CS 7641 MACHINE LEARNING
ASSIGNMENT 1: Supervised Learning

NAME: NIHAR MEHTA
GT USERNAME: nmehta80

A. Description of Problems

1. PULSAR STARS

Link: https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star

This dataset contains 18000 examples of stars and the objective is to identify whether the star is a pulsar star or not. The features of the stars provided in this dataset include:

- Mean of integrated profile
- Standard deviation of integrated profile
- Excess kurtosis of the integrated profile
- Skewness of the integrated profile
- Mean of the DM-SNR curve
- Standard deviation of the DM-SNR curve
- Excess kurtosis of the DM-SNR curve
- Skewness of the DM-SNR curve
- Target_class (1 for pulsar star,0 for not a star)

Pulsars are a rare type of Neutron stars. They can produce radio emissions detectable here on Earth. They are of considerable scientific interest as probes of space-time, the inter-stellar medium, and states of matter . Pulsars produce a detectable pattern of radio emission which is periodic in nature. These patterns depend on the rotation of the pulsar. These detections are averaged over many rotations of the pulser. The first four attributes in the dataset are simple statistics obtained from the integrated pulse profile (folded profile). The remaining four variables are similarly obtained from the DM-SNR curve . These should be sufficient to categorize a star as pulsar or not.

This dataset has a class imbalance since the legitimate pulsar examples are a minority positive class, and spurious examples the majority negative class. It contains 16,259 spurious examples caused by RFI/noise, and 1,639 real pulsar examples. These examples have all been checked by human annotators. It will be a good analysis to compare the performance of various models on such an imbalanced dataset.

2. MOBILE PRICE RANGE PREDICTION

Link: https://www.kaggle.com/iabhishekofficial/mobile-price-classification

This dataset consists of features of a mobile phone mapped to its selling price.

The features include:

Id (ID), Battery_power (Total energy a battery can store in one time measured in mAh), Blue (Has bluetooth or not), Clock_speed (speed at which microprocessor executes instructions), Dual_sim (Has dual sim support or not), Fc (Front Camera mega pixels), Four_g (Has 4G or not), Int_memory (Internal Memory in Gigabytes), M_dep (Mobile Depth in cm), Mobile_wt (Weight of mobile phone), N_cores (Number of cores of processor), Pc (Primary Camera mega pixels), Px_height (Pixel Resolution Height), Px_width (Pixel

Resolution Width), Ram (Random Access Memory in Megabytes), Sc_h (Screen Height of mobile in cm), Sc_w (Screen Width of mobile in cm), Talk_time (longest time that a single battery charge will last when you are), Three_g (Has 3G or not), Touch_screen (Has touch screen or not) and Wifi (Has wifi or not)

Better the features, higher would be the price range of the mobile phone. The price range is divided into four categories and hence its a classic problem of multiclass classification. It is important to analyze various machine learning models on such a task. Also, the dataset does not have a class imbalance so in contrast to the previous dataset,.

B. Description of the analysis

Five machine learning algorithms have been implemented and analyzed on the above datasets- Decision tree, Neural Networks, Boosting, Support Vector Machines and K Nearest Neighbors. They are implemented in python using the sklearn library. Both the datasets have been splitted into training and testing datasets with a ratio of 70:30 datapoints. On the Pulsar dataset which is a binary classification task, training and testing area under curve (AUC) score/ accuracy scores which are a measure of train and test errors have been plotted in a single graph for all the parameters. Cross validation has also been performed on the models and training score as well as validation scores have been plotted. Next, a grid search is also performed over the range of parameters to return the best parameter and accuracy classification score, ROC -AUC score and F1 scores have been reported for this parameter. The length of dataset is also varied and a learning curve is generated which plots the accuracy vs number of training data points. ROC-AUC curves also have been plotted for the best model. If class weights can be used in the model, then unweighted and weighted models have been compared. On the mobile price classification dataset, which is a multiclass classification task, apart from ROC-AUC and F1 scores, all other analyses have been performed as above. Also, the confusion matrix which shows the number of points with class i that have been predicted as class j has been printed.

