Supervised Learning Methods in Machine Learning

Darsh Thakkar

Introduction

This report contains analysis of comparison of five different supervised Machine Learning algorithms, namely – Decision Trees, Boosting Classifier, K-Nearest Neighbor, Support Vector Machine and Neural Networks. All the algorithms are applied on two different datasets and deal with classification problems for both the datasets. Each algorithm has its own section within which there are two separate section for results of respective algorithms on the two different datasets. These results contain graphs which depict relevant information, how hyper parameters are tuned and the performance of the algorithm. All the algorithm performances are compared and discussed on the classification problem datasets in the end of the report.

Datasets

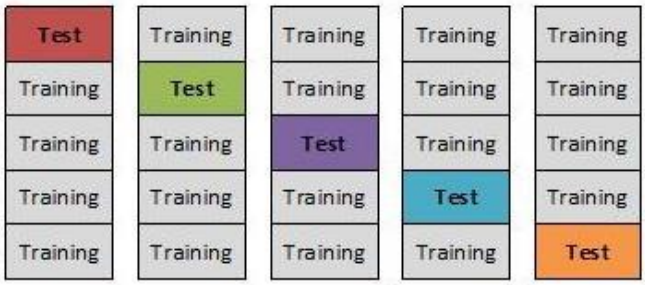
The classification problems that are selected are openly available on the internet and various features of the datasets have been given numerical encoding so that these datasets can be fed as an input to machine learning libraries in Python 3. Datasets are retrieved through the basic *pandas.read_csv( )* function except for neural networks as it required reshaping of data in order to apply one hot encoding. Following are the datasets used for this assignment:

- Breast Cancer Dataset- The dataset contains 9 features which describe the mass to classify whether breast cancer exists or not and thus it is as binary classification problem. The size of the dataset is 684 samples and is available on Kaggle, it is provided by Wisconsin breast cancer cytology report. This particular dataset is a cleaned dataset and thus good results can be expected because of this. Also, there are very clear parameters for classification, hence we can expect KNN and SVM to give better results for this dataset. One feature in the dataset “ID” is not useful for classification and thus it is omitted from all calculations.

- White Wine Quality Dataset- Size of the dataset 4879 samples and 11 features which give description and data required by the classifier to produce quality of the wine (10-outstanding, 1-very poor). Features of the dataset are continuous and hence it is interesting as it will require the values to be discretized for the classifier. Some labels have no or very few samples in the dataset and thus the algorithms will need to adapt to this situation. Moreover, there are many features and all of them almost equally important for classification and therefore clustering algorithms like KNN may perform poorly.

Implementation

Algorithms are implemented by using the traditional scikit-learn packages library in Python language. Neural network is implemented using tensorflow and Keras libraries as well. Sciikt-learn provides machine learning algorithms that can be directly called using functions and passing relevant parameters. For the evaluation purpose and to tune hyper parameters, the datasets are split into two sets: 80% training set and 20% testing set. *cross_val_score( )* is utilized with *cv* parameter equal to 5 with all algorithms. Following diagram depicts how the dataset is divided in cross validation. The mean of all scores obtained in each of the five tests is the cross-validation score.



Learning curve is provided for each algorithm, where the converging behavior of the algorithms can be observed.

Decision Tree

Algorithm

Decision tree has been implemented using the sklearn DecisionTreeClassifier with GINI index as the splitting criteria. Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. The Gini impurity can be computed by summing the probability Pi of an item with label i being chosen times the probability 1 - Pi of a mistake in categorizing that item. It reaches its minimum (zero) when all cases in the node fall into a single target category.

Pre-pruning techniques is used to implement pruning, this is where we limit the size of the decision tree, in order to consider only important factors for classification and to reduce time taken by the classifier to give output.

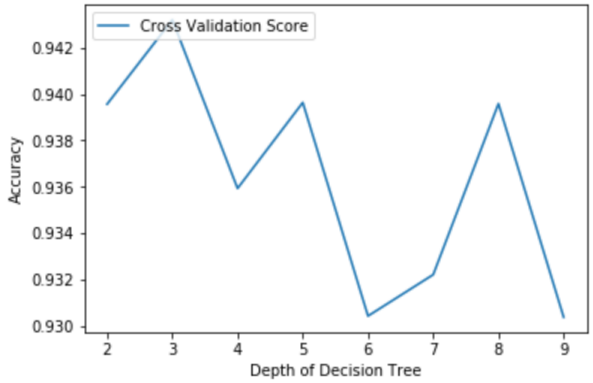
Cross validation scores are recorded for different depths of the decision tree and also the decision tree training and testing errors are observed over different depths. Based on the highest cross-validation score, depth is chosen and then learning curve is generated for that depth.

