Oracle Financial Services Anti Money Laundering Event Scoring

User Guide

Release 8.0.7.0.0

April 2019

E92945-02

ORACLE

Document Control

Version Number	Revision Date	Changes Done
1.0	Created April 2019	Created the document.
Created by: Brijesh	Reviewed by: Swetha/Akhilesh/Renjith/Nagesh	Approved by: Viru/Shandar

Table of Contents

1 Preface

Oracle Financial Services (OFS) Anti Money Laundering Event Scoring (AMLES) application scores alerts that are generated from Anti Money Laundering (AML). The number of alerts generated from AML systems do not produce accurate results to be classified as valid alerts. However, this application optimizes the filtering of valid alerts and provides highly accurate results to prioritize alerts for investigation.

1.1 About this Guide

This document provides information to users to explain concepts and instructions to perform functions on the OFS Anti Money Laundering Event Scoring application.

1.2 Audience

This document is for users (creators and approvers) of the OFS AMLES.

1.3 Related Documents

This section provides a list of additional documents related to OFS AMLES Application Pack. You can access Oracle documentation for AMLES online from the Oracle Help Center (OHC) Documentation Library.

Additionally, you may see the following documents for OFS AAI related information on OHC:

- Oracle Financial Services Advanced Analytical Applications Infrastructure (OFS AAAI) Application Pack 8.0.7.0.0 Installation and Configuration Guide
- Oracle Financial Services Analytical Applications Infrastructure User Guide
- Oracle Financial Services Analytical Applications Infrastructure Administration Guide
- Oracle Financial Services Analytical Applications (OFSAA) Licensing Information
- Oracle Financial Services Analytical Applications (OFSAA) Generic Documents

To find additional information about how Oracle Financial Services solves real business problems, see our Web site at www.oracle.com/financialservices.

1.4 Acronyms used in this guide

Table 1 Acronyms

Acronym	Description
OFS	Oracle Financial Services
OFS AMLES	Oracle Financial Services Anti Money Laundering Event Scoring
OHC	Oracle Help Center
OFS AAAI	Oracle Financial Services Advanced Analytical Applications Infrastructure

Acronym	Description
OFSAA	Oracle Financial Services Analytical Applications
AML	Anti Money Laundering
WOE	Weight of Evidence

2 About OFS Anti Money Laundering Event Scoring

Use the application user-interface to create techniques that work to filter alerts generated from AML systems. You can score the alerts by passing the alerts data through the application and using the techniques created.

This chapter includes the following topics:

1. Understanding the OFS Anti Money Laundering Event Scoring Workflow
2. Understanding User Roles in OFS Anti Money Laundering Event Scoring

2.1 Understanding the OFS Anti Money Laundering Event Scoring Workflow

The application workflow is discussed here to present to you a summary of procedures and operations that would help better your understanding of the use of the application before you see sections that deal with topics in detail. The following list is a summary of steps:

1. Login to the OFSAA Application. See Accessing OFS Anti Money Laundering Event Scoring.
2. Create Execution Runs. See Creating and Editing Execution Runs.
3. View Results and Run Details in Data Groups. See Viewing and Analyzing Results.
4. Use Model Explanation to create combinations of model variables and analyze with what-if scenarios. See Using WOE Model Explanation, Using XGBoost Model Explanation, Using Naive Bayes Model Explanation, and Using Logistic Regression Model Explanation.
5. Request for deployment of the technique for a Data Group and request for approval by a user with approver privileges. See Managing Data Groups and Approving Techniques for Data Groups.
6. Score New Alerts. See Scoring Alerts.

2.2 Understanding User Roles in OFS Anti Money Laundering Event Scoring

OFS Anti Money Laundering Event Scoring application uses four-eyes functionality to provide two levels of maker and checker to create and approve execution runs. The two levels of users are defined as creators and approvers. For example, if your user profile is mapped to a Creator role, you can create an execution run and request for deployment, which has to be approved by a user with Approver role.

3 Getting Started

This chapter describes the login to the Oracle Financial Services Analytical Applications, to perform basic first-time login processes, and getting familiar with the UI components.

The following sections are available in this chapter:

3.1 Accessing OFS Anti Money Laundering Event Scoring

To access the application, your user profile has to be mapped to the required roles. See Understanding User Roles in OFS Anti Money Laundering Event Scoring for more information. The following is the procedure to access the *OFS Anti Money Laundering Event Scoring* window:

1. Enter the OFSAA URL in your browser. The Login window is displayed.

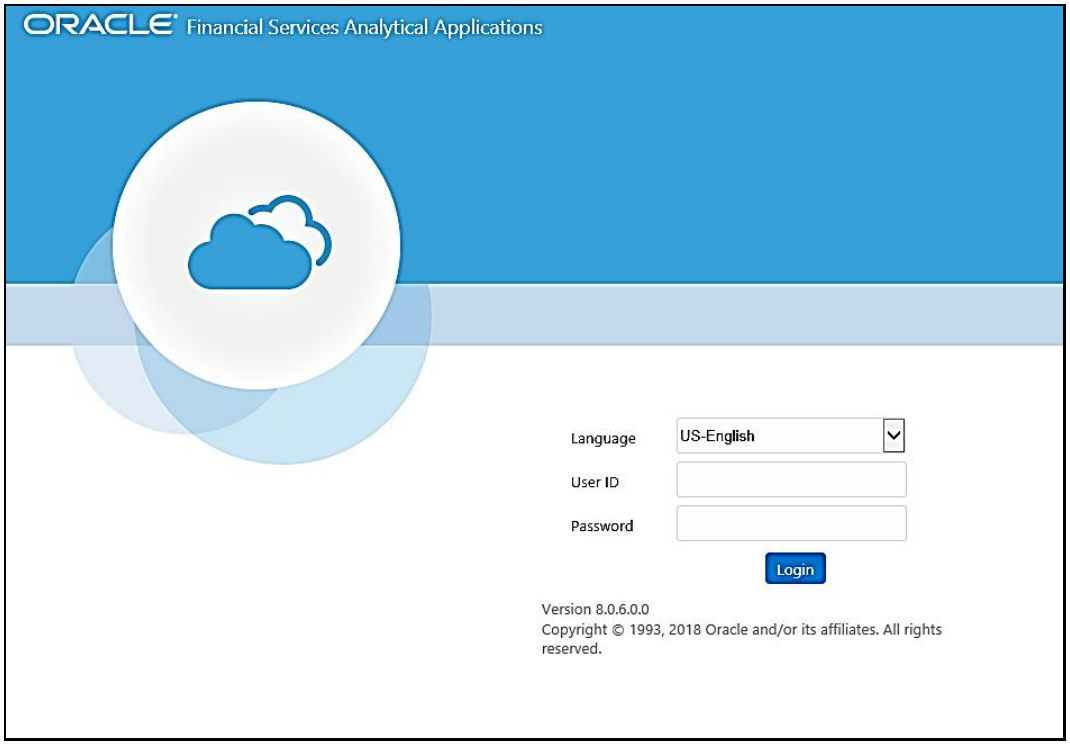

Figure 1 Login Window

2. Login to the OFSAA Application with your user credentials to view the *Applications* window. Alternatively, you can click from the header to view the *Applications* window.

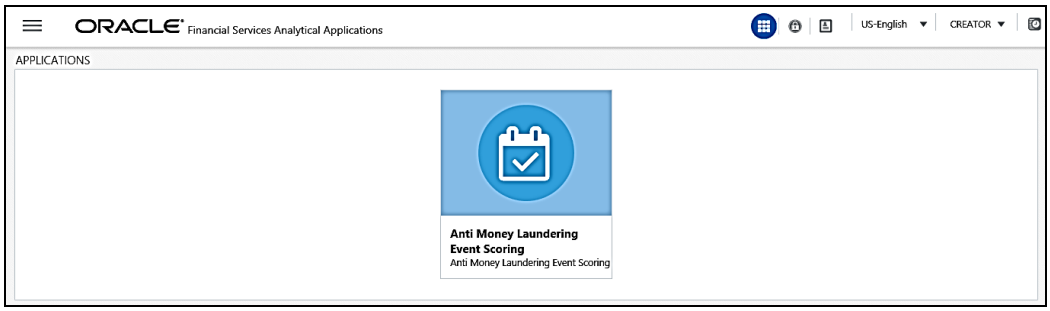

Figure 2 Applications Window

> **NOTE** The first time you login to the application, the *Change Password* window is displayed. To change password, see Changing Password.

3. Click **Anti Money Laundering Event Scoring** from the *Applications* window to view the Navigation list to the left. Click ☰ and toggle to view the list or hide it.

Figure 3 AMLES Home Window

4. Click **Anti Money Laundering Event Scoring** from the Navigation list to the left. The *Execution Run* window appears.

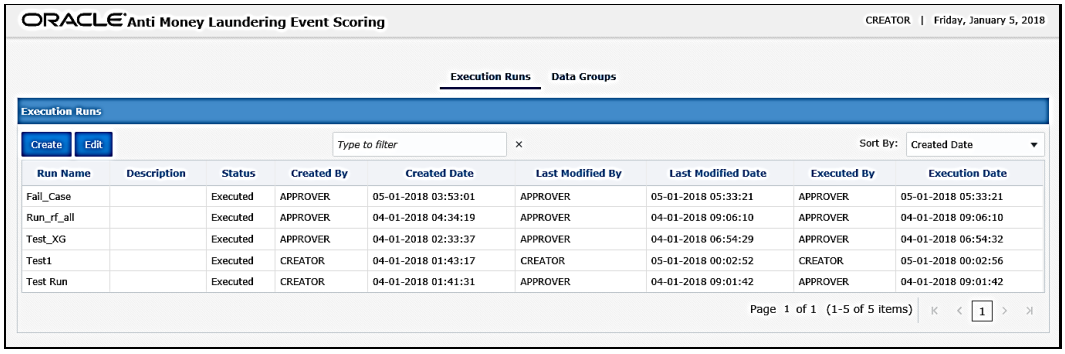

Figure 4 Execution Run Window

5. On the *Execution Run* window, a list of Execution Runs that are in saved draft or created state are displayed. Enter characters in **Type to filter** to perform a search for data groups and click the **Sort By** drop-down list to select and sort based on various options in the list. Click the pagination buttons at the bottom of the window to navigate.

Based on the roles assigned to your user profile, you can create, edit and approve execution runs. See further sections in this document for details.

3.2 Changing Password

The first time you login to the application, the *Change Password* window is displayed. Alternatively, you can also choose to change the password at any time.

To change password, follow these steps:

1. Click the **User Name** drop down and select **Change Password** from the header at the top to display the *Change Password* window.

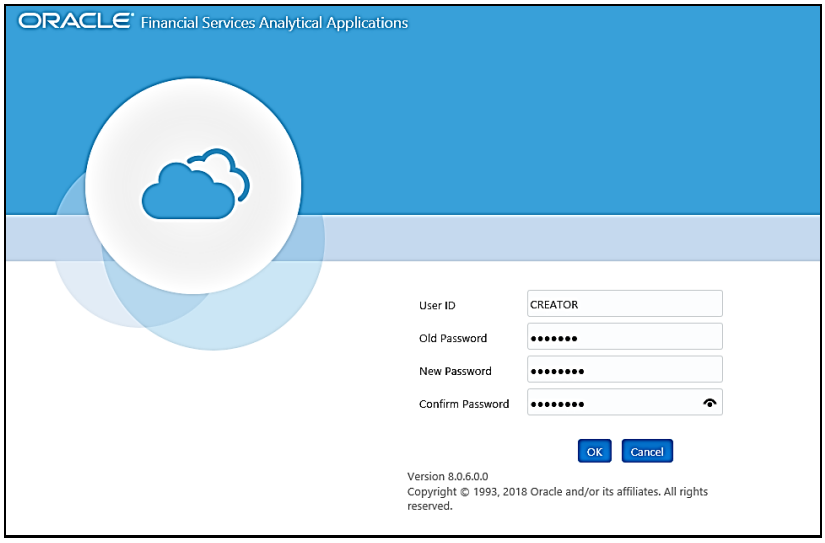

Figure 5 Change Password Window

2. Enter the **User ID**.
3. Enter the **Old Password**.
4. Enter **New Password** and re-enter in the **Confirm Password** field.
5. Click **OK**.

> **NOTE** Passwords are displayed as asterisks while you enter. This is to ensure that the password is not revealed to other users. For more information about passwords, see the Oracle Financial Services Analytical Applications Infrastructure User Guide.

3.3 Copyright Information

To access copyright information, click the **User** drop-down list and select **About** on the OFSAA login window. The Copyright text displays in a new window.

3.4 Knowing the user-interface components in AMLES

This section describes the general features of OFS AMLES UI. These are the common features that are found across the various modules of OFSAA. It describes the organization of the user interface and provides step-by-step instructions for navigating through the application.

3.4.1 Home Page Components

The Home Page contains the following sections.

- Header
- Navigation List
- Tab Bar
- Content
- Footer

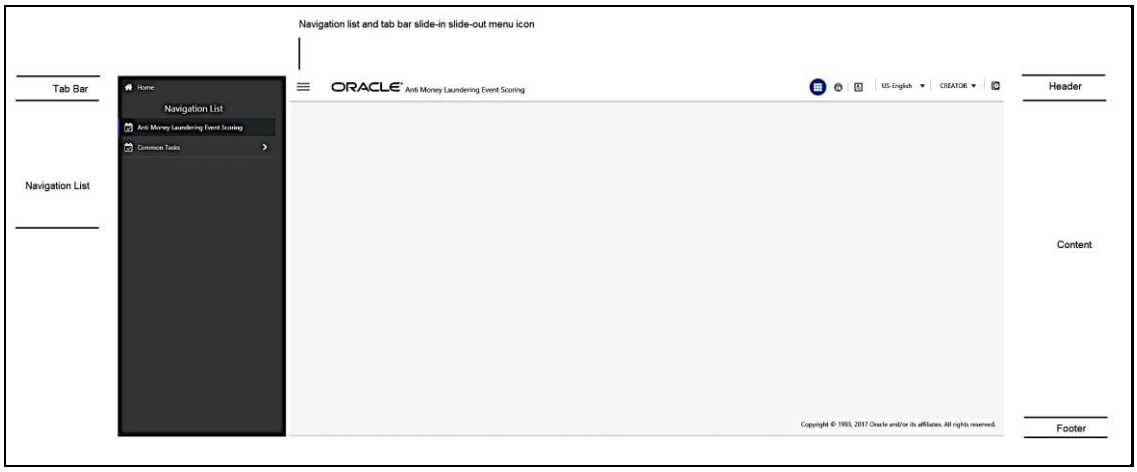

Figure 6 Home Page Components

3.4.1.1 Header

The Header displays icons, buttons and text for generic information and access to the OFSAA application's features. The following user-interface elements are displayed for AMLES:

- **Applications:** Click this icon to display applications in a Tiles menu on the content window.
- **Administration:** Click this icon to display administration tools in a Tiles menu on the content window.
- **Language:** Displays the selected language. Click to select from the options in the drop down.
- **User Name:** Displays the logged in user name. Click to select from the following options in the drop down:
  - **Preferences:** Select to set the Home page.
  - **About:** Select to view the copyrights and third-party information.
  - **Change Password:** Select to change the password in the Change Password window.
  - **Logout:** Select to log out of the application.
- **Last login date and time, and last failed login date and time:** Click to view the last login date and time, and the last failed login date and time.

3.4.1.2 Applications

The applications available are displayed in a Tiles menu on the content window. Click the Tiles to open the selected application. For AMLES, the following applications are available:

- Anti Money Laundering Event Scoring
  For detailed information on other applications, see the Oracle Financial Services Analytical Applications Infrastructure User Guide.
- Common Tasks
  For detailed information, see the Oracle Financial Services Analytical Applications Infrastructure User Guide.
  - Data Model Management
  - Data Management Framework
  - Operations
  - Run Rule Framework

3.4.1.3 Administration

The administration feature displays on Tiles menu. Click the Tiles to navigate further. For AMLES, the following administration tools are available:

- System Configuration - This tile helps System Administrators to provide security and operational framework required for the Infrastructure. System Administrators can configure Server, Database, OLAP, and Information Domains, along with other configuration processes such as segment and metadata mapping, segments to securities mapping, and rules setup. The System Configuration is a one-time activity, which helps the System Administrator make the Infrastructure system operational.
- Identity Management - This tile helps System Administrators to manage Users, User Groups, and the functions each User or User Group can access. For more information about managing Users and User Groups, see the OFS Analytical Applications Infrastructure Administration Guide.
- Database Details
- Manage OFSAA Product Licenses
- Create New Application
- Information Domain
- Processing Modeling Framework

> **NOTE** For more information on using the administration tools listed in the previous list, see the Oracle Financial Services Analytical Applications Infrastructure User Guide.

3.4.2 Navigation List

The Navigation list is a slide-in slide-out UI element that appears on clicking the Menu (triple bar) icon. This element displays a list of links in a menu based on the application selected from the Applications menu and the access rights assigned to the logged in user. Click on the various links in the list to navigate further and open in the content window. For more details, see the Oracle Financial Services Analytical Application Infrastructure User Guide.

3.4.3 Tab Bar

The Tab bar contains icons and text to navigate to specific OFSAA application modules. Click the Home button in Tabs to navigate to the Applications page on the content window. The elements displayed in Tabs is dependent on the OFSAA application logged into and the access roles assigned to the user.

4 Creating and Editing Execution Runs

Create and execute runs to explore historical data and find the best suited technique for a data group.

The following is the procedure to create Execution Runs:

1. Click **Create** from the *Execution Run* window to view the *Execution Run Details* window.

> **NOTE** Click **Back** to go to the previous window and click **Reset** to clear all entries in the fields.

