Paper SAS2604 - 2015

Take a Bite Out of Crime with SAS® Visual Scenario Designer

Yue Qi, SAS Institute Inc.

ABSTRACT

The vast and increasing demands of fraud detection and description have promoted the broad application of statistics and machine learning in fields as diverse as banking, credit card application and usage, insurance claims, trader surveillance, health care claims, and government funding and allowance management. SAS® Visual Scenario Designer enables you to derive interactive business rules, along with descriptive and predictive models, to detect and describe fraud and money laundering. This paper focuses on building interactive decision trees to classify fraud. Attention to optimizing the feature space (candidate predictors) prior to modeling is also covered. Because big data plays an increasingly vital role in fraud detection and description, SAS Visual Scenario Designer leverages the in-memory, parallel, and distributed computing abilities of a SAS in-memory server as a back end to support real-time performance on massive amounts of data.

INTRODUCTION

With the power of SAS Visual Scenario Designer, you can perform fraud detection work end-to-end, including data preparation, rule creation, and rule deployment in real-time, using SAS® Event Stream Processing Engine, or quick batch mode using a SAS in-memory server. It has an interactive graphical user interface that requires no programming knowledge. The workflow logic for SAS Visual Scenario Designer is as follows:

1. Create a Window - A window is a data table that is the result of one or more data preparation operations. These operations can include joining tables, aggregating by a group-by column or columns, aggregating by a time variable, filtering at the column or window level, and investigating specific variables.
2. Create a Scenario – A scenario is a set of user activities that might indicate the occurrence of fraud or money laundering. In this step, you can generate a set of if-then rules, and choose whether to derive the rules automatically from a decision tree. Each rule states that if a transaction satisfies some condition, then a binary fraud alert indicator or fraud alert (or compliance) score is added to the transaction. A rule set is also called a scenario.
3. Create a Deployment – A deployment is the grouping of the windows, and scenario or scenarios in real-time or in batch for execution. After a deployment is created, you can tune the scenarios, and enable or disable specific sets of rules, change the thresholds for the rules, and run simulations to update results quickly. The simulation history is automatically saved to help you track your changes and the corresponding changes to the simulation results. After you are fully satisfied with the rules and the thresholds, you can include them into a production schedule either in quick batch mode or in a real-time event streaming engine, such as SAS® Event Stream Processing Engine.

Now that you understand the big picture of how SAS Visual Scenario Designer works, you can dig deeper into the product (Figure 1). You can analyze big data in your browser to discover potentially fraudulent transactions or non-compliant entities from millions, or even billions, of transactions. Because SAS Visual Scenario Designer is a web-based application, you do not need to install anything on your office desktop, work laptop, or client machine. All you need is an application URL, a username, and a password. Then, you can start working to catch the fraudsters and defend the interests of your group.

The three blue icons on the first line are the main application tasks of SAS Visual Scenario Designer: Create a Window, Create a Scenario, and Create a Deployment. The remaining tasks on the application HUB are as follows:

1. Import Content – SAS Visual Scenario Designer uses to save your models, including windows, scenarios, and deployments. It can also import XML files to leverage the models previously saved.
2. Run a simulation – Run simulations more conveniently.
3. Manage Contents – Manage everything in SAS Visual Scenario Designer from this location, including all windows, scenarios, and deployments.

You can also open recent windows, scenarios, and deployments from beneath these icons.