1. DECISION TREE WITH PRUNING

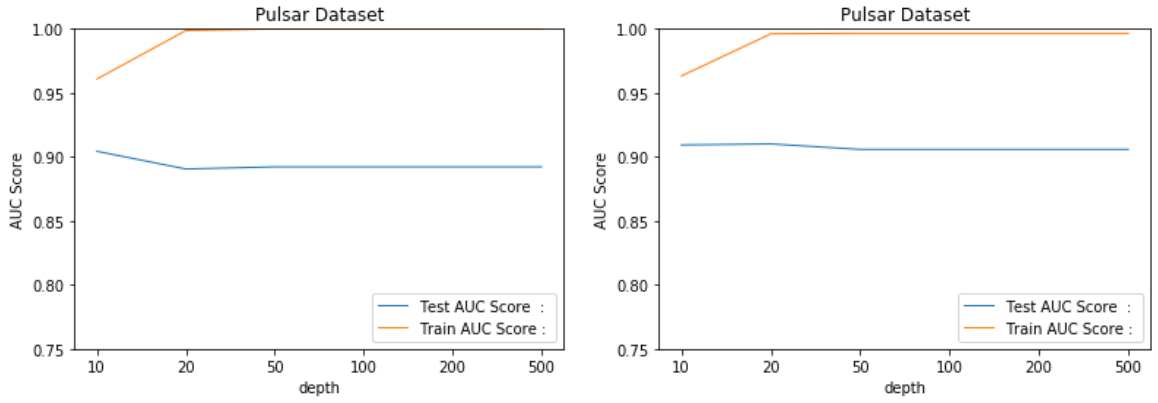
Decision trees are suitable for performing classification tasks. They are a set of rules based on the attributes arranged in a hierarchical fashion to produce the output at the leaves. Each internal node contains a splitting attribute and a split value. For ordinal attributes, the split value is a threshold below which the examples form the left children of the node whereas above which form the right children of the node. For nominal attributes, they are split for all possible values of the attribute. Thus the decision tree can effectively handle categorical as well as numeric features. More the depth of the tree, more are the rules that construct the tree and thus will result in overfitting. Hence, some form of pruning is required to validate the tree on a validation set and thus avoid overfitting. Here, minimum impurity decrease pruning is used, which is also called as reduced error pruning. This involves pruning a specific subtree by replacing it with a leaf node and checking the impurity of the validation set. If it results in reduction in impurity, the node is pruned.

1> Analysis of Pruning

In the Pulsar dataset, depth has been varied from 10 to 500 and AUC scores have been computed for each depth.

In the Mobile Dataset, the depth has been varied to only 20 as the size of dataset is ~2k examples. Thus it can be modelled using smaller trees.

The following is the result for non-pruned tree (left) and pruned tree (right): (Pulsar Dataset)



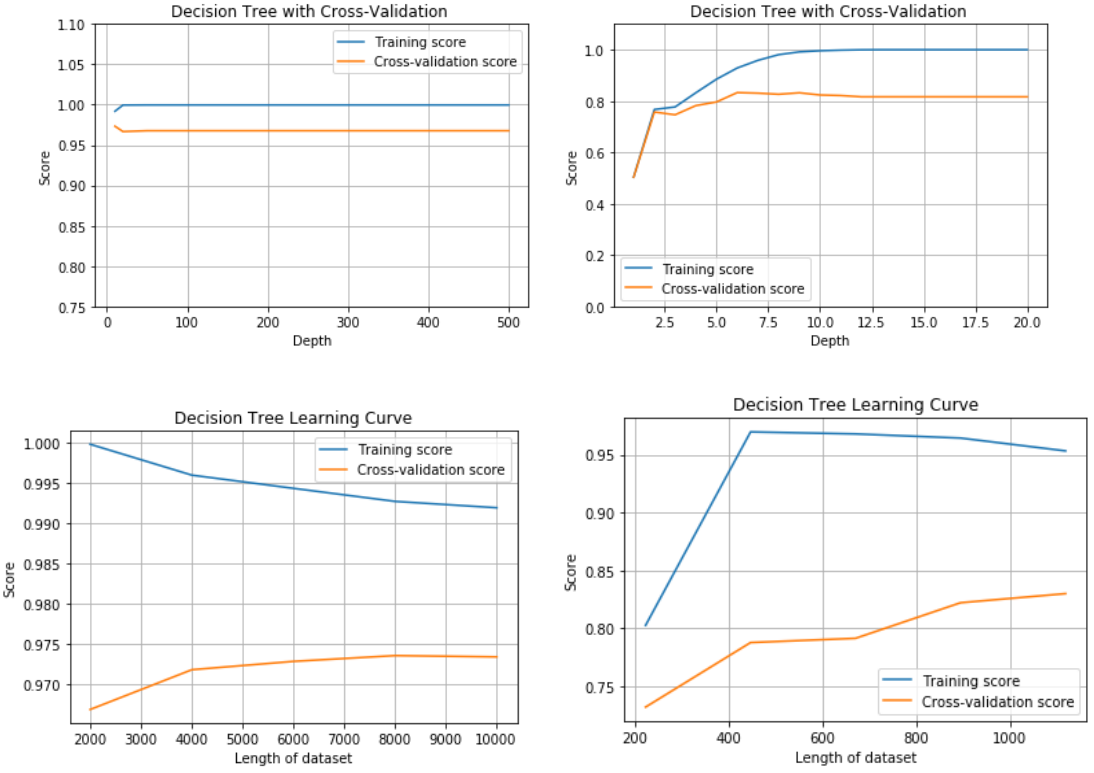
It is clear that the non pruned tree reaches AUC = 1.00 after a particular depth and thus overfits whereas the pruned tree never reaches 1.00. Also, the Test AUC Score for pruned tree is better than the non pruned tree. Hence pruning is important to avoid overfitting.

As we can see, after a certain depth, the tree achieves maximum possible train AUC score and the test AUC score doesn't improve after that depth. This shows that the tree after pruning must have an optimal depth for good performance. All the further analyses will involve pruned trees.

