Microsoft Data Science

Cortana Intelligence Suite – Azure ML Experiment

Lab Instructions

9/30/2016

Welcome to the Cortana Intelligence Suite Workshop delivered by your Microsoft team. This experiment demonstrates how we can build a binary classification model to predict income levels of adult individuals. The process includes training, testing and evaluating the model on the Adult dataset.

Binary Classification: Income Level Prediction

In this sample experiment we will train a binary classifier on the Adult dataset, to predict whether an individual's income is greater or less than $50,000. We will show how you can perform basic data processing operations, split the dataset into training and test sets, train the model, score the test dataset, and evaluate the predictions.

As you drag in each module, click the (more help...) link in the bottom right side of the

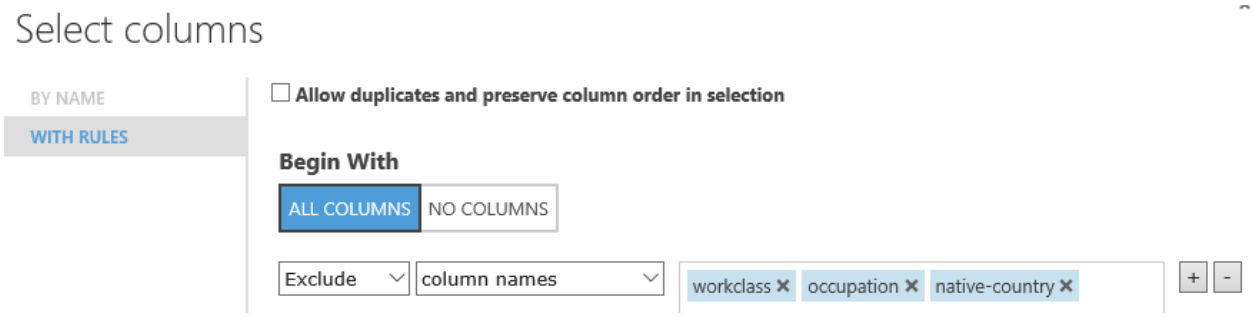
Setup the Experiment

1. In your browser, navigate to http://studio.azureml.net
2. Sign in
3. Click **Experiments** on the left, then click **+ New** at the bottom
4. Choose **Blank Experiment**

Creating the Experiment

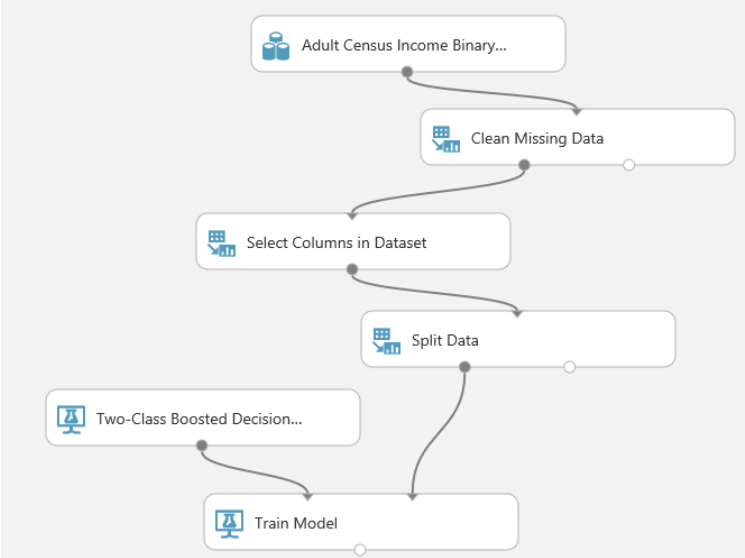
5. Drag and drop the `Adult Census Income Binary Classification dataset` module into your experiment's workspace.
6. Add a **Clean Missing Data** module, and use the default settings, to replace missing values with zeros. Connect the dataset module output to the input port.

7. Add a **Select Columns in Dataset** module, and connect the left output of **Clean Missing Data** module to the input port.
    1. Use the column selector to **exclude** these columns: `workclass`, `occupation`, and `native-country`. We are excluding these columns because we don't want their values to be used in the training process. By default, Azure ML Studio treats all columns as features except for the target variable (the Label column). Alternatively, you could use the **Edit Metadata** module, select the excluded columns, and then choose *ClearFeatures* from the **Fields** dropdown list.

