onestream

SensibleAI Forecast Guide

PV9.0.0 SV402

Table of Contents

Overview

SensibleAI Forecast lets businesses build, manage, and deploy highly accurate time-series forecast models that power downstream planning processes. With SensibleAI Forecast you can perform tasks such as:

- Create thousands of daily or weekly demand planning in forecasts across products and locations.
- Capture business intuition such as promotions, events and external factors in forecasts.
- Unify and align demand plans with driver-based sales, material costs, inventory, and labor plans across Profit and Loss, Balance Sheet and Cash Flow.
- Manage the data quality and pipelines required for modeling.
- Monitor model health and performance over time.
- Increase budget, plan and forecast confidence using drill-back and testing capabilities.

Setup and Installation

This section contains important details related to the planning, configuring, and installation of your solution. Before you install the solution, familiarize yourself with these details.

See also: OneStream Solution Modification Considerations

Dependencies

Component	Description
OneStream 9.0.0 or later	Minimum OneStream Platform version required to install this version of SensibleAI Forecast.
Xperiflow 4.0.2 or later	Minimum version required to install this version of SensibleAI Forecast.
Xperiflow Business Rules(XBR)	External API client library to allow FOR to interface with the Xperiflow Engine. The required version of XBR is packaged with all FOR versions.
Xperiflow Administration Tools SV200 (XAT)	Minimum solution version required to establish permissions for projects in FOR. Required to allow project creation.

Set Up SensibleAI Forecast

Setting up SensibleAI Forecast is a multi-step process. You must complete a separate contract outside of the standard OneStream application environment. Contact your OneStream account representative before proceeding with the next steps.

1. Enter a support ticket to have SensibleAI Forecast installed in your OneStream environment.
2. After the OneStream support team ensures that the proper contract is in place, they send a link to download the SensibleAI Forecast solution and a meeting request to complete the setup.
3. The OneStream support team installs the SensibleAI Forecast data science engine in your OneStream environment and then walks you through the process of installing the SensibleAI Forecast solution.
4. After both components are installed, the OneStream support team helps you test that SensibleAI Forecast is set up correctly and functioning properly.

> **NOTE:** Upon completion of the initial onboarding and set up process, additional upgrades and releases can be found on the OneStream Solution Exchange. Limited availability to OneStream environments enabled with AI Services.

Settings

To access the global options page, click **Settings** on the bottom of the side navigation bar.

> **NOTE:** Only power users can open the **Settings** page.

Global options include:

- Solution Information
- Engine Information
- Global Settings
- Thresholds
- Solution Setup
- Uninstall

Solution Information

Provides the following information for the SensibleAI Forecast solution:

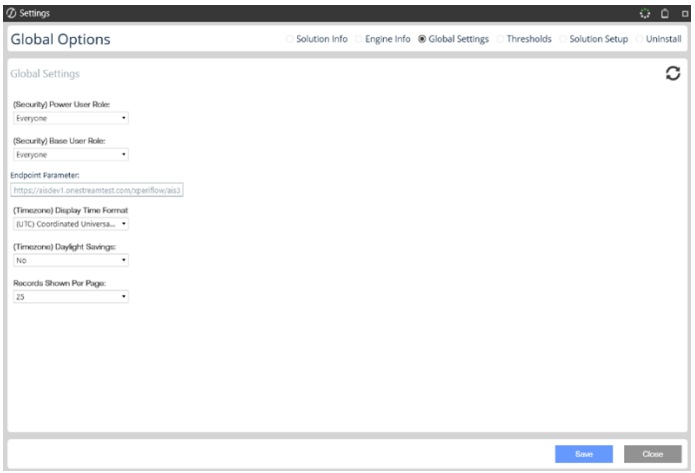
- Solution version
- Installed engine version
- Base engine version

Engine Information

Provides the following information about the Xperiflow Engine:

- **Engine Configurations** show current limits set on the Xperiflow Engine

Global Settings



(Security) Power User Role: Select users who can build and deploy models and access the global settings content. Default is Administrators.

(Security) Base User Role: Select users for this role. Default is Administrators. These users can look at models already created.

Endpoint Parameter: Predefined endpoint to access application. Do not make changes to this value unless instructed to do so.

(Timezone) Display Time Format: The time zone of the times shown in SensibleAI Forecast. The Display Time Format does not modify times for or relating to source and predicted data.

> **NOTE:** This time is not the time that is used for entries in the various log files.

(Timezone) Daylight Savings: Indicates whether the selected time zone includes daylight saving time.

Records Show Per Page: Default amount of records shown in grids that are paged throughout FOR.

Thresholds

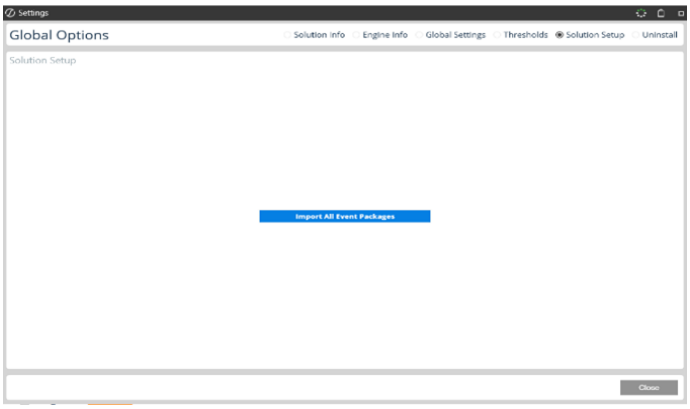
Provides the following information about the configured Build and Model Thresholds:

- **Project Thresholds** allow the user to pick a project and see the required thresholds for Feature Selection, Generation, Transformation, Grouping, and Models.
- **Default Thresholds** allow the user to configure the default required thresholds that will be used within all projects for Feature Selection, Generation, Transformation, Grouping, and Models.

Solution Setup

Import All Event Packages: Import the predefined event and location packages. See Configure Library - Events.

When you upgrade releases of SensibleAI Forecast, you can re-import the predefined event packages. When this happens, Solution Setup in the Settings displays the following:



Click **Import All Event Packages** to run the job to re-import the SensibleAI Forecast event packages.

Uninstall

You can uninstall the SensibleAI Forecast User Interface or the entire solution. If performed as part of an upgrade, any modifications performed on standard SensibleAI Forecast objects are removed. There are two uninstall options:

- **Uninstall UI** removes SensibleAI Forecast, including related dashboards and business rules but leaves the database and related tables in place.

  Choose this option if you want to accept a SensibleAI Forecast update without removing data tables.

  The SensibleAI Forecast Release Notes indicate if an over install is supported.

- **Uninstall Full** removes all related data tables, data, SensibleAI Forecast Dashboards, and Business Rules.

  Choose this option to completely remove SensibleAI Forecast or to perform an upgrade that is so significant in its changes to the data tables that this method is required.

> **NOTE:** Neither an Uninstall UI or Uninstall Full removes your SensibleAI Forecast projects. Uninstall Full only removes the stored Global Settings (Endpoint, Security, Time Format). Projects are not lost during either Uninstall.

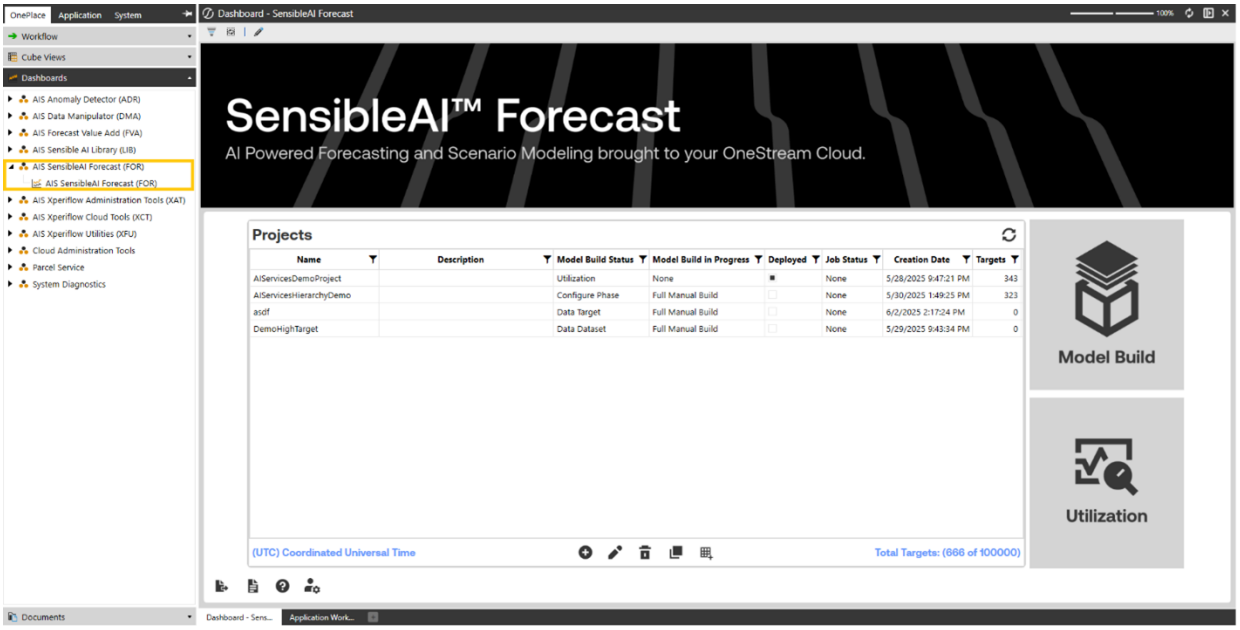
> **CAUTION:** Uninstall procedures are irreversible.

Navigate in SensibleAI Forecast

The following sections describe the different ways to navigate in SensibleAI Forecast.

SensibleAI Forecast Home Page

The Home page displays when you click on theSensibleAI Forecast. This is the starting point to creating and managing your Sensible Machine Learning projects.dashboard in Community Solution.

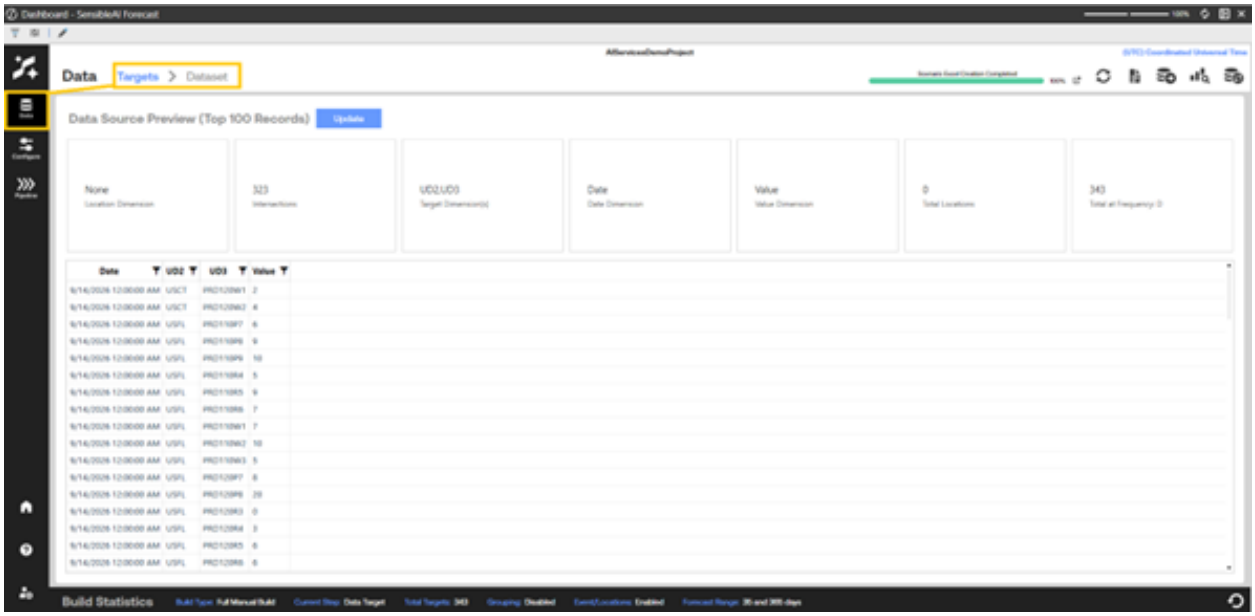


Use the **Home** page to:

- View project and build status. See Create a Model Build Project.
- Add , delete , update , and copy projects, and refresh the list of available projects.
- Update your data source for existing projects. See Update a Target or Feature Data Source.
- Manage existing projects through the Model Build and Utilization phases. See Model Build Phase and Utilization Phase.
- View and extract System Logging Tables . See View System Logging Tables.
- View the AI Services log . See View AI Services Activity.
- Access this User Guide .
- Configure global settings . See Settings.

Navigate SensibleAI Forecast Pages

When you are in Model Build or Utilization, the left side navigation includes different sections and the top left navigation shows the pages available in the selected section.



Toolbar Icons

Each page in the Model Build and Utilization phases includes a set of buttons at the top right of the page that provide additional navigation, project update, settings, or analysis functions.

Icon	Description
	Shows status of most recently started non-queued or scheduled job for the currentSensibleAI Forecast project. Click to see additional execution details of the current job in the **Job Progress** dialog box. See View Job Progress.
Refresh	Refresh and update the current SensibleAI Forecast page.

Icon	Description
 AI Services Log	Opens the AI Services log where you can review job errors, job activity, and tasks that have run or are currently running for specific jobs. See View AI Services Activity.
 Home	Navigate to the project **Home** page.
 Explore Targets and Features	View and run exploratory data to analyze each target and feature in the data set. Only available in the Model Build phase after completing the Dataset step of the Model Build. See Explore Target and Feature Data.
 Show Data Update	Opens a page where you can update target and feature data sets. This is only available after completing the Dataset step of the Model Build. See Update a Target or Feature Data Source.
 Consumption Groups	Opens the **Consumption Groups Dialog** box, which lets you create, delete, and export consumption groups. Only available after completing the Dataset step of the Model Build. See Manage Consumption Groups.

Chart and Table Toolbar Buttons

Charts and tables in the Model Build and Utilization phases include toolbar buttons for inspecting and exporting data and maximizing and minimizing the table or chart.

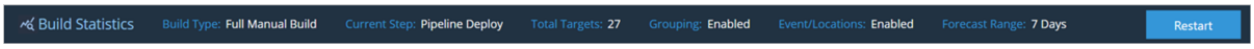
Icon	Description
Inspect	Drills into the data to view it in aggregated or raw form. All data inspect windows provide a Print Preview option from which you can save or print the data.
Export to	Exports data in the chart or table to Print Preview, PDF, Image, or Excel
Maximize	Maximizes the chart or table in the workspace.
Minimize	Minimizes the chart or table.

Review Build Statistics and Restart a Project Build

Build Statistics display at the bottom of each SensibleAI Forecast page and provide the project status and other details specific to the current page.

Model Build Phase Build Statistics

Build statistics provide you with a quick summary of information about the model in the Model Build phase.



The Build Statistics in the Model Build phase include:

Build Type: The type of build, such as Full Manual Build.

Current Step: The project build's current model build section, such as Configure Phase.

Total Targets: Total number of targets for the current project build.

Grouping: Indicates if grouping is being used in the project build.

Event/Locations: Indicates if events and locations can be used in the project build.

Forecast Range: The number of days, weeks or months to forecast forward. The default is seven days, or three months, or four weeks. Changes made on the **Forecast** page in the Model Build section are reflected here.

Restart a Project Build

You can restart a SensibleAI Forecast project using the **Build Statistics** pane. You should only restart a project if a mistake is made during the Model Build phase that would require too much effort to fix or to create another project.

When running a restart, you can redo these model building steps:

- Change the data sets used for the modeling project.
- Modify target and feature data set configurations, and groupings.
- Add new locations and events or reconfigure existing location and event data, reinstall event packages.
- Remap events and locations to targets.
- Reset the project's forecast range.
- Rerun a pipeline.

> **IMPORTANT:** A restart project job cannot be canceled once it starts.

To restart a project build for the current project:

1. Click **Restart** in the **Build Statistics** pane.
2. In the **Restart** dialog box, select a restart option:

   Restart: This reverts the project back to the Data section of the Model Build phase, prior to the **Dataset** page.

   Restart Job: Lets you begin a job based on a specified checkpoint in the project. Select the job from the list, then enter the ID of the selected job in the text box that displays. Selecting this option navigates you to the **Home** page

   > **NOTE:** Restarting using the Restart Job option can cause a loss of data such as predictions and configurations collected after the specified checkpoint.

3. Click **Confirm** to begin the restart job, then click **OK** in the message box that displays to close the **Restart** dialog box. For the **Restart Job** option, this navigates you to the **Home** page. For the **Restart** job option, this navigates you to the Data section, prior to the **Dataset** page.

Delete a Project Build

You can delete a project build using the **Build Statistics** pane. This functionality is available if the build is a rebuild. You should only delete a project build if a mistake has been made or if the rebuild is no longer required.

> **NOTE:** This only deletes the project build that is being rebuilt. The project's deployed builds and the project itself are not deleted.

To delete a project build:

1. Click **Restart** in the **Build Statistics** pane.
2. In the **Restart** dialog box, select **Delete** as the option.
3. Click **Confirm**.

SensibleAI Forecast deletes the project and returns to the**Home** page.

Restart a Job

You can also restart a job from a Job Checkpoint using the **Build Statistics** pane. This lets you select a recent checkpoint of a job (such as Data, Configuration, Pipeline, or Auto-Rebuild) and revert to that point in the model build.

> **IMPORTANT:** This capability is an advanced feature and should be used only under known circumstances. Restarting a Job can result in the loss of historic job information including but not limited to task, job, and prediction information and records. For this reason, this functionality should be reserved for known circumstances.

To restart a job:

1. Click **Restart** in the Build Statistics pane.
2. Select **Restart Job** as the restart option.
3. Select the job checkpoint you would like to restart. Click **Confirm**.

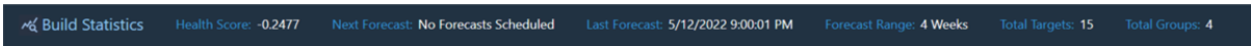
The full project now has the exact settings, configurations, and statuses that were present at the time of the selected checkpoint.

> **NOTE:** You are not able to re-enter the project in either Model Build or Utilization until the restart job is complete. The Restart job cannot be reverted or canceled.

> **IMPORTANT:** Once the job has been reverted, all previously available checkpoints no longer exist.

Utilization Phase Build Statistics

Build statistics provide you with a quick summary of information about the model in Utilization phase.



The Build Statistics in the Utilization phase include:

Health Score: Ranges from -1 to 1 indicating improvement or degradation in the model's predictive accuracy. The Health Score shows the change of Evaluation Metric Scores with each new prediction run, averaged.

Next Forecast: Date and time of next forecast. If no forecasts are scheduled, this shows **No Forecasts Scheduled**.

Last Forecast: Date and time of the last forecast.

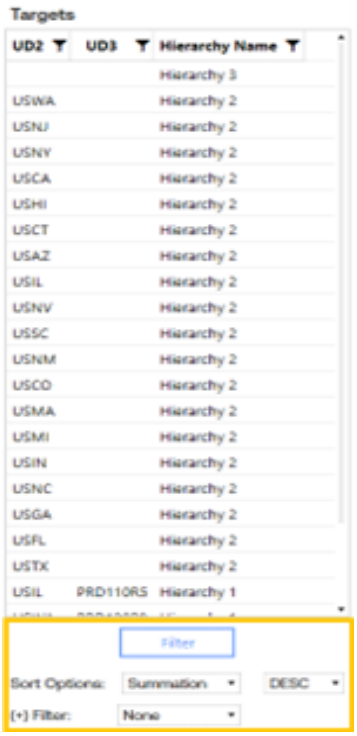
Forecast Range: The number of days, weeks or months to forecast forward.

Total Targets: Total number of targets.

Total Groups: Total number of groups.

Sort Data in Tables

Tables for targets and features may contain a lot of entries. You can filter the displayed entries by changing the sort options and series type. You can also move through specific pages to locate entries.



> **TIP:** Sort options are specific to the current page. The sort options displayed in the previous graphic are an example from the Data **Explore** page.

Sort Option: Sort by specific column in ascending (ASC) or descending (DESC) order.

Series Type: Options are targets or source features. Only available on the **Data Explore** page.

Data Type: Numeric. Only available on the **Data Explore** page.

(+) Filter: Optional. Use to filter results based on a specific column.

Filter Type: Available if **Filter** is selected. Method of filtering (equals, not equals, or contains).

Input Text Box: Available if **Filter** is selected. Select the value to apply for the filter type.

Page Number: Click a page number to go to that page.

Date Range Sliders for Line Charts

Line charts provide a date range slider that lets you drag and drop to adjust the start and end dates for the content visualized in the chart.

View AI Services Activity

The AI Services Activity Log provides critical insight into:

- Job traffic and activity.
- All tasks run for a job.
- The completion status of all SensibleAI Forecast jobs.
- Why a job may not have completed successfully.

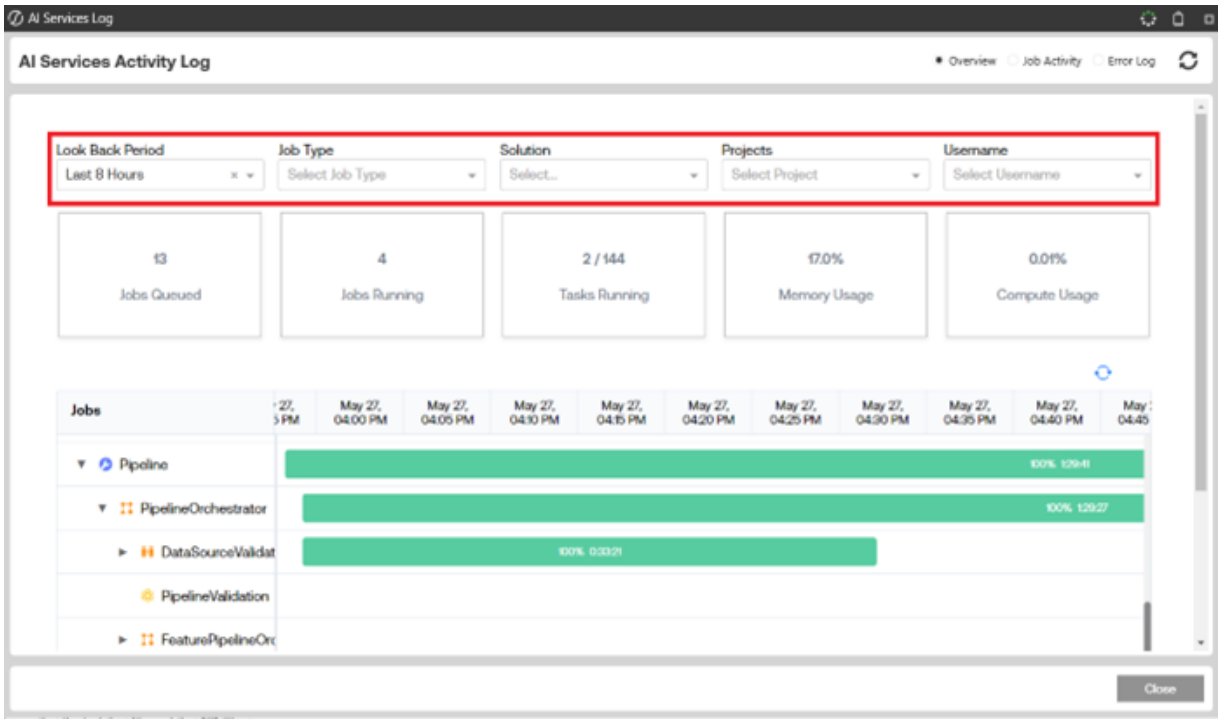
You can review this information at any time in the project creation process or during model build or utilization phases.

Access the Activity Logs

To open the **AI Services Log**, click  at the top of any SensibleAI Forecast page. This button is also available on the SensibleAI Forecast **Home** page.

The log displays an overview, the job activity, and the error log for SensibleAI Forecast. In the overview section, you can view all jobs that are running or have run. You also have the ability to narrow down jobs based on time periods, job type, solution, projects, and username. This section includes higher level statistics regarding currently running jobs and information on environment resources being used that can be cross referenced when running new jobs. In the Jobs pane, each job can be expanded to show all tasks and their status.

Overview Section



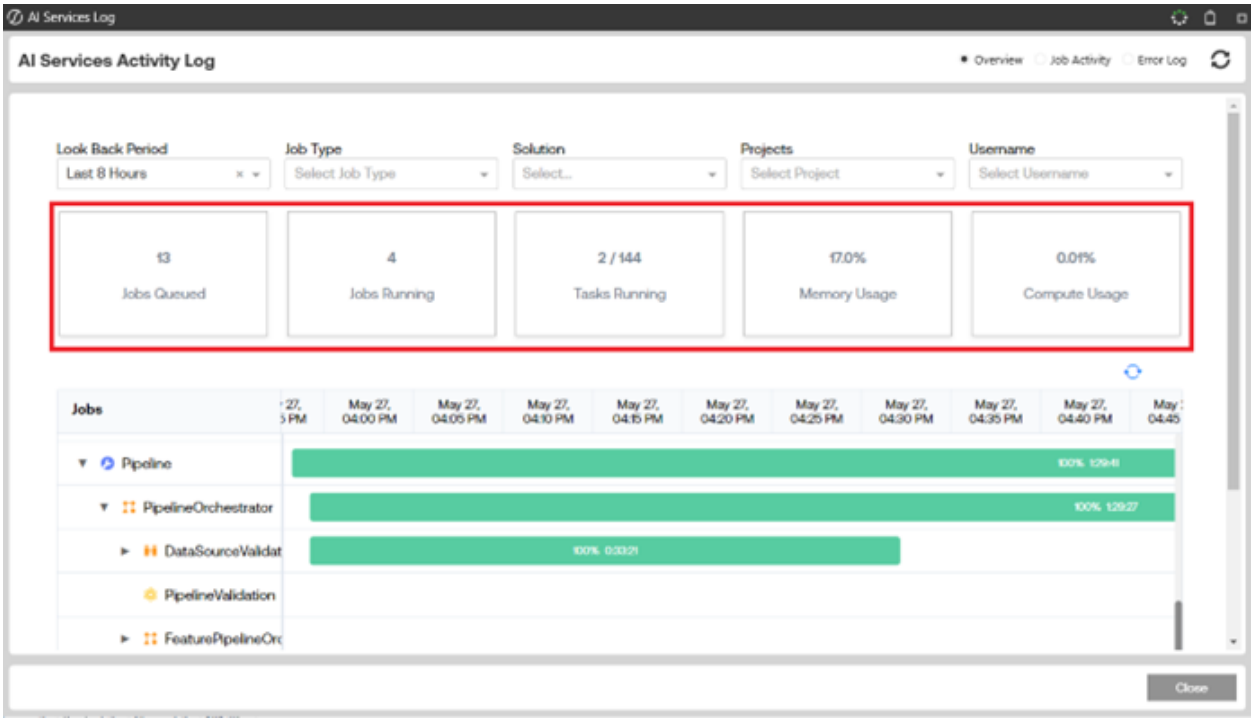
Look Back Period: Filters jobs based on the specified period of time to look back in.

Job Type: The job type that was run.

Solution: The name of the application where the job completed.

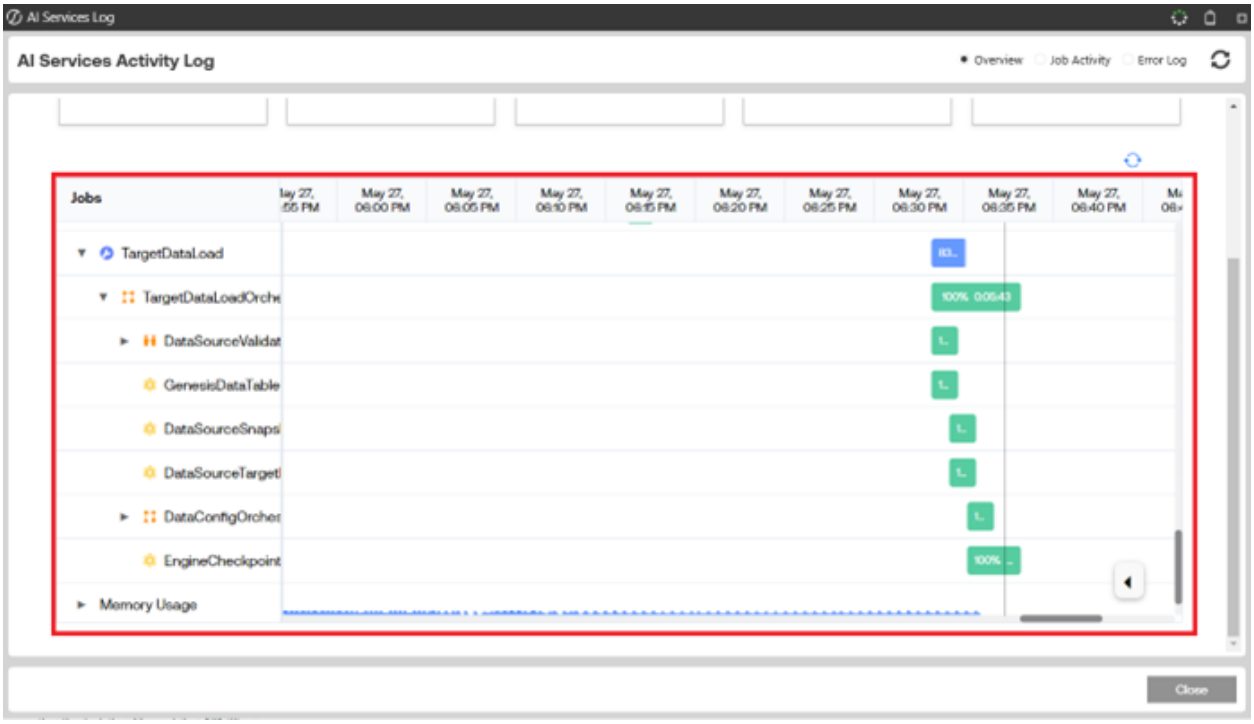
Projects: Project name given to the SensibleAI Forecast project when it was created.

Username: Username of the user that created the SensibleAI Forecast project.



Statistics Pane: Statistics about the engine resources being used to run projects/tasks.

- **Jobs Queued**: How many jobs in the environment are currently waiting in the queue to execute.
- **Jobs Running**: How many jobs in the environment are currently running.
- **Tasks Running**: How many tasks are running across the environment.
- **Memory Usage**: For the current environment, how much memory is currently being allocated towards running jobs or tasks (displayed as a percentage of possible memory).
- **Compute Usage**: For the current environment, how much compute is currently being allocated towards running jobs or tasks (displayed as a percentage of possible compute).



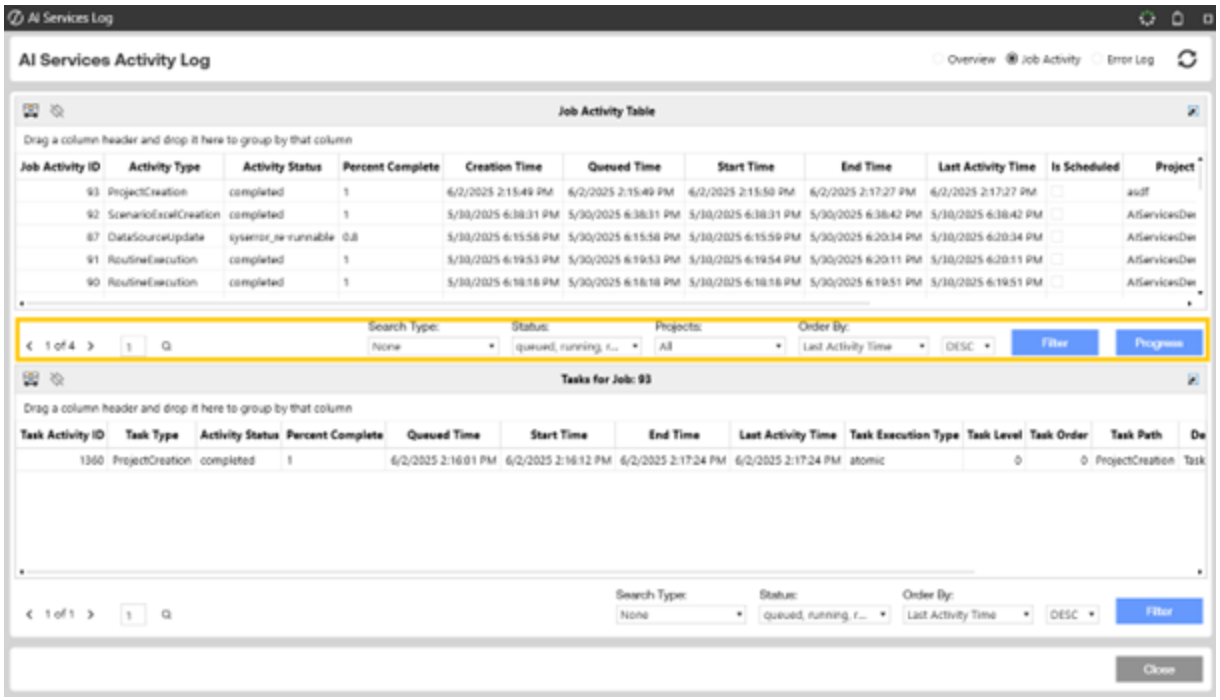
Jobs Pane: Shows jobs that have been run and status of jobs that are currently running.

- **Collapsible Jobs**: All tasks pertaining to a job are encapsulated in the overarching job that was run.
- **Time Slicing**: In the bottom right corner, users can expand the arrow to slice time periods anywhere from 5 minutes to 6 hours.
- **Color Coordinated Execution Status**: Displayed to show a certain color depending on the status of the job.
    - **Blue**: Currently Running.
    - **Green**: Successfully Completed.
    - **Red**: Job Failed.
- **Job Coordinated Icons**: Each icon is mapped to a type of job.

- : **Job.**
- : **Routine**.
- : **Atomic Task**.
- : **Sequential Task**.
- : **Parallel Task**.

• **Memory Usage**: Running graph of current memory usage over time.

• **Compute Usage**: Running graph of current compute usage over time.

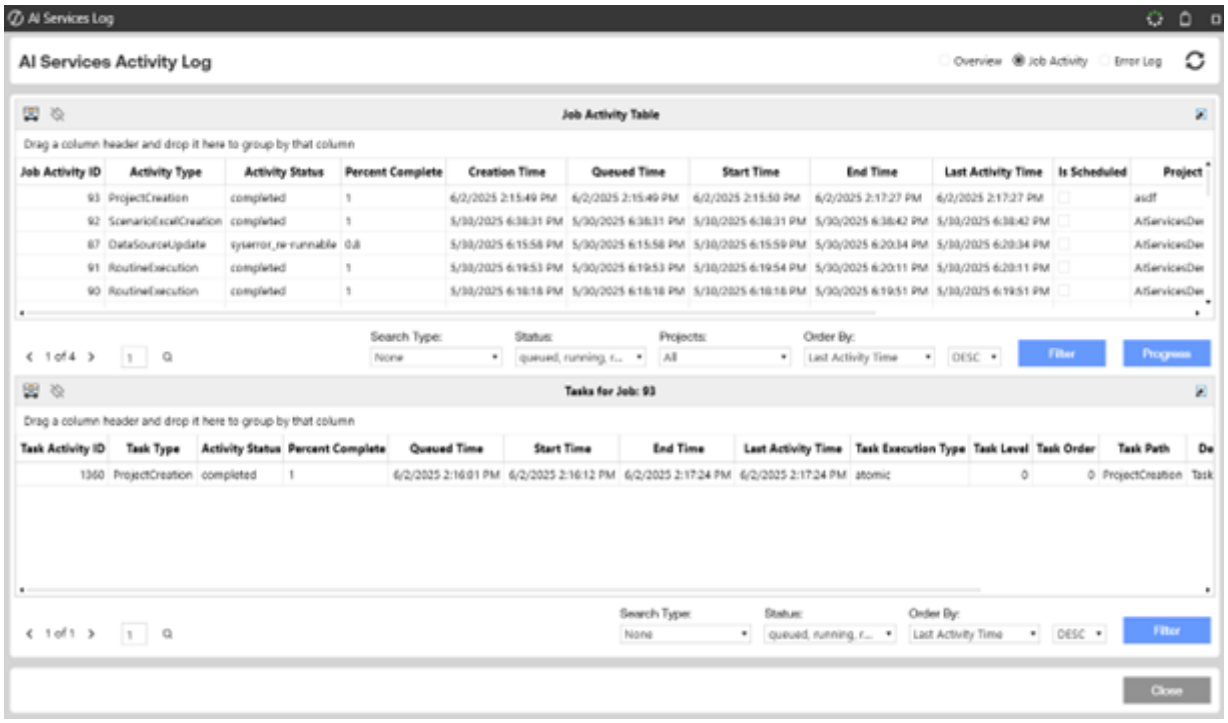


Monitor Job Activity

You can order, sort, and filter records for all views. To filter the logs, select a search type from the drop-down menu. After selecting a search type, a Search Method and Filter Value display for both jobs and tasks. Edit the Search method if needed and input a filter value. Click Filter once these settings have been selected. Select a job and click Progress to see the selected job's progress in the Job Progress dialog box, which shows the job's current activity status.

SensibleAI Forecast tracks all jobs and each job's tasks. Use the Job Activity Table in the

AI Services Activity log to see general job traffic and job details, completion status, queue order of jobs set to run immediately or on a schedule, and other job and task details.



The **Job Activity** page has a **Job Activity Table** pane and a **Tasks** pane. Both panes list activity in the OneStream grid, so items in the grid can be ordered, sorted by any column, and filtered. All items in a grid can be sorted using the sort functionality.

> **TIP:** Use the scroll bars in either pane to see all columns.

Each item in the Job Activity Table represents a job that has run in SensibleAI Forecast. Select a job to see the job's tasks in the **Tasks** pane. Each job includes the following:

Job Activity ID: Unique ID given to the job when first queued.

Activity Type: Part of the project or page in SensibleAI Forecast where the job was generated.

Activity Status: The current status of the job.

- **queued**: Indicates if the job or task is queued to run. If a job is scheduled, it is queued with a QueuedTime when the job is scheduled to run.
- **running**: The job or task is currently running.
- **running_subtasks**: An orchestrator task is currently running subtasks.
- **completed**: The job or task completed successfully.
- **syserror/syscancelled**:The job or task failed due to a system failure or the system canceling the job or task.
- **usercancelled**: The job or task was cancelled by a user.
- **syserror_re-runnable**: The job or task failed due to a data validation error and can be re-run once the issue has been resolved.
- **worker_queued**: The task is currently in the queue to be picked up by a worker. This is only an activity status for atomic tasks and not jobs.

Percent Complete: Completion percentage of all the job's tasks.

Creation Time, **Queued Time**: Time that the job was created and moved to the job execution queue.

Start Time, **End Time**: Job's start and end dates and times.

Last Activity Time: Time that the most recent job activity completed. When a job successfully completes, this is the same as the End Time.

Is Scheduled: Selected indicates that the job was originally scheduled or scheduled to run at a future date and time.

Project Name: Name given to the project when it was created. <No Project> displays if the job did not execute for a specific SensibleAI Forecast project. <Deleted Project> displays if the job run for a specific SensibleAI Forecast project has been deleted.

App Name: Application responsible for running the job.

Total Tasks, Completed Tasks: Number of tasks run by the job and the number of tasks that ran successfully.

Server Name: The name of the server where the job completed.

Project ID: ID given to the SensibleAI Forecast project when it was created.

Information displayed for tasks include:

Task Activity ID: Unique identifier assigned to the task when it is created.

Activity Status: The current status of the job.

- **queued**: Indicates if the job or task is queued to run. If a job is scheduled, it is queued with a QueuedTime when the job is scheduled to run.
- **running**: The job or task is currently running.
- **running_subtasks**: An orchestrator task is currently running subtasks.
- **completed**: The job or task completed successfully.

- **syserror/syscancelled**:The job or task failed due to a system failure or the system canceling the job or task.
- **syserror_re-runnable**: The job or task failed due to a data validation error and can be re-run once the issue is resolved.
- **data_validation_error**: The job or task failed due to a data validation error.
- **worker_queued**: The task is currently in the queue to be picked up by a worker. This is only an activity status for atomic tasks and not jobs.

Percent Complete: Completion percentage of all the job's tasks.

Queued Time, **Start Time**, **End Time**, **Last Activity Time**, **Percent Complete**: These columns have the same type of information as they do for jobs, but these are for a selected job's tasks.

Task Execution Type: Indicates whether the task is synthetic, atomic, sequential, or parallel.

Task Level, **Task Order**: Order of a job within the displayed task level.

Task Path: The routine within the job where the task runs.

- **Synthetic**: A task that runs within an atomic task. It is only included in the AI Services Task Activity log if the task fails.
- **Atomic**: One which has a simple, self-contained definition (for example, one that is not described in terms of other workflow tasks) and only one instance of the task runs when it is initiated.
- **Sequential**: Tasks are distributed across different processors and run in a specific order.
- **Parallel**: Task is distributed with other tasks across different processors and concurrently run by processes.

Process ID: Unique identifier assigned to the process in which the job was run.

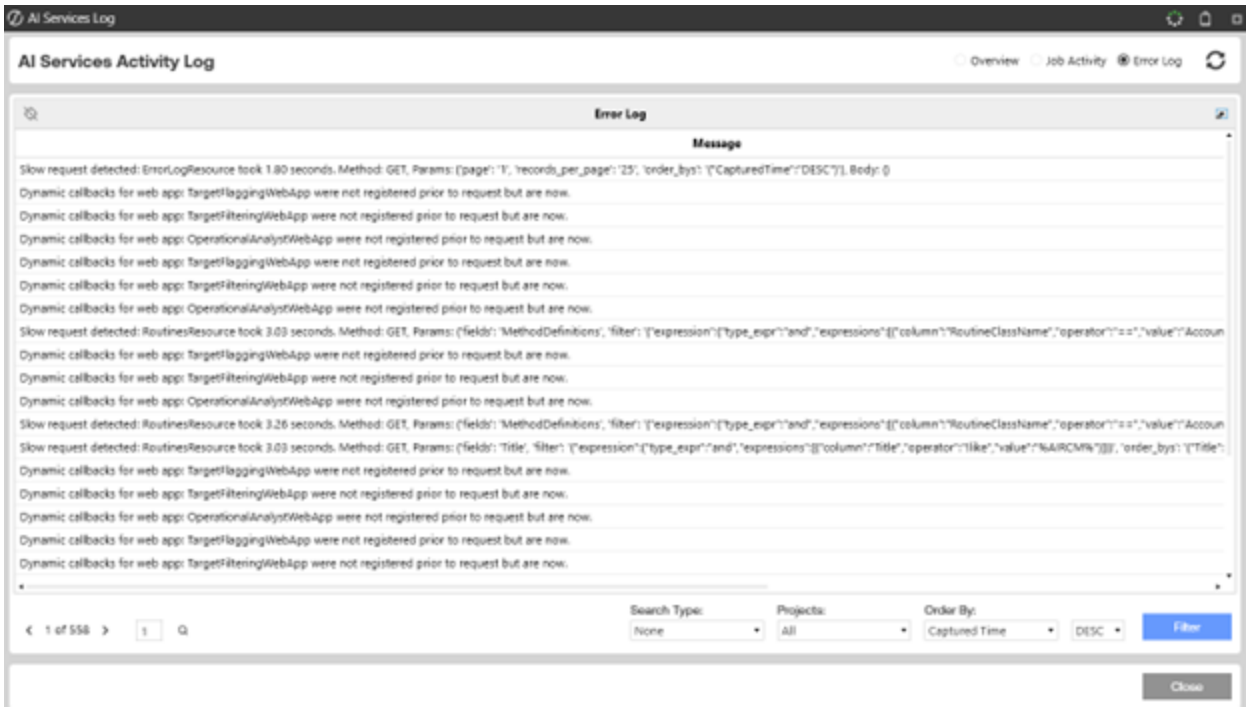
Server Name: The name of the server that completed the job.

Parent Task Activity ID: Unique identifier assigned to the task's parent when it is created.

Job Activity ID: Unique identifier assigned to the task's job when it is created.

Review Job Errors

Use the Error Log for information on why a job did not run properly or run to completion, and to aid in error diagnosis and resolution. Types of errors include job execution errors for a specific project, SensibleAI Forecast updates, and login and other connection failures.



Each entry includes the following information:

Message: Message associated with the error.

Captured Time: The time that the error occurred.

Log Level: Categorizes the severity of the error. Though warnings are logged, they do not necessarily stop a job from running.

Log Source: XperiFlow module where the error occurred.

Project Name: The name of the project that caused the error. <No Project> displays if the error did not occur for a specific SensibleAI Forecast project.

Error Category: General type of error that occurred. NA displays for errors not associated with a specific category.

Error Info: Similar to the information in the Message column, this is detailed information about what occurred to cause the error. For SQL statements that cause an error, this can include details about where in the statement the failure occurred. This information can help technical support with error diagnosis and resolution.

Server Name: The name of the server where the error occurred. Useful when a job is parallelized with multiple servers.

Process Name, Process ID: Unique identifiers that point to the specific process within a server where the error occurred.

Thread Name, Thread ID: Unique identifiers that point to the thread in which the error occurred.

Project ID: Unique identifier given to the project when the project was created. Zeros in this column indicate the error is not associated with a specific project.

Job Activity ID, Task Activity ID: Unique identifiers for the job and task within the job that encountered the error. Zeros in this column indicate the error was not associated with a specific SensibleAI Forecast task or job.

Log Level Number: The number assigned to the type of error. For example, a warning is level 30, an error is level 40.

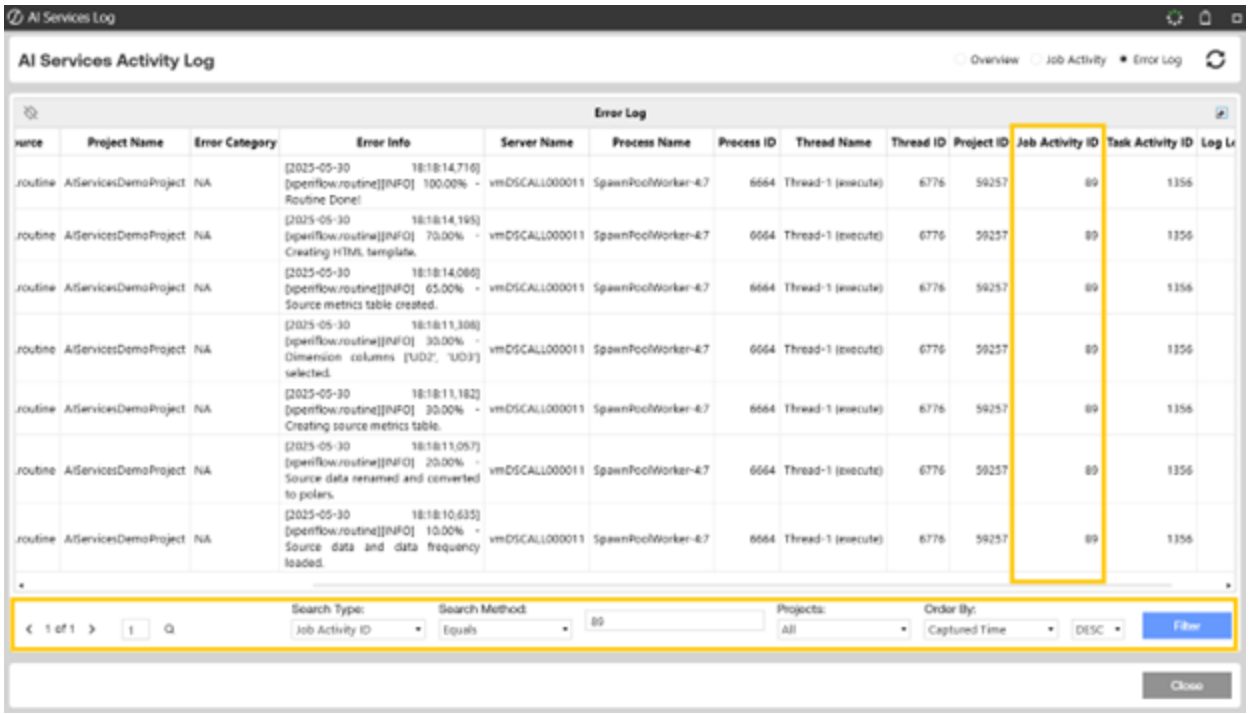
> **NOTE:** Jobs that are not associated with a specific project show zeros for the Job ID, Task ID, and Project ID.

Find Specific Jobs, Tasks, or Errors

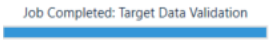
The AI Services Log also lets you search for specific jobs, tasks within jobs, and errors. This is useful when you what to find a specific item or items (jobs, tasks, or errors) within a large number of items, or your Job Activity or Error log has multiple pages. Use the search along with table sorting to speed your search of items in the AI Services Activity Log.

Select a search type, then type an ID for the item you want to search for and click **Filter**. In the Job Activity, you can find Jobs by the Job Activity ID or Project ID to find all jobs in a project. Search by Task Activity ID to find specific tasks.

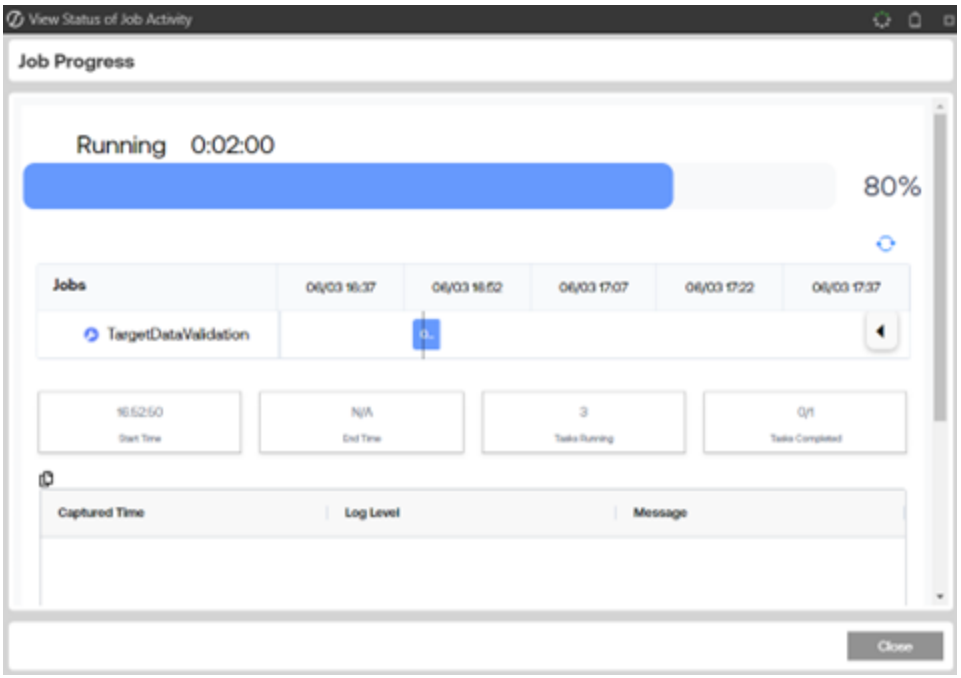
The Error log lets you find specific errors by Job Activity ID or Task Activity ID, or Project ID. The ID you enter must match the ID exactly, though lowercase letters in the search will match. The following graphic shows the search in the Error Log.



View Job Progress

You can view job progress after you have started at least one job in a SensibleAI Forecast project. Click the Open **Job Progress Bar** 

in the SensibleAI Forecast toolbar to access the **Job Progress** dialog box.



This dialog box shows the most recently active job that is not in a queued state for the current SensibleAI Forecast project. It shows a condensed view of the AI Services Log.

The following information is included in the **Job Progress** dialog box.

- **Start Time**: Time that the job started.
- **End Time**: Time that the job ended.
- **Tasks Completed**: The number of tasks that have been completed for the given job.

- **Tasks Running**: The number of tasks that are currently running for the job.
- **Progress Bar**: How close the job is to completion and how long it has been running.
- **Logging Grid**: Error log entries for the job. These may be errors, warnings, or other relevant information.

Explore Target and Feature Data Sources

After you have processed and merged the data sets for your SensibleAI Forecast project, data analysis for each target becomes available. Click the **Explore Targets and Features**  icon in the [%=Solution Names.prodname-SensibleAI Forecast%] toolbar to access the page.

> **TIP:** The **Explore Targets and Features** page is only available during the Model Build phase.

The Data Elements pane on the **Explore Targets and Features** page shows the data elements represented in the information on the page. The information on the page changes depending on whether you are viewing targets or features. Target information displays by default.

To change the data elements represented on the page, select **Features** or **Targets** in the Series Type field click **Filter**.

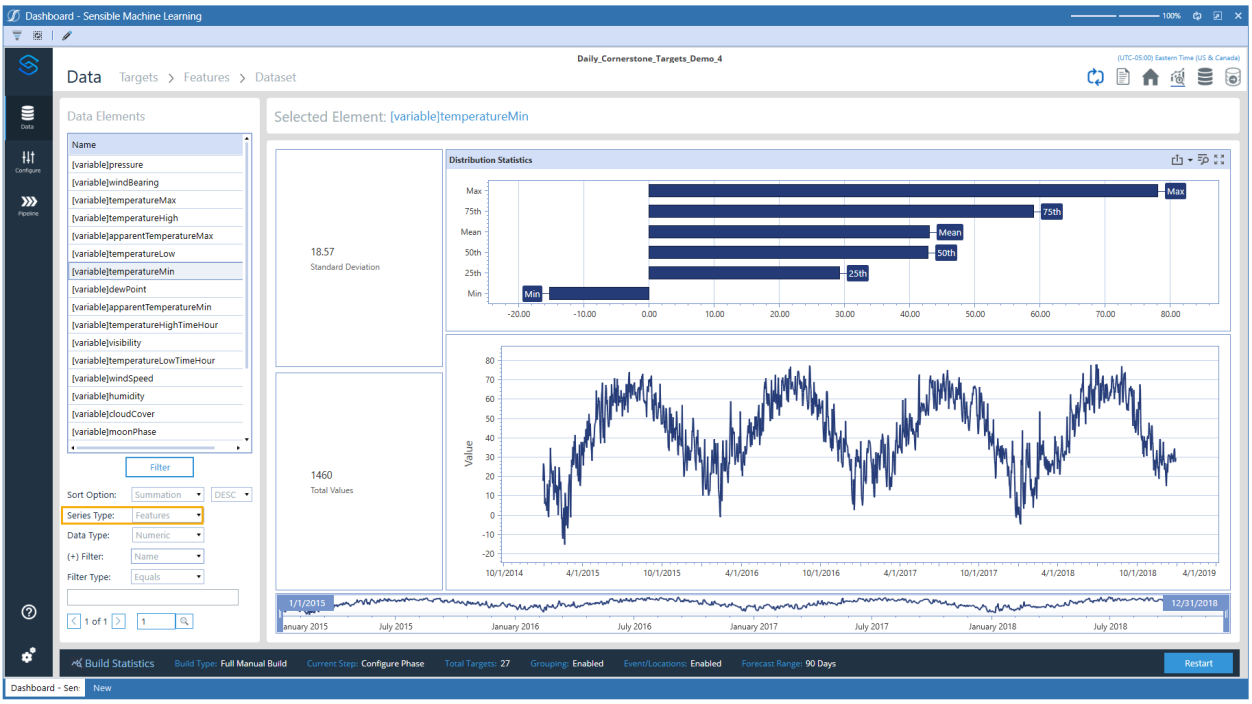
Target Data

Each target in the data source displays with the following information:

- **Random**: Indicates if a data set does not contain any recognizable patterns. This signifies that it may be difficult for models to learn from historical data.
- **Seasonality**: Indicates if a data set contains any seasonal patterns or cycles that repeat over a period of time. If a data set has seasonality, models can use it to increase predictive accuracy.
- **Standard Deviation**: Indicates the standard deviation of the target.
- **Stationarity**: Indicates if the statistical properties of the data set do not change over time.

