"Call Type" Predictions for Fire Department Calls for the city of San Francisco

Department of Information Systems, California State University Los Angeles
https://gallery.cortanaintelligence.com/Experiment/Fire-department

By: David Ward, Shivaniben Shah, Tanjina Akter Reema, Yangyang Jia
Instructor: Jongwook Woo, 05/19/2017

Overview

In this tutorial, you will train and evaluate a classification model. Classification is one of the fundamental machine learning methods used in data science. Classification models enable you to predict classes or categories of a label value. Classification algorithms can be two-class methods, where there are two possible categories or multi-class methods. In our project since we just have 2 category that is fraud or genuine transaction, we will use two-class methods such as

- Two-class Logistic Regression
- Two-class Decision Forest
- Two-class support Vector Machine

Like regression, classification is a supervised machine learning technique, wherein models are trained from labeled cases. As discussed earlier, in this tutorial you will use the data set provided to determine ithe call type group.

What You'll Need To complete this lab, you will need the following:

• An Azure ML account
• A web browser and Internet connection
• Python
• Public Dataset from "Fire Department Calls For Service" from DataSF

Data Source:
https://data.sfgov.org/Public-Safety/Fire-Department-Calls-forService/nuek-vuh 3/data

Preparing and Exploring the Data

Fire Calls-For-Service includes all fire units responses to calls. Each record includes the call number, incident number, address, unit identifier, call type, and disposition.

Upload the Data Set The full Fire Department Calls For Service data set requires a lengthy model training time.

1. If you have not already done so, open a browser and browse to https://studio.azureml.net. Then sign in using the Microsoft account associated with your Azure ML account.
2. Create a new blank experiment, and give it the title Fraud Detection Small .
3. With the Fire experiment open, at the bottom left, click NEW. Then in the NEW dialog box, click the DATASET tab as shown in the following image.
4. Click FROM LOCAL FILE. Then in the Upload a new dataset dialog box, browse to select the frauddetectionsmall.csv file.
5. Enter the following details as shown in the image below, and then click the OK icon
    - This is a new version of an existing dataset: Unselected
    - Enter a name for the new dataset: frauddetectionsmall (Clean)
    - Select a type for the new dataset: Generic CSV file with a header (.csv)
    - Provide an optional description: Fraud Detection Transaction.

6. Wait for the upload of the dataset to be completed, and then on the experiment items pane, expand Saved Datasets and My Datasets to verify that the Fire dataset is listed.

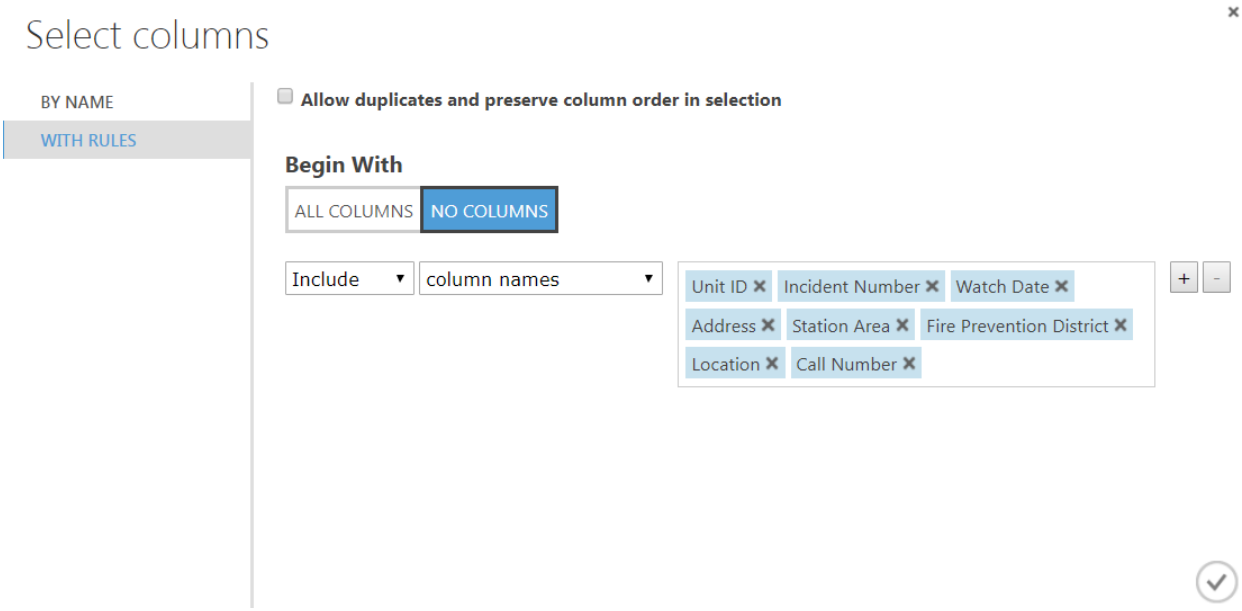
Building a Classification Model

Now that you have investigated the relationships in the data you will build, improve and validate a machine learning model. Create a New Model

1. If you are working with Python, you need to add a Edit Metadata module to your experiment by following steps a, b and c below.

a. Search for the Edit Metadata and drag it onto the canvas. Connect the output of the frauddetectionsmall(Clean) data set to the input of the Edit Metadata.