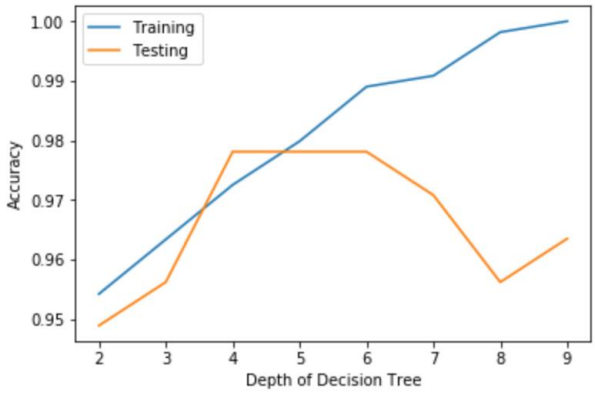
Observations and Analysis

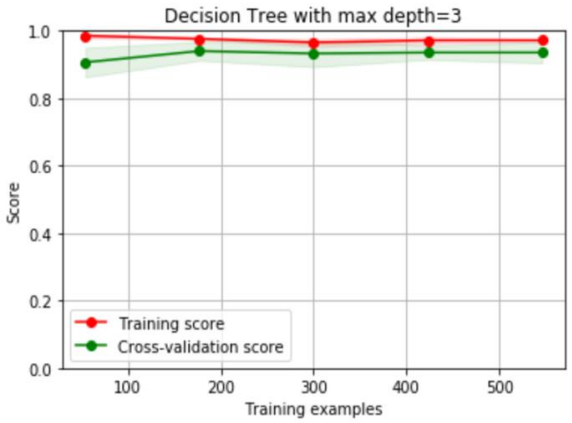
Breast Cancer

The breast cancer dataset shows the best CVS score is achieved at the depth of 3. However, when comparing the test and training accuracy, testing accuracy is highest at the depth of 4. This may be due to the fact that some factors are more important in determining the presence of breast cancer than others. Moreover, the testing accuracy drops as the depth is increased, this is very good as it displays the overfitting behavior of the algorithm on the training data. Thus, pruning the tree helps in increasing testing accuracy.

```
Best CVS 0.9432008154943935 3
```





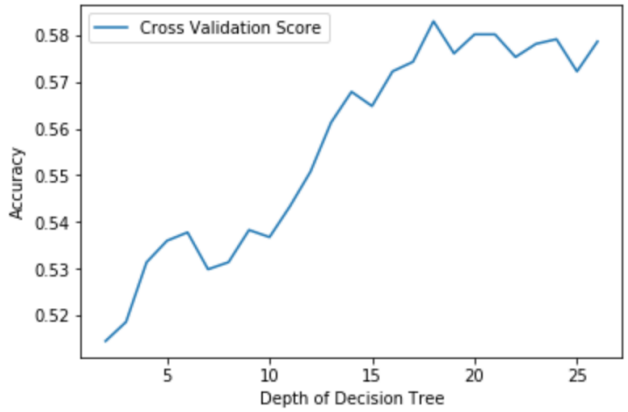


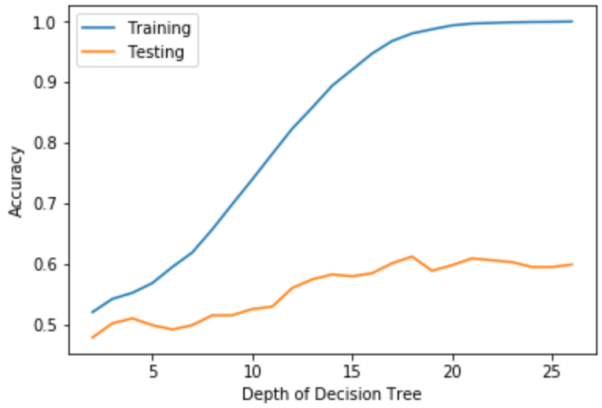

```
Time take to train Decision Tree is 0 ms
The training accuracy of Decision Tree is  0.9633699633699634
The testing accuracy of Decision Tree is 0.9562043795620438
```

