Azure machine learning Practice 3

1. Question: Selection to predict income with Azure Machine Learning designer:

Here we learn how to build a machine learning classifier without writing a single line of code using the designer. This sample trains a two-class boosted decision tree to predict adult census income (>=50Kor <=50K).

This is called a classification problem. We can apply the same fundamental process to tackle any type of machine learning problem - regression, classification, clustering, and so on.

Classification in machine learning: Classification is a supervised machine learning technique used to predict categories or classes. Learn how to create classification models using Azure Machine Learning designer.

Predict income in machine learning: Microsoft Azure Machine Learning Studio is a drag-and-drop tool that allows to visually and collaboratively build, test, and deploy machine learning models. In this Lab, We will predict income levels using census data and compare the performance of two trained models in Azure Machine Learning Studio. We will see how easy it is to build powerful models without having to write a single line of code!

Machine-learning pipelines: Azure Machine Learning pipeline is an independently executable workflow of a complete machine learning task. Subtasks are encapsulated as a series of steps within the pipeline. Pipelines should focus on machine learning tasks such as:

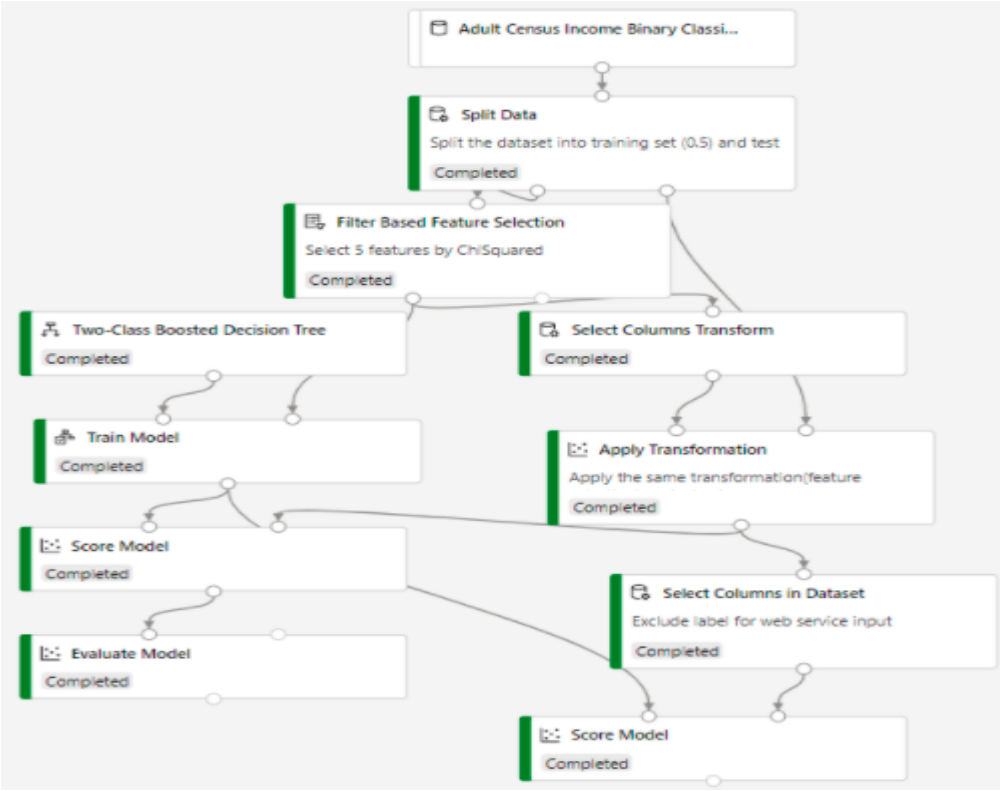
- Data preparation including importing, validating and cleaning, munging and transformation, normalization, and staging
- Training configuration including parameterizing arguments, file paths, and logging/reporting configurations
- Training and validating efficiently and repeatedly. Efficiency might come from specifying specific data subsets, different hardware compute resources, distributed processing, and progress monitoring
- Deployment, including versioning, scaling, provisioning, and access control.