2> Cross Validation

Next, 5- fold cross validation has been performed and the validation curve is plotted. Again, the train accuracy is higher than validation accuracy and both don't change much after a certain depth. Thus after a certain depth, pruning cannot improve the accuracy.

Pulsar Mobile



3> Grid Search

GridSearchCV has been performed using depth as the parameter

Pulsar Dataset:.

Best parameters for Decision Tree: {'max_depth': 10}

Accuracy score of Decision Tree with best parameters is 97.52%
ROC score of Decision Tree with best parameters is 90.93%
F1 score of Decision Tree with best parameters is 85.40%

Mobile Dataset:

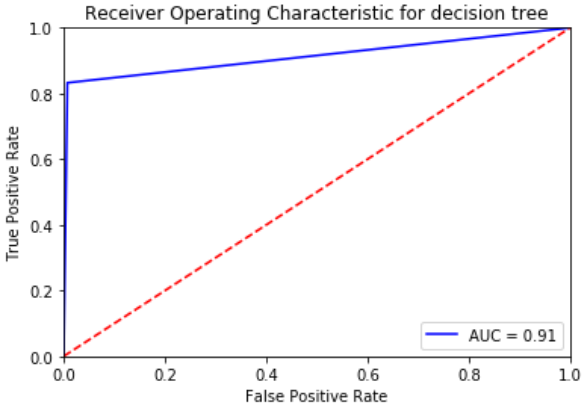
Best parameters for Decision Tree: {'max_depth': 6}
Accuracy score of Decision Tree with best parameters is 82.83%

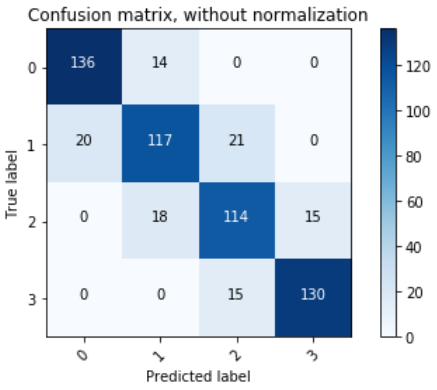
4> Learning curve

Next the learning curve has been plotted by varying the training size from 2000 to 10000 in Pulsar Dataset and from 200 to 1200 in Mobile Dataset.

In Pulsar dataset, the training score decreases with increasing training size because it is easy to overfit a smaller dataset because only a small number of rules will be required. However, it is difficult to model a large dataset In Mobile dataset, the learning curve shows the training accuracy increasing initially because it needs a certain number of examples to form sufficient rules but after a point of time, the rules are not enough to fit the whole dataset.. The cross validation score increases in both datasets which shows that the model is learning the actual patterns in the dataset and not overfitting.

5> ROC- AUC Curve





Thus we can see that the false positives are very well distinguished from true positives and thus we get a higher AUC score = 0.91.

6> Confusion Matrix

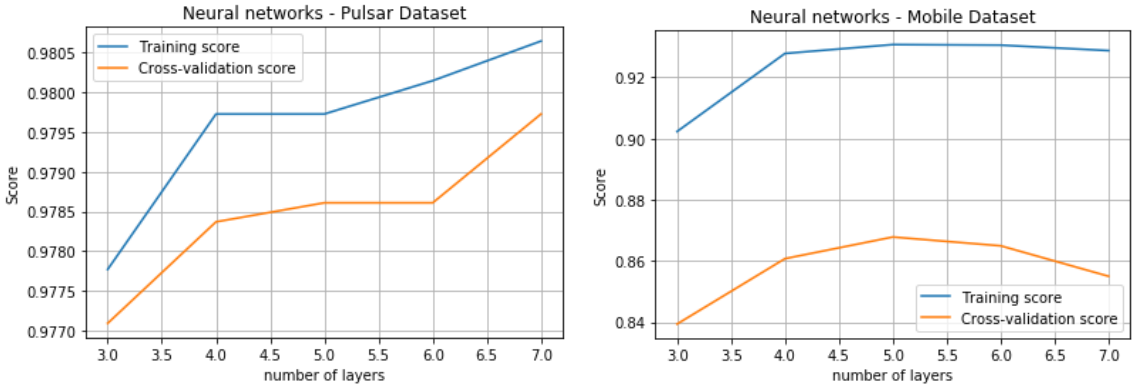
The confusion matrix for the various classes in the Mobile dataset has been plotted. The darker regions imply more data points in that cell. It is interesting to see that the price range is at most misclassified by one class. This means that the model has learnt well from the data and even if it is off in predictions, the error is not substantial.

2. NEURAL NETWORKS

These are networks with hidden layers (between input and output layers) consisting of hidden neurons that learn patterns that help in mapping the input to output. Initially the weights of the neurons are randomly initialized (or sometimes loaded from a pretrained network). During training, the input values are forward propagated to update the weights of the neurons and a loss is calculated based on the true and the predicted output. This loss is backpropagated through the network and optimizers are used to update the weights of the neurons based on the backpropagated loss. This continues through a number of iterations wherein the training data goes through the network again and again. Neural networks have been proven to outperform a lot of traditional techniques for classification tasks.

