CS 514
Applied Artificial Intelligence
Project 5

Santander customer satisfaction

Predicting customer satisfaction through decision tree learning

https://www.kaggle.com/c/santander-customer-satisfaction

INDEX

Usage Manual:

The code base is available at the following git link or the google drive link.

If you have a GIT account, you can fork the source code to your local repository using the below link.

https://github.com/smenon8/app-ai-projects/tree/master/FinalProject

Otherwise, the source code and the data file are available at the below Google drive location.

https://drive.google.com/folderview?id=0BzXUs8quIZ_xZFJKZlYxNHR2Umc&usp=sharing

Requirements:

To run the source code, you must have the below software installed in your machine.

Software	Download link
Anaconda (RECOMMENDED)	https://www.continuum.io/
OR	
Python 3.5	https://www.python.org/downloads/
sklearn	http://scikit-learn.org/stable/install.html
matplotlib	http://matplotlib.org/downloads.html
numpy	http://www.scipy.org/scipylib/download.html

Instructions:

After downloading the entire source code and the data folders, store it in any location. The python scripts are customized to automatically adjust and find the data files subject to both the 'script' and the 'data' folder are under the same parent folder.

Open a command terminal and type the following command:

cd <absolute path of the script directory>

Below command will perform attribute selection by calculating information gain of a particular attribute in comparison to the TARGET attribute. Information gain is calculated by another python script FeatureSelectionAPI.py. Please note that 25,000 examples from the training data set have been considered for parameter selection. Changing the second parameter in line number 46 (createToyFile("../data/train.csv","../data/trainToy.csv",25000)) will create another file and this file will be used for calculating information gain of each of the 571 attributes available. Changing this parameter might need some additional modifications in the code. I would not recommend the reviewer to change this parameter by a very large margin.

Once information gain for each parameter is calculated, then only those attributes which have information gain above the average information gain over all parameters are chosen. Please note that in a primitive windows machine, this script could take up to 10-15 minutes to finish execution.

python DataAnalysisFeatureSelectionSantander.py

Below command is the command responsible for learning a decision tree plotting different results and calculating the ROC curve area for different data points.

The accuracy of cross-validation predictions has been calculated for 10 different instances by splitting the train data into a train data – test data split from 10-90,20-80,30-70, ... ,70-30,80-20,90-10. The accuracy curve has been fit into a degree-2 curve. ROC curve for all the instances have been plot and triangles are plotted considering each of the FNF-TNF point as the second point in the triangle.

python LearningModelSantander.py

The output of the script is basically the area under the ROC curve for each of the sample training-test split sample. A total of 3 plots will appear (almost) simultaneously on your screen. Each of the plots and their interpretation are explained in the following pages.