2. Enter the details in the fields on the *Execution Run Details* window.

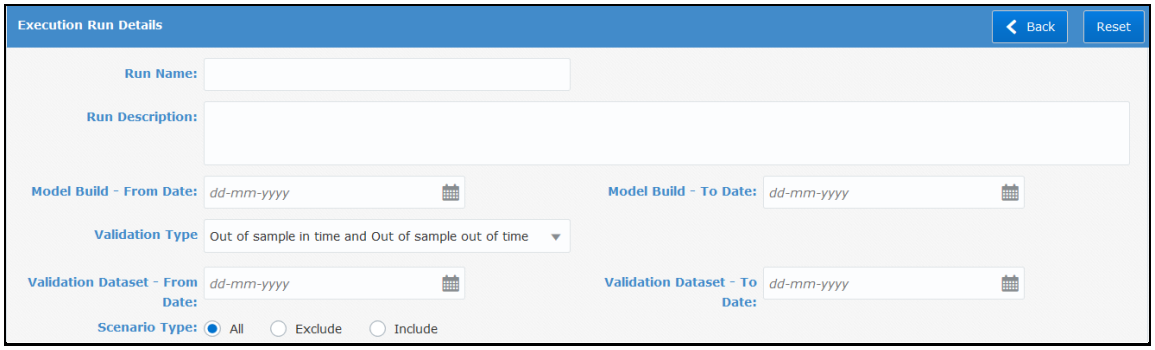

Figure 7 Execution Run Details Window

3. Enter **Run Name** and **Run Description**.
4. Click **Date-Time Editors** for **Model Build - From Date** and **Model Build - To Date** to select the **Historical Data Date Range**.
5. Select from **Validation Type**.

   Model validation step is required to test the performance of machine learning model on the data that has never been used in training the model. This data is referred to as out-of-sample. For this purpose, based on the availability, out-of-sample validation data can be sampled from the same population (In-time) on which the model is trained, or can be sourced from a different time period (out-of-time) for better evaluation. The later approach is preferred where application of a model to a population changing over time is the concern.

   The options are:

   a. **Out of sample in time** - validation, where the validation data is sampled from the population dataset used for developing the model. In this case, the validation data is separated from the population and the remaining portion is used for training the model. The performance of the model on the validation data is reported.

   b. **Out of sample out of time** - validation, where the model is tested on a different population from a different time period. In this case, the model is trained and evaluated on data from different time periods.

   c. **Out of sample in time and Out of sample out of time** - combination of Out of sample in time and Out of sample out of time, where the model is validated both on In-time data and Out-of-time data.

6. Click **Date-Time Editors** for **Validation Dataset - From Date** and **Validation Dataset - To Date** to select the validation dataset date range. This option is applicable for Validation Type - Out of sample out of time and Out of sample in time and Out of sample out of time.
7. Select the **Scenario Type** from the options **All**, **Exclude** and **Include**.

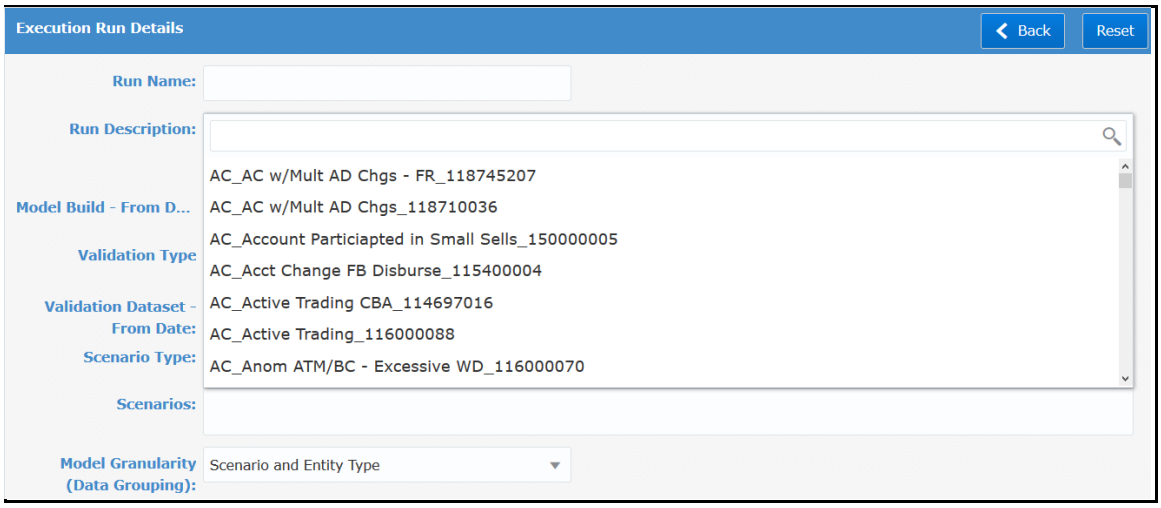

Figure 8 Scenario Type Selection

a. **ALL** - Select to include all scenarios in the run.

b. **Exclude** - Select to exclude certain scenarios from the run. On selecting this radio button, the **Scenarios** field appears. Click the field to display a drop-down list of Scenarios. Select the Scenarios that you want to exclude from the run. To remove a Scenario from the exclude list, click **X** on the selected Scenario.

> **NOTE** The Scenarios are configurable in the DIM. For more details, see OFS AMLES Administration and Configuration Guide.

c. **Include** - Select to include certain scenarios in the run. On selecting this radio button, the **Scenarios** field appears. Click the field to display a drop-down list of Scenarios. Select the Scenarios that you want to include in the run. To remove a Scenario from the include list, click **X** on the selected Scenario.

8. Click the **Model Granularity (Data Grouping)** drop-down list to select the relevant groups for the run from the following options:

a. Scenario and Entity Type

b. Scenario, Entity Type and Segment

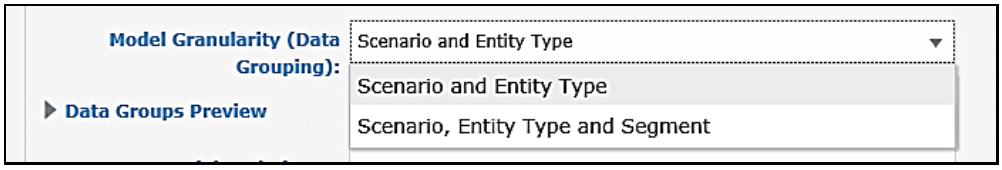

Figure 9 Model Granularity (Data Grouping)

9. Click **Data Groups Preview** to expand and view the list of the selected data groups.

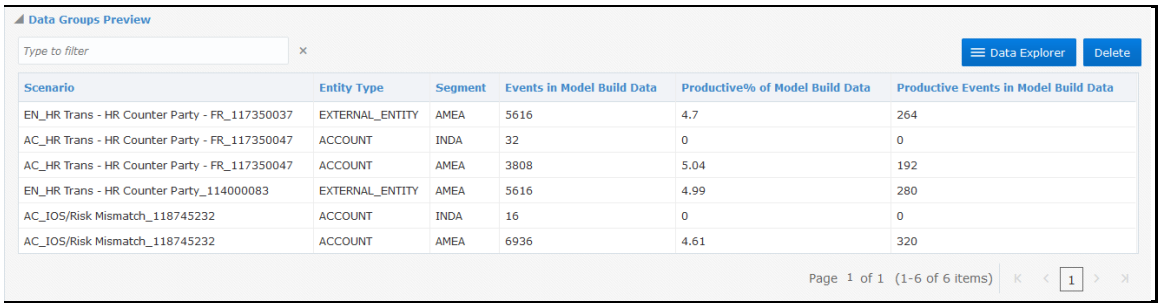

Figure 10 Data Groups Preview

10. On the *Data Groups Preview* section, you can view columns for data groups (**Scenario**, **Entity Type**, and **Segment** (if selected)), **Events in Model Build Data**, **Productive% of Model Build Data** and **Productive Events in Model Build Data**. Enter characters in the **Type to filter** field to perform a search. Select a row and click **Data Explorer** to view details in the *Data Group Details* window. To delete a data group, select the row and click **Delete**.

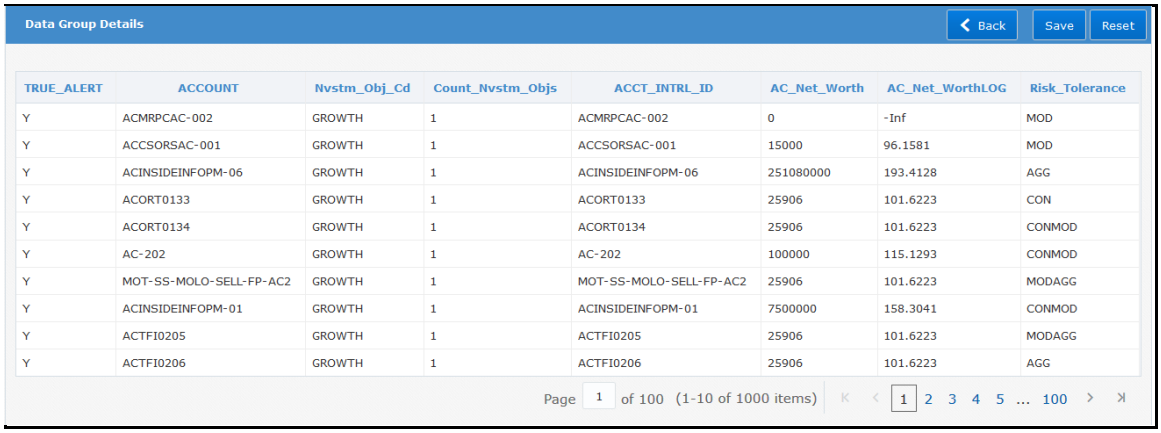

Figure 11 Data Group Details Window

> **NOTE** On the *Data Group Details* window, click **Back** to go to the previous window and click **Reset** to clear all entries in the fields.

The following is the procedure to view and plot the data from the *Data Group Details* window:

a. Enter the required percentage of data for analysis in **Sampling Percentage**. By default, this value is 33.

b. Click **Apply Transformation** to expand and view the **Transform Script** field. You can write a custom R Script in Transform Script to transform the data before building the model. This field is optional. Click **Apply** to apply the transform script and preview. Click **Save** to save the transformation script to the model.

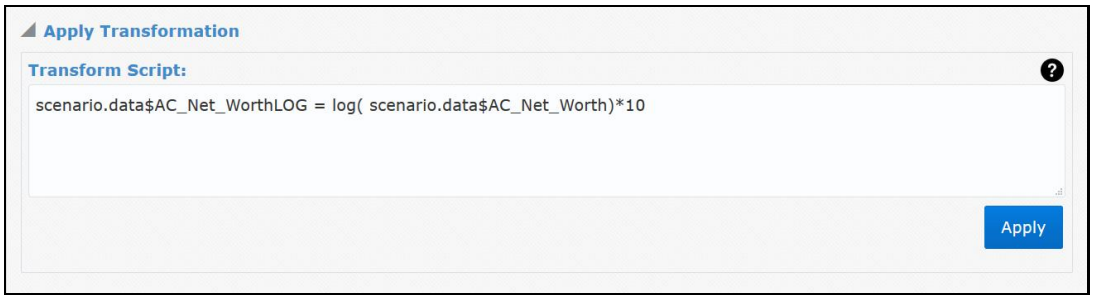

Figure 12 Transformation Script Field

NOTE Click the button to view the online help for Transformation, or see Performing Transformation of Data in this guide for information.

c. Click to select include or exclude variables required for analysis.

i. **All** - Default option. All the variables are selected.

ii. **Include** - Select to include variables that are required.

iii. **Exclude** - Select to exclude variables that are not required.

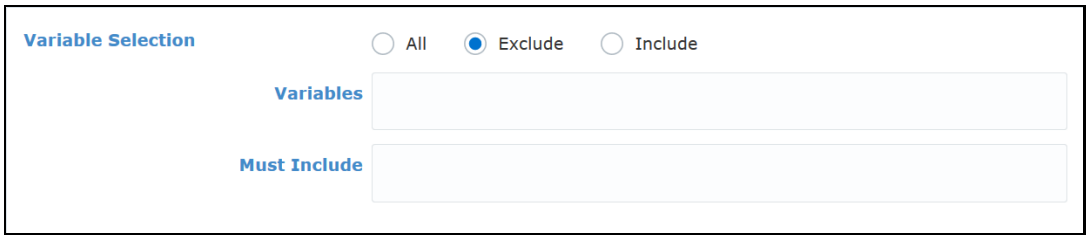

Figure 13 Variable Selection Options

d. Click to select **Univariate Plot**, **Empirical Logit Plot**, **Weight of Evidence**, **Information Value Matrix**, **Bivariate Plot** or **Correlation Heat Map** from **EDA** (Exploratory Data Analysis) to plot a visual representation of the data and click **Plot**.

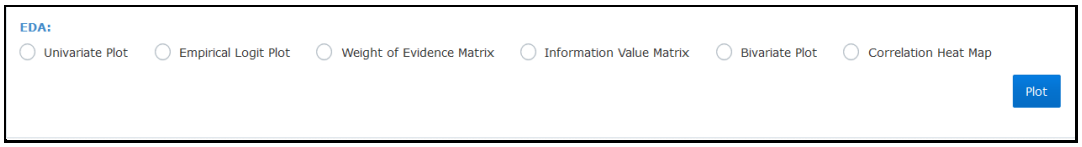

Figure 14 EDA Options

e. View the details of the plot in tabular form in the section following the EDA fields.

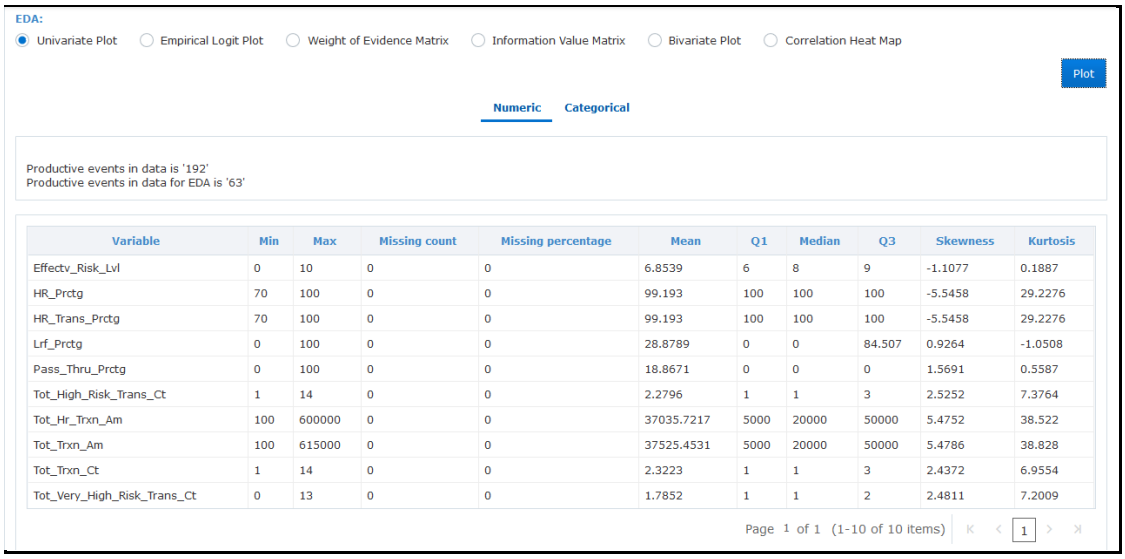

Figure 15 EDA Fields

i. For Univariate Plot, **Numeric** tab appears by default. Click a Variable in the Univariate Summary grid to view the plot.

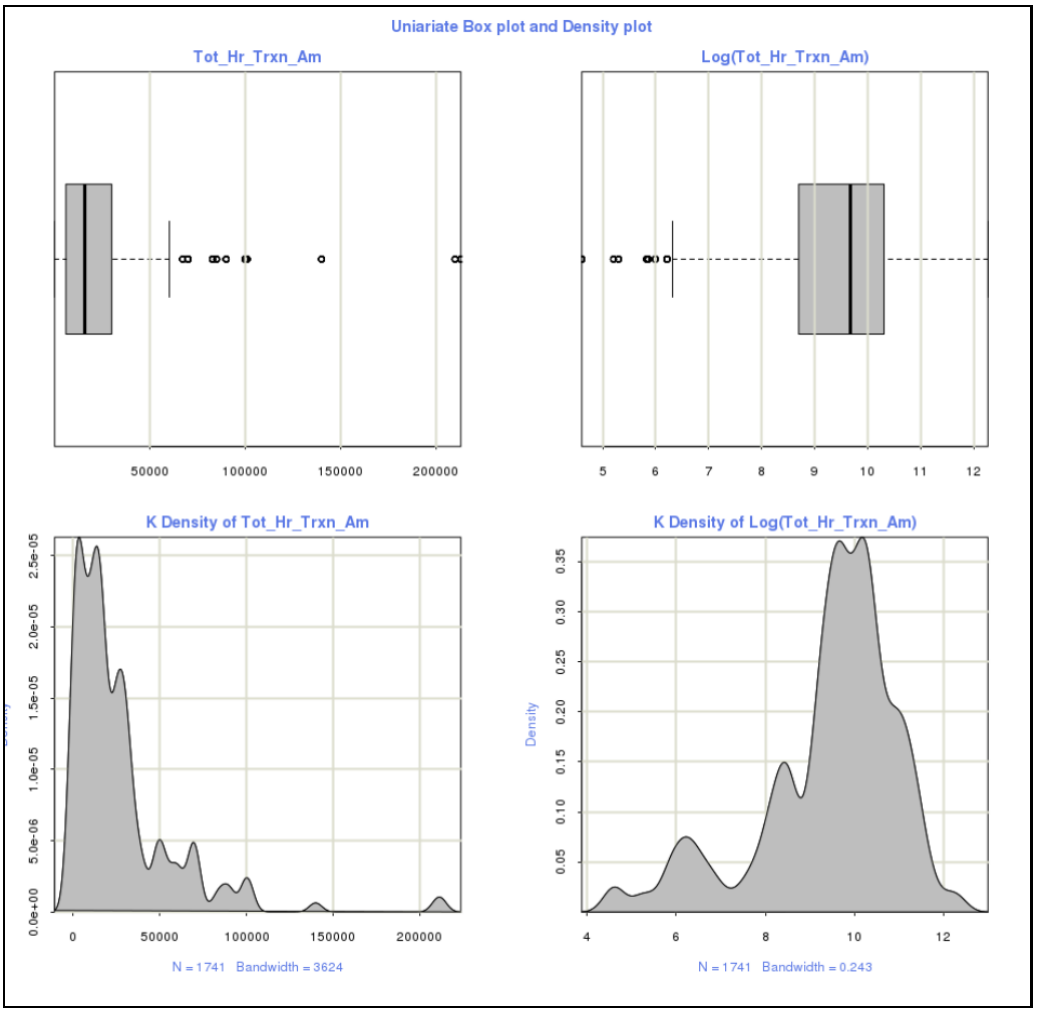

Figure 16 Plot in Numeric Tab

ii. Click **Categorical** tab and click a Variable in the Categorical Summary grid to view the Frequency Distribution Summary and Plot.