- **Trend Coefficient**: This is the result of a Mann Kendall test which produces a value between -1 and 1. A coefficient greater than zero indicates the statistical significance of a positive trend. A coefficient less than zero indicates the statistical significance of a negative trend.
- **Partial Autocorrelation Function (PACF) Plot**: Demonstrates correlation (-1 to 1) of values based on the time increment between them. For example, a daily-level data set with a PACF score of 0.5 at an x-axis point of 7 signals that, on average, today's value has a correlation coefficient of 0.5 with the value of 7 days prior.
- **Distribution Statistics**: Bar chart representing these values: Maximum, 75th percentile, Mean, 50th percentile, 25th percentile, and Minimum.
- **Data Completeness**: Bar chart representing how many zero or null values are found for each target of the data set.
- **Historical Actuals Plot**: Plots all the historical actuals from the data.

Feature Data



Each feature in the data source displays with the following information:

- **Standard Deviation**: Indicates the standard deviation of the feature.
- **Total Values**: The number of unique values created by feature data.
- **Distribution Statistics**: Bar chart representing these values: Maximum, 75th percentile, Mean, 50th percentile, 25th percentile, and Minimum.
- **Historical Actuals Plot**: Plots all the historical actuals from the data.

Update a Target or Feature Data Source

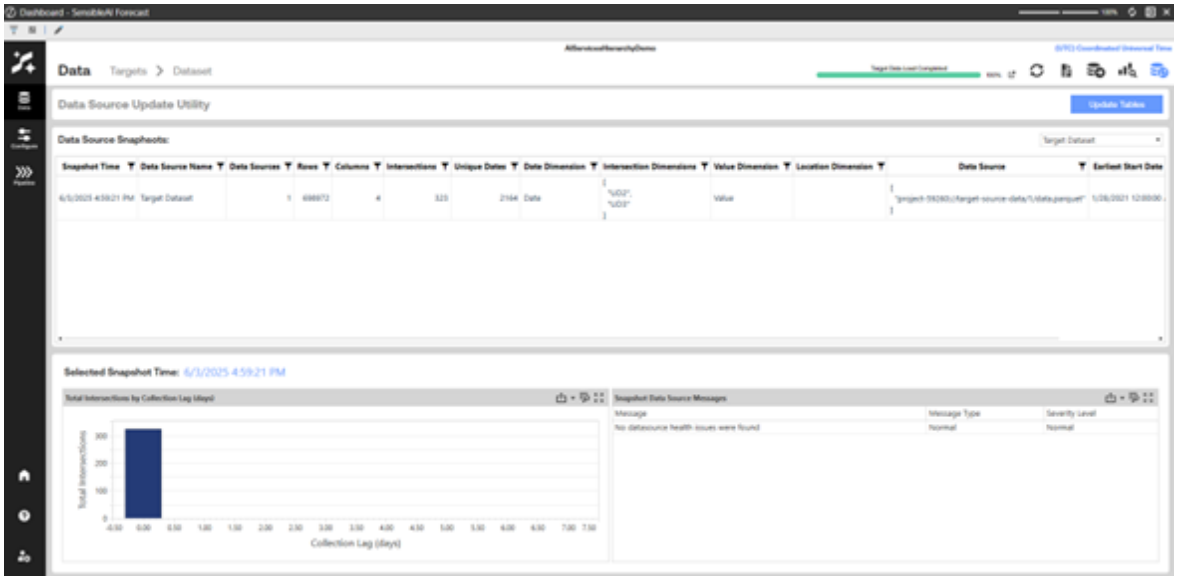
Use the **Data Source Update** page to update data tables or change the data source connection for a target or feature data source. This can be done at any time after specifying data targets and verifying your data source using the **Data** > **Dataset** page. However, it is especially useful during the Utilization phase when new data may be available in a new table that must be added to the data source. The **Data Source Update Utility** page also displays statistics about each data source.

> **TIP:** The process of updating a data source is similar to the processes used to specify targets and define the data set and to specify data features and is only available after you specify targets and features. However, instead of selecting data tables for the first time in the modeling process, you are changing the tables used.

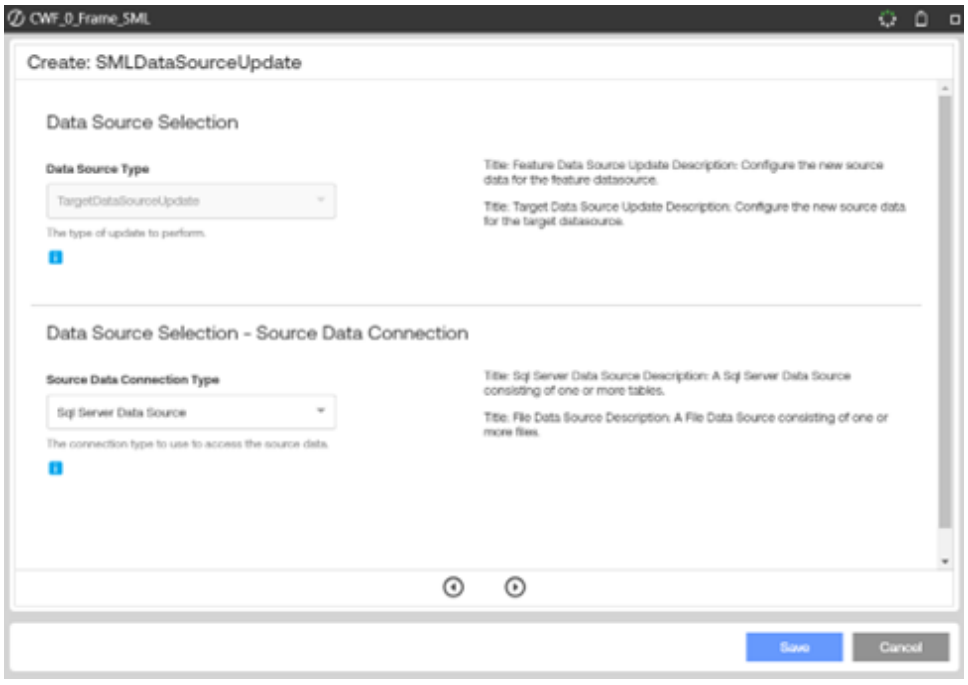
> **NOTE:** The dimensions of a data source cannot be changed. Also, you must specify targets and define the data set and verify your data sets before you can use the Data Update utility.

Update a Data Source

1. At any time after specifying data targets and features and verifying your data source, click **Show Data Update** on the current page toolbar. The **Data Source Update Utility** page displays.



2. Click **Update Tables** at the top of the **Data Source Utility** page. The **Update Dataset Connection** dialog box displays, allowing the user to choose between the target or feature data sources to update. The following graphic shows the dialog box:



3. Select the Data Source Type to be either Target Data Source or Feature Data Source, then click next.
4. Select the Source Data Connection Type and click next.
5. Select the Resource Name for the Data Connection that was selected.
6. Select the actual sources that contain the data. If they are correct, click next to review your selections. Once finished, click save to submit the new data sources

Data Source Statistics

The **Data Source Update Utility** page shows numerous statistics from the selected target or feature data source.

Data Source Snapshots: These are snapshots of the data source that are created when a user uploads data, updates data, or starts a prediction. A snapshot can be selected to populate the Total Targets by Collection Lag (days) and the Snapshot Data Source Messages at the bottom of the page.

Total Targets by Collection Lag (days): This chart visualizes the latest view of the collection lag for the target data set. It shows the number of targets that have a given collection lag by the number of days. This chart updates when you run a new prediction or snapshot. The chart also updates with each data set job and an update of target data source.

Snapshot Data Source Messages: This shows any messages from the data source snapshot. For example, warnings for missing intersections or intersections that have been added since the last snapshot.

Manage Consumption Groups

Consumption groups are the connections to a data table in OneStream. Use the Consumption Groups page to manage consumption groups that are used to load model predictions and insights into tables within OneStream.

> **NOTE:** The destination Data Connection will be the same as the specified Data Source Connection for the Target Data Source

Consumption groups can be created after you have specified targets and features and verified your data sets using the **Data > Dataset** page.

Create Consumption Groups

The consumption groups you create are available when you run a new prediction.

1. Click the **Consumption Groups** icon in the SensibleAI Forecast toolbar. The **Consumption Groups** dialog box displays.
2. Click **Add a New Consumption Group** .
3. In the **Add Consumption Group** dialog box, select the type of consumption to add in the drop down box. This determines the other settings. See the Consumption Group types here.
4. Type a name for your consumption group in the Consumption Group field.
5. Type a name for the Output name in the Output Name field.

6. Make selections from these options (or subset of options based on consumption type). The options you can select include the following values:

   Export Action: The type of action this consumption group should be auto-attached to. If auto-attached, it automatically runs and exports as a part of that type of action. Examples include prediction and pipeline jobs.

   Actuals Types: The type of actuals (engine cleaned, engine uncleaned, source) to include in the consumption group.

   Target Frequency: The frequency to export data in.

   Merge Method: The type of merge to perform on data if there are multiple data points for a date.

   Models to Return: The models to return in the consumption group.

   Prediction Intervals: Indicates if prediction intervals should be included in the export.

   Extraction Type: Select **Batch** to export only the latest prediction run. Select **Time** to use Start Date Type and End Date Type fields for the earliest prediction to the latest prediction or custom time frames.

   Start Date Type: If the earliest start date or a custom beginning date should be used.

   Start Date Time, **Start Date Hour**: The beginning date and hour of the day of the consumption group (inclusive). Available if the If Start Date Type is set to **Custom**.

   End Date Type: If the latest end date or a custom end date should be used.

   End Date TimeEnd Date Hour: The end date and end hour of the day of the consumption group (inclusive). Available if the If End Date Type is set to **Custom**.

   Group Name: Type a name for the consumption group to be exported.

   Output Table Name: The name of the outputted table. This is in the same database as the target data set.

7. Click **Save**.

Export Consumption Group Data

You can manually export all data defined in a consumption group to the associated new or existing OneStream table. If the table already exists, it must have the correct schema; otherwise, it does not populate the prediction data.

> **NOTE:** Groups can also be automatically exported by setting the Export Action option of a given consumption group.

To export a consumption group:

1. Select a group and then click **Export Group**

   > **NOTE:** Some consumption types may require you to select what to export data for (all deployed builds or a given build based on build time).

2. Confirm that you want to export the group by clicking **Export**.

Consumption Group Types

There are multiple types of consumption groups that can be configured to export a variety of information from SensibleAI Forecast. These exports can be used in downstream processes for a variety of applications.

The following list describes the consumption group types and shows the schema for each.

Feature Effect

Description: The feature effect consumption type contains data informing how a given feature and its values compare to a given target's actual values and prediction values. This provides insight into whether a correlation exists between certain feature values and predictions or actuals. See Appendix 4: Interpretability for details.

Schema: `Model, FeatureLowerBound, FeatureUpperBound, FeatureAvgValue, PredictionAvgValue, TargetAvgValue, FeatureName, FeatureShortName, TargetName, ConsumptionID, ProjectID, ConsumptionRunID, ConsumptionRunTime, XperimentKernelID, XperimentBuildID, XperimentSetID, BuildInfoID, [TargetDimensions]`

Feature Impact

Description: The feature impact consumption type contains data informing how much a given feature influences the model for a given target. A large FeatureImpactValue means that the feature is an important driver for the model predictions for that target. See Appendix 4: Interpretability for details.

Schema: `FeatureName, FeatureShortName, FeatureImpactValue, FeatureImpactType, TargetName, ModelName, Category, ConsumptionID, ProjectID, ConsumptionRunID, ConsumptionRunTime, XperimentKernelID, XperimentBuildID, XperimentSetID, BuildInfoID, [TargetDimensions]`

Model Forecast Backtest

Description: The model forecast backtest consumption type contains the backtest results (and possibly prediction intervals) made by models for each target from the model build phase of the application.

> **NOTE:** Not all model builds contain a backtest portion. It is dependent on the number of data points.

Schema: `Model, ModelCategory, TargetName, Value, LowerPI, UpperPI, Date, ModelRank, ConsumptionID, ProjectID, SplitID, ConsumptionRunID, ConsumptionRunTime, XperimentKernelID, XperimentBuildID, XperimentSetID, BuildInfoID, [TargetDimensions]`

Model Forecast Deployed

Description: The model forecast deployed consumption type contains the predictions (and possibly prediction intervals) made by models for each target from the utilization phase of the application.

Schema: `Model, ModelCategory, TargetName, Value, LowerPI, UpperPI, Date, ModelRank, PredictionCallID, PredictionScheduledTime, ConsumptionID, ProjectID, ConsumptionRunID, ConsumptionRunTime, ForecastStartDate, ForecastNumber, ForecastName, XperimentKernelID, XperimentBuildID, XperimentSetID, BuildInfoID, [TargetDimensions]`

Model Forecast V1

Description: The model forecast v1 consumption type is the same as the model forecast deployed consumption type but keeps the same table schema as SensibleAI Forecast versions SV103 and earlier. This consumption type is deprecated SensibleAI Forecast versions SV200 and later, and should not be used except for maintaining backwards compatibility for existing processes while they are transferred to newer consumption types

Schema: `Model, ModelCategory, TargetName, Value, Date, ModelRank, FKPredictionCallID, PredictionScheduledTime, FKConsumptionID, FKProjectID, ConsumptionRunID, ConsumptionRunTime, [TargetDimensions]`

Prediction Explanations Backtest

Description: The prediction explanations backtest consumption type contains the amount that the feature influenced (positive or negative) the prediction of a given model for a given target for a given date during the backtest in the model build section of the application. See Appendix 4: Interpretability for details.

Schema: `Date, FeatureName, FeatureShortName, PredictionExplanationValue, FeatureValue, PredictionExplanationType, TargetName, Model, ModelStage, ModelCategory, ModelITerationID, ConsumptionID, ProjectID, ConsumptionRunID, ConsumptionRunTime, XperimentKernelID, XperimentBuildID, XperimentSetID, BuildInfoID, [TargetDimensions]`

Prediction Explanations Deployed

Description: The prediction explanations deployed consumption type contains the amount that the feature influenced (positive or negative) the prediction of a given model for a given target for a given date during the forecasts in the utilization section of the application. See Appendix 4: Interpretability or details.

Schema: Date, FeatureName, FeatureShortName, PredictionExplanationValue, FeatureValue, PredictionExplanationType, TargetName, Model, ModelStage, ModelCategory, ModelITerationID, ConsumptionID, ProjectID, ConsumptionRunID, ConsumptionRunTime, PredictionCallID, PredictionScheduledTime, ForecastStartDate, ForecastNumber, ForecastName, XperimentKernelID, XperimentBuildID, XperimentSetID, BuildInfoID, [TargetDimensions]

View System Logging Tables

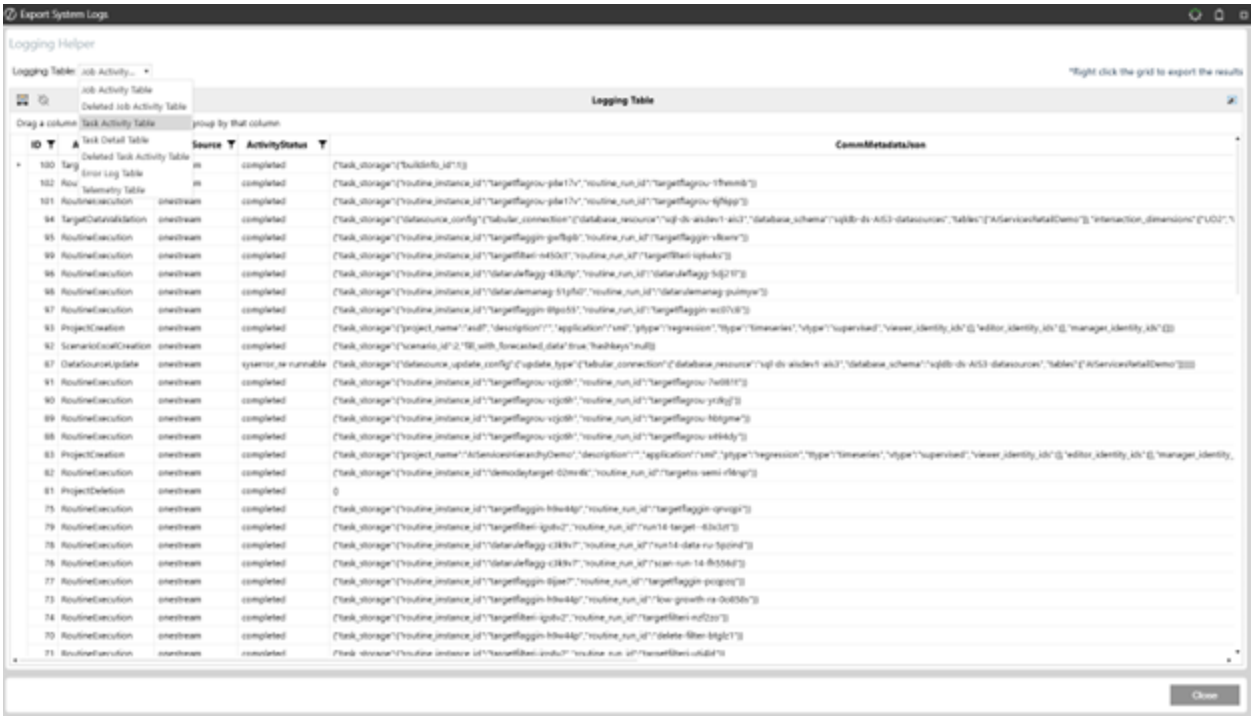
Use system logging tables to help debug any issues with a SensibleAI Forecast initial environment installation or an environment upgrade. The logging tables are meant to be used by advanced or power users.

Each table contains records ordered by a given column for each table.

These tables are not paged, so some data may be inaccessible. You can export Logging information, which retrieves all data in the selected table. To do this, select the table to display it, right-click in the table and select **Export**, then select the output type.

To quickly find a specific logging record, press **CTRL + F** to open a full text search on the selected table, then enter a string in the record.

The system logging tables are only accessible from the [%=Solution Names.prodname-SensibleAI Forecast%] Home Page. Click  on the **Home** page to view and extract system logging tables.



System Logging Tables

The following list describes each of the available logging tables.

Job Activity Table: Includes information related to jobs that have been run. Similar to the AI-services log.

Deleted Job Activity Table: Similar to the Job Activity table, but contains jobs that have been deleted from a restart job run.

Task Activity Table: Includes information related to tasks that have been run. Similar to the AI-services log.

Task Detail Table: Includes intermediate updates of what is occurring within each task.

Deleted Task Activity Table: Similar to the Task Activity table, but contains tasks that have been deleted from running a restart job.

Error Log Table: Includes information about errors or logging information that has occurred. Similar to the AI-services error log.

Telemetry Table: Contains information regarding the environment status on each machine in the environment. This includes CPU percentage, number of processes, and database connections.

Model Build Phase

The Model Build phase walks you through building a highly accurate machine learning models that are specific to the forecasting problem you are trying to solve.

> **NOTE:** You must have a business administrator or higher security role to access pages in the Model Build phase.

> **TIP:** The Consumption Groups page cannot be accessed until the **Data > Dataset** page has run the job to merge the data sets. The page can always be accessed in the Utilization phase.

Model building in SensibleAI Forecast is a three-section phase where you configure, build, and deploy machine learning or statistical models for time series forecasting in a SensibleAI Forecast project. It begins with sourcing data into SensibleAI Forecast. The data is typically sourced from an external database connection.

From there, you can add various machine learning parameters to your project. This includes parameters such as locations, events, forecast ranges, and model configurations. Most of these are defined in the Configure section.

After specifying the data and the configurations to generate, engineer, and transform the data, use the Model Build phase **Pipeline** page to run a model pipeline.

> **IMPORTANT:** You should have a thorough understanding of the data in any target or feature data source used in the machine learning or statistical model you are working with. Various steps in the Model Build and Utilization phases let you review and verify the data being used and how the data is configured for SensibleAI Forecast.

> **NOTE:** Times shown are in the specified time zone. However, custom time frames use the actual prediction data generated dates.

See Appendix 2: Use Case Example for information about models used by SensibleAI Forecast.

Create a Model Build Project

The SensibleAI Forecast Home page displays when you start the [%=Solution Names.prodname-SensibleAI Forecast%] Solution. The Project Selection grid shows each existing SensibleAI Forecast project and includes the following information for each:

Name: Name of the project.

Description: Optional description of the project. Can be added when creating or editing a project.

Creation Date: Date the project was created.

Build Status: Indicates the current model build section for each project. A value of **NONE** means the project is in the Utilization phase.

Builds in Progress: Indicates if there are builds in progress for each project. A value of **NONE** means the project is in the Utilization phase.

Is Deployed: Select to indicate that the project is deployed.

Targets: The amount of targets each project has.

You can sort each column in the grid.

Start a New Project

The first step to create a model build project is to start a new project. To do this:

1. In the Project Selection grid, click the **Add a New Project** button. The **Add Project** dialog box displays.
2. Type a name for your project in the Project Name field. The name should be descriptive enough so you can recognize the project in the list.
3. Optionally type a description for the project in the Project Description field. Descriptions are useful to differentiate multiple SensibleAI Forecast projects with similar project names.
4. Security Access (Admins Only):

- Assign different users or groups to any of the following levels of security access:
  - **Viewer**: Read-only access. Cannot modify anything within a Model Build or run any jobs.
  - **Editor**: Read/Write access. Can modify anything within a Model Build, but does not have access to higher administrative level privileges (Example: Deleting a project)
  - **Manager**: Read/Write/Delete access. Full capabilities to modify anything within a project and higher level project settings.

> **NOTE:** When assigning security roles to a project, the creator of the project will automatically be assigned a Manager role.

5. Click **Add**. A message displays in the dialog box asking you to verify running the job to create the project. To verify, click **Save** to start the project creation and view job progress.
6. When the job completes, click **Close** to close the **Project Creation** dialog box.

SensibleAI Forecast stores the project and assigns it a project ID, which you can see in the AI Services Job Activity log. All project status details are associated with the project ID.

When complete, the project displays in the Project Selection list.

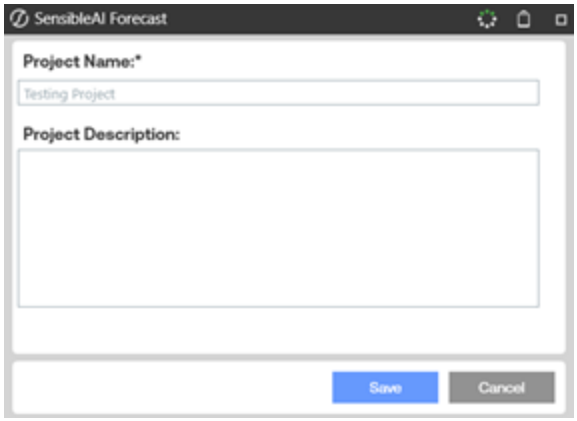
- To work with any project, click to select it, and then click the **Model Build** or **Utilization** icon.
- For a new project, click the project in the list, click **Model Build**, and then continue by specifying targets and features using the correct data sets.

> **TIP:** While working on any project, you can return to the SensibleAI Forecast Home page by clicking the **Home** button at the top of the current page.

Update Information for an Existing Project

You can change or update information for a project after it is created. To do this:

1. In the Project Selection grid, click the project you want to edit, and then click the **Update the selected Project** button. The **Update Project** dialog box displays.



2. Update the Project Name or Project Description as needed.
3. Click **Save** to update the project information.

Copy an Existing Project

You can copy a project in its entirety to add redundancy and apply different scenarios or configurations.

> **NOTE:** When copying a project with one or more predictions run or scheduled, prediction results at the time the copy is made are copied to the new project. However, the job records of any predictions running are not copied to the resulting project. These are the records that are on the **Manage > Predict** page of the original project.

1. In the Project Selection grid, click the project you want to copy, then click the **Copy the selected Project** button.
2. A message displays in the **Copy Project** dialog box to confirm running the copy project job. Verify that the project to copy is correct and type the desired name of the project that will be created as a copy. Click **Copy**.
3. A message displays to show you that the project is marked to copy. The project copy job queues and runs in a background job. You can check the status of the job in the AI Services Log.

Once the job completes, the copied project shows in the Project Selection grid, as an exact replica of the copied project with the specified project name.

Delete an Existing Project

> **NOTE:** You must have delete permissions in XAT.

You can delete a project after it is created. To do this:

1. In the Project Selection grid, click the project you want to delete, then click **Delete the selected Project** .
2. A message displays in the **Delete Project** dialog box to confirm the deletion. Type the project name in the text box to verify the deletion. Click **Delete**. You must enter the project name exactly as it is to confirm deletion.
3. A message displays to show you that the project is marked for deletion and deleted in a background job. Click **OK**.
4. After the delete project job completes, the project is removed from the Project Selection grid.

Update Project Data Sources

From the Project Selection grid, select a project and click  to update its target or feature data source. See Update a Target or Feature Data Source for details.

Model Build Phase Data Section

The Data section is where you start putting together the data from imported data sources to build data models. Pages in the Data section let you:

- Import target data sources to form the target data set.
- Optionally import feature data sources from different feature data sets.
- Select dimensions to be used by SensibleAI Forecast for each data source.
- Group or cluster targets together to optimize downstream SensibleAI Forecast job run times, or to gain predictive accuracy.

When working through the Data section pages, the first thing to know which data set is being used to build a model. The data set can be from any relational table from a OneStream external database.

Specify Targets and Define the Data Set

After configuring your source data in OneStream and creating your SensibleAI Forecast project, you can use the **Targets** page (**Data** > **Targets**) to configure your target source data to use in your model. This is a two-step process.

- Define the data source connection and data sources to use.
- Specify target data source dimensions by selecting fields that contain the desired target dimensions, value dimension, date dimension, and location dimension (optional). You can specify multiple target data tables to be used during this step.

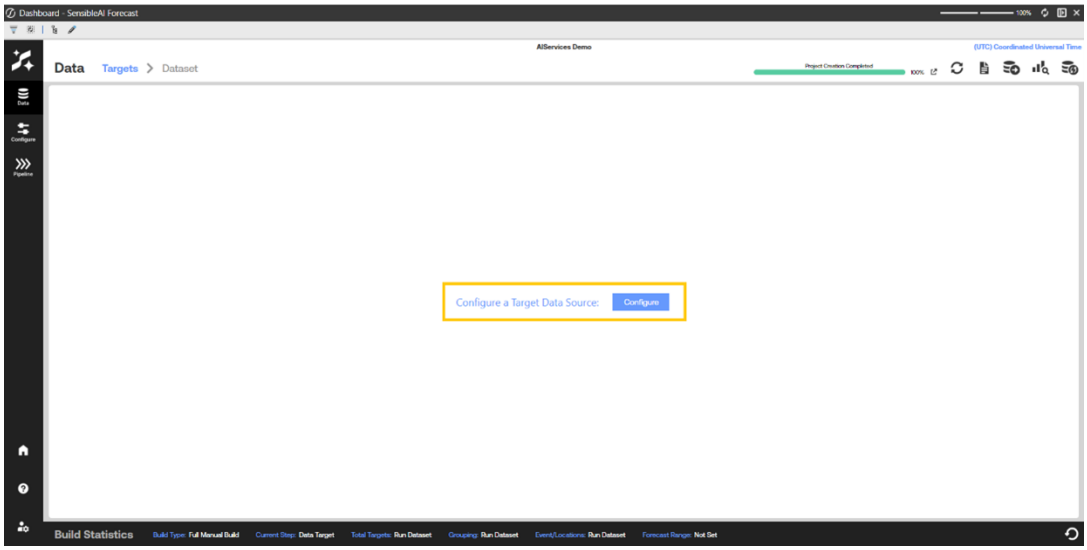
> **NOTE:** A location can be selected as both a target dimension and a location dimension. Selecting a column as a location dimension ensures that, when configuring locations for your project, all the locations within that source column are pre-populated and pre-mapped to the respective target. Selecting a location as a target dimension adds further uniqueness and more granularity to the target intersection.
>
> It is recommended to have a clustered column store index on data sources when able to avoid possible timeout issues.

> **IMPORTANT:** You should have detailed knowledge of your target data sources and how the sourced data in the data columns match to dimensions used to store that data. See Appendix 1: Data Quality Guide for information on data planning for your project.

Define Your Target Data Source Connection

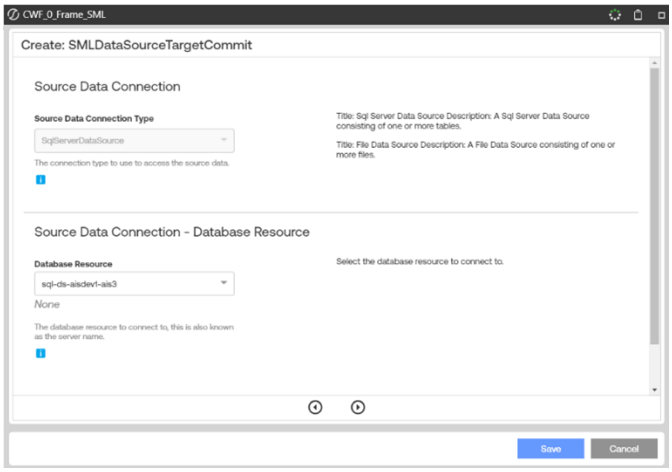
The first part to specifying targets and defining your data set for SensibleAI Forecast is to define the source connection for your data set.

1. Click **Data** > **Targets**.
2. The first time you access this page for a project, you must configure the data in your target data source.



3. Click **Configure**. The **Add Target Data Set Connection** dialog box displays.
4. In the Source Connection field, select the connection type of your Target Data Set.
5. Select the resource that contains the Target Data Set.
6. Select the Name of the resource that contains the Target Data Set.
7. Select the names of the initial set of target tables you created for importing into your SensibleAI Forecast project. If you imported multiple target data sources to use for the first model prediction, select the import table name for each.

> **TIP:** Only the first selected table name displays in the list after selecting. You can click the field to see all the selected import data files.



The **Preview** step shows data from the first imported target data set. Each row of data in the **Preview** step corresponds to a unique combination of data in the user-defined dimensions in the target source data set.

Use the information in the **Preview** step to verify that the data in the correct target data source is being used. This includes data from the source shown in the Preview table and the target data sources selected previously in the workflow.

Once you are sure the correct source connection and data sources are being used and the source data shown in the Preview pane are verified, you can select the dimensions being used for the target data source connection.

Select Target Data Source Dimensions

Continue defining the data set by specifying the intersection dimensions, value dimension, date dimension and location dimension (optional) to use in your SensibleAI Forecast model. This consists of matching the dimensions to be used for the target data in your Sensible Learning Machine model to the dimensions reserved while creating a cube for the target data source.

> **NOTE:** Only the specific dimensions reserved for SensibleAI Forecast should selected for each of the dimension types. If a dimension type does not correlate to data in the target data set, leave the field blank. If the location dimension is not being used, select **None**.

1. In the Target Dimensions field, select the target dimensions that have been defined to store data from your target data sources.

   These columns in the source data set define all the target variables that are used for predictions. The distinct combination of values across the target dimensions defines a target.

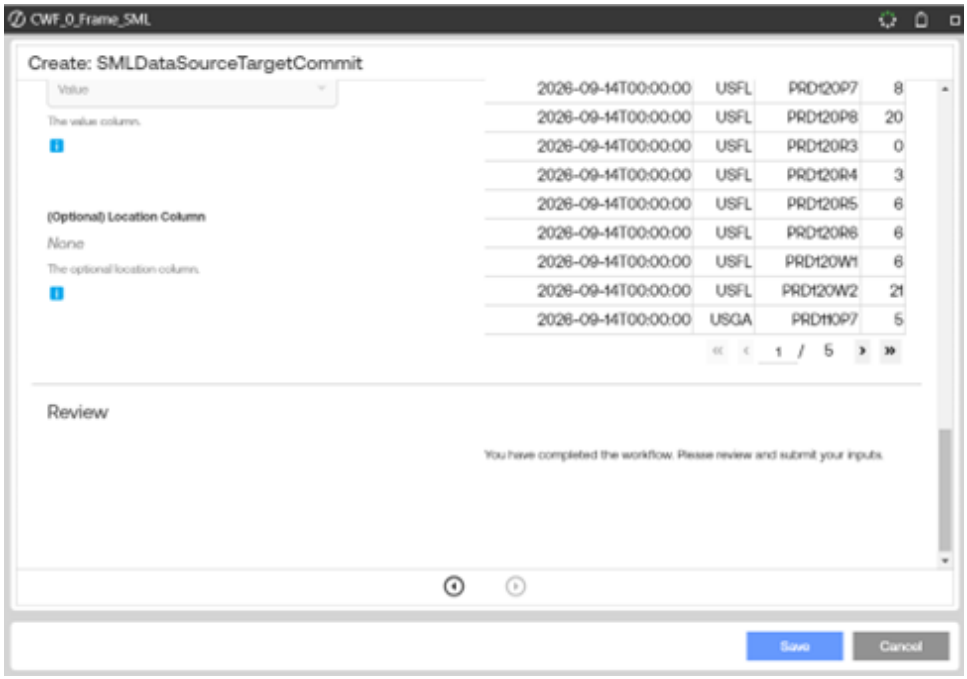
   Select the check box next to each applicable dimension. For example, if the user-defined dimensions UD1, UD2, and UD3 were reserved for source data and mapped to specific data columns in the data source, select **UD1**, **UD2**, and **UD3** from the list.

   > **NOTE:** Selecting more target dimensions leads to a higher number of unique intersections (or targets) for which to forecast.

2. In the Value Dimension field, select the dimension used for the value data coming from the target data source. Typically, this dimension is used to store source data values such as sales numbers.
3. In the Date Dimension field, select the dimension reserved for date data coming from the target data source. Typically, this dimension is used to store the date data from the target data source.
4. In the Location Dimension field, optionally select the dimension reserved for location data coming from the target data source. Select **None** if your data source does not include location information. The Location Dimension is used for mapping event and feature information to relevant targets in the Configure section.

For example, the following data table contains weekly sales dollars by location, store, store type, department, and date. The potential target dimensions include Location, Store, Store_Type, and Dept, with Dept having the highest granularity. One or more of these can be selected, depending on the desired forecast level. The value dimension, in this case, would be weekly sales dollars, and the date dimension would be Date. Location can be selected as both a target dimension and the location dimension.

	A	B	C	D	E	F
1	Location	Store	Store_Typ	Dept	Date	Weekly_Sales
2	Texas	1	A	Food	2/5/2018	24924.5
3	Texas	1	A	Hobbies	2/5/2018	50605.26953
4	Texas	1	B	Food	2/5/2018	13740.12012
5	Texas	1	B	Hobbies	2/5/2018	39954.03906
6	Texas	2	A	Food	2/5/2018	32229.38086
7	Texas	2	A	Hobbies	2/5/2018	5749.029785
8	Texas	2	B	Food	2/5/2018	21084.08008
9	Texas	2	B	Hobbies	2/5/2018	40129.01172
10	California	1	A	Food	2/5/2018	16930.99023
11	California	1	A	Hobbies	2/5/2018	30721.5
12	California	1	B	Food	2/5/2018	24213.17969
13	California	1	B	Hobbies	2/5/2018	8449.540039
14	California	2	A	Food	2/5/2018	41969.28906
15	California	2	A	Hobbies	2/5/2018	19466.91016
16	California	2	B	Food	2/5/2018	10217.5498
17	California	2	B	Hobbies	2/5/2018	13223.75977

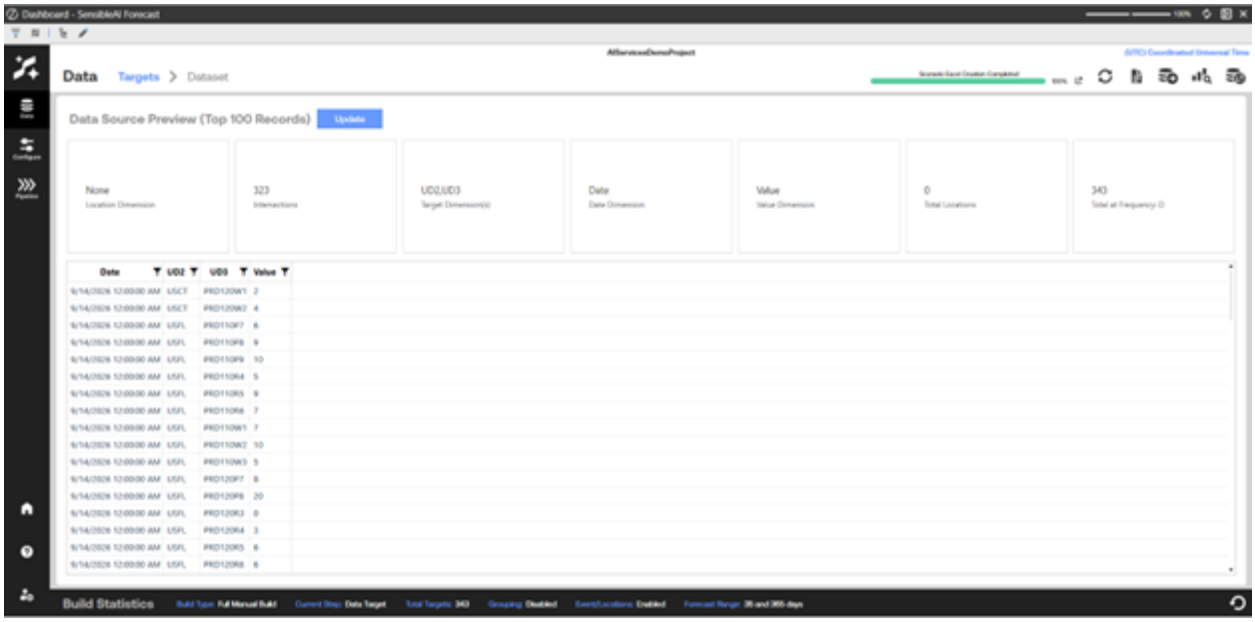


5. Click **Save** after completing the workflow. This adds a job to the job queue to validate the data and add the target data set to the model.
6. When the task completes, click **Refresh Current Page** .

The **Data Source Preview** pane displays.

> **NOTE:** Once the **Data Source Preview** pane displays in the **Targets** page, the **Configure** button no longer displays on the page. that is the default view for the page.

The data in the **Data Source Preview** pane displays information on the dimensions used to run the preview, as well as the number of data intersections in the SensibleAI Forecast data sources. Location and frequency information also displays.



Review the information in the **Data Source Preview** pane to verify the data targets are correctly defined for the model.

If any data in the **Data Source Preview** pane is not as expected, you can click **Update** to open the **Update Target Database Connection** dialog box to change target data set connection information.

> **NOTE:** The **Update** button is visible after the initial source connection is saved but is no longer visible after running the data set job in the **Data** > **Dataset** page.

Verify Your Data Sets

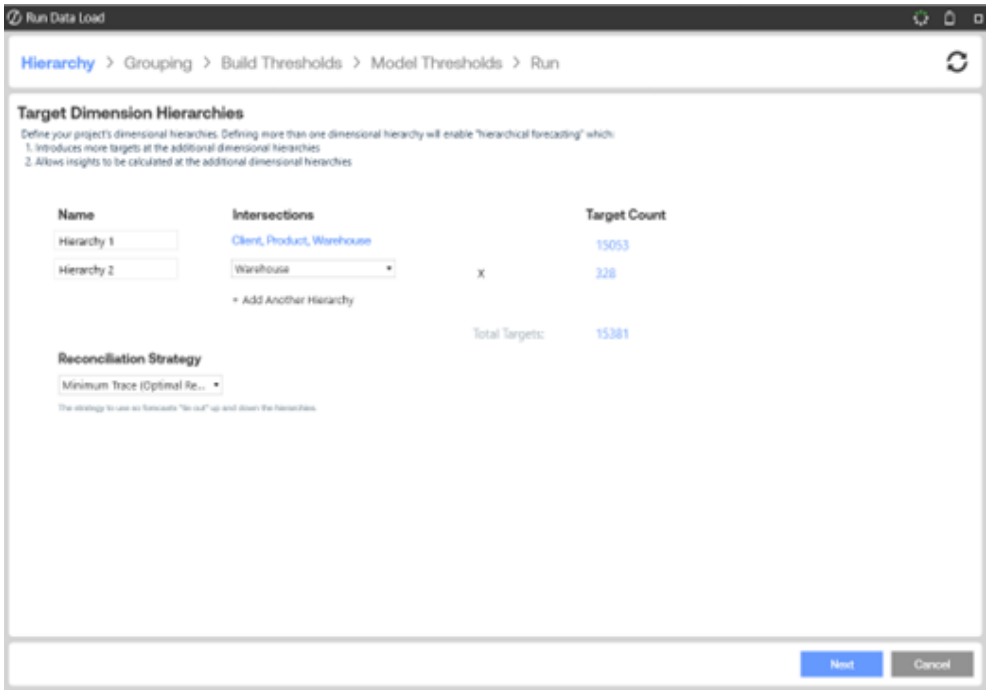
Once you have specified the targets to be used in your SensibleAI Forecast project, use the Data Set page to:

- Designate hierarchies based on your target data set.
- Review the project-level and advanced views of the merged data. Both views provide statistics on the merged data set and provide insight into how well your target data set is suited for your project.

Run Data Dataset

Use settings in the **Run Data Dataset** dialog box to designate hierarchies based on target data, determine how your data is grouped for your project, as well as configure the thresholds required to run feature engineering, grouping, and each model. Click **Run** on the **Dataset** page. The **Run Data Dataset** dialog box displays, which is broken up into the following steps:

Step 1: Hierarchies



In the Hierarchies Section, select hierarchies to be included in the project. Hierarchies cannot exceed environment restrictions for target limit. Include fields for all of the following:

Hierarchy Name: A unique name for the created hierarchy

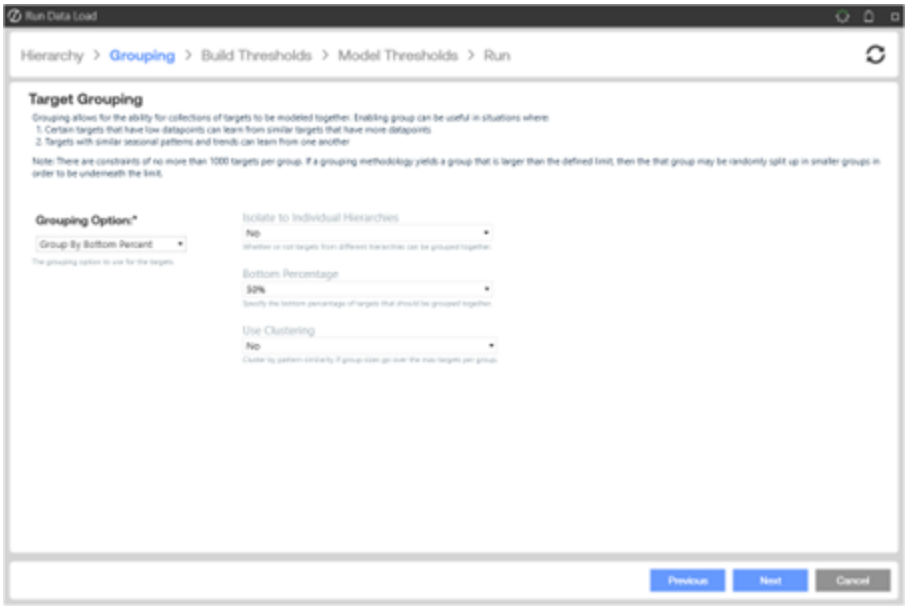
Intersections: Which dimensions the hierarchy should include.

Target Count: Targets generated for the given hierarchy.

Reconciliation Strategy: The strategy to use so forecasts “tie out” up and down the hierarchies.

- **Minimum Trace**: Minimum Trace Reconciliation is a statistical method used to adjust forecasts across hierarchical levels to ensure consistency while minimizing total forecast error variance. It leverages the forecast error covariance structure to optimally combine base forecasts from all levels, often resulting in more accurate and coherent forecasts compared to simpler approaches.
- **Bottom Up**: Bottom-Up Reconciliation is a simple method where forecasts are first generated at the most detailed (lowest) level of the hierarchy and then aggregated up to higher levels. This ensures perfect coherence across the hierarchy but may overlook useful information available at aggregated levels, potentially reducing overall forecast accuracy.
- **None**: If no reconciliation strategy is selected, forecasts are produced independently at each level of the hierarchy without any adjustment for coherence. This can lead to inconsistencies where aggregate forecasts do not match the sum of their components, but may still be acceptable if coherence is not critical for the use case.

Step 2: Grouping



In the Grouping Option field, select how you want to group targets for the model:

No Groups: Targets are not grouped for this model.

Group Bottom Percent: Groups together all targets that fall below a given percent of total significance. Use the Significance by Target charts and the Grouping Percentage (Bottom X Percent) drop-down option to understand how many targets fall below a given percent of total significance. In a situation where 50 targets out of 1,000 make up 90% of the total significance, it can be beneficial to group the lowest 10% to spend the majority of the model build resource on training the most significant targets.

If selecting Group Bottom Percent as the grouping option, you must also set the following options that display in the dialog box based on these selections:

- **Bottom Percentage**: Select the appropriate bottom percentage from the list.
- **Use Clustering**: Set to **Yes** take the targets grouped together that fall below a given percent of total significance and group further based on data similarities recognized by the XperiFlow engine.I**solate to Individual Hierarchies**: Whether or not targets from different hierarchies can be grouped together.
- **Group By Target Dimensions**: If selecting Group By Target Dimensions as the grouping option, you must also set the following options that display in the dialog box based on these selections:
- **Target Dimensions**: Shows dimensions selected when you select targets and define the data set. Select the dimensions you want to be grouped

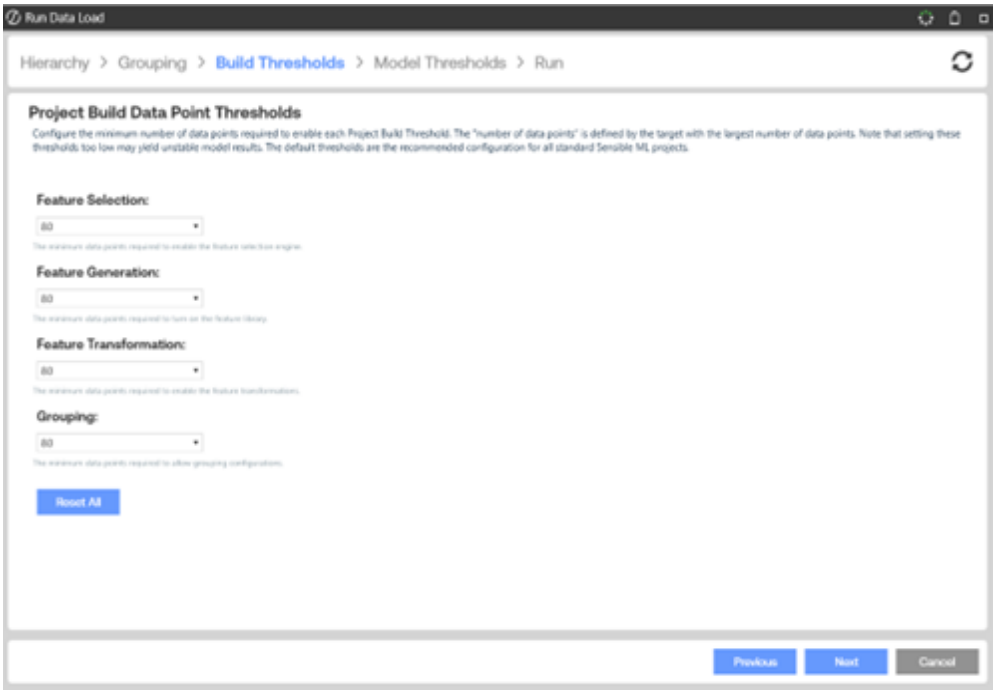
  > **TIP:** Two or more target dimensions must exist to select the dimensions to be grouped.

- **Isolate to Individual Hierarchies**: Whether or not targets from different hierarchies can be grouped together.
- **Group by Clustering**: This selection groups targets together based on data similarities recognized by the XperiFlow engine. The primary reason for grouping by clustering is to improve predictive accuracy.

Clustering involves grouping targets. A clustering algorithm classifies each target into a specific group, since targets in the same group typically have similar properties or features. Targets in different groups typically have highly dissimilar properties or features. Clustering provides valuable insights into data by showing what groups the targets fall into when clustering is applied.

Providing inverted views, both charts on this page illustrate the percent of total significance of targets along with the percentage within which they would be included for grouping.

Step 3: Build Thresholds

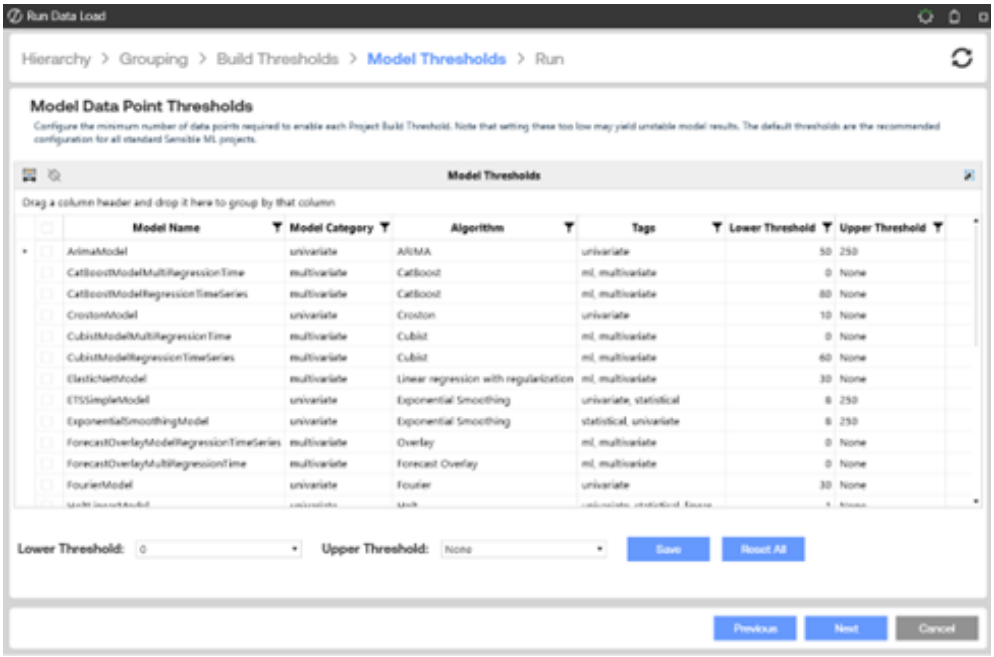


The Build Thresholds section of the dialog allows the user to configure how many data points are required to enable the following:

- **Feature Selection**
- **Feature Generation**
- **Feature Transformation**
- **Grouping**

The **Reset All** button will reset any changes made into these configurations to the defaults set in the Global Settings Dialog.

Step 4: Model Thresholds



The Model Thresholds section of the dialog allows the user to configure how many data points are required to enable each model:

- **Lower Threshold Combo Box**: The minimum required data points for the model to be used.
- **Upper Threshold Combo Box**: The maximum required data points for the model to be used.
- **Save Button**: Updates the selected models to the new thresholds configured in the Lower and Upper Threshold combo boxes.
- **Reset All Button**: Resets any changes made to these configurations to the defaults set in the Global Settings Dialog.

Step 5: Run

Click **Run** to start the data load job and monitor job progress. Click **Close** to close the **Job Progress** dialog box at any time while the job is running or after it has completed.

The XperiFlow engine analyzes the targetsand creates target groupings if grouping is selected. It also gathers descriptive statistics on the results.

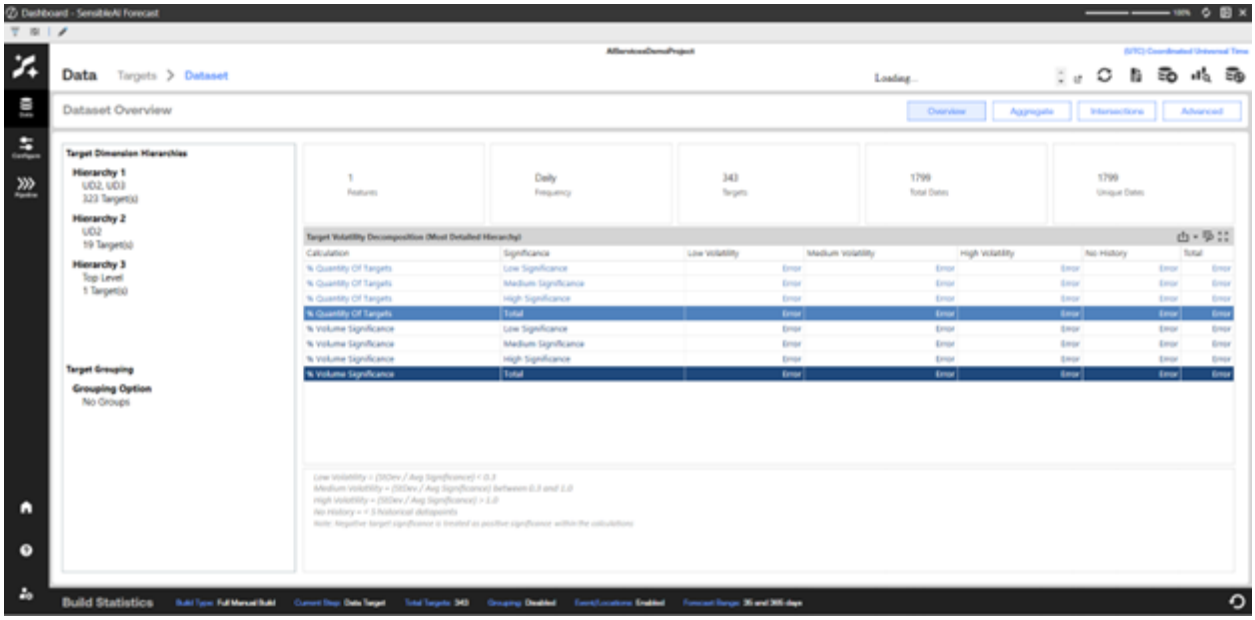
When the job successfully completes, click **Refresh Current Page** . The **Dataset** page updates and displays an overview, aggregate, and advanced pages that show statistics on the merged data set.

The **Run** button no longer displays once the data set job successfully completes. The Data Set Overview pane displays key statistics for your project, including the number of features and targets, the frequency of the data, and the number of unique dates.