1> Cross Validation Analysis

The number of hidden layers of the neural network were changed and the cross validation analysis was performed for both pulsar as well as mobile datasets:



With more the number of layers, complex patterns can be learnt by the network, but there is also a change of overfitting. So they must be trained on an optimum no. of layers.

2> Grid Search CV

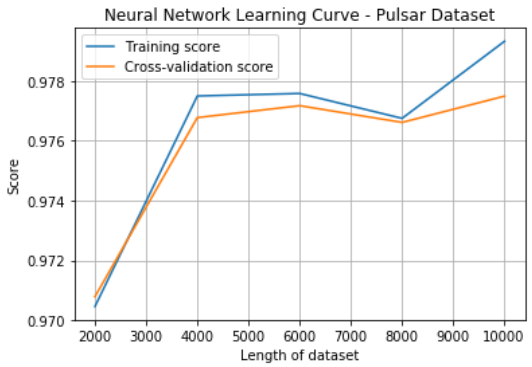
For Pulsar dataset:

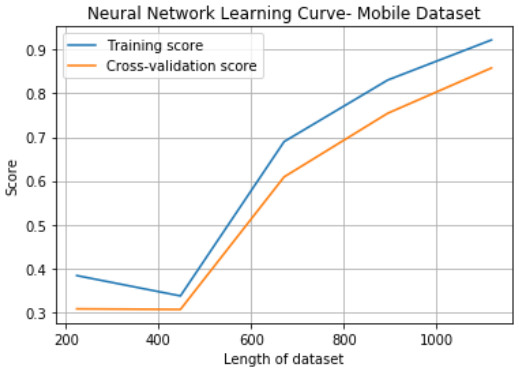
Best parameters for Neural Network: {'hidden_layer_sizes': (100, 75, 75, 50, 50)}
Accuracy score of Neural network with best parameters is 98.04%
ROC score of Neural Network with best parameters is 92.98%
F1 score of Neural Network with best parameters is 89.05%

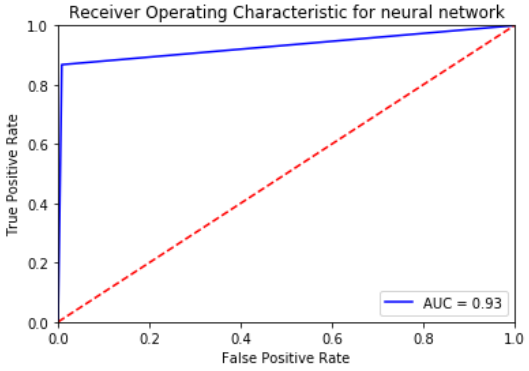
For Mobile Dataset:
Best parameters for Neural Network: {'hidden_layer_sizes': (100, 75, 75, 50)}
Accuracy score of Neural network with best parameters is 85.83%

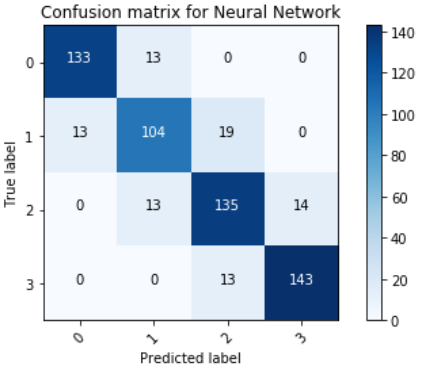
3> Learning curve

In neural networks, there are no rule based predictions like decision trees. Thus we can see that both training and validation accuracies increase when more data is available. More the data, better it is for the network to capture patterns. The accuracies in binary classification are higher than multiclass classification problems because the former is an easier task.









4> ROC Curve for Pulsar Dataset

Here the accuracies as well as AUC = 0.93 are better than decision trees. Hence neural network is a better classifier for the Pulsar dataset.

5> Confusion matrix for Mobile Dataset:

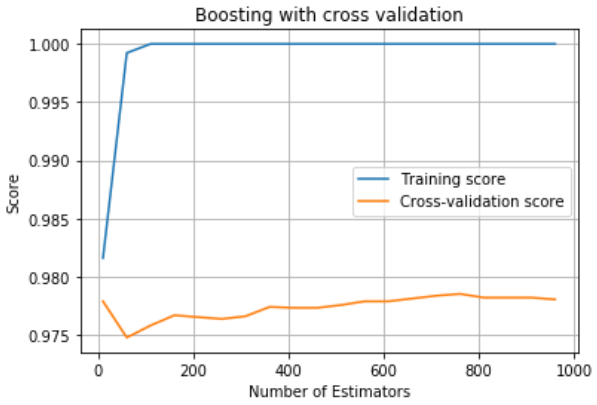
Again, no misclassifications by an offset of more than one class. Lesser misclassifications. Definitely, neural network outperforms decision trees on the Mobile dataset too.

