Sabastian Mugazambi
HMW 3 : Data Mining Spring 2017

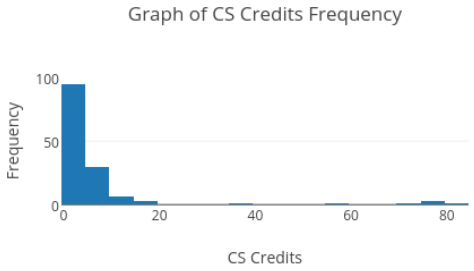
Pre-processing

Reading in data

One of the initials decisions I had to make was that of data representation or data structure for storing the data from the csv file. The first row which represents column names is represented as keys to a *deafaultdict* dictionary with each key pointing to a list which represents the column values for that key factor e.g. {'Cumulative GPA': [3.4, 3.2, 4.0, …….]}. This made it easy to calculate statistics like mean and standard deviation for the factor columns. The initial version of the dictionary included all columns but I removed 'Project ID', 'Minnesota' and 'Birth Year' since I considered then irrelevant to classifying writing abilities.

Discretization/Bins vs Probability Distribution

After analyzing the composition of each column/factor I decided to split the factors into two; Discrete Factors and Continuous Factors. **'International', 'Abroad Credits', 'AP Credits', 'CS Credits', 'English Credits', 'Science Credits'** and **'Writing Credits' where** the factors I decided to discretize. This decision was based on the fact that there are a lot of zeros in these factors and a large part of these zeros is not a representation of poor Credits but rather some students not submitting credits at all. It could be that their form of credits was not accepted. As one can see the large number of zeros distorts any continuous distributions demonstrated in the graph below.



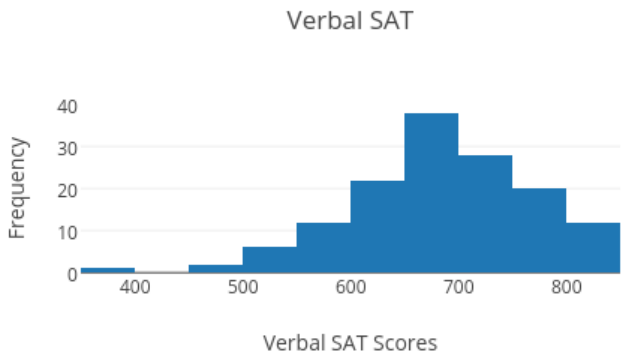
However, **'Verbal SAT', 'Math SAT', 'Cumulative GPA' and '# essays in portfolios'** did not seem to have the same problem and plotting them gave a smooth normal distribution thus I decided to estimate the probabilities using the normal distribution with mean = sample training mean and standard deviation = sample standard deviation.

Cut-off point for discretization

For all the factors that needed to be discretized I put them into two buckets of high and low represented by 0 and 1. The real job was to choose the perfect cut off point for each factor to put the data into optimal bins. I did a lot of research regarding average credits submitted to colleges by students from US high schools. I also calculated averages of Credits after removing international students and zeros. This process informed my selection of cut off points as follows. AP credits have a cut-off of 0 because they are not specific to writing skills.

FACTOR	CUT-OFF
'International'	1
'Abroad Credits'	0
'AP Credits'	33
'CS Credits'	12
'English Credits'	10
'Science Credits'	15
'Writing Credits'	10

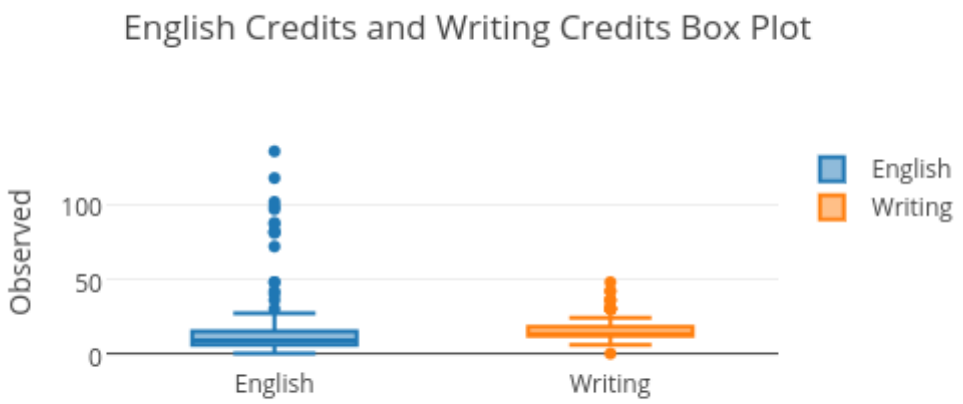
Distribution for continuous



A sample graph of the frequency plot for one of the continuous factors show that the factors chosen to remain continuous follow a distribution like a normal distribution. With the visual cue of the familiar bell shaped curve centered at the mean of the sample we can confidently estimate probabilities using the normal distribution. All factors which remained continuous where modeled as normal distributions with sample mean and sample standard deviation after leave one out cross validation.

With better computation power and complexity, we can analyze factor by factor and choose a more accurate probability distribution for each and everyone one of them. This however takes a toll on our computation time.

Conditional Independence Assumption



The box plot above shows a very close similarity in the credits acquired from English and Credits acquired from Writing. It is easy to see how a high credit in English can determine a high credit score in Writing. It is also not far-fetched to conclude that a student could have high Writing credit because they have a high English credit.

Due to this assuming conditional independence on the factors become invalid. This as a result caused our accuracy to settle at a value around 79% . If the values were truly conditionally independent then the accuracy would have been higher.

Other approaches

It is only fair to measure Naïve-Bayes approach against some other approaches previously implemented. A good example is the K-nearest neighbors approach. The k-nearest neighbors approach could have easily bean used for this classification and I speculate that it could have given a better accuracy due to the continuous nature of all the field.

Naïve-Bayes and K-nearest neighbors are similar in the sense that you use the values of data points that are in the same category, group or bin to inform the prediction of a new data point. Both deal with classifying data into already predefined classes rather than create new classes mid classification. However, they are also different in the sense that K-nearest neighbors uses distance while Naïve-Bayes uses probability to do the predictions for new data points. A summary of the pros and cons can be found in the table below.

K-nearest neighbors		Naïve-Bayes	
PROS	CONS	PROS	CONS
Robust for noisy training data. Use of squared weighted distance eliminates noise.	Complex distance calculations and choices of distance method	If Conditional independence assumption holds, a Naive Bayes classifier will converge quicker than discriminative.	Cannot learn how features are interacting thus performs poorly if conditional independence does not hold
Effective for large training data sets	Does not scale well with the number of factors and complex in picking K and other parameters	It scales linearly with the number of predictors	

Given the analysis of pros, cons similarities and differences one can make informed decisions when choosing the classifying method. If one has a huge training data set with fewer features they could possibly choose K-nearest neighbors due to its effectiveness in large training data sets. However, if the one has a lot of features which are discrete and adheres to the conditional independence assumption they could use the Naïve-Bayes approach. In the end everything relies on a careful choice of parameters and cut-off points.