Build Statistics at the bottom of the **Dataset** page update to show the number of total targets, and indicate whether Grouping, Events, and Locations are enabled. Also, the **Explore Targets and Features**, **Data Source Update** (for initial rebuild) and **Consumption Groups** pages are now enabled, and shows target and feature (if used in the data set) statistics for each unique data element in the data set.

Review Dataset Overview Statistics

The following graphic shows the **Dataset** page overview statistics:



The top of the Dataset Overview statistics view includes:

Features: The number of unique features produced by the data set job.

Frequency: Shows the time frequency of the overall data set. The time frequency is set based on the target that has the most granular level data. Frequency can be one of the following values:

- Daily
- Weekly
- Monthly
- Yearly

> **NOTE:** The frequency of an entire data set remains constant across all targets. If a data set frequency is not constant across all targets, it is recommended that the data set is split into multiple projects (one for each frequency). If kept in the same project, the most granular frequency target determines the overall data set frequency. The targets that are a less granular frequency have non-matching dates treated as missing values and are cleaned to get a complete series of the same frequency as the most granular data.

Targets: The number of unique targets in the merged data set.

Unique Dates: The number of unique dates in the merged data set.

Total Dates: The total number of unique dates in the merged data set. This may be greater than the unique dates because the data set may be missing dates based on frequency. The Total Dates statistic includes these missing dates.

Project statistics at the bottom of the Overview include:

Target Volatility Decomposition Chart: Shows the number of different volatile targets (low, medium, high). This is based on the standard deviation of the target versus the mean. It also shows how many of those targets are determined to be low, medium, or high significance. The most detailed hierarchy is used to generate these statistics.

Target Grouping: The grouping method chosen for this model build.

Target Dimension Hierarchies: The dimensions and target counts in each hierarchy.

Review Dataset Aggregate Statistics

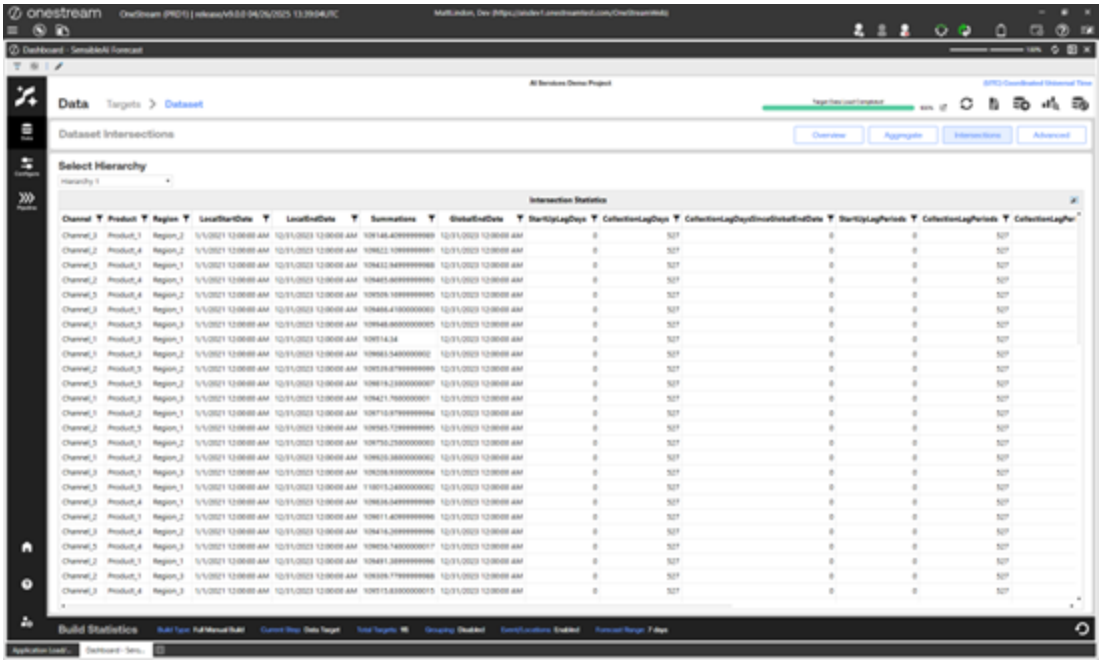
The Dataset Aggregate statistics includes Aggregate Target Time Series that can be sliced down to different granularities by dimension.

Aggregate Target Time Series: The aggregation of all targets on each given date in the data set. This helps to identify data set seasonality as a whole, on given time periods, or large trends over time.

> **TIP:** Use the date range sliders at the bottom of the page to change the time range on the Aggregate Target Time Series chart.

% Missing: The percent of targets that are missing data on each given date.

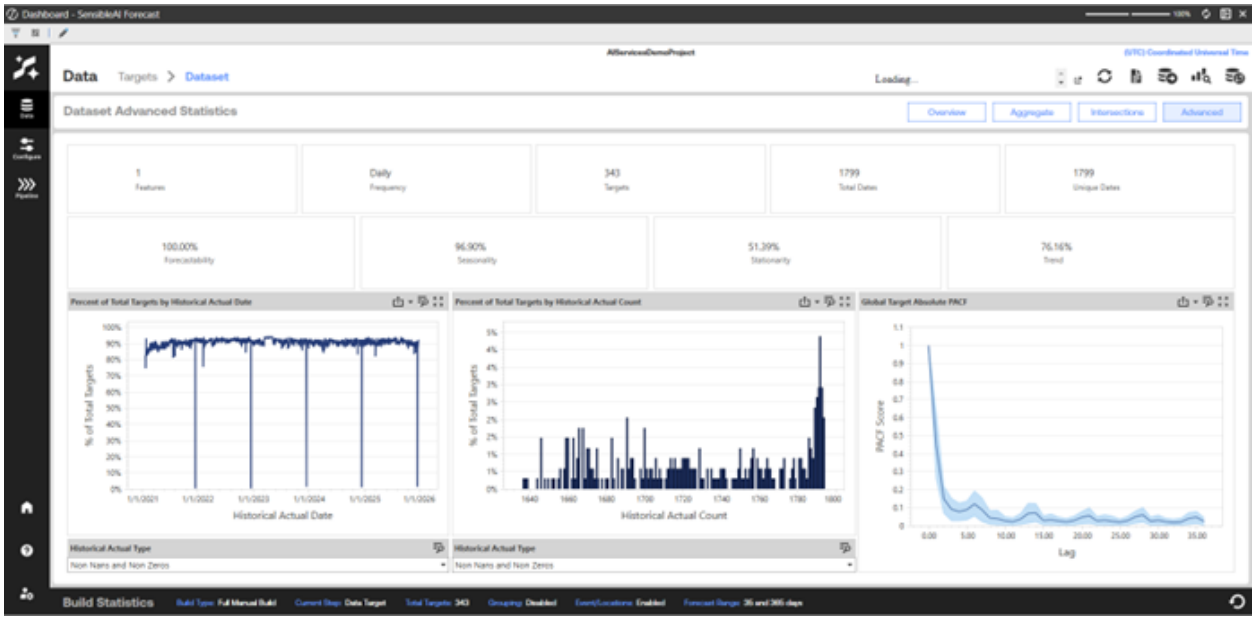
Review Intersection Data Set Statistics



The Intersection view of the data set statistics shows the generated artifact associated with each hierarchy that was created. Every hierarchy will have a unique artifact that includes every target generated on that dimension combination.

Review Advanced Data Set Statistics

The Advanced view of the data set statistics shows the statistics in the Project view along with additional charts. Click **Advanced** in the Data Set Project view to display the advanced statistics.



The Dataset Aggregate statistics include:

Forecastability: A percentage grade that is specific to SensibleAI Forecast that indicates how forecastable the target data set is. This metric is calculated as a percent of total targets (0- 100%) within the data set that are synonymous with random noise (which means no reasonable patterns can be detected). A score closer to 100% is desired.

Seasonality: A calculation of the percent of total targets (0-100%) in the target data set that have identifiable seasonality. A score closer to 100% is desired.

Stationarity: Indicates the percent of total targets (0-100%) in the target data set that are stationary, which means they do not experience a noticeable value level-shift. For example, a target whose mean value changes by 20% year-over-year would not be considered stationary. For certain time series models, it is easier to predict for stationary targets.

Trend: A calculation of the percent of total targets (0-100%) in the target data set that have an identifiable trend.

Percent of Total Targets by Historical Actual Date: This chart visualizes the percent of total targets with either non-zero and non-missing values, zero (Zeros) values, or missing (Nans) values (depending on the drop-down selection) over the data set's historical time frame. This provides the essential view on data sparsity over time.

Percent of Total Targets by Historical Actual Count: This distribution chart visualizes the percent of total targets with a given number of non-zero, non-missing, or non-zero or non-missing data points (x-axis), providing another view of data sparsity. Ideally, there should be as many targets as possible approaching the maximum number of available data points.

Global Target Absolute PACF (Partial Autocorrelation Function): A PACF chart demonstrates correlation (-1 to 1) of values based on the time increment between them. For example, a daily-level data set with a PACF score of 0.5 at an x-axis point of 7 signals that, on average, today's value has a correlation coefficient of 0.5 with the value of 7 days prior. This chart visualizes the mean, 90th percentile, and 10th percentile PACF score.

Use the information shown in the updated **Dataset** page to verify that target and grouping results are as expected. If you are satisfied with the grouping results, continue in the Model Build phase to the Configure section.

Model Build Phase Configure Section

The Configuration section of the Model Build phase is where you can:

- Import, configure, and map locations and events to targets.
- Set forecast and modeling parameters.

Use the individual pages in the Configuration section of the Modeling phase to:

- Analyze Forecasts and Set a Forecast Range
- Configure Locations
- Configure Source Features
- Configure Library - Features ** Configure Library Features
- Configure Library - Events ** Configure Event Features
- Assign Generators and Locations
- Set Modeling Options

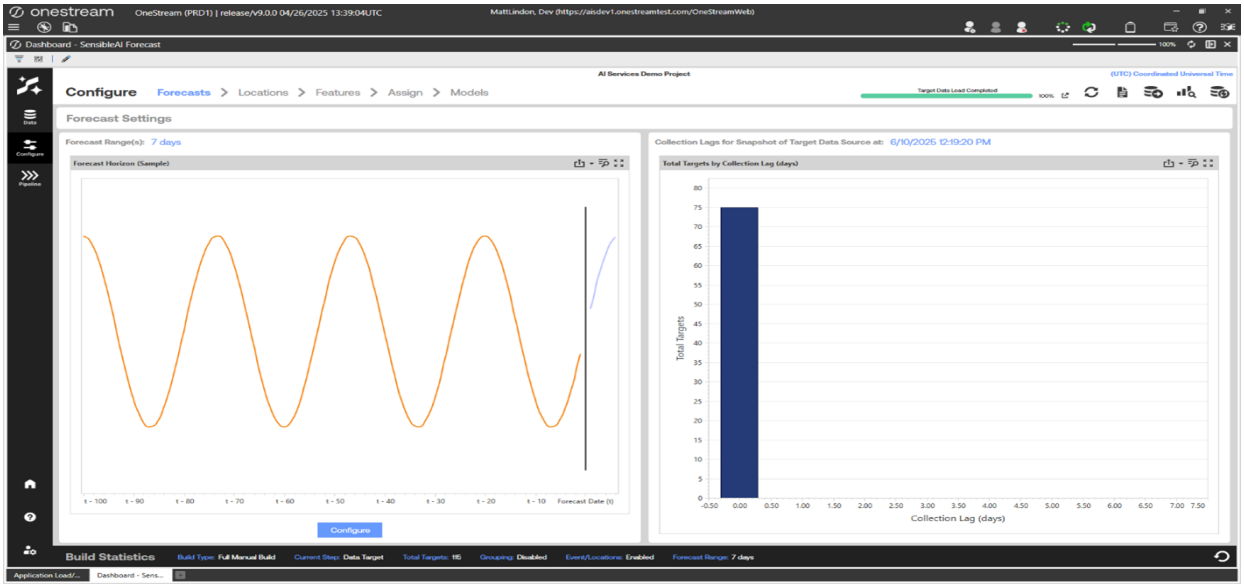
Analyze Forecasts and Set a Forecast Range

The **Forecast** page provides an easy-to-navigate step to set the desired forecast range. The forecast range is how far forward into the future the model is to forecast. The forecast range is dynamic based on the data set frequency (Days, Weeks, Months). Use the Forecast Range field in the Settings pane to set the forecast range.

The Forecast Horizon chart in the Settings pane displays an example of the configured forecast range. The line plot represents a pseudo-data set with the Actuals line representing the data set, the vertical line (Forecast Start Marker) representing the first data point of the forecast, and the Forecast line representing the additional forecasted data points. This representation shows what dates a forecast for your data would be over but is not your actual data nor your forecasted data.

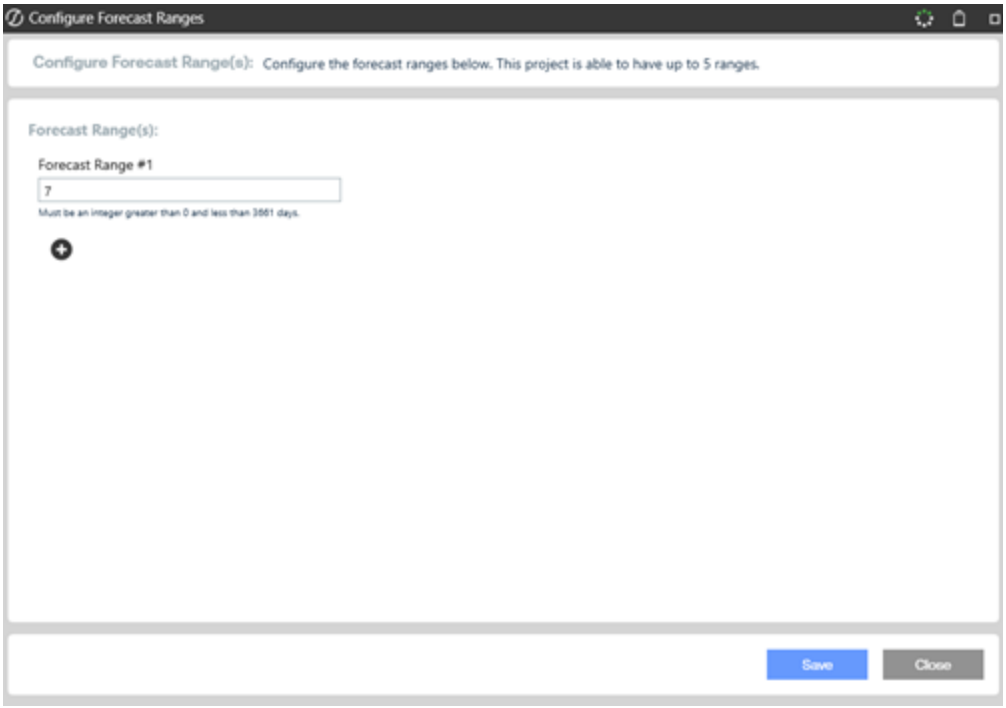
Collection lag is the length of time between when a data point occurs and when that data point is collected.

Use the **Forecast** page to select the desired forecast range for your prediction runs. You can also run new data snapshots from this page.



Set the Forecast Range

Use the **Forecast** page to set one or multiple **Forecast Ranges**, which determine how far forward from the latest date in the data set you want predictions generated for each forecast run. To configure the forecast ranges, follow these steps:



1. Click the **Configure** button to launch the dialog.
2. Input the desired forecast ranges for the Model build.
    a. Use the add ⊕ and remove 🗑 buttons to adjust the quantity of desired forecast ranges to be run.
3. Once the forecast ranges are configured, click the **Save** button to commit them to the Model Build.

Once the forecast ranges have been configured, the **Forecast Horizon** chart will update to give a visualization of the varying lengths of the forecasts.

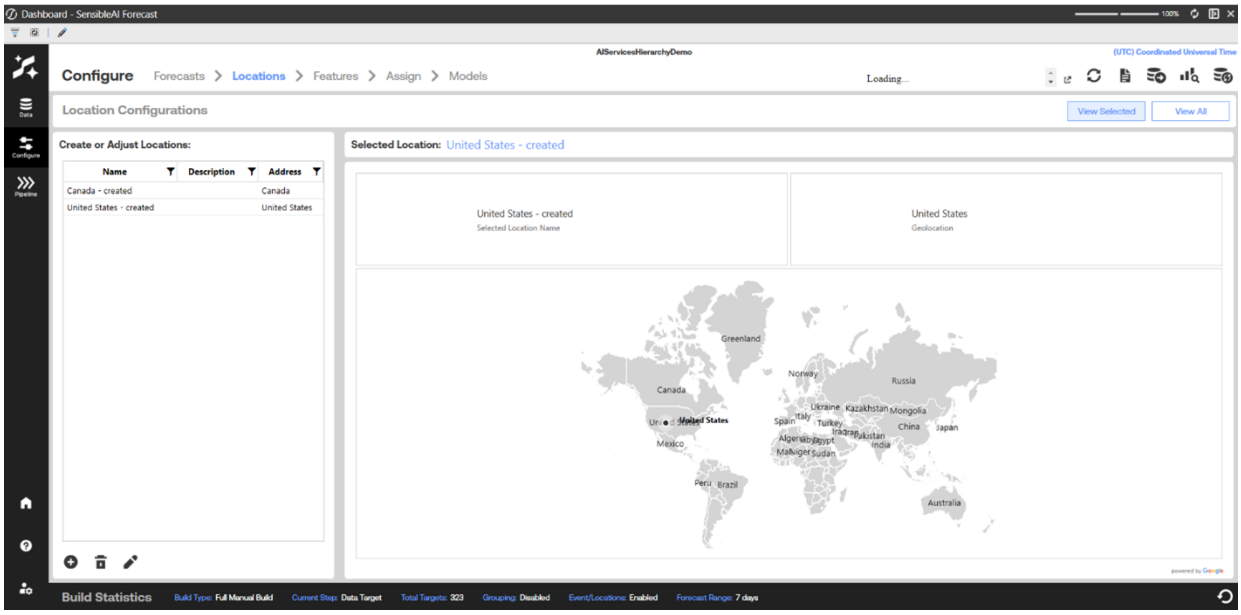
Configure Locations

Once you have grouped source data using the **Dataset** page, you can use the **Locations** page to manage locations to be used in your target data sets. You can add, delete, or edit locations using this page.

Locations provide a means to map events and features to specific targets. To do this, a location must also be mapped to one or several targets. This can either happen automatically by specifying a location dimension using the **Targets** page, or you can manually map locations to targets using the **Assign** page.

Other ways to add locations include:

- Using the location dimension of a data set job.
- Configuring generators that include locations in its features. See Configure Library Features.
- Importing event packages. See Configure Library Events.
- Uploading a locations file containing location names and addresses. This is similar to uploading an events file. See Configure Library Events for instructions.



IMPORTANT: Be aware of the geolocation assigned to a location name. For example, if a location name of **Rochester** is imported through a location dimension of the target data set, it may map to Rochester, New York by default, regardless of whether that location was meant for Rochester, Michigan. In a situation like this, the Rochester location should be edited with an address of “Rochester, Michigan."

Duplicate geolocation or location names are not allowed.

Add a Location

1. Click the **Add New Location** button at the bottom of the Locations pane. The **Add Location** dialog box displays.

2. Type a name for the location. This is the location name that displays in the location options when you assign generators and locations to targets.

> **NOTE:** The location name cannot include the keyword `created`, but any locations created through the location dimension or Generator Configuration include this keyword. If anything other than the description is updated for these locations, then the name must also change to not include the `created` keyword.

3. Optionally type a description for the location. The description also displays in the location options when you assign locations to targets.
4. Type an address for the location you are adding. The address cannot contain any special characters. The correct street address is not required.

5. Click **Save**. A message box notifies you that the location has been added.
6. Click **OK** to add the location and close the **Add Location** dialog box.

The location is added to the Locations pane and is selected as the current location. The interactive map also displays the location. **Delete** and **Update** buttons display at the bottom of the Locations pane so you can delete or edit the location.

To update a location, select it in the Locations pane and click the **Update** button, then use the **Update Location** dialog box to edit the name, description or address information for the selected location.

To delete a location, select it in the Locations pane and click the **Delete** button. A message box displays the location name and asks to confirm the deletion. Click **Delete** to delete the location from your locations list.

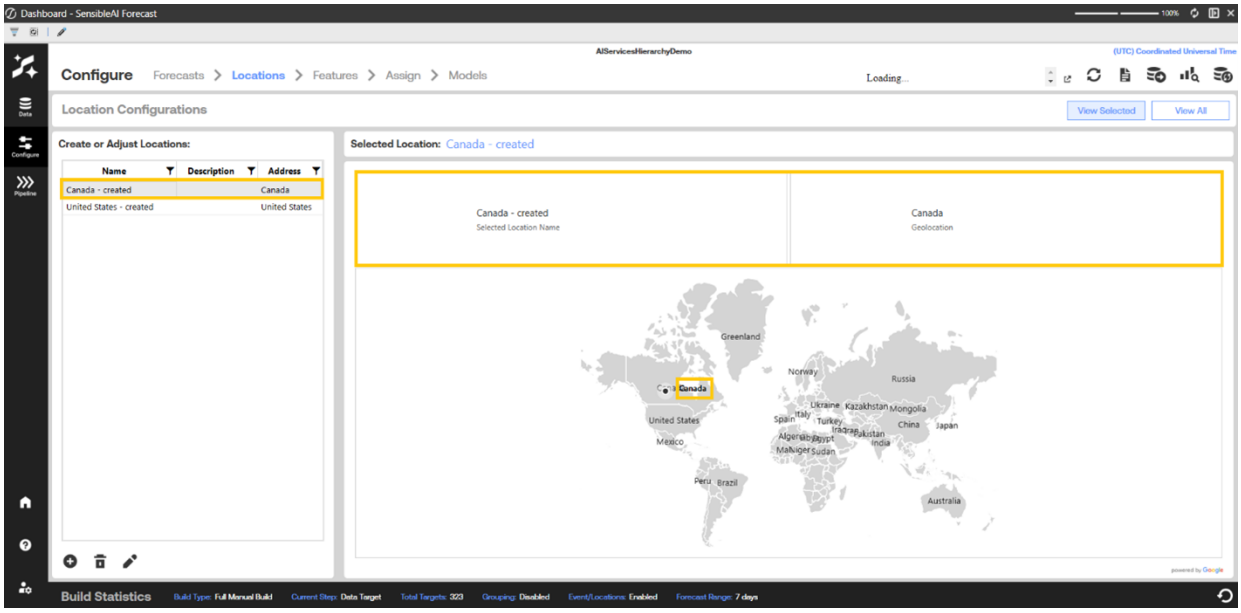
Find Locations on the Interactive Map

Use the interactive map on the **Locations** page to zoom in on specific locations anywhere on the globe. You can also move the map's center by clicking and holding down the mouse button to move the map.

There are two views you can use to view added locations: a single location view and an all locations view. Both views list all added locations in the Locations pane and the interactive map. Click the **Change view option** button at the top of the Selected Location pane to toggle between the two views.

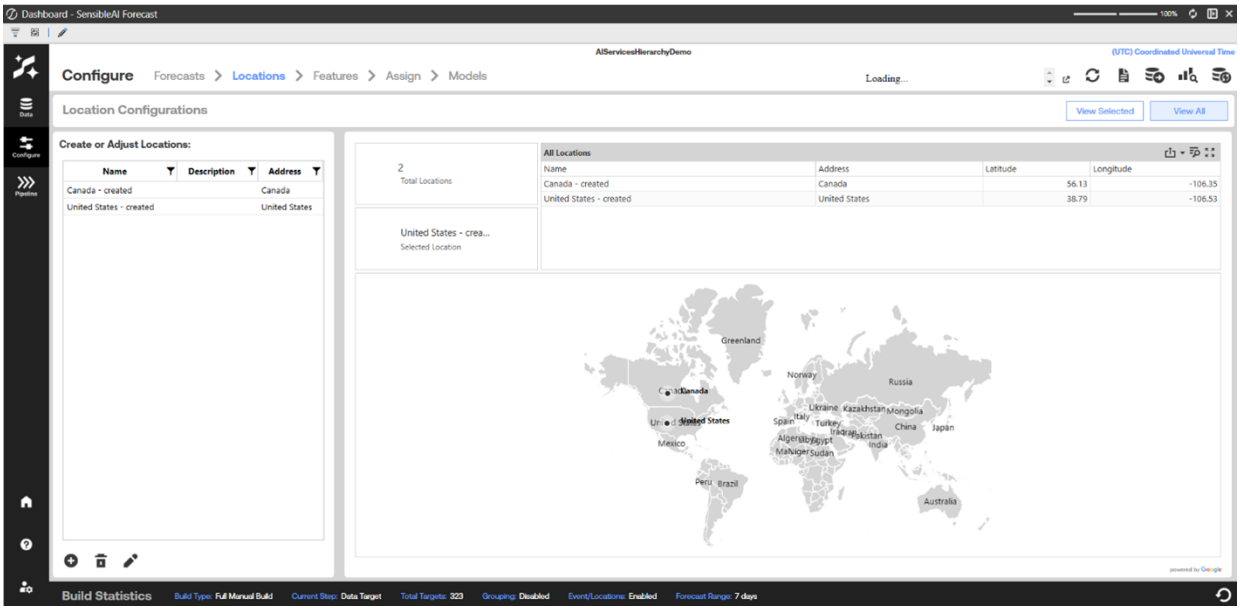
Find a Single Location

Click a location in the Locations pane to switch to the Single Location view. The Single Location view displays the selected location's name and geolocation at the top of the interactive map, and highlights the location in the interactive map.



Find All Locations

The All Locations view displays the last selected location's address and the total number of locations at the top of the interactive map. The All Locations pane lists each location, which includes the name, address, and longitude and latitude of each location. All added locations display on the interactive map.



Once you have added all the locations to be used in your project, you can continue by configuring events.

Specify Data Features

Data Sources containing features can be added on the **Configure** > **Features**> **Source** page. You can use features during modeling to help enhance prediction accuracy.

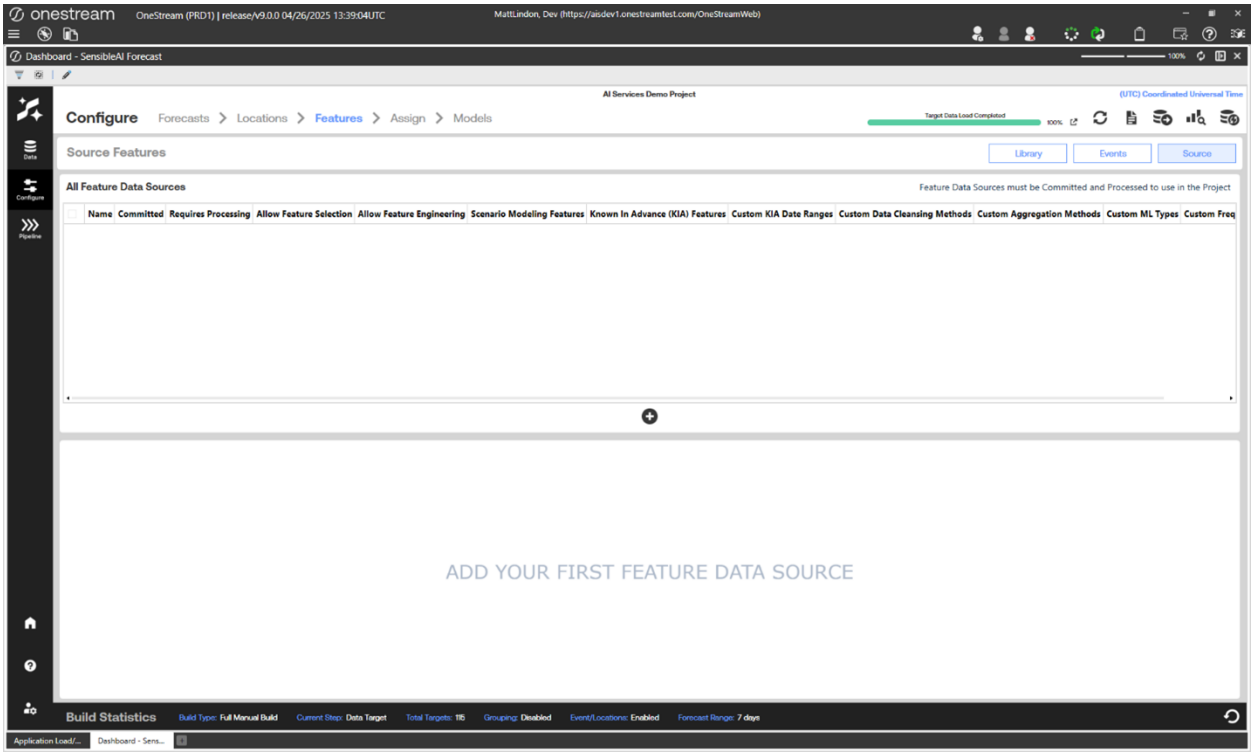
The **Features** page lets you specify multiple feature data sources. You can see previews of the features contained in each feature data source. Users can also commit or uncommit the features from the project build, as well as modify settings for each individual feature.

> **NOTE:** If you are not using a features data source in your project, you can skip this page.

Define Your Feature Data Source Connection

This page shows what your data definition looks like before configuring the definitions. Panel on the right changes after you configure it.

Click **Configure** > **Features**> **Source** to open the Source **Features** page. This first time you access this page, the Feature Data Source pane shows no feature data set information.



You can use the **Source Features** page to configure your feature source data to use in your model. Like specifying targets and defining your target data set, this is a two-step process.

- Define the data source connection and data sources to use.
- Specify feature data source dimensions by selecting fields that contain the desired feature dimensions, value dimension, date dimension, and location dimension (optional). You can specify multiple feature data sources to be used during this step.

> **IMPORTANT:** You should have detailed knowledge of your feature data sources and how the sourced data in the data columns match to dimensions used to store that data. See the *SensibleAI Forecast Data Quality Guide* for information on data planning for your project.

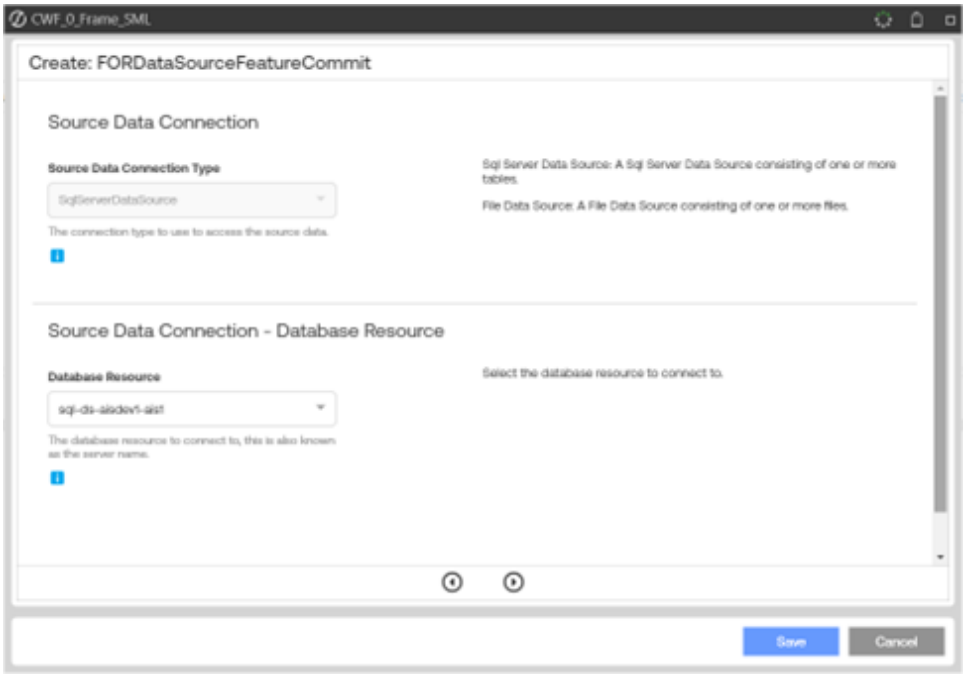
Specify the Data Source Connection

The first part to specifying features and defining your feature data set for SensibleAI Forecast is to define the source connection for your data set.

1. In the **Feature Data Sources** pane, click **Add** to add a feature data source to your SensibleAI Forecast model. The **Add Feature Data Set Connection** dialog box displays.
2. In the Source Data Connection field, select the connection type of your Feature Data Set.
3. Select your data source resource.
4. Select the Name of the resource to connect to.
5. Select the names of the initial set of feature data sources you created for importing into your SensibleAI Forecast project. If you imported multiple feature data sources for the first model prediction, select the import data source for each.

   > **TIP:** Only the first selected table name displays in the list after selecting. You can click the field to see all the selected import data files.

6. In the Data Source Name field, type a name for the feature data source you are creating.



7. Select next to open the preview pane.

> **IMPORTANT:** Use the information in the **Preview** pane to verify that the data in the correct feature data source is being used.

If you cannot verify the data in the **Preview** pane is the correct source data, or the sources are incorrect, you can navigate to previous steps to change the data source selected.

Once you are sure the correct source connection and data tables are being used and the source data shown in the **Preview** pane are verified, you can select the dimensions being used for the target data source.

Select Feature Data Source Dimensions

Continue specifying features and defining the data set by specifying any feature dimensions, value dimension, date dimension or location dimension for the data set to use in your SensibleAI Forecast model. This is basically matching the dimensions to be used for the feature data in your Sensible Learning Machine model to the dimensions reserved while creating a cube for the feature data source.

> **NOTE:** Only the specific dimensions reserved for SensibleAI Forecast should be selected for each of the dimension types. If a dimension type does not correlate to data in the feature data set, leave the field blank. If the location dimension is not being used, select **None**.

1. In the Intersection Dimensions field, select the feature dimensions that have been defined to store data from your feature data sources.

   The columns in the source feature data set define all the feature variables that are used for predictions. The distinct combination of values across the feature dimensions define a feature.

   Select the check box next to each applicable dimension. For example, if the user-defined dimensions UD1, UD2, UD3 and UD4 were reserved for source data and mapped to specific data columns in the data source, select **UD1**, **UD2**, **UD3**, and **UD4** from the list.

   If dimensions are selected for the feature data source that have the same name as a dimension in the target data source, then those dimensions are used to map features to targets. For example, UD1 is in the feature dimensions and the target dimensions, features with a value in the UD1 dimension are only mapped to targets with that same value in the UD1 dimension.

> **NOTE:** Setting the feature dimensions to the exact same dimensions specified for the target data set causes an error when running the job to validate the data and add the feature data set to the model.

> **TIP:** Selecting more feature dimensions leads to a higher number of unique intersections (or features) you can use in forecasting.

2. In the Value Dimension field, select the dimension used for the value data coming from the feature data source. Typically, this dimension is used to store source data values such as sales numbers.

   > **NOTE:** Only numeric values can be used to aid in predictions. Other types of values such as text are ignored.

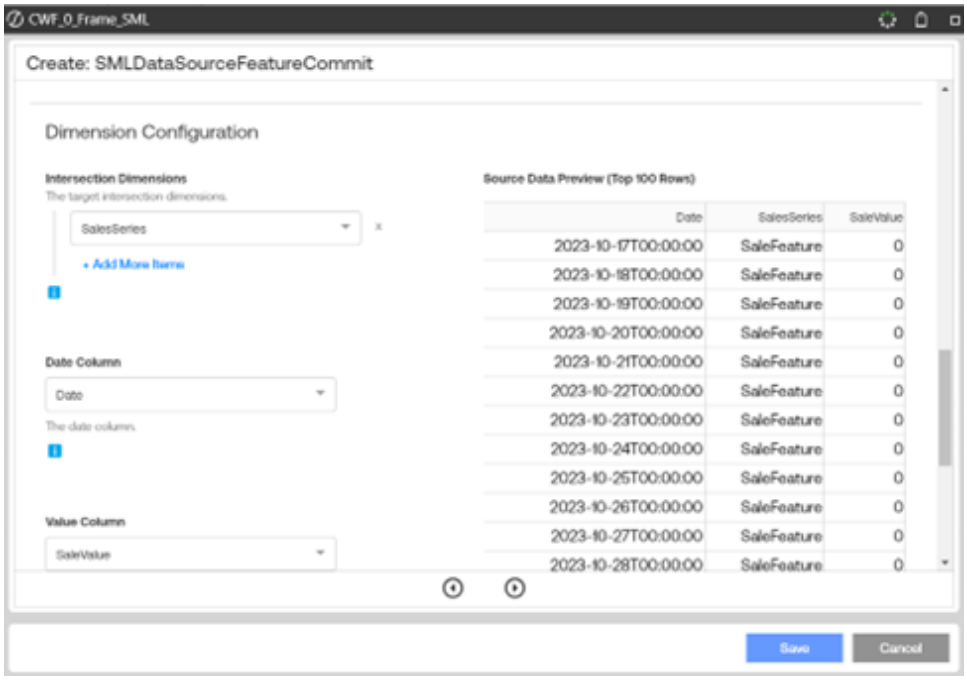
3. In the Date Dimension field, select the dimension reserved for date data coming from the feature data source.

4. In the Location Dimension field, select the dimension reserved for location data coming from the feature data source (optional).

   Select **None** if your feature data source does not include location information. The Location dimension is used during modeling to automatically map features to targets that have a location that is geographically inside of or equivalent to a given feature's location. For example, a feature with the location **Michigan** is mapped to a target with the location **Rochester, Michigan**, but is not mapped to a target with the location **USA**.

   > **TIP:** The location dimension can also be a feature dimension that adds uniqueness.

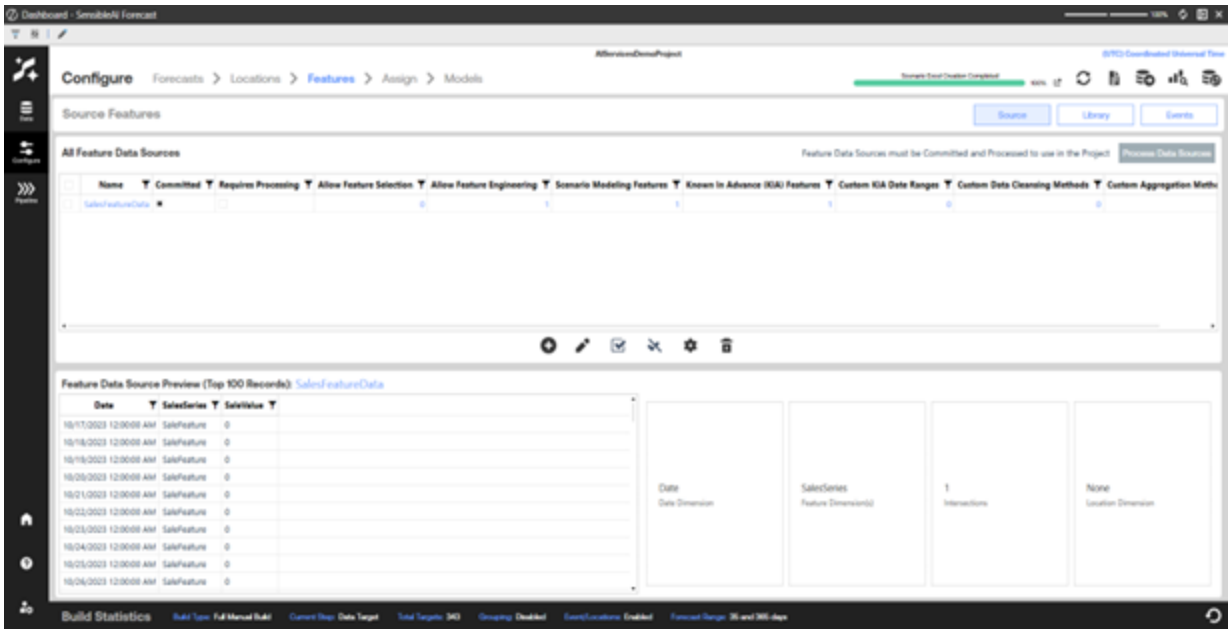
> **NOTE:** A location can be selected as both a feature dimension and a location dimension. Selecting a column as a location dimension ensures that, when configuring locations for your project, all the locations within that source column are pre-populated. Selecting a location as a feature dimension adds further uniqueness and more granularity to the feature intersection.



5. Complete the workflow after making your feature dimension selections, then click **Save**. This adds a job to the job queue to validate the data and add the feature data set to the model. The job runs tasks to complete the data definitions. A progress bar shows task progress. You can click **Cancel Task** at any time while the task is running to stop running the data definitions.
6. When the task completes, click **Refresh Current Page** .

The **Features** page displays the added feature data source listed in the **All Feature Data Source** pane. The **Feature Data Source Preview** pane displays below the **All Feature Data Source** pane, showing information for the top 100 feature records.

> **NOTE:** Once the **Data Source Preview** pane displays in the **Features** page, the **Configure** button no longer shows on the page, as it is the default view for the page.



You can also edit, delete, commit, or add a new feature set.

Edit Feature Data Source Attributes

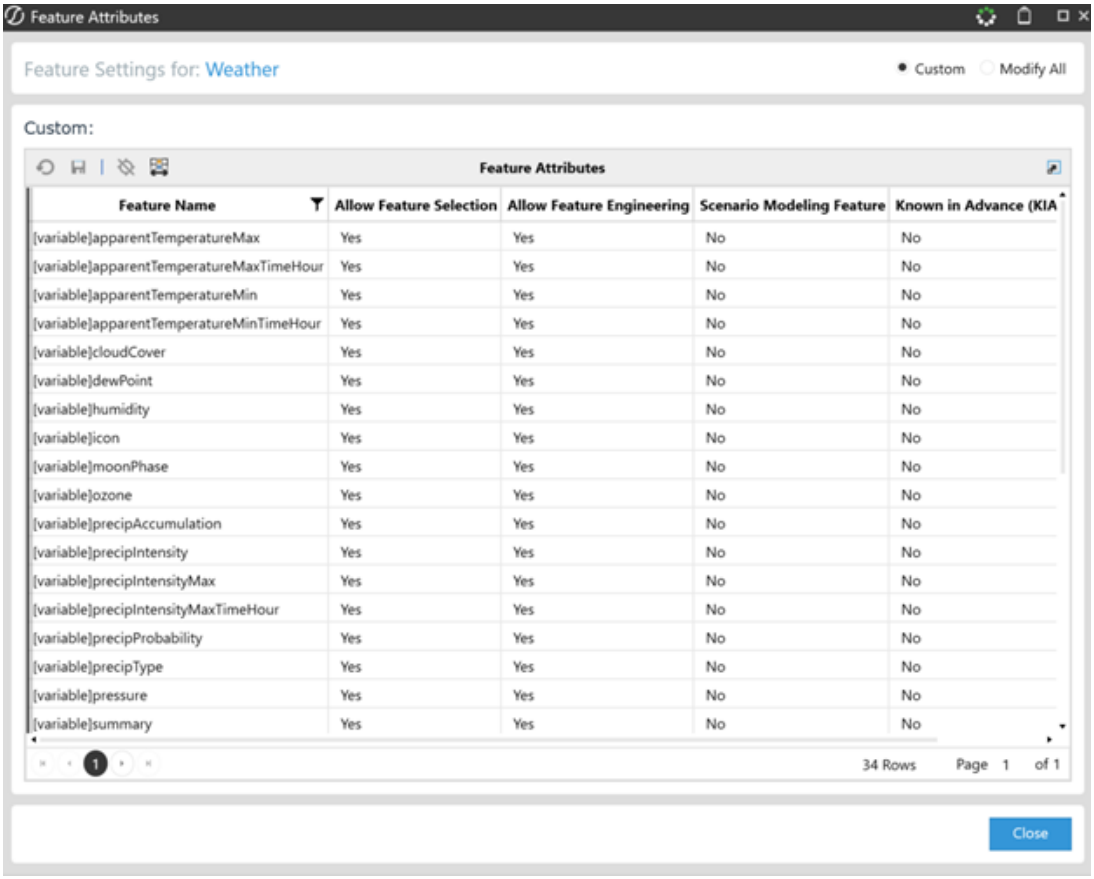
Once a feature data source has been added to the project, the **All Feature Data Sources** pane displays it at the top of the **Features** page. SensibleAI Forecast lets you set specific attributes for each feature in the data set.

Editing feature data source attributes is optional. Each feature's attributes have a default setting. Review the selections for each attribute. If you are satisfied with the defaults, click **Cancel** in the **Feature Attributes** dialog box without making changes, then commit the feature data source.

1. Click to select the data source whose attributes you want to edit.

   > **TIP:** Data information for the selected feature data set displays in the Data Source Preview pane.

2. Click the **Edit the Selected Feature Data Source's Attributes** button at the bottom of the pane. The **Feature Attributes** dialog box defaults to Custom view, which lists all the selected data source's feature attributes, and shows whether each attribute is selected (**Yes**) or not selected (**No**).



Each feature data set listed includes the following attributes:

Allow Feature Selection: The default value **Yes** allows the attribute to be filtered out during the feature selection process. Select **No** to ensure the feature is not filtered out during the feature selection process.

If too many features for a given target are set to **No**, then they still go through the feature selection process. This is to prevent too many features from being fed into any one model. This limit depends on which models are being run.

Allow Feature Engineering: The default value **Yes** indicates the feature can be engineered. Selecting **No** ensures that a feature cannot be engineered, such as lagging temperature by two weeks.

Scenario Modeling Feature: Select **Yes** if the feature should be included when defining custom Scenarios in Utilization and the intention of the project is to run predictions on different Scenarios. Otherwise, select **No**.

> **NOTE:** When selecting **Yes** for any event:
>
> - The project will be considered a Scenario Modeling project by the Xperiflow Engine.
>
> - Altering a Scenario Modeling project after the job has run requires a Restart or Manual Rebuild.
>
> - **Known In Advance** automatically changes to **Yes**.

Known In Advance (KIA): The default value No indicates that this feature does not have data that extends past the last actual data point (such as weather forecast for the next two weeks). Known-in-advance features cannot have any missing data past the forecast range (for example, five weeks for a five week forecast). Select Yes for the attributes that you know have data that extends beyond the forecast range.

> **IMPORTANT:** The prediction job cannot run if this setting is set to **Yes** and the feature is not available through the forecast when trying to run predictions.

KIA Date Range (Days): For features with **Known In Advance** set to **Yes**, this attribute allows the user to specify how many days of data will be known in advance. This attribute defaults to being blank. If a value is given to this attribute, but **Known In Advance** is not set to **Yes**, Xperiflow will automatically update the feature to **Known In Advance** set to **Yes**.

> **IMPORTANT:** The prediction job cannot run if this setting is configured and the number of days specified part of the feature data source.

Aggregation Method: This attribute allows a user to specify a preferred method of aggregating the feature data. By default, this will be set to **None** and the following options can be selected: **Sum**, **Mean**, **Median**, **Last**, **Max**, **Min**, and **Mode**.

Data Cleansing Method: This attribute allows a user to specify a preferred method of cleaning missing feature data. By default, this will be set to **None** and the following options can be selected: **Mean**, **Zero**, **Interpolate**, **Kalman**, and **Local Median**.

Frequency Override: This attribute allows a user to override the frequency of the feature data. By default, this will be set to **None**. If **None** is selected, Xperiflow will automatically determine the frequency of the feature data.

ML Type: This attribute allows a user to specify the data type of the feature data. By default, this will be set to **None** and the following options can be selected: **Binary Categorical**, **DateTime**, **Multi Categorical**, **Numerical**, and **Text**.

3. In the **Feature Attributes** dialog box, edit the feature's attributes in one of the following ways:

   Custom: Allows you to modify individual attribute values for features as desired.

   - Select a feature, then select the attributes values for that feature by clicking in each of the attribute selection fields and selecting **Yes** or **No** depending on the desired value.
   - Click the **Save** button  in the button bar to save your feature attribute changes.

   Modify All: Allows you to apply an individual attribute value to all features in a given feature data set.

   - Select the attribute option to apply the value.
   - Select the value of the attribute to apply.

- Click the **Save** button  at the bottom of the Feature Attributes dialog box, to save your feature attribute change and apply the selected value to the selected attribute for all features.

The data in the Data Source Preview pane displays information on the dimensions used to run the preview, as well as the number of data intersections in the SensibleAI Forecast data sources to be used for the model.

Verify Data Source Information

Review the information in the Data Source Preview pane to verify the data features are correctly defined for the model.

If any data in the Data Source Preview pane is not as expected, you can select the feature data source in the **Selected Feature Data Source** pane and do the following:

- Click the **Update the Selected Feature Data Source** button. This opens the **Update Feature Database Connection** dialog box so you can reselect feature data source dimensions.
- Select the feature data source in the **Selected Feature Data Source** pane and click the **Delete** button, then click **Delete** again to remove the selected feature data source from the list.

Commit or Decommit a Feature Data Source

You must commit any feature data sources to use in the SensibleAI Forecast project. You can also decommit any committed feature data source.

1. In the **Selected Feature Data Source** pane, select the feature data source and click the **Commit** button.
2. A message box informs you that the selected data source's commit status has changed. Click **OK** to close the message box.
3. Commit any other feature data sources as needed by repeating the previous steps.

Once you have committed your data sets, continue by processing Feature Data Sources to be used with your SensibleAI Forecast project.

> **NOTE:** Feature data sources can only be committed for a full build and not for a partial build.

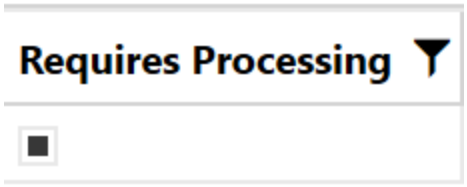
Process Feature Data Sources

After a feature data source has been committed, the user must process the feature data source. To process the data source:

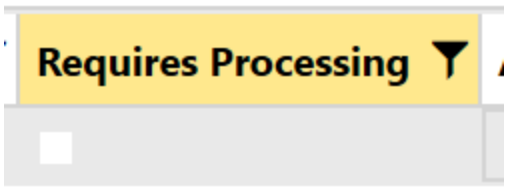
1. Upon the initial steps defined above when configuring Feature Data Sources, the **Process Data Sources** button will be disabled.

2. After committing one or multiple Feature Data Sources, the **Process Data Sources** button will become enabled.

3. When enabled the **All Feature Data Sources** grid will also display the **Requires Processing** field as on for any committed data sources. Upon these conditions, the user should click the **Process Data Sources** button, which will start a **Feature Data Load** job in the Xperiflow engine.



4. Upon completion of the **Feature Data Load** job, the **Requires Processing** field will be updated to off for all committed data sources and the **Process Data Sources** button will be disabled.



The **Feature Data Load** job is required to be run for any new changes to the Feature Data Sources. The above example is for configuring, committing, and loading a new Feature Data Source, but a **Feature Data Load** job will also be required for the following conditions:

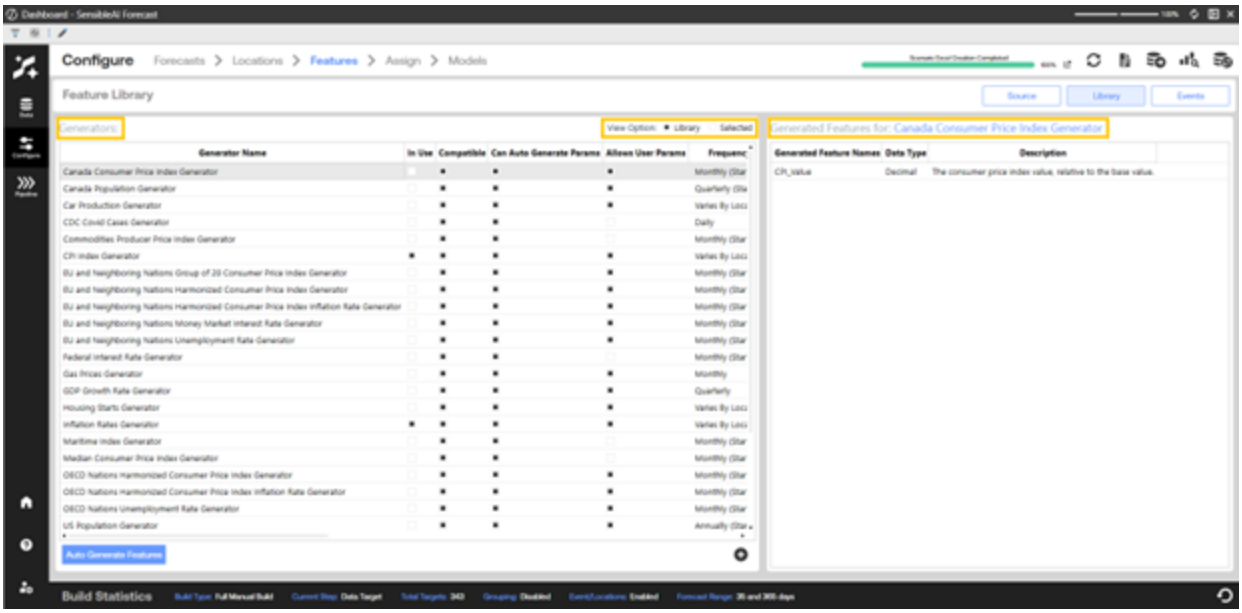
- A Feature Data Source that has been committed and been included in a **Feature Data Load** job is uncommitted.
- A Feature Data Source that has been committed and been included in a **Feature Data Load** job has updates made to its Data Source Attributes.

> **NOTE:** A user will not be able to navigate to the Pipeline Section of Model Build if any Feature Data Sources require processing. The Feature Data Load job can be run as many times as required to process all of the Feature Data Sources.

Configure Library - Features

The **Library** view of the **Features** page in the Configure section lets you select the generators used to add external features to the data set. External features can increase the predictive accuracy of the Machine Learning models. You can create multiple data sets containing features (also known as generator instances) from the library of generators that SensibleAI Forecast provides.

The Generators pane has a Library view that lists all generators currently available in the Generator library, and a Selected view that lists the generators specifically selected to add external features to the data set.



The following information displays for each generator.

In Use: A checked box means the generator is used in the current model build.

Compatible: A checked box means the generator is compatible with the current model build. This is based on number of data points, frequency, and the data set's earliest start date.

Can Auto Generate Params: A checked box means the generator can generate the necessary initial parameters required to gather external data.

Allows User Params: A checked box means you can add additional parameters to the generator. See Add Generator Configurations for instructions.

Frequency: The frequency of the external data (Daily, Weekly, Monthly).This is merged into the target data frequency.

Source: The data source and citation for the external information.

Select a generator from the Generators pane to show its generated features in the Generated Features pane. This can be one to many features, depending on the type of generator. The Generated Features pane also displays the following information for each selected generator.

Generated Feature Names: The name of each feature being generated.

Data Type: The type of data (such as integer, boolean, or decimal) that the feature contains.