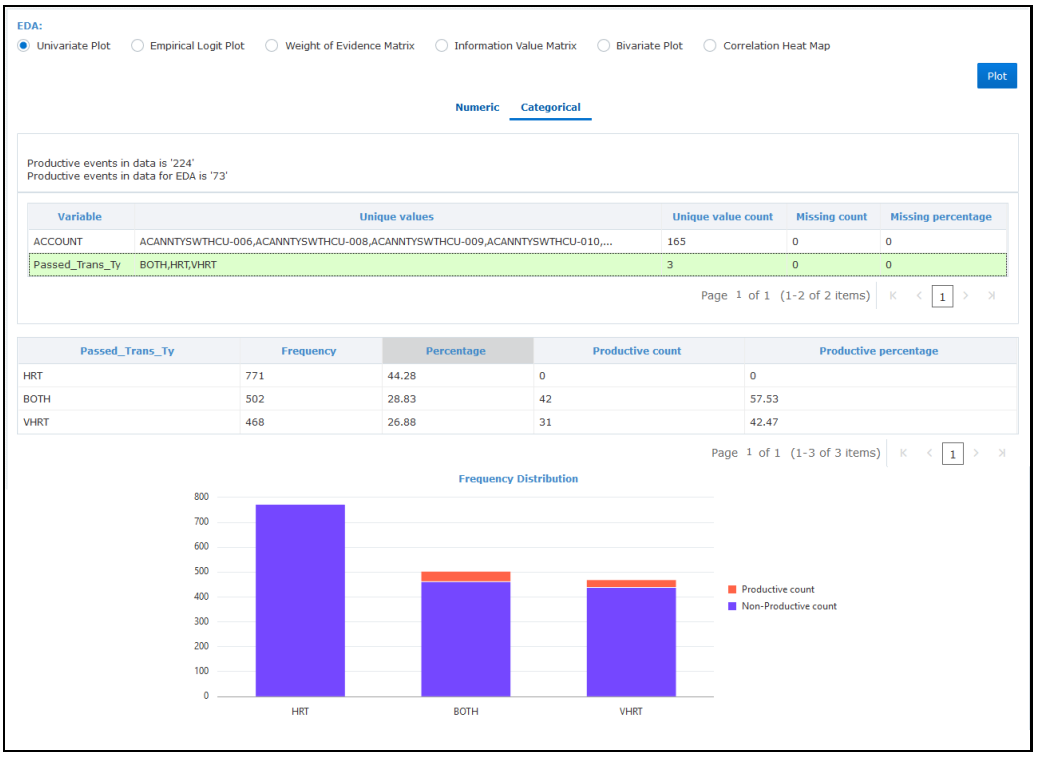

Figure 17 Plot in Categorical Tab

f. Click on a Variable and scroll down further to view the details in graphical format. Use the ‹ and › buttons to navigate through the various pages of the graph.

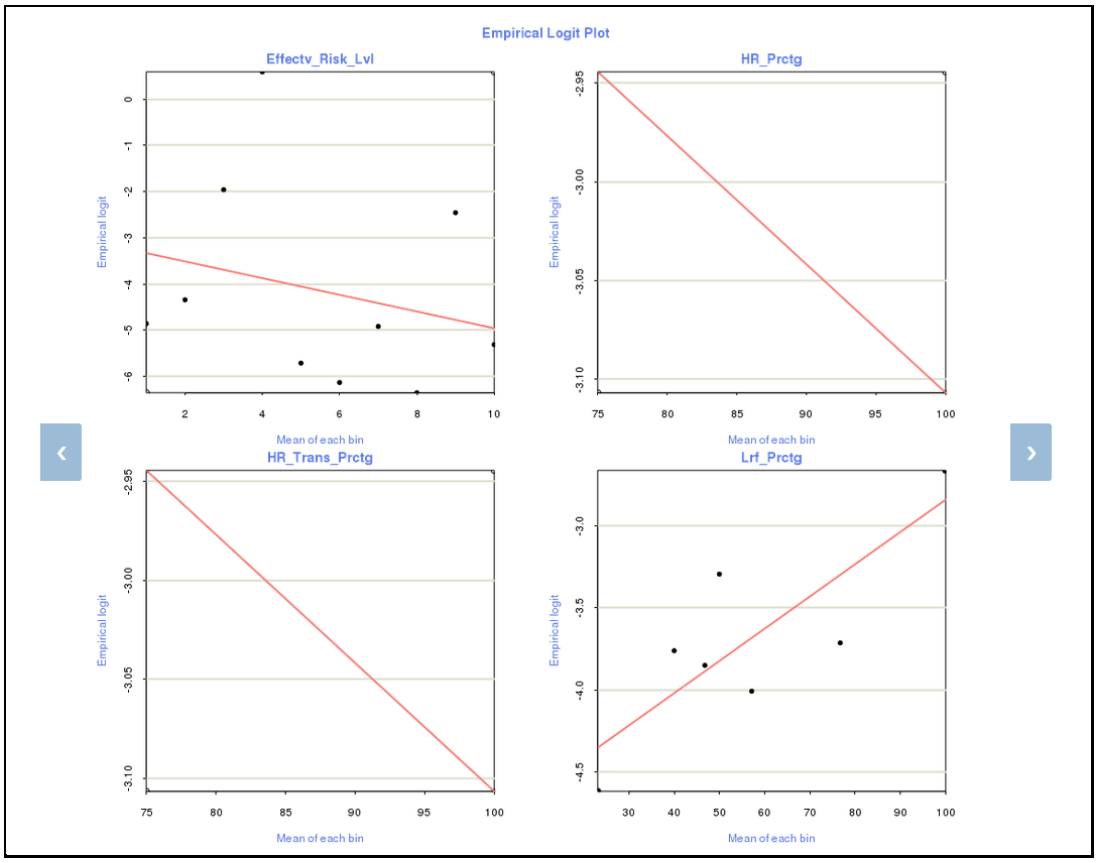

Figure 18 Plot in Graphical Format

g. Click **Back** to go to the previous window.

11. Click the **Model Techniques** field to view a drop-down list. Select from the list that you want to run for the historical data that is filtered for the conditions entered in the previous field. To remove a Model Technique from the field, click **X**.

Figure 19 Model Techniques List

12. Click **Technique Control Parameters** expand and view details for tabs **Common**, **WOE Logistic Regression**, **XG Boost**, and **Logistic Regression**. Enter data in the fields to suit your requirement to configure the number of alerts and the percentage considered for optimization. It helps improve the accuracy of the alerts. Click the headers to view the details for the respective tabs.

    a. **Common** – Select Yes or No for **Enable Null check on columns**. Enter the **Missing Value percentage Allowed** from 0 to 100. Select Yes or No for **Enable Zero Variance Check on columns**. Select Yes or No for **Handle Linear Combinations**. Enter the percentage value for **Maximum percentage of unique levels in categorical variables**. Enter the value for **Minority oversample ratio**. Enter the integer value for **Minimum minority observations in CV Folds**. Select Auto or Manual for **Data Partition**. Enter **Minimum Validation Data Percentage** from 0 to 100. Enter the integer value for **Max cross validation runs per model**

and **Max cross validation folds per repeat**. Minimum value for **Max cross validation runs per model** and **Max cross validation folds per repeat** is 2.

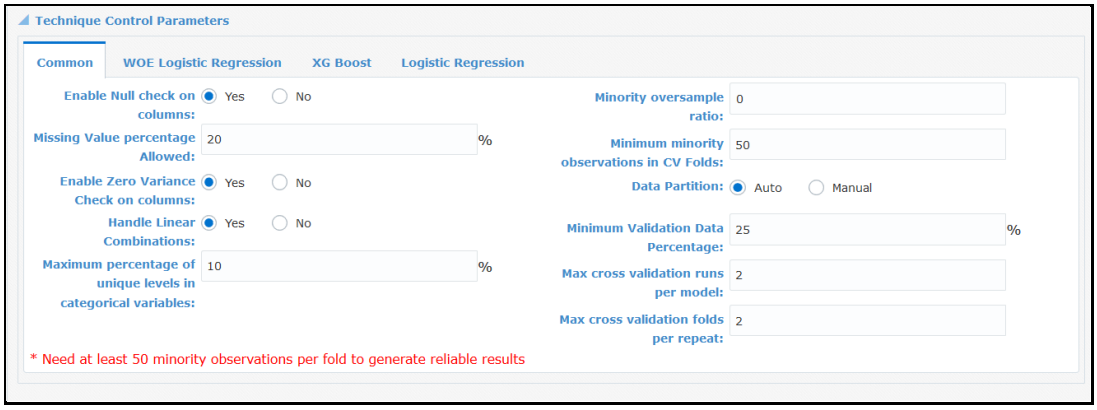

Figure 20 Common - Technique Control Parameters

b. WOE Logistic Regression – Click and select the options from the drop-down for **Binning Type**. Enter the integer value for **Number of Bins for Quantile** and **Number of Bins for Interval**.

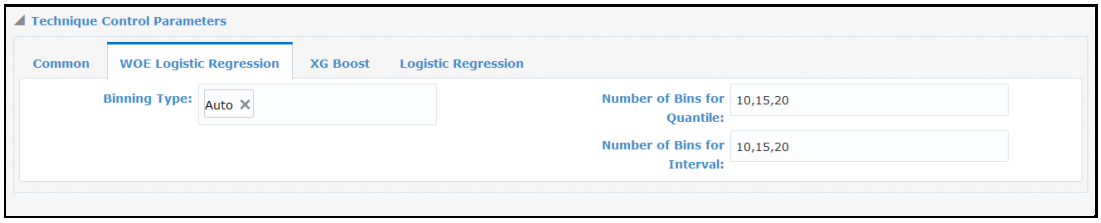

Figure 21 WOE Logistic Regression - Technique Control Parameters

c. XG Boost – Enter the numerical values for **Learning rate(eta)**, **Minimum Split Loss(gamma)**, **Maximum depth of a Tree**, **Minimum Child Weight**, **Column Sample**, **Sub Sample**, **Lambda**, **Alpha**, **Max Number of Iterations** and **Early Stop**.

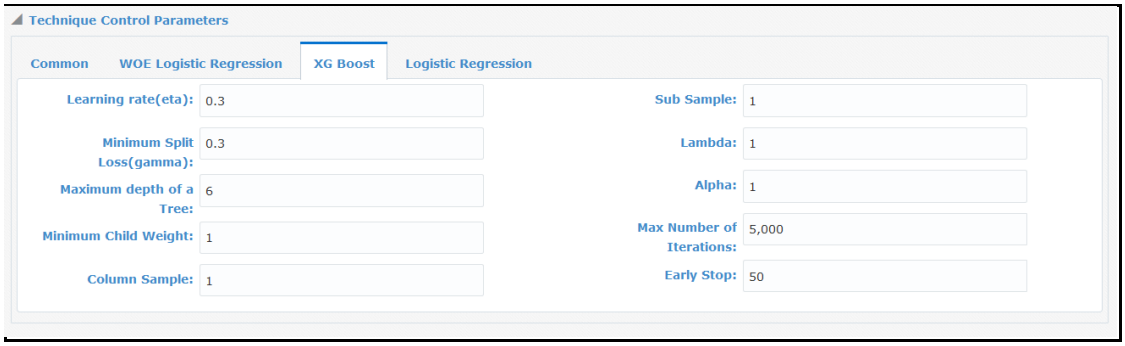

Figure 22 XG Boost - Technique Control Parameters

NOTE

- XG Boost does not predict for a record with new factor levels for any of its variables.
- Such predictions (data with new factor levels) are given a unique score of 999 and grouped under High-3 category events by the Event Scoring application. This is applicable to both model validations and real-time scoring.
- For Out of sample out of time, XG Boost Model validation will fail if factors are not common between In time (model build data) and Out time (Out of sample out of time data) data. Most of these variables are ID variables and should be removed using transformation script.
- For Out of sample in time and Out of sample out of time, if Out of sample out of time validation data contains new factor levels, then Out of sample out of time Validation AUC will be unreliable, as it uses scores such as 999.

d. **Logistic Regression** – Select True or False from **Ridge Flag**. On selecting **Ridge Value (Alpha)** field is displayed. Enter numeric values in **Ridge Value (Alpha)**.

Figure 23 Logistic Regression - Technique Control Parameters

NOTE You can access the tabs based on the selections from the Model Techniques field.

13. Click **Save** to save the entries on the creation window. Click **Update** to update changes. Click **Run** to execute the scripts. After the scripts are run, you can click **Results** to view the results of the run in the *Data Groups* window. For more information on Results, see Viewing and Analyzing Results.

4.1 Editing an Execution Run

To edit an Execution Run, select and click **Edit** on the *Execution Runs* window to view the details in the *Execution Run Details* window. Edit the fields that can be edited. **Run Name** field is read-only and you cannot edit it. For field descriptions, see Creating and Editing Execution Runs section.

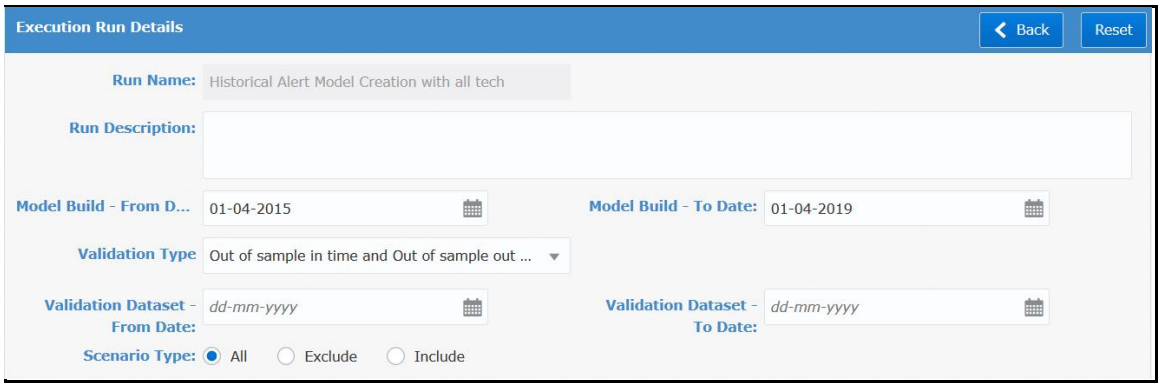

Figure 24 Execution Run Details Window

5 Performing Transformation of Data

Perform data transformation functions available on the Transformation field in the AMLES. The sections in this topic provide information to prepare and transform data.

NOTE

- ORE Frame object **scenario.data** is a data handle to the current data group. This data handle or object is the base to apply user transformations.
- Column **TRUE_ALERT** displayed in data explorer provides a clearer picture about event distribution. User Transformation should not attempt to modify or drop this column.
- Refrain from using single quotes (') in transformation scripts. Instead use double quotes where it is necessary, or use two single quotes ('') to work with transformations.
- **OREdplyr** is available.
- Do not apply cross observations' transformations such as calculate **Z-score** (how many standard deviations away from the mean) of a numeric variable.
- Transformations, which use aggregations across data or rows such as scaling and centering may lead to reduced accuracy. So it is recommended not to use these aggregations.
- Dropping and adding of columns is allowed.
- Remove columns of type ID using transformation scripts. Retaining such measures and variables reduces model accuracy.

5.1 About Preparing Data in the Database

The following basic data transformations operations can be performed on ORE Frames:

1. Selecting
2. Transforming Data
3. Sorting

5.2 Selecting

1. Selecting Data by Column
2. Selecting Data by Value

5.2.1 Snapshot of scenario data before proceeding with the transformations

Example: `head (scenario.data, 10)`

	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	Trusted_Trans_Prctg	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	Tot_Vhr_Trxn_Am	Hr_Prctg
1	N	100000	100000	100000	0	100000	100000	100000	100000	100000	100
2	N	100000	100000	100000	0	100000	100000	100000	100000	100000	100
3	N	100000	100000	100000	0	100000	100000	100000	100000	100000	100
4	N	96600	96600	96600	0	96600	96600	96600	96600	38000	100
5	N	120510	120510	120510	0	120510	120510	120510	120510	10110	100
6	N	10920	10920	10920	0	10920	10920	10920	10920	0	100

Table 2 Snapshot of scenario data before proceeding with the transformations

head() filters the record count to specified number.