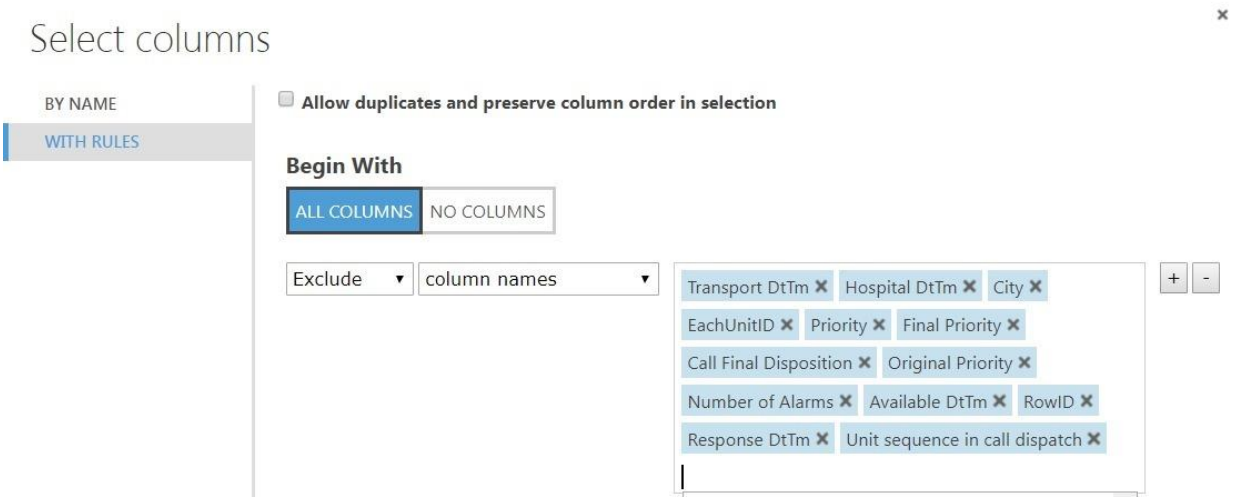
b. Click the Edit Metadata and in the properties pane, launch the Column Selector. Select all string columns as shown:



c. In the Categorical drop down list, select Make Categorical.

2. Search for the Select Columns in Dataset module and drag it onto your canvas. Connect the Results Dataset output of the Edit Metadata module to the input port of the Select Columns in Dataset module.

3. With the Select Columns in Dataset module selected, in the properties pane, launch the column selector, and exclude the following columns.



4. Search for the Split Data module. Drag this module onto your experiment canvas. Connect the Results dataset output port of the Normalize Data module to the Dataset input port of the Split Data module. Set the Properties of the Split Data module as follows:

- Splitting mode: Split Rows
- Fraction of rows in the first output: 0.7
- Randomized Split: Checked
- Random seed: 12345
- Stratified Split: False

5. Search for the Two Class Logistic Regression. Make sure you have selected the regression model version of this algorithm. Drag this module onto the canvas. Set the Properties if this module as follows:

- Create trainer mode: Single Parameter
- Train model: Call Type Group

6. Search for the Two Class Decision Forest module. Make sure you have selected the regression model version of this algorithm. Drag this module onto the canvas. Set the Properties if this module as follows:

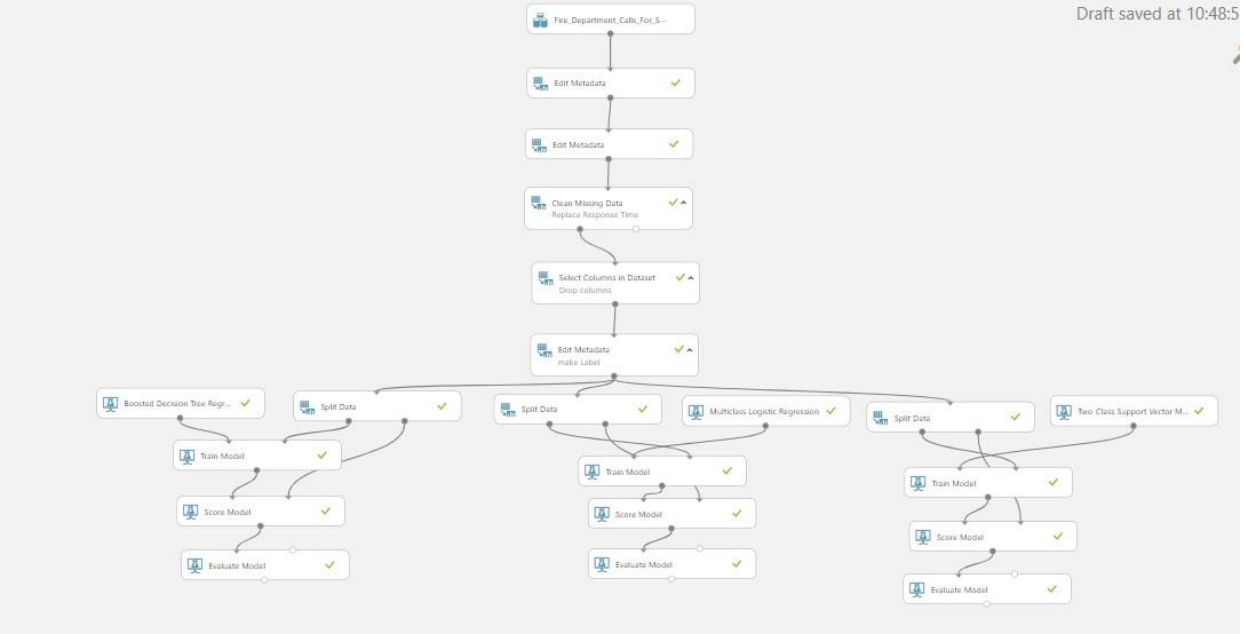
- Resampling method: Bagging
- Create trainer mode: Parameter Range
- Number of decision trees: 8
- Maximum depth of the decision trees: 32
- Number of random Splits per node: 128
- Minimum number of samples per leaf node:

7. Search for the Two Support Vector Mechanism module. Drag this module onto the canvas. Set the Properties if this module as follows:

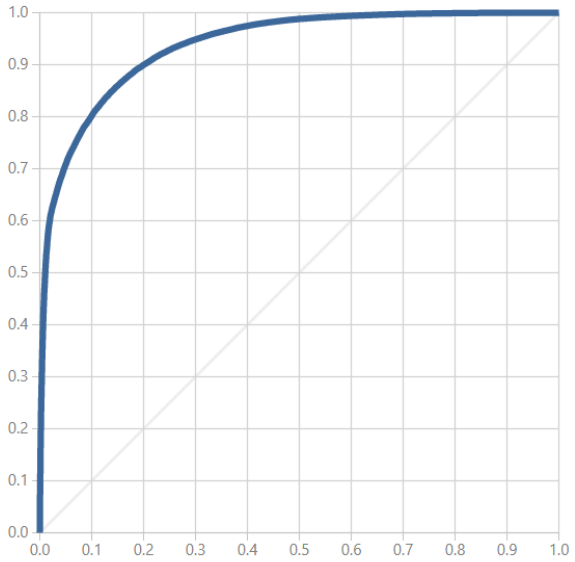
- Create trainer mode: Single Parameter
- Number of Iterations: 1

8. Search for the Evaluate Model module and drag it onto the canvas. Connect the Scored Dataset output port of the Score Model module to the left hand Scored dataset input port of the Evaluate Model module.

9. Save and run the experiment. When the experiment is finished, visualize the Evaluation Result port of the Evaluate Model module and review the ROC curve and performance statistics for the model as shown below.



10. Examine this ROC line. Notice that the bold blue line is well above the diagonal grey line, indicating the model is performing significantly better than random guessing. The AUC is 0.938



11. Next examine the performance statistics for two-class decision forest is as shown here:

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
180415	6853	0.898	0.912	0.5	0.938
False Positive	True Negative	Recall	F1 Score		
17336	32167	0.963	0.937		
Positive Label	Negative Label				
Fire	Alarm				

Summary

In this tutorial, you have constructed and evaluated 3 two class or binary classification model. Highlight from the results of this lab are:

• Visualization of the data set can help differentiate features which separate the cases from those that are unlikely to do so.

• Examining the classification behavior of features can highlight potential performance problems or provide guidance on improving a model.

• Based on the recall, precision, AUC and the time taken to train the model we have come to the conclusion that Two-classs Logistic Regression is the best model.