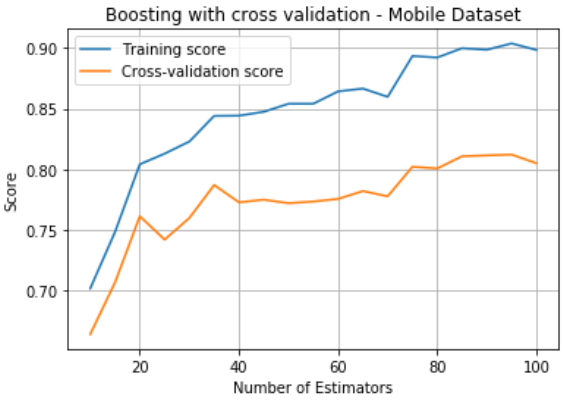
3. BOOSTING

Boosting involves converting a set of weak learners (estimators) into a strong learner. It uses gradient descent in a function space using a convex cost function in order to do so. Here, decision trees with low depth are used as weak learner and are boosted to form a combined strong learner. This process involves initialization of weights for each classifer, then for each step, train the classifiers on a single feature, choosing the classifier with lowest error and updating the weights of the classifier. The final classifier is the linear combination of the weighted classifiers.

1> Cross Validation

The number of estimators are varied from 0 to 1000 keeping the max depth of the trees = 3. With more estimators, the boosting can capture more information but after a specific number of estimators, the accuracy won't improve much.





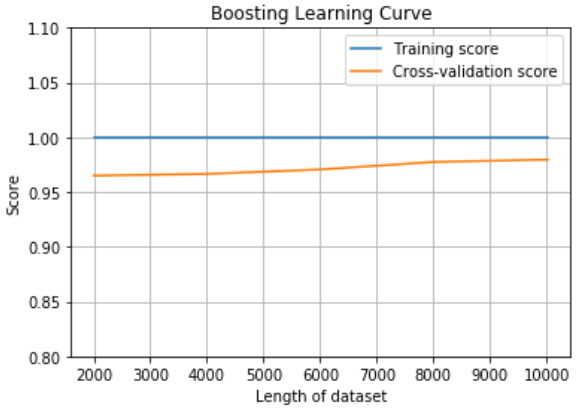
2> Grid Search CV

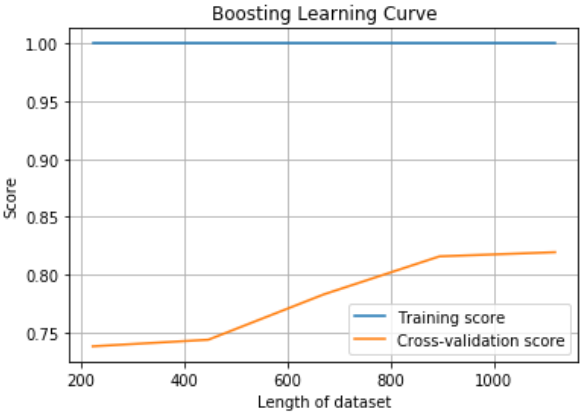
Pulsar Dataset:
Best parameters for Boosting: {'n_estimators': 360}
Accuracy score of Boosting with best parameters is 97.90%
ROC score of Boosting with best parameters is 91.62%
F1 score of Boosting with best parameters is 87.84%

Mobile Dataset:

Best parameters for Boosting: {'n_estimators': 10}
Accuracy score of Boosting with best parameters is 81.17%

3> Learning curve





Pulsar Mobile

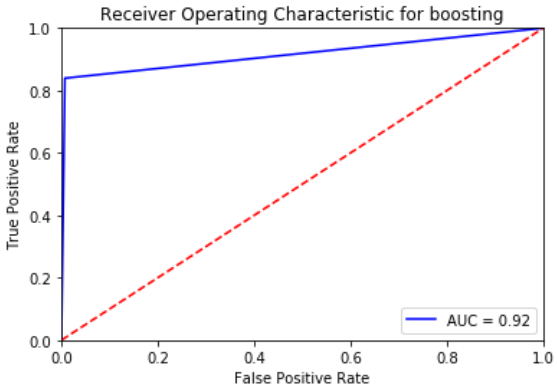
The learning curves show that the validation accuracy in boosting increases when amount of data available is large. With more data, boosting can update weights for all possible types of examples and thus have a generic strong classifier and not some rule based classifier which are prone to overfitting. Thus boosting improves the decision tree classifiers

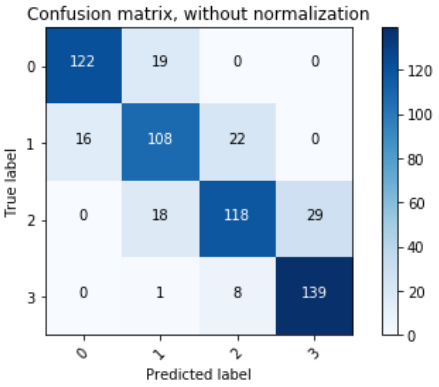
4> ROC Curve

The ROC Curve shows an AUC of 0.92 and thus we can see that it is better than decision trees.

5> Confusion Matrix

The matrix shows that the boosted learner is better than the weak decision tree classifier.



