# Machine Learning
# Assignment 4 Report

Abhishek Agarwal

2016126

## Question 1)

The accuracies obtained after using GridSearch to tune the hyperparameters for both the models are as follows:

- Decision Tree Classifier:
    - Accuracy without Grid Search:
        - Train: 100%
        - Test: 68.6%

    - Best Parameters:
        - Criterion: Gini
        - Max Depth: 4
        - Min Samples Leaf: 7
        - Min Samples Split: 20

    - Accuracy with best hyperparameters:
        - Train: 78.2%
        - Test: 68.2%

- Random Forest Classifier:
    - Accuracy without Grid Search:
        - Train: 99%
        - Test: 68.6%

    - Best Parameters:
        - Criterion: Gini
        - Max Depth: 13
        - Min Samples Leaf: 20
        - Min Samples Split: 50
        - N Estimators: 30

    - Accuracy:
        - Train: 75%

**END**

        - Test: 70.2%

Without hyperparameter tuning and using the default values, we obtain train accuracy of almost 100% and test accuracy of around 70% in both cases. This shows that the models are overfitting as there is a significant difference between train and test accuracies.
After hyperparameter tuning is performed, the decision tree still slightly overfits the data as the difference in train and test accuracy is around 7%. Random forest performs better than Decision Tree after hyper parameter tuning. This is mainly because when we increase the number of estimators i.e. the number of trees in random forest, the model diverts from overfitting and goes towards a perfect fit as can be seen from the accuracies reported above.

**(b)** The best hyperparameters have been reported above for both the models.
The hyperparameters passed through GridSearch were:

- Decision Tree:
    - **Criterion**: ["gini", "entropy"],
    - **Max Depth** :[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    - **Min Samples Split** : [10, 15, 20, 25, 30]
    - **Min_samples_leaf** : [1,3,7,10]

- Random Forest:
    - **Criterion**: ["gini", "entropy"],
    - **Max Depth** :[3, 8, 13, 18, 23, 28]
    - **Min Samples Split** : [50, 70, 90, 110, 130, 150]
    - **Min_samples_leaf** : [20, 40, 60, 80, 100]
    - **N_Estimators:** [10, 30, 50, 70, 90]

The results obtained after grid search are shown in table form in the following sheet:

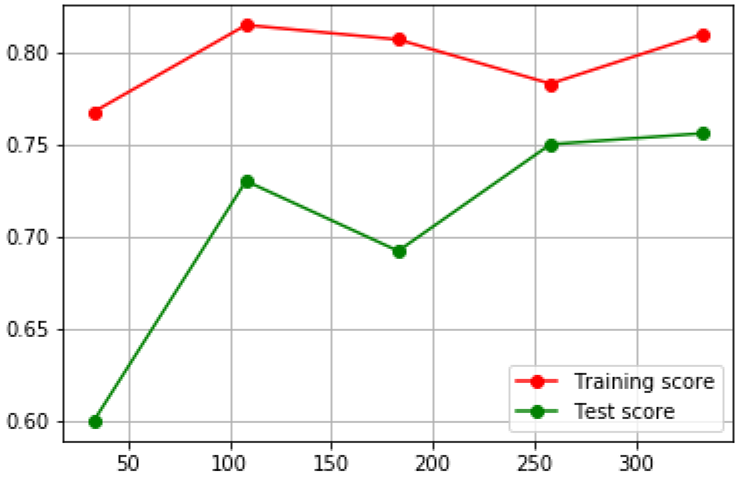
- Decision Tree: Decision_tree_results.csv
- Random Forest: Random_forest_results.csv

It can be observed from the results file for both the models that for the best hyperparameters chosen by GridSearch, the most optimal results are produced. For all other combinations of hyperparameters, the model was overfitting and was not suitable. Hence this particular choice of hyperparameters has been justified.

**END**

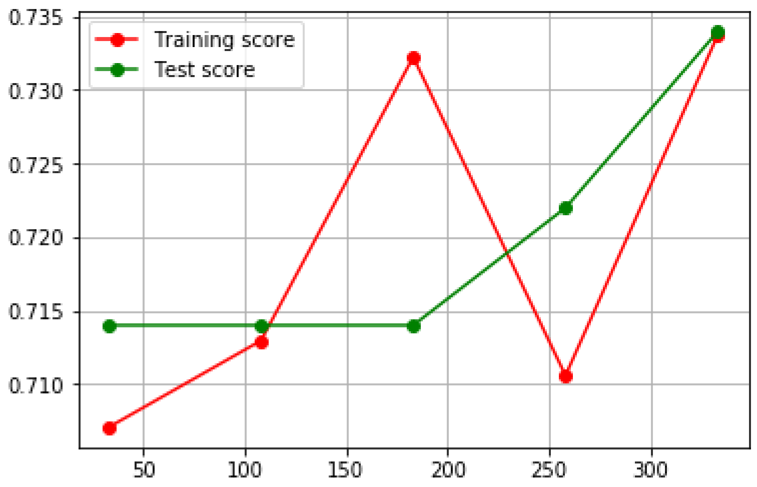
**(c)** Both the models are trained perfectly. We can check this by plotting the graph of training and test accuracies also known as the learning curve for our model. The train and the test scores converge as the training samples increase. X axis represents the number of training samples, Y axis represents the accuracy. Especially for the Random Forest classifier, the train and test scores converge perfectly, denoting a perfect model.

- Decision Tree Classifier



- Random Forest Classifier

**END**



**(d)** The cross validation was taken to be 3. That is there are 3 validation sets

**Decision Tree: [0.23952095808383234, 0.281437125748503, 0.27108433734939763]**
**Variance: 0.00031782136111014995**
**Random Forest: [0.28742514970059885, 0.28742514970059885, 0.2831325301204819]**
**Variance: 4.094796191022986e-05**

The variance obtained for the decision tree classifier is much more than that of Random Forest classifier by 10 times. This is because as we saw above, decision tree is overfitting the data and thus has a higher variance.

**(e)** The best model has been saved in the following files:

- Decision Tree:
  Grid Search fit: Decision_Tree_Classifier.pickle
  Model fit: classifier_dt.pickle
- Random Forest:
  Grid Search fit: Random_Forest_Classifier.pickle
  Model fit: classifier_rf.pickle

  The code to load pickle file and test on data has been added as the last cell in the notebook.

**END**