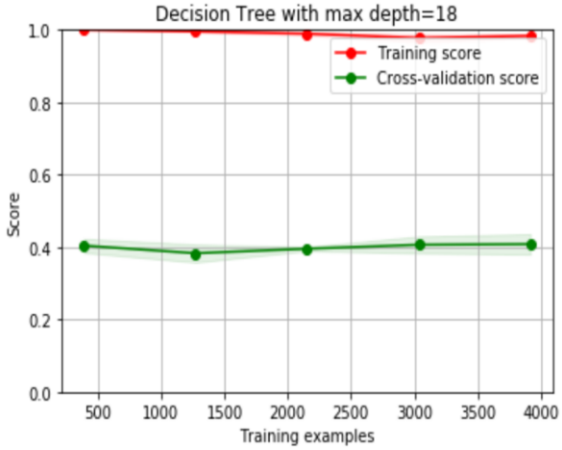
Wine Quality

For the wine quality dataset, CVS is highest for a depth of 18. By creating a graph of depth with training and testing accuracy, we are able to analyze that both training and testing accuracy become stable after a particular depth and hence there is no need to check for more depths. Unlike breast cancer dataset, lower depth of decision tree does not help in this case, this is because breast cancer is a binary classification problem and also has less number of features compared to wine dataset, also for wine dataset, most features are equally important.

```
Best CVS 0.5829414078288229 18
```








```
Time take to train Decision Tree is 38 ms
The training accuracy of Decision Tree is  0.9808575803981623
The testing accuracy of Decision Tree is 0.5938775510204082
```

Boosting Classifier

Algorithm

Ensemble methods, which combines several decision trees to produce better predictive performance than utilizing a single decision tree. The idea behind the ensemble model is that a group of weak learners come together to form a strong learner. The classifier used for this project is provided by sklearn library -GradientBoostingClassifier. The gradient Booster creates decision trees iteratively that on each iteration assign various weights to the samples depending on how useful they are to the prediction.

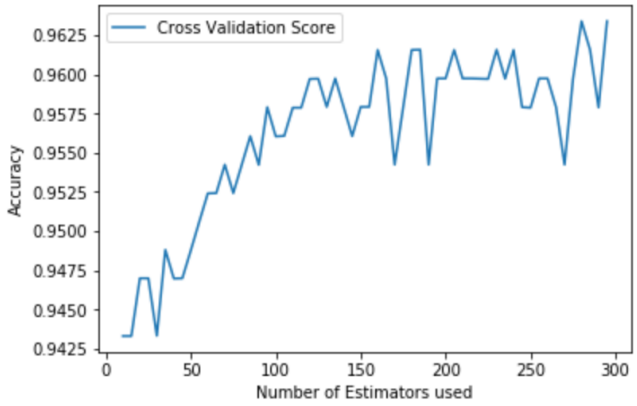
Gradient boosting uses a large number of estimators, in this case decision trees and thus the hyper parameter tuned here is the number of estimators. Pre-pruning of individual estimators has been done by limiting the depth of individual decision trees to 5.

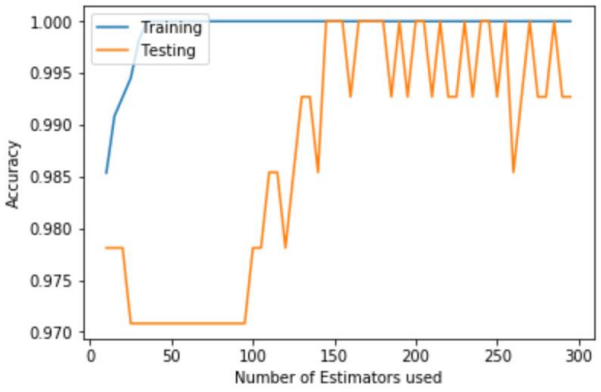
Observations and Analysis

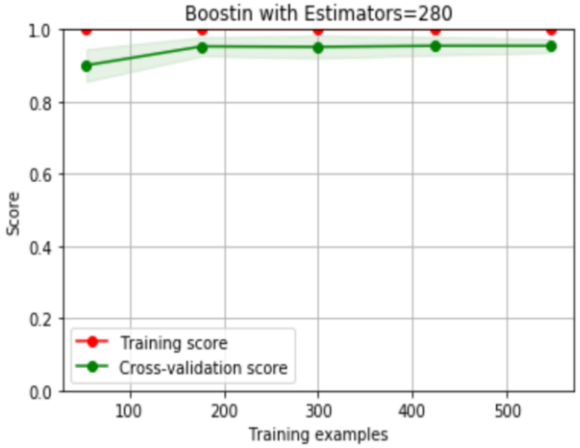
Breast Cancer

In this case there is an increase in accuracy but already ~95% accuracy was achieved with decision trees. However, this algorithm takes more time to run as compared to a decision tree which is an expected result. As we increase the number of estimators the training accuracy reaches 100% and interesting the testing accuracy follows a periodic behavior between 98.5-100%. The highest CVS is achieved at 280 estimators but also 100% testing and training accuracy is achieved at 150 estimators. Thus, not using CVS to choose hyper parameter, can make the model not so reliant for future data and that is why CVS is used to select the estimator value.

```
Best CVS 0.9634022178976307 280
```