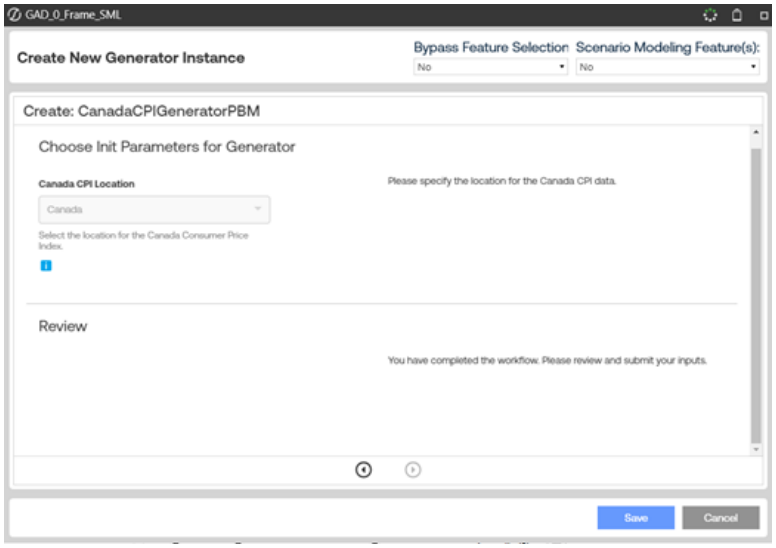
Description: A brief description of the information the feature contains.

You can add any of the compatible generators to use in the pipeline as external features.

Add a Generator Configuration

To add generator configurations:

1. Select desired generator and click **Add a new Feature Library Feature** . The **Add Feature Library Feature** dialog box displays.



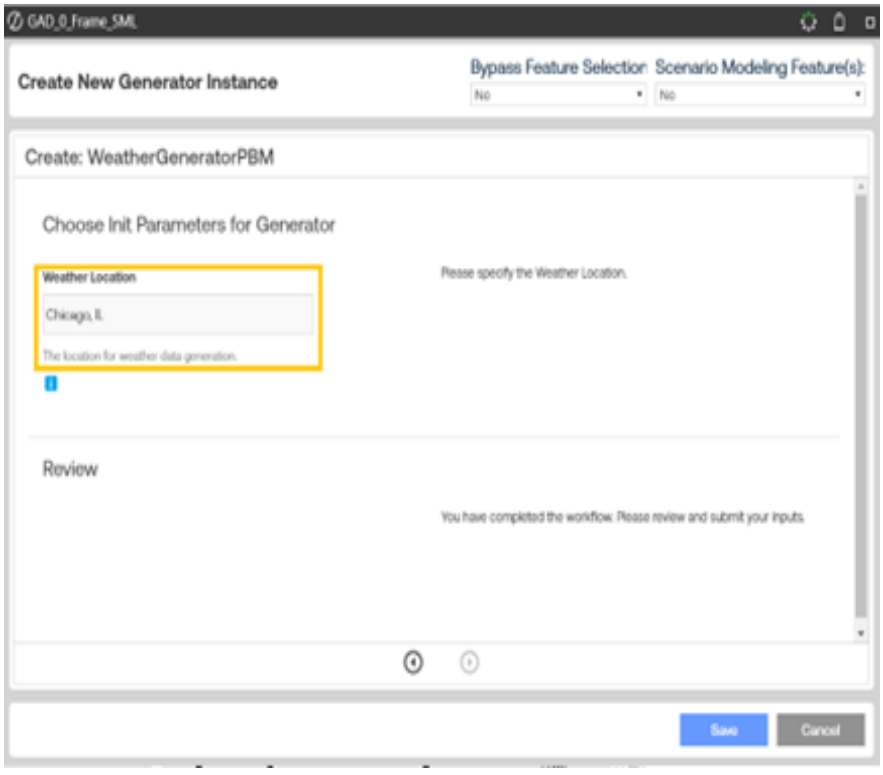
2. In the Bypass Feature Selection field, select **Yes** if the feature should bypass feature selection. Otherwise, select **No**.

> **NOTE:** Only select **Yes** for Bypass Feature Selection if it is certain that the listed generated features benefit your models. SensibleAI Forecast runs all generated features through a feature selection process to determine if the feature is important to the models.

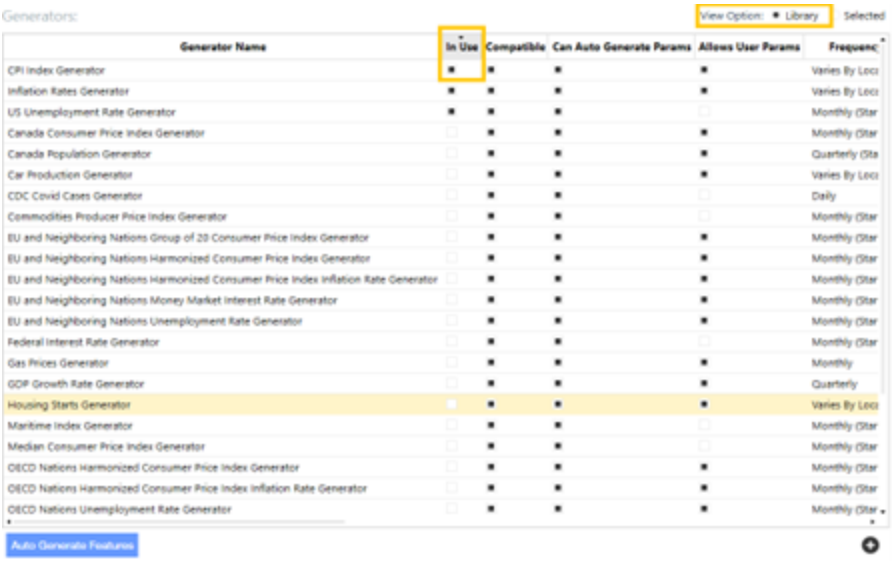
3. In the Scenario Modeling Feature field, select **Yes** if the feature should be included when defining custom Scenarios in Utilization and the intention of the project is to run predictions on different Scenarios. Otherwise, select **No**.

> **NOTE:** When selecting **Yes** for any event:
>
> - The project will be considered a Scenario Modeling project by the Xperiflow Engine.
>
> - Altering a Scenario Modeling project after the job has run requires a Restart or Full Manual Rebuild.
>
> - **Known In Advance** automatically changes to **Yes**.

4. If a generator allows for custom parameters, they must be designated through the workflow steps:

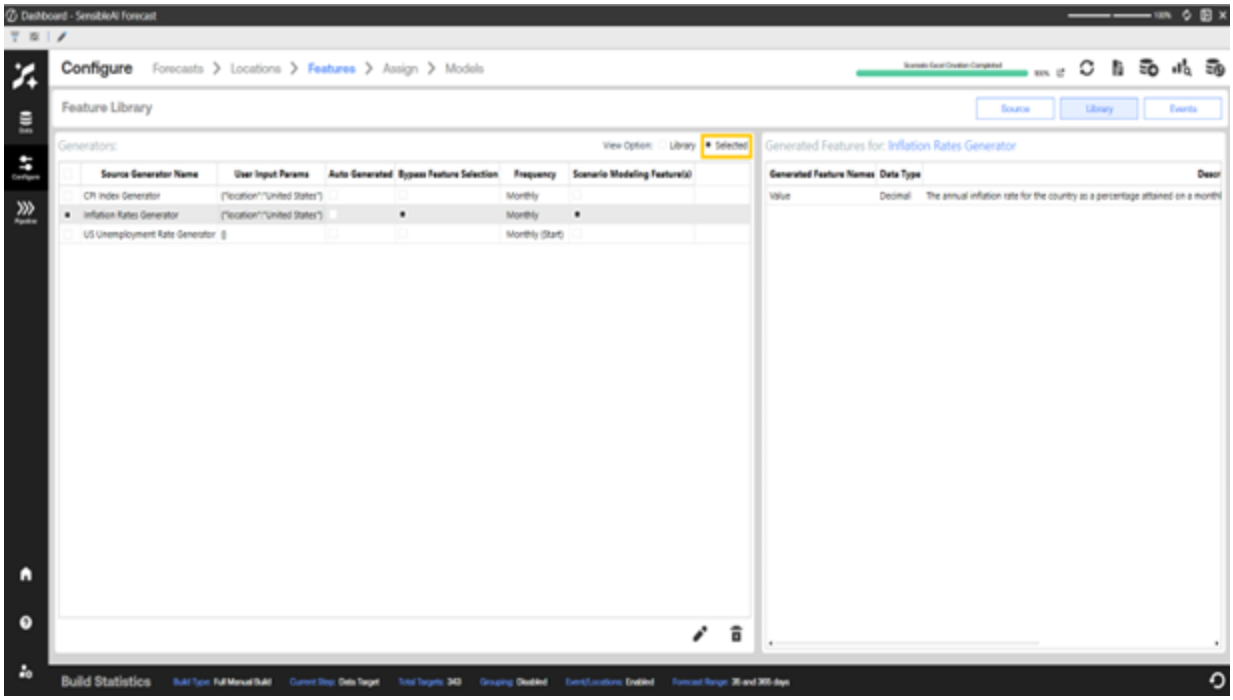


5. Click **Save** to close the **Add Feature Library Feature** dialog box. The generator configuration is validated and added to the Selected list if valid. This also checks the **In Use** option in the Generator Pane Library list so you can see the selected generators from the Library view.



> **TIP:** You can use custom parameters to further customize generated features. For example, when creating an instance using the WeatherGen Generator, you can enter a location parameter to get the WeatherGen Features for a specific location.

Once you select a generator and add it to the list of selected generators, you can click the **Selected** radio button at the top of the Generators pane to see the generators selected for your models.



Delete a Generator Configuration

To delete a generator configuration:

1. In the Selected view of the Generators pane, select the generator to delete and click **Delete the Selected Configuration** .
2. In the **Delete Feature Library Feature** dialog box, click **Delete**, then click **OK**. The generator is removed from the list .

Update a Generator Configuration

To update a generator configuration:

1. In the Selected view of the Generators pane, select the generator to edit and click **Update**  . The **Update Feature Library Feature** dialog box displays.
2. Use the **Update Feature Library Feature** dialog box to update the generator's configuration. This dialog box works the same way as the **Add Feature Library Feature** dialog box. See instructions in Add a Generator Configuration for information on using the dialog box.

Generator Configurations Information

The Generators pane lists generator configurations currently set for the model build. The following information displays for each generator configuration:

Source Generator Name: The name of the generator to be used in the configuration.

User Input Params: The input parameters provided by the user (if any).

Use Automatic Params: Indicates if the generator configuration is set to use the automatic parameters.

Bypass Feature Selection: Indicates if the features generated by this configuration bypass feature selection.

Frequency: The frequency of the external data. This merges into the target data frequency.

When you select a generator configuration, the right pane shows the features generated by the generator. This may be a single feature or many features, depending on the type of generator. The following information displays:

Generated Feature Names: The name of the feature being generated.

Data Type: The type of data, such as integer, Boolean, or decimal, that the feature contains.

Description: A brief explanation of the information the feature contains.

Generated Features for: Inflation Rates Generator

Generated Feature Names	Data Type	Description
Value	Decimal	The annual inflation rate for the country as a percentage attained on a month

After auto-assigning events and locations or assigning events and locations to specific targets, you can move to the **Model** page.

Generator Custom Parameters

Some generators can take in custom parameters to fetch specific data from external sources. While certain generators can take in a location address, they do not support all locations. Below are details on what options can be provided for the custom parameters for generators that have these limitations.

- EuroStatHarmonizedConsumerPriceIndexGen
    - Supported Locations:
        - Luxembourg, Iceland, Czechia, European Union, Malta, Latvia, Romania, Finland, Portugal, Germany, Belgium, Denmark, Poland, Cyprus, Europe, Hungary, France, Spain, Bulgaria, Albania, Sweden, Norway, Croatia, North Macedonia, Serbia, Slovenia, Slovakia, Austria, Netherlands, Italy, Lithuania, Estonia, Switzerland, Montenegro
- EuroStatGroupOf20ConsumerPriceIndexGen
    - Supported Locations:
        - Italy, France, Germany
- EuroStatHarmonizedConsumerPriceIndexInflationRateGen

    - Supported Locations:
        - Romania, Estonia, Bulgaria, Spain, Latvia, Iceland, Norway, Europe, Croatia, Sweden, Albania, Montenegro, Belgium, Portugal, Cyprus, Serbia, Luxembourg, Italy, North Macedonia, European Union, Czechia, Switzerland, Hungary, Poland, Lithuania, Malta, Slovakia, Finland, Slovenia, France, Austria, Germany, Netherlands, Denmark
- EuroStatUnemploymentRateGen
    - Supported Locations:
        - Malta, Switzerland, Norway, France, Lithuania, Latvia, Luxembourg, Netherlands, Slovakia, Bulgaria, Spain, Iceland, Italy, Estonia, Croatia, Belgium, Poland, Slovenia, Finland, Czechia, Sweden, Cyprus, Germany, Austria, Denmark, Hungary, Romania, Portugal
- EuroStatMoneyMarketInterestRateGen
    - Supported Locations:
        - Poland, Sweden, Denmark, Bulgaria, Hungary, Romania, Czechia
- EuroStatHouseholdSavingsRateGen
    - Supported Locations:
        - Poland, Slovenia, Sweden, Spain, Finland, Portugal, Austria, Belgium, France, Denmark, Italy, Germany, Norway, Czechia, Hungary, Netherlands
- Harmonized Consumer Price Index
    - Supported Locations:
        - United States
- Harmonized Consumer Price Index Inflation Rate

    - Supported Locations:
        - United States
- Unemployment
    - Supported Locations:
        - Japan, United States
- Canada Population
    - Supported Locations:
        - Canada
        - Newfoundland and Labrador, Canada
        - Prince Edward Island, Canada
        - Nova Scotia, Canada
        - Quebec, Canada
        - Ontario, Canada
        - Manitoba, Canada
        - Saskatchewan, Canada
        - Alberta, Canada
        - British Columbia, Canada
        - Yukon, Canada
        - Northwest Territories, Canada
        - Nunavit, Canada
- Canada Consumer Price Index

- Supported Locations:
  - Canada
  - Newfoundland and Labrador, Canada
  - Prince Edward Island, Canada
  - Nova Scotia, Canada
  - New Brunswick, Canada
  - Quebec, Canada
  - Ontario, Canada
  - Manitoba, Canada
  - Saskatchewan, Canada
  - Alberta, Canada
  - British Columbia, Canada
  - Whitehorse, Yukon, Canada
  - Yellowknife, Northwest Territories, Canada
  - Iqaluit, Nunavit, Canada
  - Calgary, Alberta, Canada
  - Charlottetown and Summerside, Prince Edward Island, Canada
  - Edmonton, Alberta, Canada
  - Halifax, Nova Scotia, Canada
  - Montreal, Quebec, Canada
  - Regina, Saskatchewan, Canada

- Saint John, New Brunswick, Canada
- Saskatoon, Saskatchewan, Canada
- St. John's, New Brunswick, Canada
- Thunder Bay, Ontario, Canada
- Toronto, Ontario, Canada
- Vancouver, British Columbia, Canada
- Victoria, British Columbia, Canada
- Winnipeg, Manitoba, Canada

> **NOTE:** THE FOLLOWING GENERATORS ARE PREMIUM ADD ON GENERATORS. PLEASE CONTACT YOUR ONESTREAM REPRESENTATIVE FOR ACCESS.

- Trading Economics Consumer Price Index (CPI)

- Supported locations:
  - Afghanistan, Albania, Algeria, Angola, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belgium, Belize, Benin, Bermuda, Bhutan, Bolivia, Bosnia and Herzegovina, Botswana, Brazil, Brunei, Bulgaria, Burkina Faso, Burundi, Cambodia, Cameroon, Canada, Cape Verde, Cayman Islands, Chad, Chile, China, Colombia, Congo, Costa Rica, Croatia, Cyprus, Czech Republic, Denmark, Djibouti, Dominican Republic, East Timor, Ecuador, Egypt, El Salvador, Estonia, Ethiopia, Faroe Islands, Fiji, Finland, France, Gabon, Gambia, Georgia, Germany, Ghana, Greece, Guatemala, Guinea, Guyana, Haiti, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Iran, Iraq, Ireland, Israel, Italy, Ivory Coast, Jamaica, Japan, Jordan, Kazakhstan, Kenya, Kosovo, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Lithuania, Luxembourg, Macau, Macedonia, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, Mauritania, Mauritius, Mexico, Moldova, Mongolia, Montenegro, Morocco, Mozambique, Myanmar, Namibia, Nepal, Netherlands, New Caledonia, New Zealand, Nicaragua, Niger, Nigeria, Norway, Oman, Pakistan, Palestine, Panama, Papua New Guinea, Paraguay, Peru, Philippines, Poland, Portugal, Puerto Rico, Qatar, Republic of the Congo, Romania, Russia, Rwanda, Sao Tome and Principe, Saudi Arabia, Senegal, Serbia, Seychelles, Sierra Leone, Singapore, Slovakia, Slovenia, Solomon Islands, Somalia, South Africa, South Korea, South Sudan, Spain, Sri Lanka, Suriname, Swaziland, Sweden, Switzerland, Taiwan, Tajikistan, Tanzania, Thailand, Togo, Trinidad and Tobago, Tunisia, Turkey, Uganda, Ukraine, United Arab Emirates, United Kingdom, United States, Uruguay, Vanuatu, Venezuela, Vietnam, Zambia, Zimbabwe

- Trading Economics Gas Prices

- Supported locations:
  - Albania, Argentina, Australia, Austria, Azerbaijan, Bahrain, Belarus, Belgium, Bolivia, Bosnia and Herzegovina, Brazil, Bulgaria, Cambodia, Canada, Chile, China, Colombia, Costa Rica, Croatia, Cyprus, Czech Republic, Denmark, Egypt, El Salvador, Estonia, Finland, France, Germany, Ghana, Greece, Guatemala, Haiti, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Iran, Ireland, Israel, Italy, Jamaica, Japan, Kazakhstan, Kenya, Kuwait, Latvia, Lebanon, Lithuania, Luxembourg, Macedonia, Malaysia, Malta, Mexico, Montenegro, Netherlands, New Zealand, Nicaragua, Nigeria, Norway, Oman, Pakistan, Paraguay, Peru, Philippines, Poland, Portugal, Puerto Rico, Qatar, Romania, Russia, Saudi Arabia, Serbia, Singapore, Slovakia, Slovenia, South Africa, South Korea, Spain, Sudan, Sweden, Switzerland, Tanzania, Thailand, Trinidad and Tobago, Turkey, Ukraine, United Arab Emirates, United kingdom, United states, Uruguay, Venezuela, Vietnam, Zimbabwe
- Trading Economics GDP Growth Rate

- Supported locations:
  - Albania, Angola, Argentina, Australia, Austria, Bahrain, Belgium, Belize, Bosnia and Herzegovina, Botswana, Brazil, Bulgaria, Canada, Cape Verde, Chile, China, Colombia, Costa Rica, Croatia, Cyprus, Czech Republic, Denmark, Dominican Republic, Ecuador, El Salvador, Estonia, Finland, France, Germany, Ghana, Greece, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Ireland, Israel, Italy, Jamaica, Japan, Kenya, Latvia, Lesotho, Lithuania, Luxembourg, Macedonia, Malaysia, Malta, Mauritius, Mexico, Moldova, Namibia, Netherlands, New Zealand, Nigeria, Norway, Paraguay, Peru, Philippines, Poland, Portugal, Qatar, Romania, Rwanda, Saudi Arabia, Senegal, Serbia, Singapore, Slovakia, Slovenia, South Africa, South Korea, Spain, Sweden, Switzerland, Taiwan, Thailand, Trinidad and Tobago, Tunisia, Turkey, Uganda, Ukraine, United kingdom, United states

- Trading Economics Housing Starts
  - Supported locations:
    - Bulgaria, Canada, China, Czech Republic, Denmark, Finland, France, Iceland, Israel, Japan, Kyrgyzstan, Norway, Russia, Spain, Sweden, Thailand, Turkey, United Kingdom, United states
- Trading Economics Inflation Rates

- Supported locations:
  - Albania, Algeria, Afghanistan, Angola, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belarus, Belgium, Belize, Benin, Bermuda, Bhutan, Bolivia, Bosnia and Herzegovina, Botswana, Brazil, Brunei, Bulgaria, Burkina Faso, Burundi, Cambodia, Cameroon, Canada, Cape Verde, Central African Republic, Chad, Chile, China, Colombia, Comoros, Congo, Costa Rica, Croatia, Cuba, Cyprus, Czech Republic, Denmark, Djibouti, Dominican Republic, East Timor, Ecuador, Egypt, El Salvador, Equatorial Guinea, Eritrea, Estonia, Ethiopia, Faroe Islands, Fiji, Finland, France, Gabon, Gambia, Georgia, Germany, Ghana, Greece, Guatemala, Guinea, Guinea Bissau, Guyana, Haiti, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Iran, Iraq, Ireland, Israel, Italy, Ivory Coast, Jamaica, Japan, Jordan, Kazakhstan, Kenya, Kosovo, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Luxembourg, Macau, Macedonia, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, Mauritania, Mauritius, Mexico, Moldova, Mongolia, Montenegro, Morocco, Mozambique, Myanmar, Namibia, Nepal, Netherlands, New Caledonia, New Zealand, Nicaragua, Niger, Nigeria, Norway, Oman, Pakistan, Palestine, Panama, Papua New Guinea, Paraguay, Peru, Philippines, Poland, Portugal, Puerto Rico, Qatar, Romania, Russia, Rwanda, Sao Tome and Principe, Saudi Arabia, Senegal, Serbia, Seychelles, Sierra Leone, Singapore, Slovakia, Slovenia, Solomon Islands, Somalia, South Africa, South Korea, South Sudan, Spain, Sri Lanka, Suriname, Swaziland, Sweden, Switzerland, Taiwan, Tajikistan, Tanzania, Thailand, Togo, Trinidad and Tobago, Tunisia, Turkey, Turkmenistan, Uganda, Ukraine, United Kingdom, United states, Uruguay, Vanuatu, Venezuela, Vietnam, Zambia, Zimbabwe

- Trading Economics Population

- Supported locations:

- Albania, Algeria, Afghanistan, Angola, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belarus, Belgium, Belize, Benin, Bermuda, Bhutan, Bolivia, Bosnia and Herzegovina, Botswana, Brazil, Brunei, Bulgaria, Burkina Faso, Burundi, Cambodia, Cameroon, Canada, Cape Verde, Cayman Islands Central African Republic, Chad, Chile, China, Colombia, Comoros, Congo, Costa Rica, Croatia, Cuba, Cyprus, Czech Republic, Denmark, Djibouti, Dominican Republic, East Timor, Ecuador, Egypt, El Salvador, Equatorial Guinea, Eritrea, Estonia, Ethiopia, Faroe Islands, Fiji, Finland, France, Gabon, Gambia, Georgia, Germany, Ghana, Greece, Guatemala, Guinea, Guinea Bissau, Guyana, Haiti, Honduras, Hong Kong, Hungary, Iceland, India, Indonesia, Iran, Iraq, Ireland, Israel, Italy, Ivory Coast, Jamaica, Japan, Jordan, Kazakhstan, Kenya, Kosovo, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Luxembourg, Macau, Macedonia, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, Mauritania, Mauritius, Mexico, Moldova, Mongolia, Montenegro, Morocco, Mozambique, Myanmar, Namibia, Nepal, Netherlands, New Caledonia, New Zealand, Nicaragua, Niger, Nigeria, North Korea Norway, Oman, Pakistan, Palestine, Panama, Papua New Guinea, Paraguay, Peru, Philippines, Poland, Portugal, Puerto Rico, Qatar, Republic of the Congo, Romania, Russia, Rwanda, Sao Tome and Principe, Saudi Arabia, Senegal, Serbia, Seychelles, Sierra Leone, Singapore, Slovakia, Slovenia, Solomon Islands, Somalia, South Africa, South Korea, South Sudan, Spain, Sri Lanka, Suriname, Swaziland, Sweden, Switzerland, Syria, Taiwan, Tajikistan, Tanzania, Thailand, Togo, Trinidad and Tobago, Tunisia, Turkey, Turkmenistan, Uganda, Ukraine, United Arab Emirates, United Kingdom, United states, Uruguay, Uzbekistan, Vanuatu, Venezuela, Vietnam, Zambia, Zimbabwe

- Trading Economics Unemployment Rate

- Supported locations:
  - Albania, Afghanistan, Argentina, Armenia, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belarus, Belgium, Belize, Benin, Bolivia, Botswana, Brazil, Brunei, Bulgaria, Burkina Faso, Burundi, Cambodia, Cameroon, Canada, Cape Verde, Cayman Islands Central African Republic, Chad, Chile, Colombia, Comoros, Costa Rica, Croatia, Cuba, Cyprus, Czech Republic, Denmark, Djibouti, East Timor, Egypt, El Salvador, Equatorial Guinea, Eritrea, Estonia, Faroe Islands, Fiji, Finland, France, Gabon, Gambia, Germany, Ghana, Greece, Guatemala, Guinea, Guinea Bissau, Guyana, Haiti, Honduras, Hong Kong, Hungary, Iceland, India, Iran, Iraq, Ireland, Israel, Italy, Ivory Coast, Japan, Jordan, Kazakhstan, Kosovo, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Luxembourg, Macau, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, Mauritania, Mauritius, Mexico, Moldova, Montenegro, Morocco, Myanmar, Namibia, Nepal, Netherlands, New Caledonia, New Zealand, Niger, North Korea, Norway, Oman, Pakistan, Palestine, Panama, Papua New Guinea, Paraguay, Peru, Poland, Portugal, Puerto Rico, Republic of the Congo, Romania, Russia, Sao Tome and Principe, Seychelles, Sierra Leone, Singapore, Slovakia, Slovenia, South Africa, South Korea, South Sudan, Spain, Sri Lanka, Suriname, Swaziland, Sweden, Switzerland, Syria, Taiwan, Tajikistan, Togo, Trinidad and Tobago, Turkey, Turkmenistan, Uganda, Ukraine, United Arab Emirates, United Kingdom, United states, Uruguay, Uzbekistan, Venezuela, Vietnam, Zambia, Zimbabwe

Configure Features - Events

Use the **Events** view of the **Library** page to add, edit, and delete events along with their occurrences. Events can be created manually, through an event file upload, or by selecting a pre-established event package.

The modeling process uses calendar-based events to increase the model accuracy.

Add events that you know are related to the targets being predicted. When created, an event is initially be empty with no dates (occurrences) assigned to the event. You must define one or more occurrences that define what days that particular event falls on. You can add locations to the event to map events to targets using the assigned location dimension.

> **NOTE:** This capability also exists in the **Manage Events** page in the Utilization phase. However, you cannot add or delete events from the **Manage Events** page.

Add Events

To add events, click the **Add** button  at the bottom of the Events pane. Then use the **Add Event** dialog box to add a single event, or to add an events package.

> **NOTE:** Events are validated, and all event details and mappings are stored for later use. You cannot have two separate events with the same name. However, two separate events can have different names with the same occurrences.

Add a Single Event

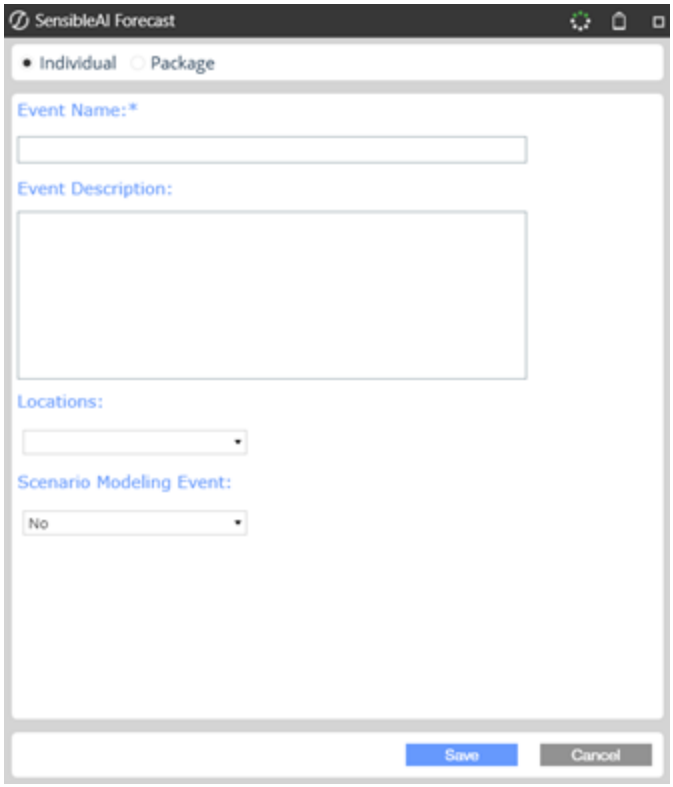
When creating a single event, you can also add locations to the event. Locations are used to map events to targets using assigned locations. This simplifies the assignment of the event to targets.

1. In the **Add Event** dialog box, select **Individual**.
2. Type a unique name for the event you want to add in the Event Name field.

   > **NOTE:** Each event in your project must have a unique Event name.

3. Optionally type a description for the event.

4. If adding one or more locations, click the **Locations** drop-down and select the check boxes next to the locations you want to add to the event. The locations in the list are created when you configure locations.

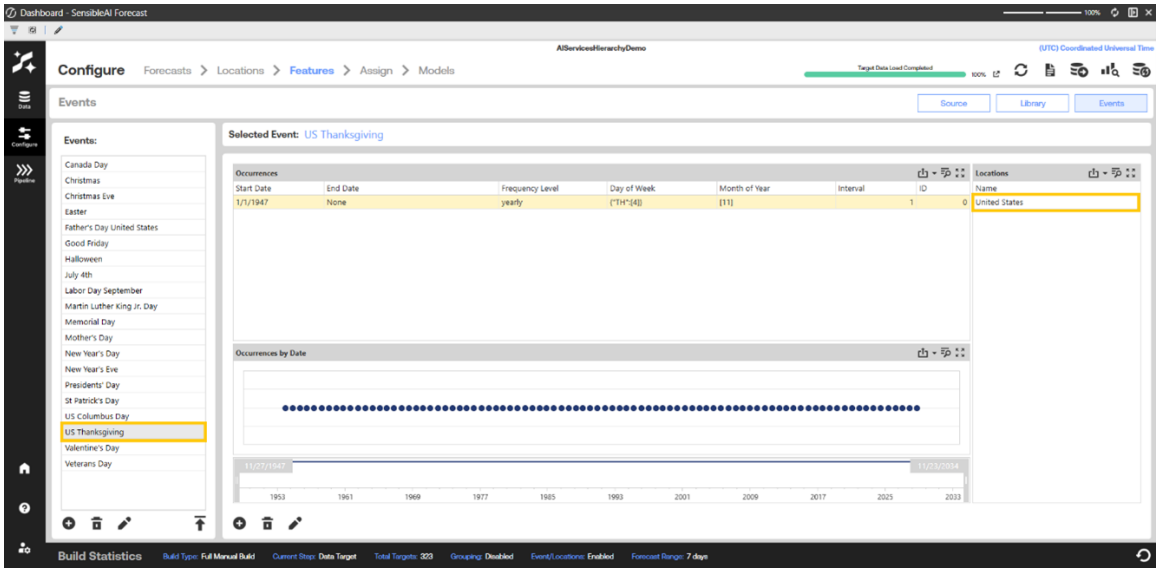


5. In the Scenario Modeling Event field, select **Yes** if the event should be included when defining custom Scenarios in Utilization. Otherwise, select **No**.

> **NOTE:** Only select **Yes** for Scenario Modeling Feature if the intention of the project is to run predictions on different Scenarios. If **Yes** is selected for any event, the project will be considered a Scenario Modeling project by the Xperiflow Engine.

6. Click **Save**. A message box informs you that the event has been added.

7. Click **OK** to add the event to the Events pane. Any locations you added to the event display in the Locations pane.



Add Events Using an Events Package

Event packages are a useful time saver when defining the events you want to assign to targets. Each event added from event packages includes an event occurrence rule.

Events that fall on the same date in a year include the appropriate specific occurrence rule. Events that fall on relative day in a year include a relative date occurrence rule (for example, the Thanksgiving U.S. holiday falls on the fourth Thursday in November).

You can add multiple events at once this way, then delete or edit individual events from the package.

To add an event package:

1. In the **Add Event** dialog box, select **Package**. A list of pre-configured event packages displays.
2. Click the check boxes for each event package whose events you want to add to your events list.

3. In the Scenario Modeling Event field, select **Yes** if the event should be included when defining custom Scenarios in Utilization. Otherwise, select **No**.

> **NOTE:** Only select **Yes** for Scenario Modeling Event if the intention of the project to run predictions on different Scenarios. If **Yes** is selected for any event, the project will be considered a Scenario Modeling project by the Xperiflow Engine.

4. Click **Add**.

5. A message box informs you that the events package has been added. Click **OK** to close the message box and add the events from the selected packages to the events list.

> **NOTE:** If an event already exists and an event package is added with the same name, the two events will be merged. The merging functionality makes a super set of all occurrences and locations between the two versions of events.
>
> Adding an event package may also add locations associated with the event package if they don't exist in the project. They are added with the special engine suffix `created`. See Add Locations for more information.

Upload an Event or Location File

You can upload an existing event or location file in CSV format that contains your company's events or location information. This saves time over having to manually enter events in the **Events** page.

> **TIP:** This functionality also exists in the **Events** page (Manage section) in the Utilization phase.

> **NOTE:** SensibleAI Forecast also lets you upload a locations file or a mapper file for events or locations. Use this procedure to upload files of those types as well. Uploading a locations file saves time over having to use the Locations page to manually add the locations that you want to assign to targets. Uploading a events or locations mapper file saves time over having to manually assign events or locations to targets on the **Assign** page.

Upload Types

You upload various types of data and event or location data mappings in a `.csv` file. The following describes the different types of uploads that can be done.

Event Upload

This upload lets you add various events and occurrences using a `.csv` file upload. If an event from the file already exists in the project, the occurrences in the file are added to the already existing event.

> **NOTE:** Occurrences uploaded through an event upload are created as single day occurrences with no occurrence rules. Occurrences are expected to be in Month/Day/Year format.

Column Definitions:

EventName: Name of event to create or add occurrences to.

Occurrence: A single date the event occurred.

Example:

	A	B	C
1	EventName	Occurrence	
2	Happy Hour	1/5/2023	
3	Happy Hour	1/6/2023	
4	Happy Hour	1/7/2023	
5	Closed	12/25/2022	
6	Closed	12/25/2021	
7	Closed	12/25/2023	

If no **Happy Hour** or **Closed** events exist in the project, their three occurrences are added to the existing events. Otherwise the events are created with their three occurrences listed in the file.

Location Upload

This upload lets you create various locations through a `.csv` file upload. The job fails if any of the below cases are encountered:

- LocationName already exists in the project.
- LocationAddress maps to a well-formatted address already in the project.

> **NOTE:** Use as specific an address as possible. For example, an address such as **Rochester** may lead to unexpected results because the state is not specified, (Rochester could mean Rochester, New York, Rochester, Michigan, Rochester, Minnesota).

Column Definitions:

LocationName: Name of the location to create.

LocationAddress: Address of location to create.

> **Example:**

LocationName	LocationAddress
Little Caesars Arena	2645 Woodward Ave, Detroit, MI 48201
White House	1600 Pennsylvania Avenue NW Washington, D.C. 20500
The Big House	1201 S Main St, Ann Arbor, MI 48104

This creates three locations (Little Caesars Arena, White House, The Big House) with the associated location addresses.

Event Target Mapper Upload

This upload lets you assign various events to targets using a `.csv` file upload. Any prior event assignments are preserved. Only new assignments are created.

> **NOTE:** If an EventName column does not exist in the project, a warning message is written to the AI Services log, but the job continues. If a TargetName column does not correspond to an existing target in a train state (in model build) a warning message is written to the AI Services log, but the job continues.

Column Definitions:

EventName: Name of the event to assign to the associated target.

TargetName: The full target name to map to assign an event to.

Example:

EventName	TargetName
Happy Hour	[UD1]Lunch~[UD2]Alcohol
Happy Hour	[UD1]Dinner~[UD2]Alcohol
Closed	[UD1]Lunch~[UD2]Burgers

This creates three new event assignments. You can see these assignments on the Model Build phase **Assign** page. The **Happy Hour** event is assigned to two targets (`[UD1]Lunch~[UD2]Alcohol` and `[UD1]Dinner~[UD2]Alcohol`) and the **Closed** event is assigned to one target (`[UD1]Lunch~[UD2]Burgers`). Any prior event assignments are preserved. Only new assignments are created.

Event Target Dimension Mapper Upload

This upload lets you assign events to targets based on dimensions using a `.csv` file upload. Any prior event assignments are preserved. Only new assignments are created.

NOTE: If an EventName column does not exist in the project, a warning message is written to the AI Services log, but the job continues. All target dimensions must be included in columns. Leave values in the column blank if not mapping to the dimension.

Column Definitions:

EventName: Name of the event to assign to the associated target.

TargetDim1*: Targets with this value in TargetDim1* along with other dimension values are assigned this event. Values can be left blank to assign to targets regardless of this dimension.

TargetDim2*: Targets with this value in TargetDim2* along with other dimension values are assigned this event. Values can be left blank to assign to targets regardless of this dimension.

Replace TargetDim1, 2 through n with the actual target dimension name, such as UD1, UD2, Scenario, or Category.

Example:

EventName	UD1	UD2
Happy Hour		Alcohol
Closed	Lunch	Burgers
Christmas		

This example assumes the below image is the target data set with only two target dimensions (UD1 and UD2). The target data set has four targets.

Date	UD1	UD2	Value
1/1/2020	Lunch	Alcohol	100
1/1/2020	Dinner	Alcohol	50
1/1/2020	Lunch	Burgers	200
1/1/2020	Dinner	Burgers	88

The upload file assigns the **Happy Hour** event to all targets with the UD2 dimension as `Alcohol` (`[UD1]Lunch~[UD2]Alcohol and [UD1]Dinner~[UD2]Alcohol`). It assigns the **Closed** event to all targets with UD1 as `Lunch` and UD2 as `Burgers` (`[UD1]Lunch~[UD2]Burgers`), and assigns **Christmas** to all targets since all target dimensions are blank.

These new event assignments can be seen on the configure assign page of the model build section of SensibleAI Forecast. Any prior event assignments are preserved. Only new assignments are created.

Event Location Mapper Upload

This upload lets you assign various locations to events using a `.csv` file upload. Any prior event-location assignments are preserved. Only new assignments are created. Event location assignments are useful when running the Auto Assign job to assign events to targets.

> **NOTE:** Use as specific an address as possible. For example, an address such as **Rochester** may lead to unexpected results because the state is not specified, (Rochester could mean Rochester, New York, Rochester, Michigan, Rochester, Minnesota). If an EventName column does not exist in the project, a warning message is written to the AI Services log, but the job continues. If a LocationAddress column does not correspond to an existing target in a train state (in model build) a warning message is written to the AI Services log, but the job continues.

Column Definitions:

EventName: Name of the event to assign LocationAddress to.

LocationAddress: Address of the location to assign to the event.

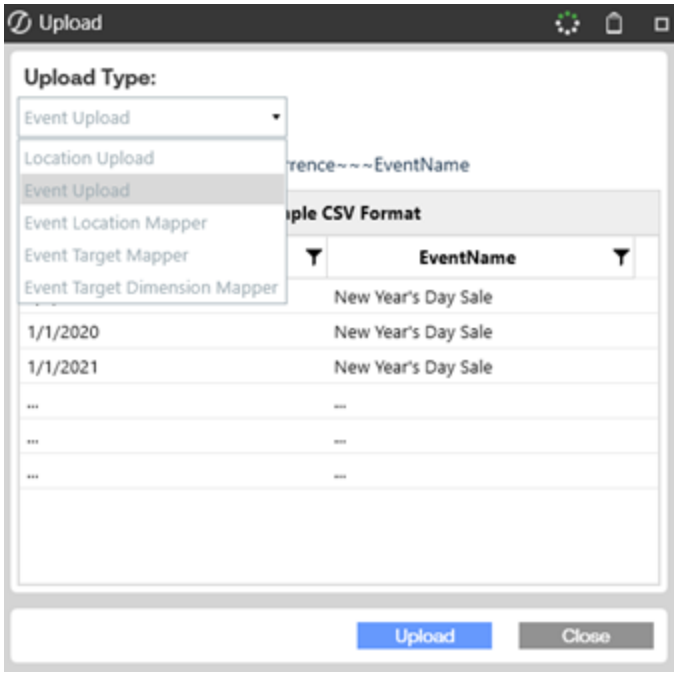
> **Example:**

EventName	LocationAddress
Happy Hour	2645 Woodward Ave, Detroit, MI 48201
Closed	1600 Pennsylvania Avenue NW Washington, D.C. 20500
Christmas	1201 S Main St, Ann Arbor, MI 48104

This adds new locations to the specified events. You can see this on the **Events** page. The Happy Hour event has the 2645 Woodward location. The Closed event has the 1600 Pennsylvania Ave location. The Christmas event has the 1201 S Main St. location. Any prior locations added to these events are preserved, only new locations are added to the events.

Upload an Event File

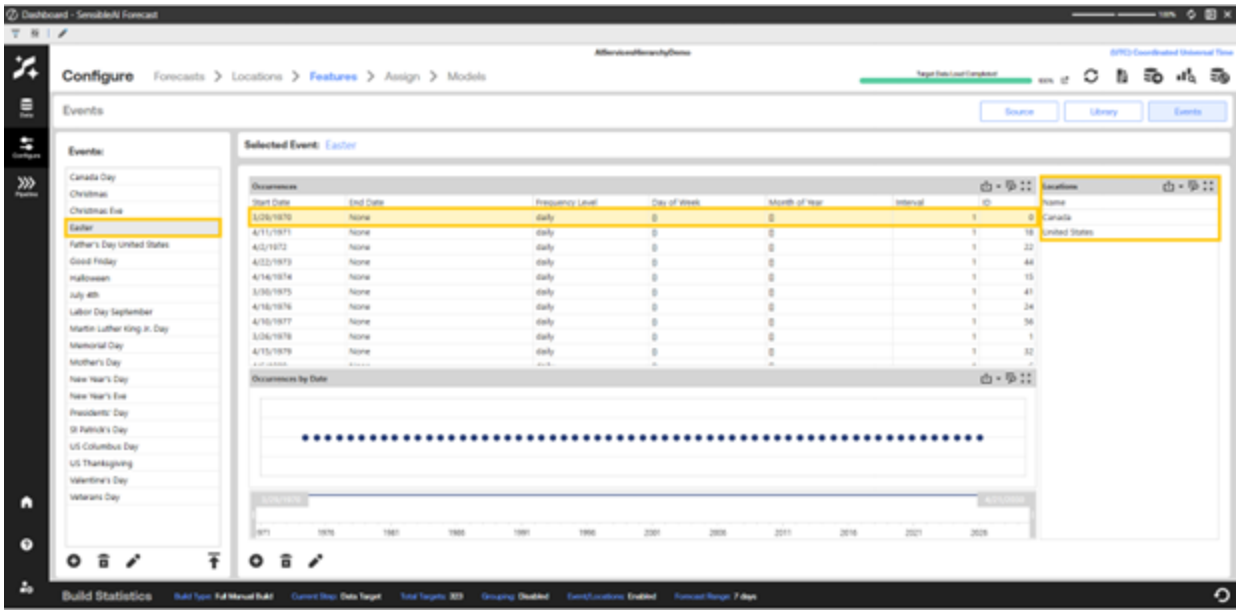
1. Click the **Upload** button at the bottom of the Events pane. The **Upload** dialog box displays.
2. In the Upload Type field, select the type of file you want to upload, or use the default **Event** upload type.



3. Click **Upload**, then use the Windows **Open** dialog box to navigate to and select the file you want to upload.

4. Click **OK**. SensibleAI Forecast queues the job to upload the selected file. The Windows **Open** dialog box closes when the job completes.
5. Click **Update** to add the information from the file to the appropriate page.

The events defined in the events file display in the Events pane. Information for the event selected in the Events pane displays in the Occurrences pane and the Occurrences by Date pane maps the event occurrences for the selected event. Any locations for the selected event that were in the uploaded events file display in the Locations pane.



Delete an Event

You can delete any event that is not assigned to a target that has started or completed the pipeline job. If unsure that an event can be useful, the machine learning engine can determine if the event is useful for the model.

When you delete an event, any associated occurrence rules for the event are also deleted. If the event is currently assigned to a target that has not been through the pipeline job, then the associated assignment is also deleted. This applies to any event, whether it was added manually, from an events package, or as part of an events file.

To delete a single event:

1. Select the event from the Events pane and click **Delete** .
2. A message box lists the selected event and asks to confirm the deletion. Click **Delete**.

3. Click **OK** to close the message box and delete the event.

To delete multiple events:

1. Click **Delete**on the **Events** pane.
2. In the **Delete Events** dialog box, select the **Multiple Events** option.
3. Select check boxes for all events that are to be deleted. Click **Delete**.

4. Click **OK** to close the message box and delete the selected events.

> **NOTE:** You can delete all events if the pipeline job has not run yet.

Add an Occurrence Rule to an Event

To add an occurrence rule, in the **Events** pane, select the event you want to add the occurrence to and click **Add a New Occurrence** ⊕ . Then use the **Add Occurrence Rule** dialog box to add a specific occurrence rule or a relative occurrence rule.

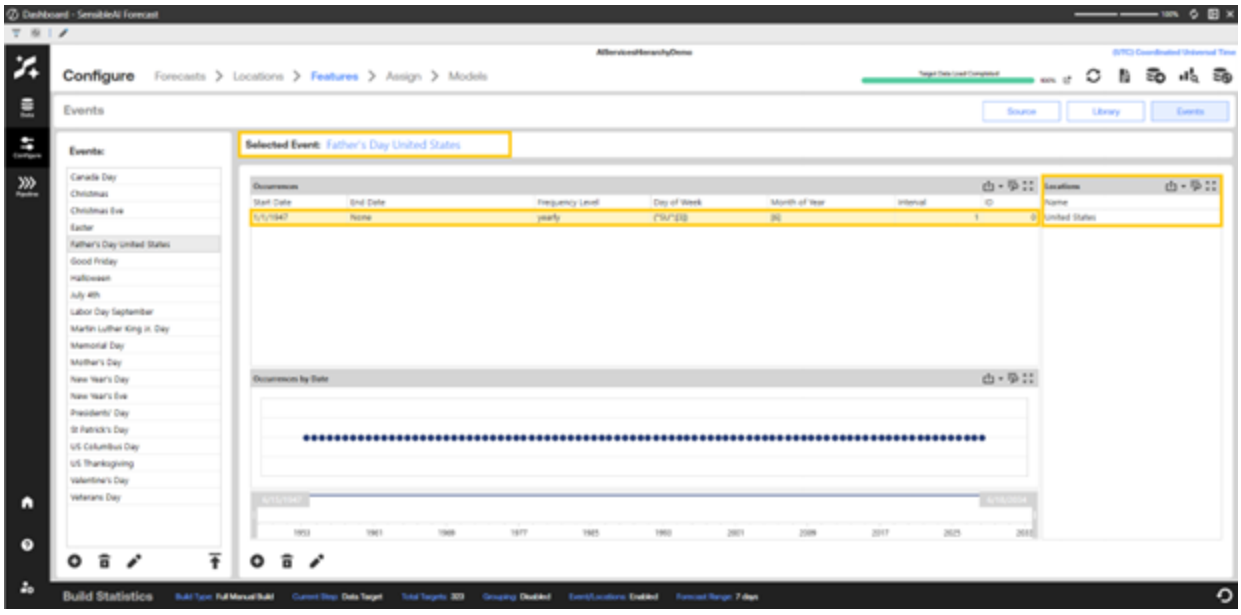
Add a Specific Occurrence Rule

1. In the **Add Occurrence Rule** dialog box, click **Specific**.
2. Select values in the Date Start fields to specify the Month, Date and Year of the event.
3. In the Date End field, do one of the following:
    - Select **None** to have the occurrence fall on the same day of each year.

- Select **Custom**, then use the Month, Date, and Year fields to set the last date you want the occurrence rule to apply to.

4. In the Interval field, select the rule's interval. The interval is how many steps of the frequency are between the occurrences. For example, an interval of 2 with a frequency of yearly will be once every other year.
5. In the Frequency field, select the rule's frequency from the list (Yearly, Monthly, Weekly or Daily).
6. Click **Save**. A message box informs you that the events package has been added.
7. Click **OK** to close the message box and the **Add Occurrence Rule** dialog box, and add the occurrence rule to the events list.

The occurrence rule displays in the selected event's list of occurrences, and the **Occurrences by Date** chart updates to include the added occurrence.

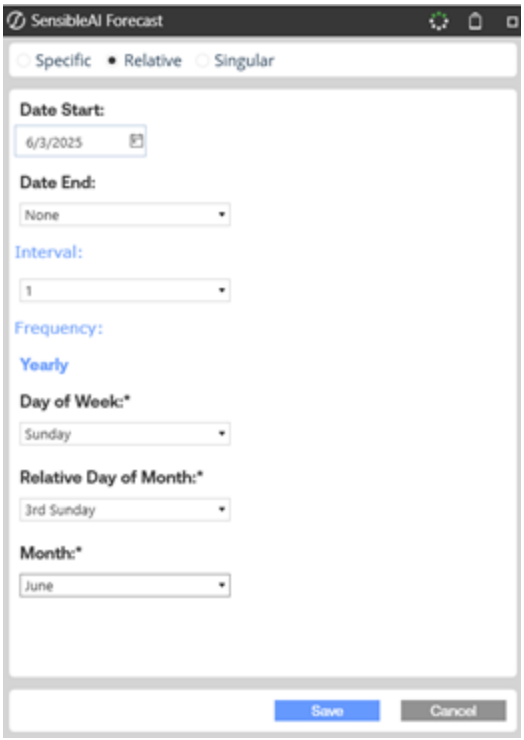


Add a Relative Occurrence Rule

1. In the **Add Occurrence Rule** dialog box, click **Relative**.
2. Select values in the Date Start fields to specify the Month, Date and Year of the event.
3. In the Date End field, do one of the following:
    - Select **None** to have the occurrence fall on the same day of each year.
    - Select **Custom**, then use the Month, Date, and Year fields to set the last date you want the occurrence rule to apply to.

      > **Example:** If the occurrence rule is for a multiple-day event that lasts from October 1st to October 4th each year, set the date start to Month=10, Day=1, and year to the specific year for the occurrence. Then set the date end to Month=10, Day=4, and year to the specific year for the occurrence.
4. In the Interval field, select the rule's interval.
5. In the Day of Week field, select the day of the week on which the event falls.
6. In the Relative Day of Month field, select the month in which the event falls.
7. Click **Save**. A message box informs you that the events package has been added.
8. Click **OK** to close the message box and the **Add Occurrence Rule** dialog box, and add the occurrence rule to the events list.

The following graphic shows a relative occurrence rule set up for Father's day in the U.S., which occurs on the third Sunday in June of each year.



Add a Single Occurrence Rule

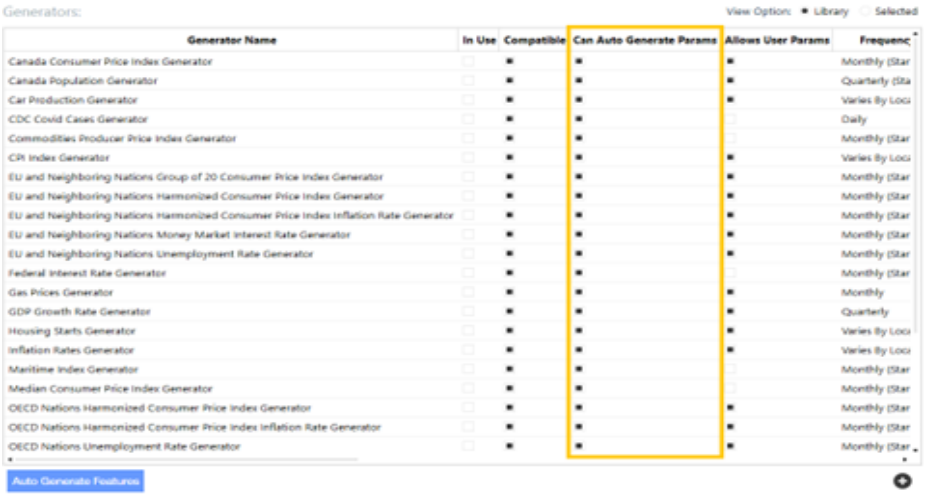
1. In the **Add Occurrence Rule** dialog box, click **Single**.
2. Select the single date of occurrence for the event.
3. Click **Save**. A message box informs you that the events package has been added.
4. Click **OK** to close the message box and the **Add Occurrence Rule** dialog box, and add the occurrence rule to the events list.

The following graphic shows a single occurrence rule set up for New Year's Day for 2000.

Once you have created and configured all the events for your project, you can assign the generators and locations you have configured for your project.

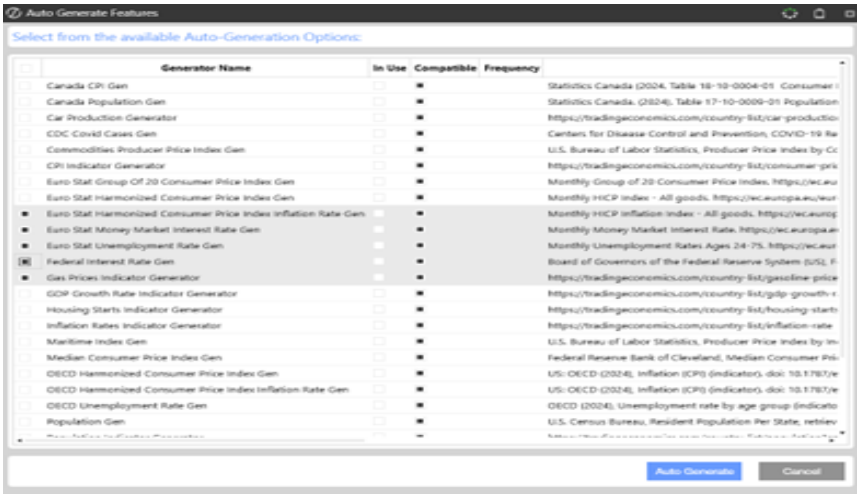
Auto Generate Features

Some generators support the automatic creation of configurations based on the user's project configuration and dataset. Features that are available for auto-generation are flagged in the **Can Auto Generate Params** field in the Library view of the Generators pane.



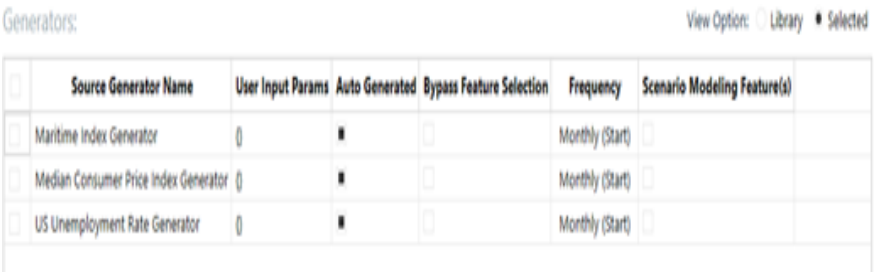
To use **Auto Generate Features**:

1. Click the **Auto Generate Features** button.
2. Select which generators to auto-generate.



3. Click the **Auto Generate** button.
4. The **Job Progress** dialog box appears and can be closed at any time.

5. Once the **Auto Generate Feature** job is completed, the selected generators can be seen in the **Selected** pane.



Assign Generators and Locations

Use the **Assign** page to map events and locations to targets. This page has a Project view and Target view.

Any location dimension specified on the **Targets** are automatically assigned locations to your targets. Use the Assign page to map events and locations to targets for use during Pipeline, where these events and locations can generate predictive features or serve as predictive features themselves.

