Name:- Sahil Chitnis
Class: MSc. Data Analytics
Student ID: 23100637

0.1 Divide your training data in train and validation set and then test with the testing set provided in assignment 1.

Validation Dataset:- Validation Data is mostly used to assess the performance of a machine learning model and also to tune the hyper-parameters. It helps in selection of models as validating on different models can help to compare the performance. Before doing any hyper-parameter tuning below is the accuracy score for both the algorithms on the testing dataset is given below.

1. **Decision Tree**:- 70.88 ± 0.54%
2. **Random Forest**:- 74.12 ± 0.33%

0.2 Compare the two classifiers (which ever you used in your assignment 1) using a t-test and report which one is significantly better than the other. Calculate accuracy, confusion matrix, ROC Curve, and Area Under ROC (AUROC).

Confusion Matrix:- It is a table which is used to summarize the performance of a binary classification problem. Below are few terms used in the matrix.**[1]**

1. **True Positive(TP)**:-It represents actual positive values which are predicted as positive.
2. **False Positive(FP)**:- It represents how many times the model wrongly predicts the negative class as positive.
3. **True Negative(TN)**:-It represents how many times the model predicted negative class as negative
4. **False Negative(FN)**:-It represents how many times the model predicted negative class as positive

For decision tree, before doing the hyper-parameter tuning the number of TP was 17640 but after doing the tuning the number got increased to more than 20 thousand which indicates that the precision and the accuracy for decision tree has slightly increased to about approximately 2% , but there has been a significant impact on the recall and the F1 score.

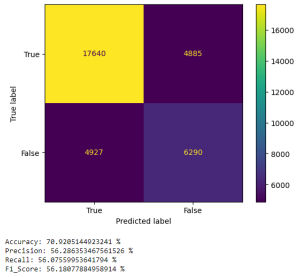

Figure 1: Before Tuning for Decison Tree

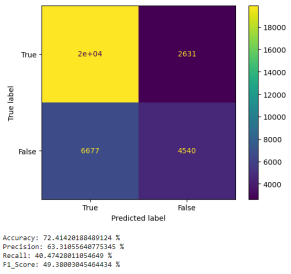

Figure 2: After Tuning for Decision Tree

Similar is the case for random forest algorithm where the recall got affected due to increase in the precision. Below is the pictorial representation for the same.

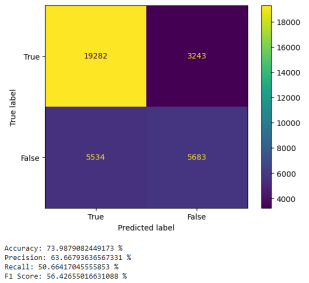

Figure 3: Before Tuning for Random Forest

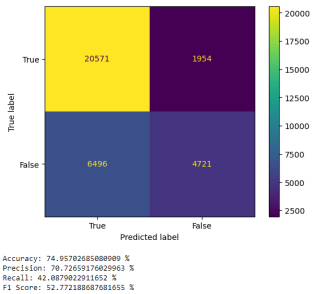

Figure 4: After Tuning for Random Forest

ROC:- It stands for Receiver Operating Characteristic curve, it is used to visualize the trade-off between the true positive and the false positive rate i.e nothing but the sensitivity and the specificity**[2]**. For Decision tree, it can be seen that there is a significant change in the ROC curve after the parameter tuning, below picture shows the comparison between pre and post implementation of hyper-parameter tuning.

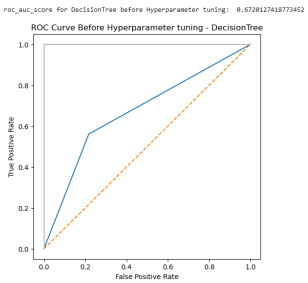

Figure 5: ROC before tuning for Decision Tree

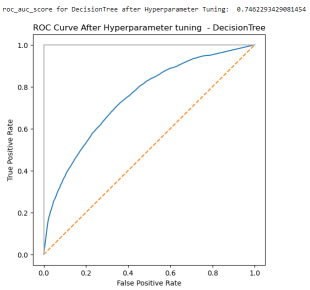

Figure 6: ROC after tuning for Decision Tree

Similarly, for random forest there is a very less change in the ROC curve as well as the ROC score which can be seen with the pictorial representation. If the curve is more closer to the top left side then it is usually a good sign that the model is a well-performing classification model.

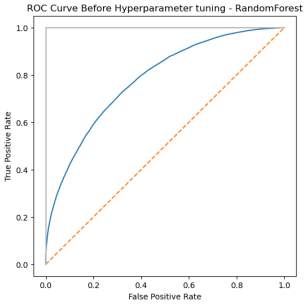

Figure 7: ROC before tuning for Random Forest

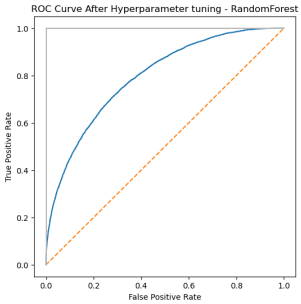

Figure 8: ROC after tuning for Random Forest

T-test:- It is a test which is used to if there is any difference between the means of two samples. After comparing the accuracy's of the two models we will get to know the p-value which is used to compare to the significant value in this assignment the significant value is 5%. Hence, it can be seen that the model rejects the null hypothesis and accepts the alternate hypothesis as the p value is less than 5%.

```
t_statistic,p_value=ttest_rel(metrics_dt["Accuracy"],metrics_RF["Accuracy"])
alpha = 0.05

if p_value < alpha:
    print(f"The difference in accuracies is statistically significant (p-value: {p_value})")
    if accuracy_Dec_Tree > accuracy_Rand_Forest:
        print("Decision Tree better.")
    else:
        print("Random Forest better.")
else:
    print(f"The difference in accuracies is not statistically significant (P-value: {p_value})")
```

```
The difference in accuracies is statistically significant (p-value: 4.960473705820592e-10)
Random Forest is significantly better.
```

Figure 9: T-test result

0.3 Use a k-fold (k=10) cross validation and report the performance of the testing set. Explain which model you used finally for generating the predictions of each sample test data.