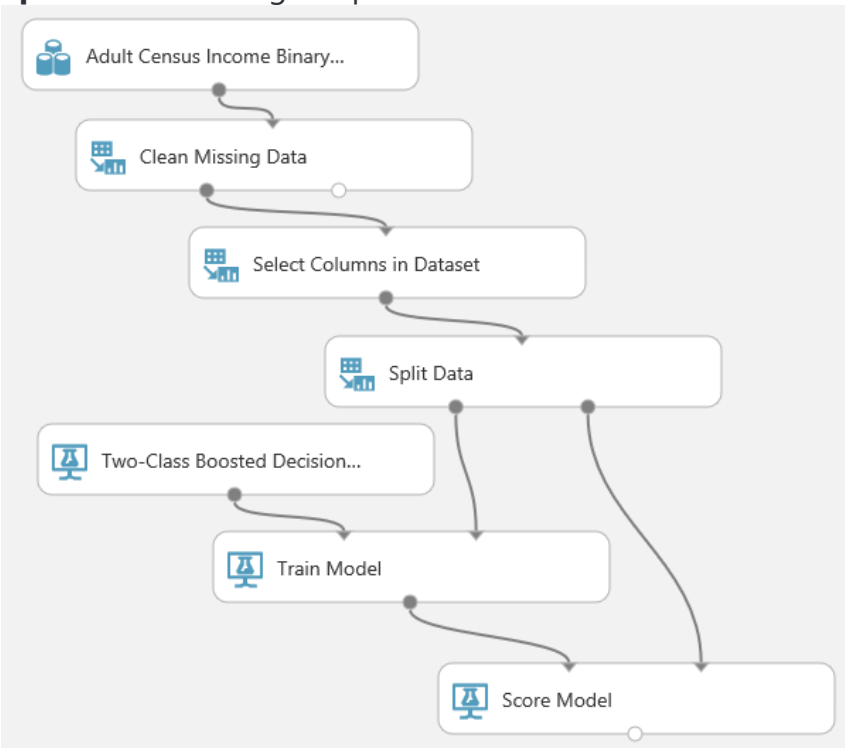


8. Add a **Split Data** module to create the testing and test sets. Set the *Fraction of rows in the first output dataset* to 0.7. This means that 70% of the data will be output to the left port and the rest to the right port of this module. We will use the left dataset for training and the right one for testing.
9. Add a **Two-Class Boosted Decision Tree** module to initialize a boosted decision tree classifier. (Don't connect it yet)
10. Add a **Train Model** module and connect the classifier (step 5) and the training set (left output port of the **Split** module) to the left and right input ports respectively. This module will perform the training of the classifier. Select **income** as the Label

column.



11. Add a **Score Model** module and connect the **Train Model** to the left input, and the **Split Data** to the right input.

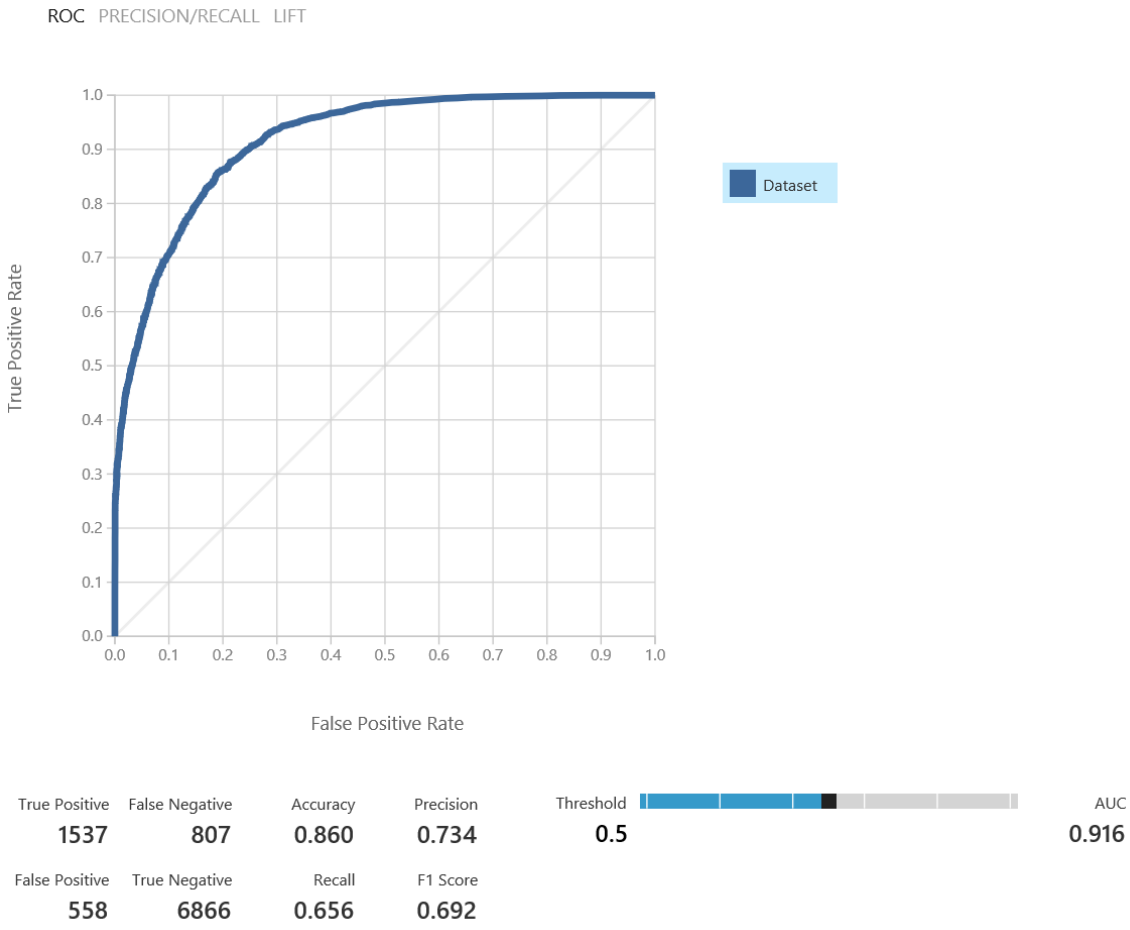


12.
13. Add an **Evaluate Model** module and connect the scored dataset to the left input port.
14. Change the name at the top of the page, and save your experiment.
15. Run your experiment.

16. To see the evaluation results, click on the output port of the **Evaluate Model** module and select *Visualize*.

Results

From these results, you can see that the **Two-Class Boosted Decision Tree** is fairly accurate in predicting income for the `Adult Census Income` dataset. Is it too accurate? Can you add another algorithm and compare how well they do? Which one should you choose?



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
1537	807	0.860	0.734	0.5	0.916

False Positive	True Negative	Recall	F1 Score
558	6866	0.656	0.692

Review Validation

Now navigate to this topic and review the difference between Cross-Validation and Evaluation: https://azure.microsoft.com/en-gb/documentation/articles/machine-learning-evaluate-model-performance/