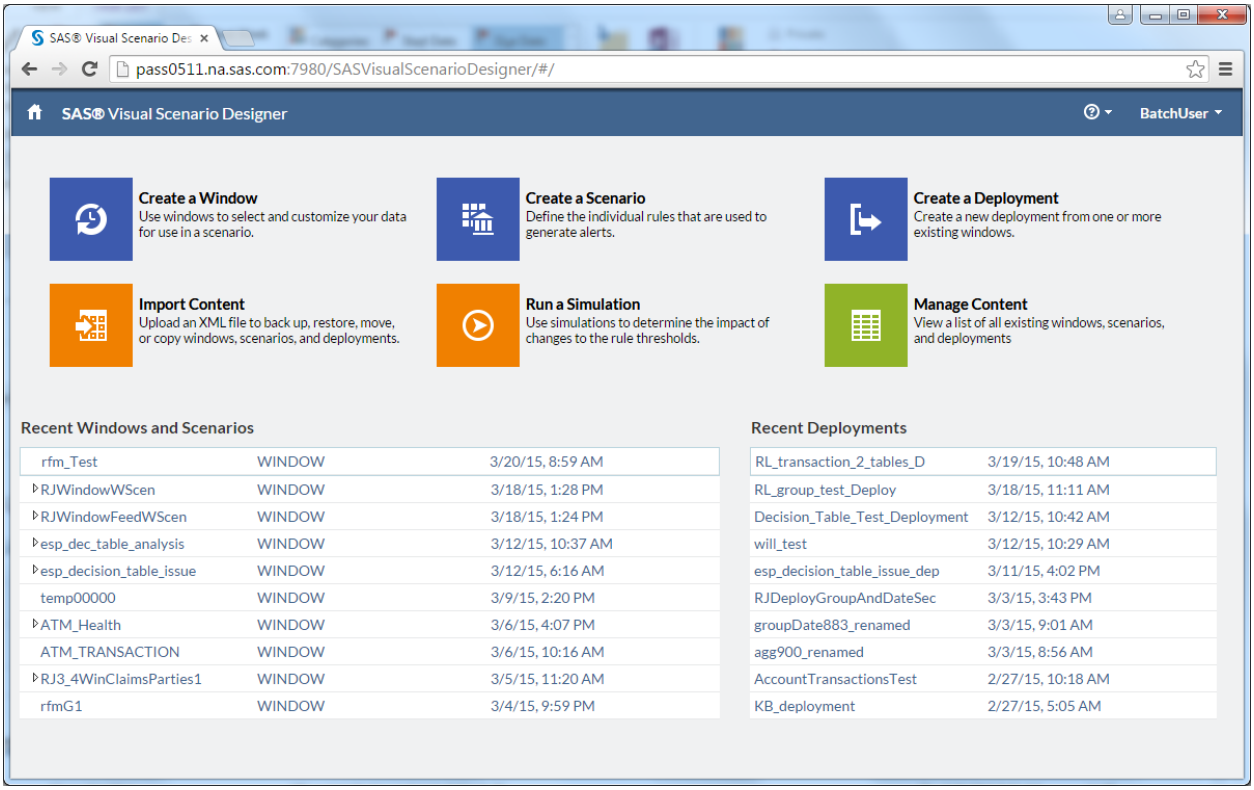

Figure 1. Main Window of SAS Visual Scenario Designer

DATA DESCRIPTION

The data used in this paper is that of fraudulent automobile insurance claims. The initial data was actual data from an insurance company, however, all customer-related information has been removed or anonymized. There are three tables in the analysis: the policy information table (SGF_CLAIM_POLICY), the vehicle information table (SGF_CLAIM_PAYMENT), and the payment information table (SGF_CLAIM_VEHICLE). The policy information table is the main table in the analysis, because it carries most of the information. This information includes a fraud flag that represents whether a claim is fraudulent (coded as 1) or legitimate (coded as 0). The payment information table and vehicle information table are joined to the policy information table because they also provide helpful information.

Here is a brief description of each table:

- Policy – This file contains the insurance policy information for the claims. Of course, there can be multiple claims on the same policy. Each row represents a single claim. There are 44 columns, including fraud flag, policy id, claim id, policy effective dates, incident date and time, claim description, total annual premium, etc.
- Vehicle – This file contains the detailed information of the vehicles insured in the claims. Some vehicles are claimed multiple times. Each row represents a single claim. There are ten columns in the table, including vehicle registration number, claim id, manufacturer, model, vehicle identification

number, manufacture year, damage description, damage location, damage amount, and direction of travel.

- Payment – This file contains the detailed payment information for the claims. There are eight columns in the table, including claim ID, claim revision date, payment date, payment amount, bank code, account number, payee name, and employee clerk number.

