CT475
Machine Learning & Data Mining

Assignment 2

Name: Taidgh Murray

Student I.D: 15315901

Discipline: College of Science

Course: 4BS2 Undenominated Science (Computing)

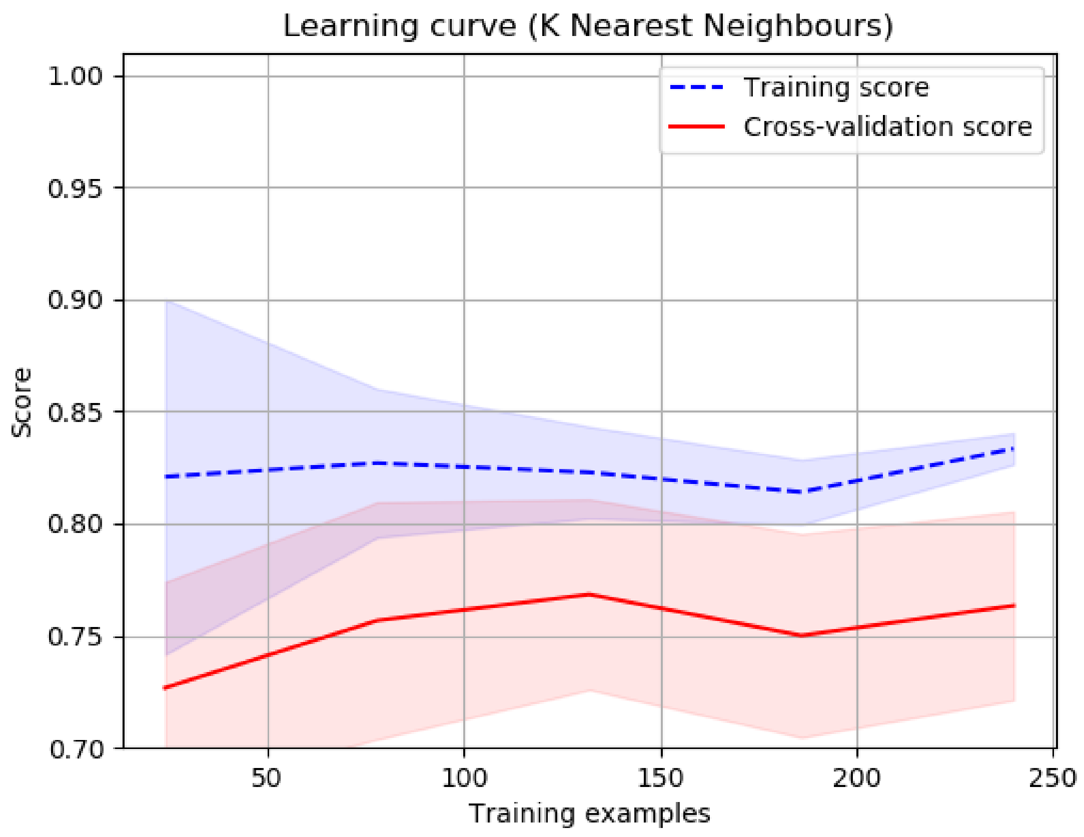
For this assignment, we were asked to construct either learning curves, or ROC curves for the two classification algorithms used in assignment 1.

I used the K-Nearest-Neighbours algorithm, and the Naïve Bayes algorithm. I decided to continue using SciPy, the same package used in assignment 1, it supports both algorithms natively. I decided to plot learning curves, as I felt it was easier to compare and contrast. Learning curves are plots of how accurate a machine learning algorithm, versus how big the training set is. It's designed to show you how much better the algorithm gets at prediction, as more data is introduced.