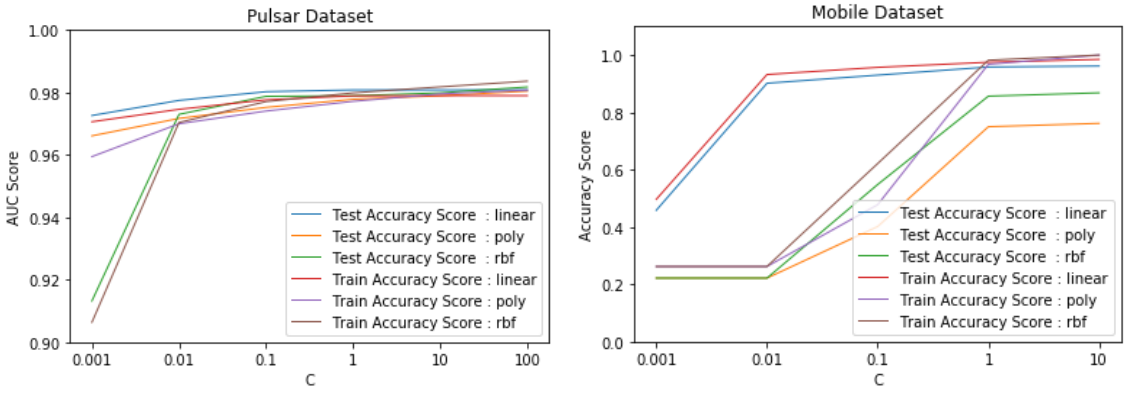


4. SUPPORT VECTOR MACHINES

Support Vector Machines are supervised learning models which are suitable for classification tasks. They can learn complex decision boundaries in the data. They use the best possible hyperplanes and a margin to classify the data points. They may be hard margin or soft margin classifiers. Hard margin classifiers try to choose a margin such that most of the data points are outside the margin. Soft margin classifiers allow misclassification to a certain extent. SVMs use kernels to learn nonlinear decision boundaries. Linear, radial basis function (rbf) and polynomial are some examples of the kernels.

1> Parameter analyses

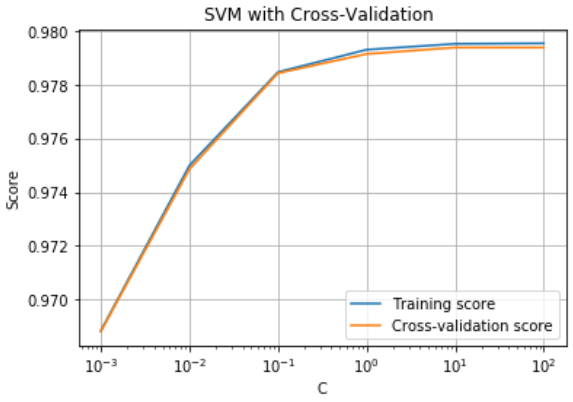
Two parameters are analyzed - C and kernel. Linear, poly and rbf kernels are used. C is varied from 0.001 to 100 in Pulsar dataset and from 0.001 to 10 in Mobile Dataset.

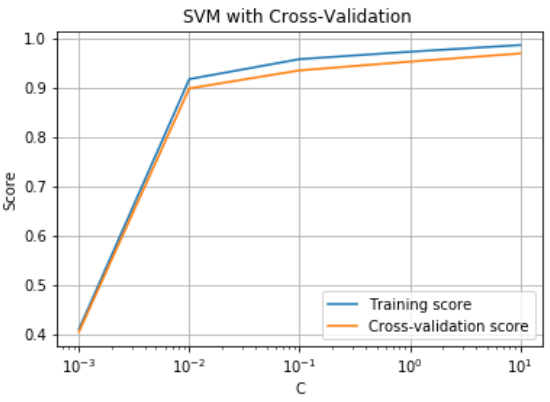


We can observe that for low values of C, the linear kernel outperforms the poly kernel which in turn outperforms the rbf kernel. C is the regularization parameter. Very low values of C allow a lot of error in classification and hence the accuracies are low, but higher values of C may result in overfitting. So, we see that an optimal C should be chosen for training. Also the choice of kernel totally depends on how the data looks like. Here, the datasets seem to have a more direct relation to the features. For example, in the mobile dataset, the price range will be higher if the phone has a higher battery capacity etc.

2> Cross Validation

Next, 5 fold cross validation is performed and the accuracies seem to increase with C and then become steady after a high C is reached.





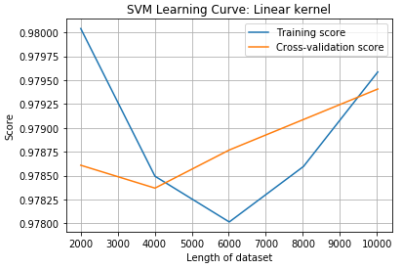
Pulsar dataset

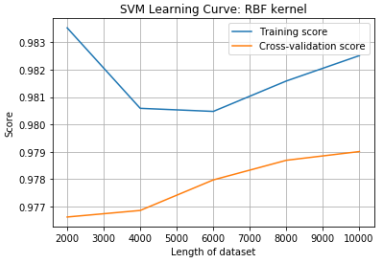
Mobile Dataset

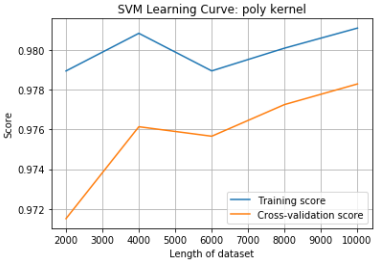
3> GridSearchCV

Pulsar Dataset:
Best parameters for SVM: {'C': 10.0}
Accuracy score of Linear SVM with best parameters is 97.95%
ROC score of Linear SVM with best parameters is 90.89%
F1 score of Linear SVM with best parameters is 87.78%
Mobile Dataset:
Best parameters for SVM: {'C': 10.0}
Accuracy score of Linear SVM with best parameters is 96.67%