> **NOTE:** There is a limited number of locations and events that can be assigned to any individual target.

You can map locations and events to targets in these ways:

Auto Assign: All events are assigned to targets based on their locations. If any of the target's locations are geographically within any of the event's locations, the event is assigned to the target. A target without a location, whether removed on the Target view or not present, cannot have events assigned to it through a shared location. Click **Auto Assign** to assign events based on the currently applied locations for all targets. See Auto-Assign Generators and Locations.

> **NOTE:** You must have at least one location in the Target view mapped (selected and applied) to use Auto Assign. The Auto Assign job uses the selected locations in order to automatically map Generators which contain the selected location(s).

Manual Assignments: In the Target view, you can assign events and locations to a target using arrow buttons and clicking **Apply**. Selecting **Apply to All** applies whatever mappings are present for the currently selected target to all other targets in the data set. See Assign Generators and Locations to a Specific Target.

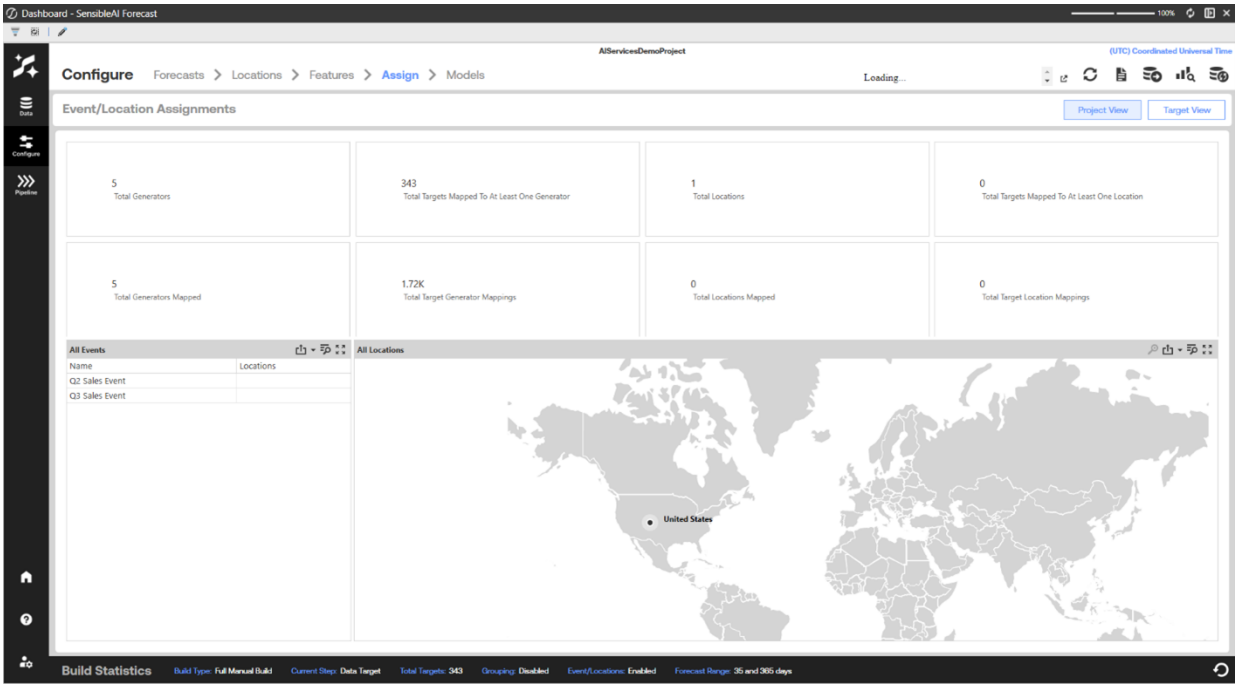
Event Target Mapper File Upload: Upload a file that maps events to target names. See Upload an Event File.

Event Target Dimension Mapper File Upload: Upload a file that maps events to targets based on dimensionality. See Upload an Event File.

View Project Location and Generator Information

Use the Project view to see the number of:

- Generators and locations currently being used by your model.
    - Generators/Locations mapped to at least one target.
    - Total Generators/Locations Mapped
    - Total Target-Generators/Location Mappings
- Locations mapped to at least one target.
- A list of all events defined in your project. See Configure Library - Features.
- The interactive map showing all locations defined in your project. This is the same interactive map on the **Locations** page.

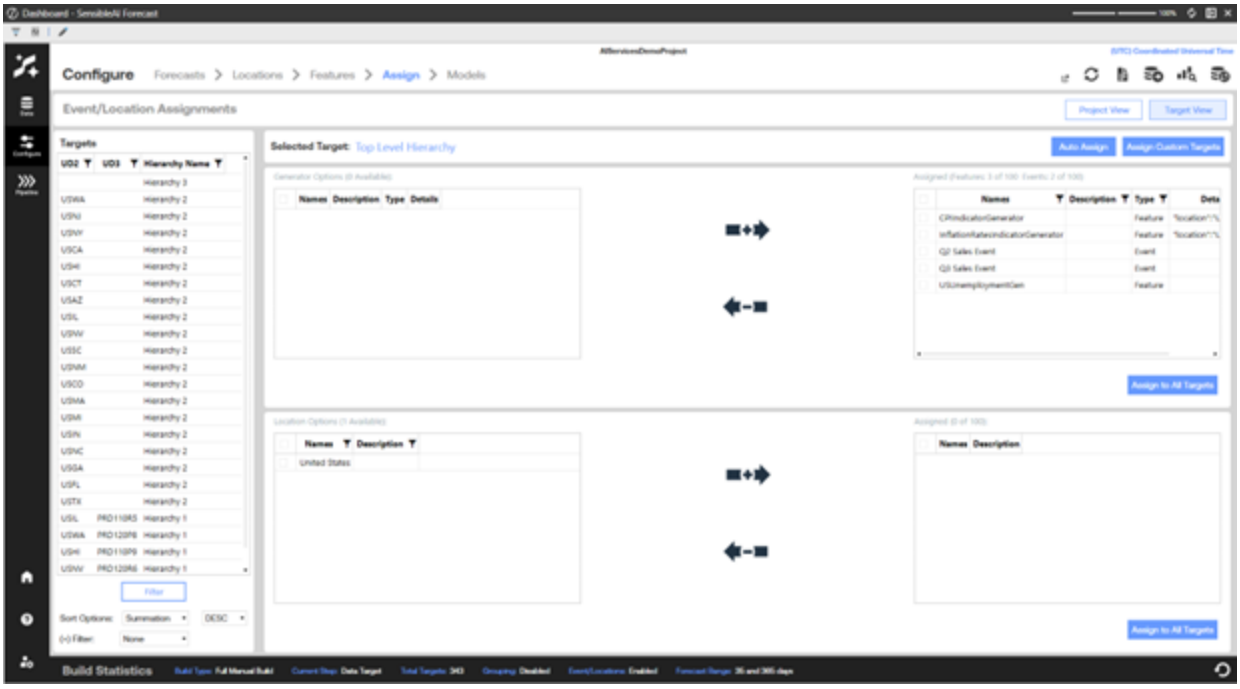


View Generator and Location Assignments by Target

For the target selected in a list on the left, you can add all the events and locations added in the previous pages to be associated with the target. To do this, click the plus arrow to add them. Click the minus arrow to remove them.

In Target view, you can see the events and locations assigned to each target.

> **TIP:** When a location dimension is selected for the target data set, the respective location for each target displays in the selected locations pane.

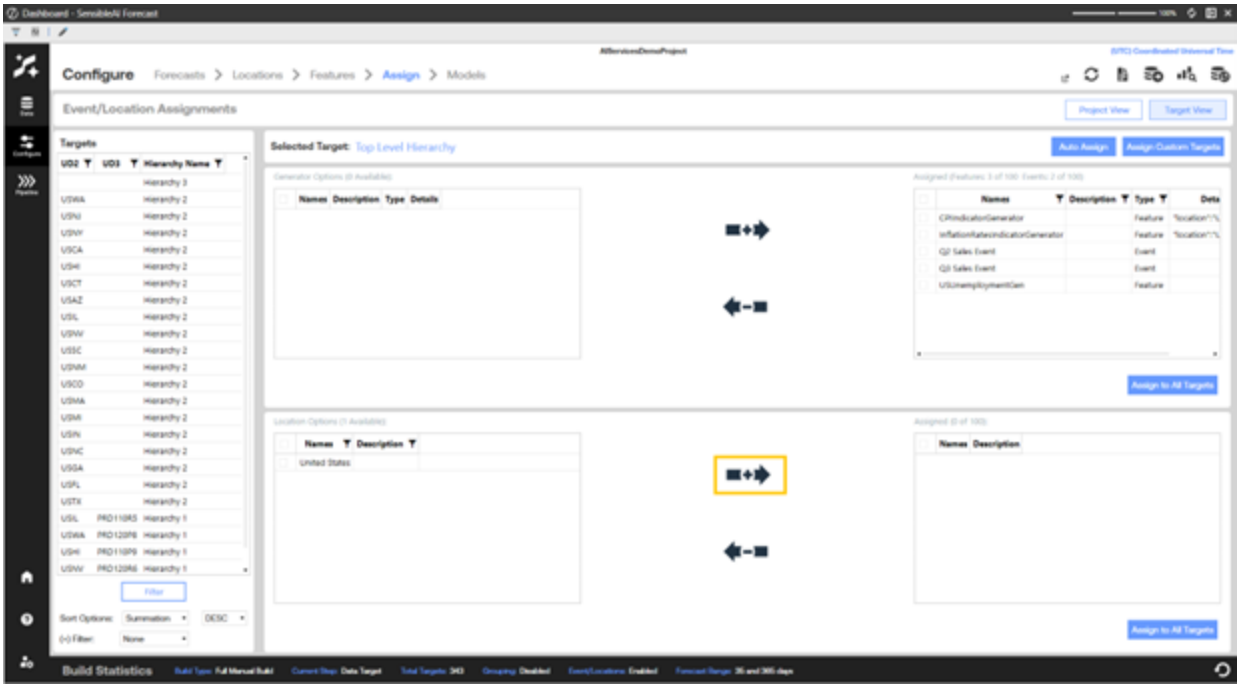


Assign Generators and Locations to a Specific Target

Assigning events and locations to a target allows you to use inputted information in the modeling process. Generators and locations can be assigned to a single target or to all targets in the model build.

1. If in Project view, click **Target View** to switch to the **Assign** page Target view.
2. In the Targets pane, select a target to which you want to assign events and locations. The Selected Target pane shows the target name and all available events and locations that can be assigned.
3. In the Generator Options table, select check boxes next to the generators you want to assign to the target, or select the check box in the table header row to select all available events.

4. Click the **Select the Checked Generators** button to move the selected generators from the Event Options table to the Selected table.
5. In the Location Options table, select check boxes next to the locations you want to assign to the target, or select the check box in the table header row to select all available locations.
6. Click the **Select the Checked Locations** button to move the selected locations from the Location Options table to the Selected table.
7. The add and remove buttons will only apply the events and locations to the currently selected target. To apply these changes to all targets:
    - Click **Apply to All** to apply the current event and location assignments to the selected target.
8. Select another target in the Targets and repeat steps 3 through 7 to make assignments for that target. Repeat until you have assigned generators and locations for all targets in the project.



> **TIP:** To deselect events or locations, select them from the respective Selected table and click Remove to remove the selected events or locations from the selected lists.

Auto-assign Generators and Locations to Targets

Auto Assign uses the selected locations to automatically map Generators that contain the selected locations.

1. Click **Target View** to assign one or more locations from the Locations Options list to the Selected list.

   > **NOTE:** To run Auto Assign, you must have at least one location under Selected Locations (both selected and applied) on the Target View page.

2. Click **Auto Assign**. A message box asks you to confirm that you want to run Auto Assign to assign events to targets based on their locations.
3. Click **Run**. The auto assign job posts to the job queue.
4. Click **OK** to close the jobs message box. The **Job Progress** dialog box displays.
5. Click **Close** at any time to close the **Job Progress** dialog box. The number of total mapped events and locations updates in the **Assign** page Project view.

After auto-assigning events and locations or assigning events and locations to specific targets, you can move to the **Forecast** page to set forecast ranges.

Configure Custom Target Feature and Location Assignment

Allows a user to filter a specific group of targets and apply selected generator or location assignments to those filtered targets. Once completing the dialog, the configuration of assignments for whichever target is currently selected on the Target View page is applied to the filtered targets.

1. Click **Target View** to view all targets for the project

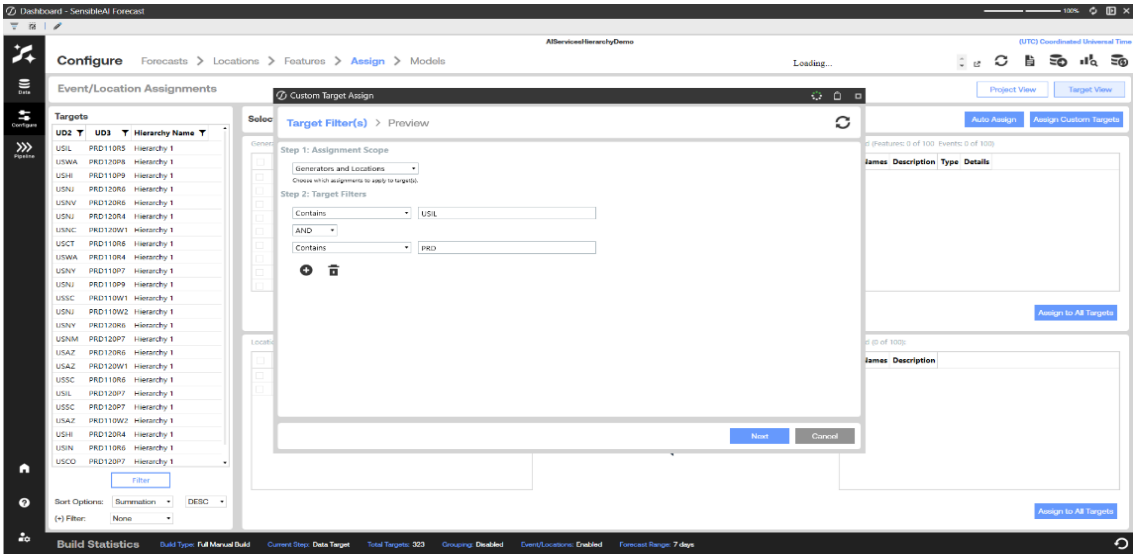
   > **NOTE:** Features and Locations that will be applied to your filtered projects will be from the current selected target on the Target View Page.

2. Click **Assign to Custom Targets**. A dialog will appear allowing you to filter down targets that you want to apply features and locations to.
3. Under **Step 1: Assignment Scope** select either **Generators**, **Locations**, or **Generators and Locations** from the drop-down menu to choose which assignments you would like to apply to your filtered targets.
4. Under **Step 2: Target Filters** select either **Contains**, **Equals**, or **Not Equals** from the drop-down menu to begin your target filter.
   a. In the text box to the right of the drop-drown menu, write out the text that your target either contains, is equal to, or is not equal to.
5. To apply an additional filter, click the **Add Filter** button.

a. Another filter will appear with an extra drop-down allowing you to select either **AND** or **OR**.

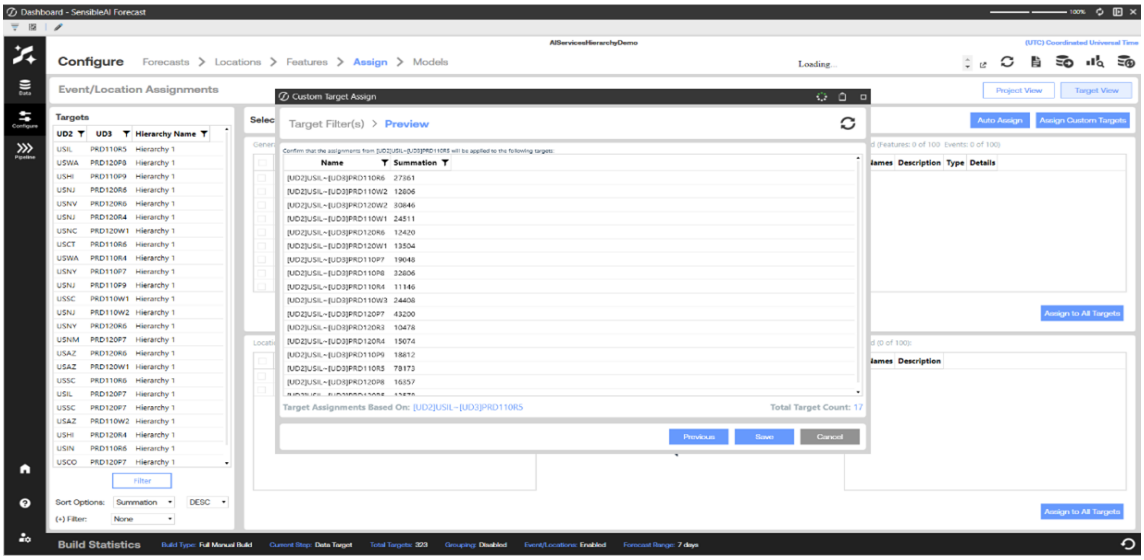
> **NOTE:** By selecting **AND**, it will chain the second filter to the first filter. By selecting **OR**, it will return the first filter results along with the second filter results.

6. To remove a filter, click the **Remove Filter** button.



7. Once all filters have been added and configured, click **Next**.

8. All filtered targets will appear on the **Preview** page of the dialog. Confirm that the targets have been filtered correctly and click **Save** to apply assignments to the list of filtered targets.



Set Modeling Options

The **Model** page in the Configure section lets you set various options that manage how the SensibleAI Forecast engine runs your modeling pipeline. These parameters tweak processes related to feature transformation and modeling.

Use this page to configure the model settings that run in the model pipeline. Adjusting these settings for your modeling project is optional. You can accept the default settings and run the pipeline.

View Options

View options on the **Model** page let you set options for the entire project or create different options for each target in the model build.

Set Project-wide Model Options

The Project View lets you set options that apply to all targets.

1. Set a value for **Train Intensity**. This parameter varies between 1 and 5. It is a scale of how many hyperparameter tuning iterations you want to occur during training. The higher the training intensity number, the more iterations are performed to find the optimal hyperparameters, which potentially leads to increased accuracy. However the run time for the modeling job increases with a higher intensity.

   Typically, training intensities between 3 and 5 prove to achieve both high-quality accuracy and reasonable computing run times.

2. Set a value for **Deployment Strategy**. This specifies which models to deploy after the model training phase. Select Auto if you want SensibleAI Forecast to determine the most effective deployment strategy for your project.

   > **NOTE:** Selecting Top Model or Best 3 Models for the deployment strategy treats targets in groups as a whole, not as individual targets.

3. Set a value for **Allow Negative Targets**. The default setting of **False** ensures that SensibleAI Forecast does not predict values below zero for the project.

4. Set a value for **Evaluation Metric**. Specify the error metric that evaluates model accuracy during training and testing. See Appendix 3: Error Metrics for an error metric list and details.

5. Set a value for **Clean Missing Method**. This setting determines the method used to handle missing data values during data cleaning. Values are:

   Interpolate: Fills in missing data using linear interpolation between the surrounding non-missing data points. For example [1.0, nan, 3.0, nan, -5.0] becomes [1.0, 2.0, 3.0, -1.0, -5.0]

   Kalman: Fills in missing data using a Kalman filter.

   Local Median: Fills in missing data using the median of the past *n* and future *n* days from the missing data point where *n* depends on the frequency. For example, (n=2) [1.0, nan, 2.0, 3.0, nan, 6.0] becomes [1.0, 2.0, 3.0, 3.0, 6.0]

Mean: Fills in missing data using the mean value of the non-missing data. For example [1.0, nan, 3.0, nan, 5.0] becomes [1.0, 3.0, 3.0, 3.0, 5.0]

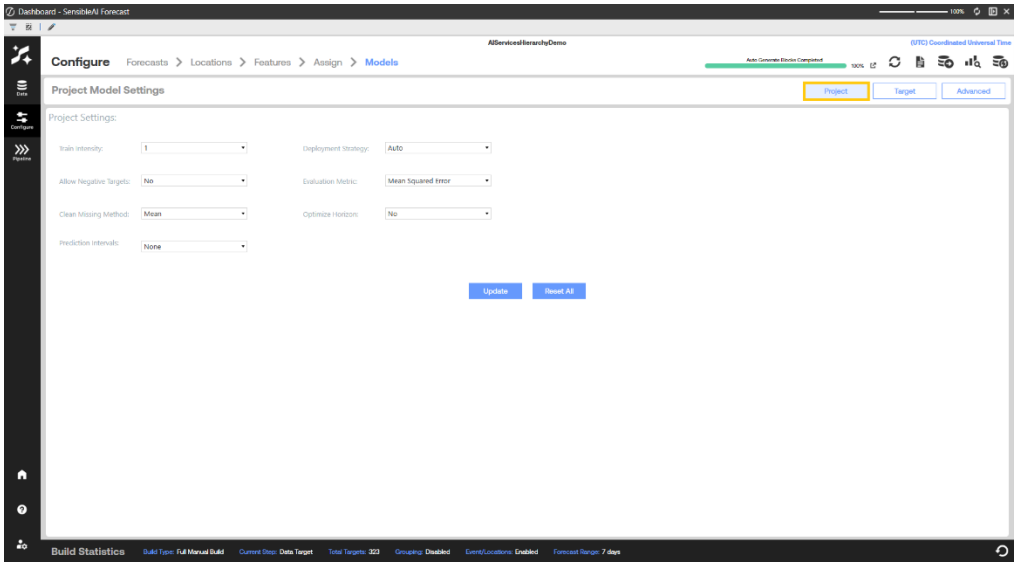
Zero: Fills in missing data with 0. For example [1.2, nan, 3.0, nan, nan] becomes [1.2, 0, 3.0, 0, 0]

6. Set a value for **Optimize Horizon**. Set to **Yes** to split the test portion of the largest split into smaller chunks equal to the forecast range (or only 10 chunks, if the amount of chunks created by separating the split into forecast range sized chunks is more than 10).

   These chunks are used in the model selection stage to perform individual predictions on each chunk. This results in a better test of which model performs best on the given forecast range, but also results in longer run times.

7. Set a value for **Prediction Intervals**. If set to a percentage, prediction intervals run for models and give a confidence band for predictions and back tests of the selected percentage.

8. Set a value for **Compute Intervals for**. This setting only displays if the prediction interval value is set to something other than **None**. It specifies which models' prediction intervals are calculated in the prediction. Back tests compute prediction intervals for all models (if prediction interval value is set to something other than **None**).



9. Set a value for **Model Ranking**. This setting is only for projects that have a grouping strategy configured. Select from the following options:
    - **Targets**: Calculates the model ranking for each individual target in a project.
    - **Groups**: Calculates the model ranking for each group of targets in a project.
10. When you have made your settings, click **Update** at the bottom of the Project Settings pane.
11. A message box informs you that project level settings have been updated. Click **OK** to close the message box.

> **NOTE:** Targets set to **Project** use these setting. Targets set to **Advanced** do not use these settings unless reset to **Project**.

> **TIP:** Click **Reset All** to reset all targets to use the project level settings.

Set Model Options for Each Target

The Targets view lets you select advanced settings on a target-by-target basis. This provides finer-grained modeling during the pipeline run, but can increase the run time of the pipeline job.

To use the advanced settings:

1. In the Targets pane, select a target, then use the drop-downs to change setting values for the selected target or group. Settings are as follows:

   Models: Specify the models offered by SensibleAI Forecast to be trained for a selected target.

   Evaluation Metric: This is the same as the Project view.

   Tuning Strategy: Select the strategy by which to tune hyperparameters during training.

   Tuning Iterations: This is the same as Training Intensity on the Project view.

   Clean Missing Method. This setting determines the method used to handle missing data values during data cleaning. Values are:

   - **Interpolate**: Fills in missing data using linear interpolation between the surrounding non-missing data points. For example [1.0, nan, 3.0, nan, -5.0] becomes [1.0, 2.0, 3.0, -1.0, -5.0]
   - **Kalman**: Fills in missing data using a Kalman filter.
   - **Local Median**: Fills in missing data using the median of the past *n* and future *n* days from the missing data point where *n* depends on the frequency. For example, (n=2) [1.0, nan, 2.0, 3.0, nan, 6.0] becomes [1.0, 2.0, 3.0, 3.0, 6.0]
   - **Mean**: Fills in missing data using the mean value of the non-missing data. For example [1.0, nan, 3.0, nan, 5.0] becomes [1.0, 3.0, 3.0, 3.0, 5.0]

- **Zero**: Fills in missing data with 0. For example [1.2, nan, 3.0, nan, nan] becomes [1.2, 0, 3.0, 0, 0]

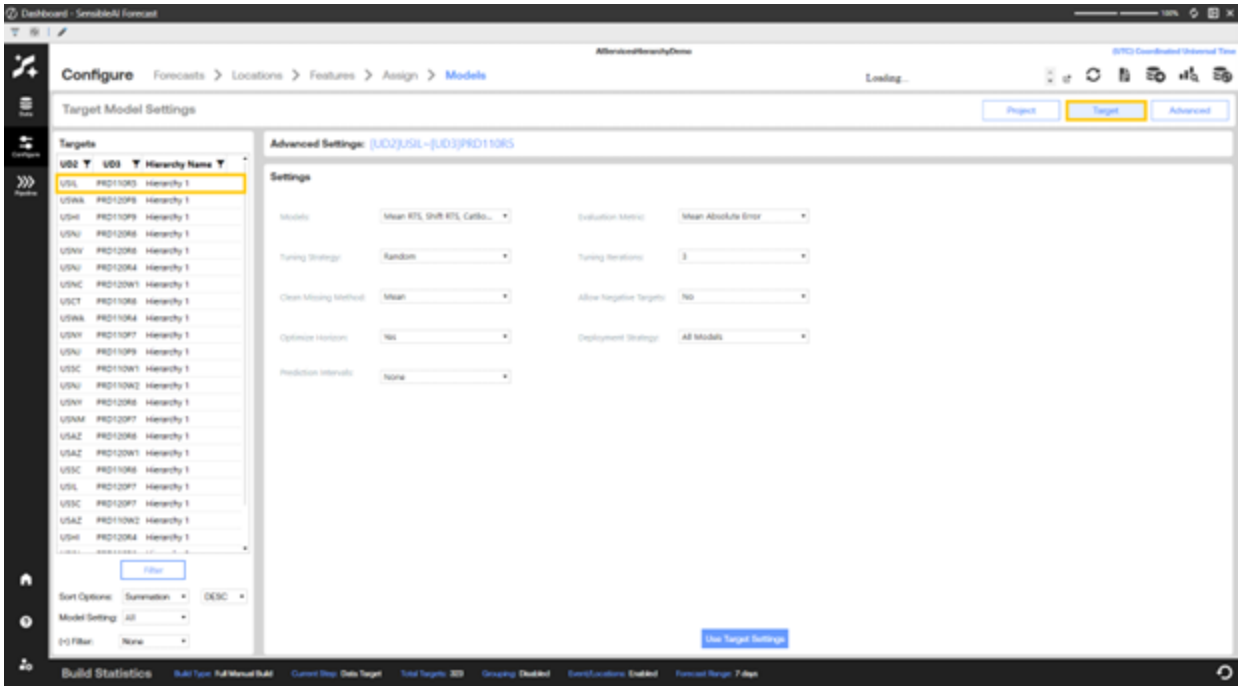
Allow Negative Targets: This is the same as the Project view.

Optimize Horizon: This is the same as the Project view.

Deployment Strategy: This is the same as the Project view.

Prediction Intervals: This is the same as the Project view.

Compute Intervals for: This is the same as the Project view.



2. Click **Use Advanced** at the bottom of the Settings pane. A message box notifies you that the selected advanced settings will be used for the selected target.
3. Click **OK** to close the message box. Sensible Machine learning then uses your advanced settings for that target when running the pipeline job.

4. Repeat the previous steps for any other targets in your project.
5. Once advanced settings are made for a target, if you want to change those advanced settings, use the previous steps to select the target and change values for each setting, then click **Update** at the bottom of the Settings pane.

> **TIP:** Click **Use Project** at the bottom of the Settings pane to use the values set in the Project view for the selected target. Click **Refresh** to see the targets with the Advanced model settings in the Targets pane.

Set Advanced Settings

The Advanced view lets you select advanced settings for the project.

> **IMPORTANT:** This is an advanced page and should only be used by users with a deeper understanding of cross-validation and modeling techniques.

Configure Cross Validation

The Cross Validation Settings pane sets the desired cross-validation configuration for the model build.

The first selection must be the cross-validation type which is one of the following:

Total Splits: Set the total number of splits to be used in the model build.

Custom Dates: Add rows to the table editor using the **Add +** button. Each split number must have every data set type listed as **True** on the Saved settings pane on the right. Data set types for a given split cannot overlap and must be contiguous. Each row must contain the following information:

- **Split Number**: The split to which the row's information corresponds.
- **Dataset Type**: The data set type for which the row belongs.
- **Start (Datetime)**: The start date of the split portion. The range of valid dates is displayed at the top of the table.
- **End (Datetime)**: The end date of the split portion. The range of valid dates is displayed at the top of the table.

Custom Percentiles: The same as the Custom Dates settings but using start and end percentages for the data set portions instead of dates. The percentages are decimals that can be on or after zero and before or on one. For example 0.1 thru 0.8 for each split number.

> **NOTE:** For custom dates and custom percentiles, the data set type column values should be in this order for each split number: **Train**, **Validation**, **Test**, **Holdout**.

Default: The default settings recommended by SensibleAI Forecast based on the data set.

Has Holdout: Can be configured to not use a holdout set based on the Boolean value selected.

Configure Avoidance Periods

The Avoidance Periods table contains portions of the data set that should not be used when evaluating metrics for model performance.

To add avoidance periods:

1. Click the **Add +** button.
2. Enter a start and end date.
3. Click **Save** on table editor.
4. Click **Save** after configuring cross-validation settings and avoidance periods.

Settings are validated. If settings are determined to be valid, they are saved and the page refreshes with newly saved settings on right pane. If settings are invalid, a message displays with the validation errors of the configured settings and avoidance periods.

View Saved Settings

The Saved Settings pane shows what the current cross-validation settings are for the model build. It includes the following:

Current Splits: This chart shows the different test, train, validation, and holdout portions and which dates from the data set belong to each portion.

Has Holdout Set: Indicates if the cross-validation strategy has a holdout set (True, False). This is dependent on the number of dates in the data set.

Has Test Set: Indicates if the cross-validation strategy has a test set (True, False). This is dependent on the number of dates in the data set.

Has Training Set: Indicates if the cross-validation strategy has a training set (True, False).

Has Validation Set: Indicates if the cross-validation strategy has a validation set (True, False). This is dependent on the number of dates in the data set.

Strategy Name: The name of the cross-validation strategy.

Total Splits: The total number of splits in the cross-validation strategy.

If you are satisfied with the cross-validation and avoidance period settings, continue in the Model Build phase Pipeline section by running the pipeline.

Model Build Phase Pipeline Section

The Pipeline section of the Model Build phase is where you run the SensibleAI Forecast pipeline. This brings all prior configurations and parameter selections together. The pipeline:

- Generates, transforms, and selects features based on predictive capability.
- Selects optimal hyperparameter sets for each model of each target.
- Trains models and tests them on historical data.

Running a pipeline produces the best model for each given target. After running a pipeline, you can view various statistics and metrics before deploying the models for utilization.

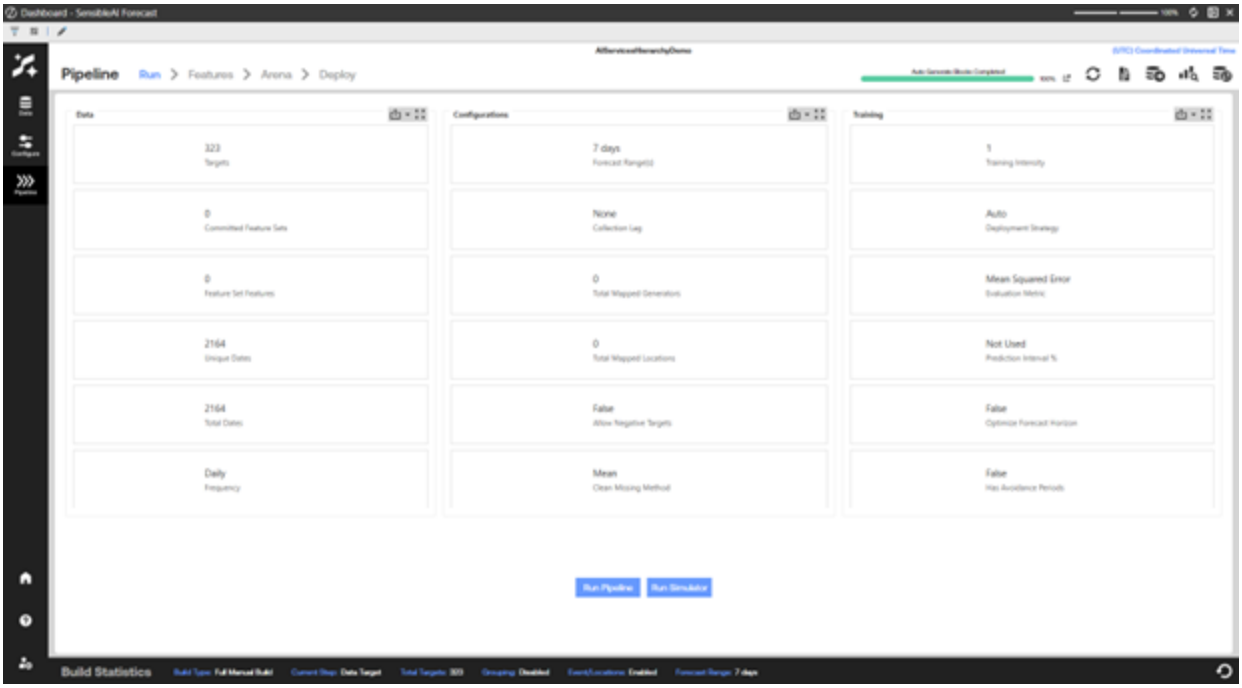
In the Pipeline section, the XperiFlow engine uses all the information and configurations gathered from the pages in the Data and Configuration sections to run feature engineering for each target.

While creating numerous new features, the engine also iteratively selects the best features to keep for each target. These are later used for all the models run for each specific target.

Run the Pipeline

When running a pipeline, all specified data configurations are brought together to generate and transform the data. It then selects the most predictive features and iteratively trains and tests models against historical data. This is the longest run of the solution because it is where the most data science work completes.

When you first navigate to the **Run** page in the Pipeline section, a **Run Pipeline** button displays in the center of the page with a variety of statistics and settings showing some of the project's currently configured settings. This indicates your data is ready for modeling.



Click **Run Pipeline** to run the pipeline job. You can also automate pipeline steps by completing the **Run Pipeline Simulator**. Clicking the button will open a component workflow where you can designate which steps to simulate.

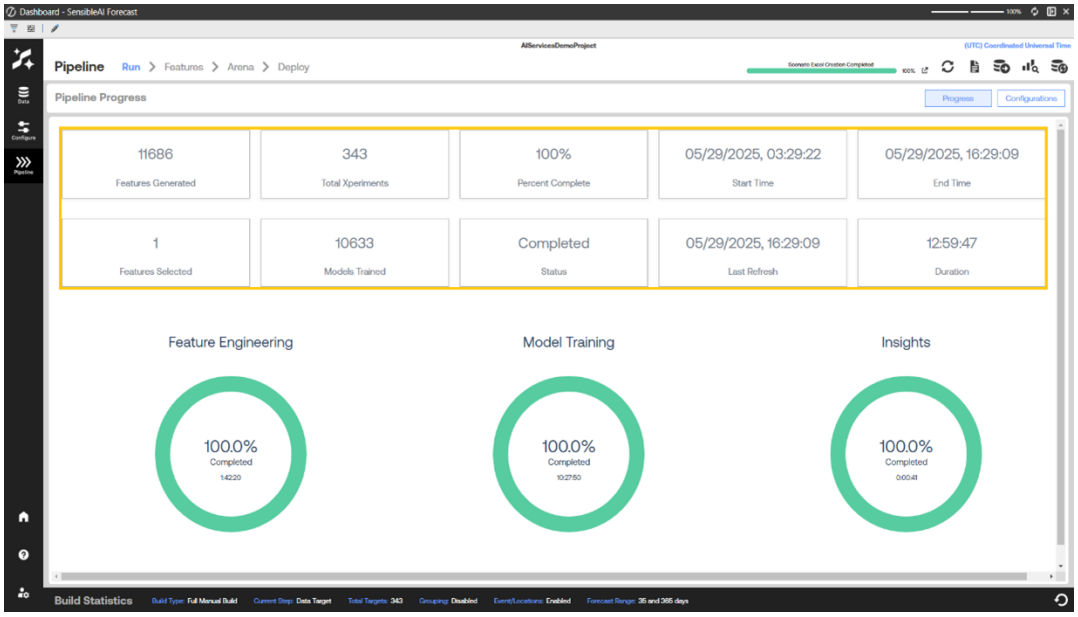
> **NOTE:** The pipeline job must run to successful completion before you can access other pages in the Pipeline section.

During the pipeline run, the XperiFlow engine does the following:

Feature Generation, Transformation, and Selection: The engine takes in all the data and information that is added to the project in the Data and Configure sections and begins the process of running feature engineering for each target. While creating numerous new features, the engine also selects the best features to keep for each target. These are used later for all the models that run for a target to increase the predictive accuracy.

Hyperparameter Tuning, Model Training, and Model Selection: With all the configurations and the newly found important features, the engine runs multiple models for each configuration to find the best ones. This process involves hyperparameter tuning each model on multiple splits of the data and then saving the accuracy metrics of each model.

When the pipeline run completes, click **Refresh** to view a summary page that displays pipeline job results. Run statistics for the most recently completed pipeline job display, as shown in the following graphic:



After the pipeline run job completes, the top half **Run** page shows:

Features Generated: Total number of features generated by the pipeline job.

Total Experiments: Total number of groups plus single targets being run in the model build. The AI Services engine is running an experiment for each of these targets or groups to find the best model possible.

Percent Complete: The current completion percentage of the pipeline job.

Features Selected: Number of times models were iterated with different hyperparameter settings during the pipeline job.

Progress: The current completion percentage of the pipeline job.

Models Iterated: Number of times models were iterated with different hyperparameter settings during the pipeline job.

Models Trained: Number of unique models that were trained.

Status: The completion status of the most recently started pipeline job.

Start Time, **End Time**: Start and end time of the most recently completed pipeline job.

Last Refresh, **Queued Time**: Date and time the pipeline run page was last refreshed.

The bottom half of the **Run** page shows:

Percent Complete and Duration of Pipeline Stages: There are 3 progress circles which indicate how far along in the pipeline a job is. These are for Feature Engineering, Model Training, and Insights.

Analyze Features

The **Features** page in the Pipeline section is an exploratory page that visualizes the feature generation, transformation, and selection that occur during the Pipeline run.

You can get valuable insights by analyzing the types of features selected for given targets. This helps to better understand what influences predictive accuracy.

This page includes an Generalization, Transformers, and a Targets view. Click the buttons at the top of the page to switch between the views.

View Predictive Features

The Transformers view provides insights into the predictive features generated and selected during Pipeline across all targets. Metrics for how long feature engineering ran during the pipeline are also provided.

Feature Engineering Duration: The number of seconds feature engineering ran.

Feature Transformation Rounds: The number of rounds that occurred during feature engineering.

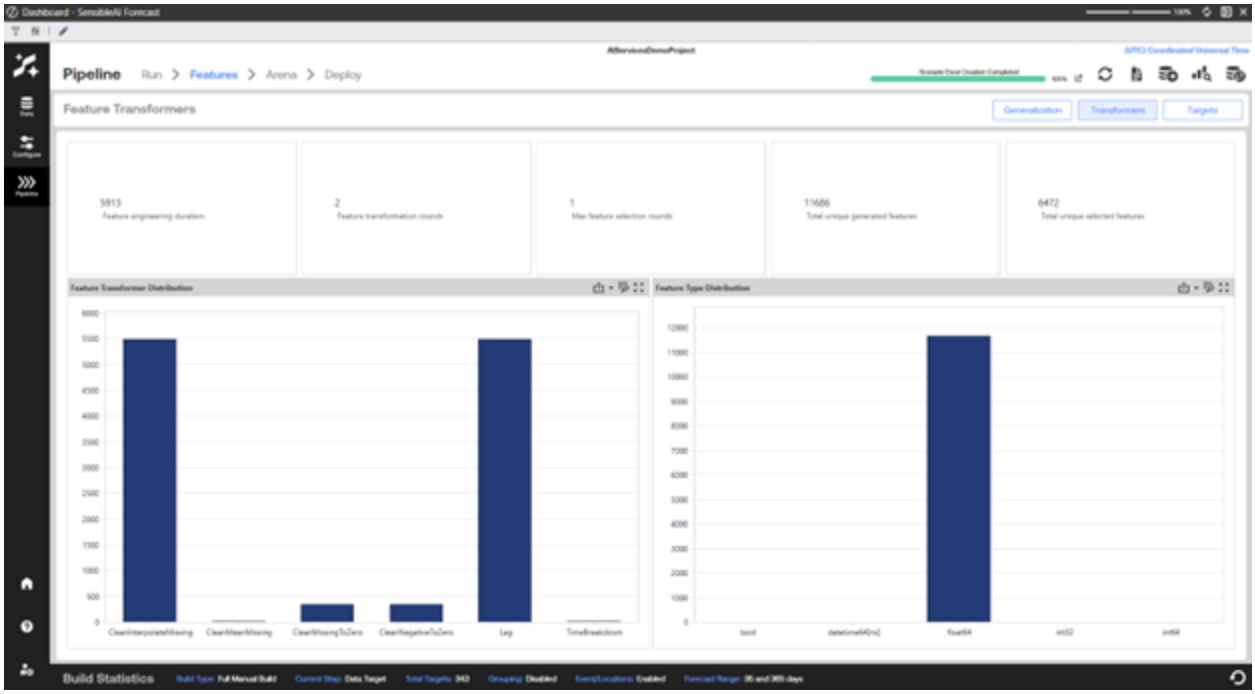
Max Feature Selection Rounds: The maximum number of rounds of feature selection run across all targets.

Total Unique Generated Features: The number of features generated or transformed by the XperiFlow engine.

Total Unique Selected Features: The number of features used by at least one model.

Feature Transformer Distribution Chart: A bar chart showing how many features were created using each type of transformer.

Feature Type Distribution: A bar chart showing the number of features for each data type.



The naming convention for features signals the order in which transformations to them occurred during Pipeline and can include the following terms:

CleanNegativeToZero: If you chose to not allow negative targets, all negative target values are cleaned by assigning them as zeros.

CleanMissing: This is a standard job by XperiFlow to impute for missing values.

From Source: Shows how many features and events are from the data source.

Lag(frequency=[X],lag_step=[Y]|): A feature like this is a lag of Y periods with X frequency.

TimeBreakdown-[X]: This feature analyzes the minute, hour, day, week, or month that a data point occurred.

View Generalization Pipeline Features

The Generalization view provides insights into the how generalized different features were across targets. The data shown can be broken down by target dimension, significance, and which hierarchies to include. The grid shows the following information:

Feature Name: The shorter common feature name for features. For example, “SalesLunch-Lag7” and “SalesDinner-Lag7” would both be “Lag7” (assuming SalesLunch and SalesDinner are targets).

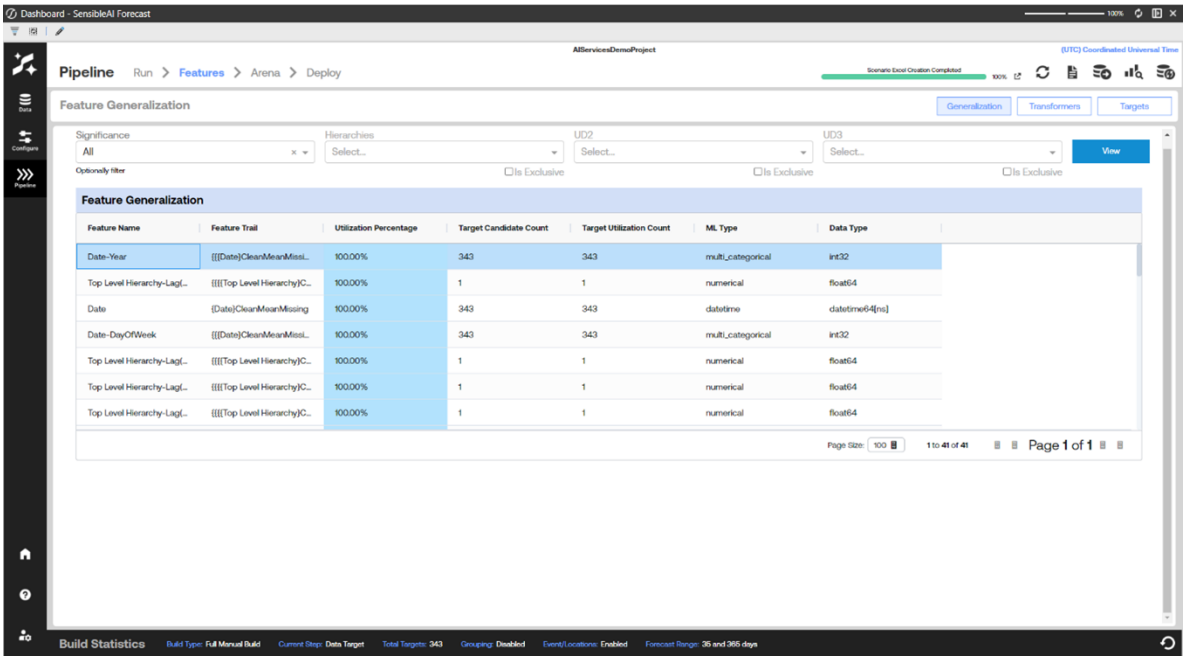
Feature Trail: The full common feature name for features. Similar to Feature Name, but the full name instead of the short name.

Utilization Percentage: The percentage of eligible targets that used the feature in at least one model.

Target Candidate Count: The number of targets that were eligible to use the feature.

Target Utilization Count: The number of targets that used the feature in at least one model.

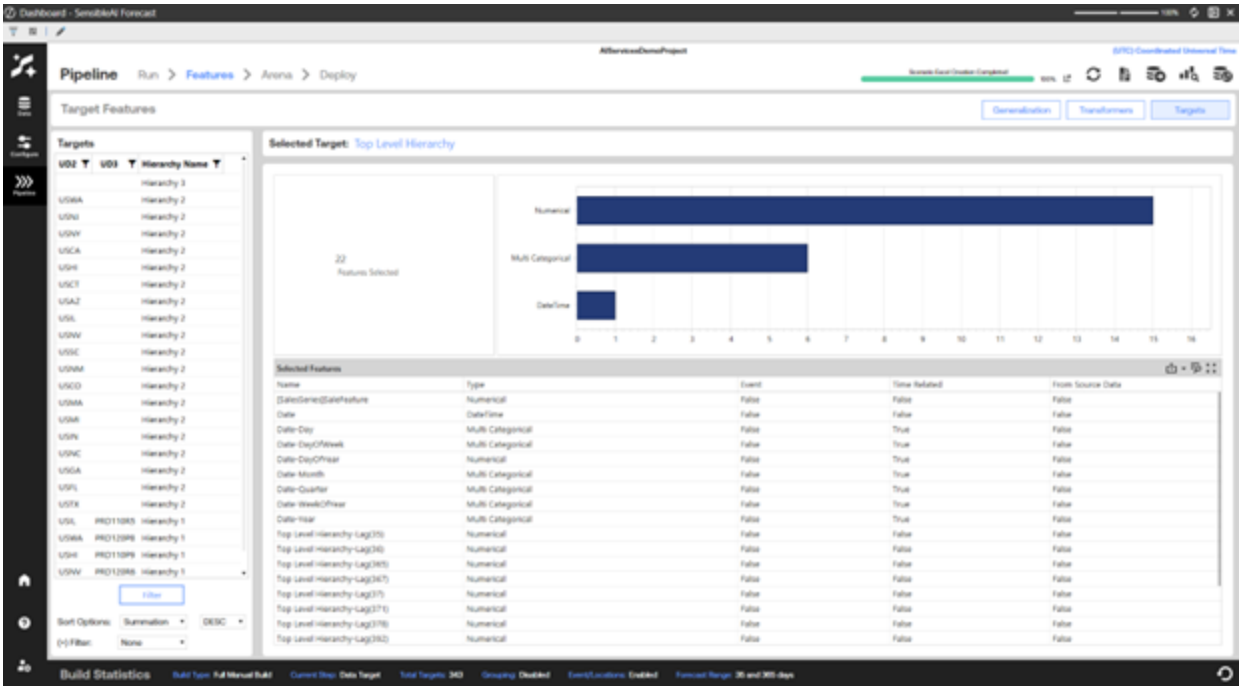
Data Type: The data type of the given feature.



View Predictive Features for Targets

The Targets view provides insights into the different predictive features that were selected for each given target and used by models during Pipeline.

Click a target in the Targets pane to see how many features were selected for it and used by models during the pipeline run and how many of those selected features are in each feature category. The table lists each selected feature for the current target, with its type and name, and whether the feature is an event feature, time-related feature, or if it originated from source data.



Selected features are broken down by types and represented visually on this page. Types include:

Numerical: Examples of this include temperature (67 degrees) or a 7-day lagged value (a feature for avocado sales today is the avocado sales from 7 days ago).

Multi Categorical: An example of this would be day of the week (1-7).

DateTime: The date dimension is always included in this.

Binary Categorical For example: “For the given date (row of data), did the event St. Patrick’s Day occur (0 or 1)?”

Analyze the Arena Summary

The **Arena** page is an exploratory page that does not require any specific action. It provides valuable insight by analyzing evaluation metrics and features across models and targets gathered during the model arena.

The **Arena** page consists of different views (Forecast, Feature Impact, Beeswarm, and Tug of War). To select a view, click on its button at the top of the page.

For all views available for the **Arena** page, use the left-most panel to filter the models available in the **Leaderboard** pane. Then select a specific Model to view its performance.

Arena Accuracy View

The Accuracy view allows the user to view source actuals overlaid with the selected model's predictions. Error metrics are displayed in the correlated subplot below. Select a datapoint on the plot to view explanations for a given prediction.

This helps to answer questions such as:

- Which type of model has the best accuracy for a given target?
- By how much did the model win?

It also helps you understand how closely the forecasted values overlay the actuals in the line chart, which can provide answers to questions such as:

- Are there spikes that aren't being caught by the forecasts?
- If so, could adding any events help catch these spikes?

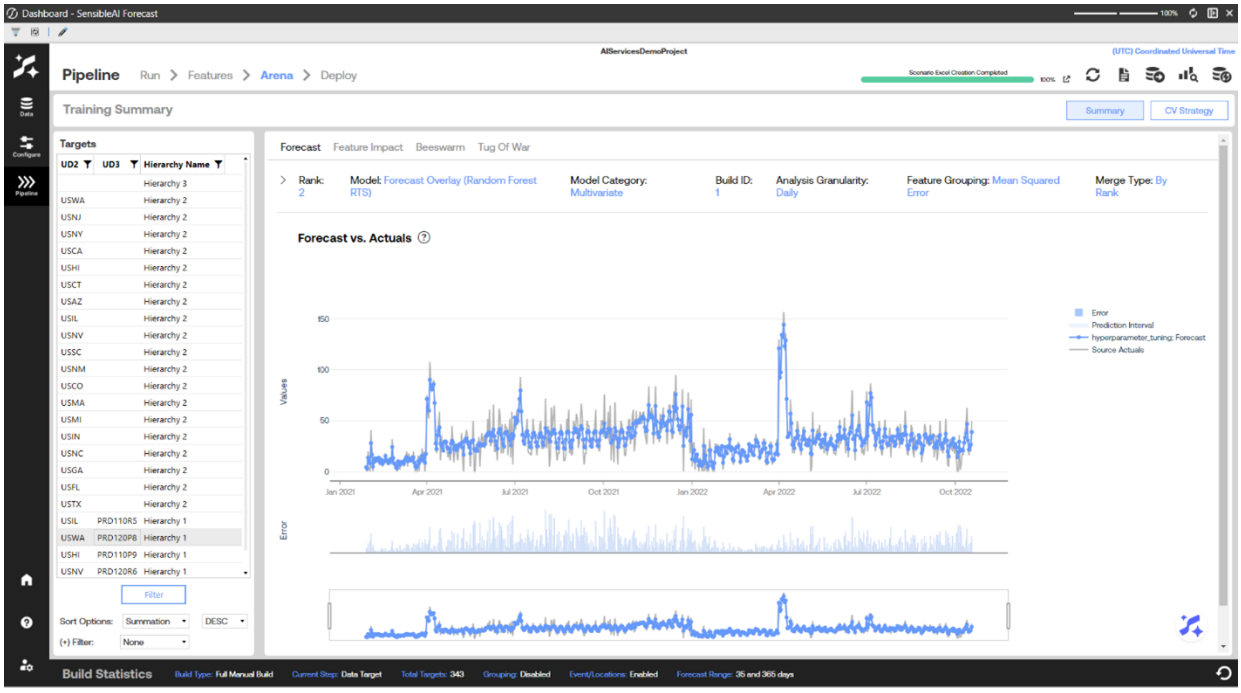
> **NOTE:** Error metric scores do not dynamically adjust based on the time period specified by the range slider at the bottom of the page.

The table on this page displays:

- The model algorithms run for a given target (such as XGBoost, CatBoost, or Shift).
- The type of model algorithm (ML, Statistical, or Baseline).
- The evaluation metric (such as Mean Squared Error, Mean Absolute Error, and Mean Absolute Percentage Error) and the associated score.
- The train time (How long did it take to train the given model during Pipeline?).

With all the configurations and the newly found important features, the engine runs multiple models per configuration to find the best. This process involves hyperparameter tuning each model on multiple splits of the data and then saving the accuracy metrics of each model.

To analyze the training results, click a target in the Targets pane to see the accuracy metrics of each it its deployed models if implemented over the course of its past data. Each model listed shows its name and category, along with the type of evaluation metric used and the evaluation metric score. Select a model name in the models list to view a line chart that shows how close the forecasted values are to the actuals.



The line chart corresponds to the highlighted model in the table. It visualizes both the predictions made for the historical actual test period and the historical actuals. The time frame in this chart is only a subset of the total time frame for the historical data, as this time frame is for a specific portion of a split.

At a high level, this page provides the view of how the best version of each model (such as XGBoost, CatBoost, ExponentialSmoothing, Shift, and Mean) has performed against unseen historical data for each target and more specifically, on the test set of the historical data.