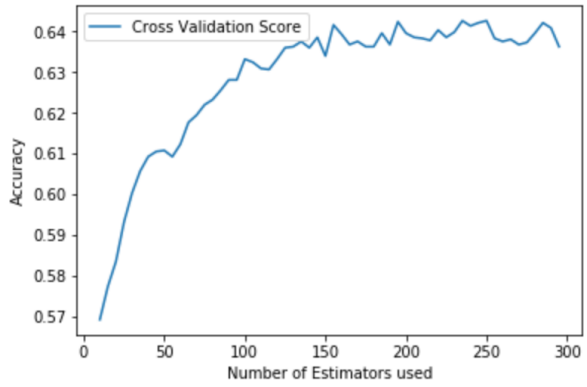

```
Time take to train Decision Tree is 217 ms
The training accuracy of Decision Tree is  1.0
The testing accuracy of Decision Tree is 0.9927007299270073
```

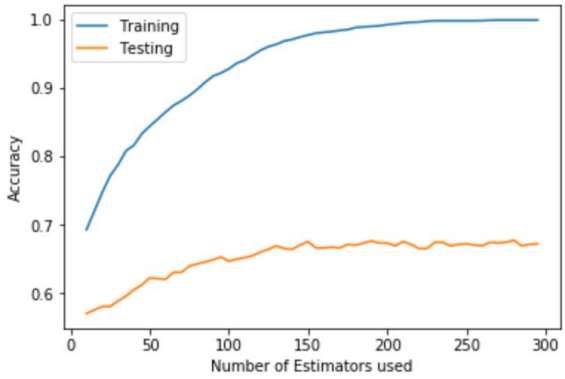
Wine Quality

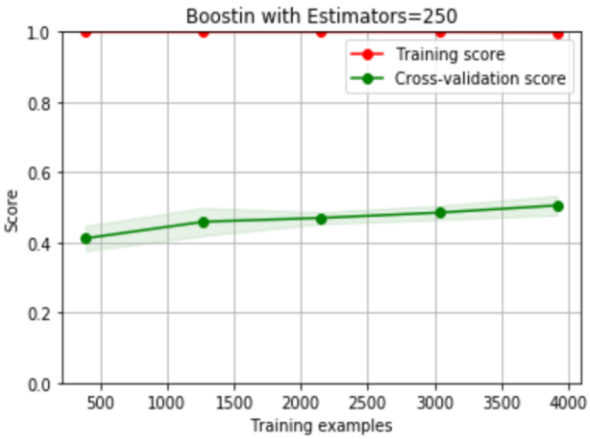
From the graphs shown below, it is evident that gradient boosting gives better results than a single decision tree for this dataset. The accuracy increases from 59% in decision trees to 67% in Gradient Booster Classification for Wine Quality Dataset but again the tradeoff is that this method takes more time then decision trees. Also, the testing accuracy levels off at around 150 estimators and there also

isn’t a significant change in CVS after that. Thus, taking number of estimators as 150 instead of 250 would result in similar results with less time than 150 estimators.

```
Best CVS 0.6426637355402179 250
```








```
Time take to train Decision Tree is 8255 ms
The training accuracy of Decision Tree is  0.998468606431853
The testing accuracy of Decision Tree is 0.6724489795918367
```

K-Nearest Neighbor

Algorithm

The KNN algorithm implementation is from the sklearn library KNeighborsClassifier. Here the value of K can be set as an argument. The classifier runs the KNN algorithm and then tries to match the label values with the cluster that are in. KNN in its base implementation is used for clustering but the sklearn library transforms it so that it can be used as a classifier too.

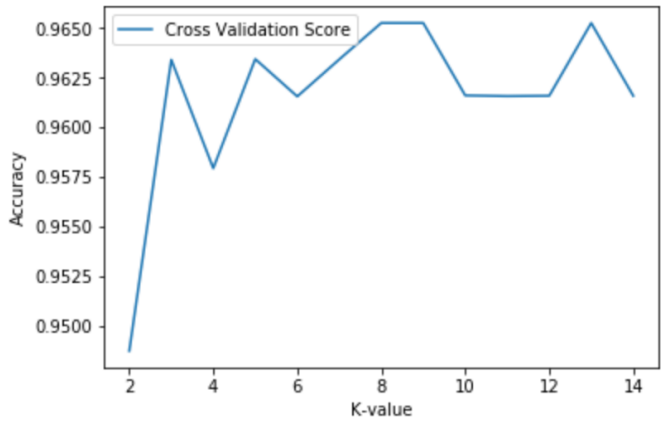
The hyper parameter which needs to be tuned for this algorithm is the value of K. Depending on the CVS for different values of K. The best and most reasonable value of K can be chosen