CREATE A WINDOW

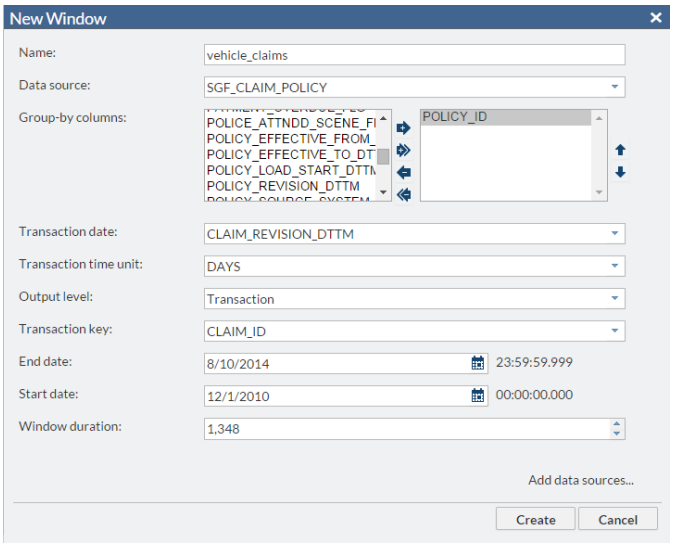

Figure 2. Create a New Window that Aggregates on the Policy Level

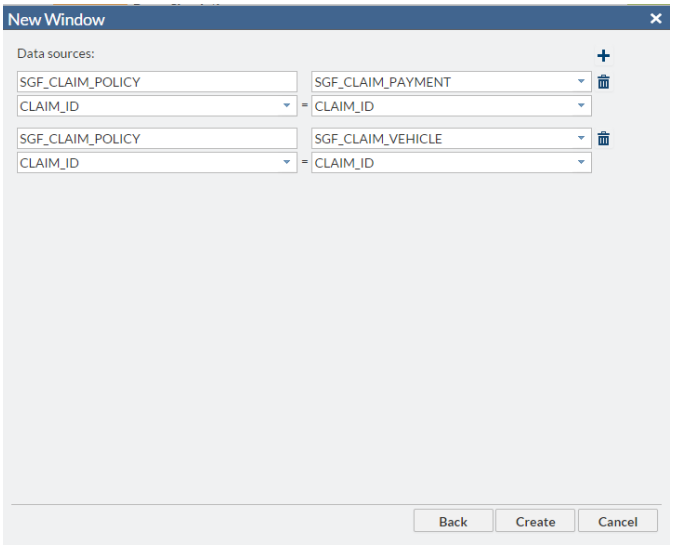

Figure 3. Join Policy, Payment, and Vehicle Tables using CLAIM_ID as the Transaction Key.

The policy information table is aggregated on the policy level, as shown in Figure 2, where POLICY_ID is used as the group-by column. SAS Visual Scenario Designer supports multiple group-by columns, though only one group-by column is needed in this case.

Because each table has different information about the claims, it is essential to join the tables to get all of the information necessary for fraud detection. Figure 3 shows how conveniently SAS Visual Scenario

Designer can create a join of the payment information table and the vehicle information table to the policy table using CLAIM_ID as the join key. Additionally, SAS Visual Scenario Designer can generate a relationships diagram, which shows the table joining structure, as shown in Figure 4. You can see that the payment table is coded as t1 in Figure 5. Its variables are tagged with prefix t1. Similarly, the vehicle table is coded as t2.

You can also change the default aggregation function and look-back period in the Properties window (Figure 5). When you drag and drop the columns to add them to the window, the default aggregation function and look-back period are automatically applied.

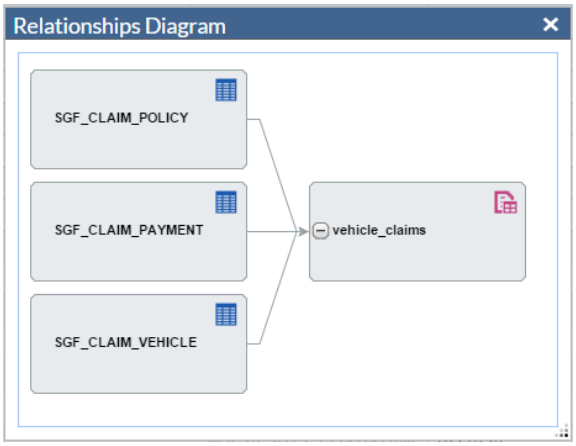

Figure 4. Relationships Diagram

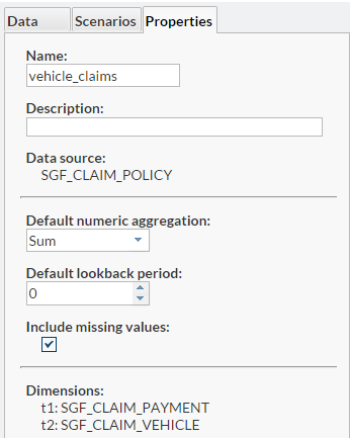

Figure 5. Properties of the Created Window

WINDOW FILTER

In SAS Visual Scenario Designer, you can set window filters, which apply to all columns in the window. For example, you can specify a window filter to exclude claims that are for a very small payment amount. The payment table has the payment amount information and you already have joined the payment amount column to the policy table, which is automatically renamed as “t1_PAYMENT_AMT”. You can click the window filter icon to add the window filter shown in Figure 6.

