1 Introduction

In this experiment, I implemented three classifiers which are; a decision tree, random forest and K-nearest neighbours. All three classifiers were optimised and evaluated on the UCI breast cancer dataset available in the sklearn library. For each classifier, I optimised the performance by doing a 5 k-fold cross validation on each one with a range of parameters and selecting the parameters that produced the best median accuracy. Finally, I then evaluated and compared the performance of each classifier with the optimised parameters on an unseen test set, to find the the classifier that performs best on the UCI breast cancer data set.

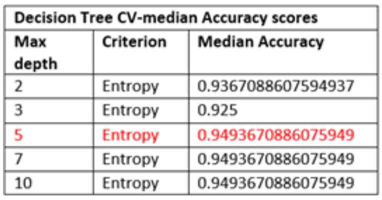
2 Methods

The first step I took was to split the breast cancer data set into 70:30 train_validation and test split using the train_test_split method in SKlearn.

Next, I moved onto the optimisation of parameters for each classifier. For this I had nested for loops that ran through all combinations of parameter values for the particular classifier that were given. For each combination of parameter values, I then used the cross_val_score method in sklearn on the classifier. I used a k-fold object with 5 splits for this and used the train_validation dataset. This produced 5 accuracy scores for that specific classifier parameter set up. I then found the median accuracy of the 5 accuracy scores by using the numpy median method. Next I appended the median accuracy for that set of parameters into an array. To keep track of the best median accuracy score and parameter values for each classifier. I initialised variables, one being the best median accuracy score and the others being the best combination of parameter values. For each median accuracy produced, I would then compare to see if this was better than the current best median accuracy score and if it was, I would then update the best median accuracy score, to the new best median accuracy score and update the best parameter values to the ones that the classifiers used. This allowed me to find out the best parameters to use to get the most optimised classifiers.

The next step that I took was to test the three classifiers on the unseen test dataset (this is called generalisation) to get its predicted accuracy. To do this I built three new classifiers with the best parameter values that I obtained from the optimisation previously in the experiment. For this I built a Decision tree classifier with a max_depth vale of 5 and a criterion of "entropy", A random forest classifier with a n_estimators value of 100 and random_state of 1 and finally a K-nearest neighbours' classifier with a k value of 21 and the distance metric being "Manhattan". Next, I trained all the classifiers on the train_validation set and then made the predictions for the test_set using the predict method available in sklearn. I then stored all the predictions in an array. Finally, I got the accuracy of each classifier by using the accuracy_score method in sklearn. This method takes two inputs, one being an array of your predictions and the other an array of the actual class labels, which I stored in y_test. The method then returns the accuracy score for your classifier on the predictions you passed it.

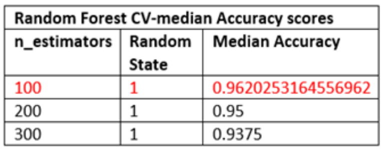
3 Results



Decision Tree CV-median Accuracy scores		
Max depth	Criterion	Median Accuracy
2	Entropy	0.9367088607594937
3	Entropy	0.925
5	Entropy	0.9493670886075949
7	Entropy	0.9493670886075949
10	Entropy	0.9493670886075949

Figure 1: Decision Tree optimisation median Accuracy Results

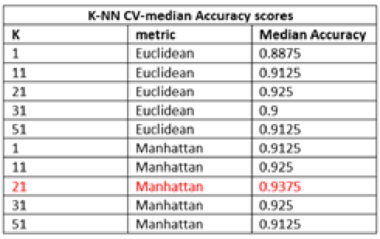
The results from the cross validation of parameters for each classifier showed that best parameters to use for the decision tree classifier was a max_depth value of 5 and then a criterion of "entropy". Using this set of parameters produced a median accuracy of 0.949 (to 3.d.p) on the train_validation set, which can be seen in Figure 1 and 5. Both max_depth values of 7 and 10 also achieved the same median accuracy as the max_depth value of 5. However, I selected 5 since it was achieved first and would take less time to build due to a smaller max depth.



Random Forest CV-median Accuracy scores		
n_estimators	Random State	Median Accuracy
100	1	0.9620253164556962
200	1	0.95
300	1	0.9375

Figure 2: Random Forest Optimisation median Accuracy Results

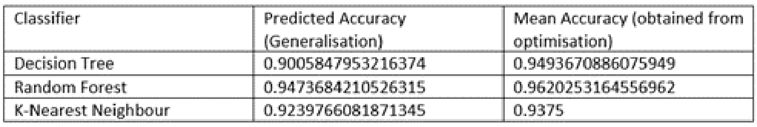
For random forest the best parameter set up that was produced by the cross validation was an n_estimators value of 100 and the random_state value set to 1. Using this configuration achieved a median accuracy of 0.962(to 3.d.p) on the train_validation set. The median accuracy scores for all the combinations of parameters can be seen in Figure 2 and 6.



K-NN CV-median Accuracy scores		
K	metric	Median Accuracy
1	Euclidean	0.8875
11	Euclidean	0.9125
21	Euclidean	0.925
31	Euclidean	0.9
51	Euclidean	0.9125
1	Manhattan	0.9125
11	Manhattan	0.925
21	Manhattan	0.9375
31	Manhattan	0.925
51	Manhattan	0.9125

Figure 3: K Nearest Neighbours Optimisation median Accuracy Results

Finally, the optimisation of parameters on the K-nearest neighbours classifier showed that the best parameter set up was a K value of 21 and the distance metric being "Manhattan". The median accuracy this K-nearest neighbours classifier achieved on the train_validation was 0.9375. This accuracy score was 0.0125 (1.25%) better than second best median accuracy achieved by another set of parameters. The median accuracy scores for all the combinations of parameters can be seen in Figure 3 and 7.