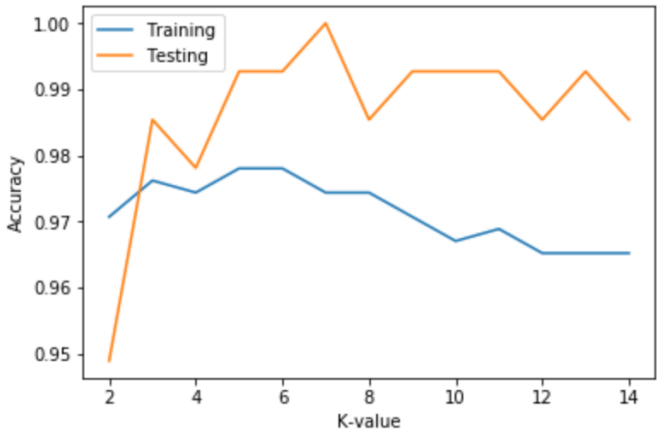
Observations and Analysis

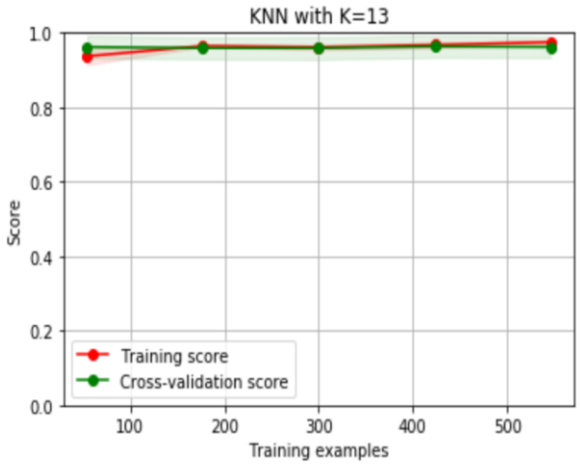
Breast Cancer

Breast Cancer classification is a Binary classifier problem and thus the expectation would be that K=2 will lead to the best results. However, the CVS score as well as testing accuracy is the minimum for K=2, which is very unexpected. Maximum CVS score is achieved for K=8 as well as K=13, where K=13 has slightly higher CVS score. When compared to testing accuracy though, it is better for K=7 but with a very minute difference and therefore K is chosen on basis of CVS score as it is more reliant. Time taken to run KNN is very small for this dataset, mainly because of less size and cleaner data. Test accuracy is higher than train accuracy very minutely in probably because very less test samples are present.

```
Best CVS 0.9652540697494825 13
```







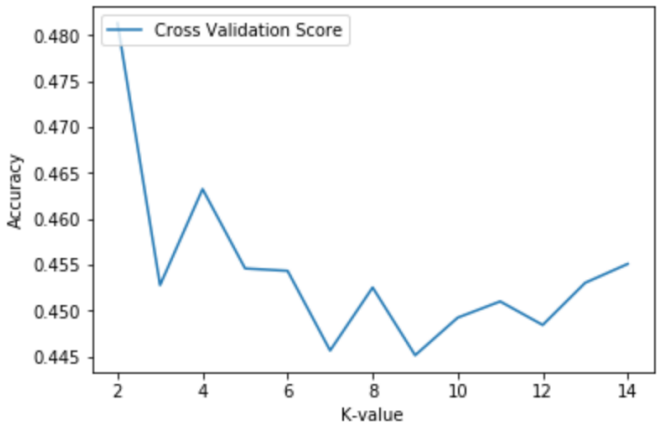

```
Time taken to train KNN is 0 ms
The training accuracy of Decision Tree is  0.9652014652014652
The testing accuracy of Decision Tree is 0.9927007299270073
```