Figure 6. Window Filter to Exclude Claims

COLUMN PROPERTIES EDITOR

SAS Visual Scenario Designer provides column-level aggregation, conditioning, and look-back period specification. For example, Figure 7 shows how to aggregate the following columns:

- Total number of claims for each policy
- Total payment amount for all claims on each policy, looking back over the past 720 days
- Average payment amount for claims where the claim type is collision

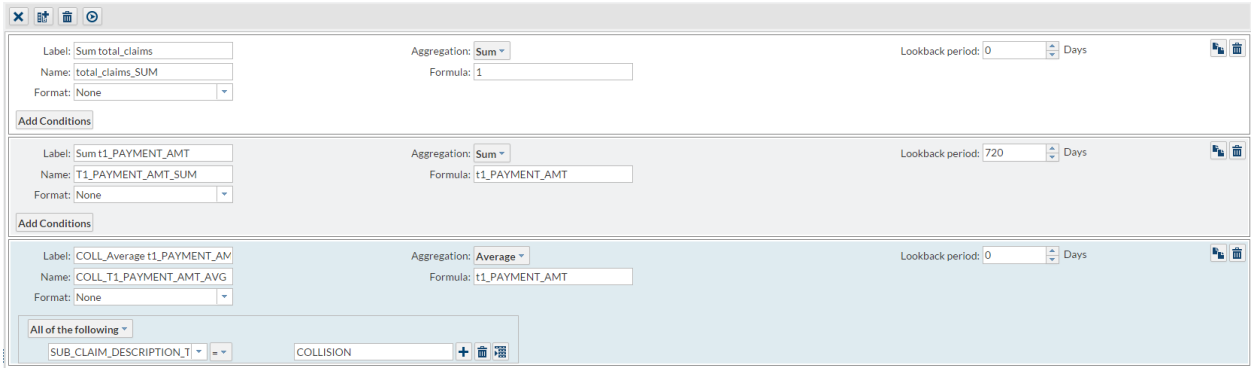

Figure 7. Column Properties Editor

CREATE A SCENARIO

Now that you are finished the data preparation, you are ready to start detecting fraudulent claims. Click "New Scenario" in the New Menu dropdown list in the upper left corner to open the New Scenario window. This window enables you to create a scenario that uses either a decision table approach or an exploration approach.

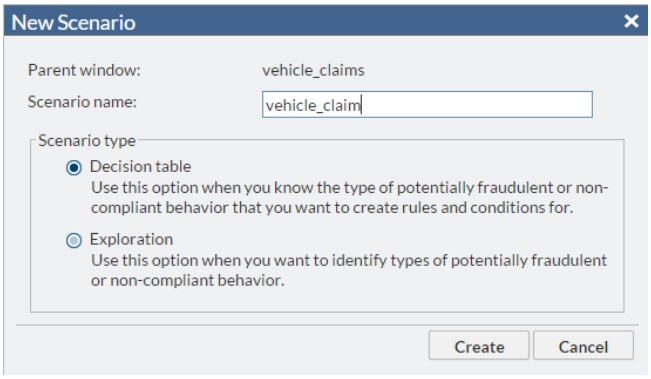

Figure 8. Create a New Scenario

DECISION TABLE

In "Decision table", you use a decision tree algorithm to automatically generate rules, which can be handcrafted afterwards. If you choose to manually add rules, you can do so by dragging and dropping columns onto the decision table.

