Oracle® FCCM Automated Scenario Calibration Cloud Service

Using Automated Scenario Calibration

Release 25.01.01
G25333-01
January 2025

ORACLE

Oracle FCCM Automated Scenario Calibration Cloud Service Using Automated Scenario Calibration, Release 25.01.01

G25333-01

Contents

Preface

Using Automated Scenario Calibration Cloud Service simplifies and automates the tuning and optimization of scenarios.

Audience

This document is intended for the Data Scientists and Users who are working in the Financial Crime and Compliance Management.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see these Oracle resources:

- Oracle Public Cloud: http://cloud.oracle.com
- Community: Use https://community.oracle.com/customerconnect/ to get information from experts at Oracle, the partner community, and other users.
- Training: Take courses on Oracle Cloud from https://education.oracle.com/oracle-cloud-learning-subscriptions.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: https://support.oracle.com/portal/.

1 Overview

Automated Scenario Calibration Cloud Service is designed to simplify and automate the tuning and optimization of Scenarios.

In this service, you can:

- Simulate events from any scenario for a chosen threshold set on demand.
- Use an industry standard methodology to determine thresholds for new scenarios based on historical data.
- Carry out Pre-prod analysis to determine initial thresholds for scenarios.

2 Getting Started

This topic provides step-by-step instructions to log in to the application.

Prerequisites

Before using Automated Scenario Calibration (ASC), you must configure the scenario pipelines to calculate “Overall Risk”. This is a key risk indicator that ASC uses to suggest adequate parameters depending on the “Overall Risk” of the customer.

To Add Overall Risk in pipelines:

1. In the Pipeline Designer, duplicate the Out of the Box pipeline, as described in the Copying Pipelines.
2. Open the newly created pipeline, then open the **Risk Indicator** widget.
3. Add the **Overall Ris**k Indicator, as described in Managing Risk Indicators.
4. Save the modified pipeline.

ASC is available with the following scenarios:

- Anomalies in ATM Bank Card - Foreign Transactions - Customer Focus
- Focal High Risk Entity - Customer Focus
- High Risk Counter Party - Customer Focus
- Large Reportable Transaction - Customer Focus
- Possible Currency Transaction Report - Customer Focus
- Rapid Loading and Redemption of Stored Value Cards - Customer Focus
- Rapid Movement of Funds - Customer Focus
- Single or Multiple Cash Transactions - Large Significant Transactions - Customer Focus
- Structuring - Avoidance of Reporting Thresholds - Customer Focus
- Structuring - Deposits Withdrawals of Mixed Monetary Instruments - Customer Focus
- Structuring - Potential Structuring in Cash and Equivalents - Customer Focus

Accessing Automated Scenario Calibration

To access the application:

1. Enter URL in the web browser.
2. The **Oracle Cloud** login page is displayed.
3. Enter your **User Name** and **Password**.
4. Click **Sign In**. The Application landing page is displayed.
5. In the **Navigation List**, click **Oracle Financial Services Crime and Compliance Management Anti Money Laundering Cloud Service** module. The menu options are displayed.
6. Click **Compliance Studio** and then click **Model Management and Governance**.

The Workspace Summary of the Automated Scenario Calibration landing page is displayed with pre-seeded workspace based on the logged in user.

2.1 User Profile

This topic provides information about functionality of users in the ASC.

Admin User (Graph Studio Administrator)

An Admin user is mapped with **GSADMIN** workspace to create a conda environment for executing the ASC notebooks.

To create the conda environment:

1. Login as Administrator. The Workspace Summary page is displayed with pre-seeded GSADMIN workspace.
2. Launch the **GSADMIN** workspace.
3. On the **Dashboard** page, click **Modeling** and select **Pipelines**.
4. Click **ASC** Objective and then click **Conda** to view the Conda Notebook.
5. Open the **Conda Notebook** and execute all the pre-seeded paragraphs in the notebook to create the conda environment.

Analyst User (ASC Analyst User)

An Analyst user is mapped with **ASC** workspace to automate and simplify the scenario tuning process. For Analyst action, see the following sections.

- Admin Notebook Activity
- User Notebook Activity

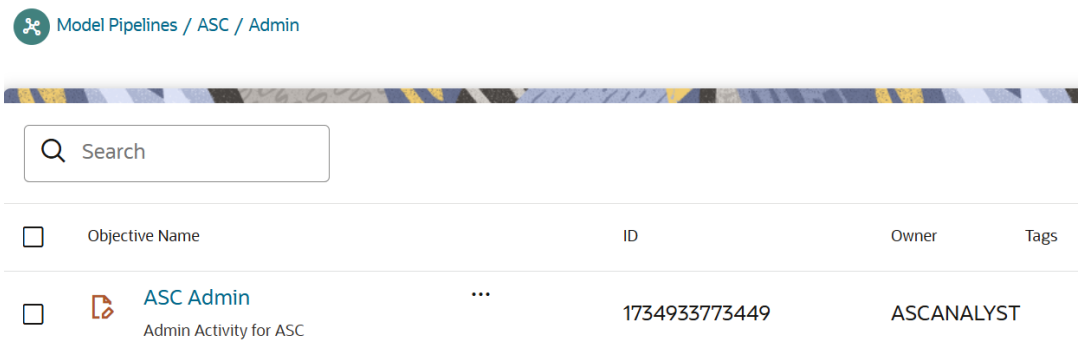
3
Admin Notebook Activity

The ASCAdmin Notebook provides import analysis template for PreProd scenarios.

To access the ASC Admin notebook:

1. Launch the **ASC** Workspace.
2. On the Dashboard page, click **Modeling** and select **Pipelines**.
3. Click **ASC** Objective and then click **Admin** to view the ASC Admin notebook.

Figure 3-1 ASC Admin Notebook



3.1 ASC Instance Initialization for Admin Notebook

Users can import the necessary packages and create an instance for the Admin Notebook.

Download/Activate Conda Environment

Navigate to the **Download/Activate Conda Environment** paragraph and run this paragraph to download/activate the Conda environment.

ASC Instance Initialization

To create an instance, execute the following.

```
%python-pgx

from ds_interpreter_client.context.ds_context import PyDataStudioContext
ds = PyDataStudioContext()

from ofs_auto_ml.model_deployment.output_tracking import *;
from ofs_auto_ml.db_connection import *
from ofs_aif.ofs_asc.asc import asc
asc = asc()
```

> **Note:**
>
> Some of the notebook's UI element like Drop-downs, Check boxes, and Text boxes might error out due to paragraph's synchronization issues.
>
> For such cases, the paragraph should be executed "twice or thrice" to get the successful selection of the paragraph and this is applicable for all the notebook execution part of ASC analysis.

3.2 Import Templates for ASC Analysis

This **asc.import_analysis_templates** provides a functionality to create a set of analysis templates that would require to execute PreProd scenarios.