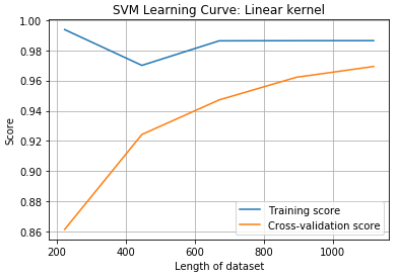
Pulsar:

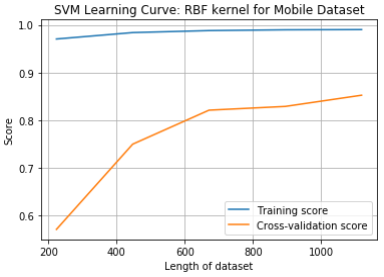


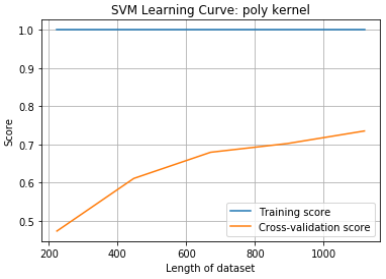




Mobile:







4> Learning curves

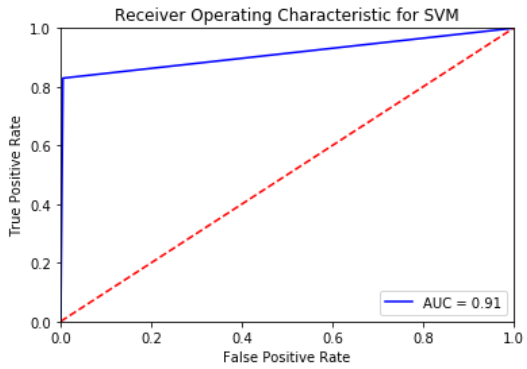
The curves show that the train accuracy reduces after a speciifc length and validation accuracy keeps on increasing with more data. Thus more data is always helpful for the model.

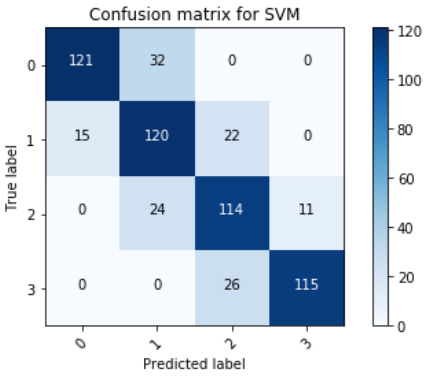
5> ROC Curve

An AUC of 0.91 is achieved. SVMs are comparable to decision tree in this respect.

5> Confusion Matrix

Again the misclassification does not vary by an offset more than a class.

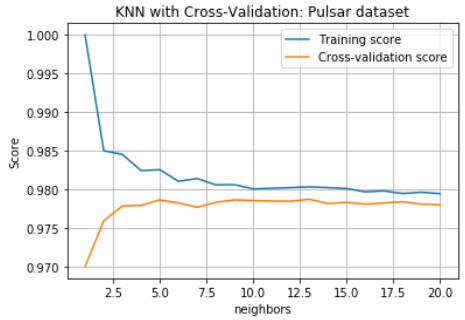


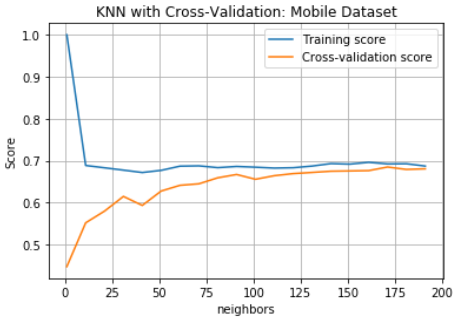


5. K NEAREST NEIGHBORS

K-Nearest Neighbors Classifier finds out the K nearest neighbors of the data points and assign the most common label among these neighbors to the data point.