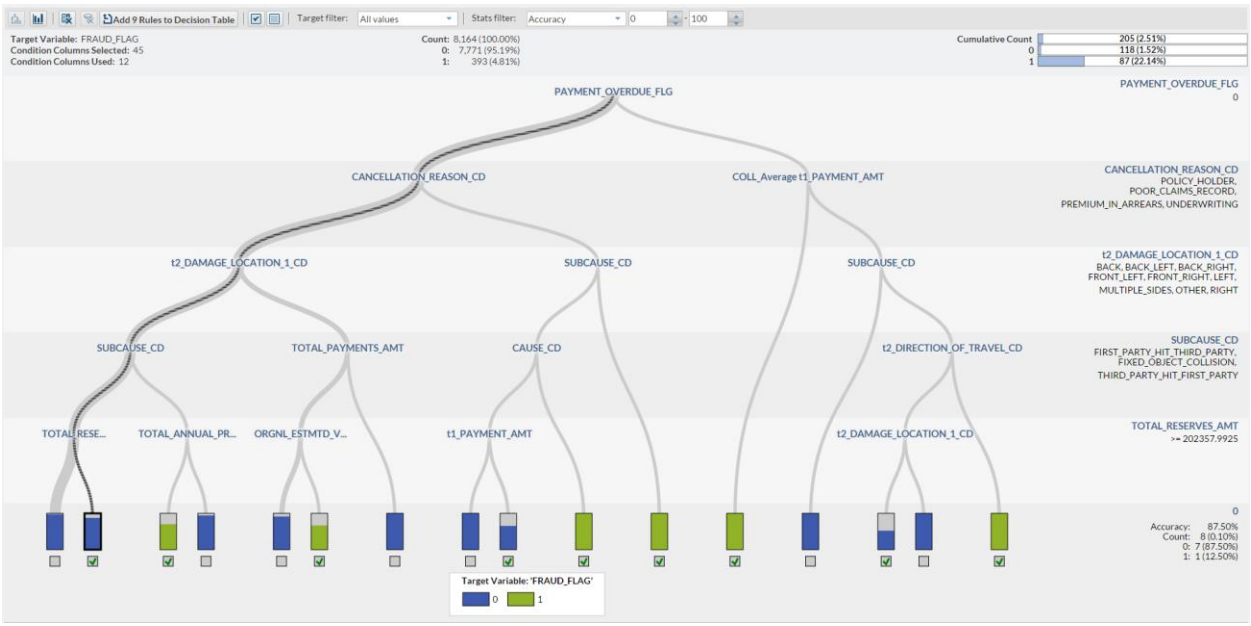

Figure 9. Using a Decision Tree to Derive Rules

In a decision tree session, it is very easy to build your own model by dragging and dropping target and predictor columns. A decision tree is a rule-based classifier that optimizes the classification of the response column, FRAUD_FLAG. A decision tree produces paths from the root node to leaves, which are a set of if-then rules. Those rules can be added to a decision table by selecting the boxes below the leaves that you like and by clicking the button on the top “Add # Rules to Decision Table”. The leaves are classified and colored by the majority class level of the target. In this example, leaves with more 1’s (fraudulent claims) are colored green and leaves with more 0’s (legitimate claims) are colored blue. Select all six green and the three blue leaves with a significant number of fraudulent claims. The cumulative counts in the top right corner indicate that you selected 205 transactions, of which 87 are fraudulent and 118 are legitimate.

You can change decision tree options, including maximum number of branches, maximum depth level, and minimum number of records per rule (leaf node). You can also toggle the aggressive search algorithm in the “Tree options” tab. Aggressive search that rebalances a decision tree based on the smaller counts when target counts differ considerably. This enables a decision tree to find rules that can detect more frauds at the price of more false positives.

The Variable importance view displays the variables ranked according to their variable importance as derived by the total Gini reduction of decision tree, as shown in Figure 10.