Navigate to the **Configure Analysis and Import Templates** paragraph and run this paragraph to import the following analysis templates:

- **PreProd Scenario**
  - Scenario Execution
  - PreProd Scenario
  - Threshold

3.3 Create Scenario Definition and Import Templates

You can create and import the scenario definition for analysis.

To create the analysis definition:

1. Navigate to **Create Scenario Definition** paragraph.
2. From the **Scenario** drop-down list, select the scenario.
3. Enter the **Definition Name** (folder name in which templates will be imported).
4. From the **Analysis** drop-down list, select the analysis type. The available options are **Prod** and **PreProd**.
5. From the **Overwrite** drop-down list, select the option. The available options are:

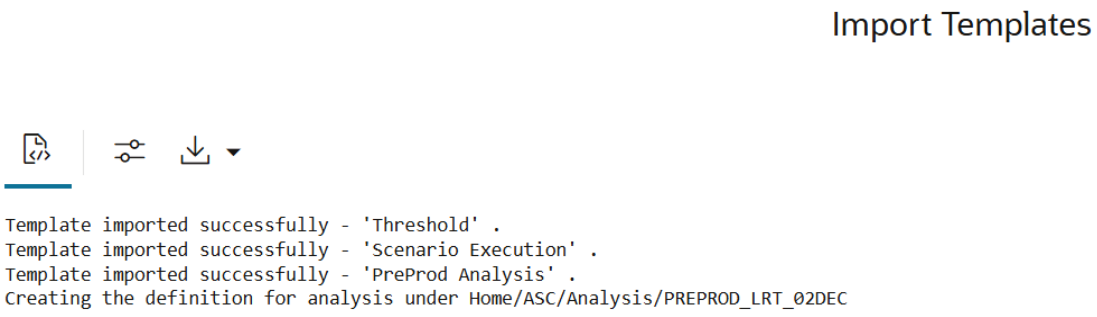
   True: Select when you want to delete and recreate the existing definition.

   False: Select when you want to create the definition.
6. **Run** the paragraph to create definition or recreate the existing definition.

Figure 3-2 Create Scenario Definition

Run the **Import Templates** paragraph to import templates based on the selected analysis and created the definition under specific path. The scenario definition path should be `Home/ASC/<Analysis>/<Definition Name>`.

Figure 3-3 Import Templates



4

User Notebook Activity

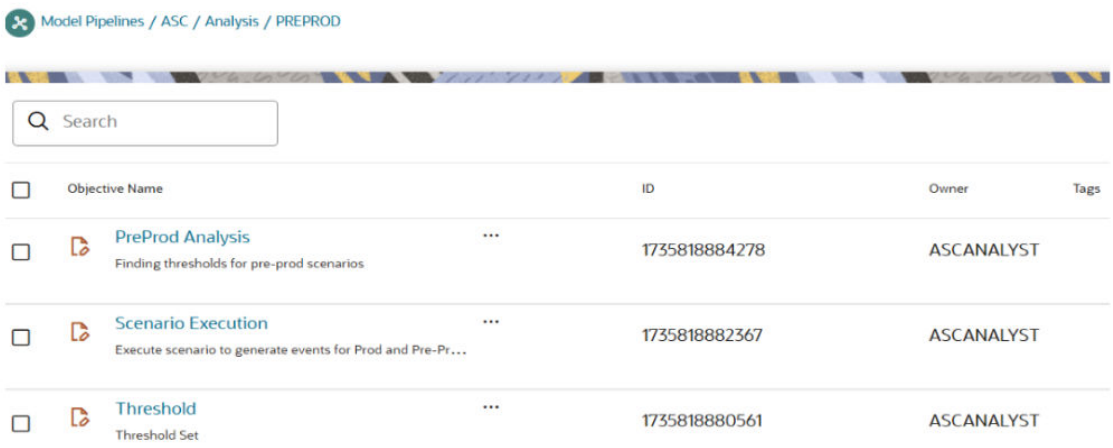
Users can navigate to the respective analysis folder, where they have the ability to define the scope and parameters of their specific scenario.

4.1 PreProd Scenario Analysis

Preprod Scenario Analysis is a crucial phase where preliminary assessments and preparations are conducted before actual production. It involves defining the scope and parameters of the scenario, collecting and preprocessing data, establishing assumptions and constraints, and performing initial modeling and simulations.

You can see the imported template for PreProd analysis as shown in the following figure.

Figure 4-1 PreProd Analysis



4.1.1 Threshold

Creating threshold set can assess scenario performance, identify areas for improvement, and ensure that the calibrated scenarios meet predefined standards and requirements.

4.1.1.1 ASC Instance Initialization for Threshold

You can import the necessary packages and create an instance for threshold set.

Navigate to the Threshold Notebook and the path is `Home / Modeling / Pipelines / ASC / Analysis / PreProd / Threshold Notebook`.

Download/Activate Conda Environment

Navigate to the **Download/Activate Conda Environment** paragraph and run this paragraph to download/activate the Conda environment.

ASC Instance Initialization

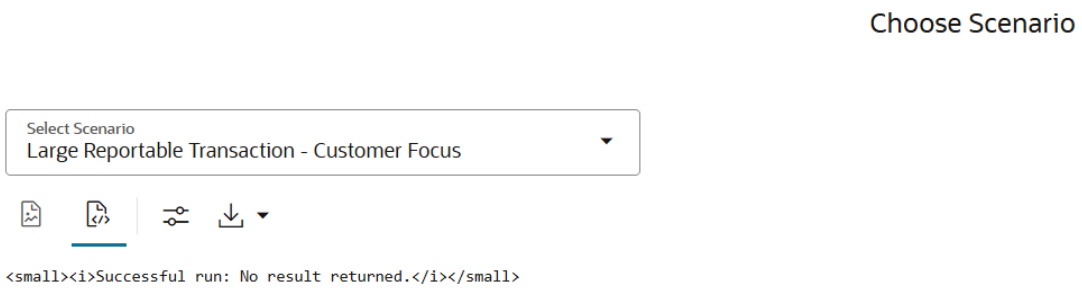
```
%python-pgx

from ds_interpreter_client.context.ds_context import PyDataStudioContext
ds = PyDataStudioContext()

from ofs_auto_ml.model_deployment.output_tracking import *;
from ofs_auto_ml.db_connection import *;
from ofs_aif.ofs_asc.asc import asc;
asc = asc()
```

4.1.1.2 Create Threshold Set

Users can create threshold set for scenario that will be used for further analysis.

You can select available scenario from the **Select Scenario**drop-down list.