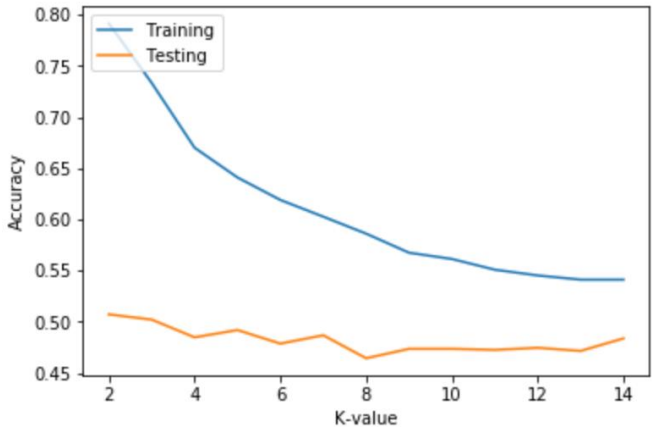
Wine Quality

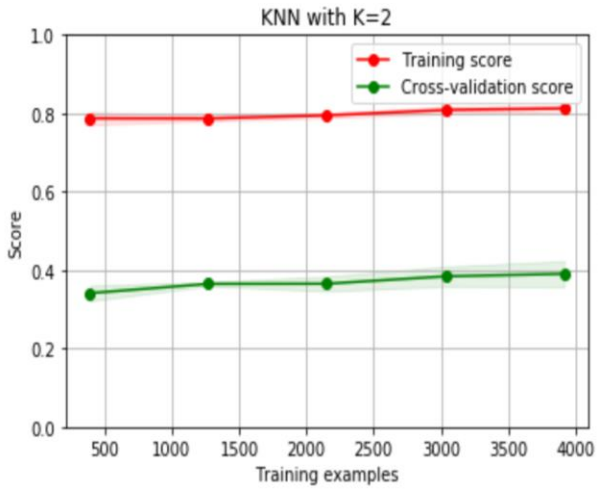
For Wine Quality Dataset, I was getting Highest CVS score with Testing accuracy=65% for K=1. I realized that this was because the algorithm was keeping the rows with the wine quality whose entries are highest (here wine quality=6) in one cluster and everything else not part of the cluster. However, the K-NN cannot work with K<2 as this is the flaw and it will not make the algorithm work better for future data.

Subsequently, highest test accuracy and CVS score was achieved with K=2 because most of the examples in the dataset belong to the class 5 or 6 (that is, wine quality =5 or wine quality=6) and thus

the algorithm only classified those two labels correctly and disregarded other labels. Therefore, the testing accuracy is also ~50%, which is very poor compared to the previous two algorithms and was expected for KNN in this particular dataset.








```
Time taken to train KNN is 5 ms
The training accuracy of Decision Tree is  0.7914752424706483
The testing accuracy of Decision Tree is 0.5071428571428571
```

Neural Networks

Algorithm

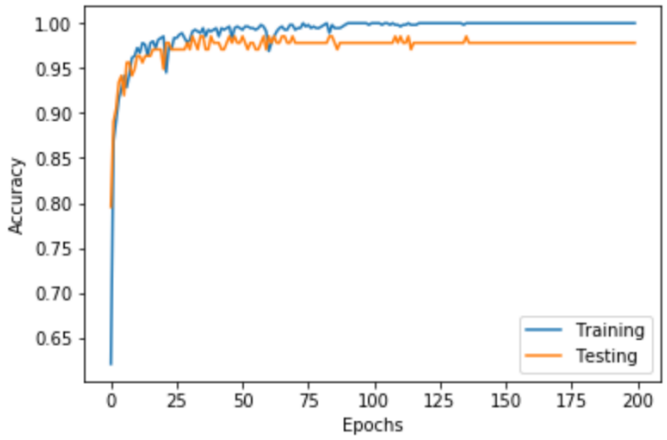
The neural network classifier has been implemented using the Tensorflow and Keras framework in Python 3. The model uses back propagation with batch gradient descent to update weights and learn the parameters. The output labels have been converted into a one hot vectors and then are being fed into the model and the last layer of the model has a softmax function which is often the best choice for multiclass classification. The network architecture has two hidden layers with 128 and 64 units respectively. This network was found to give the best results after checking empirically.

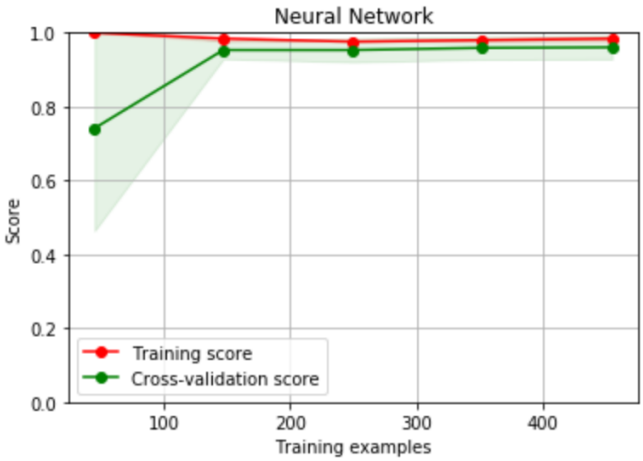
Here the graph of testing and training accuracy is observed with number of epochs, in order to decide for how many epochs will lead to a good stable testing accuracy.