Here's the final pipeline graph for this sample: At first we need to create a pipeline graph. In this case, we need to must maintain the sequence for this sample step by step. Such as …

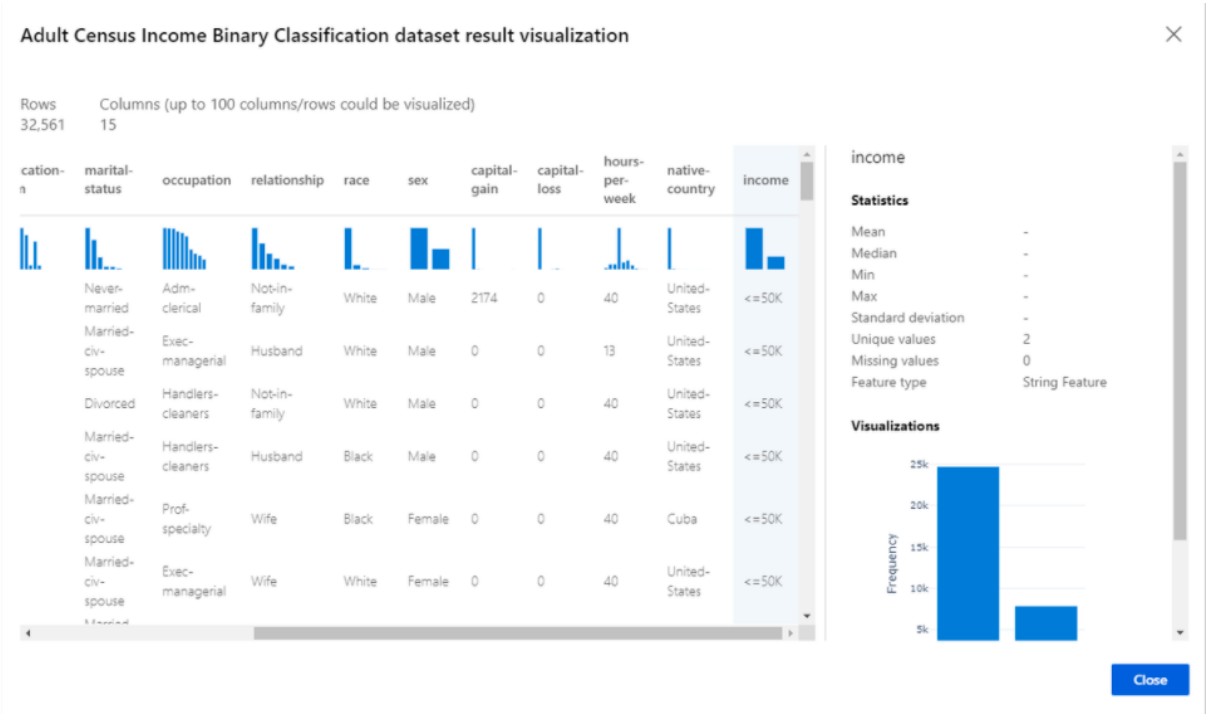
1. Adult sensual income Binary classification
2. Split data
3. Filter-Based future selection
4. Two class Boosted Decision Tree
5. Select columns Transform
6. Train Model
7. Apply Transformation
8. Score model
9. Select column in Dataset
10. Evaluate Model
11. Score Model

Pipeline Summary: When we want to create a pipeline. We need to follow these steps to create the pipeline:

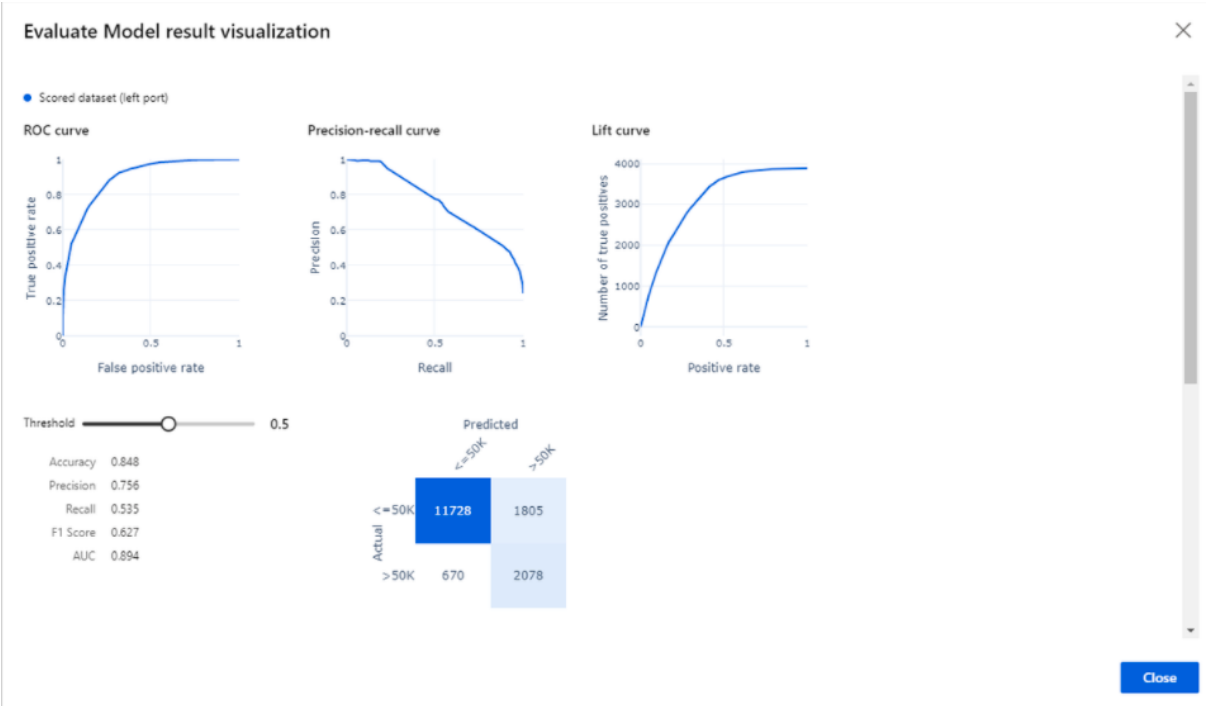
1. Drag the Adult Census Income Binary dataset module into the pipeline canvas.
2. Add a Split Data module to create the training and test sets. Set the fraction of rows in the first output dataset to 0.7. This setting specifies that 70% of the data will be output to the left port of the module and the rest to the right port. We use the left dataset for training and the right one for testing.
3. Add the Filter Based Feature Selection module to select 5 features by ChiSquared.
4. Add a Two-Class Boosted Decision Tree module to initialize a boosted decision tree classifier.
5. Add a Train Model module. Connect the classifier from the previous step to the left input port of the Train Model. Connect the filtered dataset from Filter Based Feature Selection module as the training dataset. The Train Model will train the classifier.
6. Add Select Columns Transformation and Apply Transformation module to apply the same transformation (filtered based feature selection) to the test dataset.
7. Add the Score Model module and connect the Train Model module to it. Then add the test set (the output of Apply Transformation module which applies feature selection to the test set too) to the Score Model. The Score Model will make the predictions. You can select its output port to see the predictions and the positive class probabilities. This pipeline has two score modules, the one on the right has excluded label column before making the prediction. This is prepared to deploy a real-time endpoint because the web service input will expect only features not label.
8. Add an Evaluate Model module and connect the scored dataset to its left input port. To see the evaluation results, select the output port of the Evaluate Model module and select Visualize.



Data: The dataset contains 14 features and one label column. There are multiple types of features, including numerical and categorical. The following diagram shows an excerpt from the dataset:


Adult Census Income Binary Classification dataset result visualization
Rows
32,561
Columns (up to 100 columns/rows could be visualized)
15
cation-n
marital-status
occupation
relationship
race
sex
capital-gain
capital-loss
hours-per-week
native-country
income
Never-married
Adm-clerical
Not-in-family
White
Male
2174
0
40
United-States
<=50K
Married-civ-spouse
Exec-managerial
Husband
White
Male
0
0
13
United-States
<=50K
Divorced
Handlers-cleaners
Not-in-family
White
Male
0
0
40
United-States
<=50K
Married-civ-spouse
Handlers-cleaners
Husband
Black
Male
0
0
40
United-States
<=50K
Married-civ-spouse
Prof-specialty
Wife
Black
Female
0
0
40
Cuba
<=50K
Married-civ-spouse
Exec-managerial
Wife
White
Female
0
0
40
United-States
<=50K
income
Statistics
Mean -
Median -
Min -
Max -
Standard deviation -
Unique values 2
Missing values 0
Feature type String Feature
Visualizations
Frequency
25k
20k
15k
10k
5k
Close