Hyperparameters and Tuning:- Hyperparameters are those parameters/values which are use to control the learning process of a model **[6]**. Hyperparameter tuning is a method of finding the best set of hyperparameter for a particular machine learning model to have a optimal/best performance for a given problem **[5]**. In order to choose the right set of hyperparameters I have used randomisedCV function which uses random sampling to choose the correct set from a predefined distributions **[4]**.

Cross Validation:- It is a technique which is used to know how well a model will perform to unknown/unseen data-set.There are various types of cross validation techniques which are listed below.

1. **Holdout Method**:- It is the simplest CV method in which the data-set is divided into training and testing dataset. **[7]**.
2. **K-fold CV**:- In this method the training dataset is divided into K equal parts, then at each k-fold one part is used as a validation set while the other is used for training the model **[7]**. Here, k-fold cross validation is used with 10 foldes meaning the training data is divided into 10 different parts, each fold act as a validation set while the other remaining part act as a training dataset. This method is not suitable for imbalanced dataset.
3. **Stratified K-fold**:- To resolve the problem of K-fold CV which cannot be applied for imbalanced dataset, stratified K-fold comes into picture. In this each fold of the cross-validation has similar target variables that are present in the whole dataset this helps in datasets which are imbalanced. **[7]**

There are various functions which are available in the sklearn library for doing cross-validation out of which RandomizedCV and GridSearchCV are the two most popular ones. In GridSearchCV an exhaustive search is carried out over a given hyper-parameters to find the best combination of those. It is computationally exhaustive and can take lot of time for a larger dataset to go through. On the other hand, RandomizedCV performs random search to take the hyper-parameter values randomly from a defined range. It is much more efficient than GridSearchCV as it uses random sampling and does not perform an exhaustive search over the whole dataset. One drawback of this is, it required more number of iterations in-order to find the best combination of hyper-parameters.

Out[511]:

Validation set #	Accuracy	Precision	Recall	F1-Score	AUC
1	71.46	56.21	58.23	57.24	0.68
2	70.66	55.03	55.88	55.17	0.67
3	70.53	54.69	54.65	54.93	0.67
4	69.92	53.74	54.37	54.11	0.66
5	70.77	54.24	55.43	54.96	0.67
6	71.47	56.21	56.76	56.52	0.68
7	71.64	56.26	56.41	56.36	0.68
8	70.75	54.89	55.04	55.43	0.67
9	70.30	53.73	56.25	54.83	0.66
10	71.31	56.23	55.04	56.00	0.67

Figure 10: Accuracy at each K-Fold for Decision Tree

[64]:

Validation set #	Accuracy	Precision	Recall	F1-Score	AUC
1	74.43	63.44	53.07	57.16	0.79
2	73.81	61.83	50.57	55.23	0.78
3	73.82	62.27	50.08	55.45	0.77
4	73.44	62.07	50.20	55.54	0.77
5	73.82	62.75	50.55	55.76	0.78
6	74.06	62.33	51.72	56.32	0.78
7	74.22	63.13	51.41	56.19	0.78
8	73.72	62.24	50.04	55.83	0.77
9	74.05	61.08	51.33	56.08	0.77
10	74.62	63.86	50.55	56.16	0.79

Figure 11: Accuracy at each K-Fold for Random Forest

NOTE:- For calculating the cross validation score for random forest(Figure 10) I have used Kaggle GPU due to which there may be slight change in the image.

0.4 Conclusion

Based on the overall assessment of the performed t-test and the hyper-parameter tuning as well as the ROC score it can be seen that the random forest is performing better than the decision tree for the given data-set. Below image represents the difference between the ROC curve for both the algorithms.

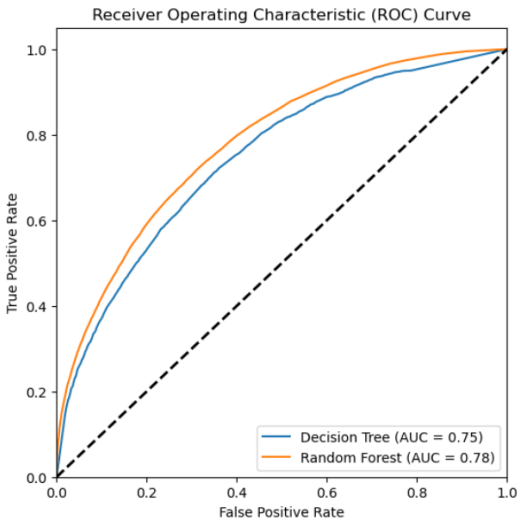

Figure 12: ROC comparison for both the algorithms

1 REFERENCES

1] **https://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning**
2] **https://www.simplilearn.com/what-is-a-roc-curve-and-how-to-use-it-in-performance-modeling-article**
3] **https://www.analyticsvidhya.com/blog/2022/02/different-types-of-cross-validations-in-machine-learning/**
4] **https://medium.com/chinmaygaikwad/hyperparameter-tuning-for-tree-models-f99a66446742**
5] **https://aws.amazon.com/what-is/hyperparameter-tuning/**
6] **https://www.turing.com/kb/different-types-of-cross-validations-in-machine-learning-and-their-explanations**
7] **https://www.geeksforgeeks.org/t-test/**