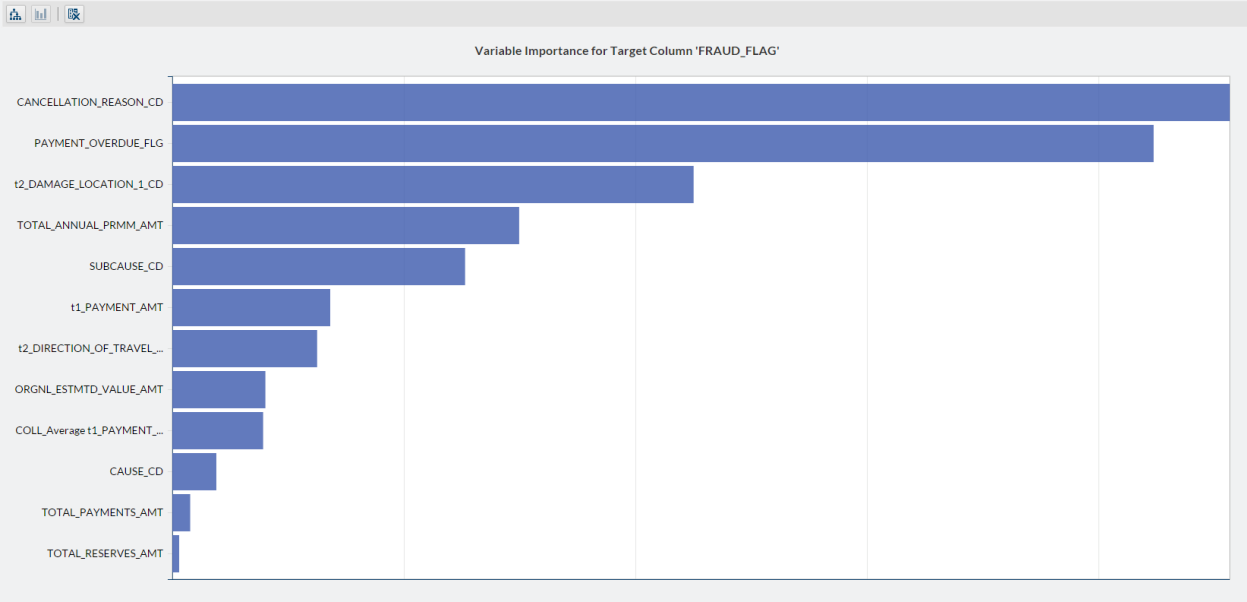

Figure 10. Variable Importance View, where Variables are Ranked According to Variable Importance

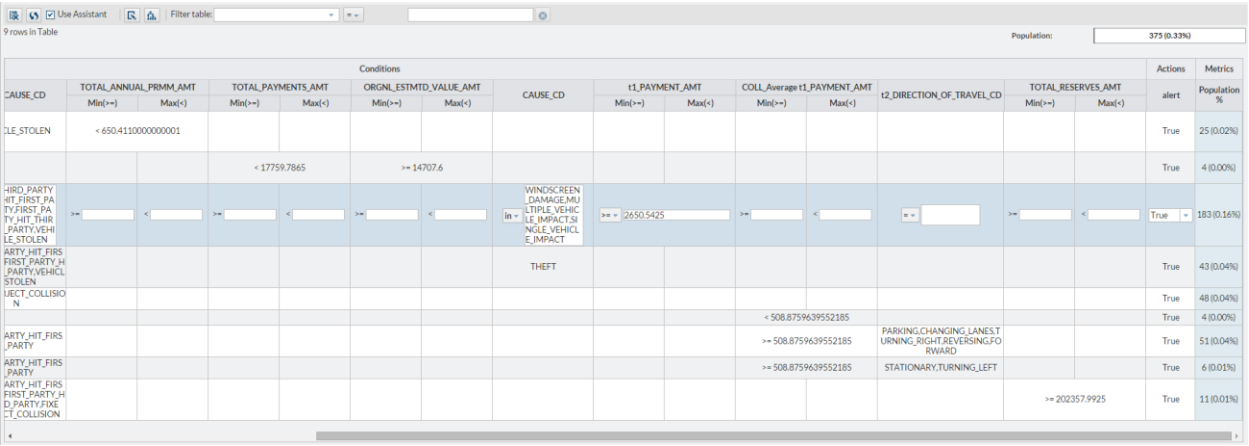

Figure 11. A Decision Table with Rules Derived from the Decision Tree. The Rule Highlighted is the Rule Being Edited.

In a decision table, you can generate and modify the rules. For example, you can change threshold values, drag and drop to add more columns, and can delete columns by right-clicking on the names of columns and selecting “Delete Column”.

In a decision table, ““Action columns” show you if an alert has been triggered and if you need to investigate it. Action columns are added through the “Edit Action Columns” feature. There are two types of actions available in “Edit type” – a binary True/False label and a numeric score. After it is created, the action is displayed under “Scenario Output Columns” at the bottom of the “Condition Columns” window in the left side of the window. To apply the created action, you can drag-and-drop the action to the “Actions” field in the middle panel to mark the rules. In this example, you created a binary True/False label and marked all the rules as true, because the rules are all derived from the decision tree to detect true frauds.

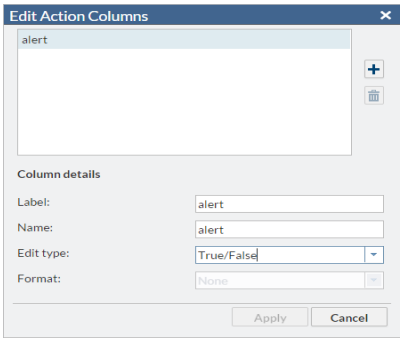

Figure 12. Create a Binary True/False Action Named and Labeled "alert" in the "Edit Action Columns" Window.