5.2.2 Select Data by Column Names

Example :

```
scenario.data = scenario.data[,c(   "Tot_Dep_Am",
"Curr_Mnth_Cr_Amt",
"Trusted_Trans_Prctg",
"Hr_Prctg")];
```

	Tot_Dep_Am	Curr_Mnth_Cr_Amt	Trusted_Trans_Prctg	Hr_Prctg
1	100000	100000	0	100
2	3002.735	3002.735	0	100

	Tot_Dep_Am	Curr_Mnth_Cr_Amt	Trusted_Trans_Prctg	Hr_Prctg
3	100000	100000	0	100
4	500	500	0	100
5	100000	100000	0	100
6	1270	1270	0	100

Table 3 Select Data by Column Names

5.2.2.1 Select Data by Column Index

Example:

```
scenario.data = scenario.data[,c( 1,5,9,11 ) ];
```

OR

```
scenario.data = scenario.data[,1:5];
```

	Tot_Dep_Am	Curr_Mnth_Cr_Amt	Trusted_Trans_Prctg	Hr_Prctg
1	100000	100000	0	100
2	3002.735	3002.735	0	100
3	100000	100000	0	100
4	500	500	0	100
5	100000	100000	0	100
6	1270	1270	0	100

Table 4 Select Data by Column Index

5.2.3 Data filters or select data by value

Example 1:

```
scenario.data = scenario.data[scenario.data$Tot_Dep_Am> 10000,];
```

	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	Trusted_Trans_Prctg	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	Tot_Vhr_Trxn_Am	Hr_Prctg
1	N	100000	100000	100000	0	100000	100000	100000	100000	100000	100
2	N	100000	100000	100000	0	100000	100000	100000	100000	100000	100
3	N	100000	100000	100000	0	100000	100000	100000	100000	100000	100
4	N	96600	96600	96600	0	96600	96600	96600	96600	38000	100
5	N	120510	120510	120510	0	120510	120510	120510	120510	10110	100
6	N	10920	10920	10920	0	10920	10920	10920	10920	0	100

	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	Trusted_Trans_Prctg	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	Tot_Vhr_Trxn_Am	Hr_Prctg
214	N	126000	126000	126000	0	126000	126000	126000	126000	126000	100
215	N	600000	600000	600000	0	600000	600000	600000	600000	600000	100
216	N	600000	600000	600000	0	600000	600000	600000	600000	600000	100
217	N	20000	20000	20000	0	20000	20000	20000	20000	20000	100
218	N	20000	20000	20000	0	20000	20000	20000	20000	20000	100

	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	Trusted_Trans_Prctg	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	Tot_Vhr_Trxn_Am	Hr_Prctg
219	N	30000	30000	30000	0	30000	30000	30000	30000	0	100

Table 5 Data filters or select data by value, Example 1

Example 2:

```
scenario.data = scenario.data[scenario.data$Tot_Dep_Am> 10000 &scenario.data$TRUE_ALERT == 1];
```

	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	Trusted_Trans_Prctg	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	Tot_Vhr_Trxn_Am	Hr_Prctg
1	Y	45600	45600	45600	0	45600	45600	45600	45600	10000	100
2	Y	30000	30000	30000	0	30000	30000	30000	30000	30000	100
3	Y	15000	15000	15000	0	15000	15000	15000	15000	10000	100
4	Y	200000	200000	200000	0	200000	200000	200000	200000	200000	100
5	Y	6000000	6000000	6000000	0	6000000	6000000	6000000	6000000	4800000	100

	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	Trusted_Trans_Prctg	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	Tot_Vhr_Trxn_Am	Hr_Prctg
19	Y	20000	20000	20000	0	20000	20000	20000	20000	20000	100

	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	Trusted_Trans_Prctg	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	Tot_Vhr_Trxn_Am	Hr_Prctg
20	Y	55000	55000	55000	0	55000	55000	55000	55000	0	100
21	Y	50000	50000	50000	0	50000	50000	50000	50000	0	100
22	Y	15000	15000	15000	0	15000	15000	15000	15000	0	100

Table 6 Data filters or select data by value, Example 2

5.3 Transforming Data

In preparing data for analysis, a typical step is to transform data by reformatting it or deriving new columns and adding them to the data set.

The examples in this topic demonstrate two ways of formatting data and deriving columns.

5.3.1 Formatting Data

Example 1: using standard R/ORE functions.

```
head( scenario.data );
```

	N_EVENT_ID	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	TRUSTED_TRANS_PRCTG	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt
1	125105	0	100000	100000	100000	0	100000	100000	100000	
2	125116	0	3002.735	3002.735	3002.735	0	3002.735	3002.735	3002.735	3002.735
3	125126	0	100000	100000	100000	0	100000	100000	100000	100000

	N_EVENT_ID	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	TRUSTED_TRANS_PRCTG	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt
4	125129	0	500	500	500	0	500	500	500	500
5	125137	0	100000	100000	100000	0	100000	100000	100000	100000
6	125154	0	1270	1270	1270	0	1270	1270	1270	1270

Table 7 Formatting Data, Example 1 - 1

```
scenario.data$Tot_Dep_Am_LOG = log( scenario.data$Tot_Dep_Am )
```

	N_EVENT_ID	TRUE_ALERT	Tot_Dep_Am_LOG	Curr_Mnth_Cr_Amt	HRG_Trans_Am	TRUSTED_TRANS_PRCTG	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	TOT_VHR_TRXN_AM
1	125105	0	11.51292546	100000	100000	0	100000	100000	100000	100000	100000
2	125116	0	8.007278819	3002.735	3002.735	0	3002.735	3002.735	3002.735	3002.735	3002.735
3	125126	0	11.51292546	100000	100000	0	100000	100000	100000	100000	100000
4	125129	0	6.214608098	500	500	0	500	500	500	500	500
5	125137	0	11.51292546	100000	100000	0	100000	100000	100000	100000	100000
6	125154	0	7.146772179	1270	1270	0	1270	1270	1270	1270	1270

Table 8 Formatting Data, Example 1 - 2

```
scenario.data$Tot_Dep_Am_LOG10 = log10( scenario.data$Tot_Dep_Am )
```

	N_EVENT_ID	TRUE_ALERT	Tot_Dep_Am_LOG10	Curr_Mnth_Cr_Amt	HRG_Trans_Am	TRUSTED_TRANS_PRCTG	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt	TOT_VHR_TRXN_AM
1	125105	0	5	100000	100000	0	100000	100000	100000	100000	100000
2	125116	0	3.477517006	3002.735	3002.735	0	3002.735	3002.735	3002.735	3002.735	3002.735
3	125126	0	5	100000	100000	0	100000	100000	100000	100000	100000
4	125129	0	2.698970004	500	500	0	500	500	500	500	500
5	125137	0	5	100000	100000	0	100000	100000	100000	100000	100000
6	125154	0	3.103803721	1270	1270	0	1270	1270	1270	1270	1270

Table 9 Formatting Data, Example 1 - 3

> **NOTE** Most of the standard R transformation functions work on ORE frames.

5.4 Sorting

ore.sort()

Provides flexible sorting for **ore.frame** objects.

Example :

```
scenario.data = ore.sort(data = scenario.data,by = "Tot_Dep_Am");
print( head(scenario.data) )
```

	N_EVENT_ID	TRUE_ALERT	Tot_Dep_Am	Curr_Mnth_Cr_Amt	HRG_Trans_Am	TRUSTED_TRANS_PRCTG	Tot_Hr_Trxn_Am	Trans_Amt_Cdt	Wthdrwl_Trans_Amt	Tot_Trans_Amt
125105	125105	0	100000	100000	100000	0	100000	100000	100000	100000
125116	125116	0	3002.735	3002.735	3002.735	0	3002.735	3002.735	3002.735	3002.735
125126	125126	0	100000	100000	100000	0	100000	100000	100000	100000
125129	125129	0	500	500	500	0	500	500	500	500
125137	125137	0	100000	100000	100000	0	100000	100000	100000	100000
125154	125154	0	1270	1270	1270	0	1270	1270	1270	1270

Table 10 Sorting

6 Viewing and Analyzing Results

Results provide representation of the run and is used to analyze the scores. You can analyze the results and submit the run for approval.

The following is the procedure to view and analyze the results:

1. Run the scripts on the *Execution Run Details* window. For more information, see Creating and Editing Execution Runs.
2. Click **Results** to view the results of the run in the *Data Groups* window.

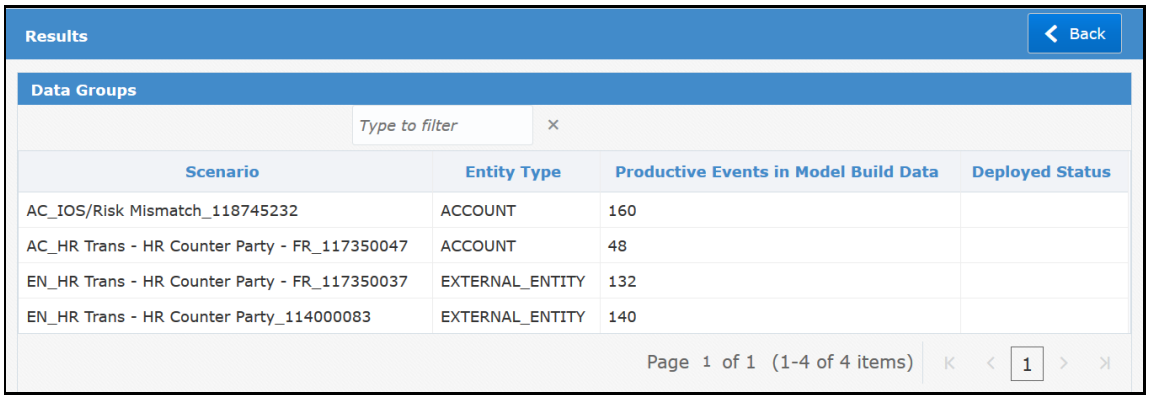

Figure 25 Data Groups Window

3. Select a Scenario row on the *Data Groups* window and the *Execution Results* pane appears at the bottom. This pane displays the Execution Log and the Model Techniques that were selected on the *Execution Run Details* window. Click **Execution Log** to view execution details. The *Execution Results* pane also displays **Model Fit Summary** and **Plots**. Click **Submit For Approval** to deploy the Techniques for approval by a user with Approver privileges. For more information on approval, see Approving Techniques for Data Groups.

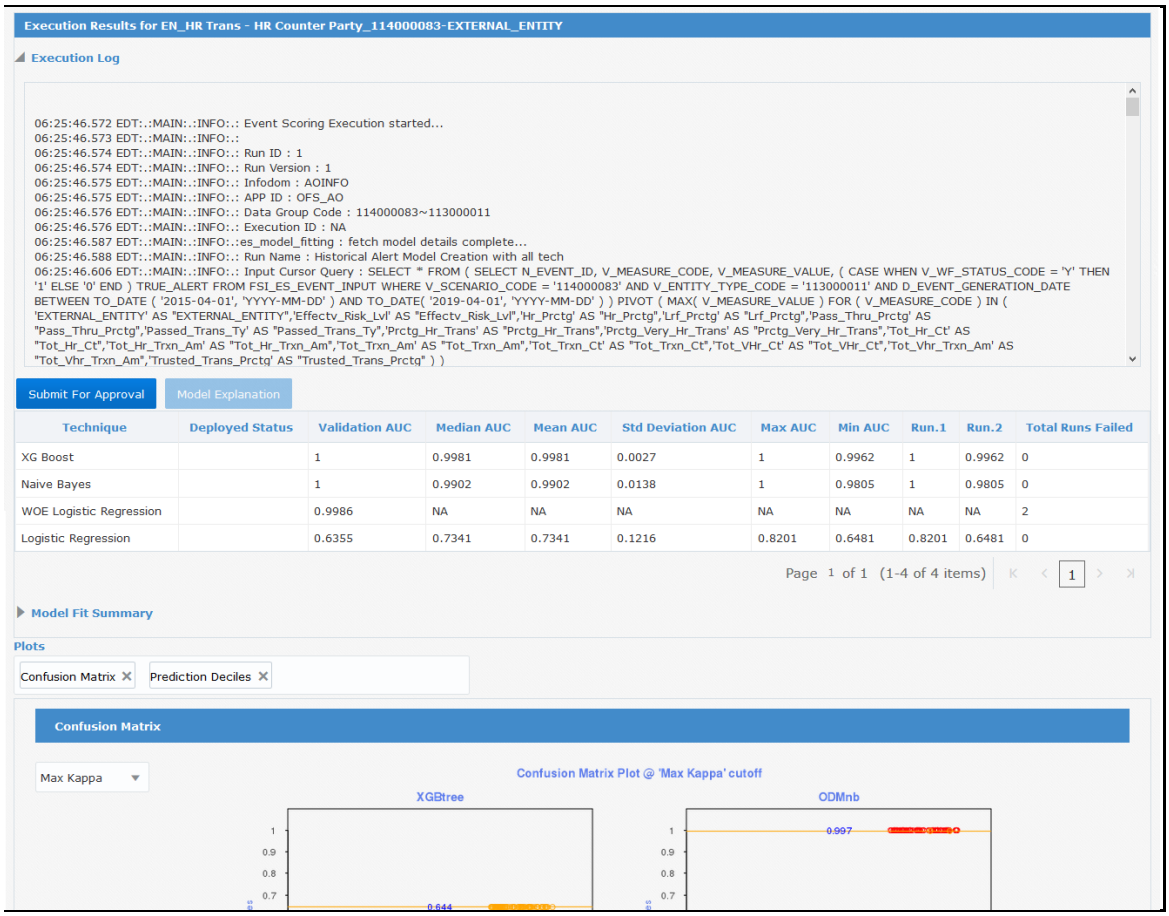

Figure 26 Execution Results Pane

NOTE

1. **Submit for Approval** button is displayed if you are a user assigned the Creator role.
2. **Model Explanation** button is active if you click on a Technique name in Model AUC grid. Model Explanation lets you create views of various combinations of Events and helps in What-If analysis of the risk involved in a transaction.. For detailed information, see Using Model Explanation.

4. Click **Model Fit Summary** to expand and view details for the selected Model.

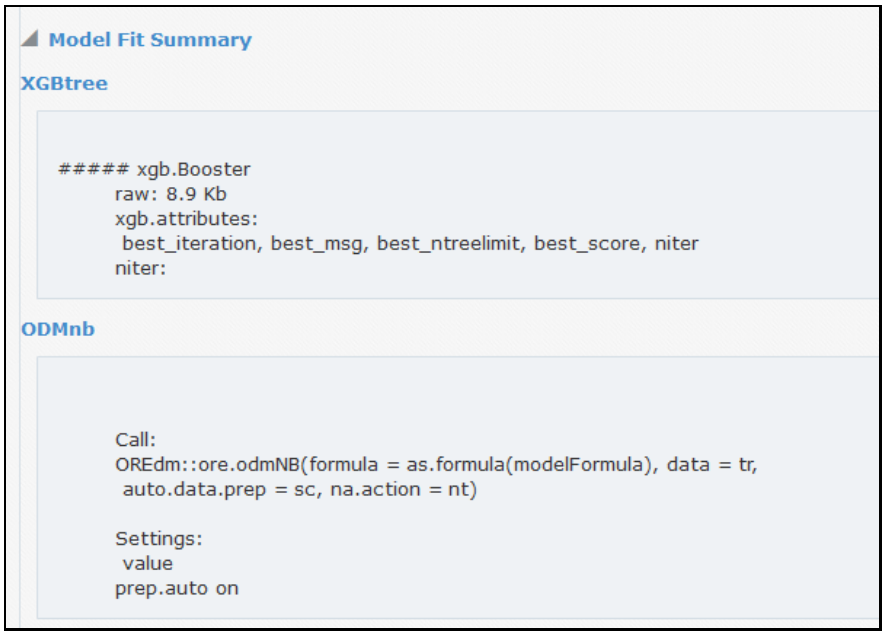

Figure 27 Model Fit Summary

5. **Plots** displays a graphical view of the results of the Techniques used in the Model. You can review and analyze the results of the Techniques here.
   For example, Confusion Matrix can be used to compare alerts with analysis of historical data to determine the fit or match. The framework provides Confusion Matrix for various cut-offs such as Kappa, KS, F Value, BKVN, and High3 till Low1.

Figure 28 Graphical view of the results of the techniques used in the model

6. Click ‹ to view the previous technique and › to view the next Plot. Select the available studies from the drop-down list on Plots and apply on the charts.

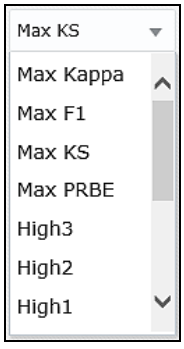

Figure 29 Available Studies List

7. Click the **Plots** field to display the Plots drop-down list and select a Plot to add. To remove a Plot, click **X** on the Plots displayed in the field.
8. Click **Back** to go to the previous window.

7 Using WOE Model Explanation

WOE Model Explanation lets you create model variables that contribute to an event, which helps in What-If analysis of the risk involved in a transaction. This window is available if you have configured WOE Logistic Regression on the *Execution Run Details* window. For steps to access this window, see Viewing and Analyzing Results.

> **NOTE** If you are interested in understanding and analyzing the details behind WOE based Logistic Regression, see OFS AML Event Scoring WOE Logistic Regression Reference Guide.