Figure 4-2 Choose Scenario

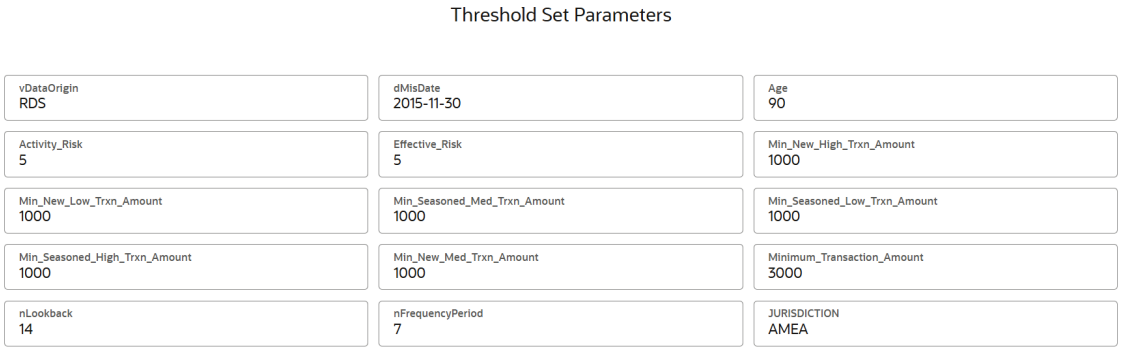


You can view the list of existing threshold set names in the **Existing Threshold Set Names** paragraph. Enter the Threshold Set Name in the **Input Threshold Set Name** paragraph and run it.

Figure 4-3 Input Threshold Set Name

You will see parameters for threshold set in the **Threshold Set Parameters** paragraph.

Figure 4-4 Threshold Set Parameters



If you want to update threshold set parameters, make the necessary changes in the required fields and then run the paragraph. This will generate updated threshold values, and you can view the updated threshold set parameters. This threshold set is used to run scenarios in the **Scenario Execution** notebook.

4.1.2 Scenario Execution

This topic describes how to generate tuning alerts by executing a scenario using specified thresholds for any time window.

The first step of the tuning process is to create tuning alerts by setting thresholds to some suitably low value. This generates a population of alerts below the current threshold, which can then be analyzed.

> **Note:**
>
> If you want to evaluate multiple thresholds sets for a given jurisdiction or group of jurisdictions, this should be done across different versions within the same definition. Within a single definition version, you should create only one threshold set per jurisdiction.

For example, if you want to evaluate threshold sets for jurisdictions AMEA, APAC and US, then you can create as mentioned in the following table. This assumes you want to apply the same thresholds for AMEA and APAC.

Table 4-1 Threshold Sets for Jurisdiction

Model Group Name	Definition ID	Version	Threshold Set ID	Thresholds
SIG CASH – 2024	123	1	1	{“Min_Trans_Amt”: 15000, “Min_Trans_Ct”:5 , “Incl_Jrsdcn_Cd: [”AMEA”,”JAPAC”]}

Table 4-1 (Cont.) Threshold Sets for Jurisdiction

Model Group Name	Definition ID	Version	Threshold Set ID	Thresholds
SIG CASH – 2024	123	1	2	{“Min_Trans_Amt”: 12000, “Min_Trans_Ct”: 3 , “Incl_Jrsdcn_Cd: ”US”}
SIG CASH – 2024	456	2	1	{“Min_Trans_Amt”: 13000, “Min_Trans_Ct”:2 , “Incl_Jrsdcn_Cd: [”AMEA”.”JAPAC”]}
SIG CASH – 2024	456	2	2	{“Min_Trans_Amt”: 12500, “Min_Trans_Ct”: 6 , “Incl_Jrsdcn_Cd: ”US”}

4.1.2.1 ASC Instance Initialization for Scenario Execution

You can import the necessary packages and create an instance for the scenario execution.

Navigate to the Scenario Execution Notebook and the path is `Home / Modeling / Pipelines / ASC / Analysis / PreProd / Scenario Notebook`.

Download/Activate Conda Environment

Navigate to the **Download/Activate Conda Environment** paragraph and run this paragraph to download/activate the Conda environment.

ASC Instance Initialization

To create an instance, execute the following.

```
%python-pgx

from ds_interpreter_client.context.ds_context import PyDataStudioContext
ds = PyDataStudioContext()

from ofs_auto_ml.model_deployment.output_tracking import *;
from ofs_auto_ml.db_connection import *;
from ofs_aif.ofs_asc.asc import asc;

asc = asc()
setattr(asc,'objectiveId',objectiveId)
```

Scenario

The notebook itself shows the scenario with which the analysis has been attached to.

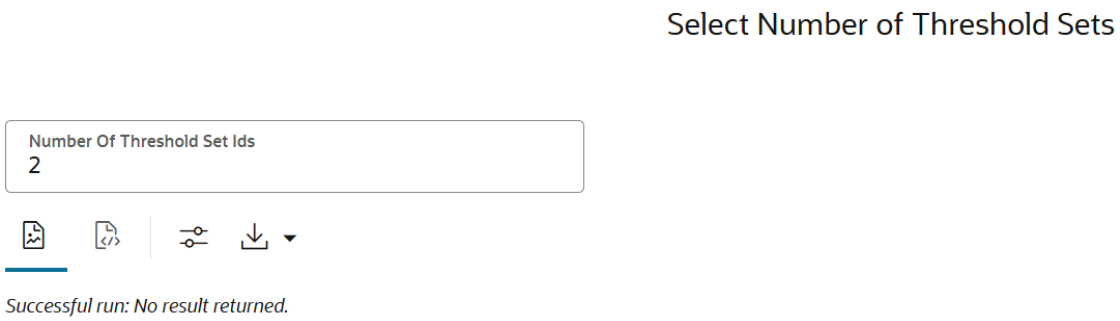
4.1.2.2 Specify Execution Parameters

The scenario can be executed for single or multiple thresholds for a given scenario. User must give the number of threshold set ids to be executed for a given scenario.

Select Number of Threshold Sets

Depending on the number of thresholds selected by the user, as many number of drop-downs will be appeared in next paragraph where different threshold set ids could be selected for the scenario.

