Project Development Phase
Model Performance Test

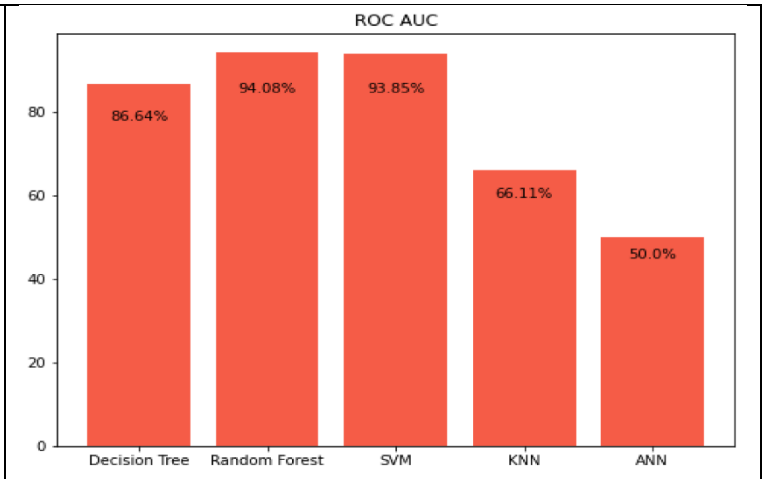
Date	15 November 2022
Team ID	PNT2022TMID40372
Project Name	Statistical Machine Learning Approaches to Liver Disease Prediction
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix, Accuracy Score, Classification Report – Random Forest	In [27]: `from sklearn.metrics import accuracy_score, confusion_matrix, classification_report` In [28]: `from sklearn.ensemble import RandomForestClassifier` `RandomForest = RandomForestClassifier()` `RandomForest = RandomForest.fit(X_train,y_train)` `y_pred = RandomForest.predict(X_test)` `print('Accuracy:', accuracy_score(y_test,y_pred))` `print(confusion_matrix(y_test,y_pred))` `print(classification_report(y_test,y_pred))` Accuracy: 0.6929824561403509 [[13 25] [10 66]] precision recall f1-score support 0 0.57 0.34 0.43 38 1 0.73 0.87 0.79 76 accuracy 0.69 114 macro avg 0.65 0.61 0.61 114 weighted avg 0.67 0.69 0.67 114

	Metrics	**Classification Model:** Confusion Matrix, Accuracy Score, Classification Report-KNN	In [30]: ``` from sklearn.neighbors import KNeighborsClassifier knn_classifier = KNeighborsClassifier() knn_classifier = knn_classifier.fit(X_train, y_train) y_pred = knn_classifier.predict(X_test) print('Accuracy:', accuracy_score(y_test,y_pred)) print(confusion_matrix(y_test,y_pred)) print(classification_report(y_test,y_pred)) ``` ``` Accuracy: 0.5964912280701754 [[13 25] [21 55]] precision recall f1-score support 0 0.38 0.34 0.36 38 1 0.69 0.72 0.71 76 accuracy 0.60 114 macro avg 0.53 0.53 0.53 114 weighted avg 0.59 0.60 0.59 114 ```
	Metrics	**Classification Model:** Confusion Matrix, Accuracy Score, Classification Report-SVM	In [32]: ``` from sklearn.svm import SVC svm_classifier = SVC() svm_classifier = svm_classifier.fit(X_train, y_train) y_pred = svm_classifier.predict(X_test) print('Accuracy:', accuracy_score(y_test,y_pred)) print(confusion_matrix(y_test,y_pred)) print(classification_report(y_test,y_pred)) ``` ``` Accuracy: 0.6666666666666666 [[0 38] [0 76]] precision recall f1-score support 0 0.00 0.00 0.00 38 1 0.67 1.00 0.80 76 accuracy 0.67 114 macro avg 0.33 0.50 0.40 114 weighted avg 0.44 0.67 0.53 114 ```

2.	Tune the Model	Hyperparameter Tuning - Grid SearchCV, finding best estimators for each algorithm in ensemble model Validation Method -cross validation	
3.			In [ ]: ``` from sklearn.model_selection import GridSearchCV # Create the parameter grid based on the results of random search param_grid = { 'bootstrap': [True], 'max_depth': [10,15], 'max_features': [2, 3], 'min_samples_leaf': [3, 4, 5,6], 'min_samples_split': [3,4,5,6], 'n_estimators': [1150, 1200, 1250, 1300,1350] } # Create a based model rf = RandomForestClassifier() # Instantiate the grid search model grid_search = GridSearchCV(estimator = rf, param_grid = param_grid, cv = 3, n_jobs = -1, verbose = 2) # Fit the grid search to the data grid_search.fit(x_train,y_train) ``` best parameter from gridseachCV