The WOE Model Explanation window displays the following sections:

- Model Build
- Formula
- Variable Importance
- Measures Omitted
- Customize your score card
- Scorecard Plot
- WOE Variable Plots
- Scorecard (WOE Bins)
- Scorecard (Raw)

For more information on the preceding list, go to the respective subsections.

7.1 Model Build

Model Build displays a summary of the run definition (see Creating and Editing Execution Runs) with additional information such as Validation AUC.

7.2 Formula

Formula displays the standard logistic regression formula, where all the variables are in WOE transformed format.

7.3 Variable Importance

Variable Importance displays a plot that shows the importance for final model variables based on type III test during model training on a scale of 0 to 1 and is normalized to maximum importance = 1. The plot shows a comparative view for each variable.

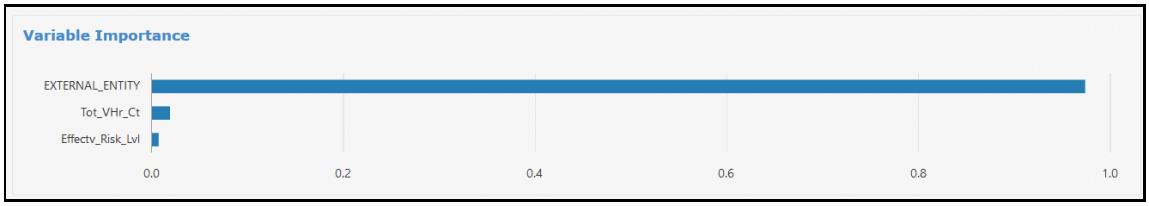

Figure 30 WOE Variable Importance

7.4 Measures Omitted

View the list of Measures that are omitted from the run.

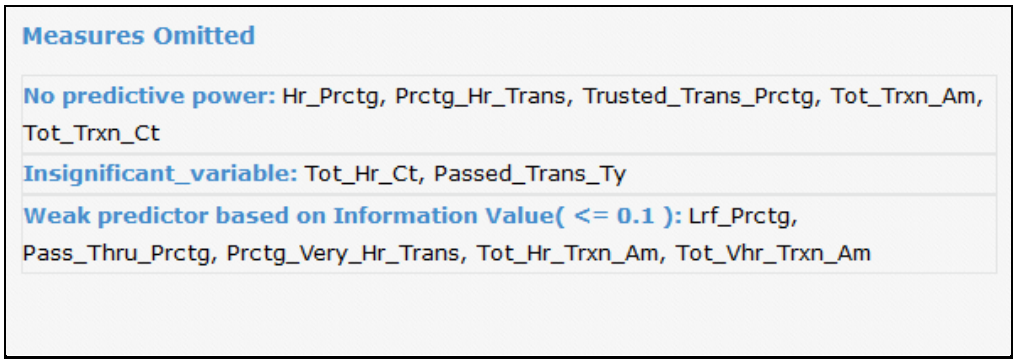

Figure 31 Measures Omitted

7.5 Customize your score card

Custom scorecard provides user with the flexibility of scaling probability scores associated with prediction of productive events.

A Scorecard plot displays the linear relationship between log-odds and the corresponding points.

- Assuming that median odds is 1:100. You want the median points to be 500 and every 50 points increase should result in double odds, your scorecard model would be computed as shown in the following:
  - 500 = α+β*ln(1/100)
  - 550 = α+β*ln(1/50)
- Solve for α and β

Enter values for **Total points for median odds** and **Points to double the odds**. Click **Run** to view the customized WOE score card.

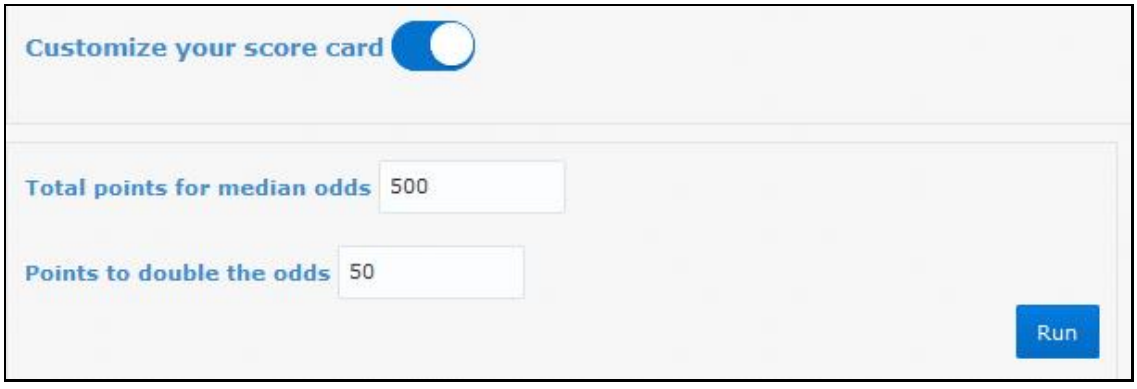

Figure 32 Customize your score card

7.6 Scorecard Plot

Illustration of a Scorecard plot displaying the linear relationship between log-odds and the corresponding points.

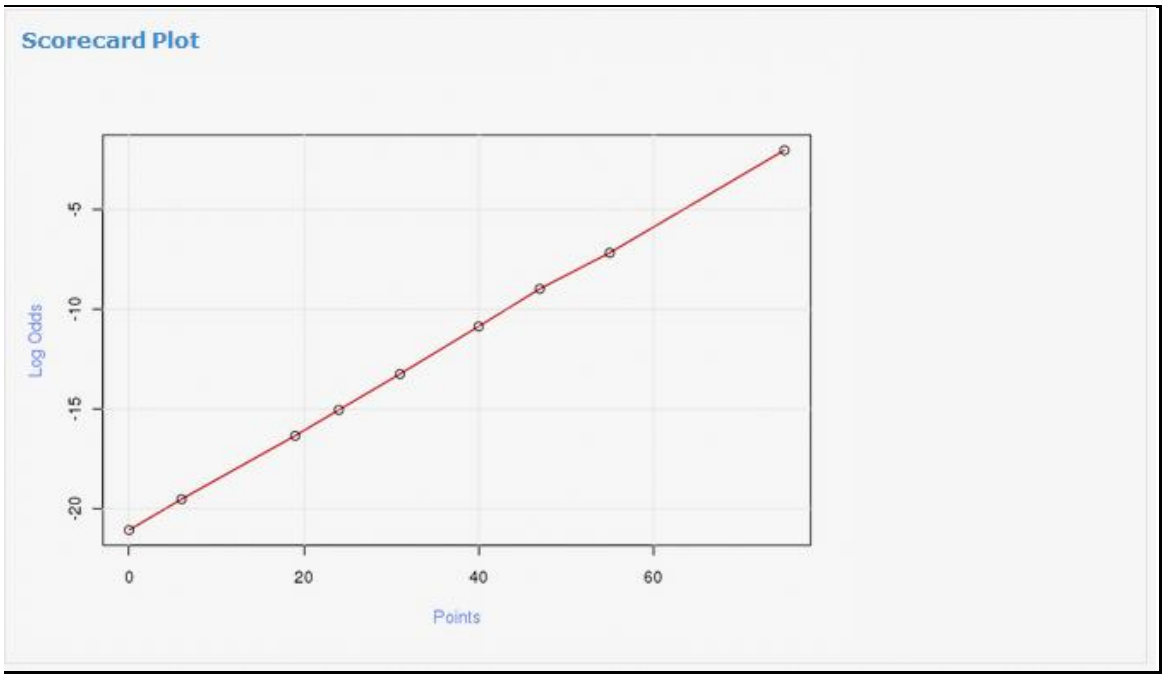

Figure 33 Scorecard Plot

7.7 WOE Variable Plots

WOE Variable Plots displays the WOE Value for Bins that are contained in Variables. The WOE Value is based on how each Bin is contributing to the result. Values are based on distribution of events and non events within the Bin categories of the respective variables.

7.7.1 Information

Bins: Binning is pre-processing technique that is used to group a number of more or less continuous values into a smaller number of "bins". For categorical variables: each bin refers to a distinct value within the variable.

WOE should be interpreted as a log-odds ratio for each Variable after applying controls on all other predictors. The following is the information to interpret the plot:

- WOE greater than 0: odds of event (bad) at the Bin level of Variable exceed the overall odds by a factor of exp (WOE).
- WOE is equal to 0: odds of event (bad) at the Bin level of Variable is at the overall odds.
- WOE less than 0: odds of event (bad) at the Bin level of Variable is below overall odds by a factor of exp (WOE).

Select a Variable from the drop-down list and the Bin scores appear in a plot.

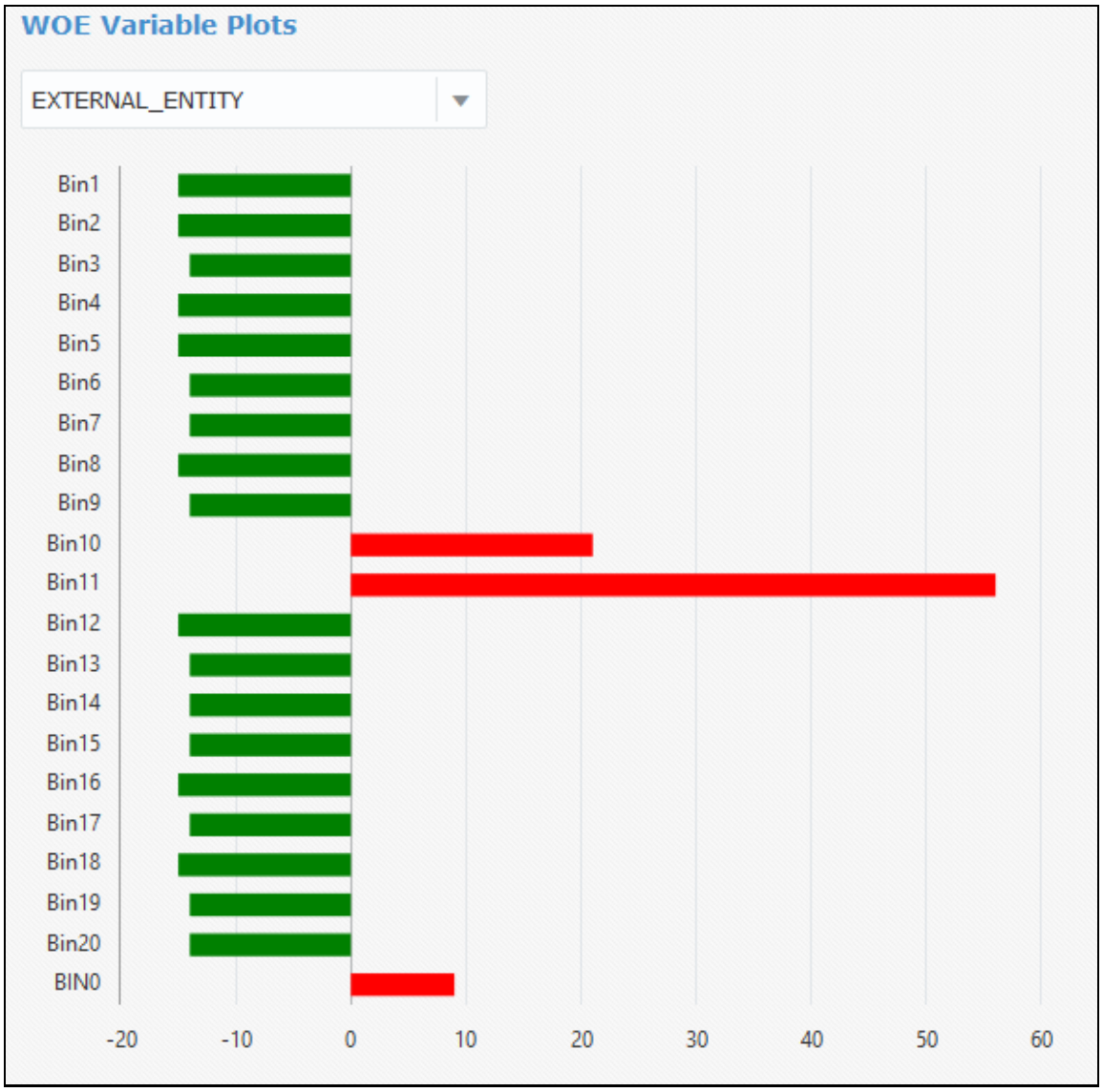

Figure 34 WOE Variable Plots

7.8 Scorecard (WOE Bins)

Scorecard (WOE Bins) provides a summarized view of the total points required for an event to be placed in each tile and the points value per bin for each feature. You can do a What-If analysis based on an existing event chosen by an event id or on an event tile picked up randomly by the application. Click **Explanation with Random Event** for What-If analysis based on a random event for the selected tile. Click **Explanation for an Event** for What-If analysis based on an event ID entered by you.

You can select a Bin (gets highlighted), and move the application of the Bin from the existing one to the selected one, and click **Run**. The result is displayed in **Result Changed**. Compare it with the result in **Result**.

What-if analysis provides insights on how to push an Event from High Risk to Low Risk. In addition, it indicates the range of input values that contribute to the Risk factor. It also provides an opportunity to identify the combination of variables that could push an event from high risk to low risk.

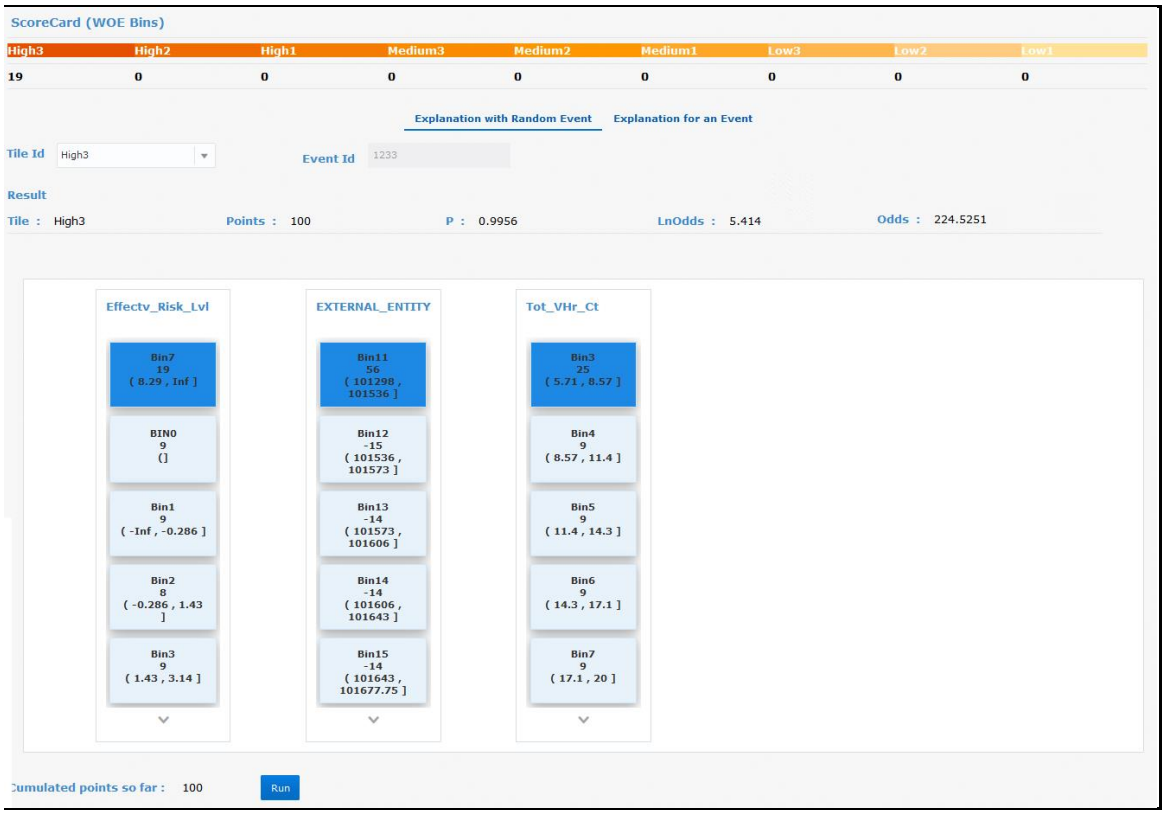

Figure 35 Scorecard (WOE Bins)

7.9 Scorecard (Raw)

Scorecard (Raw) provides What-If analysis based on values configured by you. Event distribution and results is shown for the specified Event ID. You can move the slider bar on each numeric variable and drop down for categorical variable. Click **Run**, the result changed is displayed in **Result Changed**.

What-if analysis provides insights on how to push an Event from High Risk to Low Risk. In addition, it indicates the range of input values that contribute to the Risk factor. It also provides an opportunity to identify the combination of variables that could push an event from high risk to low risk.