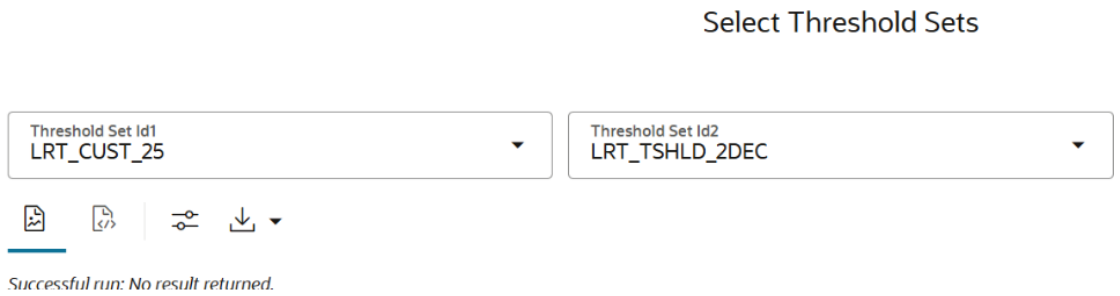
Figure 4-5 Select Number of Threshold Sets



Select Threshold Set IDs

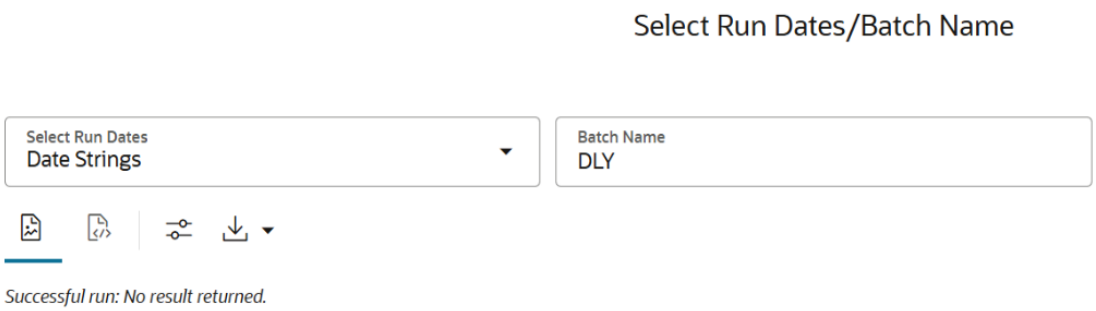
You can choose the base threshold set id or threshold set id which was generated through Threshold utility notebook depending on the requirement of the analysis.

Figure 4-6 Select Threshold Set IDs



Select Run Dates and Batch Name

Figure 4-7 Select Run Dates - Date Strings

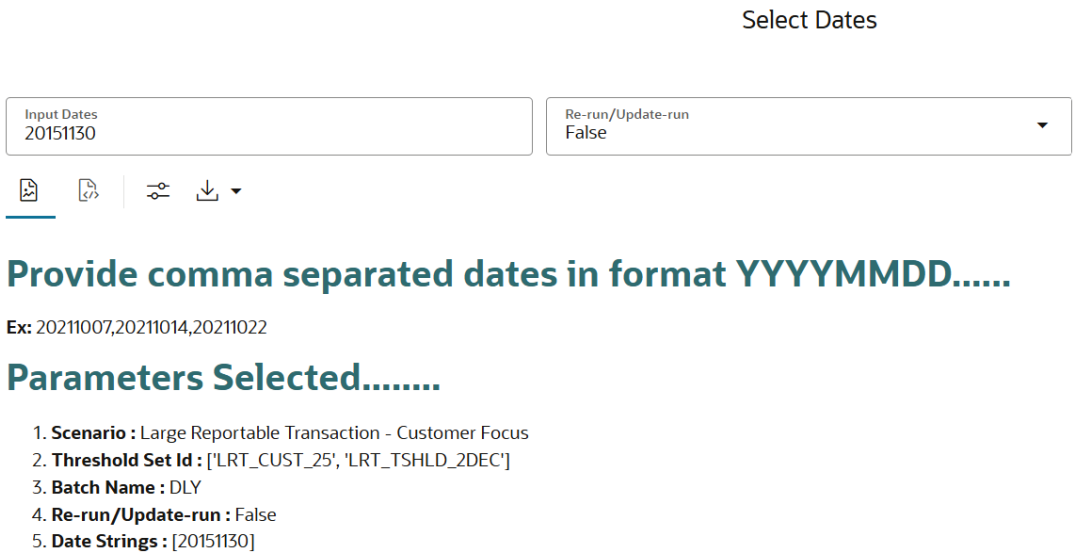


- When Select Run Dates is **Date Strings** the input text box takes the comma separated dates. Run dates when passed as a **comma separated**, then all possible combinations with threshold set ids will be created. For example: If run dates are passed as a comma separated **20211007**, **20211014**, **20211022** then threshold set ids are **116000048**, **118860048**, **78**. The possible combinations are:

```
{"20211007": [116000068, 118860048, 78],
    "20211014": [116000068, 118860048, 78],
    "20211022": [116000068, 118860048, 78]}
```

> **Note:**
>
> Make sure that you enclosed keys of dictionary into the double quotes.

Figure 4-8 Select Dates - Date Strings



The other option is selective **run date - threshold set id** combinations by passing **dictionary** as an input.

For example:

```
{"20211007": [116000068],
    "20211014": [118860048, 78],
    "20211022": [116000068, 118860048, 78]}
```

> **Note:**
>
> Make sure that you enclosed keys of dictionary into the double quotes.

- **Re-run/Update-run** should be **True** if you are re-executing the scenarios for **failed** run_dates/threshold_set_ids or **appending** any new run_dates/threshold_set_ids to existing version.
  - Re-run/Update-run is only applicable when Input Run Dates is selected as **Date Strings**.
  - The run_dates can be passed either **comma separated or dict of (run_dates,threshold_set_id)** combinations.
  - If it was re-executed with flag is **False**, then entries into **DB tables will not be updated**. However, scenario would be executed.
- When Select Run Dates is **Date Ranges** the input text box takes date format as shown in the following figure.
  The description for date range inputs are as follows:
  - **From Date**: Enter the starting date. This field is mandatory. For example, 20211007 .
  - **To Date**: Enter the ending date. If it is **None**, then the count should be mandatory to pass. This field is optional. For example, 20211022.
  - **Frequency**: Frequency of a scenario. By default, it is**Monthly**. This field is mandatory. For example, Weekly.
  - **Count**: The maximum number of recurrences to generate. This field is optional.
  - **By Month Day**: Integer or a sequence of integers, meaning the month days to apply the recurrence. This field is optional.
  - **Re-run/Update-run**: A Boolean flag (True/False) is provided to re-run the failed run dates or append the new run dates with the existing definition. The value is either True or False.

  The selected parameters are as follows:
  - **From Date**: 20211007
  - **To Date**: 20211022
  - **Frequency**: Weekly
  - **Count**: None
  - **By Month Day**: None
  - **Re-run/Update-run**: False

Show Selected Parameters

```
%python-pgx
asc.show_scenario_parameters()
```

> **Note:**
>
> You must examine all parameters carefully especially **Run Dates**before these are feeding to scenario execution notebook. In case of any discrepancies, set the “Run Dates” again and verify using this API.

Create Definition

You can create a definition that will be used to track all data that is to be generated by executing this scenario and any analysis to be carried out thereafter on this data.