Classifier	Predicted Accuracy (Generalisation)	Mean Accuracy (obtained from optimisation)
Decision Tree	0.9005847953216374	0.9493670886075949
Random Forest	0.9473684210526315	0.9620253164556962
K-Nearest Neighbour	0.9239766081871345	0.9375

Figure 4: Evaluation of predicted accuracy for each classifier on their optimised parameters

4 Discussion

The predicted accuracies that I got for all three of the classifiers can be seen in Figure 4. The results of this showed that the Random Forest classifier had the best predicted accuracy out of the three, with an accuracy of 0.947 (3.d.p). This was 0.023 (2.3%) better than K-NN, which was second best with 0.924 (3.d.p) accuracy and then also 0.046 (4.6%) better then the decision tree classifier, which had an accuracy of 0.901 (3.d.p). Comparing each of the classifiers predicted accuracy scores on the test data set, to their optimisation accuracy scores on the train_validation set. All the optimisation accuracy scores on the train_validation set achieved higher accuracy then predicted accuracy scores on the test set, this was expected since it had been trained on this data whereas it had not seen the test data for generalisation. The results of these both can be seen in Figure 4 once again. One result that should be mentioned is that on the train_validation set the decision tree classifier was the second-best performing classifier. However, on the test set the K-NN classifier did in fact perform better and achieved a higher accuracy then the decision tree classifier during generalisation.

5 Conclusion

In conclusion, the best classifier to use on the UCI breast cancer dataset is the Random Forest classifier with the a n_estimators parameter value of 100 and a random state of 1, as it achieved the highest accuracy out of all three classifiers on both the train_validation and test set accuracies, with accuracies of 0.947 and 0.962. Then followed by K-NN classifier and finally the worst performing classifier was the decision tree classifier.

A Graphs

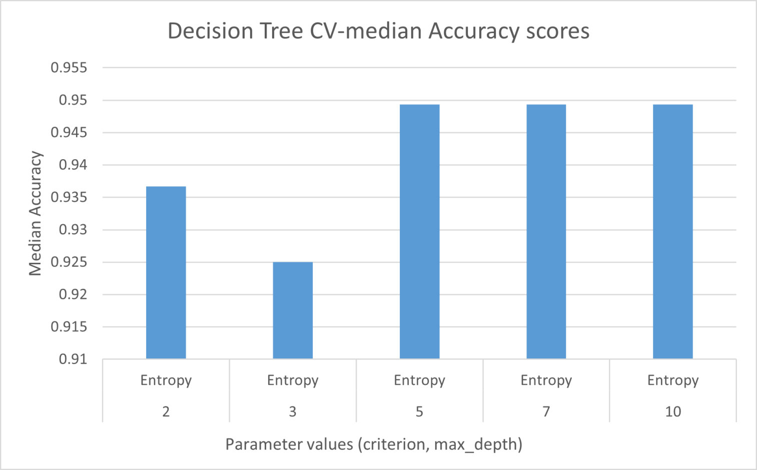


Figure 5: Graph showing the median accuracy of the Decision Tree cross validation parameter optimisation

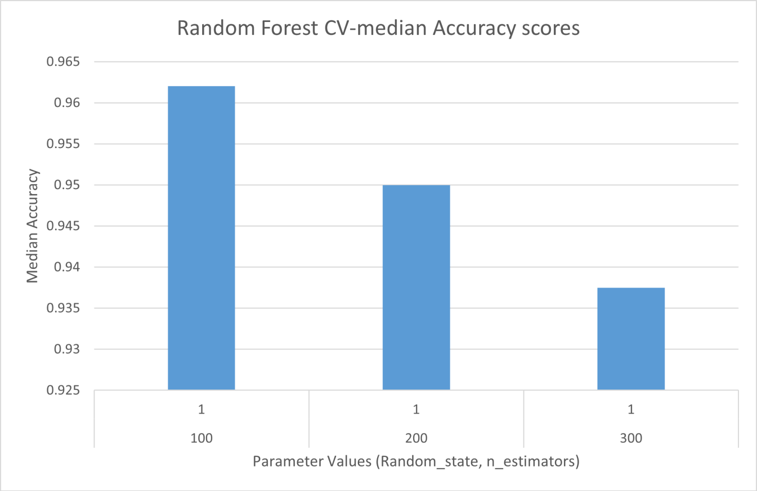


Figure 6: Graph showing the median accuracy of the Random Forest cross validation parameter optimisation

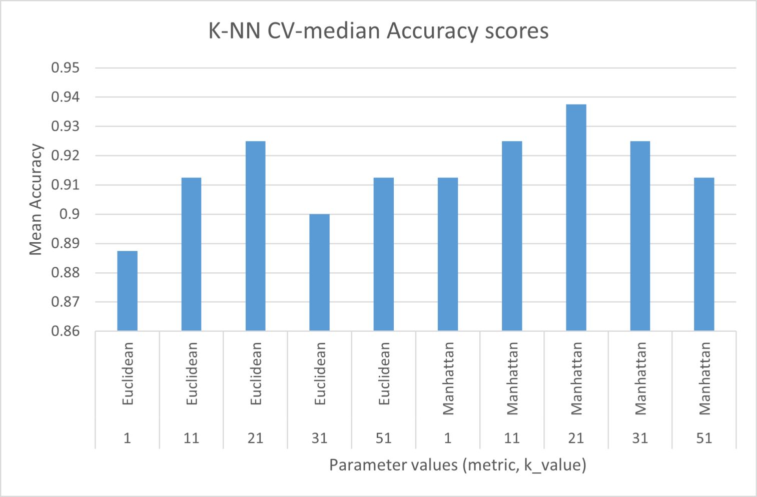


Figure 7: Graph showing the median accuracy of the K-NN cross validation parameter optimisation