CAB320 – Assignment 2
Machine Learning

Report

Submitted By:
n9400001 Moira Quinn
n7226209 Maurice Cafun
n9708651 Christopher O'Rafferty

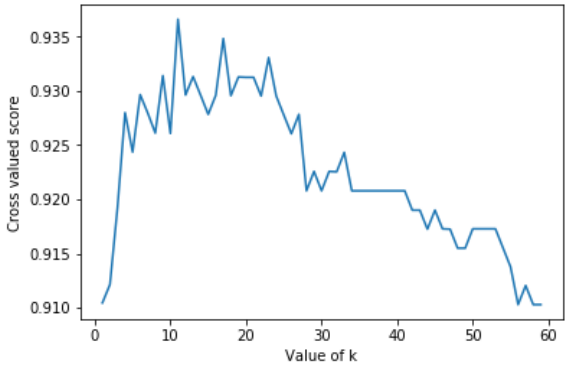
04 June 2017

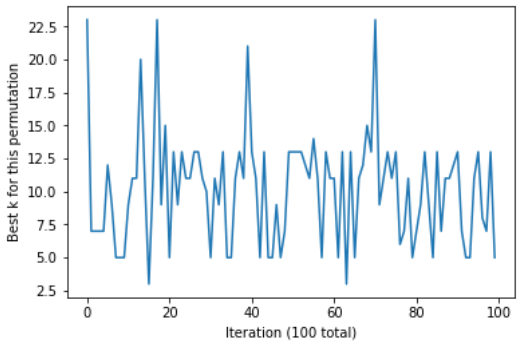
Section 1: Optimising the classifiers

1.1 Nearest Neighbours Classifier

In the Sklearn Library, the K-Nearest Neighbours classifier has a parameter called k, which refers to the number of nearest neighbours to be used for classification. This parameter can influence the score that the classifier will achieve when tested against data that is tested with it. By using cross-validation, we can optimise the k parameter by testing a range of numbers, and determining which has the highest score. This means that the final classifier to be returned by the function can have the most optimised k parameter for the dataset being tested.

By using the test data supplied, "medical_records.data", we were able to find the most optimised k value for many permutations of the data. We decided to choose the largest k value that gives the highest score, meaning that if two k values give the same highest score, the larger k value will be chosen. This is to suppress the effects of noise.



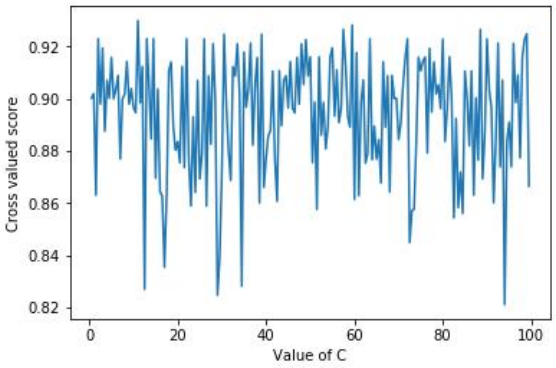


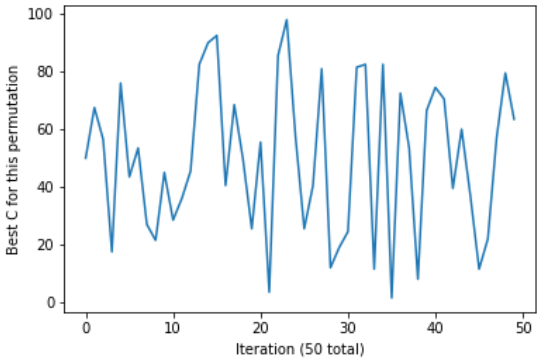

```
The average best k value is: 10.19
```

The graph on the left shows the highest scores for this permutation are given by k values around the range 5 – 20, with the best value being 11. The graph on the right shows the trend of the best k-values for 100 random permutations of this dataset. The best average k value for our dataset is 10.19, which rounds to 10.

1.2 Support Vector Machine Classifier

In the Sklearn Library, the linear SVM classifier has a parameter called C, which is a penalty parameter. We attempted to optimise this parameter by testing a larger range of values, however our graphs returned were very similar for our data. We tried small numbers, going from 0 to 1, stepping at 0.01, and large numbers from 0 to 50 million stepping at 10,000, however there was no clear trend to the effects of the c-value for our data, so in the end we decided to test values from 1 to 100 stepping by 0.5, so that 200 tests occurred in total. The C value that gives the highest score is deemed the best, so it is used to build the final classifier.