Figure 4-9 Create Definition

- To use an existing definition and overwrite the existing results, set **Save With New Version = False, Clean Results = True** and set **Version** to the version you want to cleanup the existing results and start again. If the **Version** is set to **None**, the most recent version will be cleaned up.
- To use an existing definition and merge results to existing results, set **Save With New Version = False, Clean Results = False** and set **Version** to the version you want to merge. If the **Version** is set to **None**, the most recent version will be used.
- To create a new version and map the results to this version, set **Save With New Version = True** and **Version** to the version you want to create. If you set **Version** to **None**, a new version will be created after incrementing the most recent version by 1. **Clean Results** will have no impact when creating a new version.

Show Definition

To show definition, execute the following.

```
%python-pgx

asc.show_definition()
```

4.1.2.3 Trigger Scenario Execution

Execute scenario for given combination of Run Dates and Threshold Set Ids.

Execute Scenario

To execute the scenario, run the following.

```
%python-pgx

if run_flag == True:
    asc.asc_cleanup4rerun()

asc.execute_scenario()
```

Execution Status

You can verify the execution status for each run date using the Execution Status paragraph.

Figure 4-10 Execution Status

There are two types of execution status as follows:

- **Current Executions**: It shows only executions which are currently running in the notebook python session.
- **All Executions**: It shows current executions along with executions from the previous runs/ sessions for the same definition id and version.

Detailed Execution Status

If it is **True** then it gives detailed status of runs for each Job Ids. It shows error in case execution is failed, start time, end time and execution time for each job.

> **Note:**
>
> Scenario having multiple Job Ids, each Job Id status must have COMPLETED, then only execution for given run date would be considered as COMPLETED. Otherwise, it will be marked as FAILED.
> Error log will display the error occurred in paragraph from the Utility Scenario notebook.

Upon successful completion of each run, the post processing step can be completed as described below. This moves the generated events into a table in the schema which can be retrieved for further analysis.

Re-run/Update Flag and All Execution status

The user may want to execute the failed run dates again, considering the post-processing which was not yet executed for the current definition id and version. The run dates status in **All Execution Status** would be updated with the latest status of the run dates, whether the run_flag is True or False.

Only after executing post-processing, the asc_runid_lookup table gets entry for the current definition id and version.

Users must use the **Re-run/Update** flag is set to **True** for making any updates to existing run dates or the addition of any new run dates to the existing definition and version in the **asc_runid_lookup** table.

Upon completion of each run, only the **COMPLETED** run dates will be considered for the post processing step, which is described below. This moves the generated events into a table in the schema, which can be retrieved for further analysis.

Show Available Scenario Bindings

To show available scenario bindings, execute the following.

```
%python-pgx

asc.show_scenario_bindings()
```

Set Expression for Tunable Parameters

To set expression for tunable parameters, you must specify the parameters to be analyzed. This must be expressed in the same form as the scenario logic.

> **Note:**
>
> For many scenarios, the available bindings include both a **Base** and **Func** versions of a parameter. In such cases, determine the **Curr_Type** the scenario has been configured to use. If it has been set to **F**, the **Func** version of the parameter should be used; otherwise, the **Base** version should be used.

To set expression for tunable parameters, execute the following.

```
%python-pgx
tunable_parameters = "TOTAL_DEPOSIT_AMOUNT | TOTAL_WITHDRAWAL_AMOUNT"
setattr(asc,'tunable_parameters',tunable_parameters)
```

> **Note:**
>
> Using **setattr** method, the expression for tunable parameters is stored into class object for PreProd analysis.

Post Execution Activity

To perform post processing activity, execute the following.

```
%python-pgx

asc.scenario_post_processing()
```

This API perform the post scenario execution activities as follows:

1. Creating a unique **asc_run_id** in the **ASC_RUNID_LOOKUP** table.
2. Loading ATL alerts for given run_id in the **ASC_INVESTIGATED_ENTITIES** table.
3. Loading scenario data for given Run Dates into **ASC_EVENT_MASTER** table.

Review Results

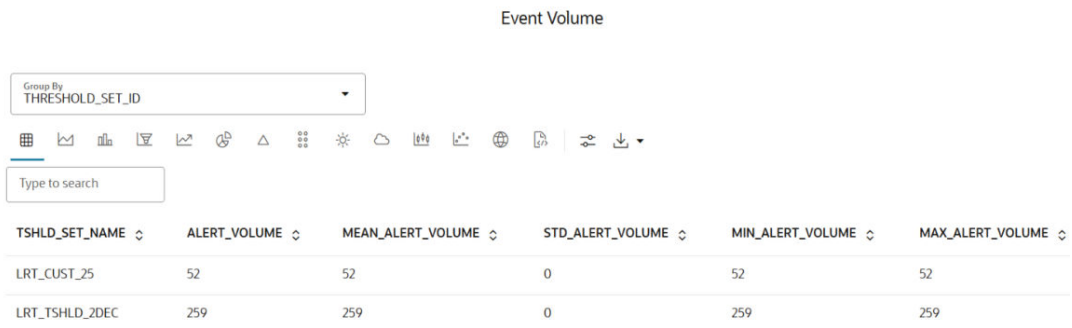
You can review the results of scenario execution. The summary of the event volumes is generated as shown below.

Event Volume

You are provided with multiple groups options across run dates as follows:

- **RUN_DATE**: Aggregate alerts by different run dates.
- **EVENT_TAG**: Aggregate alerts by event tag (ATL or BTL).
- **RUN_DATE_AND_EVENT_TAG**: Aggregate alerts by RUN DATE and EVENT TAG.
- **JURISDICTION_AND_RUN_DATE**: Aggregate alerts by JURISDICTION and RUN_DATE.
- **RUN_DATE_AND_THRESHOLD_SET_ID**: Aggregate alerts by RUN DATE and THRESHOLD SET ID.
- **EVENT_TAG_AND_THRESHOLD_SET_ID**: Aggregate alerts by EVENT_TAG and THRESHOLD SET ID.
- **THRESHOLD_SET_ID**: Aggregate alerts by THRESHOLD SET ID.
- **ALL**: Aggregate alerts by all inputs together (by SEGMENT_ID, RUN_DATE, EVENT_TAG).

Figure 4-11 Event Volume




```
asc.save_object()
```

Save Object

It saves the current ASC object in DB tracking table for later use in ATL, BTL, Impact and PreProd Analysis notebooks.

To save object, execute the following:

```
%python-pgx

asc.save_object()
```

4.1.3 PreProd Analysis

PreProd Analysis notebook provides thresholds for preprod scenarios.

4.1.3.1 ASC Instance Initialization for PreProd Analysis

You can import the necessary packages and create an instance for the PreProd analysis.

Navigate to the PreProd Analysis Notebook and the path is `Home / Modeling / Pipelines / ASC / Analysis / PreProd / PreProd Analysis Notebook`.