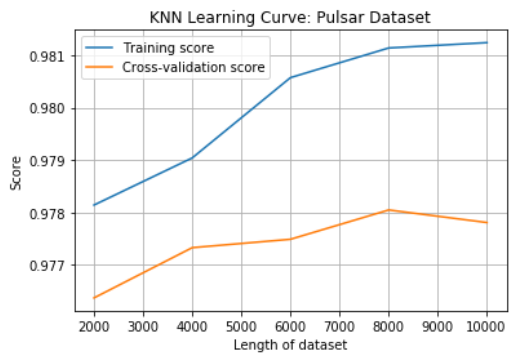
1> Cross Validation

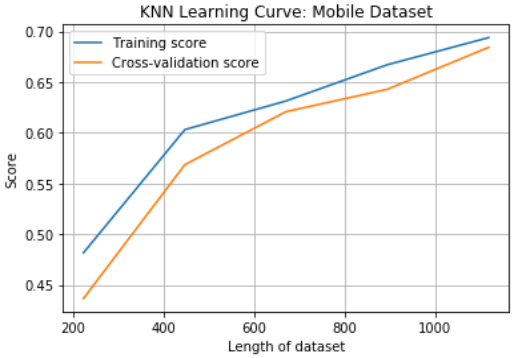




In case of just 1 neighbor, the model overfits the training dataset and thus performs very poorly on the cross validation dataset. A the neighbors increase, the training accuracy is compromised as some of the neighbors may lie on the other side of the decision boundary. At a certain point of time, neighbors become so large that the model is totally biased and the accuracy is the same for the train as well as validation datasets. An optimum value of K is thus required for ideal fitting.

2> Learning curve



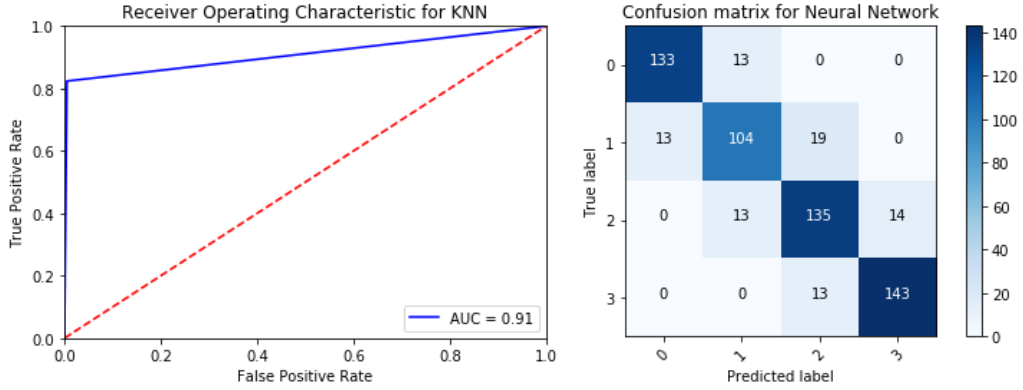


As more data is feeded into the model, the training accuracy increases as well as the validation accuracy increases. This is because more the data points, better will be the neighborhood and capturing patterns without overfitting will be easier.

3> GridSearch

Pulsar Dataset:
Best parameters for Neural Network: {'n_neighbors': 7}
Accuracy score of KNN with best parameters is 97.82%
ROC score of KNN with best parameters is 90.93%
F1 score of KNN with best parameters is 88.02%
Mobile Dataset:
Best parameters for Neural Network:{'n_neighbors': 111}
Accuracy score of KNN with best parameters is 63.50%

4> ROC



The ROC Curve gives an AUC of 0.91 so it is comparable to decision tree and SVM.

5> Confusion Matrix

Again in case of the mobile dataset, the distribution appears such that the ones with very different price range are never very close to each other and hence the neighborhood is ideal to predict the price range of a new test data point.

C. Summary and Conclusions

Pulsar dataset:

Algorithm	**Accuracy**	**F1-score**	**Train Time (ms)**	**Inference Time (ms)**	**AUC**
Decision Tree	96.72	90.79	64	**0.546**	82.81
Boosting	97.90	**91.62**	10200	460	87.84
NN	**98.04**	89.05	551	22	**92.98**
KNN	97.82	90.93	**5**	125	88.02
SVM(Linear)	98.03	88.60	510	59.6	91.98
SVM(RBF)	**98.04**	88.60	367	229	91.62
SVM (Poly)	97.99	88.21	455	91	91.22

In the Pulsar dataset, Neural Network acheived the highest Accuracy as well as the AUC Score. The highest F1 score was obtained by Boosting. Boosting methods are thus in general better than using plain decision trees. Also we observed that the decision trees must be pruned to avoid overfitting. The neural networks can easily capture the nuances in the dataset and thus are appropriate for binary classification. In terms of training time, KNN was the fastest while in terms of the inference time, decision tree was the fastest. In both cases, boosting was incredibly slow because of 360 esimators.

Mobile Dataset:

Algorithm	**Accuracy**	**Train Time (ms)**	**Inference Time (ms)**
Decision Tree	82.83	**5**	**0.118**
Boosting	81.17	35	3
NN	85.83	199	2
KNN	63.50	194	3
SVM(Linear)	**96.67**	68	4
SVM(RBF)	86.67	84	64
SVM (Poly)	78.33	66	36

In the Mobile Dataset, Linear SVM achieved the highest accuracy. It maybe because the data is linearly correlated and thus a linear SVM will be a good choice. The price range of the mobile phone is directly proportional to most of its specifications like RAM size, battery capacity, internal memory etc. In terms of training and inference time, decision tree was the fastest.