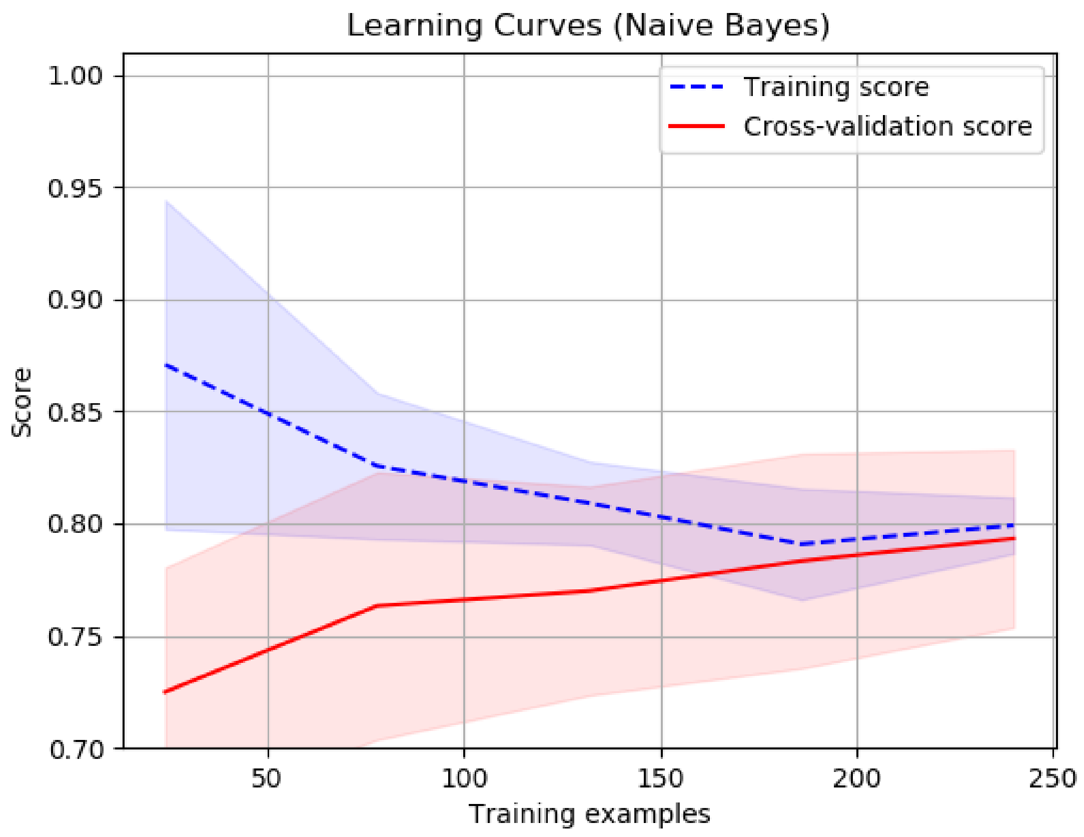
The sciPy package, which includes scikit-learn, includes a learning curve function. However, I needed to write a function to draw a graph for both functions. The function read in the ML Algorithm, and the formatted data from Assignment 1. A 10-fold cross validation was set up to split the data into training and test groups. In general, I tried to keep all data and variables as close as possible to those of semester 1, to allow for ease of comparison. This 10-fold cross validation is run within the function. This info was run through the learning_curve function included with scikit-learn.

The matplotlib library in python handled the actual rendering of the graph, but I felt it worthwhile to also show the standard deviation. Showing this was trivial however, numpy has a function for both the mean and the standard deviation (for reference, the transparent coloured areas in the graphs below). Calculating & drawing the standard deviation 'areas' was also trivial. Matplotlib includes the fill_between function, which, like it says on the box, allows you to fil between two areas.

The function was called, using the two algorithms (Naïve Bayes & KNN) as arguments, and the data as described above. Below are the learning curves returned.



Learning curve (K Nearest Neighbours)
Score
Training examples
Training score
Cross-validation score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
50
100
150
200
250



Learning Curves (Naive Bayes)
Score
Training examples
Training score
Cross-validation score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
50
100
150
200
250

Looking at the first curve, the KNN curve, we can see a number of interesting developments from the curve. In both the Training Score & the Cross-validation score, we can see that overall trend is upwards, indicating that algorithm can still get more accurate.

We can see that the standard deviation is getting smaller as the curve tends left to right, indicating that the algorithm is becoming more accurate over the training examples. As the area looks quite small, one could assume that the function won’t become much more accurate.

It’s interesting to note that the Training Score line & the Cross-Validation line don’t seem to be tending towards one another. This could also be an indicator that there weren’t enough training examples to show where the function becomes most accurate.

The Naïve Bayes curve is much more interesting to look at.

Like the previous curve, the standard deviation is also getting smaller as the curve tends left to right. It’s interesting to note that the area seems to be much larger than in the KNN curve. It’s likely that function can get more accurate yet, provided there are more training samples used.

It’s surprising to see that the accuracy score is going down for the training score, and up for the cross-validation score. It could be a case that the algorithm is most accurate at the start, and immediately becomes inaccurate. It could also be the case that the training data set is affecting the testing data set.

Most interesting to see is that the Training Score, and the Cross-Validation score are tending towards one another, unlike the KNN graph. This could mean that the amount of training samples provided are most likely adequate to showcase the function.