Results:


Evaluate Model result visualization
Scored dataset (left port)
ROC curve
True positive rate
False positive rate
Precision-recall curve
Precision
Recall
Lift curve
Number of true positives
Positive rate
Threshold 0.5
Accuracy 0.848
Precision 0.756
Recall 0.535
F1 Score 0.627
AUC 0.894
Predicted
<=50K
>50K
Actual
<=50K
11728
1805
>50K
670
2078
Close

[Source: From google]

2. Question: The Two-Class Boosted Decision Tree algorithm in MS Azure documentation and the advantages and disadvantages of this algorithm.

The Two-Class Boosted Decision Tree algorithm: The Two-Class Boosted Decision Tree module in Machine Learning Studio (classic), creates a machine learning model that is based on the boosted decision trees algorithm.

Generally, when properly configured, boosted decision trees are the easiest methods with which to get top performance on a wide variety of machine learning tasks. However, they are also one of the more memory-intensive learners, and the current implementation holds everything in memory. Therefore, a boosted decision tree model might not be able to process the very large datasets that some linear learners can handle.

The advantages and disadvantages of this algorithm: The decision Tree is a very popular machine learning algorithm. Decision Tree solves the problem of machine learning by transforming the data into a tree representation. Each internal node of the tree representation denotes an attribute and each leaf node denotes a class label.

A decision tree algorithm can be used to solve both regression and classification problems.

Advantages:

- Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
- A decision tree does not require the normalization of data.
- A decision tree does not require the scaling of data as well.
- Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
- A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

Disadvantage:

- A small change in the data can cause a large change in the structure of the decision tree causing instability.
- For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
- Decision tree often involves higher time to train the model.
- Decision tree training is relatively expensive as the complexity and time have taken are more.

- The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

[Source: From google]

3. Question: <u>How can income predictions be used in a business process of a company? Describe 2-3 examples.</u>

Income prediction is valuable to businesses because it gives them the ability to make informed business decisions and develop data-driven strategies. Past data is aggregated and analyzed to find patterns, used to predict future trends and changes. Forecasting allows the company to be proactive instead of reactive

Three examples are given below:

 # Straight-line Method: The straight-line method is one of the simplest and easy-to-follow Income prediction methods. A financial analyst uses historical figures and trends to predict future revenue**.** In the example provided below, we will look at how straight-line forecasting is done by a retail business that assumes a constant sales growth rate of 4% for the next five years.

- To forecast future revenues, take the previous year's figure and multiply it by the growth rate. The formula used to calculate 2017 revenue is =C7*(1+D5).

 # Moving Average: Moving averages are a smoothing technique that looks at the underlying pattern of a set of data to establish an estimate of future values. The most common types are the 3-month and 5-month moving averages.

- To perform a moving average forecast, the revenue data should be placed in the vertical column. Create two columns, 3-month moving averages, and 5-month moving averages.

 # Simple Linear Regression: Regression analysis is a widely used tool for analyzing the relationship between variables for prediction purposes. In this example, we will look at the relationship between radio ads and revenue by running a regression analysis on the two variables.

- Select the Radio ads and Revenue data in cells B4 to C15, then go to Insert > Chart > Scatter.

[Source: From google]

Practice 4

1. Questions: Build a classifier & use Python scripts to predict credit risk using Azure Machine Learning designer

This article shows how to build a complex machine learning pipeline using the designer. You'll learn how to implement custom logic using Python scripts and compare multiple models to choose the best option.

This sample trains a classifier to predict credit risk using credit application information such as credit history, age, and the number of credit cards. We can apply the concepts in this article to tackle our own machine learning problems.

Machine-learning pipelines: Azure Machine Learning pipeline is an independently executable workflow of a complete machine learning task. Subtasks are encapsulated as a series of steps within the pipeline. Pipelines should focus on machine learning tasks such as:

- Data preparation including importing, validating and cleaning, munging and transformation, normalization, and staging
- Training configuration including parameterizing arguments, file paths, and logging/reporting configurations
- Training and validating efficiently and repeatedly. Efficiency might come from specifying specific data subsets, different hardware compute resources, distributed processing, and progress monitoring
- Deployment, including versioning, scaling, provisioning, and access control.