Observations and Analysis

Breast Cancer

For breast cancer dataset, number of epochs are cut off at 200 as both training and testing accuracy becomes constant after that. The accuracy reached by neural network is 98.5% and has very small difference then training accuracy (100%), hence the model does not over fit the data even when number of epochs are higher than when the accuracies stabilize. One interesting observation is that the time taken for running neural network is far more than the previous algorithms for this dataset and also similar or better results were obtained with those algorithms, thus concluding that it is better to use other algorithms for this dataset compared to neural network.

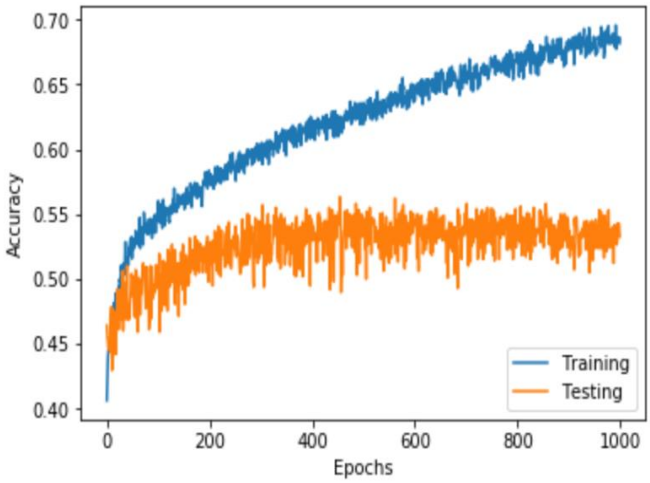


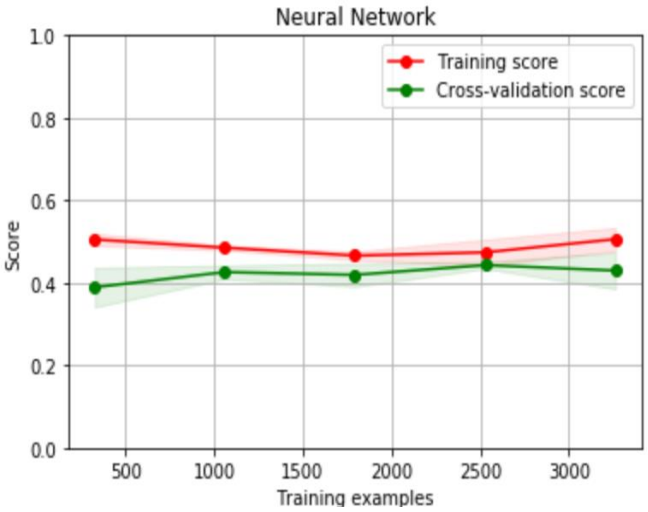



```
Time taken to train Neural Network is 10315 ms
The training accuracy of Neural Network is  1.0
The testing accuracy of Neural Network is 0.9854014598540146
```

Wine Quality

The Wine quality dataset was run for 1000 epochs to check the models performance and the classifier reaches maximum accuracy up to 56.3% but at the same time the time taken by the classifier is significantly higher than the other algorithms, Boosting Classifier provided better accuracy in less time for this dataset. The accuracy changes on each iteration because batch gradient is used. Other network architectures were tried but there was no significant change in accuracy, this is probably because the algorithm needs more variant data or more features to accurately classify the data. Interestingly, after 200 epochs the testing accuracy of the model becomes constant but the training accuracy of the model keeps increasing, this is probably because the model is over-fitting on the training data, however it has no result on the testing accuracy. This is again probably because the model is not learning anything significantly new which reflects on the testing data.






```
Time taken to train Neural Network is 273719 ms
The training accuracy of Neural Network is  0.6957631444310338
The testing accuracy of Neural Network is 0.5632653060008068
```

Support Vector Machines

Algorithm

The SVM classifier given by the sklearn package is used to implement this part of the algorithm. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. The points that are the boundary of this gap are the support vectors. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In the project, StandardScaler functionality of sklearn is also used to normalize the data as this helps the learner in classification. If not used, the time required to converge to a local minima is greatly more.

