2016

P2 – Student Intervention

MACHINE LEARNING NANODEGREE

PRATHEERTH PADMAN

1. Classification vs. Regression

A. This is a classification supervised ML problem. The output or the label for the test data is discrete. It is not a continuous variable/number which is the criteria for regression. Here, the "output" would be either a yes or a no.

2. Exploring the Data

- Total number of students - 395
- Number of students who passed - 265
- Number of students who failed - 130
- Graduation rate of the class (%) – 67.09%
- Number of features (excluding the label/target column) – 30

3. Preparing the Data

Refer the iPython notebook.

4. Training and Evaluation Models

Model 1 - Decision Tree

Decision Tree is a top-down, greedy algorithm classifier. The "best split" or the most appropriate attribute for splitting the data is chosen by using "entropy" and "information gain". Entropy is the measure of homogeneity in a classified set whereas Information Gain expresses how well an attribute splits the data into groups based on the applied classifier.

Decision trees work very well where speed and simplicity are key requirements. They also tend to respond well to classification problems such as image classification or even this particular problem.

Advantages:

- Trees can be visualized and hence is easier to understand and interpret.
- It requires little data preparation. It avoids complex data normalization, creation of dummy variables and removing blank spaces.

Disadvantages:

- They can create over-complex trees, over fitting the data and failing to generalize well (as seen in this particular project. The F1 score for training data is always 1.0).
- If a particular class dominates, decision tree learners can create biased trees.

This model was chosen because a decision tree classifier tends to work well in supervised machine learning problems. It is easier to interpret and visualize. The dominant features that indicate the failure of students can be identified through a decision tree classifier.

	Training set Size		
	100	200	300
Training Time(secs)	0.001	0.001	0.002
Prediction Time(secs)	0.000	0.7611	0.000
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.6341	0.7230	0.7552

Model 2 – SVM

Support Vector Machines (SVM) are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having

different class memberships. Support vector classifier (SVC) aims to maximize the distance between the data points of each class.

SVM's work well in problems with higher dimensional data and also where the decision boundaries tend to be complex using the "kernel trick".

Advantages:

- They're effective in high dimensional spaces.
- They're also very versatile. Different kernels can be specified for different purposes.

Disadvantages:

- If the number of features is greater than the number of samples, this method is likely to give poor performances.

This model was chosen given the high number of features in the dataset.

	Training set Size		
	100	200	300
Training Time(secs)	0.002	0.004	0.008
Prediction Time(secs)	0.001	0.003	0.006
F1 score for training set	0.9027	0.8571	0.8738
F1 score for test set	0.8407	0.8461	0.8354

Model 3 – Gaussian Naïve Bayes

GaussianNB uses Bayesian reasoning to develop a set of most likely hypotheses. The 'naïve' in the name comes from the fact that it ignores the inter-dependence between attributes.

Naïve Bayes models tend to work well with text analysis, spam filtering, NLP etc. and it's relatively low complexity and high speed lends itself to real world applications.

Advantages:

- They tend to work well with a small amount of training data and are fast, which plays well into the projects requirement of lower server time.

Disadvantages:

- Attribute independency leads to problems if the outcome is based on their inter-connectedness.

This model was chosen owing to the fact that Gaussian Naïve Bayes works well with a small amount of training data (300 in this case)

	Training set Size		
	100	200	300
Training Time(secs)	0.002	0.001	0.002
Prediction Time(secs)	0.000	0.001	0.001
F1 score for training set	0.3076	0.7985	0.7872
F1 score for test set	0.2790	0.8299	0.8258

5. Choosing the Best Model.

Based on the performed experiments, I choose the **Gaussian Naïve Bayes** model.

This is mainly based on the model's superior balancing act between Prediction time and its F1 scores.

If you compare the Gaussian Naïve Bayes model with the SVM, you can see that although the F1 scores end up being very similar when you consider the entire training set (300 samples), the

prediction time for SVM is almost 6 times that of the Gaussian Naïve Bayes. Minimizing the computation cost for the chosen model is one of the prime objectives of the project and Gaussian Naïve Bayes definitely trumps it in this round.

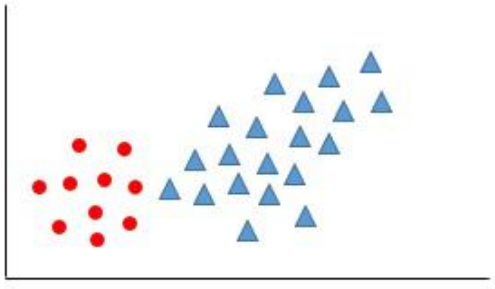
Coming onto the comparison between Gaussian Naïve Bayes and the Decision Tree model, you could say that although the training time isn't the differentiating factor, the F1 score for the test set certainly is (0.7552 vs 0.8258). Also, when you look at the F1 scores for the training set, the Decision Tree model with its score constantly at 1.0 tends to over fit which might be a reason why the performance across the test set precipitously decreases.

A concern with the GaussianNB is how it is thoroughly biased when the data is very small, as seen from the very poor F1 score for the training and testing set. It shoots up exponentially as the data set gets larger

Working of the chosen model

To understand the working of this model, let's take a simple example.
In the banking industry, while giving out credit cards, banks are always looking at ways of effectively distinguishing customers who default and those who don't.

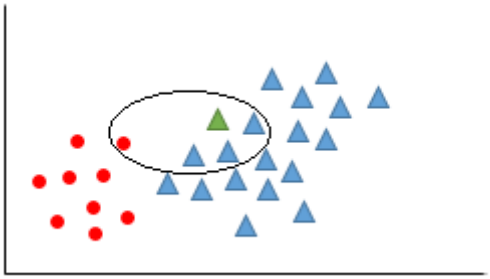
Below is a simple graph that a particular bank has made to distinguish between existing defaulters and good customers.



The blue triangles represent the good customers and red dots, the defaulters.

Intuitively, you can see triangles significantly outnumber the dots. In fact there are twice as many triangles as there are dots (20 vs 10), i.e 20 out of 30 points are triangles (let's call this **A**) and 10 out of 30 points are dots(**B**).

So, without any other information, we assume that the possibility that a new customer would be twice as likely to be good than a defaulter.



Let's introduce a new customer.

In the graph above, the green triangle would be the new customer. As you can see, there are more blue triangles surrounding the new customer than there are dots. Hence there is a better probability that the customer would be good.

To measure that particular likelihood, we have drawn a circle around some points irrespective of whether they are red dots or blue triangles.

There are three triangles around the green triangle and one red dot. Now, we compare these figures to the total triangles or dots i.e, the circle encompasses 3 out of 20 triangles(**C**) and 1 out of 10 dots (**D**).

Taking this into consideration, we multiple A with C to arrive at a figure X and B with D to arrive at a figure Y. We will do this for any new customer who can be placed anywhere within the graph. The higher X and Y is, the better our model will rate and consequently, choose it.

Fine Tuning

In the Gaussian Naïve Bayes model, there are no parameters to tune. Our model then, would be the original model.

Final F1 Score = 0.8258

Bibliography

- http://www.statsoft.com/Textbook/Support-Vector-Machines
- http://scikit-learn.org/stable/
- http://analyticstraining.com/2015/intuition-of-naive-bayes/