Here's the final pipeline graph for this sample: At first we need to create a pipeline graph. In this case, we need to must maintain the sequence for this sample step by step. Such as …

1. German credit card UCI dataset
2. Edit Metadata
3. Split data
4. Execute python script
5. Normalize data
6. Execute python script
7. Normalize data
8. Two class support vector machine
9. Normalize data, Normalize data
10. Train Model, Train Model

11. Two c;lass-Boosted decision Tree
12. Score model, Score model
13. Train Model, Train Model
14. Evaluate model
15. Score model, Score model
16. Add rows
17. Evaluate model
18. Execute python script
19. Select column in data set

Pipeline Summary: In this pipeline, We can compare two different approaches for generating models to solve this problem:

- Training with the original dataset.
- Training with a replicated dataset.

With both approaches, we evaluate the models by using the test dataset with replication to ensure that results are aligned with the cost function. Test two classifiers with both approaches: Two-Class Support Vector Machine and Two-Class Boosted Decision Tree.

Data processing: Start by using the Metadata Editor module to add column names to replace the default column names with more meaningful names, obtained from the dataset description on the UCI site. Provide the new column names as comma-separated values in the New column name field of the Metadata Editor.

Next, generate the training and test sets used to develop the risk prediction model. Split the original dataset into training and test sets of the same size by using the Split Data module. To create sets of equal size, set the Fraction of rows in the first output dataset option to 0.7.

Generate the new dataset: Because the cost of underestimating risk is high, set the cost of misclassification like this:

- For high-risk cases misclassified as low risk: 5
- For low-risk cases misclassified as high risk: 1

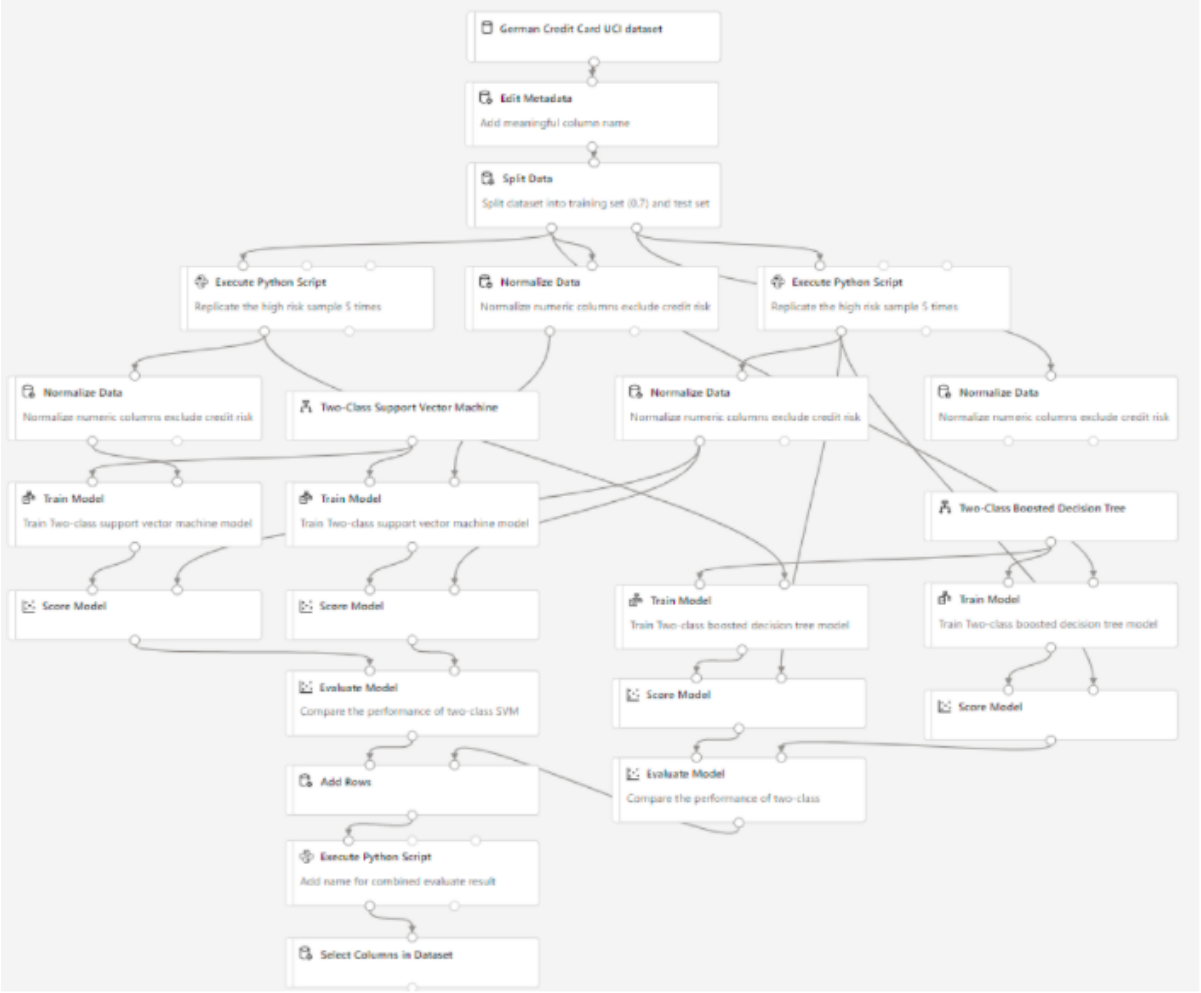
Feature engineering: The Two-Class Support Vector Machine module handles string features, converting them to categorical features and then to binary features with a value of zero or one. So you don't need to normalize these features.