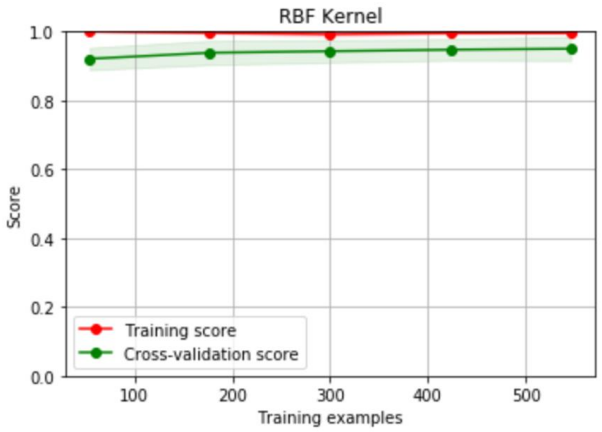
Depending on which kernel used for SVM, the accuracy varies significantly. Three popular kernels are thus used to evaluate performance of SVM – Linear, Polynomial and Gaussian (RBF).

Observations and Analysis

Breast Cancer

For the Breast Cancer dataset, the Gaussian Kernel gives the best performance with testing accuracy ~97.5% as this kernel does not require space to be separable by a polynomial. Moreover, the polynomial kernel gives the poorest result comparatively ~94.1%, which is worse than linear kernel (~96%), this is probably because the polynomial kernel tries to over fit the training data which affects the testing accuracy. The linear kernel however is not affected by this as it is more general and does not

fit to the preciseness of training data. Time taken by SVM is very much comparable to other algorithms which work well with this dataset, as expected.

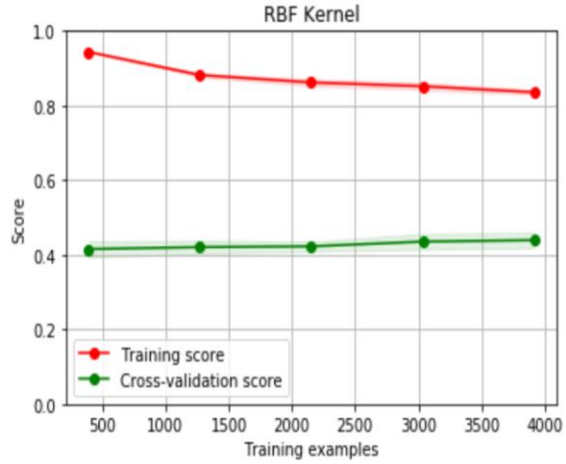



```
Time take to train SVM Classifier is 3 ms
The training accuracy of SVM Classifier is  0.9790794979079498
The testing accuracy of SVM Classifier is 0.975609756097561
```

Kernel	Linear	Polynomial	RBF
Testing Accuracy	96.09%	94.14%	97.56%

Wine Quality

This dataset is not a binary classification problem and thus the linear kernel does not do well, also the examples cannot be separated by a polynomial function which gives a good accuracy. The best test accuracy is achieved through RBF which is ~55.3% but this is still ot comparable to ~67% achieved through boosting classifier for this dataset.




```
Time take to train SVM Classifier is 783 ms
The training accuracy of SVM Classifier is  0.6202143950995406
The testing accuracy of SVM Classifier is 0.5530612244897959
```

Kernel	Linear	Polynomial	RBF
Testing Accuracy	50.71%	51.2%	55.3%

Comparison and Analysis of Algorithms

Below is the performance comparison of all algorithms together. This shows that for the given datasets, Boosting Classifier and KNN both will get the best results for Breast Cancer Dataset. Whereas for Wine Quality Dataaset, Boosting Classifier is the clear winner.

For Breast Cancer dataset all algorithms are very much comparable except decision tree which gives lowest testing accuracy compared to others.

KNN gives the worst result for wine quality because it chooses K=2 as the best K and overthrows all wine qualities except 5 and 6 out of consideration.

Decision Tree over fits the data significantly in Wine Quality dataset.

Breast Cancer Dataset has higher test accuracy than train accuracy because of smaller size of test data.

Overall, results are better for Breast cancer dataset as it is smaller in size, cleaner and most importantly because it has clear signs in its features to classify the problem and it's a Binary classifier. In case of Wine Quality Dataset, it is a multi-class classification problem which is much harder to solve and thus requires even more data to learn.

Dataset		**Decision Tree**	**Boosting**	**K Nearest Neighbors**	**Neural Network**	**Support Vector Machine**
Breast Cancer	**Training Accuracy**	96.33%	100%	96.5%	100%	97.9%
	Testing Accuracy	95.62%	99.27%	99.2%	98.5%	97.5%
Wine Quality	**Training Accuracy**	98.08%	99.84%	79.1%	69.5%	62%
	Testing Accuracy	59.38%	67.24%	50.7%	56.3%	55.3%

References:

- Code Source: Jonathan Tay(https://github.com/JonathanTay/CS-7641-assignment-1)
- Scikit Learn: http://scikit-learn.org/stable/index.htm
- Dataset Sources: https://www.kaggle.com/