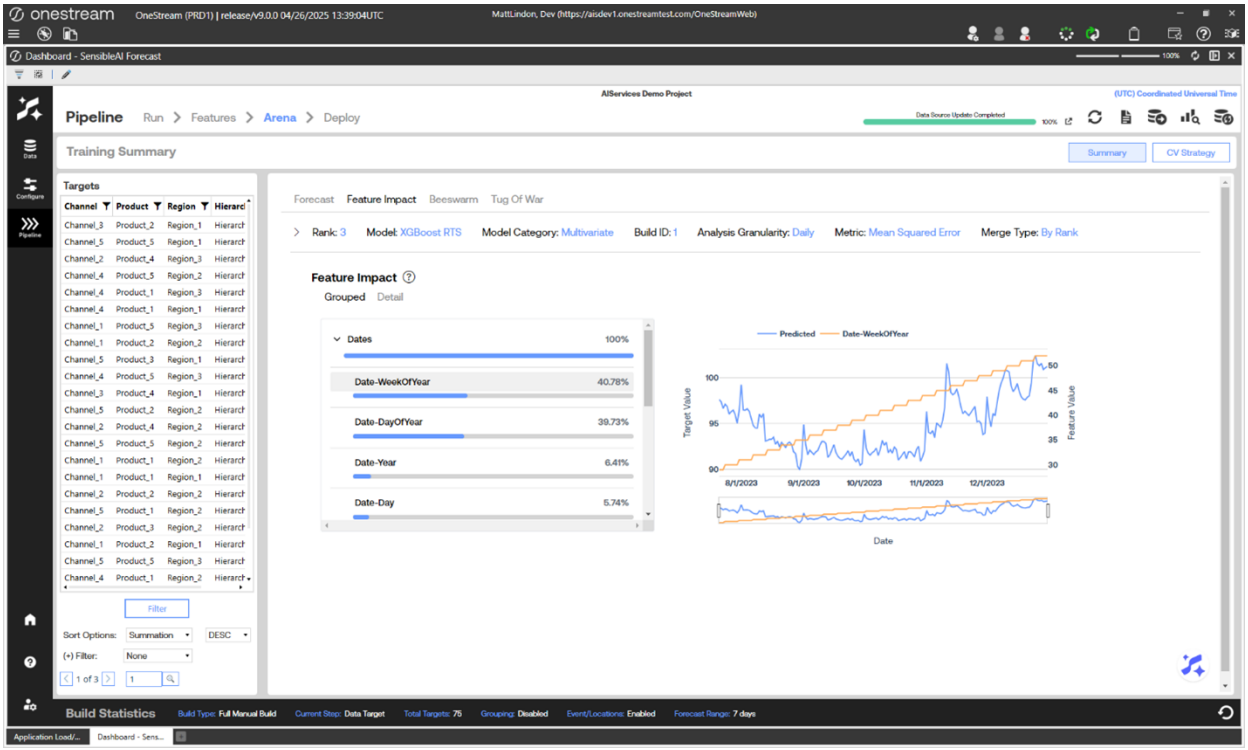
The optimal error metric score can be the lowest score, the highest score, or the closest to zero. It is dependent on the type of metric. See Appendix 3: Error Metrics for more information about the error metrics SensibleAI Forecast uses.

The line chart corresponds to the highlighted model in the table. It visualizes both the predictions made for the historical actual test period and the historical actuals. The time frame in this chart is only a subset of the total time frame for the historical data, as this time frame is for a specific portion of a split.

Arena Feature Impact View

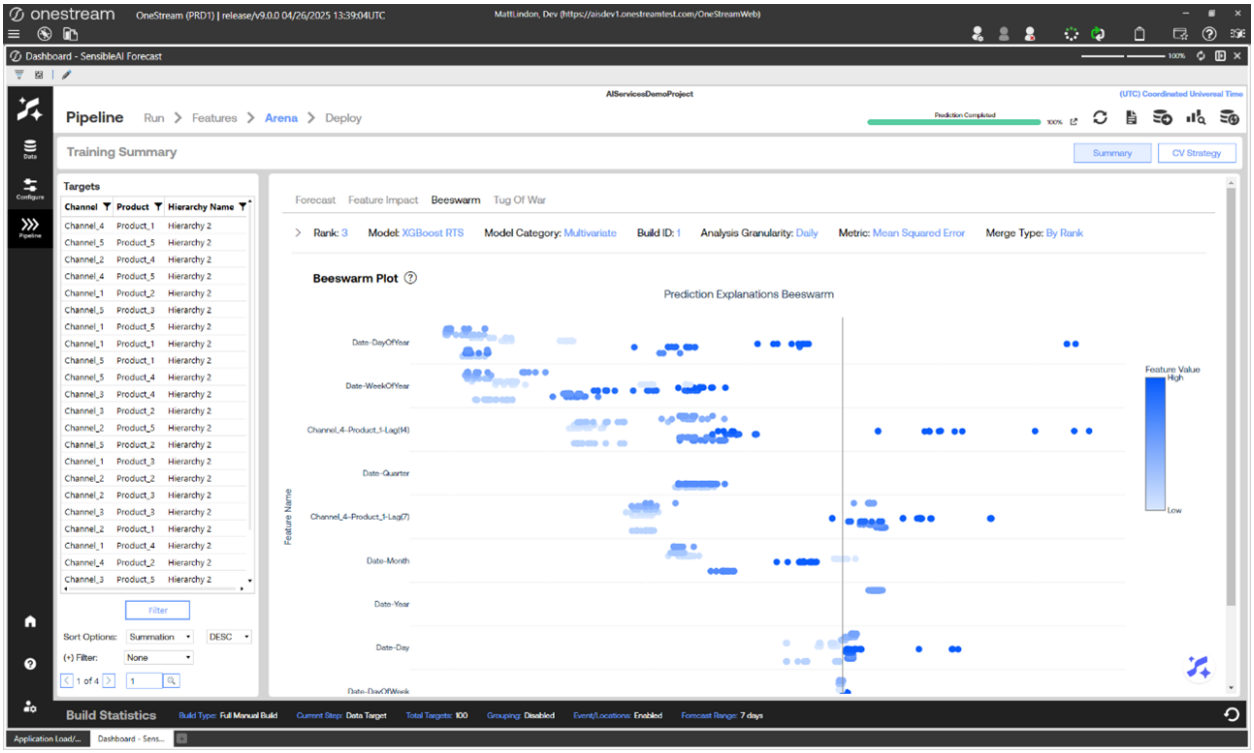
The Feature Impact view shows the hierarchical (Grouped) or individual (Detail) feature impact percentages aggregated across all the model's prediction values. The user can click on a feature name from the table to view its actual values compared with the predictions and actuals in the visual on the right. The prediction and actual values are bound by the primary y-axis on the left hand side of the graph, while the feature values are bound by the secondary y-axis on the right. The feature impact score shows how much influence the feature had for a given model.

> **NOTE:** Feature impact data is dependent on the type of model. Not all models have feature impact data.



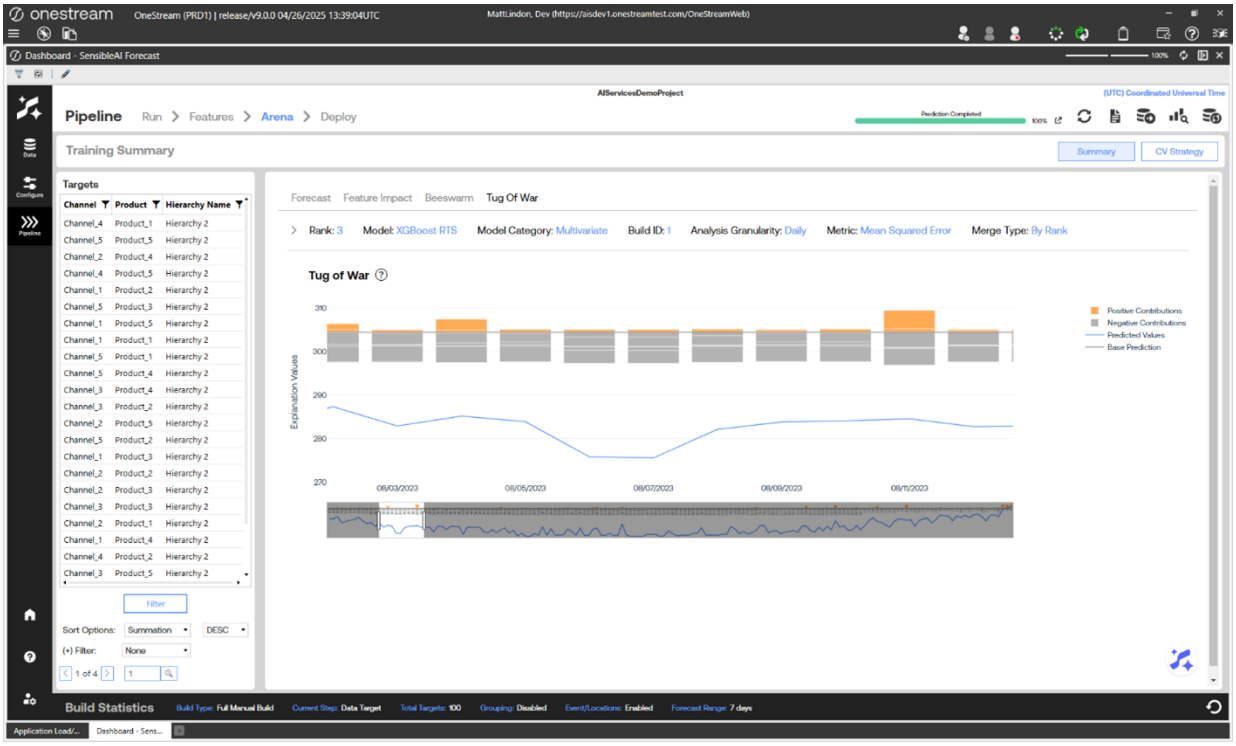
Arena Beeswarm View

The Beeswarm view visualizes the impact of features on model predictions by displaying SHAP values as individual dots, where each dot represents a feature's SHAP value for a specific instance. Features are ordered by importance, with the most influential ones at the top. The color of each dot indicates the actual feature value. The horizontal spread of dots reflects the distribution of SHAP values for each feature, showing how much they contribute to model predictions. Plot is based on source granularity and changing the Analysis Granularity will not change the plot.



Arena Tug of War View

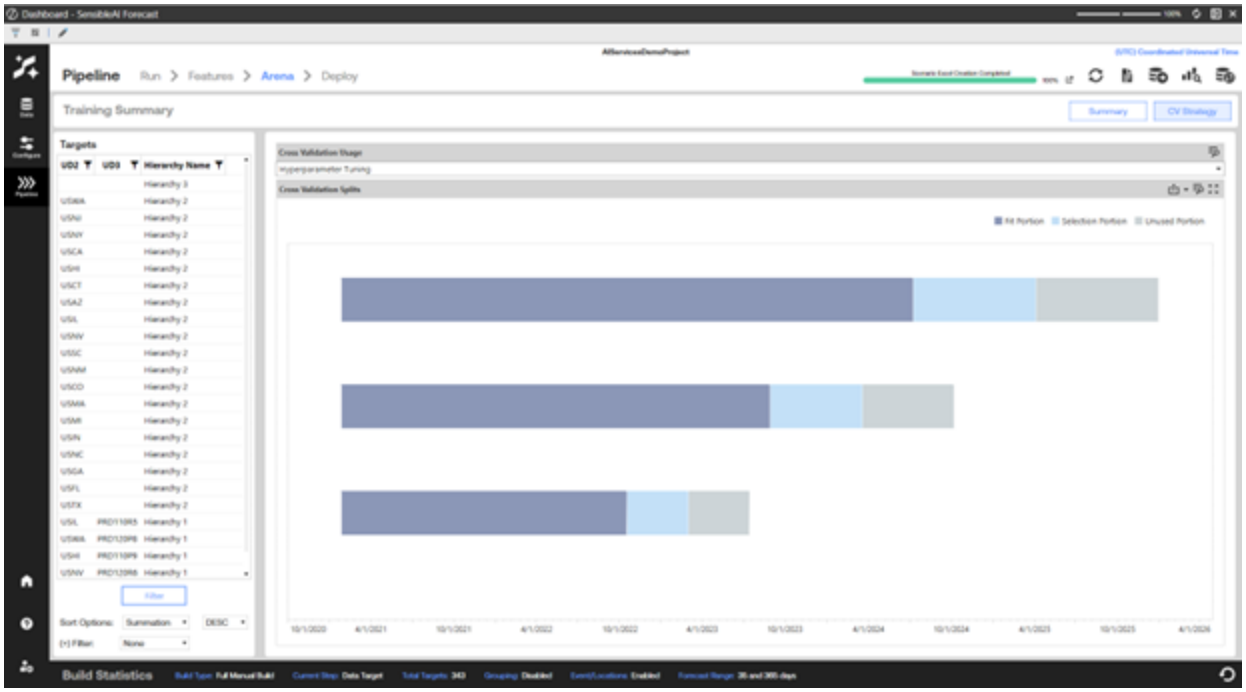
> **NOTE:** Only models that use features have feature explanation data.



The Tug of War view visualizes the four most significant positive and negative prediction explanations for all dates at the selected Analysis Granularity of the selected model. Compare with base predictions and actual prediction values for all dates. Users can toggle any component of the visual on or off by selecting it from the legend on the right. Users can also narrow in on specific dates by interacting with the range slider.

Arena CV Strategy View

The CV Strategy view shows how the splits were used in each model stage and the portion of the splits. Select the Model stage using the Cross Validation Usage drop-down. Split usage can then display in the Cross Validation Splits Chart.



The number of splits and size of each portion of the splits can be configured when you Set Modeling Options. A description of how each portion of the split is used follows.

Train Set: The split of historical data on which the models initially train and learn patterns, seasonality, and trends.

Validation Set: The split of historical data on which the optimal hyperparameter set is selected for each model, if applicable. A model makes predictions on the validation set time period for each hyperparameter iteration. The hyperparameter set with the best error metric when comparing predictions to the actuals in the validation set time period is selected. This split does not occur when the historical data set does not have enough data points.

Test Set: The split of historical data used to select the best model algorithm compared to the others. For example, an XGBoost model gets ranked higher than a baseline model based on evaluation metric score. This split does not occur when the historical data set does not have enough data points.

Holdout Set: The split of historical data used to simulate live performance for the model algorithms. This is the truest test of model accuracy. This set can also serve as a check for overfit models. This split does not occur when the historical data set does not have enough data points.

Deploy Your Model

The **Deploy** page provides information that lets you fully analyze and understand the effectiveness of your model before deploying it to production. Once satisfied, you deploy your model using this page, which collects necessary information from the pipeline job to be able to run the deployed models in utilization. This information includes:

- The most optimal hyperparameters for deployed models.
- How to generate and transform features selected for the deployed models.

Analyze Pipeline Performance Overview Statistics

General statistics shown here include:

Features Generated: Number of features generated for the entire data set.

Features Selected: Number of features selected for the data set based on being able to positively contribute to predictive accuracy.

Models Iterated: Number of times models were iterated with different hyperparameter settings during the pipeline job.

Train Time: Total train time across all targets and target groups during Pipeline. This total time is not sequential however, as much of the Pipeline is run in parallel through the XperiFlow Conduit Orchestration.

The charts on this page include:

“Best” Models: Descending bar chart that visualizes the breakdown of best models selected across all targets, so you can understand how frequently different models and model types are winning.

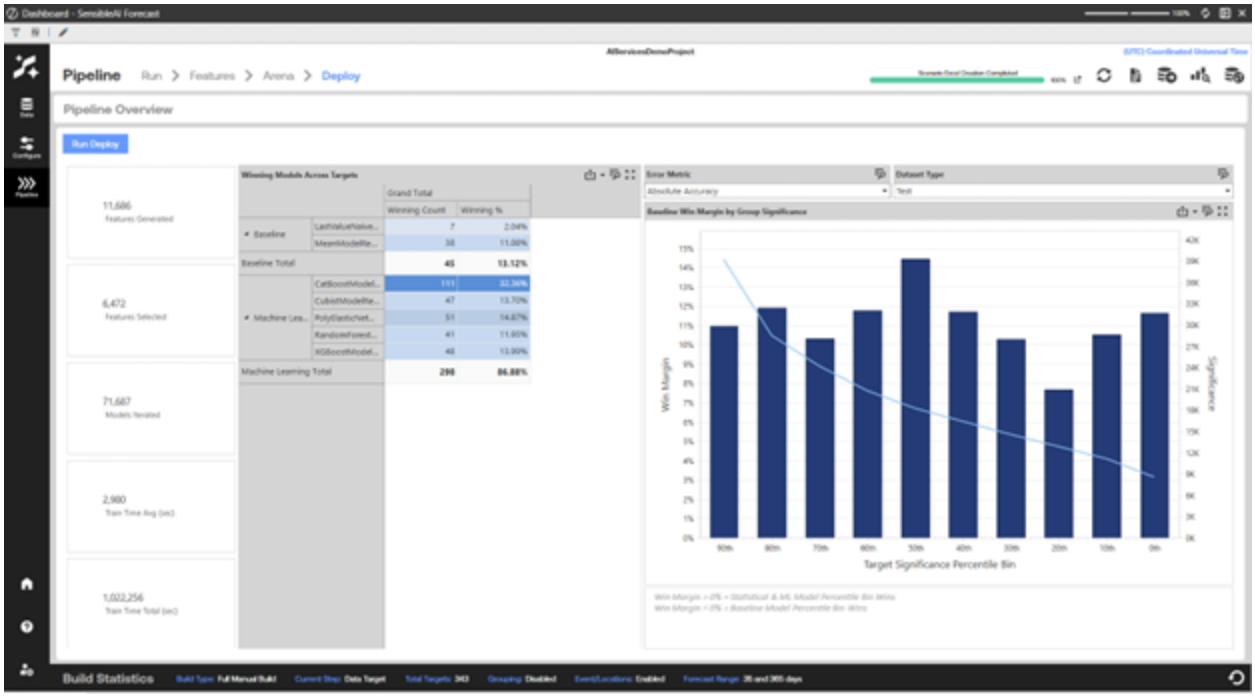
Baseline Win Margin by Group Significance: Each bar in this chart represents an even-sized bin of targets. Bar heights indicate the percent by which the best statistical or machine learning model beat or lost to the best baseline model based on the selected error metric, summed across all targets within the bin.

The light blue line in the chart represents the bin's total significance. Bin refers to the value amount selected during the Data section, such as total units or dollars. Positive win margins are ideal. This means that machine learning and statistical models are beating the simplistic models on average.

It is possible however, for a bar to be negative due to an instance where the best baseline model beats the best machine learning or statistical model. For example, in a ten-target bin, nine machine learning and statistical models can beat the best baseline by 10 percent each, but one baseline model that wins by 120 percent can swing the bar to be negative.

Use the Overview view to get valuable insight by analyzing the Best Models and Baseline Win Margin by Group Significance charts. This can help answer questions such as:

- How often are my machine learning and statistical models beating the best baseline model?
- By how much are my best machine learning and statistical models beating the best baseline model?
- Are the best baseline models being beaten for my most significant targets? This is specific to non-units-based value dimensions, such as sales dollars.



Deploy Your Model

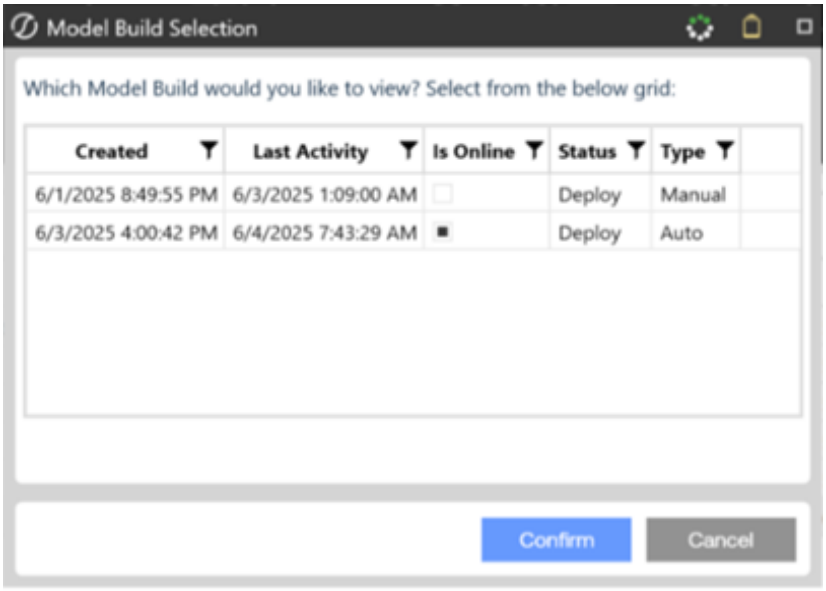
After reviewing the available statistics and visualizations for each target, click the **Run Deploy** button. This creates the deployment job, which upon completion changes the project's status and moves the project from the Model Build phase to the Utilization phase.

The deployment job takes the best models selected during pipeline and deploys them for generating forecasts.

Additionally, after the models have been chosen for deployment, SensibleAI Forecast creates the feature schemas and pre-trained models needed to run predictions against the models in the Utilization phase.

Viewing Historical Model Builds

In the event of a Restart or Rebuild being run on a Model Build, a project may contain several historical Model Builds. If this is the case, when entering the Model Build Phase of FOR, a dialog to select from the historical Model Builds will be provided.



Model Build Information

In the provided dialog, the following information displays for each model build available:

Created: The date and time the Model Build was created.

Last Activity: The date and time of the last activity.

Is Online: Indicates if the Model Build is still active.

> **NOTE:** A build will be offline after a Restart or a new build is deployed to Utilization.

Status: Indicates if the Model Build is being configured (Train) or is used in Utilization (Deploy).

Type: Indicates if the Model Build is configured and deployed by the user (Manual) or an Auto Rebuild was run on the Model Build (Auto).

Utilization Phase

The Utilization phase consists of these sections:

Manage: Run predictions, monitor model health, audit model builds, and update event occurrences.

Analysis: Analyze results from model forecasts, view statistics on how well the deployed models are performing.

Monitor: Flag targets and create filters for when certain metric thresholds are met.

Utilization Phase Manage Section

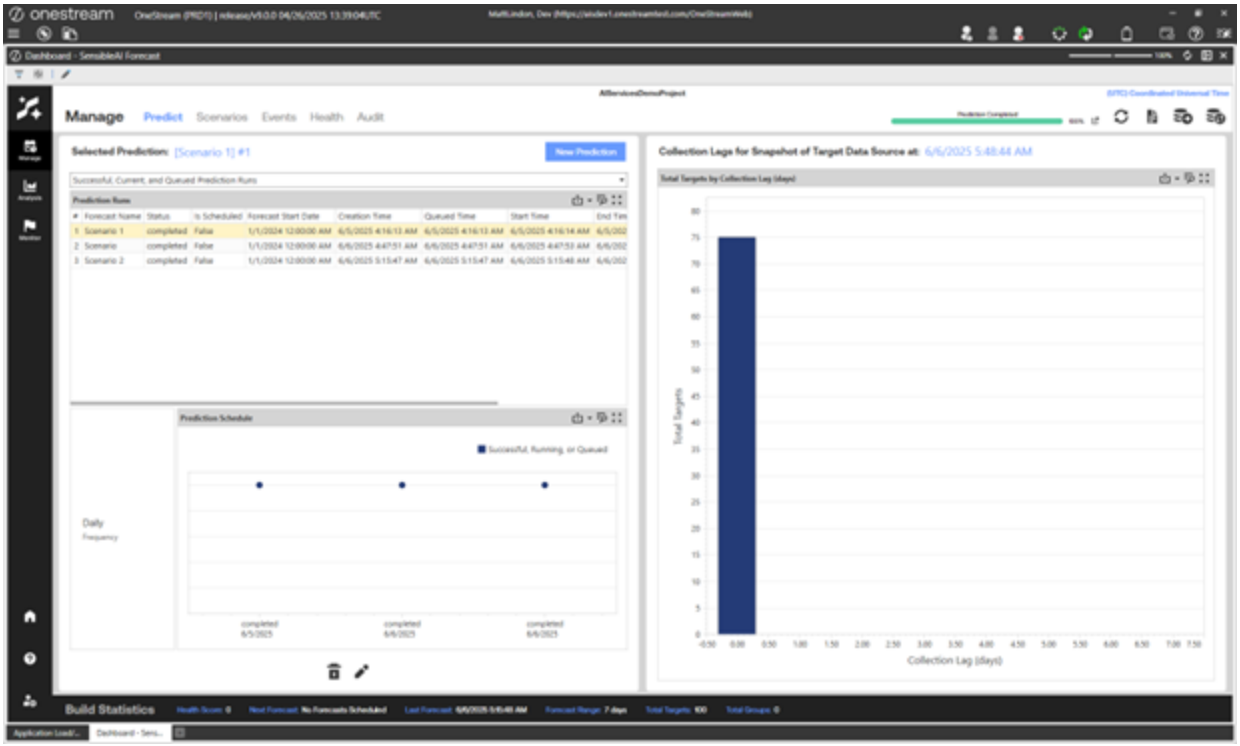
The Manage section consists of these pages:

- **Predict**: Run, schedule, and delete scheduled predictions.
- **Scenarios**: Review and modify event occurrences that are defined in the Events page of the Model Build phase.
- **Events**: Review and modify event occurrences that are defined in the Events page of the Model Build phase.
- **Health**: Monitor model health by project and target and run full or partial rebuilds to restore model health.
- **Audit**: View detailed audit information for all model builds for the project.

Run or Schedule a Prediction

From the **Predict** page, you can:

- Run and schedule predictions.
- Delete scheduled prediction runs.
- View prediction status.
- View the collection lag for the target data set.
- Create a new snapshot.

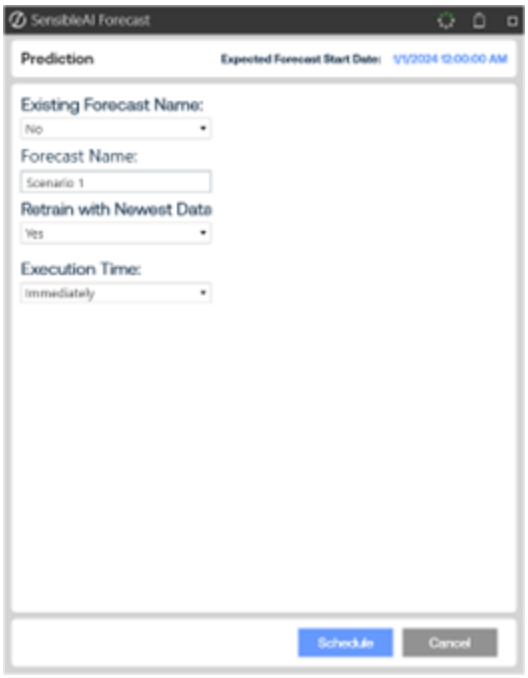


When you run a new prediction, the XperiFlow engine uses the forecast range established on the **Forecast** page of the Configure section to generate a forecast for each target using the deployed models. This prediction is made from the date of the last data point in the loaded data and extends to the forecast range. The XperiFlow engine also recognizes and collects any actual data points following the date of deployment, bringing it in as actual values on the **Prediction** page of the Analysis section.

> **IMPORTANT:** Before running a new prediction, you must load and transform any additional data sets being used for your model. Perform the steps in Upload and Associate Data Sets for the next data set. Verify rows in the transformation table are approximately the same as in the actual data file.

To run a new prediction:

1. Click **New Prediction**. The **New Prediction** dialog box displays.



2. Make selections from these options:

    - **Retrain with Newest Data**: Select **Yes** to retrain all models on any new data that has been uploaded to the project before making a prediction. For example, if the model learned from three years of historical data during the Pipeline section, and three months of new data has been uploaded during the Utilization phase, the model can now learn from three years and three months of data.
    - **Existing Forecast Name**: If a forecast name should be included and used to label the prediction run.

- **Forecast Name**: Enter a name to label the prediction. This name displays in different consumption group export tables. This name must be unique for a given forecast start date. For example, you can't have 2 editions of "Scenario 1" for the forecast start date of 1/1/2020, but you can have "Scenario 1" and "Scenario 2" for the forecast start date of 1/1/2020.

  > **TIP:** Forecast names are a great way to link forecasts across multiple prediction runs. Prediction runs with the same forecast name are linked and can be visualized on the **Prediction** page.

- **Execution Time**: Select **Immediately**to move the prediction job to the Job queue. Otherwise, select **Schedule** and use the date/time fields that display to set the hour, day, month, and year when you want the prediction job to run.

  > **NOTE:** Scheduled time is based on local time specified in the Global Settings.

3. Click **Schedule**.

Analyze Predictions

After running an initial prediction, the **Predict** page displays the following information.

Total Targets by Collection Lag: This chart in the Collection Lags pane visualizes the latest view of the collection lag for the target data set. This chart updates when you run a new prediction, snapshot, target data source update, or data set job.

Prediction Runs: This table in the Selected Predictions pane displays information about predictions that have been run or are scheduled to run, including status, queued time, creation time, start time, end time, and job ID. To delete a queued or scheduled prediction, select it and click .

> **NOTE:** At least three prediction jobs must run to completion for SensibleAI Forecast to have enough data to produce a health score for the model. See Manage Model Health for more information.

Prediction Schedule: This chart in the Selected Predictions pane shows completed and scheduled predictions.

Update a Scheduled Prediction

1. Click **Update**  below the Prediction Schedule pane.
2. In the **Update Prediction** dialog box, use the Existing Forecast Name field to indicate whether you want to update the schedule for an existing forecast. If selecting Yes, select the name of the forecast to update from the Forecast Name drop-down. If selecting No, type the name of the forecast to update.
3. Click **Save**. A message box informs you that the Prediction Call forecast name is modified.
4. Click **OK** to close the message box and the **Update Prediction** dialog box.

Manage Model Health

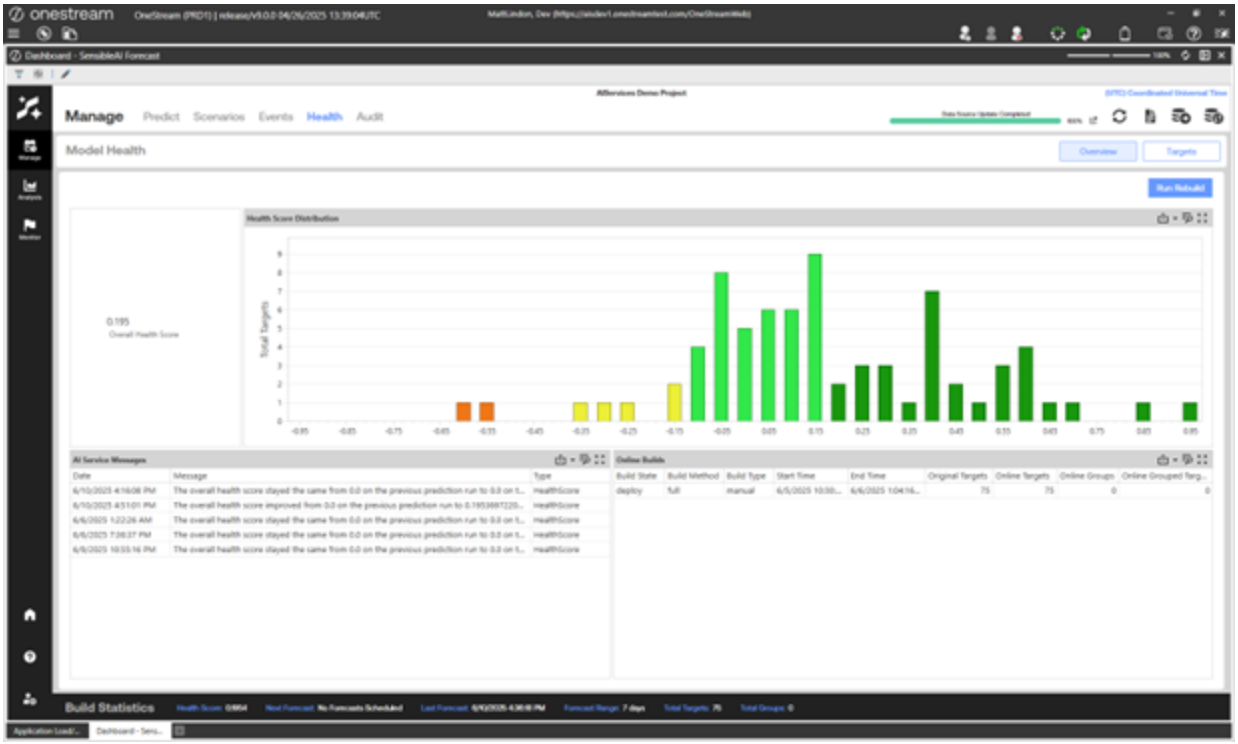
The **Health** page has two views into the health of your model: Project Overview and Targets. It's also where you can run a rebuild of the project.

Analyze Model Health by Project

The Project Overview view shows a color-coded distribution chart visualizing the health of the best models. The Overall Health Score is a scaled metric ranging between -1 and 1, which indicates improvement or degradation in a model’s predictive accuracy.

A positive (green) health score signals that a model's predictive accuracy has improved since it was deployed. A negative (red) health score signals a decrease in predictive accuracy since initial deployment.

The AI Service Messages provides information on how the overall health score has changed with each prediction. The Online Builds pane lists details of all builds for the project.



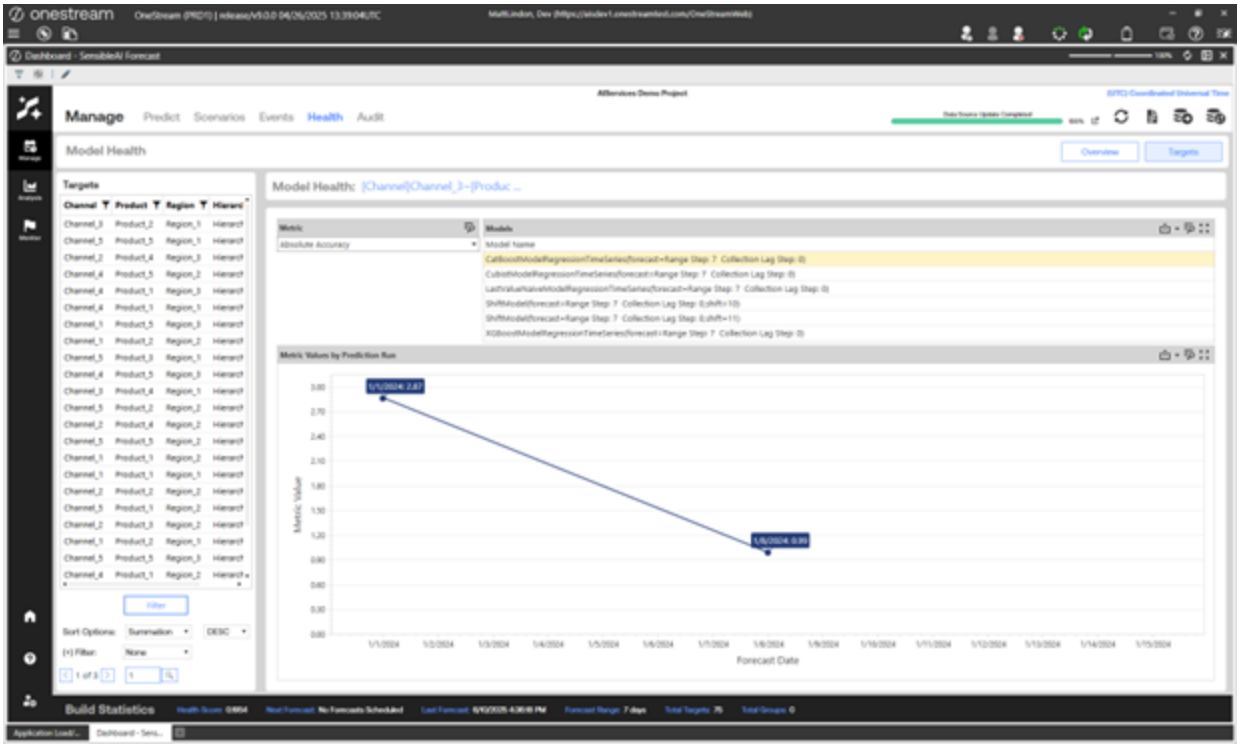
Analyze Model Health by Target

In the Targets view, you can see how the evaluation metric score for a given target/model/metric score combination changed over time.

Click a target in the **Targets** pane to see the average metric score for each of its models, based on the error metric selected in the Metric Name field. Use the Metric Name drop-down to select the type of error metric. See Appendix 3: Error Metrics for more information on the error metrics SensibleAI Forecast uses.

For each model of the selected target, you can view a line graph that shows the metric scores for each of its prediction runs. Click a model name in the top pane to view the model's prediction run information in the line chart. This includes the date of each of the model's prediction runs and the average metric score for each.

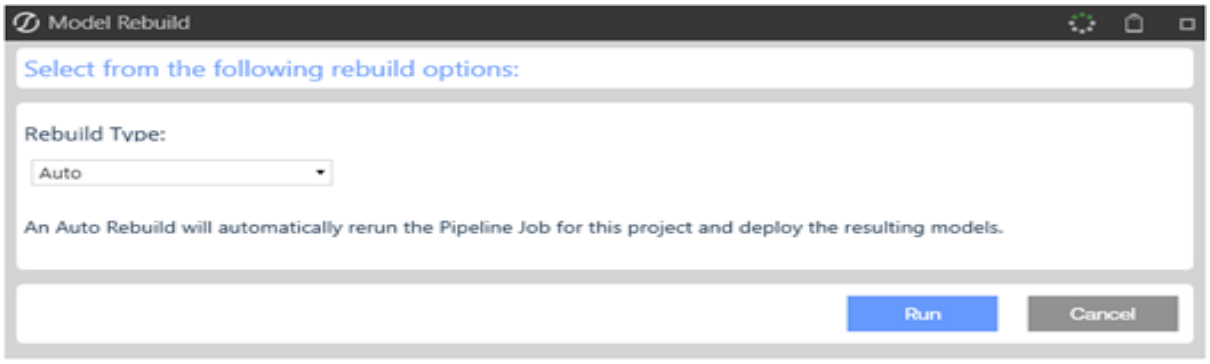
> **TIP:** A forecast is not included in the line chart if no actuals are present for the metrics to be calculated. The date displays on the x-axis, but no corresponding value displays.



Rebuild the Model

You can rebuild the model when needed. The rebuild process can be automated to run as a data management job in the background, or you can click **Run Rebuild** on the **Health** page to run it manually. There are two rebuild methods:

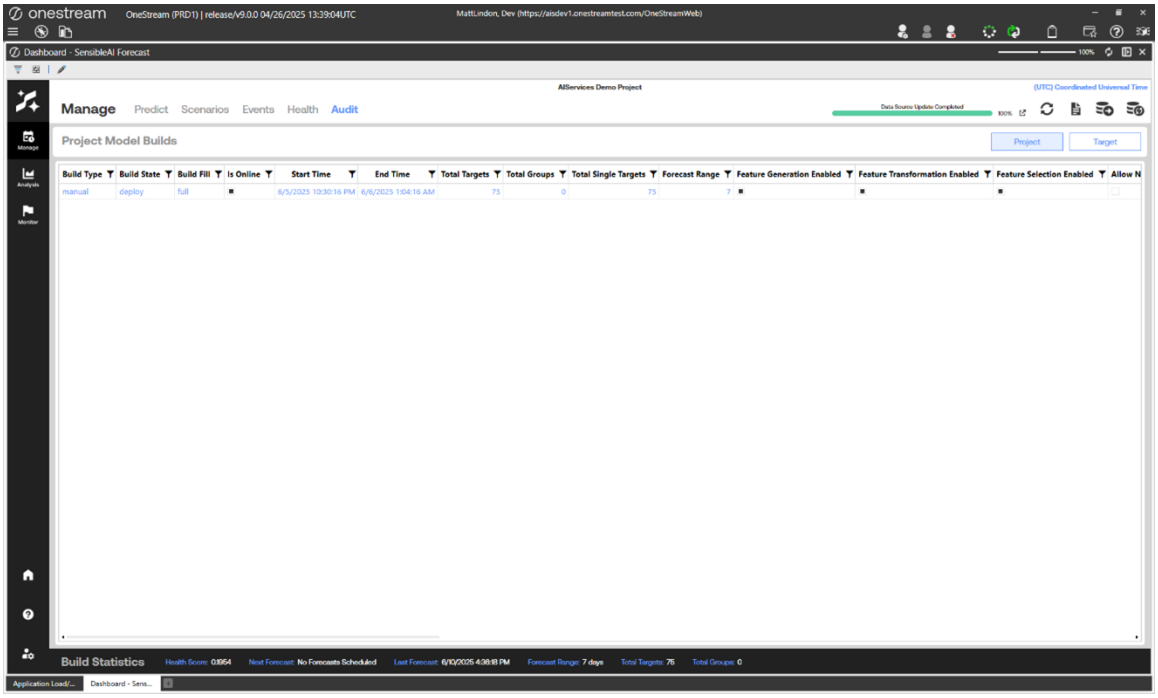
- **Auto Rebuild**: Automatically rebuilds and deploys the entire project using the same configurations from the latest build. The new build's models are trained on more historical data than in the initial build when additional data is being used since the latest build.

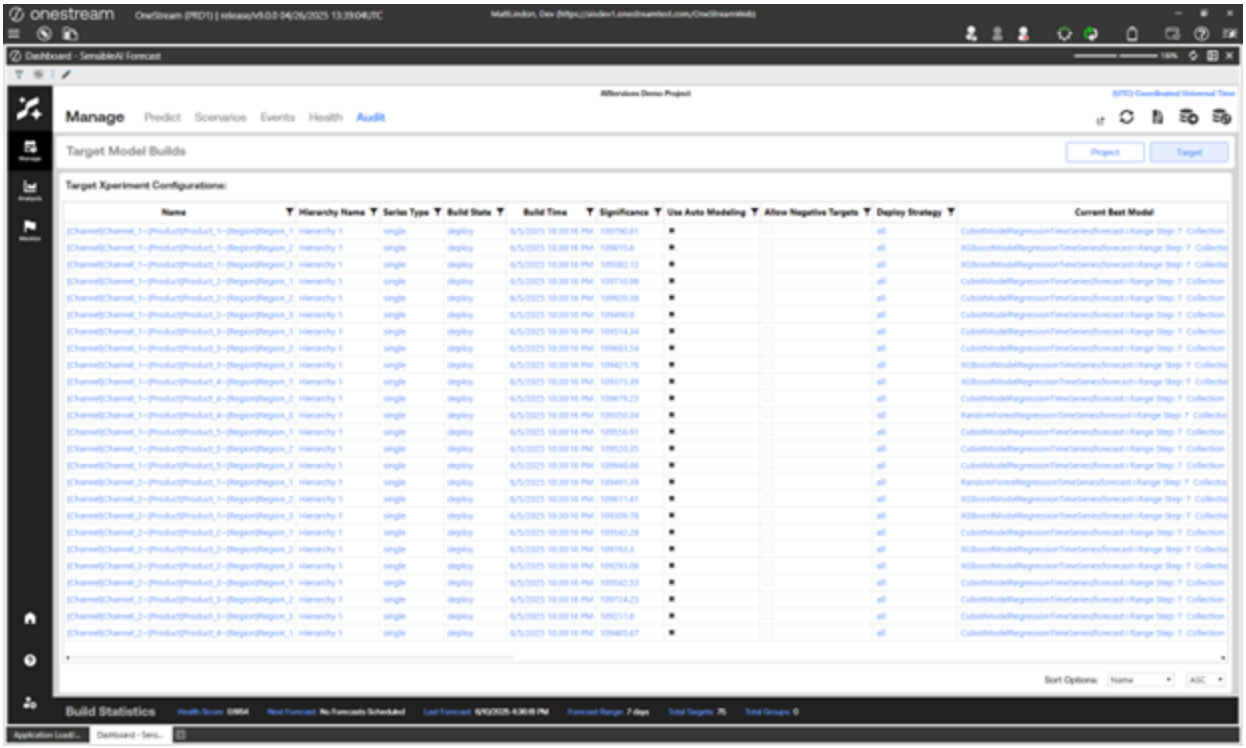


- **Manual Rebuild**: Returns you to the **Dataset** page of the Data Section in the Model Build phase. You can walk through the Model Build phase, setting different configurations as needed.

Audit Project Model Builds

The **Audit** page lets you view details of builds that have been run on the project and the target configurations. The page provides a full audit of what has been run.


onestream
Manage
Predict
Scenarios
Events
Health
Audit
Project Model Builds
Build Statistics


onestream
Manage
Predict
Scenarios
Events
Health
Audit
Target Model Builds
Target Xperiment Configurations:
Build Statistics

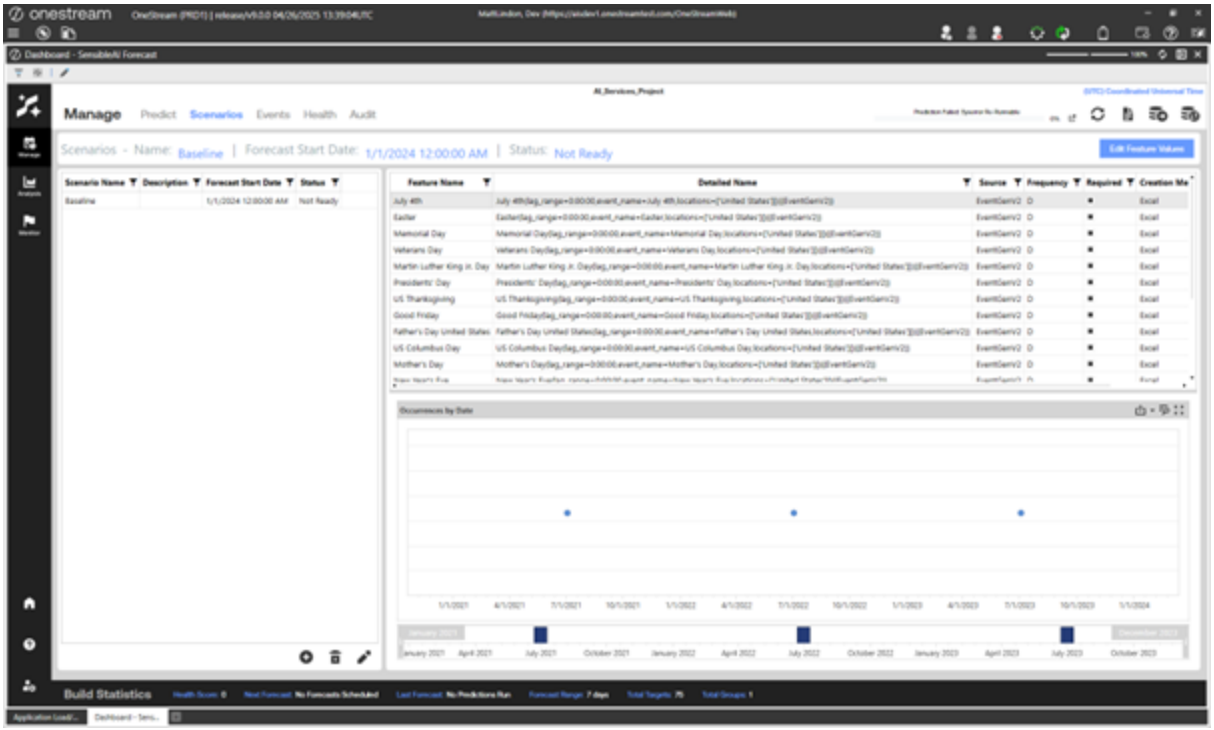
Manage Configured Events

The **Events** page in the Utilization phase lets you manage and modify events defined in your project during the Events step in the Modeling phase.

The Utilization phase **Events** page in the works the same way as the **Events** page in the Modeling phase. See Configure Events for information on how to modify the events defined during modeling.

Manage Scenarios

The **Scenarios** page in the Utilization phase is where **Scenario Modeling** configurations can be made. This feature is designed to empower users with the ability to simulate various future scenarios. This functionality enables the configuration of hypothetical sets of feature values or potential events.



Overview

- **Configure Scenarios**: Users can define and customize scenarios by specifying future feature values or predicting the occurrence of certain events.
- **Run Predictions**: After scenarios are configured, predictions can be run for each Scenario. The Xperiflow Engine will factor the user submitted values for feature values and event occurrences in the prediction.
- **Forecasting Impacts**: By leveraging this feature, users can explore how different features and events might influence their forecasts. This predictive capability enables strategic planning and preparing for a range of potential outcomes.

Key Benefits:

- **Enhanced Preparedness**: Scenario Modeling equips users with the tools to anticipate and plan for various future conditions, thereby improving decision-making processes.
- **Customizable Simulations**: The feature offers flexibility, enabling users to tailor Scenarios according to their specific forecasting needs.
- **Predictive Insights**: Through precise modeling, users gain insights into how different variables and events could potentially impact their forecasts.

Scenarios Table

The following information is displayed for each Scenario in the left pane of the page.

Scenario Name: The name of the Scenario.

Description: The description of the Scenario.

Forecast Start Date: The expected start date of the forecast when the Scenario is run.

> **IMPORTANT:** Only Scenarios for the current forecast period will be shown on the **Scenarios** page. Please ensure that all target and feature data sources are up-to-date and a **Date Source Update** job has been run for all data sources. This allows Xperiflow to generate an accurate Forecast Start Date.

Status: The current status of the Scenario showing if it is ready to be used in a prediction. The available statuses are:

- **Not Ready**: Scenario has not received Feature/Event values and a prediction cannot be run for this scenario.
- **Ready**: Scenario has received Feature/Event values and can be used to run a prediction.
- **Complete**: A prediction was run using this Scenario.

Create a Scenario

To create a Scenario:

1. Click the **Add** button ⊕ at the bottom left of the pane.
2. Type a unique name for the scenario.

   > **NOTE:** Each scenario in the project must have a unique name and cannot include special characters, excluding underscores.

3. Type a description for the scenario. This is optional.
4. Click the **Create** button .

Edit a Scenario

1. Click the **Edit** button  at the bottom left of the pane.
2. Change the scenario name.

   > **NOTE:** Each scenario in the project must have a unique name and cannot include special characters, except underscores.

3. Update or add a description for the scenario. This is optional.
4. Click the **Save** button.

   > **NOTE:** If a prediction has already been run on the Scenario being edited, the Forecast Name for the prediction will be updated on all other pages in Utilization that it appears.

Edit Feature Values and Event Occurrences

To edit the Feature Values and Event Occurrences, click the **Edit Feature Values** button at the top right of the page.

1. On the first entry of the Edit Feature Values dialog box for each Scenario, an Excel file will be generated by Xperiflow. A status bar for the Excel Creation job is shown and will be replaced with the Excel file once fully generated.

   > **NOTE:** This file will contain historical values for every Feature and Event set to Scenario Modeling Feature or Scenario Modeling Event during Model Build.

2. Once the Excel file is displayed, the Features and Events required to run a Scenario will appear in each column. The following rules should be followed when providing values:

- **All required cells must be completed before submitting to Xperiflow**. Cells that require a value from the user are highlighted yellow. The frequency of the Feature or Event will determine how much data is necessary.
- **The Save button must be clicked in the Excel sheet**. If the Excel is not saved before submission, the last saved version of the sheet will be submitted to Xperiflow.
- **Data Type Guidelines**:
    - **Events**:
        - **0** – No Occurrence on that date
        - **1** – Event Occurrence on that date
    - **Features**:
        - Reference historical data to match the proper data type of that Feature.

3. After inputting the Scenario data into the Excel sheet, press the **Submit** button. Xperiflow will validate the data in the Excel sheet and give feedback on any changes that need to be made.
4. Upon successful submission to Xperiflow, the status of the Scenario will update to **Ready**, and it is ready to be used in a prediction.

Scenario Feature Table

The following information displays for each Scenario in the pane of the page.

Feature Name: The name of the Feature/Event.

Detailed Name: The detailed version of the name used to generate the Feature/Event. This includes any Custom Parameters or Location configurations.

Source: The name of the generator used to generate the Feature/Event.

Required: Indicates whether or not the Feature/Event is required for completing a Scenario.

Creation Method: The method that provided data for the Feature/Event.

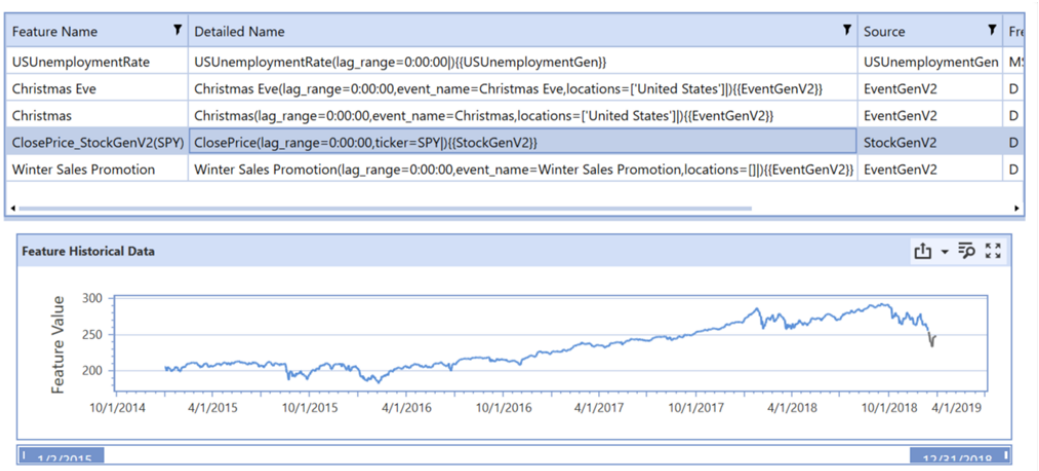
Status: The current status of the Feature/Event showing if it is ready to be used in a prediction. The available statuses are:

- **Not Ready**: Feature/Event value has not been provided, and a prediction cannot be run for this scenario.
- **Ready**: Feature/Event value has been provided and can be used to run a prediction.

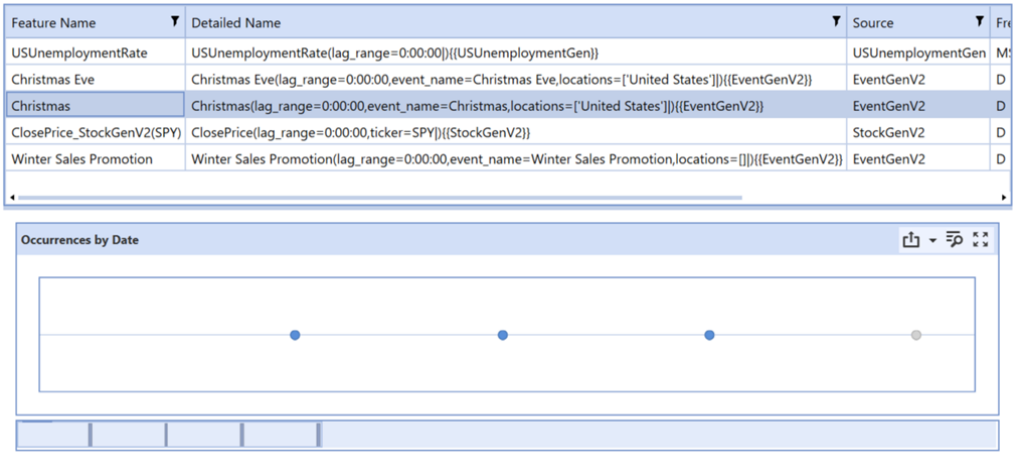
Explore Scenario Data

To view the historical and user input values of a Scenario, select different Features or Events from the **Scenario Features** table. With each selection change, the bottom pane will update with a time series view of Feature data and occurrence view of Event data:

Feature Time Series Data



Event Occurance Data



Feature Name	Detailed Name	Source	Fr
USUnemploymentRate	USUnemploymentRate(lag_range=0:00:00]){{USUnemploymentGen}}	USUnemploymentGen	M
Christmas Eve	Christmas Eve(lag_range=0:00:00,event_name=Christmas Eve,locations=['United States']]){{EventGenV2}}	EventGenV2	D
Christmas	Christmas(lag_range=0:00:00,event_name=Christmas,locations=['United States']]){{EventGenV2}}	EventGenV2	D
ClosePrice_StockGenV2(SPY)	ClosePrice(lag_range=0:00:00,ticker=SPY]){{StockGenV2}}	StockGenV2	D
Winter Sales Promotion	Winter Sales Promotion(lag_range=0:00:00,event_name=Winter Sales Promotion,locations=[]){{EventGenV2}}	EventGenV2	D

Run Predictions

Once at least one Scenario has a status of **Ready**, a prediction can be run using any Ready Scenarios. Reference the Manage Predict page section of the documentation for more detailed instructions on how to run predictions.