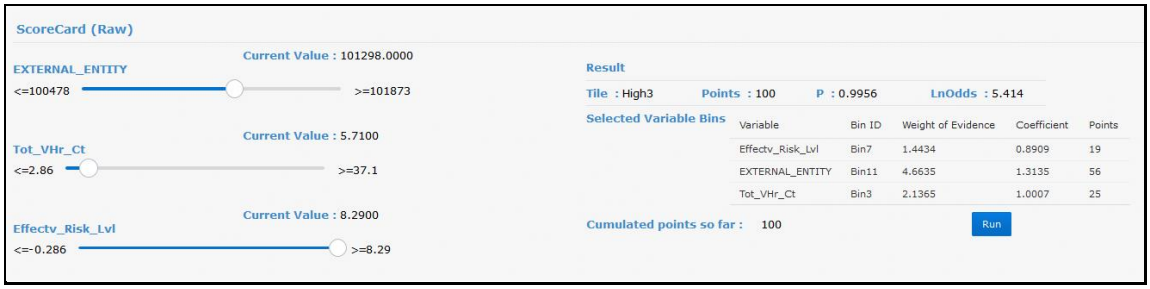

Figure 36 Scorecard (Raw) - Result Changed for Tile is High3

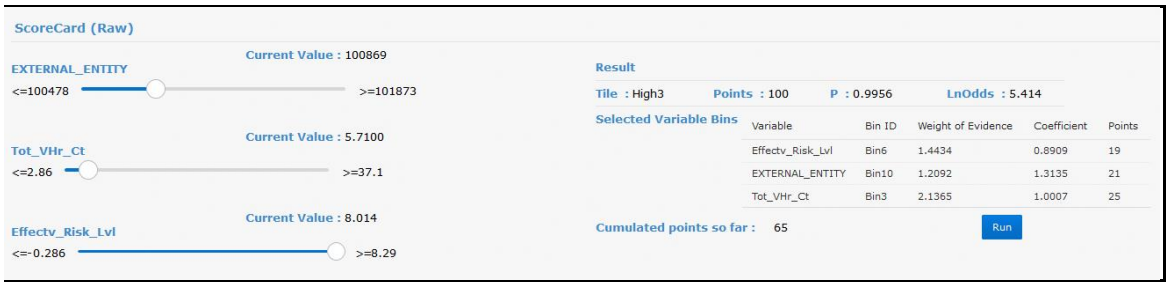

Figure 37 Scorecard (Raw) - Result Changed for Tile is Medium3

8 Using XGBoost Model Explanation

XGBoost Model Explanation lets you create model variables that contribute to an event, which helps in What-If analysis of the risk involved in a transaction. This window is available if you have configured XGBoost on the *Execution Run Details* window. For steps to access this window, see Viewing and Analyzing Results.

The XGBoost Model Explanation window displays the following sections:

- Model Build
- Variable Importance
- Measures Omitted
- Input Variables

For more information on the preceding list, go to the respective subsections.

8.1 Model Build

Model Build displays a summary of the run definition (see Creating and Editing Execution Runs) with additional information such as Validation AUC.

8.2 Variable Importance

Variable Importance displays a plot that shows the importance for top variables based on Gain as computed by XGBoost during model training on a scale of 0 to 1 and is normalized to maximum importance = 1. The plot shows a comparative view for each variable.

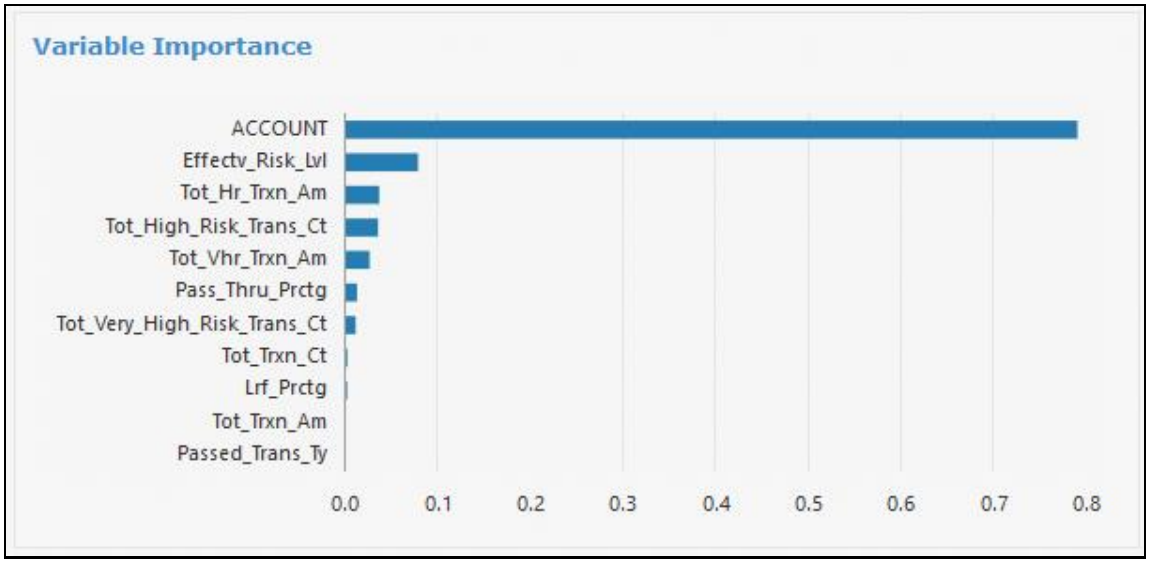

Figure 38 XGBoost Variable Importance

8.3 Measures Omitted

View the list of Measures that are omitted from the run.

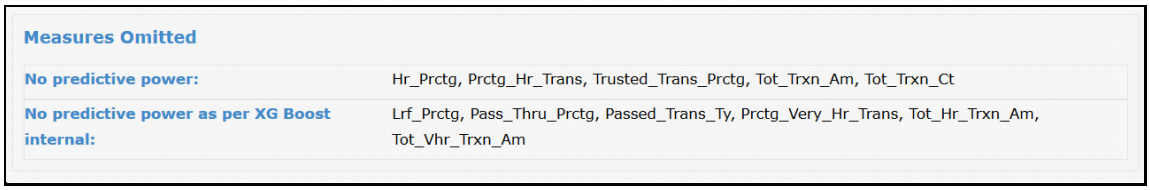

Figure 39 Measures Omitted

8.4 Input Variables

XGBoost Model Explanation provides What-If analysis based on values configured by users. Click **Explanation with Random Event** for What-If analysis based on a random event for the selected tile. Click **Explanation for an Event** for What-If analysis based on an event ID entered by you. You can move the slider bar on each Input Variable and observe the changed values in results. The result is displayed in **Result Changed**.

The following set of illustrations provide an example of the procedure described in the preceding paragraph:

1. Navigate to the *XGBoost Model Explanation* window.

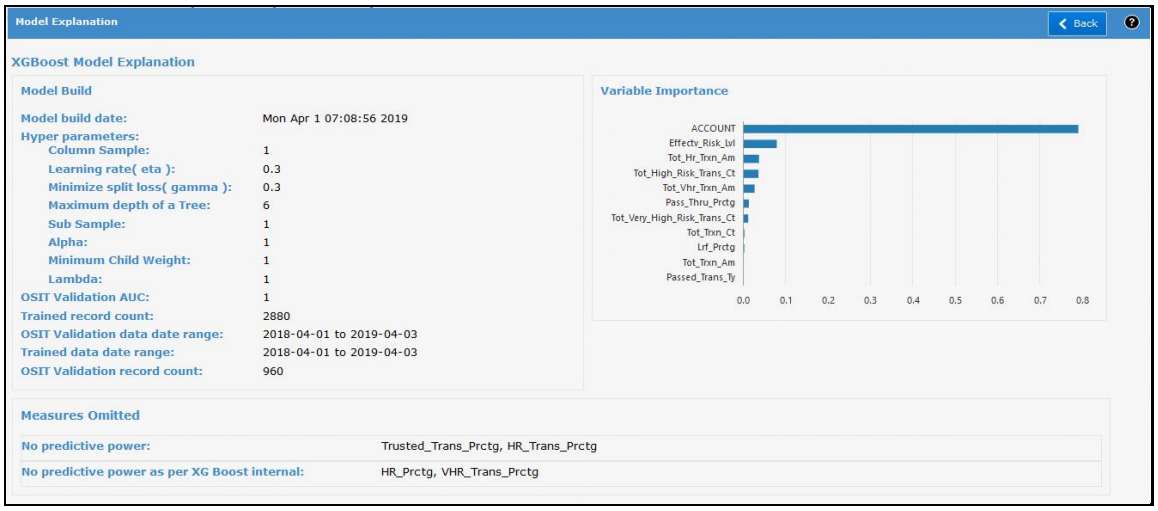

Figure 40 XGBoost Model Explanation

2. Select from **Tile ID**.

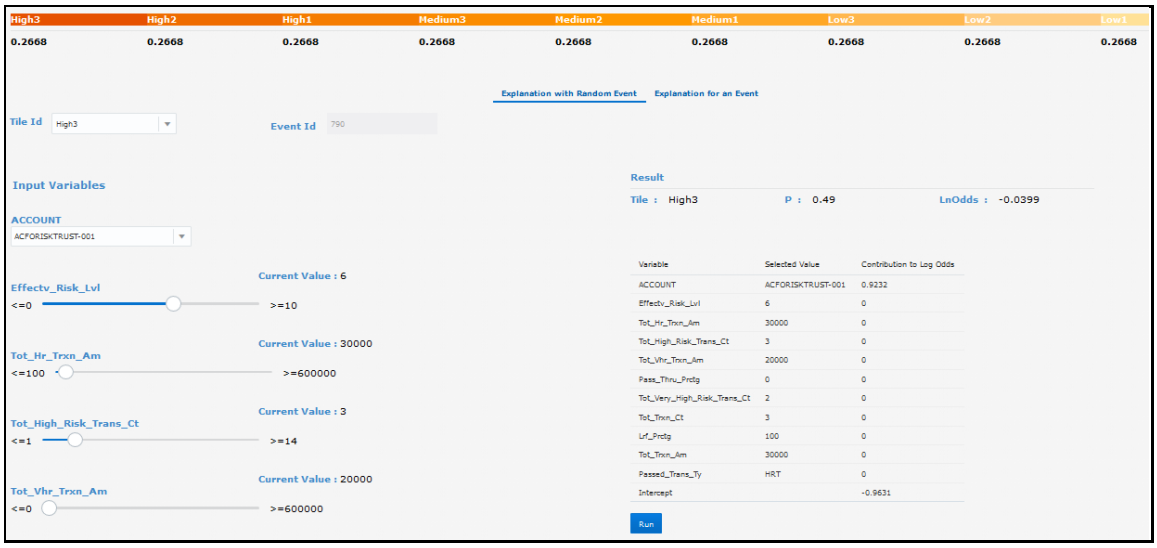

Figure 41 XGBoost Model Explanation Slider

3. Move Sliders to the required values and click **Run**. View the results in **Results Changed**.

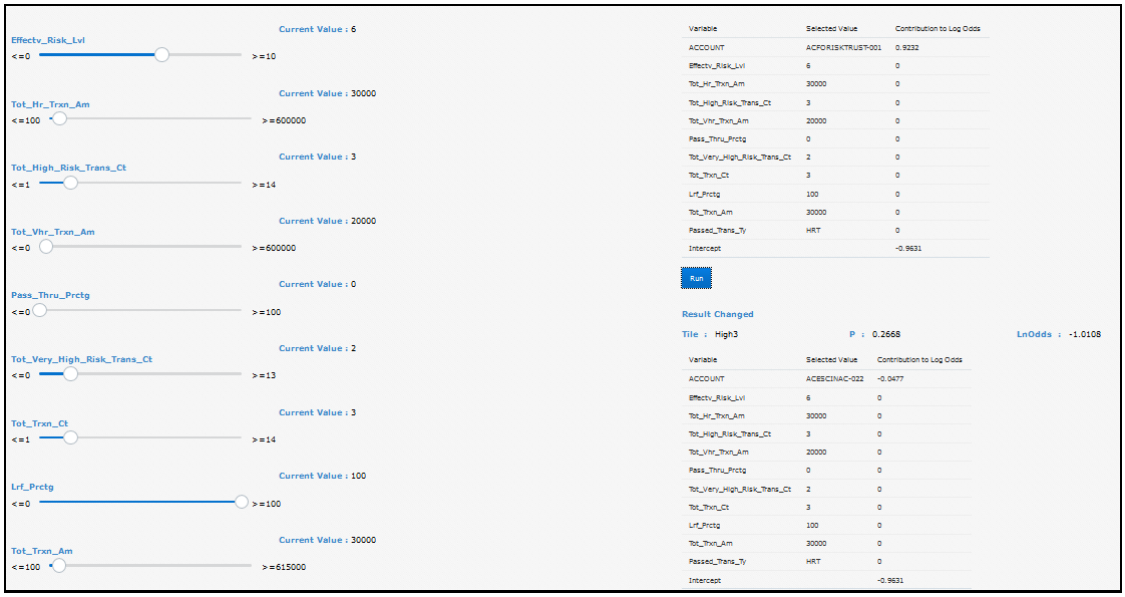

Figure 42 XGBoost model explanation slider run result changed

9 Using Naive Bayes Model Explanation

Naive Bayes Model Explanation lets you create model variables that use probabilities based on Bayes' theorem with the assumption that there is very little dependence between events. This window is available if you have configured Naive Bayes on the *Execution Run Details* window. For steps to access this window, see Viewing and Analyzing Results.

The Naive Bayes Model Explanation window displays the following sections:

- Model Build
- Formula
- Variable Importance
- Conditional Probabilities
- Measures Omitted
- Input Variables

For more information on the preceding list, go to the respective subsections.

9.1 Model Build

Model Build displays a summary of the run definition (see Creating and Editing Execution Runs) with additional information such as Validation AUC.

9.2 Formula

Formula displays the probability formula, where all the variables have Naive Bayes theorem applied.

9.3 Variable Importance

Variable Importance displays a plot that shows the importance for top variables based on the probabilities as computed by Naive Bayes during model training on a scale of 0 to 1 and is normalized to maximum importance = 1. The plot shows a comparative view for each variable.

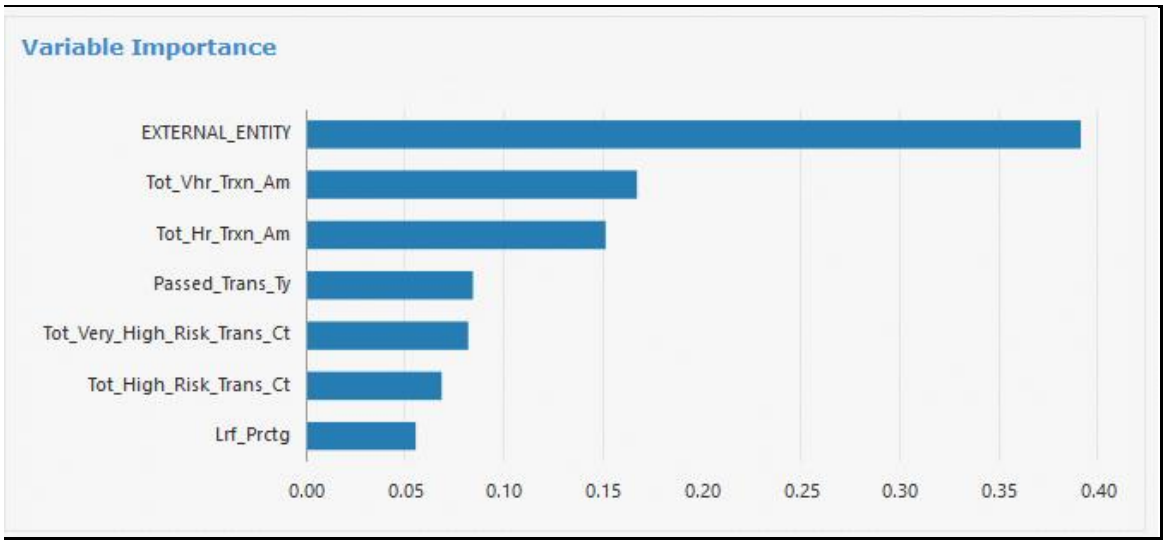

Figure 43 Naive Bayes Variable Importance

9.4 Conditional Probabilities

Bayes' theorem is stated mathematically as shown in the following equation:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A) * P(A)}{P(B)}$$

In case of Event Scoring:

- **A** represents a Productive/Non-productive Event.
- **B** represents an attribute of predictor variable.

In the preceding equation:

- **P(A|B)** is a conditional probability: the likelihood of A is dependent (that is, conditional) on what happened with B.
- **P(B|A)** is also a conditional probability: the likelihood of B is dependent (that is, conditional) on what happened with A. In the model explanation for Naïve-Bayes, each conditional probability table corresponds to a predictor column.
- **P(A)** is called prior probability (Apriori): The prior probability is the estimated probability of a particular class (1-Productive event or 0-Non-productive event) before observing any of the predictors.
- **P(B)** is known as the Marginal likelihood.

Figure 44 Naive Bayes Conditional Probabilities

9.5 Measures Omitted

View the list of Measures that are omitted from the run.

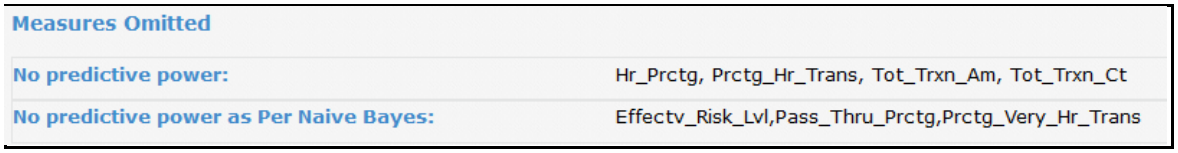

Figure 45 Measures Omitted

9.6 Input Variables

Naive Bayes Model Explanation provides probable scenarios based on values configured by users. Click **Explanation with Random Event** for probabilities based on a random event for the selected tile. Click **Explanation for an Event** for probabilities based on an event ID entered by you. You can move the slider bar on each Input Variable and observe the changed values in results. The result is displayed in **Result Changed**.