Download/Activate Conda Environment

Navigate to the **Download/Activate Conda Environment** paragraph and run this paragraph to download/activate the Conda environment.

ASC Instance Initialization

To create an instance, execute the following.

```
%python-pgx

from ds_interpreter_client.context.ds_context import PyDataStudioContext
ds = PyDataStudioContext()

from ofs_auto_ml.model_deployment.output_tracking import *;
from ofs_auto_ml.db_connection import *;
from ofs_aif.ofs_asc.asc import asc;

asc_obj = asc()
setattr(asc_obj,'objectiveId',objectiveId)
```

4.1.3.2 Attach to an Existing Definition Version

You must load the same object version which was used during Scenario Execution.

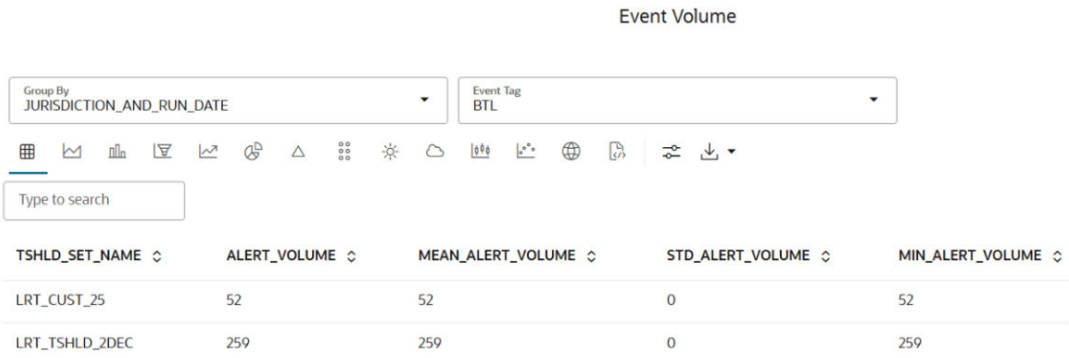
Figure 4-12 Load Existing ASC Object

Event Volume

Events volume by different inputs as follows:

- **RUN_DATE**: Aggregate alerts by RUN_DATE.
- **JURISDICTION**: Aggregate alerts by JURISDICTION.
- **JURISDICTION_AND_RUN_DATE**: Aggregate alerts by JURISDICTION and RUN_DATE.
- **JURISDICTION_AND_THRESHOLD_SET_ID**: Aggregate alerts by THRESHOLD_SET_ID and JURISDICTION.
- **THRESHOLD_SET_ID**: Aggregate alerts by THRESHOLD_SET_ID.

Figure 4-13 Event Volume



4.1.3.3 Data Loading

You can get data for selected Threshold Set IDs.

Getting Data

It retrieves the data from DB tables for selected tunable parameters. The tunable_parameters is an optional input.

- Input expression will be taken from class variable **asc.tunbale_parameters** if it was set during scenario execution.
- Input text box will only appear for user, if class variable **asc.tunable_parameters** is **None**. User can then set the expression again.

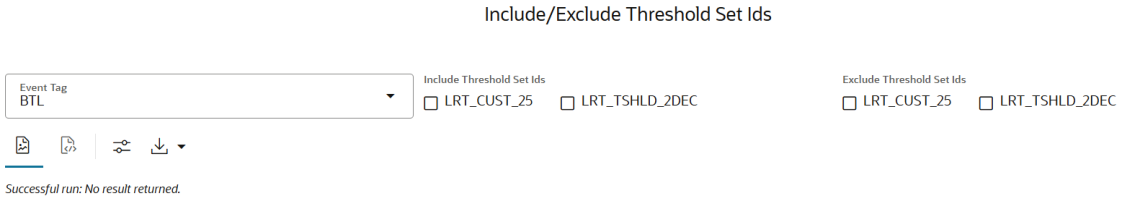
For example:

```
tunable_parameters = "Trans_Base_Amt | Trans_Ct"
```

Include/Exclude Threshold Set Ids

Use the **Include/Exclude Threshold Set Ids** paragraph to include or exclude relevant threshold set ids. If **None** of the threshold set ids are selected then all threshold set ids will be included.

Figure 4-14 Include/Exclude Threshold Set Ids



> **Note:**
>
> The PreProd analysis only supports the **BTL** event_tag. Selecting other options may result in an error.

Get Data

To get the data, execute the following.

```
%python-pgx

evented_data_pdf = asc.get_data(tag = event_tag,
                                tunable_parameters = tunable_param,
                                include_threshold_set_ids =
include_threshold_set_ids,
                                exclude_threshold_set_ids =
exclude_threshold_set_ids,
                                is_pre_prod_analysis=True)
z.show(evented_data_pdf.head())
```

4.1.3.4 Create/Add New Feature

You can create new features and add to the existing dataframe.

Create New Feature

To create a new feature, execute the following.

```
%python-pgx

import pandas as pd
def custom_feature(X, key_var='EVENT_ID'):
```

```
    V =
abs(X["TOTAL_DEPOSIT_AMOUNT"].fillna(0)+X["TOTAL_WITHDRAWAL_AMOUNT"].fillna(0)
)
    return pd.DataFrame({key_var : X[key_var], 'AVG_TRXN_PER_CT' : V})

df = custom_feature(evented_data_pdf)
```

Most scenarios use simple parameters which trigger an alert when the parameter value is greater than a threshold. However, some scenarios employ more complex parameters, which trigger an alert only when the parameter value is within a range. Such complex parameters e.g, Min_Percentage in the Rapid Movement of Funds scenario have to be appropriately transformed before the default scenario tuning methodology can be applied. For more information, see the Handling Complex Parameters section.

View Newly Created Feature

To view the newly created feature, execute the following.

```
%python-pgx

z.show(df.head())
```

Add New Feature with Existing Dataframe

A new feature will be added to the existing dataframe, so expression should also be updated. Use parameter **tunable_parameters** to update the existing expressions for the tunable parameters. Based on the expression, strata will be find out.

If it is **None**, then it would use the same expression which would have set in the scenario execution notebook.

To add a new feature with existing dataframe, execute the following.

```
%python-pgx

asc.add_to_event_dataset(df_list=[df])
```

Get Updated Dataframe

To get an updated dataframe, execute the following.

```
%python-pgx
evented_data_pdf = asc.get_event_dataset()
z.show(evented_data_pdf.head())
```

4.1.3.5 Computing Recommended Threshold

This topic provides option for choosing appropriate outlier technique and threshold computing technique.