Scenario Analysis

Once at least one Scenario has been used to run a prediction, the results of that prediction run can be found in the Analysis section of Utilization. Please reference the Analysis section of the documentation, for more detailed instructions on viewing prediction results.

Local Excel Storage and Upload

While using Scenario Modeling, the Excel files generated by Xperiflow are available to be used locally and re-uploaded. Here is some information on how to use this capability:

- The upload functionality in the **Edit Feature Values** dialog box is intended for uploading a local version of the Xperiflow generated Excel file. This will not accept any custom .csv or .xlsx files.

- To get a local copy of the generated Excel file, use the **Save As** option in the Excel toolbar and select **Save As File on Local Folder**.

Utilization Phase Analysis Section

The Analysis section consists of these pages:

- **Forecast**: Analyze results from model forecasts, different builds, and different stages of the project.
- **Overview**: Provides general statistics on how well the deployed models are performing in utilization across all targets.

Analyze Forecast Results for Targets

This page includes Forecast, Feature Impact, Beeswarm, Tug of War, Waterfall, and Periodic Explanation pages. Use the Forecast page in the Analysis section to analyze results from model forecasts, different builds, and different stages of the project. This page visualizes the forecasted values for each model and back filled actuals for each target. It is similar to the **Arena Summary** view of the Model Build phase Pipeline section, but this page displays in-production results.

This page includes an Accuracy view (default), an Impact view, and an Explanation view. Use the fields at the top of the page to filter the information displayed on any of the views. These fields include:

> **NOTE:** Feature impact data is dependent on the type of model. Not all models have feature impact data.

Analyze Forecast Accuracy View Information

The Forecast view allows the user to view source actuals overlaid with the selected model's predictions. Error metrics are displayed in the correlated subplot below. Select a datapoint on the plot to view explanations for a given prediction..

This helps to answer questions such as:

- Which type of model has the best accuracy for a given target?
- By how much did the model win?

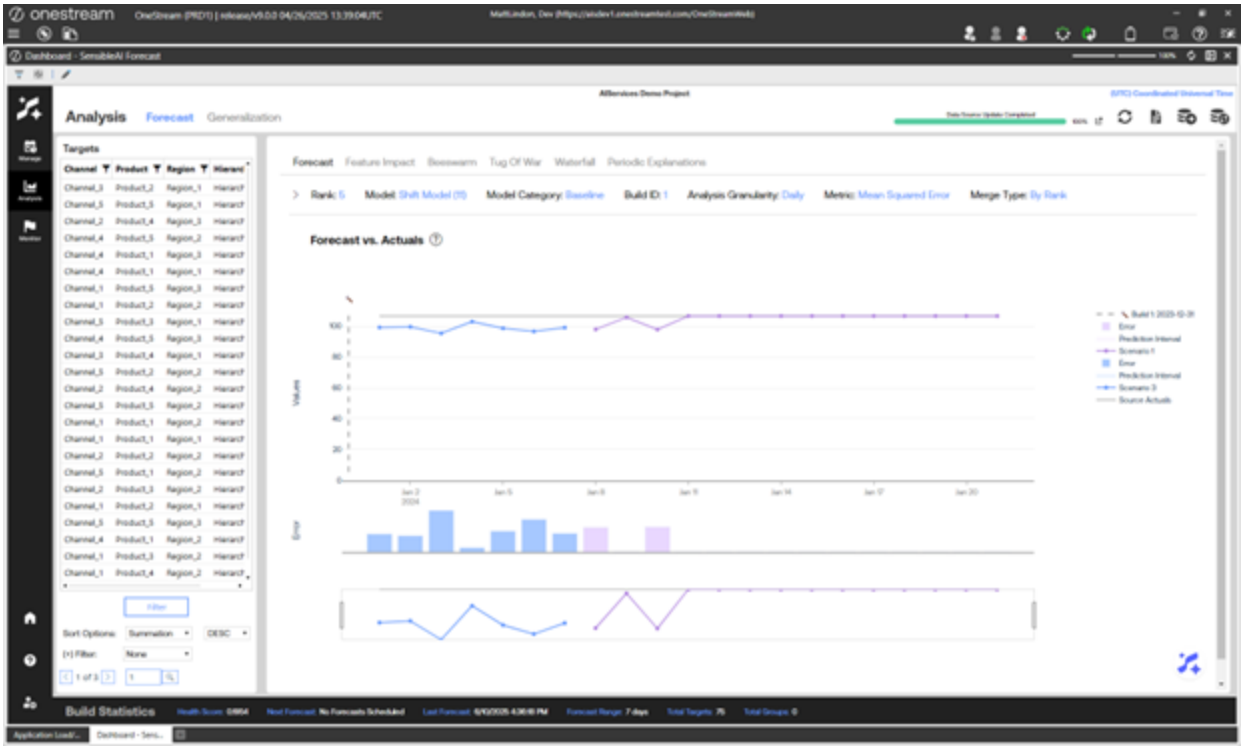
It also helps you understand how closely the forecasted values overlay the actuals in the line chart, which can provide answers to questions such as:

- Are there spikes that aren't being caught by the forecasts?
- If so, could adding any events help catch these spikes?

The table on this page displays:

- The model algorithms run for a given target (such as XGBoost, CatBoost, or Shift).
- The type of model algorithm (ML, Statistical, or Baseline).
- The evaluation metric (such as Mean Squared Error, Mean Absolute Error, and Mean Absolute Percentage Error) and the associated score.
- The train time (How long did it take to train the given model during Pipeline?).

With all the configurations and the newly found important features, the engine runs multiple models per configuration to find the best. This process involves hyperparameter tuning each model on multiple splits of the data and then saving the accuracy metrics of each model.



The line chart corresponds to the highlighted model in the table. It visualizes both the predictions made for the historical actual test period and the historical actuals. The time frame in this chart is only a subset of the total time frame for the historical data, as this time frame is for a specific portion of a split.

At a high level, this page provides the view of how the best version of each model (such as XGBoost, CatBoost, ExponentialSmoothing, Shift, and Mean) has performed against unseen historical data for each target and more specifically, on the test set of the historical data.

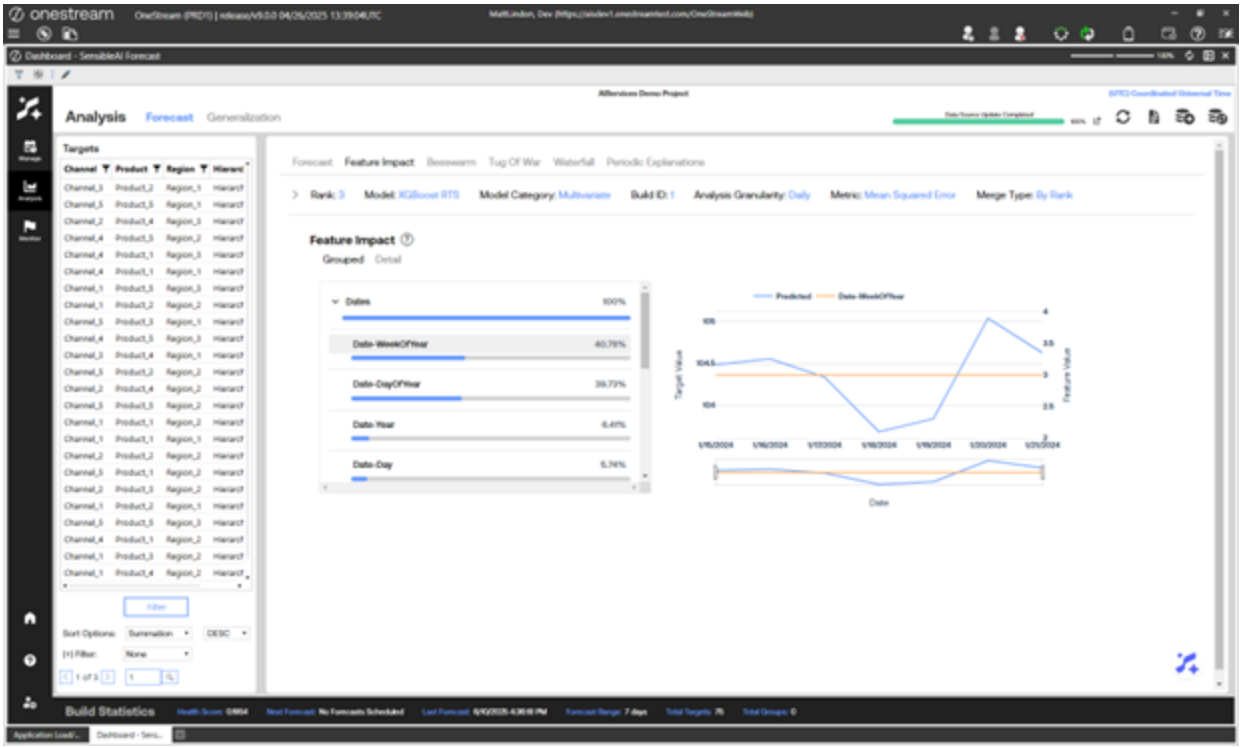
The optimal error metric score can be the lowest score, the highest score, or the closest to zero. It is dependent on the type of metric. See Appendix 3: Error Metrics for more information about the error metrics SensibleAI Forecast uses.

If you select a prediction point on the line chart, it will open a graph below providing prediction explanations for whichever data point was selected. This will include feature explanations along with the impact contribution of the feature when selected.

Analyze Feature Impact

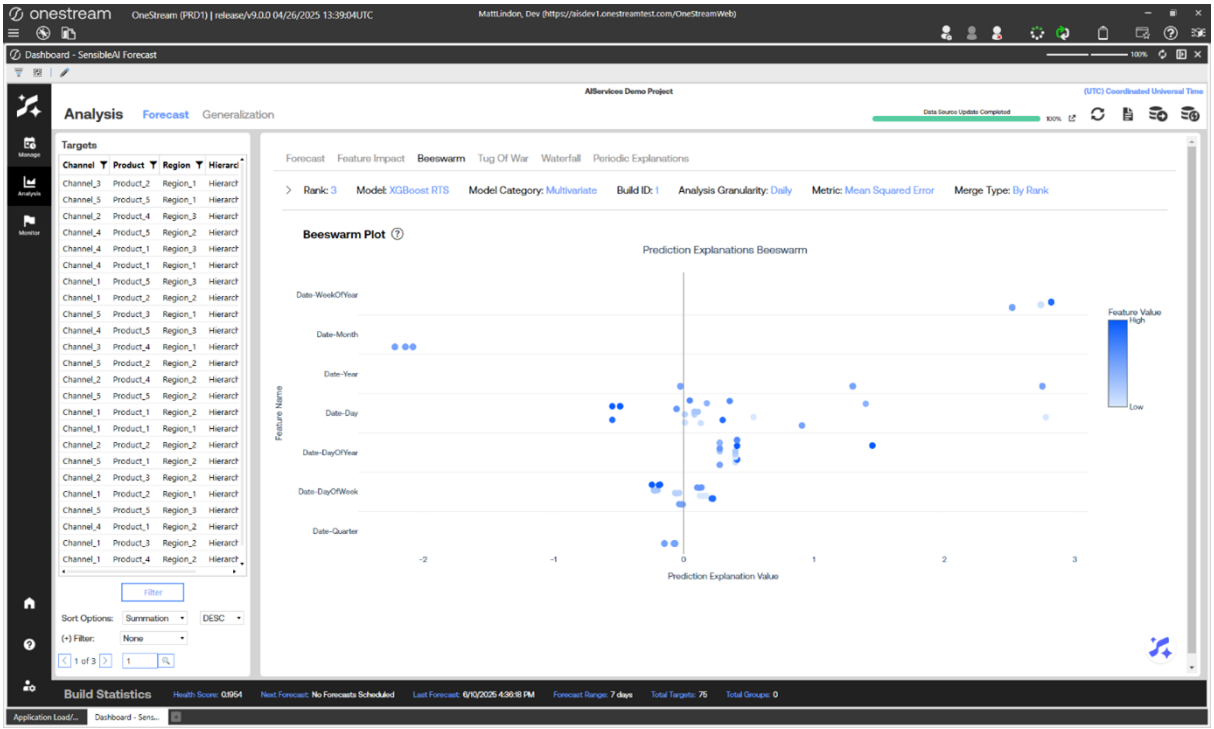
The Feature Impact view shows the hierarchical (Grouped) or individual (Detail) feature impact percentages aggregated across all the model's prediction values. The user can click on a feature name from the table to view its actual values compared with the predictions and actuals in the visual on the right. The prediction and actual values are bound by the primary y-axis on the left hand side of the graph, while the feature values are bound by the secondary y-axis on the right. The feature impact score shows how much influence the feature had for a given model.

> **NOTE:** Feature impact data is dependent on the type of model. Not all models have feature impact data.



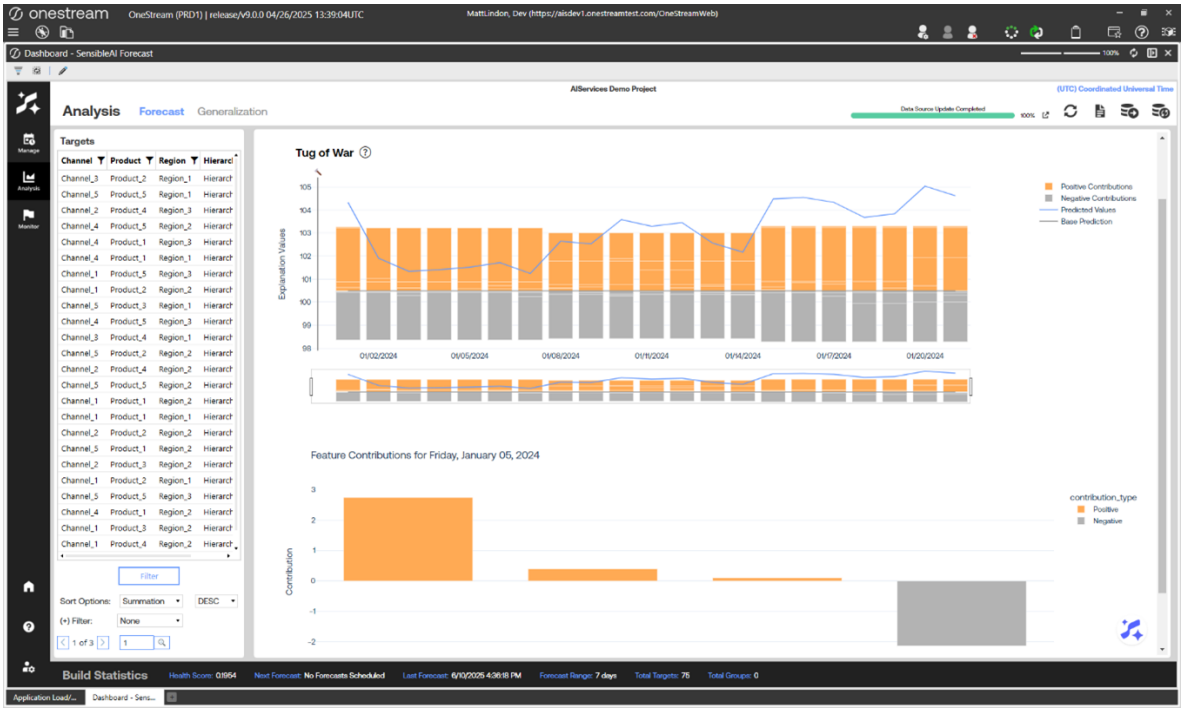
Analyze Beeswarm View

The Beeswarm view visualizes the impact of features on model predictions by displaying SHAP values as individual dots, where each dot represents a feature's SHAP value for a specific instance. Features are ordered by importance, with the most influential ones at the top. The color of each dot indicates the actual feature value. The horizontal spread of dots reflects the distribution of SHAP values for each feature, showing how much they contribute to model predictions. Plot is based on source granularity and changing the Analysis Granularity will not change the plot.



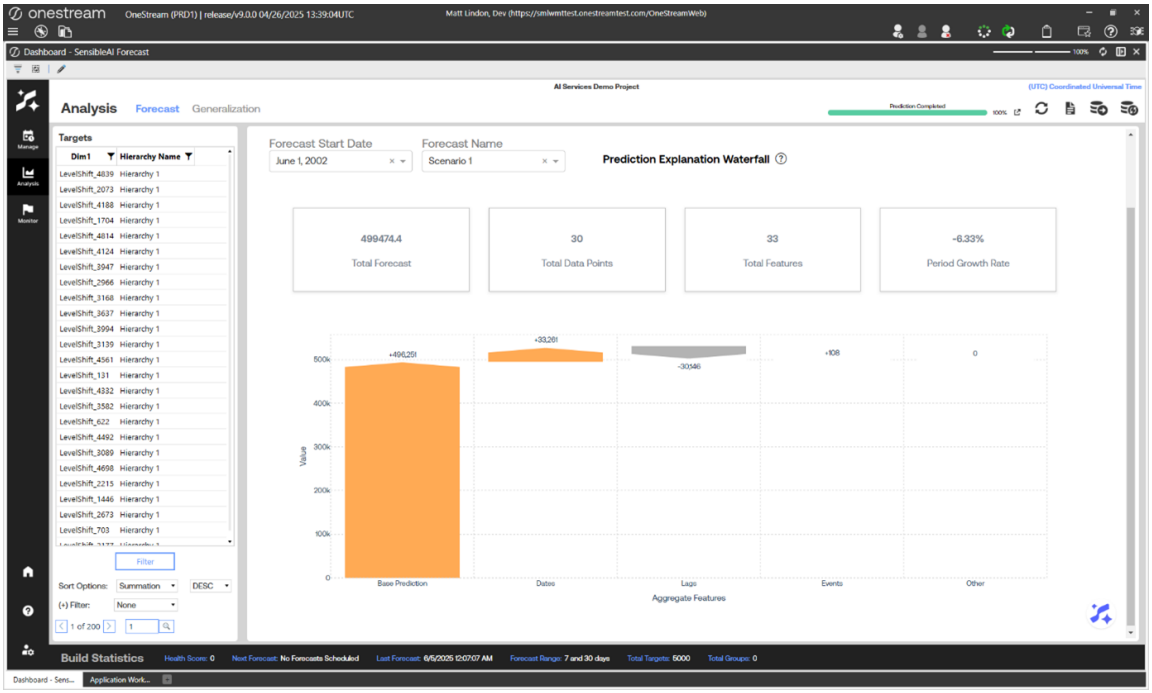
Analyze Tug of War View

The Tug of War view visualizes the four most significant positive and negative prediction explanations for all dates at the selected Analysis Granularity of the selected model. Compare with base predictions and actual prediction values for all dates. Users can toggle any component of the visual on or off by selecting it from the legend on the right. Users can also narrow in on specific dates by interacting with the range slider. By clicking on one of the bars in the graph, it will expand the feature contributions.



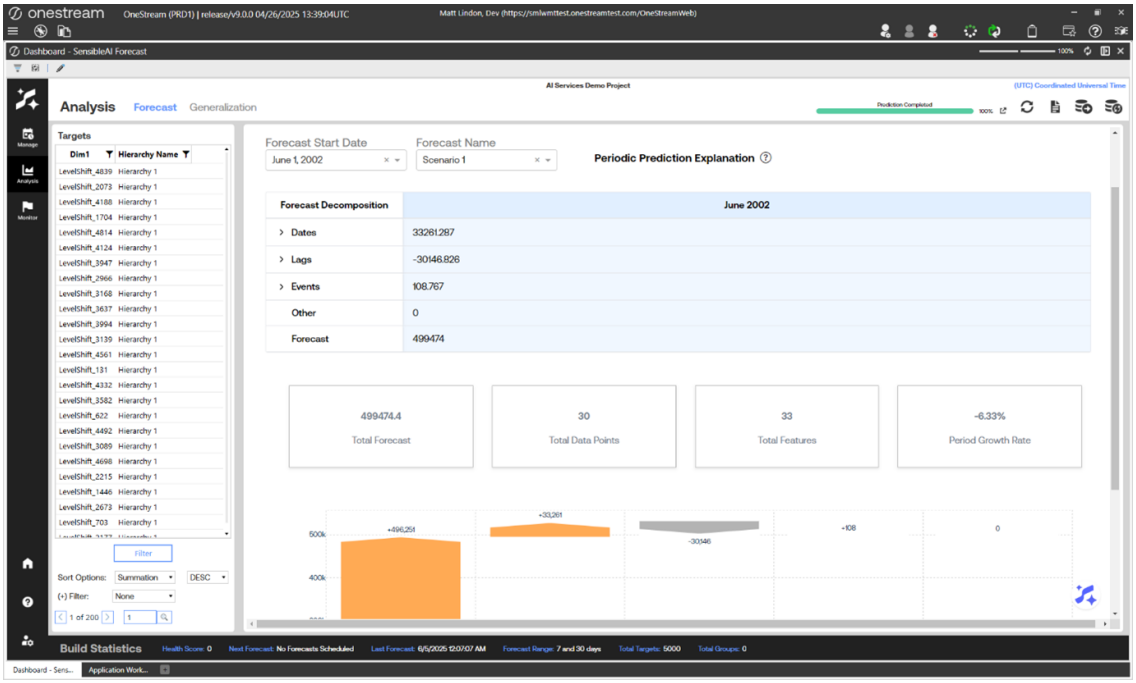
Analyze Prediction Explanations Waterfall View

View the aggregated predictions explanations for the selected forecast version of the selected model. Feature groupings are bucketed based on the Feature Grouping selection. Hover over a grouping to see the specific features that make up the given group. The figure below is only based on the source granularity. Only the Total Data Points will change when Analysis Granularity is changed.



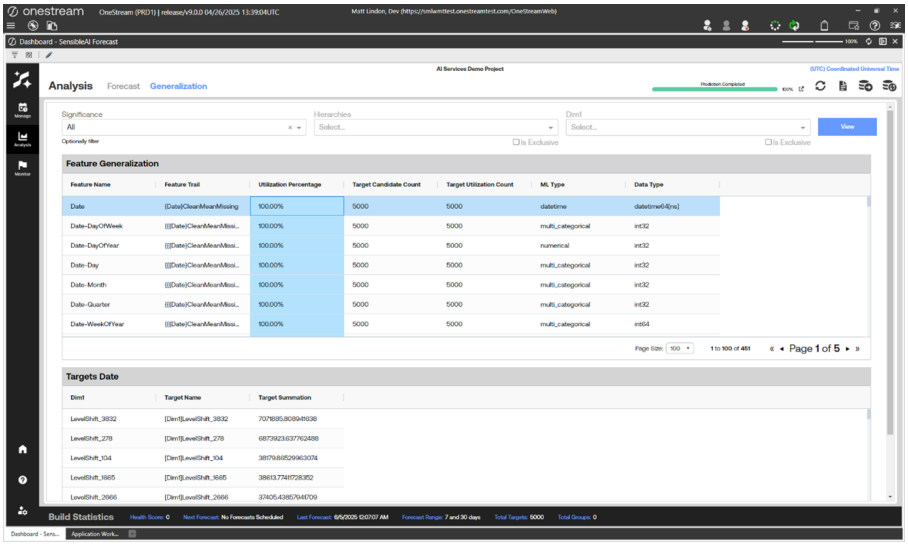
Analyze Periodic Explanations View

View the aggregated predictions explanations for the selected forecast version of the selected model. Feature groupings are bucketed based on the Feature Grouping selection. Click on a grouping to see the specific features that make up the given group. Click on a column header to generate a waterfall explanation chart for the given date range. The figure below is only based on the source granularity. Only the Total Data Points will change when Analysis Granularity is changed.



Analyze Generalization

The Generalization view gives insight into how widely used the features were across deployed targets. The Features Generalization grid in this view is a similar view to the Pipeline Features Generalization view. The use of different features displays as a percentage of total targets for which they are eligible.



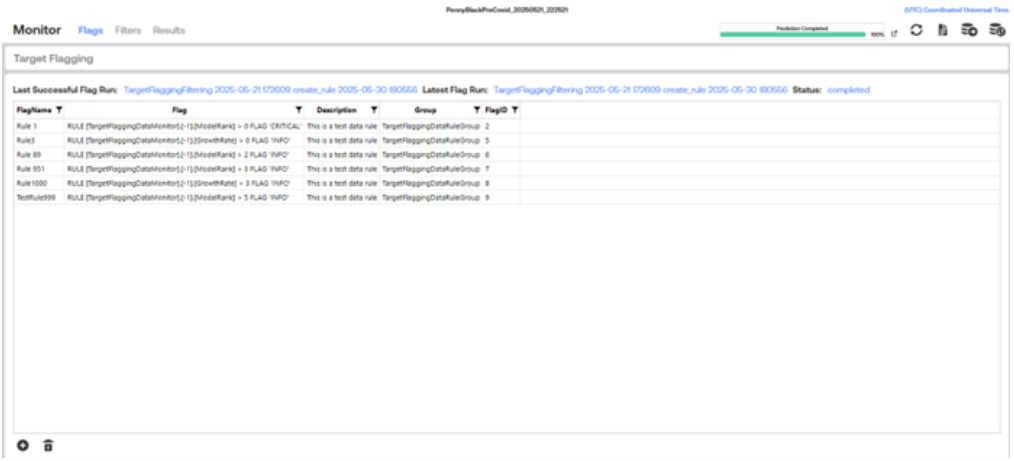
Utilization Phase Monitor Section

The Insights section consists of the following pages:

- **Flags**: Create new flags to analyze when models pass certain threshold values.
- **Filters**: Create new filters based on the created flags used to filter down each target to use one model post prediction runs.
- **Results**: Analyze the results of the analyzed flags and filters. Provides an understanding of the system filtered each target down to one model based on the specific flag/filter combination.

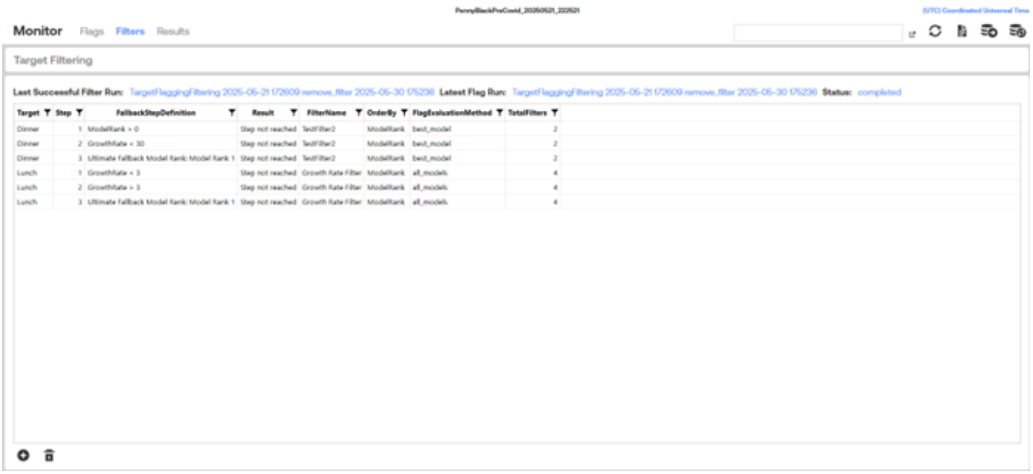
Create Flags for Targets

Flags are used to find models that have unwanted metric values. For example, if a user wants to find all models that have a low growth rate, they would create a flag that evaluates whether the growth rate for a specific model is lower than a specified value.



Create Filters for Targets

Filters will be used to filter down each specific target to one model post prediction runs. Targets can only have one filter but a filter can have multiple targets.



Analyze Monitor Results for Targets

Use the Results page in the Monitor section to analyze which model was selected for each specific target. This page will show how the filtering system decides which model should be picked for each target based on the created Filter. Each Filter has the following options when creating one:

Order by: How to order the models for filtering, defaults to model rank.

Flag to analyze: Which flag to tie to the filter. This flag will be evaluated on either the best model for the target or all models based on the input for models to evalute.

Model to evaluate: Whether to evaluate the best model or all models. If best model is selected, the filter will evaluate if the best model trips the flag tied to the filter. If it does not that model will be selected for that target. If all models is selected then the filter will find if any model does not trip the flag in order of whichever column is selected in the order by field. The “best” model that does not trip the flag will be selected. If the filter does not find any models that fit the flag, the filter will enter the fallback steps.

Dimension to apply: Pick which dimensions this filter will be applied to. Multiple targets can have the same filter, however a singular target cannot have multiple filters.

Fallbacks: Steps to take if there are no models that do not trip the flag specified for the filter.

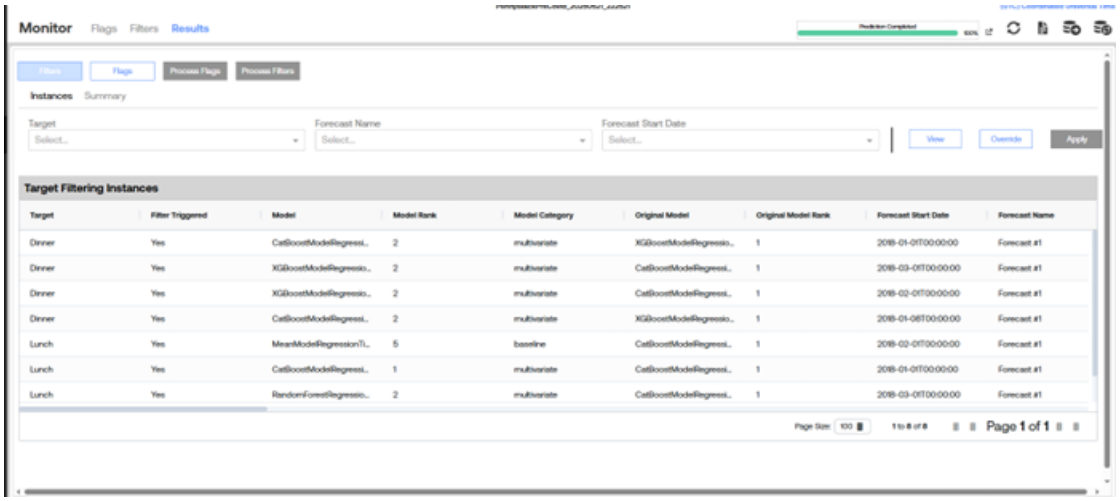
- Fallback steps:
    - How to “fallback” if no models are selected after evaluating the models for the flag specified on the filter. Each fallback can have more than one step. Example: “If column x > y and column d <= w.
    - Fact to evaluate
        - Which column/metric to analyze for the fallback step.

- Evaluation
  - Which comparison to use (greater than, less than, etc).
- Value
  - The value to compare to, like Absolute Error < 10.

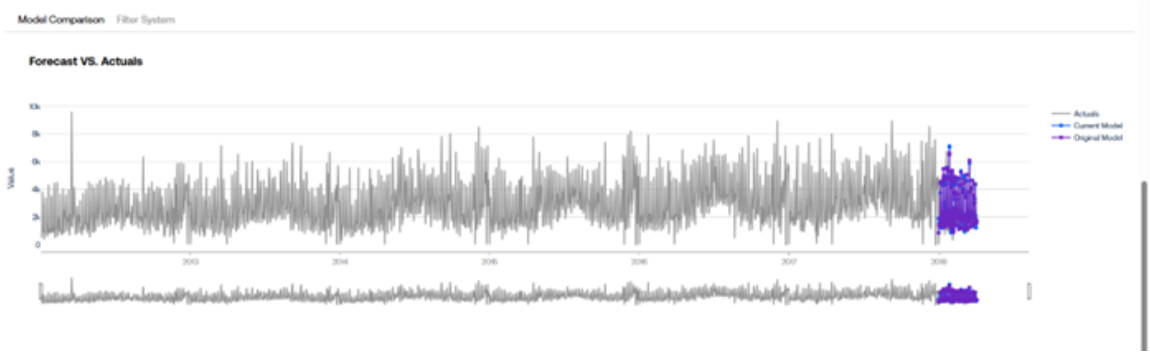
Ultimate Fallback Model: If there is no models that fit any criteria after both analyzing of the flag and the fallback steps, use the model selected here as an ultimate fallback. This input defaults to model rank 1, but also allows the user to select a specific model. If that model is not ran for the specific target being analyzed, then model rank 1 will be used.

Process Flags/Filters post creation

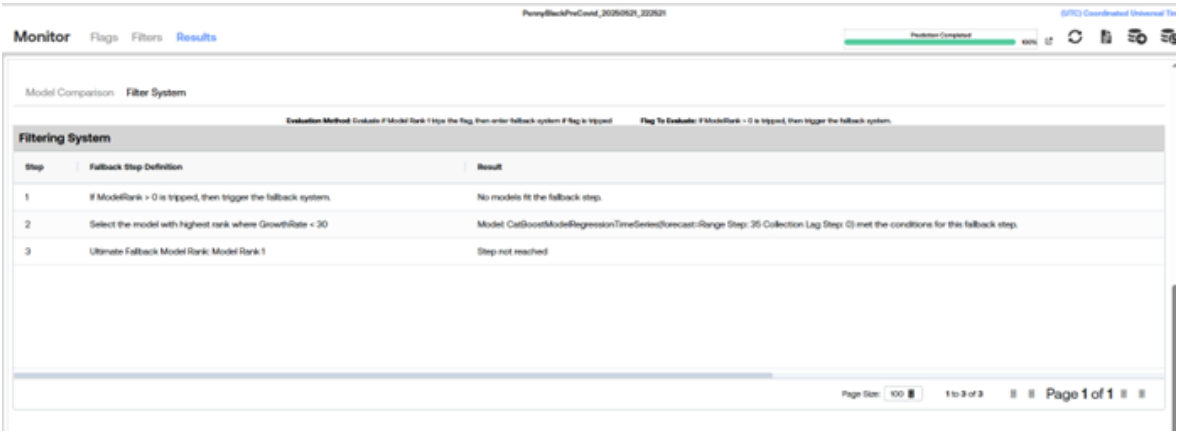
Once flags/filters are created, they will be able to be processed in the results page of the monitor section. When users process flags/filters, the buttons to process each system will be disabled until new flags or filters are created:



After Filters are processed, there will be data in both the Filters and Flags page that show the results of the evaluation on both systems. If only the flags are processed, there will be no data on the filters page. The filters page will show which model was selected for each target after processing, clicking on a row in the data table above will show the actuals for that target, the original model (model rank 1) as well as the selected model post filtering, aka the Model value in the selected row.

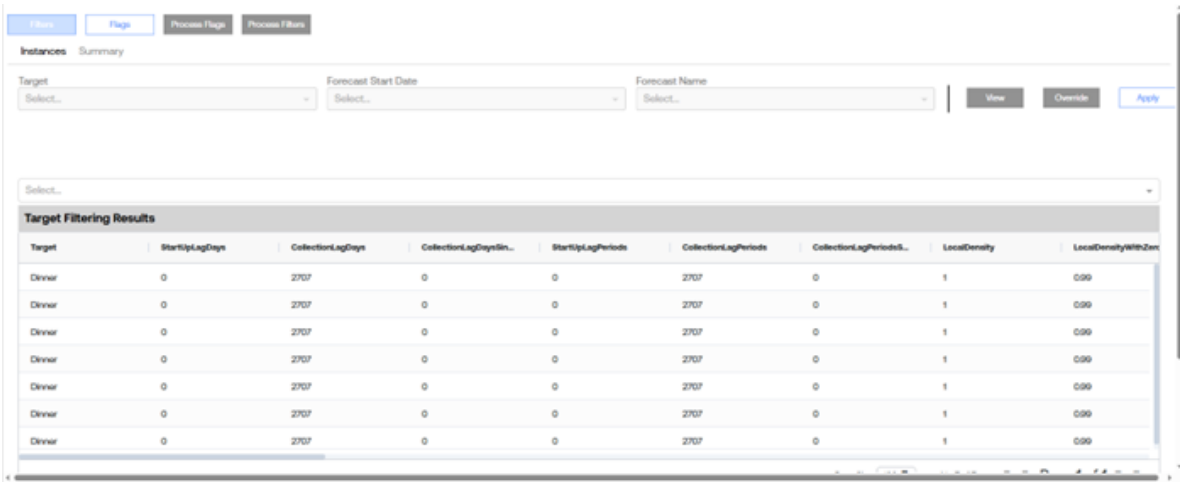


The filter system tab will show which step triggered the system to land on a certain model.



Create Model Overrides

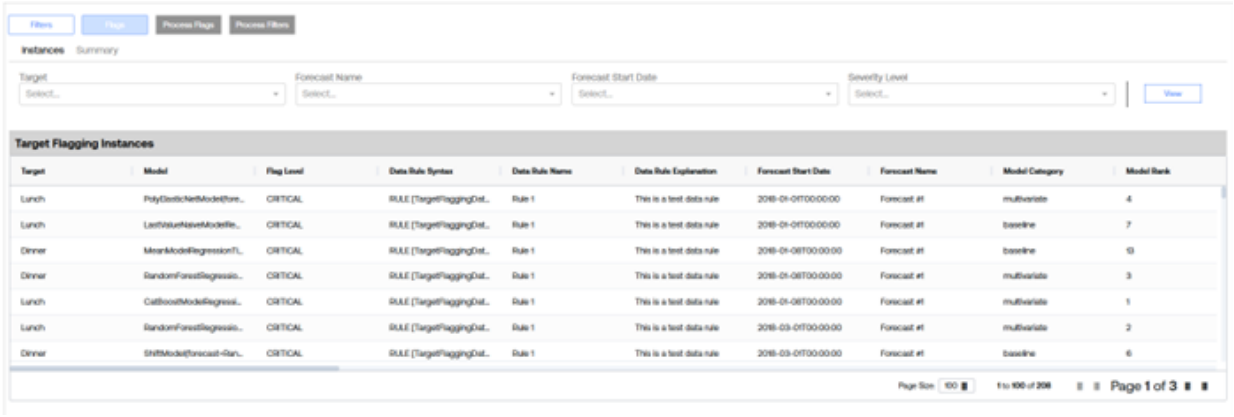
If after filtering the model for a specific target does not satisfy the user, there is an option to “override” the filter system in which the user can select another model to represent the target at hand by clicking on the override button in the top right corner once a row is selected on the table. The user will be able to select anyone of the models ran on the selected target, this view will show a table of all models ran on a target and the specific model metrics as well. Once a model is selected in the dropdown, there is an apply button in the top right corner that will apply the override to that target.



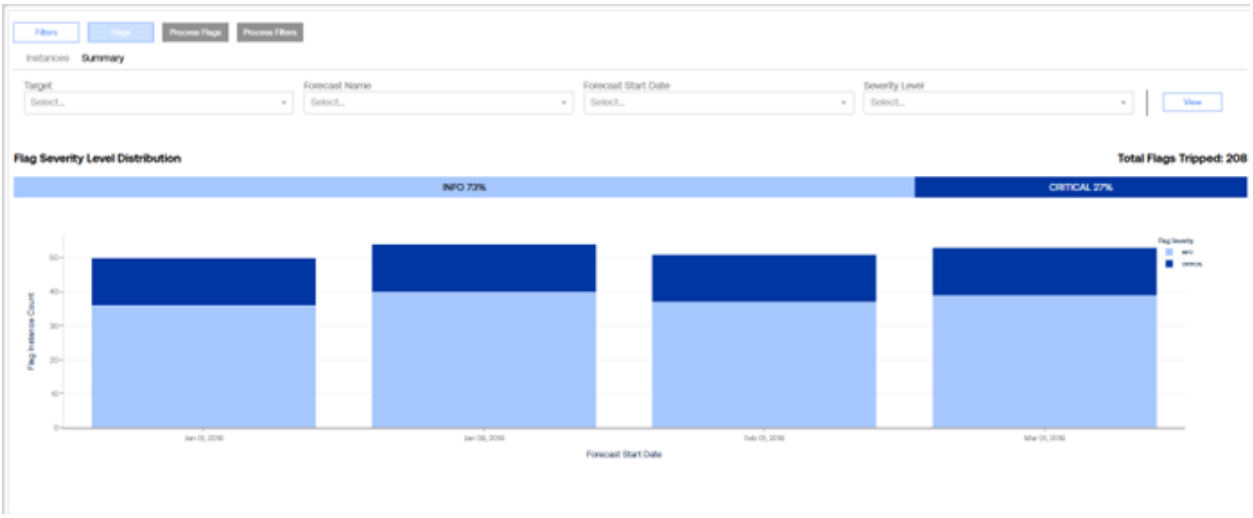
The summary tab in the filters page will show how many targets had filters tripped versus how many targets did not have filtered tripped:

Analyze Flagging Results

When either process flags or filters is complete users will be able to analyze the flags that were tripped for each model target combination.



The summary tab will show many of each type of flag was tripped for each different forecast start date, letting the user know if there are any forecast start dates that cause more flags to be tripped than another. Each severity level will have it's own color allowing users to be able to easioly distinguish how many of each type were tripped.


Instances
Summary
Target
Forecast Name
Forecast Start Date
Severity Level
View
Flag Severity Level Distribution
Total Flags Tripped: 208
INFO 73%
CRITICAL 27%
Flag Instance Count
Forecast Start Date

Help and Miscellaneous Information

Display Settings

OneStream Solutions frequently require the display of multiple data elements for proper data entry and analysis. Therefore, the recommended screen resolution is a minimum of 1920 x 1080 for optimal rendering of forms and reports.

Additionally, OneStream recommends that you adjust the Windows System Display text setting to 100% and do not apply any Custom Scaling options.

Package Contents and Naming Conventions

The package file name contains multiple identifiers that correspond with the platform. Renaming any of the elements contained in a package is discouraged in order to preserve the integrity of the naming conventions.

Example Package Name: FOR_PV9.0.0_SV402_PackageContents.zip

Identifier	Description
FOR	Solution ID
PV9.0.0	Minimum Platform version required to run solution

Identifier	Description
SV402	Solution version
PackageContents	File name

OneStream Solution Modification Considerations

A few cautions and considerations regarding the modification of OneStream Solutions:

- Major changes to business rules or custom tables within a OneStream Solution will not be supported through normal channels as the resulting solution is significantly different from the core solution.
- If changes are made to any dashboard object or business rule, consider renaming it or copying it to a new object first. This is important because if there is an upgrade to the OneStream Solution in the future and the customer applies the upgrade, this will overlay and wipe out the changes. This also applies when updating any of the standard reports and dashboards.
- If modifications are made to a OneStream Solution, upgrading to later versions will be more complex depending on the degree of customization. Simple changes such as changing a logo or colors on a dashboard do not impact upgrades significantly. Making changes to the custom database tables and business rules, which should be avoided, will make an upgrade even more complicated.

Appendix 1: Data Quality Guide

This section describes the importance and different aspects of data quality, how SensibleAI Forecast perceives the data based on varying levels of data quality, and how data volume affects the XperiFlow engine and SensibleAI Forecast functionality.

Why Data Quality Matters

The single biggest indicator of effective forecasting produced by SensibleAI Forecast is the quality of the source data. The following sections describe the components and best practices.

Collection Lag

Collection lag is the time between the most recent date in the data set (across all targets) and the latest received data for a single target.

The following image is a daily level data set containing a date column and two target columns named Apples and Bananas. As seen in the data set, the most recent date in the data set is 10/17/2021 (provided by Bananas). In this example, Apples has a collection lag of two days, which implies Apple's data is received two days after the most recent date in the data set.

A two-day collection lag is considered normal.

Date	Apples	Bananas
10/9/2021	100	82
10/10/2021	95	22
10/11/2021	64	14
10/12/2021	62	97
10/13/2021	29	89
10/14/2021	99	93
10/15/2021	43	73
10/16/2021		46
10/17/2021		(Most Recent Date) 60

Target Collection Lag: The time between the most recent date of the data set and when you receive the corresponding data for a specific target variable for that moment in time. In the following image, the target collections are: Apples – 2 days, Bananas – 0 days, Carrots – 4 days.

Date	Apples	Bananas	Carrots
10/9/2021	100	82	24
10/10/2021	95	22	73
10/11/2021	64	14	40
10/12/2021	62	97	10
10/13/2021	29	89	39
10/14/2021	99	93	
10/15/2021	43	73	
10/16/2021		46	
10/17/2021		(Most Recent Date) 60	

Data Collection Process

Having a good understanding of how your data is collected is extremely important. The quality of the data collection process determines the amount of effort needed to pre-process data prior to using SensibleAI Forecast.

The following sections describe what defines a good source data collection process.

Uniform Data Collection

Uniform Data Collection is defined as having a consistent data collection procedure for building the source data set.

Important questions to consider are:

- Is there a consistent data collection procedure across all business units or do business units implement their own practices?
- Do all business units report the same information, on the same frequency, and at the same time?

Ensuring that data is sourced using the same procedure and practices across all sources minimizes the effort to identify and correct any discovered inconsistencies. Overall, a highly fragmented and disjointed data collection process should be addressed and fixed before using a data source in SensibleAI Forecast.

Uniform Intra-Target Collection

Uniform intra-target collection is defined as a particular target maintaining the same data collection practices and procedures over time. Important questions to consider are:

- Does the frequency and the collection lag of the target remain consistent over time?
- Do the number of sources feeding a target remain consistent over the course of time?

Intra-target collection process integrity is important to maintain over the course of time. Otherwise, you risk the statistical and ML models mistaking a data collection inconsistency for changes in the underlying data pattern for that target.

The following example illustrates the difference between a non-uniform intra-target collection procedure versus a uniform procedure.

A clothing company that owns a variety of clothing brands wants to predict the unit sales of shirts and pants. They have historical retail sales data dating back to 2014. In 2017, this clothing company merged Brand B that they own with Brand A (having Brand B be absorbed by Brand A).

Consolidate Data at Merger (Non-Uniform)

The following table shows a non-uniform intra-target collection pattern. It is non-uniform because from 01/01/2017 onward there are two sources feeding BrandA-Shirts and BrandA-Pants. Before 01/01/2017 there was only one source. This fundamentally changes the collection process for these targets.

Date	BrandA-Shirts	BrandA-Pants	BrandB-Shirts	BrandB-Pants
10/1/2014	30	14	17	13
10/2/2014	35	15	21	16
...	...	...	...	...
12/30/2016	34	18	20	15
12/31/2016	36	20	22	17
1/1/2017	60	38	DNE	DNE
1/2/2017	59	35	DNE	DNE
1/3/2017	58	37	DNE	DNE

The problem with this method is that the models that run against BrandA-Shirts and BrandA-Pants are not aware that this merger happened. The models assume that BrandA targets magically and organically doubled their sales at the start of the year. In future projections, the models may see this doubling as a common occurrence and predict this to happen at the start of every year which would be wrong.

The Correct Way: Backdate the Consolidation of Brands to the Beginning of the Data (Uniform)

This is the correct way to handle this merger from a machine learning data perspective because it maintains uniformity of target data collection over time. With the uniform option, even though the merger officially occurred on 01/01/2017, the values of Brand A and Brand B are aggregated back to the beginning of the data set.

Date	BrandAB-Shirts	BrandAB-Pants
10/1/2014	47	27
10/2/2014	56	31
...	...	...
12/30/2016	54	33
12/31/2016	58	37
1/1/2017	60	38
1/2/2017	59	35
1/3/2017	58	37

This has two benefits over a non-uniform intra-target collection.

- The models are trained off the combined Brand A and B which is the case moving forward.
- This removes the Shutdown Brand B from the data set that serves no purpose moving forward and should not be receiving predictions.

Data Set Frequency

Data Set Frequency refers to the time frequency of the overall data set. The time frequency is set based on the target that has the most granular level data.

It is expected that the frequency of an entire data set remains constant across all targets. If a data set frequency is not constant across all targets, the most granular frequency target determines the overall data set frequency. The targets that are of a less granular frequency are based on the configured cleaning method selected in the **Configure** > **Model** page in the Model Build phase to get a complete series of the same frequency as the most granular data.

This is illustrated by the following two data sets.

(Left Dataset)

Date	DayOfWeek	Target A	Target B
1/4/2021	Mon	7	32
1/5/2021	Tue	4	
1/6/2021	Wed	7	
1/7/2021	Thu	10	
1/8/2021	Fri	11	
1/9/2021	Sat	5	
1/10/2021	Sun	7	
1/11/2021	Mon	6	49
1/12/2021	Tue	2	
1/13/2021	Wed	4	

(Right Dataset)

Date	DayOfWeek	Target A	Target B
1/4/2021	Mon	7	32
1/5/2021	Tue	4	40
1/6/2021	Wed	7	40
1/7/2021	Thu	10	40
1/8/2021	Fri	11	40
1/9/2021	Sat	5	40
1/10/2021	Sun	7	40
1/11/2021	Mon	6	49
1/12/2021	Tue	2	40
1/13/2021	Wed	4	40

Mean Imputation

The Left Dataset is the raw data given to SensibleAI Forecast. The Right Dataset is the data set processed by SensibleAI Forecast after the initial data load. In this scenario, Target A is a daily frequency and determines the underlying frequency of the entire data set. This means all other targets are expected to also be daily frequency. If any additional targets are not of the same frequency, SensibleAI Forecast fills their missing values based on the configured cleaning method selected in the **Configure** > **Model** page in the Model Build phase. Target B illustrates this as a weekly granularity in the Left data set and being filled with the mean value of the entire column in the Right Dataset.

Data Collection Best Practices

The most concise way to describe the best practices is to avoid all the data quality problems shown in Data Quality, Data Collection Process, and Data Set Frequency.

Additional best practices include:

Use the Same Target Collection Lags for all Targets
All targets should have close to the same target collection lags. If there are large differences in collection lags across targets, break up the data into two or more separate projects where you can separate the project based on similar target collection lags across the data.

Ensure the Target Collection Lag Remains Constant Over Time
The source data should be routinely updated at a consistent interval. This minimizes the need to interpolate the most recent dates. If the actual collection lag changes, then you must reconfigure the models by doing a full model rebuild.

Provide Complete Data
It's okay to fill a few missing values or a few partial sections of target data if they can be reasonably interpolated. If greater than five percent of data is missing, performance can become unstable for targets. This is because models are learning from fake interpolated patterns found in the data that may not match. The more complete the data is, the better the forecast results.

Do Not Use Fake Data

No reasonable model accuracy can be assumed if source data is manufactured to represent real data.

For example, take a target that is only available at a yearly frequency and guess its allocation at a monthly frequency, then provide this to SensibleAI Forecast as a monthly frequency target. The model would learn from a fake monthly variation that most likely does not match the monthly reality. Therefore, you cannot reasonably assume that the model accuracy for that target at the monthly level produces accurate forecasts.

In general, if the data provided to SensibleAI Forecast (or any model) does not match the reality of the historical data patterns, then no reasonable forecasts should be expected from SensibleAI Forecast for those targets. Learning from fake data can lead to inaccurate results.

Ensure Uniform Data Collection Practices

Ensure that all data collection practices remain consistent across the entire history and future of the source data. This ensures that models have consistent data patterns to learn from. A model can mistake a change in the data collection pattern as a new trend or seasonal data pattern which can cause inaccurate results.

Ensure a Constant Frequency Across All Targets

Ensure that all targets included in the source data are of the same frequency. Any targets that are less granular than the data set frequency produce inaccurate results.

Do Not Change Source Data While SensibleAI Forecast is Running

Changing the data source targets while a SensibleAI Forecast job is running may cause SensibleAI Forecast to stop responding. This is because SensibleAI Forecast avoids making a copy of the data source used to power the solution.

Align the Target Units to the Business Problem

It is important to align the target units to the business problem that SensibleAI Forecast is being used to solve.

For example, if the downstream use case is supply chain demand planning, it is best to have the targets' units be unit sales rather than dollar sales. This is because the raw units sold more closely align to the downstream use case when estimating how much product to move to certain retail locations.

Using dollar sales does not effectively align to this use case and creates these challenges:

- Models are expected to learn price appreciation or changes in the price per unit. Price-per-unit changes are a form of non-uniform target collection which should be avoided.
- The forecast dollar sales must be converted back into unit sales to align to the demand planning use case. This can lead to conversion errors, which complicates the inherent error that exists in any forecast.

Data Volume

This section explores all aspects of data volume, from an educational perspective, to show how data volume components influence performance and use case alignment.

The size and shape of the source data set determines:

- Data patterns that can be learned.
- Model algorithms that can be leveraged.
- How far forward you can accurately forecast data.
- Effectiveness of features (external variables).
- Model train and build time.
- Overall model accuracy and use case performance.

Data Granularity and Learnable Data Patterns

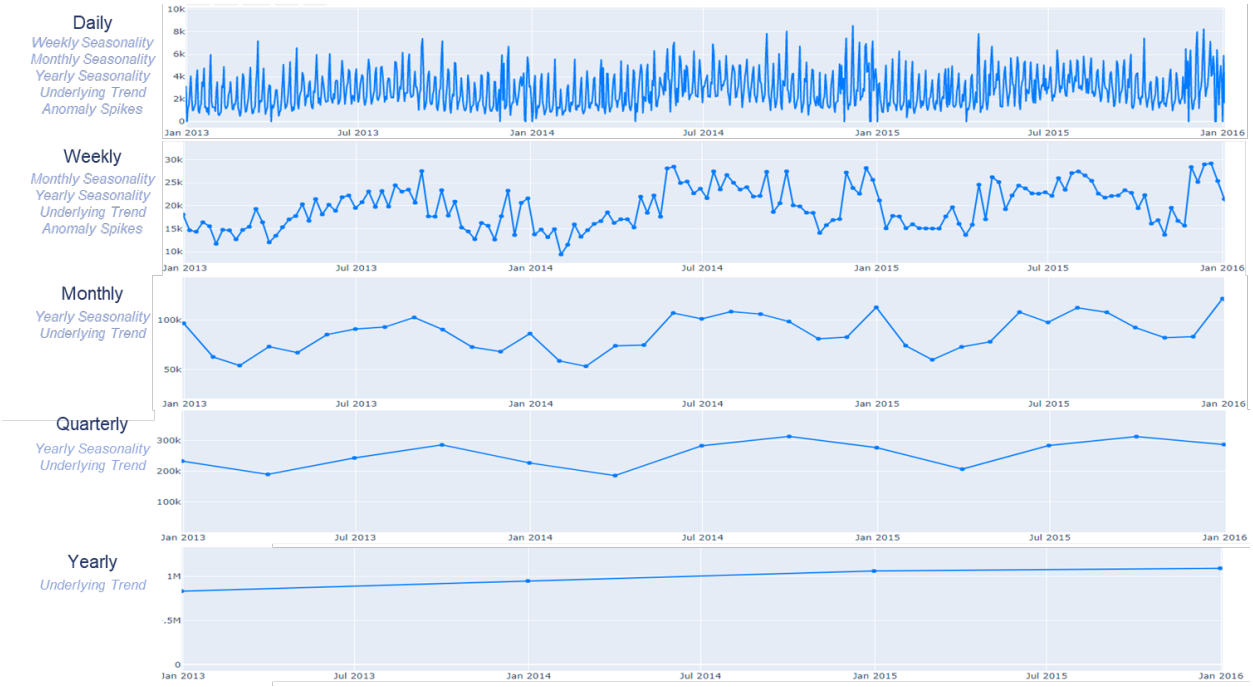
Before understanding the influence of data volumes, it is important to understand how data granularity determines what data patterns can be learned by models. A *data pattern* is an underlying structure of a time series.