Results

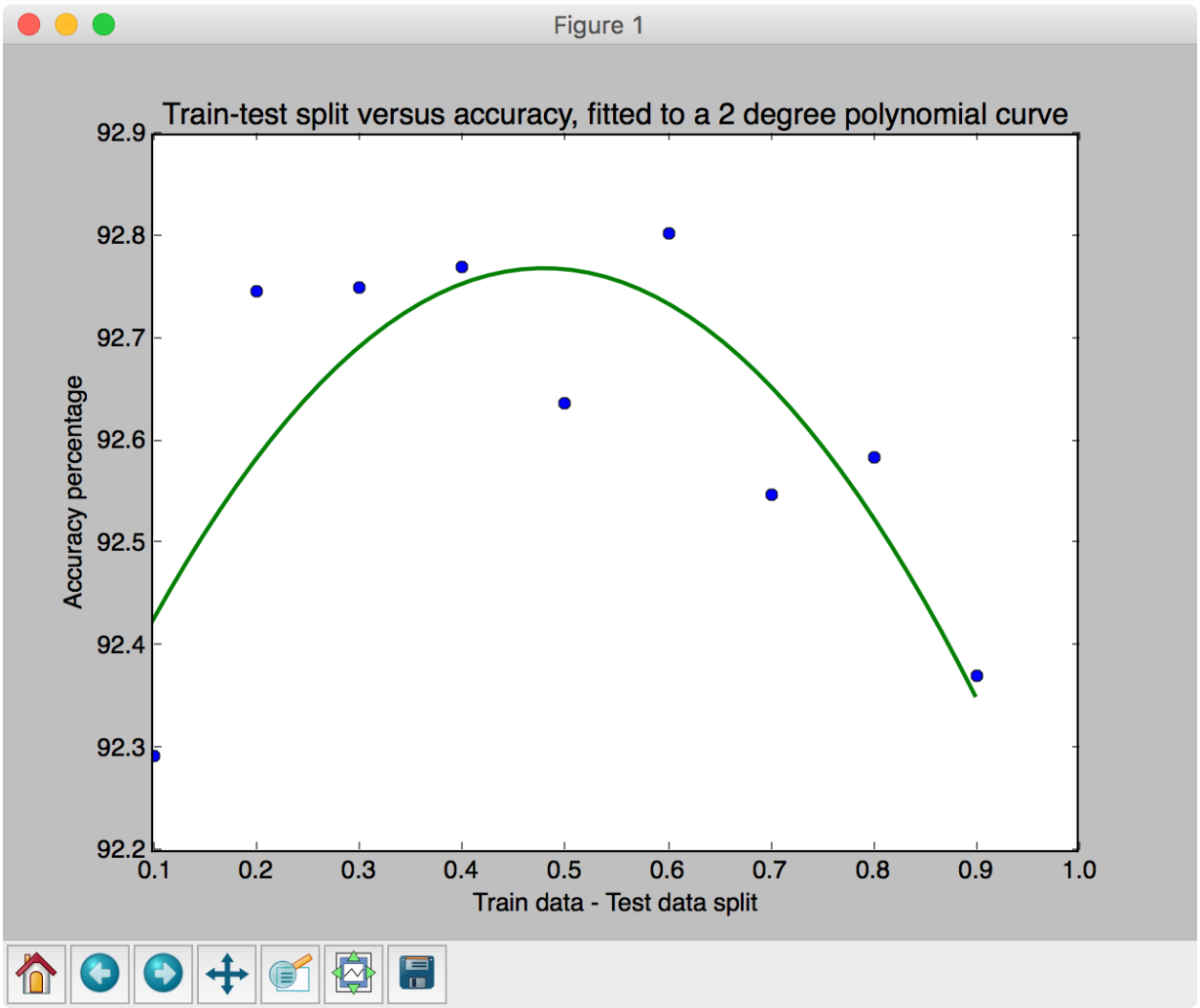
Please note that the results will slightly vary since a different split is selected randomly at each run, the values on an average must remain the same.

Current split 10.000000 - 90.000000
Accuracy : 92.580900
Area under the ROC curve : 0.461205
Current split 20.000000 - 80.000000
Accuracy : 92.811102
Area under the ROC curve : 0.462491
Current split 30.000000 - 70.000000
Accuracy : 92.646994
Area under the ROC curve : 0.461700
Current split 40.000000 - 60.000000
Accuracy : 92.837411
Area under the ROC curve : 0.462676
Current split 50.000000 - 50.000000
Accuracy : 92.570376
Area under the ROC curve : 0.461297
Current split 60.000000 - 40.000000
Accuracy : 92.747523
Area under the ROC curve : 0.462236
Current split 70.000000 - 30.000000
Accuracy : 92.519027
Area under the ROC curve : 0.461055
Current split 80.000000 - 20.000000
Accuracy : 92.633517
Area under the ROC curve : 0.461654
Current split 90.000000 - 10.000000
Accuracy : 92.399661
Area under the ROC curve : 0.460438

Plot # 1:

Plot of train-test data split percentage versus accuracy. Cross-validation accuracies are calculated on different splits of training and testing data. A value of 0.1 on x-axis means that the entire set of ~72,000 training examples were broken down into 10% training data and 90% test data and so on.