The following set of illustrations provide an example of the procedure described in the preceding paragraph:

1. Navigate to the *Naive Bayes Model Explanation* window.

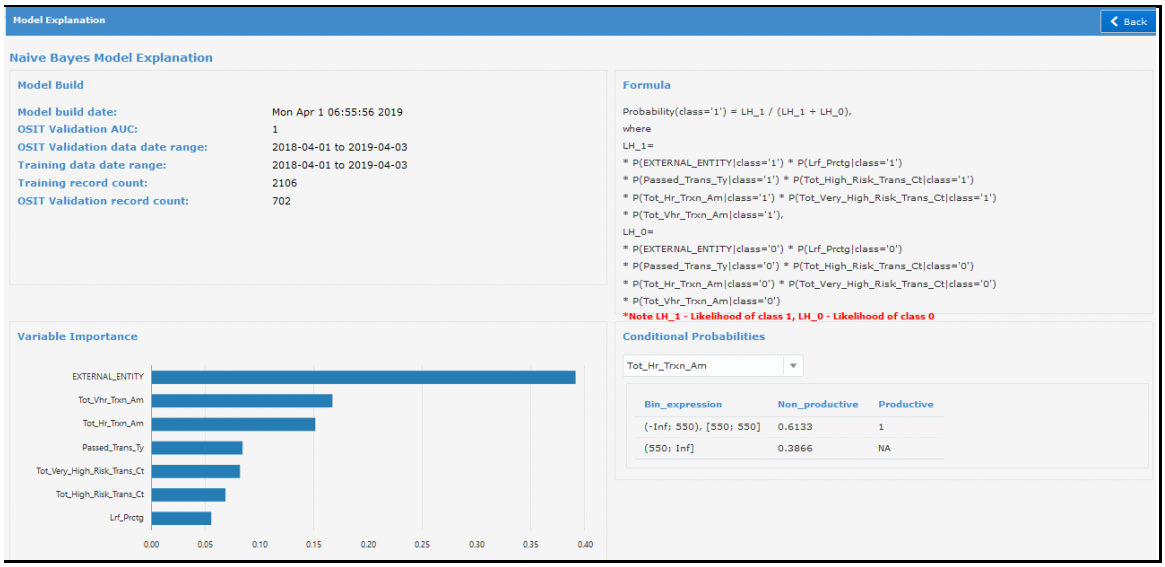

Figure 46 Naive Bayes Model Explanation

2. Select from **Tile ID**.

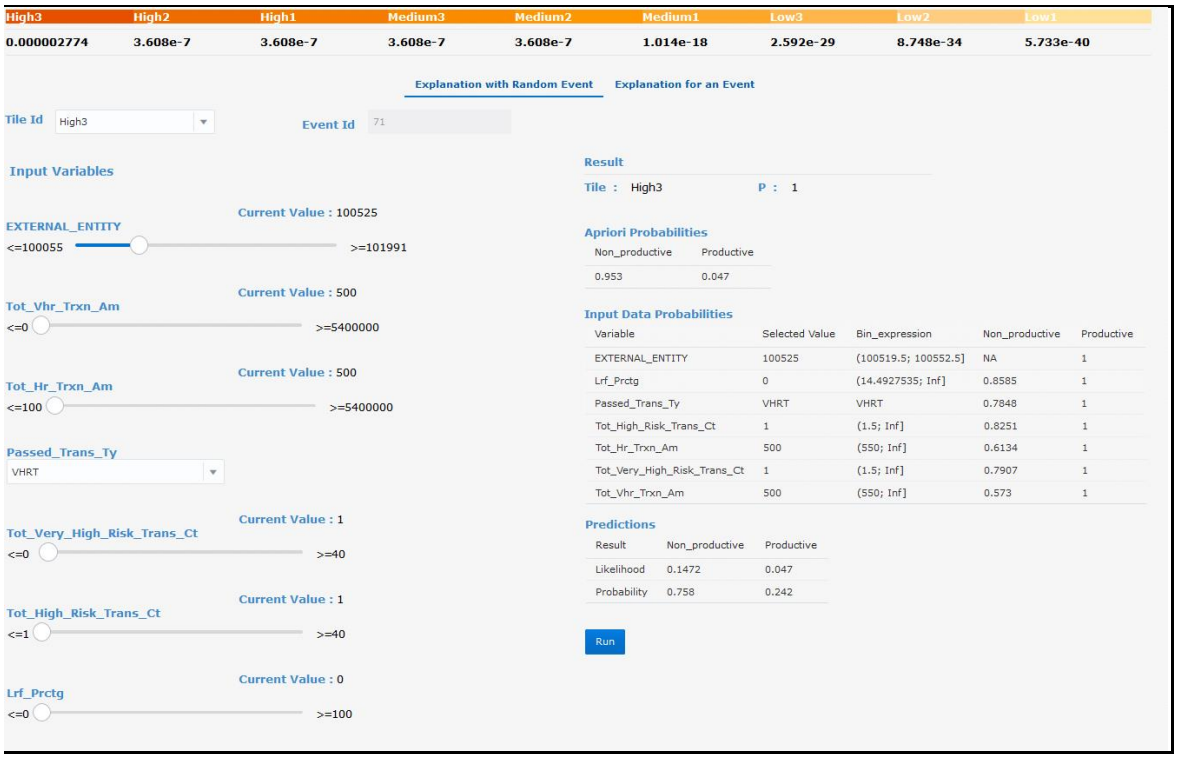

Figure 47 Naive Bayes Model Explanation Slider

3. Move Sliders to the required values and click **Run**. View the results in **Results Changed**.

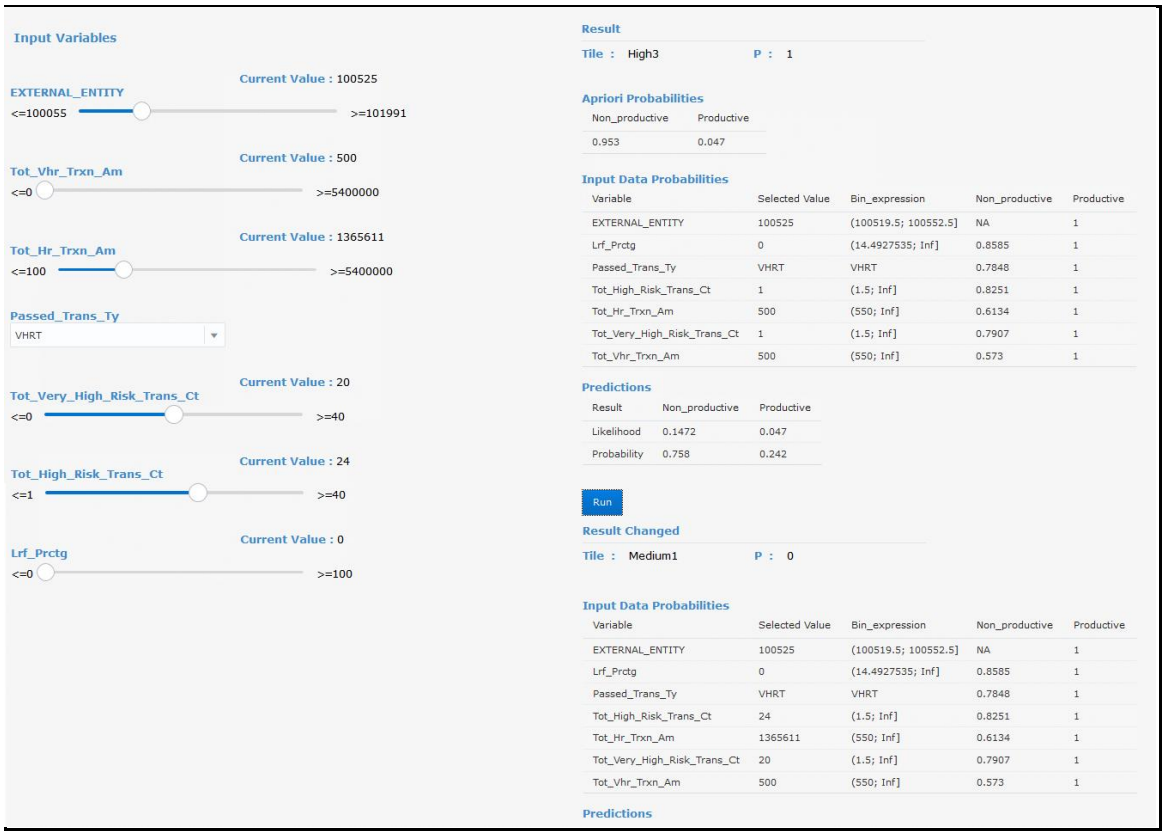

Figure 48 Naive Bayes model explanation slider run result changed

10 Using Logistic Regression Model Explanation

Logistic Regression Model Explanation lets you create model variables in binary format to allow predictive analysis of data. Generalized Linear Models (GLM) Logistic Regression helps create predictive models on continuous data. This window is available if you have configured Logistic Regression on the *Execution Run Details* window. For steps to access this window, see Viewing and Analyzing Results.

The Logistic Regression Model Explanation window displays the following sections:

- Model Build
- Formula
- Variable Importance
- Measures Omitted
- Input Variables

For more information on the preceding list, go to the respective subsections.

10.1 Model Build

Model Build displays a summary of the run definition (see Creating and Editing Execution Runs) with additional information such as Validation AUC.

10.2 Formula

Formula displays the Logistic Regression formula applied on the model.

10.3 Variable Importance

Variable Importance displays a plot that shows the importance for top variables based on the comparison of data computed during model training on a scale of 0 to 1 and is normalized to maximum importance = 1. The plot shows a comparative view for each variable.

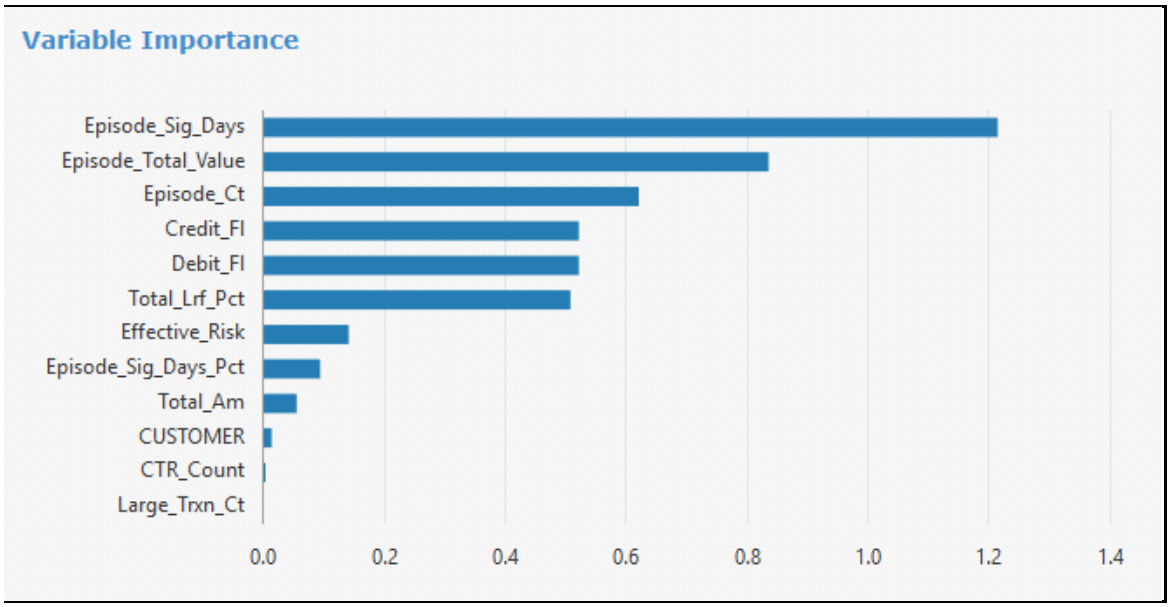

Figure 49 Logistic Regression Variable Importance

10.4 Measures Omitted

View the list of Measures that are omitted from the run.

Figure 50 Measures Omitted

10.5 Input Variables

Logistic Regression Model Explanation provides predictive analysis based on values configured by users. Click **Explanation with Random Event** for analysis based on a random event for the selected tile. Click **Explanation for an Event** for analysis based on an event ID entered by you. You can move the slider bar on each Input Variable and observe the changed values in results. The result is displayed in **Result Changed**.

The following set of illustrations provide an example of the procedure described in the preceding paragraph:

1. Navigate to the *Logistic Regression Model Explanation* window.

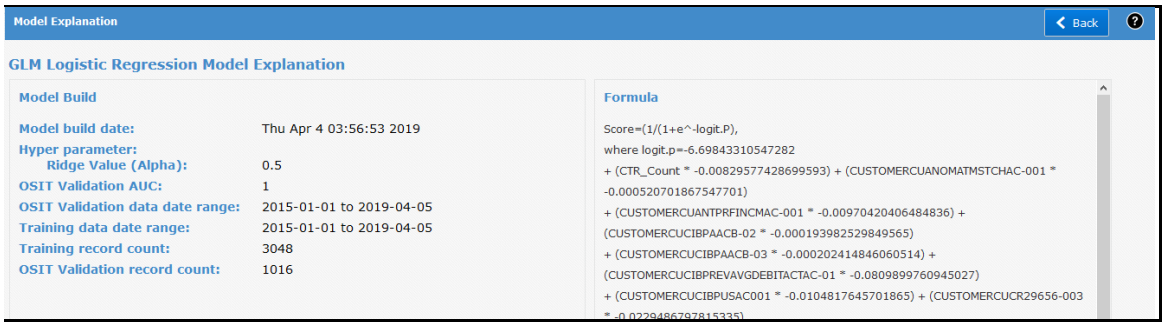

Figure 51 Logistic Regression Model Explanation

2. Select from **Tile ID**.

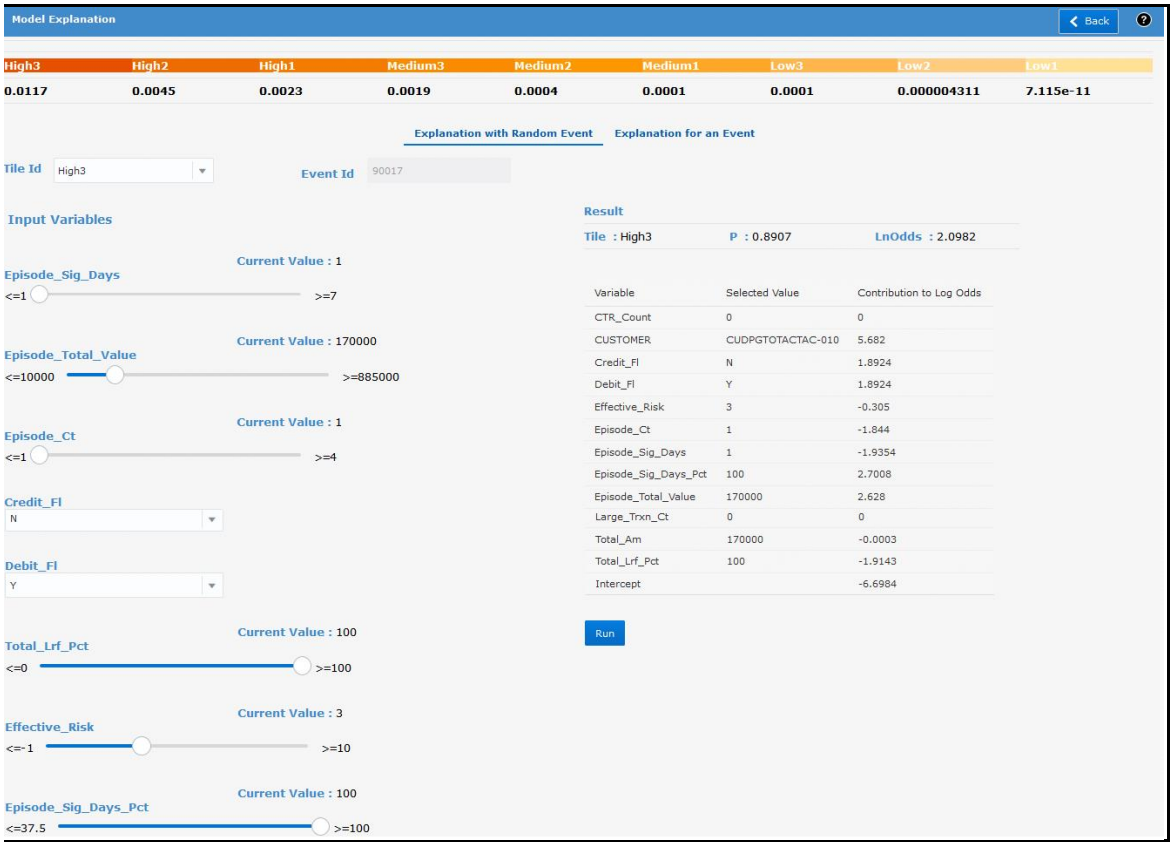

Figure 52 Logistic Regression Model Explanation Slider

3. Move Sliders to the required values and click **Run**. View the results in **Results Changed**.

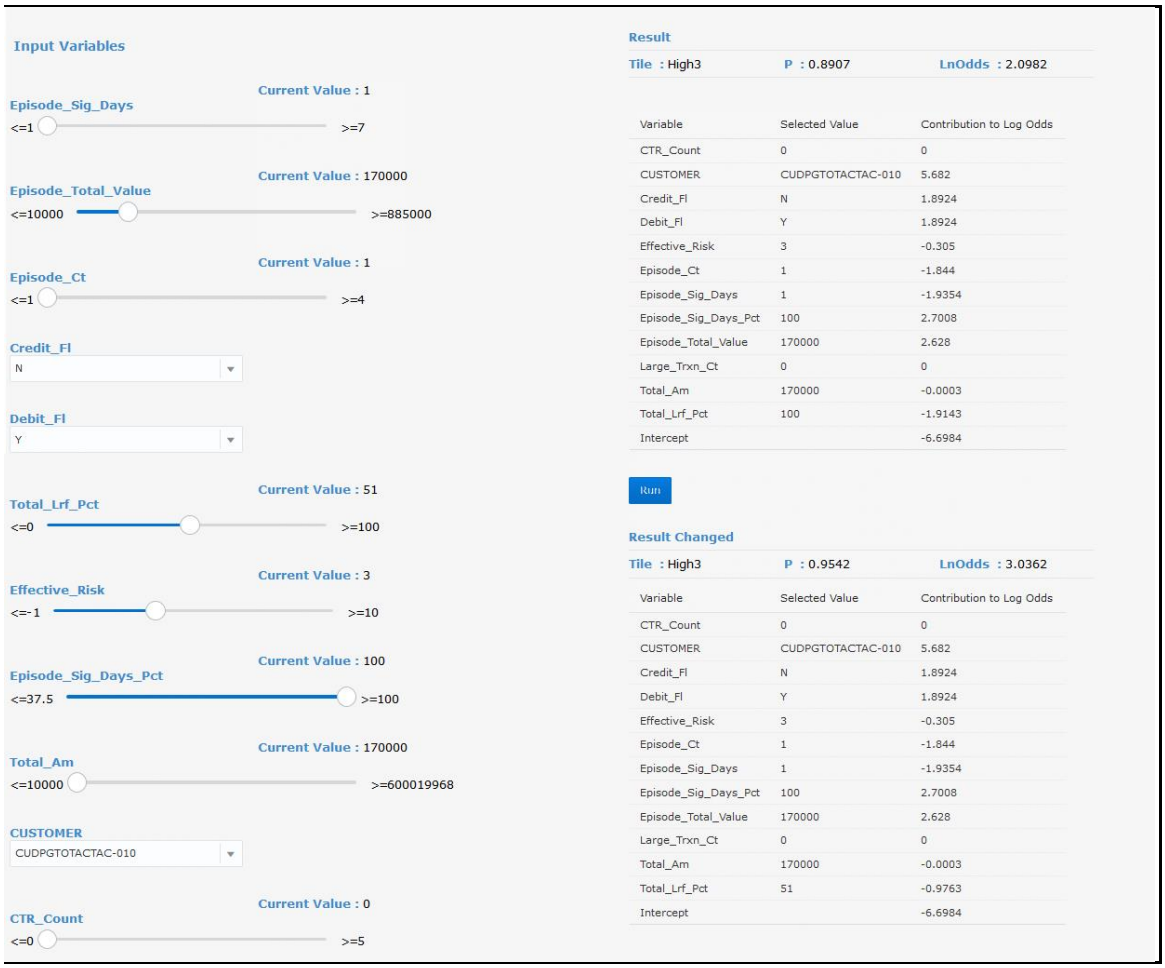

Figure 53 Logistic Regression model explanation slider run result changed

11 Managing Data Groups

Data Groups provides information for the status that a Run is in the workflow. For more information on workflow, see Understanding the OFS Anti Money Laundering Event Scoring Workflow. You can view Runs that are approved, or you can approve Runs if you have Approver Privileges. For more information on how to approve a Run, see Approving Techniques for Data Groups.

The following is the procedure to use the Data Groups user-interface:

1. Click **Data Groups** to view the *Data Groups* window. On the *Data Groups* window, the Runs are grouped in **Approved**, **Pending Approval**, **Rejected** and **Inactive** tabs. The data groups are grouped based on their current status and you can click on the respective tabs to view the data groups.

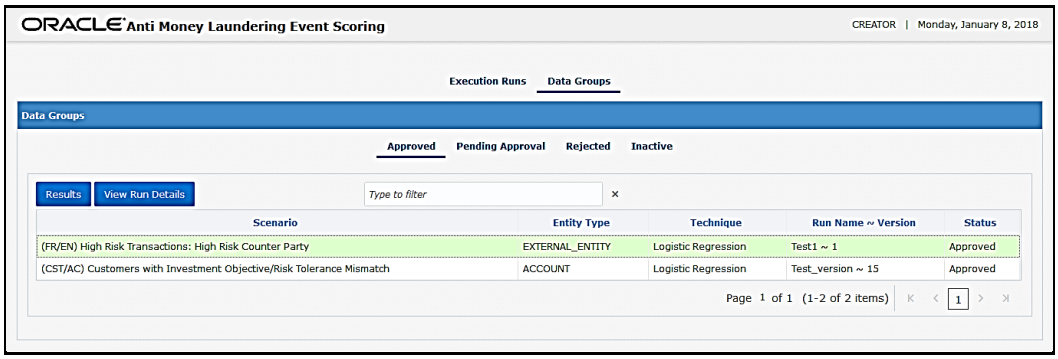

Figure 54 Managing Data Groups

The following is the description for the statuses:

a. **Approved** – Data Groups deployed by approvers and used in the application to predict new alerts.

b. **Pending for Approval** – Data Groups submitted for approval by creators.

c. **Rejected** – Data Groups rejected by approvers. Creators can rerun after applying recommended changes.

d. **Inactive** – Data Groups which have techniques that were replaced by newer and more effective techniques.

2. Select a Data Group and click **Results** to view the execution run results. For more information on Results, see Viewing and Analyzing Results. Select a Data Group and click **View Run Details** to see details of the run. For more information on Runs, see Creating and Editing Execution Runs.

3. Click **Approved** to view a list of approved runs, select a row and click **Results** to view the Technique that is deployed for the Run in the *Deployed Status* column.

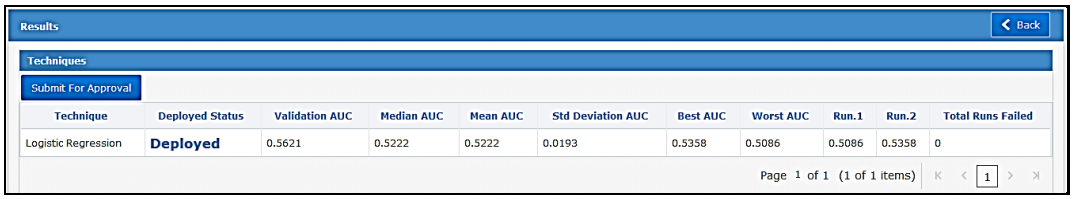

Figure 55 View technique that is deployed for the run

12 Approving Techniques for Data Groups

Users create Execution Runs and submit for approval to users with Approval privileges.

> **NOTE** You must have Approver role privileges assigned to your profile to approve Techniques for Data Groups.

See the following sections for related topics:

- Understanding the OFS Anti Money Laundering Event Scoring Workflow
- Understanding User Roles in OFS Anti Money Laundering Event Scoring
- Managing Data Groups
- Viewing and Analyzing Results

The following is the procedure to approve a Run:

1. Login to OFSAA application. See Accessing OFS Anti Money Laundering Event Scoring for more information.
2. Click **Data Groups** to view the *Data Groups* window. On the *Data Groups* window, the Runs are grouped in **Approved**, **Pending Approval**, **Rejected** and **Inactive** tabs. The Runs are grouped based on the current status and you can click on the respective tabs to view.

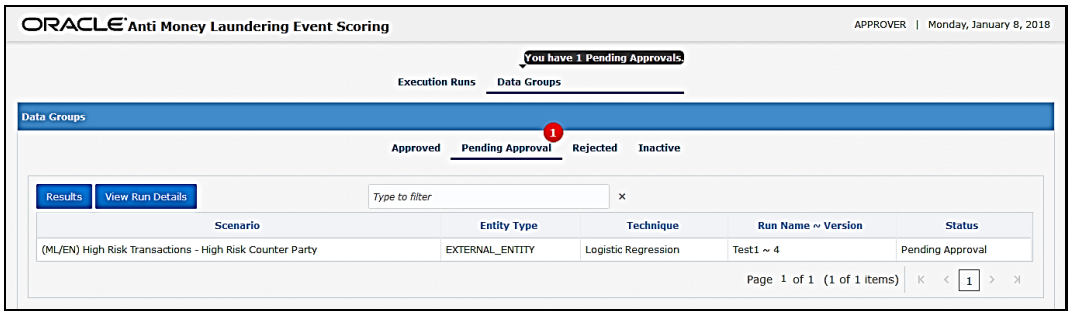

Figure 56 Approving a Run

3. Click **Pending Approval** to view the Runs that are in Pending Approval status. Select a Run/Data Group and click **Results** to view the execution run results. For more information on Results, see Viewing and Analyzing Results. Select a Run/Data Group and click **View Run Details** to see details of the run. For more information on Runs, see Creating and Editing Execution Runs.

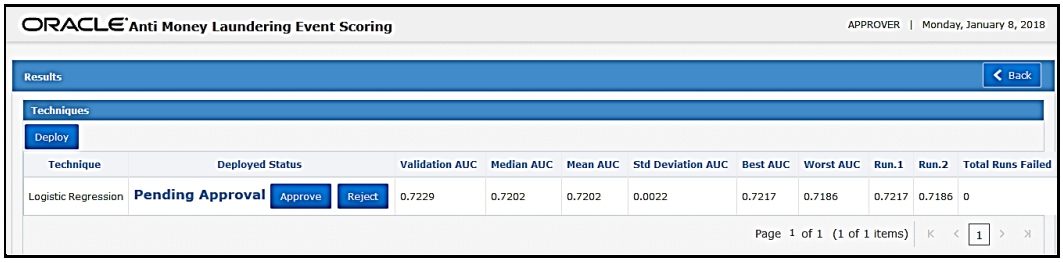

Figure 57 Runs that are in pending approval status

4. On the *Result* window, click **Approve** to approve and deploy a technique. Click **Reject** to reject by providing appropriate comments. The rejected Technique will be available to the creator to modify and resubmit. Click **Deploy** to deploy an approved technique.

13 Scoring Alerts

Score alerts by using alerts data generated from BD application and by posting JSON requests to AMLES. The following subsections provide details for the two scenarios.

13.1 Scoring alerts using AML batches for BD alerts

Score alerts by scoring events in AML batches and access the data from the behavior detection (BD) application. See **Accessing event scoring batches from behavior detection application** section of the OFS AMLES Administration and Configuration Guide 8.0.7.0.0 for more information.

13.2 Scoring newly generated events using REST services

Newly generated events (production data) can be scored by posting the events in JSON format to the Event Scoring application using a REST Client. You can then check the data and analyze for further processing. The subsections in this topic provides information on how to post data in JSON format using HTTP requests and receive responses from the server.

13.2.1 Prerequisites

The following are the prerequisites to post JSON requests in AMLES:

1. Install a REST client on your local computer.
2. User access for authentication on the server. Users must have the required permissions to post requests to the server with Approver or Creator roles.
3. Knowledge about posting JSON requests.
4. Execute and deploy techniques for specific datagroups (Scenario-Entity) from the Event Scoring Application.

13.2.2 Using POST service to send JSON requests

Send the data in JSON format to the server using the POST Service to score the events. You can then use the POST service message to view the result using the GET service.

The following is the procedure to send JSON requests for POST services:

1. Open the JSON client.
2. Select or enter Method as **POST**.
3. Enter the server URL in **Request URL**. To access the Post EVENT Service, use the following URL:

   ```
   <PROTOCOL>:/<HOSTNAME>:<PORT>/<CONTEXT>/rest-api/aorest/AlertOptimizationRequestService/locale/en_US
   ```

4. Enter the Header parameters as shown in the following table:

No.	Header Name	Header Value	Description
1	username	Enter the user name to login to the server.	This value is used for user authentication.

No.	Header Name	Header Value	Description
2	password	Enter the password to login to the server.	This value is used for user authentication.
3	content-type	Select or enter application/json.	This value denotes that the data in JSON format.
4	accept	Select or enter application/json.	This value denotes that the data in JSON format

Table 11 JSON Request Header Parameters

5. Select **application/json** for **Body content type**.
6. Enter the JSON message in the body. The following code sample is an example:

```
{
  "alertDetails":[
{
   "V_ENTITY_TYPE": "<ENTITY_NAME>",
   "D_EVENT_GENERATION_DATE": "<DATE_OF_ALERT_GENERATION>",
   "V_SCENARIO_NAME": "<SCENARIO_NAME>",
   "N_EVENT_ID": <ALERT_ID/EVENT_ID>,
   "V_MEASURE_VALUE": "<BINDING_VALUE>",
   "V_MEASURE_NAME": "<BINDING_NAME>",
   "V_SEGMENT_NAME": "<ANY_VALID_SEGMENTATION_EXAMPLE_KDD_JRSDCN>"
}
]
}
```

> **NOTE** You can post multiple events in a JSON request. The previous code snippet is applicable for a one-event one-measure. The events and all its associated measures should be part of the same JSON.
> You must use the uppercase format for the keys (V_ENTITY_TYPE, D_EVENT_GENERATION_DATE and so on) shown in the previous example.

The following is an example for the JSON in BD:

```
{
  "alertDetails":[
{
   "V_ENTITY_TYPE": "<ENTITY_NAME>",
   "D_EVENT_GENERATION_DATE": "<ALERT_CREATE_DATE_FROM_KDD_REVIEW>",
   "V_SCENARIO_NAME": "<SCENARIO_NAME>",
```

```
        "N_EVENT_ID": <ALERT_ID_FROM_KDD_REVIEW>,
        "V_MEASURE_VALUE": "<BINDING_VALUE_FROM_KDD_BREAK_BINDING>",
        "V_MEASURE_NAME": "<BINDING_NAME_FROM_KDD_BREAK_BINDING>",
        "V_SEGMENT_NAME": "<ANY_VALID_SEGMENTATION_EXAMPLE_KDD_JRSDCN>"
    }
    ]
    }
```

13.2.2.1 Example for one event with all its measures

```
 {
  "alertDetails":[
{
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1000,
   "V_MEASURE_VALUE": "1",
   "V_MEASURE_NAME": "MEASURE1",
   "V_SEGMENT_NAME": "AMEA"
},
{
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1000,
   "V_MEASURE_VALUE": "ABC1",
   "V_MEASURE_NAME": "MEASURE2",
   "V_SEGMENT_NAME": "AMEA"
},
{
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1000,
   "V_MEASURE_VALUE": "ABC",
   "V_MEASURE_NAME": "MEASURE3",
   "V_SEGMENT_NAME": "AMEA"
}
```

```
]
}
```

13.2.2.2 Example for two events with all their measures

```
{
  "alertDetails":[
{
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1000,
   "V_MEASURE_VALUE": "1",
   "V_MEASURE_NAME": "MEASURE1",
   "V_SEGMENT_NAME": "AMEA"
},
{
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1000,
   "V_MEASURE_VALUE": "ABC1",
   "V_MEASURE_NAME": "MEASURE2",
   "V_SEGMENT_NAME": "AMEA"
},
{
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1000,
   "V_MEASURE_VALUE": "ABC",
   "V_MEASURE_NAME": "MEASURE3",
   "V_SEGMENT_NAME": "AMEA"
},
  {
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1001,
```

```
   "V_MEASURE_VALUE": "2",
   "V_MEASURE_NAME": "MEASURE1",
   "V_SEGMENT_NAME": "AMEA"
},
{
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1001,
   "V_MEASURE_VALUE": "ABC2",
   "V_MEASURE_NAME": "MEASURE2",
   "V_SEGMENT_NAME": "AMEA"
},
{
   "V_ENTITY_TYPE": "ACCOUNT",
   "D_EVENT_GENERATION_DATE": "07-May-18",
   "V_SCENARIO_NAME": "SCENARIO1",
   "N_EVENT_ID": 1001,
   "V_MEASURE_VALUE": "DEF",
   "V_MEASURE_NAME": "MEASURE3",
   "V_SEGMENT_NAME": "AMEA"
}
]
}
```

After you post the event, the POST service returns the following message:

```
Model Execution Triggered for request ID 1.Please Check after some time for response
```

For more information, see Using GET service to view results.

13.2.3 Using GET service to view results

Use the GET service to view the result. After posting the event, the POST service returns a message, which shows details for successful events and errors, if any.

The following step describes the procedure to view results:

Use the GET service after the posted event is scored, to obtain the result. To get the scored EVENT data, use the following URL:

```
<PROTOCOL>:/<HOSTNAME>:<PORT>/<CONTEXT>/rest-api/aorest/AlertOptimizationResponseService/locale/en_US/requestId/<Request_ID>
```

Where <Request_ID> is the Request ID returned by the POST service.

The format of the GET response is shown in the following code snippet:

```
[{
                "DATAGROUPING": "Scenario and Entity Type",
                "DATAGROUPS": [{
                                "STATUS": "SUCCESS",
                                "EVENTS": {
                                                "failed": {
                                                                "Events
omitted due to R prediction behaviour": ""
                                                },
                                                "successful":
"[{\"N_EVENT_ID\":21406,\"SCORE\":0.0073,\"BUCKET\":\"Medium3\"},{\"N_E
VENT_ID\":21399,\"SCORE\":0.0019,\"BUCKET\":\"Medium2\"}]"
                                },
                                "DATAGROUP": "CIB - Product Utilization
~ ACCOUNT"
                }]
}, {
                "DATAGROUPING": "DATA GROUP NOT DEPLOYED",
                "DATAGROUPS": [{
                                "STATUS": "FAILED",
                                "ERROR_MESSAGE":
"{\"status\":\"FAILED\",\"contents\":\" error while getting the details
for this data group... \\n Data group details for '110000085~AC' not
found, check whether the data group is deployed ? \"}",
                                "DATAGROUP": "NO DATA GROUP"
                }
}]
```

The previous response is an example for a successful event and a failed event. The error message means that Datagroup 110000085~AC was not deployed and the corresponding event was posted.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.