EXPLORATION

Another method to derive rules in SAS Visual Scenario Designer is through exploration. It is very convenient to do preliminary data exploration work before designing complicated scenarios. For example, if you drag and drop t2_DAMAGE_DESCRIPTION_TXT to the middle panel, all of the 12 types of damage and their percentages are displayed. You can display counts by toggling the switching button at the top of the middle panel.

Suppose that you are interested in three damage types – collision, fire, and theft. You can select these damage types, add them to scenario conditions, and work only on these damage types. More conditions can be added by adding more columns to the middle panel.

In Figure 13, payment amount is added, automatically binned into five equal-width buckets, and the two bins with the highest values of payment amount are selected. Total population percentage and count of the three types of accidents for which the payment amounts are higher than 6,080.01 are displayed in the top right corner.

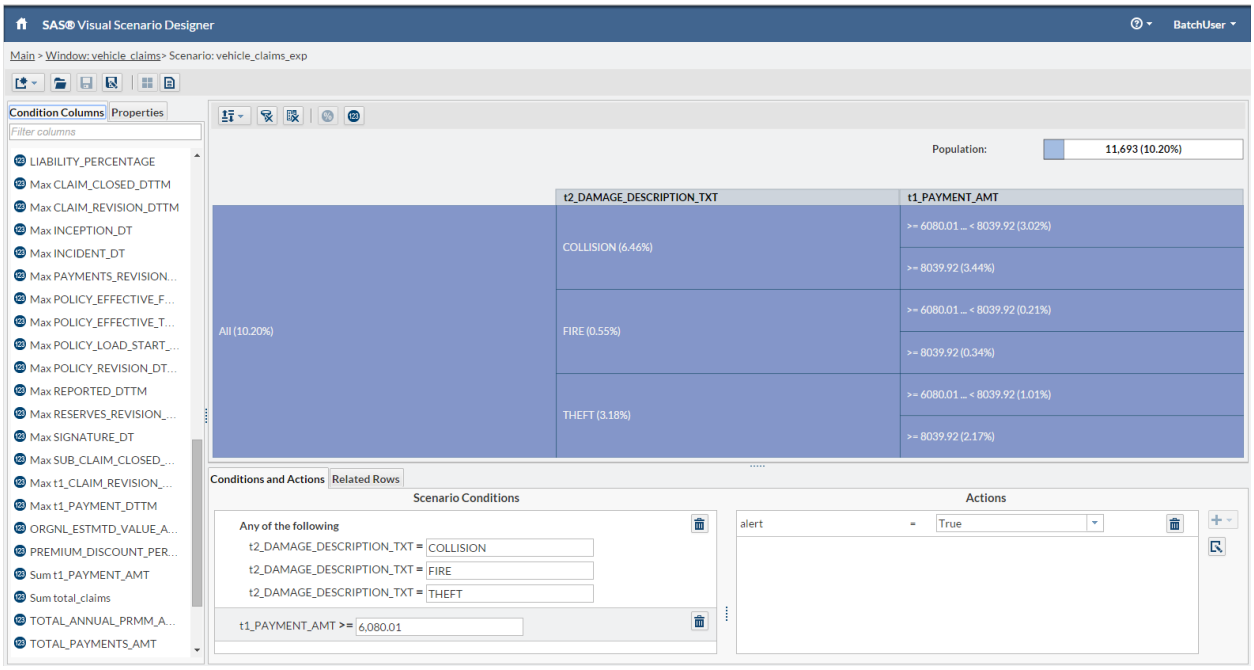

Figure 13. Explore, Subset Data and Derive Rules Using Exploration

CREATE A DEPLOYMENT

In a deployment session, you can organize your work by choosing which window you want to deploy and what scenarios you want to apply. This is helpful when you have built multiple windows and many scenarios. In this example, you use the scenario built in the decision table and exclude the scenario built in exploration by disabling it.

Before you deploy the window-level conditions and scenarios, it is very important to tune the parameters (which are the numeric thresholds and the categorical levels included in the conditions). Traditionally, this tuning work not only requires much effort, but is also very hard to manage, because when you run many simulations, the previous simulation parameters and results are often lost or forgotten. Therefore, it can be cumbersome to track the changes of the values of parameter and results, especially when there are many parameters to tune.