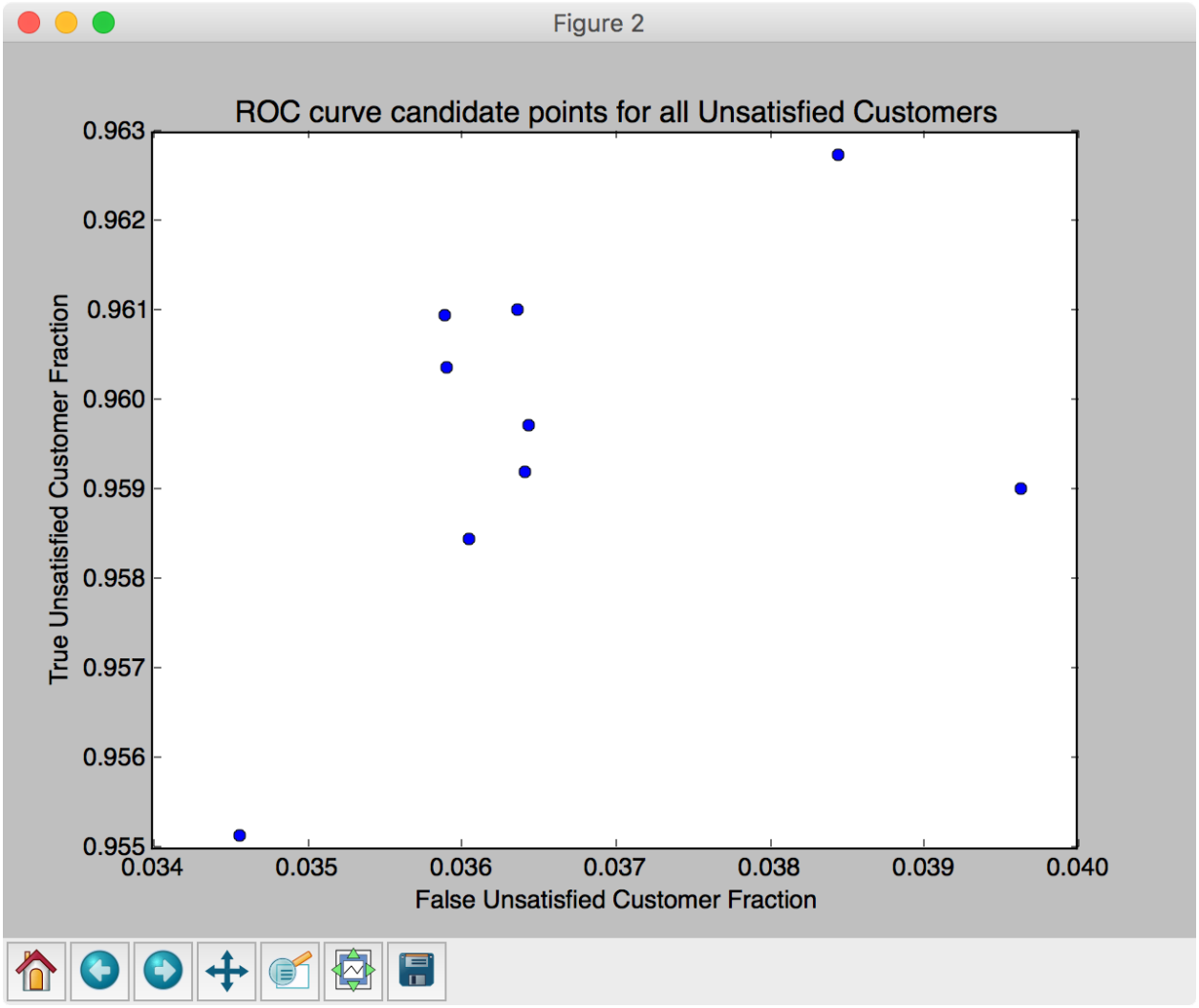
A 2-degree polynomial curve has been best fit in order to show to trend in accuracy. As you can see, accuracy peaks when there is an equal split of training and test data. With increase in the training data, the resulting trained decision tree undergoes **over-fitting** and thus the accuracy decreases. For low proportion of training data, the decision tree undergoes **under-fitting**.



Plot # 2:

ROC curve candidate points -

Each of the point represents False Negatives and True negatives for different splits.



Plot # 3:

ROC curve representation for calculating the area under the curve. The cluster of all points can be seen by zooming into the resultant curve after running the scripts using the zoom tool in the plot. For calculating area, each split has been considered separately and a triangle is constructed with the resulting ROC candidate as the second points and (0,0) and (1,1) as the other two points.