Titanic Predictions Report

Uddesh Soman

5th Semester, UECU

Null Accuracy for the data is 61.6162 %

LogisticRegression(random\_state=42, solver='liblinear')

For Training data

Correct Predictions: 568

False Positives: 62

False Negatives: 82

Accuracy: 0.7977

Precision: 0.75

Sensitivity: 0.6940

Specificity: 0.8603

F1: 0.7209

ROC AUC score 0.8470

Tuned model :LogisticRegression(C=79, random\_state=42, solver='liblinear')

For Training data

Correct Predictions: 570

False Positives: 60

False Negatives: 82

Accuracy: 0.8005

Precision: 0.7560

Sensitivity: 0.6940

Specificity: 0.8648

F1: 0.7237

ROC AUC score 0.8495

For Test data

Correct Predictions: 145

False Positives: 15

False Negatives: 19

Accuracy: 0.8100

Precision: 0.7857

Sensitivity: 0.7432

Specificity: 0.8571

F1: 0.7638

ROC AUC score 0.8818

DecisionTreeClassifier(max\_depth=4)

For Training data

Correct Predictions: 598

False Positives: 15

False Negatives: 99

Accuracy: 0.8398

Precision: 0.9184

Sensitivity: 0.6305

Specificity: 0.9662

F1: 0.7477

ROC AUC score 0.8814

Tuned model :DecisionTreeClassifier(criterion='entropy', max\_depth=3)

For Training data

Correct Predictions: 594

False Positives: 43

False Negatives: 75

Accuracy: 0.8342

Precision: 0.8177

Sensitivity: 0.7201

Specificity: 0.9031

F1: 0.7658

ROC AUC score 0.8687

For Test data

Correct Predictions: 143

False Positives: 9

False Negatives: 27

Accuracy: 0.7988

Precision: 0.8392

Sensitivity: 0.6351

Specificity: 0.9142

F1: 0.7230

ROC AUC score 0.8487

RandomForestClassifier(max\_depth=4, n\_estimators=50)

For Training data

Correct Predictions: 598

False Positives: 34

False Negatives: 80

Accuracy: 0.8398

Precision: 0.8468

Sensitivity: 0.7014

Specificity: 0.9234

F1: 0.7673

ROC AUC score 0.8810

Tuned model: RandomForestClassifier(bootstrap=False, max\_depth=5, n\_estimators=10)

For Training data

Correct Predictions: 614

False Positives: 25

False Negatives: 73

Accuracy: 0.8623

Precision: 0.8863

Sensitivity: 0.7276

Specificity: 0.9436

F1: 0.7991

ROC AUC score 0.9075

For Test data

Correct Predictions: 144

False Positives: 13

False Negatives: 22

Accuracy: 0.8044

Precision: 0.8

Sensitivity: 0.7027

Specificity: 0.8761

F1: 0.7482

ROC AUC score 0.8933

XGBClassifier(alpha=10, base\_score=0.5, booster='gbtree', colsample\_bylevel=1, colsample\_bynode=1, colsample\_bytree=1, eval\_metric=['logloss'],gamma=0, gpu\_id=-1, importance\_type='gain', interaction\_constraints='', learning\_rate=1, max\_delta\_step=0,

max\_depth=3, min\_child\_weight=1, missing=nan, monotone\_constraints='()', n\_estimators=1000, n\_jobs=8, num\_parallel\_tree=1, random\_state=0, reg\_alpha=10, reg\_lambda=1,

scale\_pos\_weight=1, subsample=1, tree\_method='exact', validate\_parameters=1, verbosity=None)

For training data

Correct Predictions: 591

False Positives: 27

False Negatives: 94

Accuracy: 0.8300561797752809

Precision: 0.8656716417910447

Sensitivity: 0.6492537313432836

Specificity: 0.9391891891891891

F1: 0.7420042643923241

ROC AUC score 0.8659

Tuned model: XGBClassifier(alpha=10, base\_score=0.5, booster='gbtree', colsample\_bylevel=1,colsample\_bynode=1, colsample\_bytree=0.7, eval\_metric=['logloss'],gamma=0.48, gpu\_id=-1, importance\_type='gain',interaction\_constraints='', learning\_rate=0.6, max\_delta\_step=0,max\_depth=12, min\_child\_weight=4,monotone\_constraints='()', n\_estimators=10, n\_jobs=8,num\_parallel\_tree=1, random\_state=0, reg\_alpha=1, reg\_lambda=1,scale\_pos\_weight=1, subsample=0.71, tree\_method='exact',validate\_parameters=1, verbosity=None)

For Training data

Correct Predictions: 612

False Positives: 21

False Negatives: 79

Accuracy: 0.8595505617977528

Precision: 0.9

Sensitivity: 0.7052238805970149

Specificity: 0.9527027027027027

F1: 0.7907949790794979

ROC AUC score 0.9105275984940164

For test data

Correct Predictions: 145

False Positives: 11

False Negatives: 23

Accuracy: 0.8100558659217877

Precision: 0.8225806451612904

Sensitivity: 0.6891891891891891

Specificity: 0.8952380952380953

F1: 0.75

ROC AUC score 0.8796010296010295

So far Random Forest is the best model according to our analysis, it has given Highest accuracy and ROC for traing data and despite little overfitting it has good accuracy and roc scoreds for test data as well.

Logistic regression was our baseline model which preformed better than Decision tree and test scores were better than XGBoost as well.

XGBoost is considerable as there is low overfitting but accuracy and ROC scores are less. Hyperparameter tuning didn't help either. After manually changing max\_depth, learning rate and n\_estimators in tuned model, considerable scores were obtained.

So in Conclusion, as far as supervised learning is concerned Random Forest Classifier should be preferred for the given dataset followed by XGBClassifier and simple logistic regression if Overfitting is a concern.

So, there is no clear distinction among clusters that could predict binary attribute survival. Hence KMeans clustering is not appropriate for the task at all.

For Final Unknown data

Similarity between Logistic regression and Random Forest is 88.03 %

Similarity between Logistic regression and XGBoost is 91.14 %

Similarity between XGBoost and Random Forest is 92.58 %