SAS Visual Scenario Designer offers a convenient way to solve this problem. Each time you change the values of parameters and run a simulation, the number of alerts generated and values of the parameters changed are automatically recorded. The version number of the simulation is also tracked so that you can see the results of any previous simulations. What is even more convenient is that you can see the simulation history in a histogram or parallel coordinates plot. The histogram shows the history of the alert number with simulation version number as the horizontal axis. The parallel coordinates plot shows not only the alert number history, but also the changes of the parameter values in each simulation. This enables you to track your simulation work and improve the efficiency of your tuning work.

Currently, SAS Visual Scenario Designer provides two ways to deploy the window-level conditions and scenarios. You can directly deploy the window-level conditions and scenarios to run in fast-batch using SAS In-Memory Server, or to SAS Event Stream Processing Engine for real-time process.

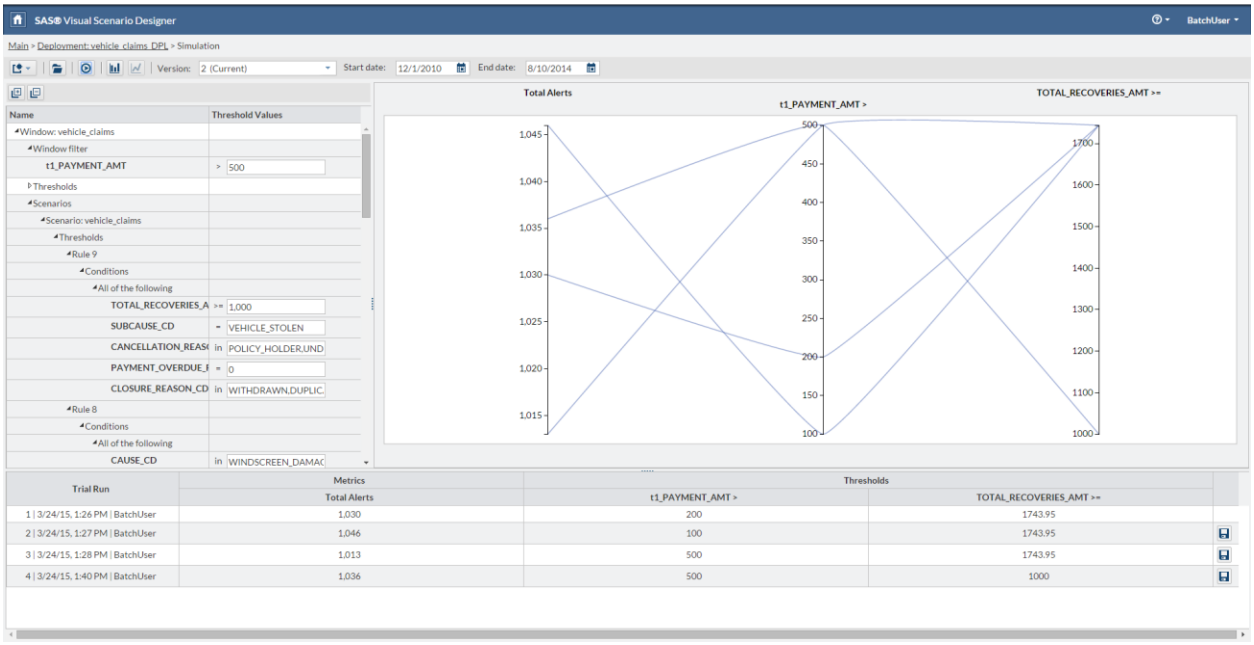

Figure 14. The Parallel Coordinates of Simulation History

CONCLUSION

SAS Visual Scenario Designer is a visual tool that uses data to identify anomalies and events. It leverages distributed in-memory computations to instantaneously analyze enormous amounts of data. It covers the entire analytical cycle for fraud detection, including data preparation, model building, parameter tuning, and deployment.

REFERENCE

SAS® LASR™ Analytic Server 2.5: Reference Guide. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/documentation/cdl/en/inmsref/67629/HTML/default/viewer.htm

SAS® Visual Scenario Designer. http://support.sas.com/documentation/onlinedoc/vsd/index.html

ACKNOWLEDGMENTS

The authors would like to thank Jamie Hutton and Rick Matthews for providing the data and developing the preliminary work flow.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Yue Qi
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Yue.Qi@sas.com
http://www.sas.com