Common data patterns include:

- **Seasonality**: A repeated pattern occurring on a constant frequency, for example weekly seasonality where the same sort of high and low point would occur on the same day of the week. There can be many forms of seasonality occurring within the same time series.
- **Trend**: An underlying slope (linear or non-linear) that increases or decreases the time series average over time.
- **Anomalies** (also known as outliers): An explainable or unexplainable data pattern that deviates from the normal seasonality or trend. These are typically one-off high or low points. In more extreme cases, these can surface as longer-term data shifts lasting weeks, months, or years.

  It is best to address anomalies either by removing them or using features and events to inform models of their occurrence.

The following graphic overlays the same sales data at varying levels of aggregation (daily, weekly, monthly, quarterly, yearly). As aggregations continue up to yearly, different forms of seasonal variation and anomalies become hidden.



It is important to use the right tools for the forecasting job. Sophisticated machine learning models perform best when multiple seasonal variations and an underlying trend exist. Machine learning models leverage features to learn intricate seasonal patterns that exist within a time series to get the best line following result. Machine learning models are better suited for short range demand planning scenarios, where high daily forecast accuracy is needed, compared to more statistical based models.

Statistical based models are better suited for monthly, quarterly, or yearly data sets with fewer data patterns. This makes statistical models better at solving long range or growth-based use cases, since the business cares more about the underlying trend.

The following graphic shows general implications of fine grained versus coarse grained data granularity.

Data Volume Definitions

Target Historical Data Points: The number of historical actuals in a target data series. A target that has monthly frequency going back five years would have 60 data points (12 * 5 = 60).

Data Set Historical Data Points: The maximum number of data points across all targets in the data set.

Training Depth: A scale of 1 to 5 of how long models will train. The larger the training depth, the more iterations of the model will run. This may lead to better model accuracy, but with longer train times.

The following graphic summarizes how the number of data points, model training depth, and number of targets, influence various components used to build and utilize models for generating predictions.

Impact of Data Points and Data Granularity

The number of data points determines the functionality that can be used during the model build process. For data sets of coarser data granularity, SensibleAI Forecast adjusts to not waste resources on functionality that does not contribute to model performance. For example, SensibleAI Forecast does not run machine learning models on a monthly data set because there are too few seasonal data patterns to learn from monthly level data. SensibleAI Forecast is optimized to produce the best performance out of any level of data set provided.

Functional Overview by Data Granularity

XperiFlow Engine Functionality by Data Point Range

Total Data Points	16 – 36 Data Points	36 – 80 Data Points		80 – 300 Data Points		300+ Data Points	
Data Granularity	Quarterly/Monthly	Monthly/Weekly		Weekly/Daily		Daily	
Train Data Points	< 80 Train Data Points	< 80 Train Data Points	>= 80 Train Data Points	< 80 Train Data Points	>= 80 Train Data Points	< 80 Train Data Points	>= 80 Train Data Points
Auto ML	☑	☑	☑	☑	☑	☑	☑
Data Cleansing	☑	☑	☑	☑	☑	☑	☑
Event Builder	☐	☐	☐	☑	☑	☑	☑
Auto External Data Collection	☐	☐	☐	☑	☑	☑	☑
Auto Feature Engineering	☐	☐	☐	☑	☑	☑	☑
Univariate Models (Statistical Models)	☑	☑	☑	☑	☑	☑	☑
Multivariate Models (ML Models)	☐	☐	☑	☐	☑	☐	☑
Multi-Series Models (Target Grouping)	☐	☐	☐	☑	☑	☑	☑
Allows External Features	☐	☐	☐	☑	☑	☑	☑
Baseline Comparisons	☑	☑	☑	☑	☑	☑	☑
Backtest Performance	☐	☐	☐	☐	☐	☑	☑

16 to 36 Data Points (Quarterly to Monthly)

This is the most restrictive model build pipeline offered. The data is most likely monthly data with only one to three seasonal cycles. All models offered for this data point range are univariate, simple models, meaning they are trend-based models with no features to enhance predictive capability.

Data: Most likely monthly data with little to no seasonality. Typically, only one to three seasonal cycles exist within the number of historical data points. Limited seasonality implies that models trained against this data rely exclusively on learning the underlying trend.

Functionality: The functionality for this data point range is the most restrictive offered. Functionality is restricted due to low data volumes. Models that can be used with low data volumes do not accept features. Therefore, events, locations, external data, external features, feature engineering, and feature selection cannot be provided or run in SensibleAI Forecast for these data volumes.

With the 16 to 36 data points range, there may be enough data for a validation set for training to compare model accuracy. This is not a perfect situation, since this validation set is being used for model hyperparameter tuning and selecting the best model, during training.

Since there is not enough data that can be held out during training to compare model accuracy, backtest accuracy comparisons are deactivated on the **Deploy** page. Grouping functionality is also not available since there are no grouping models that can be run.

Features: Features cannot be used for this data point range. This is because there is not enough data to learn any meaningful relationships between any feature and target variable.

Models: Models in this data point range are basic statistical models that extract an underlying trend or basic seasonality. Given the limited amount of data to train on, these models typically forecast conservatively and do not project aggressive increases or decreases in the underlying trend. With only one to three yearly seasonal cycles typically existing with this amount of data, the models struggle to find the seasonality with accuracy. The models typically catch only apparent seasonality patterns spotted over at least three seasonal cycles.

Models running against this amount of data can train much faster than larger data point ranges and more complex models.

Model Selection: Model selection can be difficult for models with a low number of data points. There is no back-test procedure or train-test procedure that can validate which models will perform the best in production outside of using the validation set once there are 20 data points.

The best performing model is chosen based on the best fit for the validation data if available. This implies that models could be overfit to the training data. To deal with this, SensibleAI Forecast selects models from the Model Arena of the model build phase based on some of the structural integrity of a particular target. For example, if the historical data offers little to no repeated seasonality, the seasonal statistical models do not run against that target in the Model Arena.

Selecting the Models to Train: SensibleAI Forecast inspects a target's data patterns to determine which models should be allowed to train. Inspecting the data patterns includes determining whether there is seasonality in the existing underlying data. SensibleAI Forecast only trains models that match the structural data patterns of the data.

Selecting the Best Trained Model: The best trained model is chosen by whatever model best overfits the training data or validation data, if applicable. The effectiveness of the Selecting the Model to Train process is very important.

Use Cases: This data point range is best associated with long term growth, high-level or long-range planning, or strategic growth use cases, since there are only underlying trends and slight seasonality.

It is recommended to leverage all the models for your targets. This gives downstream users from SensibleAI Forecast the flexibility to choose the model forecasts to use.

36 to 80 Data Points (Monthly to Weekly)

This is the second most restrictive model build pipeline offered. The data is typically monthly or weekly with three or more seasonal cycles. All models offered for this data point range are univariate simple trend and seasonal models. This means these models can detect a single seasonality with an underlying trend. However, there are still not enough underlying data patterns to warrant the use of features.

Data: Most likely monthly or weekly level data with a single seasonality and trend.

Functionality: The same limitations for the 16 to 36 data point range exist with a few exceptions.

With the 36 to 80 data points range, there is enough data for a validation set for training to compare model accuracy. This is not a perfect situation, since this validation set is being used for model hyperparameter tuning and selecting the best model during training. Backtest accuracy comparisons are deactivated on the **Deploy** page. Grouping functionality is not available since there are no grouping models for this data point range.

Features: Features are not allowed for this data point range. This is because there is not enough data to learn meaningful relationships between any feature and the target variable, since there are minimal data patterns to learn.

Models: Models for this data point range are statistical models that extract underlying trends and some seasonality. With limited data to train on, these models typically forecast conservatively and do not project aggressive increases or decreases in the underlying trend. These models typically only catch seasonal patterns spotted over at least three seasonal cycles.

Models running against this amount of data can train very fast, compared to larger data point ranges and more complex models.

Model Selection: As mentioned in the 16 to 36 Data Points (Quarterly to Monthly) section, the same difficulties apply to model selection and Selecting the Best Trained Model in most situations. However, as the data points approach 20 or more, SensibleAI Forecast uses a validation data set for hyperparameter tuning.

> **Selecting the Models to Train**: SensibleAI Forecast inspects the target's data patterns to determine which models should be allowed to train. This includes determining whether there is seasonality or an underlying trend that exists in the data. SensibleAI Forecast only trains models that match the structural data patterns.
>
> **Selecting the Best Trained Model**: The best trained model can be chosen by whatever model overfit the most to the training data for 20 or less data points. If there are more than 20 data points, a validation set is used for hyperparameter tuning and choosing the model that had the lowest error on this section.

Use Cases: This data point range is best associated with long term growth, or high-level or long-range planning use cases given that there exists an underlying trend and single seasonality. This provides some understanding of the months or time periods within a given year that are spiking or dipping.

80 to 300 Data Points (Weekly to Daily)

A data set with 80 to 300 data points can leverage almost all capabilities of SensibleAI Forecast.

Data: The data will most likely be daily level data with multiple data patterns that can be learned.

These data patterns consist of:

- **Seasonality**: Multiple seasonalities may exist within the data. This may be overlaid seasonality of weekly, monthly, quarterly, or yearly.
- **Trend**: An underlying change in the mean value over time.
- **Anomalies**: Spikes or dips in the data that can be explained by re-occurring events and holidays.

These data patterns may be difficult to learn since there are likely not enough data pattern repetitions. For example, a daily level data set with only 300 data points does not have a complete picture of yearly seasonality. Therefore, a model running against this data set most likely cannot assume any yearly seasonality exists, even if it does exist.

Functionality: Almost all the functionality available in SensibleAI Forecast may be leveraged. The cross-validation strategy used improves the closer you get to 300 data points.

Features: SensibleAI Forecast uses all possible feature types to get the most highly performing models. However, weekly level data sets may not see an effective benefit from event-based features since events occur daily.

Models: All different model types can be leveraged with at least 80 data points in the train set of the largest split. Models that run against these 80-300 data points are typically a mix of machine learning and statistical models. This data point range blends the usage of models that leverage features and pit them against models that do not use features.

Model Selection: The models that perform best are typically ML models or more advanced statistical models, since there is a decent amount of data patterns to learn from.

> **Selecting the Models to Train**: SensibleAI Forecast gets a list of candidate models to run in the Model Arena. It defaults to running a recommended set of machine learning models. Two of these are XGBoostTimeSeries and CatBoostTimeSeries. After the models have been selected, the Model Arena then trains these models and compares them against each other. SensibleAI Forecast also includes common baseline models (shift and mean models).
>
> **Selecting the Best Trained Model**: In the Model Arena, the cross-validation strategy used improves the closer you get to 500 data points.

Use Case: This data point range is best associated with an annual demand plan. This is because SensibleAI Forecast can learn a fair amount of seasonal data patterns which provide accurate forecasts on any given day or weekly interval.

Daily granular data sets within this data point range may struggle to produce accurate forecasts longer than six months given that there may not have been enough daily history to learn yearly seasonal data patterns.

300+ Data Points (Daily)

All capabilities of SensibleAI Forecast are unlocked for data sets with more than 300 data points.

Data: The data is most likely daily level data with multiple data patterns that can be learned.

These data patterns consist of:

- **Seasonality**: Multiple seasonality may exist within the data. This may be overlaid seasonality of weekly, monthly, quarterly, or yearly.
- **Trend**: An underlying change in the mean value over time.

- **Anomalies**: Spikes or dips in the data that can be explained by re-occurring events and holidays.

Functionality: All functionality of SensibleAI Forecast is available for data sets with more than 500 data points.

Features: SensibleAI Forecast leverages all possible feature types to get the most performing models.

Models: All different model types can be leveraged with at least 80 data points in the train set of the largest split. Models that run against data with more than 300 data points take the longest to train. This is because the models running against larger data sets have more parameters to tune, more data for models to consume, and more data patterns to learn. The train time duration per target is higher than other data point ranges.

The models that perform best here are typically machine learning models due to high volume of data patterns to learn from.

Model Selection

Selecting the Models to Train: Before running the Model Arena with 300+ data points, SensibleAI Forecast gets a list of candidate models to run in the Model Arena. It defaults to running a recommended set of machine learning models. Two of these models are XGBoostTimeSeries, and CatBoostTimeSeries. After the models have been selected, the Model Arena trains these models and compares them against each other. SensibleAI Forecast also includes common baseline models (shift and mean models).

Selecting the Best Trained Model: In the Model Arena, a comprehensive cross-validation strategy is leveraged for hyperparameter tuning and model validation to determine which models are the most likely to perform best in production. A nested time series cross-validation strategy is leveraged.

Use Cases: This data point range is best associated with annual demand planning or operational level demand planning use cases. This is because SensibleAI Forecast can learn from multiple seasonal data patterns which provide highly accurate forecasts on any given day. These granular forecasts can be used to drive operational level decisions.

Grouping: Modeling Targets Together

By default, SensibleAI Forecast builds at least one model per target. It allows grouping on the **Dataset** page of the Model Build phase which is advanced functionality that allows targets to be grouped and treated as if they are a single target. Grouped targets are trained using multi-series models which work by treating the target name as a feature and collective target values as the target column.

The following graphic illustrates what a grouped data set looks like:

TargetA	TargetB	TargetC	Date		TargetName	TargetValue	Date
7	14	20	1/1/2019		TargetA	7	1/1/2019
5	8	18	1/2/2019		TargetA	5	1/2/2019
3	4	15	1/3/2019		TargetA	3	1/3/2019
...	...	...			TargetA	...	...
					TargetB	14	1/1/2019
					TargetB	8	1/2/2019
					TargetB	4	1/3/2019
					TargetB	...	...
					TargetC	20	1/1/2019
					TargetC	18	1/2/2019
					TargetC	15	1/3/2019
					TargetC	...	...

On the left is what a single target machine learning data set looks like. On the right, is the data format that a multi-series model expects.

Single Targets vs. Grouped Targets

Single targets run with single series models and grouped targets run with multi-series models. Each approach comes with pros and cons depending on the data and the business problem.

With grouped targets, a multi-series model can establish relationships between the targets that are being grouped. If the targets are highly correlated and exhibit similar data patterns, it is likely that this can lead to better target accuracy than if the targets were only treated as single targets with single series models. However, if the targets are not correlated or exhibit little to no common historical data patterns, it is equally likely that the target accuracy would be worse than if the targets were only treated as single target with single series models.

Additionally, with grouped targets and multi-series models, it is more difficult for XperiFlow to choose features that will provide benefit to all targets involved in the group. This is because there are limits on the total number of features that can be used for a given model. This can potentially lead to important features that would typically only benefit a single or few targets that a part of a group from being included in the multi-series model. Therefore, it is possible to see target accuracy suffer for certain targets in a group.

Group Targets

There are circumstances where it may be beneficial to group targets.

Highly Correlated Targets

Grouping targets that are highly correlated or related can lead to better individual target accuracy. This happens because the multi-series model can establish non-linear relationships between the different target values.

For example, consider two targets, Dinner Sales and Alcohol Sales, for a restaurant where we want to forecast the daily sales for the next 14 days. It is likely that people will order alcoholic beverages around dinner time. As a result, it will likely be beneficial to group these two targets and allow the multi-series model to establish relationships between Dinner Sales and Alcohol Sales to positively influence the accuracy of both targets.

New Targets with Little Historical Data

Grouping targets that have little historical data with well established targets that have a healthy amount of historical data can yield better accuracy for those targets.

For example, a new suite of high-top sneakers is introduced to the market by a shoe company and have only been sold for the past six months. The shoe company has never sold high-top sneakers before but has been selling a suite of standard sneakers for over four years. For this shoe company, it may be beneficial to group the high-top sneaker targets with the standard sneaker products. This is because the high-top sneaker may exhibit similar sales patterns to the standard sneakers, therefore, a multi-series model may be able to establish relationships between the high-top sneakers and the standard sneaker, allowing the high-top sneakers to rely on the standard sneakers' historical patterns as its own. If grouping was not used in this scenario, then the new high-top sneakers would only have six months of historical data to try and establish a meaningful forecast.

Understanding Accuracy

Interpreting model accuracy can be difficult in data science, since there are many ways to quantify accuracy. SensibleAI Forecast uses multiple viewpoints of model performance to provide a complete picture.

Accuracy Degradation Over Time

It is normal and expected that model accuracy degrades over time. This is a symptom of the underlying data patterns changing since the model was initially deployed.

The Importance of Model Refits

Model Refit: Taking a deployed model and refitting the same model configuration on the latest data.

SensibleAI Forecast attempts to automatically protect against model accuracy degradation by giving the option to **Refit with Latest Data** before running its next prediction. This functionality defaults to **True**, and the prediction takes longer because every model is refit. This functionality refits the production model with the latest refreshed source data. Often, this functionality is enough to keep a model healthy. In some cases, this may even increase model accuracy over time leading to a positive health score.

OneStream recommends leaving **Refit with Latest Data** set to **True** unless the project needs quick and frequent predictions.

The Importance of Model Rebuilds

Rebuild: A rebuild consists of creating new models, data sources, and configurations for a set of targets.

The underlying data patterns are expected to change over the course of time which will cause model accuracy degradation. This implies that:

- Some of the features that were once important may no longer be useful.
- There may be new events that can positively influence accuracy.
- A different intelligent model may better represent the data.

The Purpose of a Partial Rebuild

In large projects, there may be some targets that degrade to unacceptable levels faster than others. Instead of rebuilding all targets and spending time retraining, the partial rebuild option allows you to rebuild only the poorly performing targets.

Know When to Rebuild

SensibleAI Forecast provides indicators throughout the Model Utilization section.

Model Health Distribution Chart: The chart is available on the **Manage Health** page and the **Analysis Overview** page. It plots and color codes the health score for all targets; green = healthy, yellow = warning, and red = unhealthy. If a cluster of red is found with mostly green, this may be an indicator to execute a Partial Rebuild to get the red targets back into the green. If there is mostly yellow and red, this may be an indicator to execute a Full Rebuild.

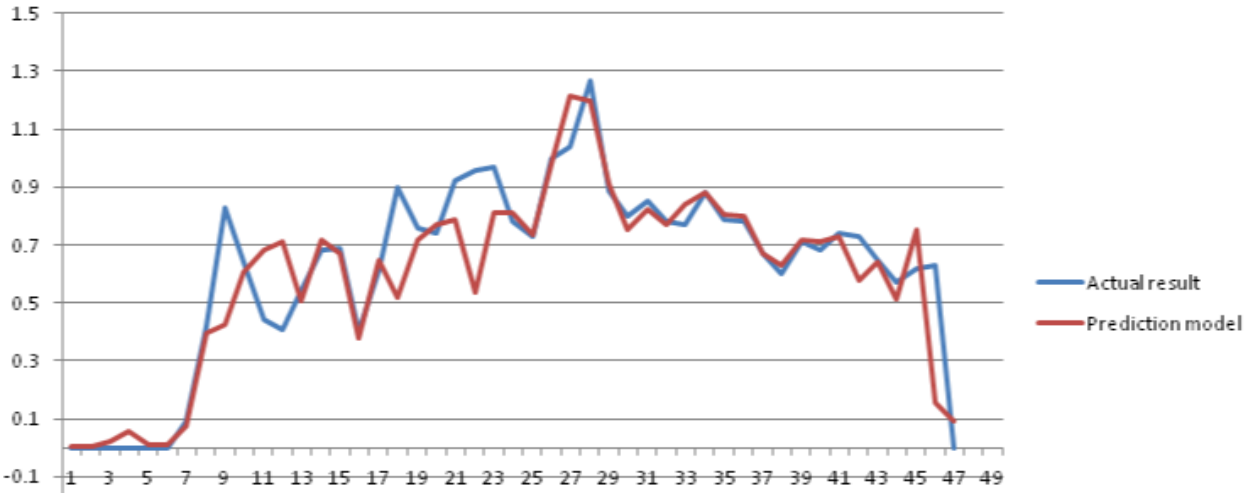
XperiFlow Suggestions: A message grid view that lets you know if a rebuild is suggested.

Quantifying Model Accuracy

Error Metrics

Error metrics evaluate a model's accuracy and quantify how far off predictions are from actuals. All error metrics typically work with two inputs: actual data and associated model predictions that line up against those actuals. An arbitrary error metric output is a single number that can be compared relative to predictions made on the same data.

The following graphic shows an arbitrary set of actuals and predictions:



All error metrics work by quantifying the difference between a set of predictions and actuals. The further away a prediction from its associated actual, the higher the respective error is for that given data point pair.

Variations in quantifying this difference between actuals and predictions produce several different types of error metrics.

For example, Mean Squared Error (MSE) works by squaring the absolute difference between each actual-prediction pair while Mean Absolute Error (MAE) takes the absolute difference between each actual-prediction pair.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$$

A smaller number for most error metrics is better than a higher number (minimizing error). However, there are certain error metrics such as R2 Score that look to maximize its value to **1**. R2 Score is the variance proportion in the dependent variable that can be predicted from independent variables and is closely related to Mean Square Error (MSE).

Evaluation Metrics

An evaluation metric is the error metric that evaluates model accuracy.

You can specify the evaluation metric on the **Modeling Configuration** page of the Model Build section. You can select the evaluation metric or let SensibleAI Forecast set the evaluation metric to what it thinks is the best metric to use for the modeling task.

The selected evaluation metric is used to:

- Choose the best model parameters for a model on a target.
- Compare accuracy across models for a target.
- Select the best performing model for a target in training that is then put into production.
- Compare statistical or machine learning models to baseline models.
- Calculate the health scores. The health score is a mathematical equation that quantifies how evaluation metric changes over time.

Comparisons to Baseline Models

SensibleAI Forecast compares statistical or machine learning models to baseline models to quantify the benefit of using SensibleAI Forecast versus traditional forecasting methodologies. This comparison provides an understanding of the benefit of using intelligent models over more traditional forecasting methodologies.

SensibleAI Forecast creates these comparisons by comparing the evaluation metric of an intelligent model and the evaluation metric of a baseline model. The **Pipeline Deploy** page of the Model Build section shows the comparison between baselines and machine learning models in different ways. This provides a quality assurance check that intelligent models can learn more than traditional baseline forecasting methodologies for this forecasting problem.

When Baseline Models Outperform Machine Learning Models

You may experience a situation where baseline models, like shift and mean models, outperform their machine learning counterparts because:

- There is not enough meaningful training data that the machine learning models can learn. Providing machine learning models with more features and data instances to learn from (seasonal, event, weather) typically leads to improved predictive capability. If the time on a machine learning model is trained is too short, the model could perform poorly. If making the training time frame longer is not possible, consider adding additional features and events that could help capture some important data relationships.
- The level of target dimension aggregation selected within the data may not be ideal. Depending on the data set, sparse data can originate from creating too deep, or specific, a target dimension. Typically, it is difficult for machine learning or statistical models to learn anything meaningful from a highly sparse target (many values are 0).

  Rolling up to a high-level, or generic, target dimension could hide underlying trends and insights. In the case of a generic target dimension, it is best to use dollar sales over unit sales to ensure that quantified products and services sold are weighted by their dollar amount. This can make it easier for the models to understand value. Intuitively, this makes sense when you consider a situation where the selected target dimension and value are too generic, such as unit sales of clothing.

In this scenario, unit sales could be dominated by high-quantity products with low-value sales compared to low-quantity products with high-value dollar amounts (for example, socks versus Cashmere sweaters). Given unit sales with an overly generic target dimension, a model will struggle to learn seasonality and trend. Keep in mind that this case will align more closely to a sales planning use case than a supply chain use case.

The following provides a conceptual example of target dimension aggregation level. This is not the same across all data sets. Methods for target dimension aggregation should always be assessed to ensure optimized machine learning model learning capabilities.

NOTE: OneStream recommends that you create small project experiments to test varying levels of target dimension aggregation, different amounts of date instances, and different events and features on a small subset of the entire data set. This allows you to determine the best data set format for your project. Do this experimentation until you have Statistical and Machine Learning Models winning consistently over baselines for the most important 60-80% of targets.

XperiFlow Health Score

The XperiFlow Health Score indicates how a deployed model's performance changes over time. The health score range is between -1 and 1. A health score value of zero means that the model performance has remained constant while the model has been in production. A negative health score implies that the model performance has degraded while the model has been in production. A positive health score implies that the model performance has improved since it has been deployed.

The health score is a calculated, weighted rate of change of the evaluation metric at discrete time intervals over the course of the model being in production.

Other Model Performance Considerations

Fundamental Changes to Business Over Time

In almost every data science problem, *the more data, the better*. Time series forecasting is one of the few data science problems that has an exception to that statement. To properly adjust this statement for time series, it should be phrased as: *the more data, the better… as long as the data patterns remain largely consistent from the beginning of the data history to now*.

This means that if the data patterns change wildly over the course of time, the old and outdated data patterns have little to no importance to the data patterns that exist and need to be forecasted now. From a model point of view, blending old and outdated data patterns with new patterns may only lead to confuse the models and lead to bad performance. Examples of this include data sets or businesses that have been heavily affected by the COVID-19 pandemic in 2020, and completely changes the underlying sales patterns and consumer behavior to look nothing like pre-COVID-19.

Long Forecast Ranges Using Daily Data

Long forecast ranges will typically yield lower performance to shorter forecast ranges. This is because there will be more meaningful features able to aid in the prediction when the forecast range is shorter.

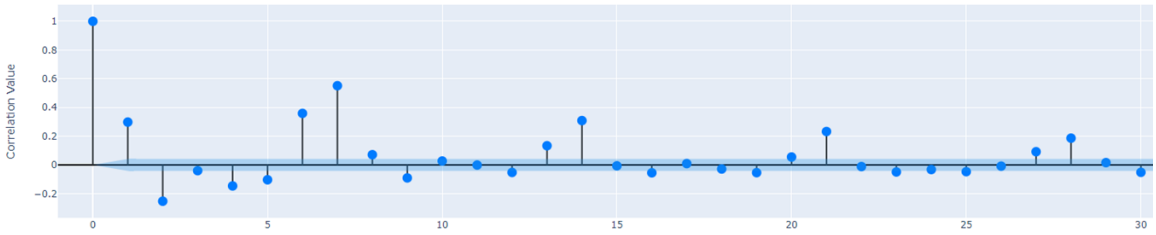
Explaining with Lags

In time series forecasting, a target variable is usually heavily correlated with its most recent prior values.

In a daily granularity restaurant sales forecasting example, the sales value today will be heavily correlated with the sales value from 7 days ago. This is because a strong weekly seasonal data pattern found in the data in a daily level data set, today's (Monday's) sales value will be correlated with prior values.

The following graphic shows the daily sales data. There are routine spikes near the weekend. This is weekly seasonality which makes intuitive sense given that this is a restaurant bar that will experience higher sales on the weekend.

The following auto-correlation plots how correlated each lag of the target variable is with the actual target variable. The 6th and 7th day target lags are highly correlated (.35, .55 respectively) with the target. Additionally, as you move further out, 14, 21, and 28 day target lags are the next highly correlated with the target. The key takeaway is that the further away you get from the actual day (lag days), the less correlated that value is. The smaller the lag number, the more likely the correlation value is high, the more important the lag is an aiding in the prediction of the target.



What do lags have to do with a Long Forecast Range?

In time series, lags of the target variable are powerful features that ultimately lead to a big performance boost. Like the correlation plot shows, a lag feature will typically become less impactful to the model performance the further away you get from the current day.

In time series forecasting, you are forced to lag any features values that you do not know in advance by the length of the forecast range. That means if we want to forecast 21 days ahead on this data set, we can only leverage lags greater than 21 days. Lags of 1, 6, 7, 13, 14, and 20 days will not be able to be used as features in this case. If we were to attempt to produce a 21-day forecast with the previous lags, all of those lags would be missing data necessary to make a prediction.

To summarize:

- The smaller the lag number, the more likely the correlation value is high, the more important the lag is when aiding in the prediction of the target.
- You are forced to lag any features values that you do not know in advance by the length of the forecast range.

Having a large forecast range leads to having less impactful lag features that give lower model performance compared to a short forecast range.

> **NOTE:** Features that are known in advance are features such as events since they happen on a repeated basis. There is no need to lag events.

Appendix 2: Use Case Example

A grocery store company has ten store locations across Florida. This parent company wants to accurately forecast daily sales for 100 different products such as bread, hamburgers, and soda at each location. Provided with daily historical sales over the last three years, SensibleAI Forecast generates hundreds to thousands of performance-enhancing features to train and select the most accurate forecasting model possible to optimize downstream business processes.

There are four main feature types recognized by SensibleAI Forecast. The feature types are largely categorized by how they are created or gathered.

Common Definitions

Time Series Forecasting: *Time series forecasting* uses a model to predict future values based on previously observed values. A simple example of this predicting future grocery store hamburger sales based on historical sales over time.

Other features (see below) could be incorporated to improve the predictive capability of a model.

SensibleAI Forecast Project: A SensibleAI Forecast project is a collection of targets, data sources, and model configurations.

Targets and Features: A *target* is a subject that is to be forecasted and is represented by a single series of historical data (such as beer, martini, and sandwich sales).

A *feature* is a measurable characteristic of a phenomenon that can be represented as a data series that can be leveraged to improve the predictive accuracy of a model. Multivariate models can learn complex non-linear relationships between a target variable and multiple feature variables.

In this example, machine and statistical models produce forecasts for each target such as beer, martini, and sandwiches. Models that produce the forecasts can use feature variables such as New Year's or Gas Price to enhance forecast accuracy.

Model Training: In machine learning context, model training is the process of providing machine learning algorithms with data used to learn trends and patterns.

Iterations (Machine Learning): In machine learning, iterations are the number of times an algorithm's parameters are updated. Iterations are done to achieve optimal algorithm performance.

Hyperparameters:

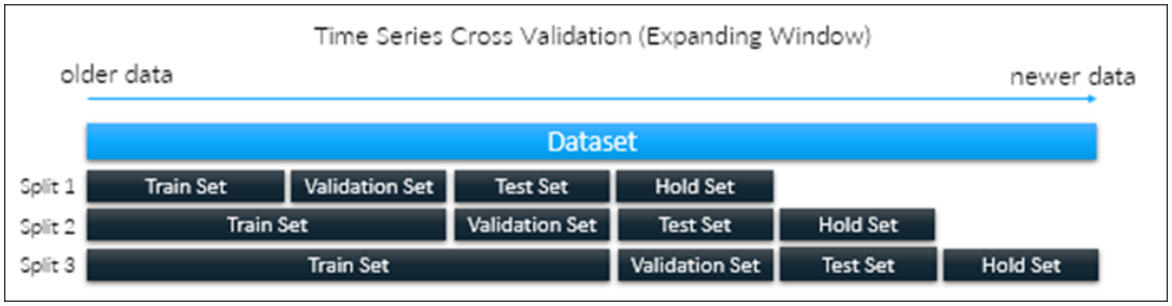
- Cannot be learned from the data.
- Are tunable (hyperparameter tuning). Different hyperparameter tunings result in different model parameter optimizations. This leads to different levels of accuracy.
- Directly control the behavior of the training algorithm.
- Have significant impact on the performance of the model being trained.
- Express high-level structural settings for algorithms.

Cross-Validation: A technique that tests statistical or machine learning model performance prior to putting the model into production. Cross-validation works by reserving specific data set samples on which the model is not trained. The model makes predictions on these untrained samples to evaluate its accuracy. Cross-validation provides understanding of how well a model performs before putting the model into production.

Cross-validation is used to :

- Choose the best hyperparameters (the best variation) of a model.
- Compare different models to determine the best model to deploy.

A cross-validation for a time series uses methods such as walk-forward cross-validation (or sliding-window) and expanding-window cross-validation (or forward-chaining). The following images illustrate these methods.



Meta-Learning: Machine learning algorithms that learn from the output of other machine learning algorithms. Rather than increasing the performance of a single model, several complementary models can be combined to increase model performance.

Ensembles: A meta-learning approach that uses the principle of creating a varied team of experts. Ensemble methods are based on the idea that, by combining multiple weaker learners, a stronger learner is created.

Boosting: An ensemble technique that sequentially boosts the performance of weak learners to construct a stronger algorithmic ensemble as a linear combination of simple weak algorithms. Each weak learner in the sequence tries to improve or correct mistakes made by the previous learner. At each iteration of the Boosting process:

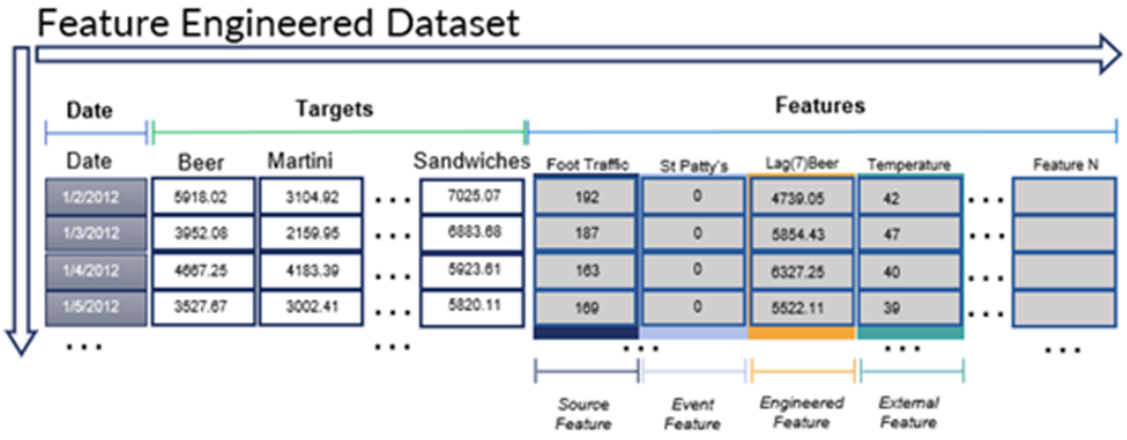
- Re-sampled data sets are constructed specifically to generate complementary learners.
- Each learner's vote is weighted based on its past performance and errors.

Some of the most popular boosting techniques include:

- AdaBoost (Adaptive Boosting)
- Gradient Boosting
- XGBoost (Extreme Gradient Boosting)

Feature Types

There are four main feature types recognized by Sensible ML. Feature types are largely categorized by how they are created or gathered.



Source Feature: A data column in the source data set that enters SensibleAI Forecast. Specify source features in the **Data Features** page of SensibleAI Forecast.

You can create additional features if you have already collected features outside of SensibleAI Forecast and you know these features help the modeling process. However, creating additional features in SensibleAI Forecast is not necessary and not heavily used since so much of SensibleAI Forecast is geared towards generating external features, event features, and engineered features for you. Source features are largely considered a Data Science function of Sensible Machine Learning.

External Feature: A data column that is grafted onto the source data and not originally included in the source data. External features are largely found through external data APIs such as weather or macro-economic data. The most common ways to graft an external feature to the source data is by leveraging the Location capability in SensibleAI Forecast that fetches features such as weather data.

Event Feature: A special subset of an external feature generated using the Event Builder in SensibleAI Forecast. An event feature is always a binary categorical data column generated from the events included in the SensibleAI Forecast Model Build phase.

Event Features have a profound positive effect on model performance.

Engineered Feature: A data column built by augmenting and transforming existing columns in the source data as well as other previously engineered features. Examples of this transformation process include cleaning missing values, lagging, moving averages, time breakdowns, and scaling values in an existing feature to create new engineered features.

Typically, a huge part of a data scientist's time goes toward manufacturing engineered features to improve overall model performance. SensibleAI Forecast creates engineered features by autonomously running statistical column transformations against each column in the source data set. This genetic algorithm style creation of engineered features leads to creating thousands of unique engineered features that SensibleAI Forecast can choose from and leverage in the modeling process.

Model Types

Univariate (statistical) models: A model that does not accept external variables to make predictions. Univariate models are used when there are not many data patterns to learn from, such as with one or two seasons.

Multivariate (machine learning) model: A model that leverages external variables in its predictions. These models are used when there are many data patterns to learn from, such as multiple seasons, anomalies, and a trend.

Multi-series model: A type of multivariate model that is trained to make predictions on a group of targets. A multi-series model requires data to be pivoted in a special long-form data format.

Single series model: A model that is trained to make predictions on a single target. A single series model can be a multivariate model or univariate model.

Baseline model: A type of naïve model that emulates what traditional forecasting methodologies may look like. A baseline model acts as an initial comparison benchmark for more intelligent models.

Appendix 3: Error Metrics

> **NOTE:** Only some of these Error Metrics can be used as evaluation metrics. The rest are computed metrics (not used for evaluating which model is better than another).

Mean Absolute Error (MAE)

An error measurement between paired observations expressing the same phenomenon. Examples of **Y** versus **X** include predicted versus observed comparisons, subsequent time versus initial time, and one measurement technique versus an alternative measurement technique.

Formula:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| y_{actual} - \hat{y}_{forecast} \right|$$

Interpretation: Lower is better.

Benefits:

- Easily interpretable.
- No favoritism towards over- or under-predictions.

Shortcomings:

- Relative size of the error is not obvious as with percentages.

Mean Absolute Percent Error (MAPE)

A prediction accuracy measurement of a forecasting method. Also known as mean absolute percentage deviation.

Formula:

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

where A_t is the actual value and F_t is the forecast value. Their difference is divided by the actual value A_t.
The absolute value in this ratio is summed for every forecasted point in time and divided by the number of fitted points n

Interpretation: Lower is better.

Benefits:

- Easily explainable.
- Scale-independent / expressed as a percentage.

Shortcomings:

- Returns undefined values when forecasting for actual values of zero.
- Favors models that under-forecast due to heavier penalty on forecasts higher than actuals. Forecasts below actuals cannot be worse than 100% Mean Absolute Percent Error (MAPE).

Mean Absolute Scaled Error (MASE)

A measure that determines how effective forecasts generated through an algorithm are by comparing the forecast predictions with the output of a naïve forecasting approach.

Formula:

$$MASE = \frac{MAE}{\left(1/n-1\right)\sum_{i=2}^{n}\left|a_i - a_{i-1}\right|}$$

Interpretation: Lower is better

Benefit: Independent of the data's scale, so it can be used to compare forecasts across data sets with different scales.

Mean Asymmetric Over Error (MAOE)

A combination of both Mean Squared Error (MSE) and Mean Absolute Error (MAE) depending on whether the predicted value is over or under the actual value. If the predicted value is over the actual value, then the error metric applied is MSE (squared difference). If the predicted value is under the actual value, then the error metric applied is MAE (absolute difference).

Formula:

$$MAOE = \frac{1}{n}\sum_{t=1}^{n}\begin{cases}(A_t - F_t)^2 \text{ for } A_t \leq F_t \\ |A_t - F_t| \text{ for } A_t > F_t\end{cases}$$

Interpretation: Lower is better.

Benefit: Favors models that under predict. May be useful if there is a difference in penalty in real world application for over predictions.

Shortcomings: Can over penalize a model for over predicting just once or twice way more than under predicting consistently.

Mean Asymmetric Under Error (MAUE)

Similar to Mean Asymmetric Over Error (MAOE), but applies Mean Squared Error (MSE) to predictions below the actual value and Mean Absolute Error (MAE) to predictions above the actual value.

Formula:

$$MAUE = \frac{1}{n}\sum_{t=1}^{n}\begin{cases}|A_t - F_t| \text{ for } A_t \leq F_t \\ (A_t - F_t)^2 \text{ for } A_t > F_t\end{cases}$$

Interpretation: Lower is better.

Benefit: Favors models that over predict. May be useful if there is a penalty differences in real world application for under predictions.

Shortcomings: Can over penalize a model for under predicting just once or twice way more than over predicting consistently.

Mean Bias Error (MBE)

Mean Bias Error (MBE) is primarily used to estimate the average bias in a model and determine what is needed to correct the model bias. MBE captures the average bias in the prediction.

Formula:

$$MBE = \frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)$$

where Oi is the observation value and Pi is the forecast value.

Interpretation: Typically not used as a measure of model error, as high individual errors in prediction can also produce a low MBE. MBE reflects a prediction's average bias. A positive value represents overestimating bias and a negative value represents an underestimating bias.

Shortcomings: This is typically not used to measure model error.

Mean Percent Error (MPE)

The computed average of percentage errors by which a model's forecasts differ from actual values of the forecasted quantity.

Formula:

Interpretation: Closer to zero is better.

Benefits: Can be used as a measure of the bias in the forecasts

Shortcomings:

- Measure is undefined when a single actual value is zero.
- Positive and negative forecast errors can offset each other.

Mean Squared Error (MSE)

The average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where N is the number of data points,
f_i the value returned by the model and
y_i the actual value for data point i.

Interpretation: Lower is better.

Benefits:

- Increased penalty for larger errors.
- Ensures positives values for easy interpretation.

Shortcomings:

- Less intuitive than MAE.

Mean Squared Logarithmic Error (MSLE)

A measure of the ratio between the true and predicted values. MSLE is a variation of Mean Squared Error (MSE).

Formula:

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(Y_i) - \log(\hat{Y}_i))^2$$

Interpretation: The loss is the mean over the seen data of the squared differences between the log-transformed true and predicted values.

Benefit: Treats small differences between small true and predicted values approximately the same as large differences between large true and predicted values.

Shortcomings: Penalizes underestimates more than overestimates.

Median Absolute Error (MedAE)

The loss is calculated by taking the median of all absolute differences between the target and the prediction. If $\hat{y}$ is the predicted value of the sample and y_1 is the corresponding true value, then the median absolute error estimated over *n* samples is defined as follows:

Formula:

$$MedAE\left(y,\hat{y}\right)=median\left(\left|y_1-\hat{y}_1\right|,\ldots,\left|y_n-\hat{y}_n\right|\right)$$

Interpretation: Lower is better

Benefit: Robust to outliers

Symmetric Mean Absolute Percent Error (SMAPE)

An accuracy measure based on percentage (or relative) errors.

Formula:

$$\text{SMAPE}=\frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t-A_t|}{(|A_t|+|F_t|)/2}$$

where A_t is the actual value and F_t is the forecast value.

The absolute difference between A_t and F_t is divided by half the sum of absolute values of the actual value A_t and the forecast value F_t. The value of this calculation is summed for every fitted point *t* and divided again by the number of fitted points *n*.

Interpretation: Lower is better

Benefits:

- Expressed as a percentage.
- Lower and upper bounds (0% - 200%).

Shortcomings:

- Unstable when both the true value and the forecast value are close to zero.
- Not as intuitive as MAPE (Mean Absolute Percent Error).
- Can return a negative value.

R2 (R-Squared)

A statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variables in a regression model.

Formula:

X	Y	X^2	Y^2	XY
4	5	16	25	20
8	10	64	100	80
12	15	144	225	180
16	20	256	400	320
$\sum X = 40$	$\sum Y = 50$	$\sum X^2 = 480$	$\sum Y^2 = 750$	$\sum XY = 600$

Now,

$$R^2 = \frac{N \times \sum XY - (\sum X \sum Y)}{\sqrt{[N \sum x^2 - (\sum x)^2][N \sum y^2 - (\sum y)^2]}}$$

Putting all the values,

$$R^2 = \frac{4 \times 600 - (40 \times 50)}{\sqrt{[4 \times 480 - (40)^2][4 \times 750 - (50)^2]}}$$

Solving we get

$$R^2 = \frac{400}{17.89 \times 22.36}$$

$$= \frac{400}{400}$$

$$= 1$$

Therefore correlation coefficient is 1.

Interpretation: Higher is usually better; Measures the strength of the relationship between independent and dependent variables in a regression model; Ranges between 0 (No Correlation 0 and 1 (Perfect Correlation)

Benefits: Commonly used as a measurement technique

Shortcomings:

- Sometimes, a high R-Squared value can indicate problems with the regression model.
- Does not reveal the causation relationship between the independent and dependent variables.

Symmetric Mean Absolute Percent Error (SMAPE)

A measure to compare true observed response with predicted response in regression tasks.

Formula:

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

where A_t is the actual value and F_t is the forecast value.

Interpretation: The value of this calculation is summed for every fitted point *t* and divided again by the number of fitted points *n*.

Benefits: Has both a lower bound and an upper bound.

Shortcomings: If the actual value or forecast value is 0, the value of error goes to the error upper limit.

Appendix 4: Interpretability

Feature Effects

For a specific trained model, a specific feature of that model and its value, the feature effects shows the average prediction of that model versus the average value of the actuals, The average is taken across all dates where that feature was equal to the specific feature value. Use this information to see how accurate a model is on average for a specific feature. Used with the feature impact score of the feature, this is a useful diagnostic for decomposing the model performance and identifying potential areas for improvement. This could mean adding different impactful features so a model is less reliant on a feature that shows large deviances between average predictions and average actuals, or removing a poorly performing feature.

Take for example, given model A, a daily data set between 1/1/2020 and 1/1/2021, binary feature B that takes on values 0 and 1 and is used by model A:

- The feature effects graph for feature B would have on the X axis 0 and 1.
- The Y axis over 0 would be the average value of the actuals on all days the feature took on 0, as well as the average prediction on all days the feature took on value 0. The same applies for a feature value of 1.

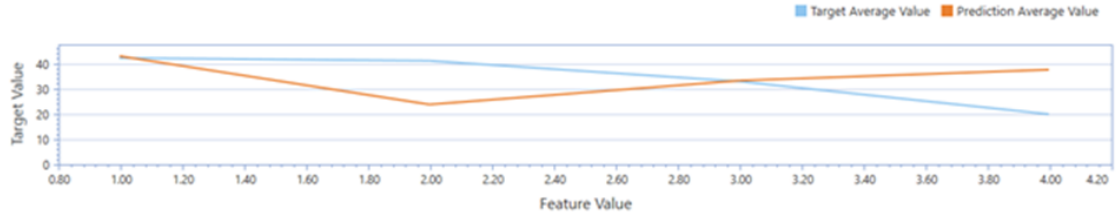
The following example looks at the feature effects of the Cubist Model for the feature Date-Quarter and a specific target. The average prediction on all the days the Date-Quarter takes on a value of 2 is roughly 20, and the average actuals is roughly 40. This same logic can be applied to all the feature values Date-Quarter takes on, namely, 1, 2, 3, and 4.

ModelName	Feature Impact
CatBoostModelRegressionTimeSeries(forecast=Range Step: 7 Collection Lag Step: 0)	False
CubistModelRegressionTimeSeries(forecast=Range Step: 7 Collection Lag Step: 0)	False
MeanModelRegressionTimeSeries(forecast=Range Step: 7 Collection Lag Step: 0)	False
PolyElasticNetModel(forecast=Range Step: 7 Collection Lag Step: 0)	False
ShiftModel(forecast=Range Step: 7 Collection Lag Step: 0;shift=10)	False
ShiftModel(forecast=Range Step: 7 Collection Lag Step: 0;shift=14)	False

Features

FeatureShortName	Feature Impact
Date-Day	36.00
Date-Month	12.50
Date-WeekOfYear	5.50
Date-DayOfWeek	2.00
Date-Quarter	0.00

Feature Effects



Feature Impact

Feature impact is a way to understand the overall importance a feature has to a model's predictions. The higher a feature's impact score, the more significant that feature is to the model's outputs.

For example, the following chart shows the Date-Year feature has the most significant impact on the model's predictions, where Date-Month was rather trivial in comparison. Specifically, changes in the Date-Year feature cause large changes in the model's output, whereas changes in the Date-Month feature cause much smaller changes in the model output.

Feature Impact	
FeatureShortName	FeatureImpact (Average)
Date-Year	21.91
Date-DayOfYear	12.36
Date-WeekOfYear	4.86
Date-Day	4.32
Date-DayOfWeek	2.37
Date-Quarter	2.29
Date-Month	1.28

There are various and evolving mathematical methods for calculating feature's impact on a model. The two methods implemented by the engine are Permutation Importance and Shapley Additive Explanations (SHAP).

The permutation importance of a feature is the decrease in the model's prediction quality when that feature is randomly shuffled. This removes the model's ability to use that feature, but not its reliance since the model is already trained on the unshuffled version of the feature. The performance decrease's magnitude is the feature's relative importance for the model. If shuffling the feature has little effect on the model's error, that feature is not important since the model did not rely heavily on the feature. If shuffling a feature causes a large performance decrease, the model relies heavily on that feature, meaning the feature is important.

The SHAP feature impact value indicates the average magnitude this feature moved the model's prediction from the model's average prediction across all data points. SHAP is an algorithm for calculating Shapley values, where the Shapley values are the feature's fair attribution contributions to the model output for a specific data point.

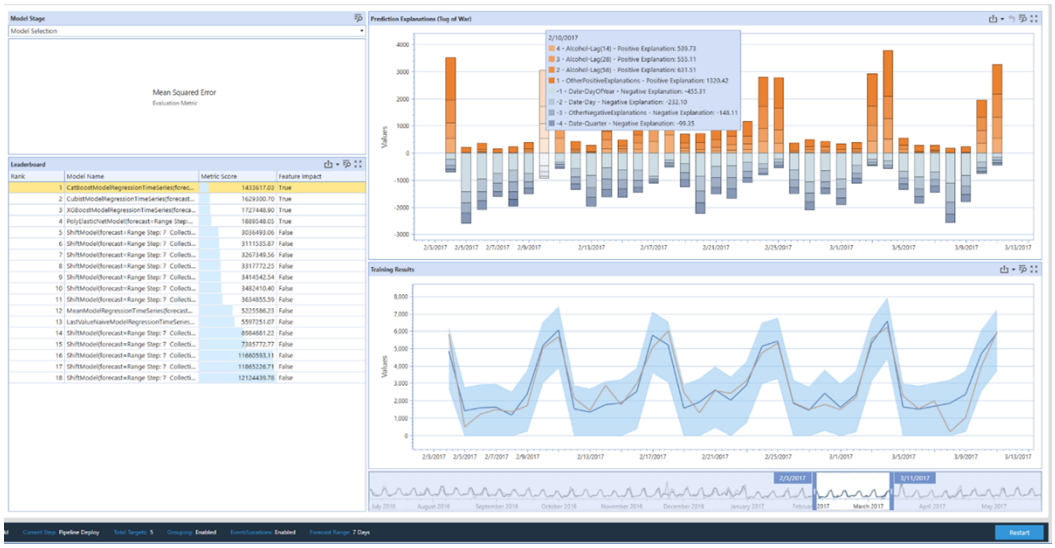
If a model's average prediction across all data points is x, the Shapley values of the features for a specific data point x' tells how the model got from its average prediction x to the specific prediction for x', where $x' = x + \text{sum}(\text{Shapley values})$ and the sum is taken across all features the model uses. The average magnitude of a specific feature's Shapley value for across all data points predicted can go from specific data point explanations to an overall feature impact.

For more information on the specific math behind the SHAP implementation, see: https://christophm.github.io/interpretable-ml-book/shap.html

Prediction Explanations

For a specific predicted data pot, prediction explanations show how the model uses the features to form that prediction. XperiFlow uses Shapley Additive Explanations (SHAP) as its prediction explanation method. SHAP for prediction explanations and for feature impact is the same algorithm. The only difference in in prediction explanations, the Shapley magnitude average values of the Shapley are not taken. Instead, the Shapley values at each data point are shown to explain how the model got from its average prediction to the actual prediction.

This can be seen in the following graphic. Zooming in on the first predicted point, the model's prediction is higher than average, and the Shapley values represent this. The orange boxes show positive Shapley values, driving the prediction upward from the average. The blue boxes show negative Shapley values, driving the prediction downwards. For the first data point, the positive Shapley values far outweigh the negative Shapley values, leading to a higher prediction. Each box is associated with a specific feature, and shows for the feature and data point how the feature affects the prediction output; either by driving it down, driving it up, or being insignificant. The sum of the Shapley values plus the average model prediction gives the model prediction for a specific data point.

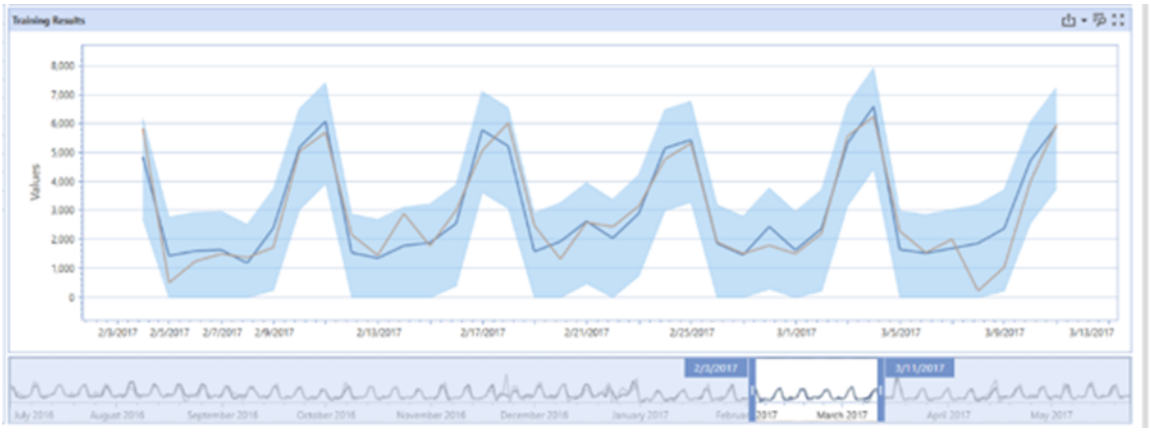


Prediction Intervals

A prediction interval is a value range for a data point prediction that likely contains the true actual future value of the data point. Prediction intervals are useful since time series data is a stochastic process, which means an underlying process generates the actual data has randomness involved. The prediction interval can help account for this randomness.

Prediction intervals instantiate using an alpha value. For example, an alpha of `.05` gives a 95% prediction interval. This means if the stochastic process generating the data is sampled an infinite number of times, you could expect 95% of the actual values to fall within this interval. These intervals move the models from simply predicting a point forecast, which can be interpreted as the mean of all the potential futures, to forecasting the distribution of the potential futures.

In the following example, the orange line is the point forecast and the shaded blue region around the orange line is the prediction interval.



There are various and evolving methods to calculate prediction intervals. The XperiFlow engine uses conformal prediction intervals, parametric prediction intervals, and non-parametric prediction intervals.

Parametric prediction intervals assume the error of the data follows a specific distribution (where the error is the difference between the actual value and the predicted value). The error metrics statistics are used to fit this distribution. Then, depending on the alpha level that instantiates the prediction interval, the proper values can be extracted from the distribution to surround the point forecast and form the prediction interval.

The non-parametric approach works similarly without assuming the error metrics. The non-parametric approach orders errors by size and uses the (1-alpha)% error as the top of the interval, and the (alpha)% error as the bottom. Specifically, if alpha is `.05`, the 95% percent largest error forms the top of the interval, and the 95% smallest error (which is either negative or zero) forms the bottom of the interval.