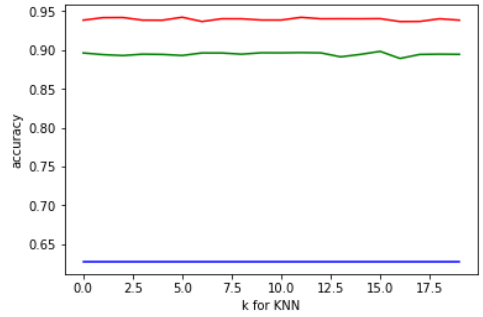

```
11.0
```

```
The average best C value is: 49.89
```

The graph on the left shows the scores given by this range of k values is similar in the range 0 to 100, which we found to be true for small values (0 to 1) and large values which we tested up to 50 million, with the best value in this case being 11. The graph on the right shows the trend of the best C-values for 50 random permutations of this dataset. The best average C value for our dataset is 49.89, which rounds to 50. This also shows that the best C is random (or at least seems to be), as it averages exactly halfway which is what you would expect random numbers to do.

1.3 Naïve Bayes Classifier

In the Sklearn Library, the Naïve Bayes classifiers do not have parameters that you modify to increase the efficiency and results of the scores. We still decided to slightly optimise the function, however, by determining which of the three classifiers (Gaussian, Multinomial and Bernoulli) gives the best average result for many random permutations of the inputted data.




```
0.940431898712
```

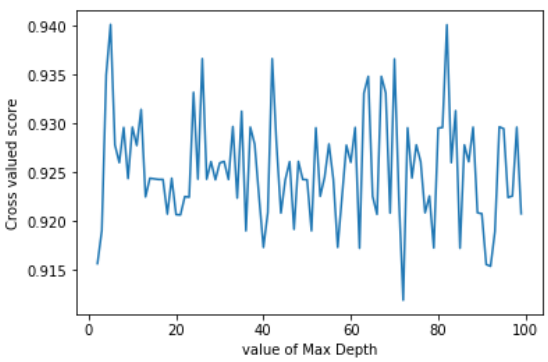
Gaussian, Multinomial and Bernoulli

In this graph, the red line represents Gaussian, the green represents Multinomial, and the red represents Bernoulli. It is obvious by the results that Gaussian is the best fit for our data. This also produced a consistently high score (in this case about 0.94).

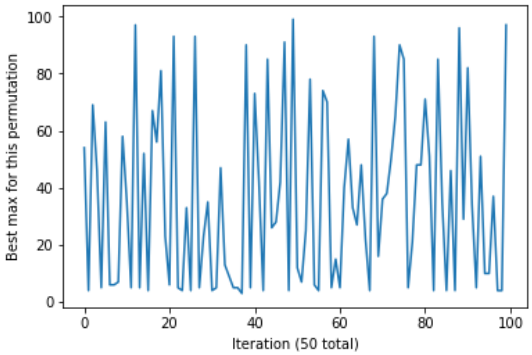
1.4 Decision Tree Classifier

In the Sklearn Library, the Decision Tree classifier has a parameter called max Depth, which is a parameter that controls the size of the tree to avoid overfitting. This was chosen over the min samples split as it gave much better results. The classifier does use random values, so results weren't always accurate after fitting the data with the best max depth found. We ended designing the

function to test all integers from 2 to 100, and choosing the max depth value that produced the highest score to construct the final classifier.



5

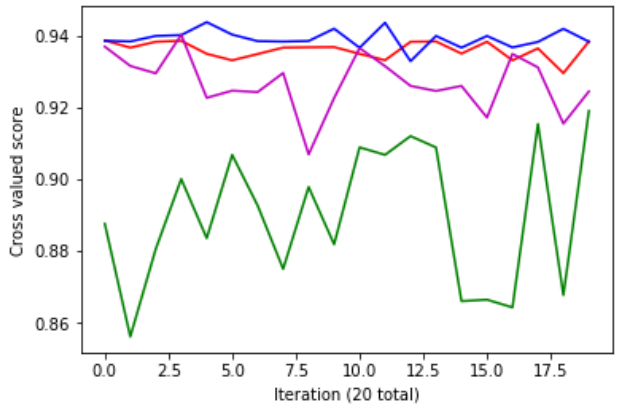



```
The average best max value is: 36.1
```

The graph on the left shows the scores given by this range of max depth values is similar in the range 0 to 100, we found true for values up to at least 4000, with the best value in this case being 5. The graph on the right shows the trend of the best C-values for 100 random permutations of this dataset. The best average C value for our dataset is 36.1, which rounds to 36. This shows that whilst the results are somewhat random, there is a small trend where lower values are better.

Section 2: Comparing the Classifiers

To compare all the types of classifiers, we decided to generate 20 different permutations of the data and feed them to each classifier, we would then plot the results on a graph and find the average score that each classifier produced. The results are as follows:




```
Averages:
knn: 0.936157203353
svm: 0.889910228157
nb: 0.939275505574
dt: 0.926932903379
```

Naïve Bayes, Nearest Neighbours, Support Vector machine and Decision Tree results.

From these results it is clear that Naïve Bayes is the best scoring classifier for this data set, followed by the K-Nearest neighbours classifier, then the Decision tree classifier, and finally the Support vector machine. All of these classifiers scored very highly, with the average being very close to or above 90%. This shows that optimising these classifiers for the inputted data certainly helps increase their accuracy at predicting and classifying new data.