Select Parameters for Tuning

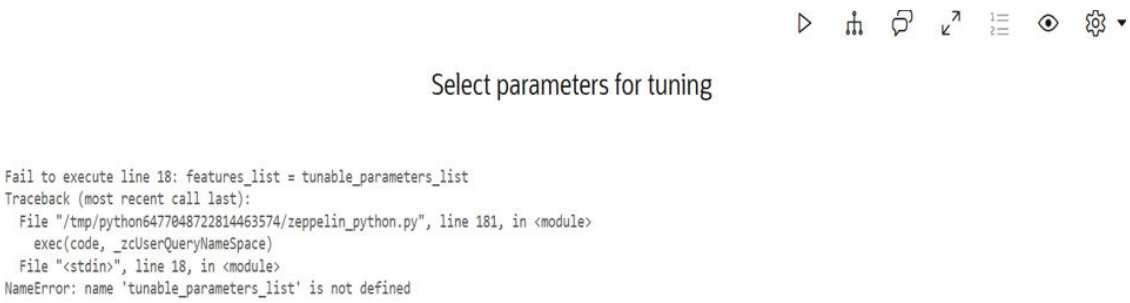
In this option, you need to select the parameters for which thresholds will be calculated. The default parameters are taken from the expression that the user must have set in the scenario

execution notebook. The parameters are stored in the class variable **asc.tunable_parameters_list**.

Limitation

The Select Parameters for Tuning paragraph might be failed because of missing class object before the variable used.

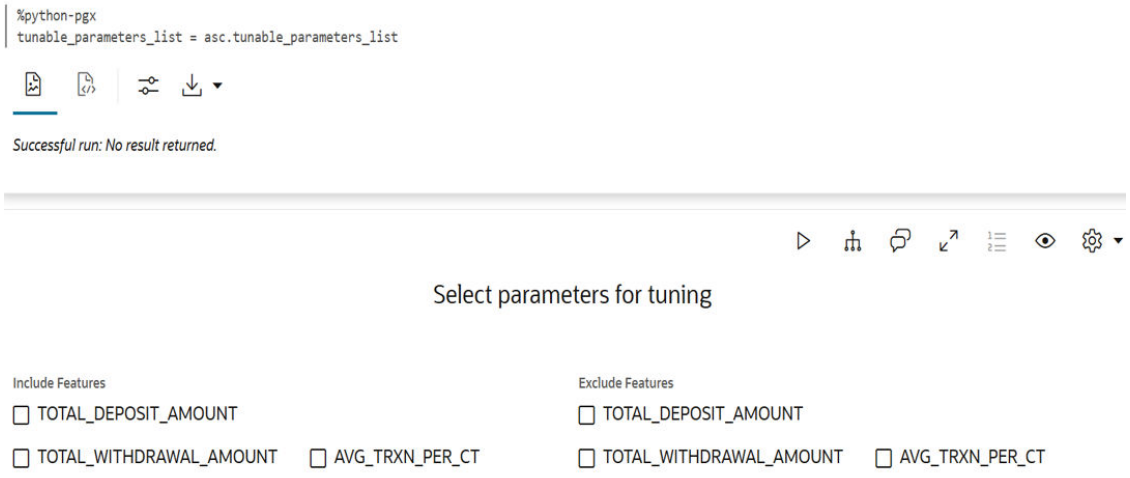
Figure 4-15 Error Message - Select Parameters for Tuning



To resolve this issue, add another paragraph above the failed paragraph with the following and run the paragraph.

```
%python-pgx
tunable_parameters_list = asc.tunable_parameters_list
```

Figure 4-16 Select Parameters for Tuning

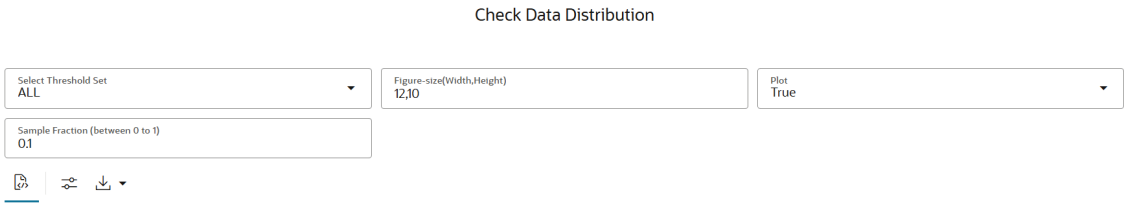


Check Data Distribution

You can see the data distribution for selected threshold set id and parameters and decide on the outlier techniques for calculating the thresholds to use.

Parameter Sample Fraction can be used to take the subset of data. The Parameter Sample Fraction value is between 0 to 1.

Figure 4-17 Check Data Distribution



Select Outlier Detection Method

You are provided with out-of-box univariate outlier methods. The outlier methods are divided into two groups based on the type of population. For each of the method, you can select the tail either **RIGHT**, **LEFT** or **BOTH** for removing outliers.

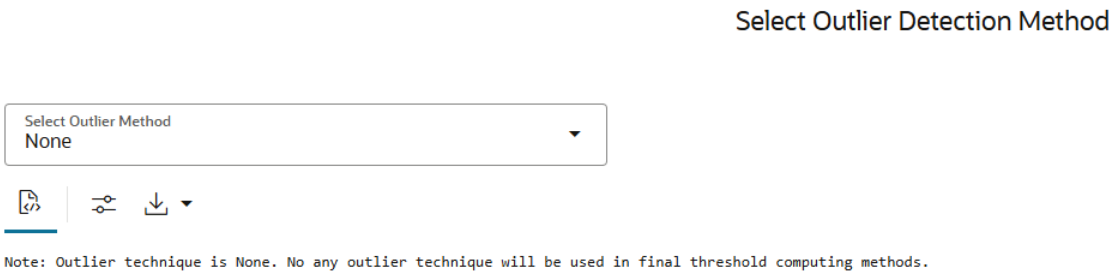
For Normal Population

- **zscore**: Outliers are number of standard deviations away from the mean.
  - Parameter **nstdev**. Default is 3.
- **IQR**: Inter-Quantile-Range (Q75-Q25). Outliers are considered which is away from Q75 + IQR*iqr_cut_off.
  - Parameter **iqr_cut_off**. Default is 3.
- **percent_outliers**: Percent of data points are considered to be outliers.
  - Parameter **outliers_proportion**. Default is 5%.

For Skewed Population

- **robust_zscore**: It is similar to Z-score method with some changes in parameters.
  - Parameter **nstdev**. Default is 3.
  - Since mean and standard deviations are heavily influenced by outliers, instead of them it uses median and absolute deviation from median.
  - Also called Median Absolute Deviation (MAD) method.
- **adjusted_boxplot**: It is similar to IQR method with some changes in parameters.
  - It customize the range of valid data for both side of tails differently.
  - Exponential model is used for fitting the data.