Models: Because applied two classifiers, Two-Class Support Vector Machine (SVM) and Two-Class Boosted Decision Tree, and two datasets, you generate a total of four models:

- SVM trained with original data.
- SVM trained with replicated data.
- Boosted Decision Tree trained with original data.
- Boosted Decision Tree trained with replicated data.

This sample uses the standard data science workflow to create, train, and test the models:

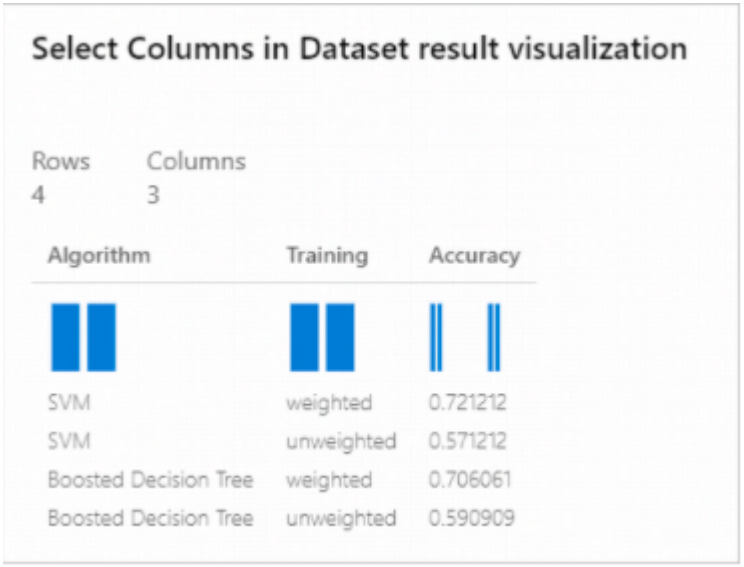
1. Initialize the learning algorithms, using Two-Class Support Vector Machine and Two-Class Boosted Decision Tree.
2. Use Train Model to apply the algorithm to the data and create the actual model.
3. Use Score Model to produce scores by using the test examples.



Data: This sample uses the German Credit Card dataset from the UC Irvine repository. It contains 1,000 samples with 20 features and one label. Each sample represents a person. The

20 features include numerical and categorical features. For more information about the dataset, see the UCI website. The last column is the label, which denotes the credit risk and has only two possible values: high credit risk = 2, and low credit risk = 1.

Results:



The first column lists the machine learning algorithm used to generate the model.

The second column indicates the type of the training set.

The third column contains the cost-sensitive accuracy value.

From these results, we can see that the best accuracy is provided by the model that was created with a Two-Class Support Vector Machine and trained on the replicated training dataset.

[Source: From Google]