Figure 4-18 Select Outlier Detection Method

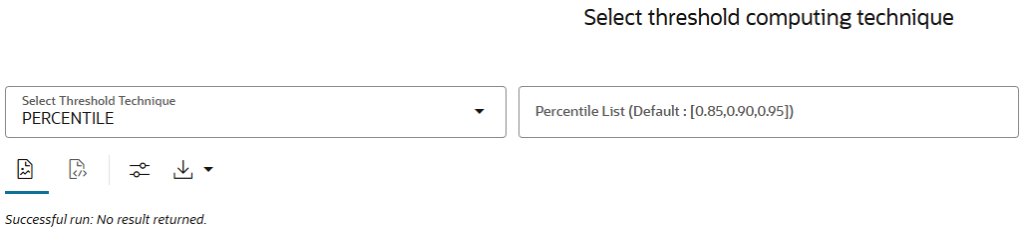


Select Threshold Computing Techniques

You are provided with following two out-of-boxes threshold computing techniques:

- **Percentile**: Thresholds are calculated based on the percentiles.
  - parameter **perc_list**. Default is [85,90,95]
  - User can pass its own percentile's list.
  - If **None** is passed, then threshold will be set at 85%, 90% and 95% for HR, MR and RR respectively.

Figure 4-19 Select Threshold Computing Techniques



- **Jump**: Thresholds are calculated based on the highest peaks found within the range defined.
  - Parameter **range**. Percentile range in which peaks will be found. It works similar like a range method in python.
  - If **range** is **None**:
    - Default range (85,100,0.1) is used.
    - Highest peak is found for HR between 85-90%, MR between 90-95% and RR between 95-99.9%.
    - If none of the peaks is found in range specified, then threshold is set at minimum percentile at 85%, 90% and 95% for HR, MR and RR respectively.
    - If any peak is missing for any range specified, then threshold for missing peak is set at same to last peak found. For example, No peak found for RR, then threshold for RR will be set to same threshold as of MR.
  - If **range** is **not None**:
    - range is passed as a **tuple**.
    - Three highest peaks will be found between the range defined by the user. For example, (80,100) and (85,100,0.2).
      If none of the peaks is found in range specified, then thresholds for all will be set at minimum percentile. For example, 85%.

      If any peak is missing for any range specified, then threshold for missing peak will be set at same to last peak found. For example, No peak found for RR, then threshold for RR will be set to same threshold as of MR.
    - range is passed as a **dict**
      - Highest peak will be found as per the range defined by the user for each risk level.

* For example, {'HR':(80,90,0.2),'MR':(90,95),'RR':(95,99,0.1)}, {'HR':(85,90),'MR':(90,95,0.1),'RR':(95,100,0.2)}
* If none of the peaks is found in range specified, then threshold will be set at minimum percentile passed by the user in specified range.

Compute Recommended Thresholds

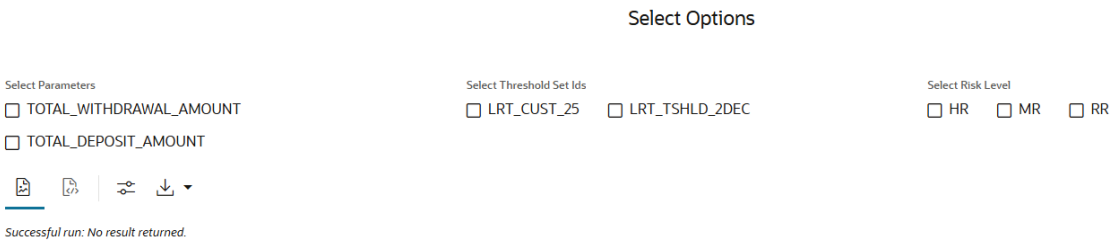
The API takes all the parameters (tunable parameters, outlier technique, and threshold computing technique) as an input and calculates the thresholds for each of the tunable parameters across the selected threshold set ids.

```
%python-pgx

asc.compute_initial_thresholds(features=features_list,
                               outlier_method=outlier_method,
                               technique=technique,
                               outliers_proportion=outliers_proportion,
                               nstdev=nstdev,
                               robust_zscore_cut_off=robust_zscore_nstdev,
                               iqr_cut_off=iqr_cutoff,
                               anomaly_proportion=anomaly_proportion,
                               perc_list=perc_list,
                               search_range=search_range,
                               tail=tail)
```

4.1.3.6 Show Recommended Thresholds

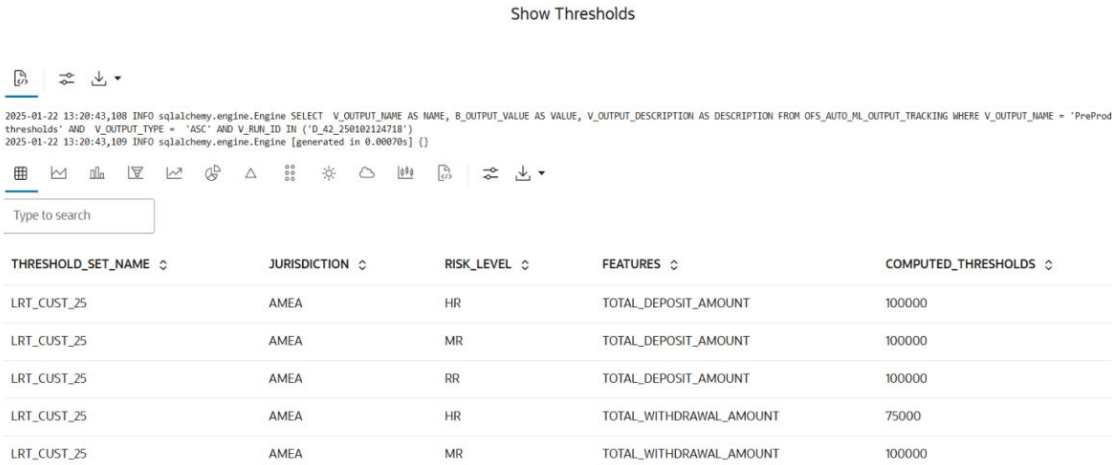
You can select the calculated thresholds for selected parameters, threshold set ids and risk levels.

Figure 4-20 Select Options



After selecting the options, run the paragraph to view the recommended thresholds.

Figure 4-21 Show Thresholds



THRESHOLD_SET_NAME	JURISDICTION	RISK_LEVEL	FEATURES	COMPUTED_THRESHOLDS
LRT_CUST_25	AMEA	HR	TOTAL_DEPOSIT_AMOUNT	100000
LRT_CUST_25	AMEA	MR	TOTAL_DEPOSIT_AMOUNT	100000
LRT_CUST_25	AMEA	RR	TOTAL_DEPOSIT_AMOUNT	100000
LRT_CUST_25	AMEA	HR	TOTAL_WITHDRAWAL_AMOUNT	75000
LRT_CUST_25	AMEA	MR	TOTAL_WITHDRAWAL_AMOUNT	100000