Report: Parkinson's Telemonitoring Dataset Analysis

Ali Rostami 610302040

Introduction

This report outlines the progress made on the project utilizing the Parkinson's Telemonitoring dataset. The objective is to transform the dataset from a regression problem to a two-class classification problem, exploring various methods and evaluating their effectiveness.

Dataset Preparation and Feature Selection

Initially, the code begins with library imports and variable declarations, allowing for easy parameter adjustments. The Parkinson's Telemonitoring dataset, specifically, has been employed for this analysis.

Upon reviewing the dataset, I identified two potential targets: ‘motor_UPDRS’ and ‘total_UPDRS’. After consulting the relevant literature, I concluded that ‘total_UPDRS’ is more suitable for our needs, given its comprehensive data representation. Consequently, this column was added to our feature set.

Classification Approach and Data Transformation

A critical decision was the transformation of the dataset from a regression to a classification problem. To classify the data into two distinct classes, I explored three methods: separation by mean, median, and the KMeans clustering method. Each method offers a unique approach to class assignment.

To facilitate this, I developed two functions:

A KMeans clustering function, configured for two classes.

A function with a ‘type’ input, allowing users to select either the mean or median for classification.

The user can choose either function to add a binary target column (0 and 1) to the dataset, following which the ‘total_UPDRS’ column is removed.

Data Splitting and Normalization

The dataset was then split into training and testing subsets. This enables model training and subsequent accuracy assessment on the test data. To improve model performance, data normalization was carried out using the StandardScaler.

Model Development and Evaluation

In the subsequent sections, various models were created and evaluated. The GridSearch method was utilized to fine-tune model parameters. The best parameters identified by GridSearch will be discussed later in this report.

Post-parameter tuning, the cross_val_score function was employed to obtain scoring data, vital for project requirements. Finally, the chosen model was used to predict targets on the test set. The results were stored for later analysis.

Visualization and Reporting

The last section of the code is dedicated to visualization. It includes the generation of confusion matrices and box plots for the evaluation scores. These visualizations, imported into this report, aid in understanding model performance and scoring variations.

Results:

So now we want to check the results I've gotten from this code for this purpose I run this code for all 3 classify methods I mentioned before and captured the results and we are going to see all of them now.

KMeans Classify:

First of all, we will see results in this table and I have to mention that we get really good results with this method and the most of our models got 100% accuracy on test data.

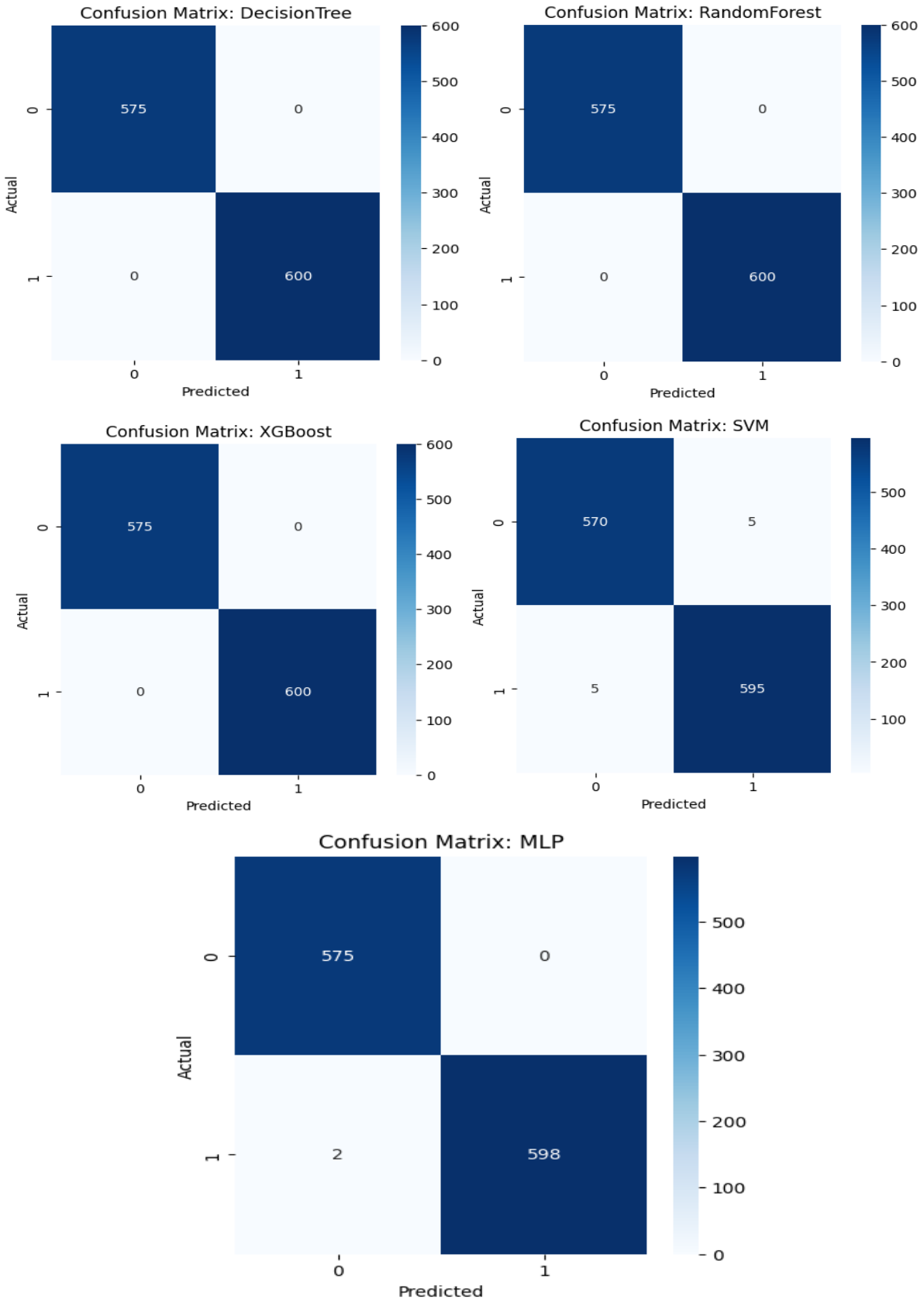
Method	Value Pameters	Accuracy	Precision	Recall	F1
Decision Tree	'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'	1	1	1	1
Random Forest	'criterion': 'gini', 'max_depth': 5, 'min_samples_split': 2, 'n_estimators': 10	1	1	1	1
XGBoost	'criterion': 'gini', 'max_depth': 5, 'min_samples_split': 2, 'n_estimators': 10	0.999	1	0.999	0.999
SVM	'C': 10, 'gamma': 0.01, 'kernel': 'rbf'	0.994	0.995	0.992	0.994
MLP	'activation': 'tanh', 'alpha': 0.001, 'hidden_layer_sizes': (200, 100), 'learning_rate': 'constant', 'max_iter': 500	0.999	1	0.999	0.999

Accuracies on predicting test data:

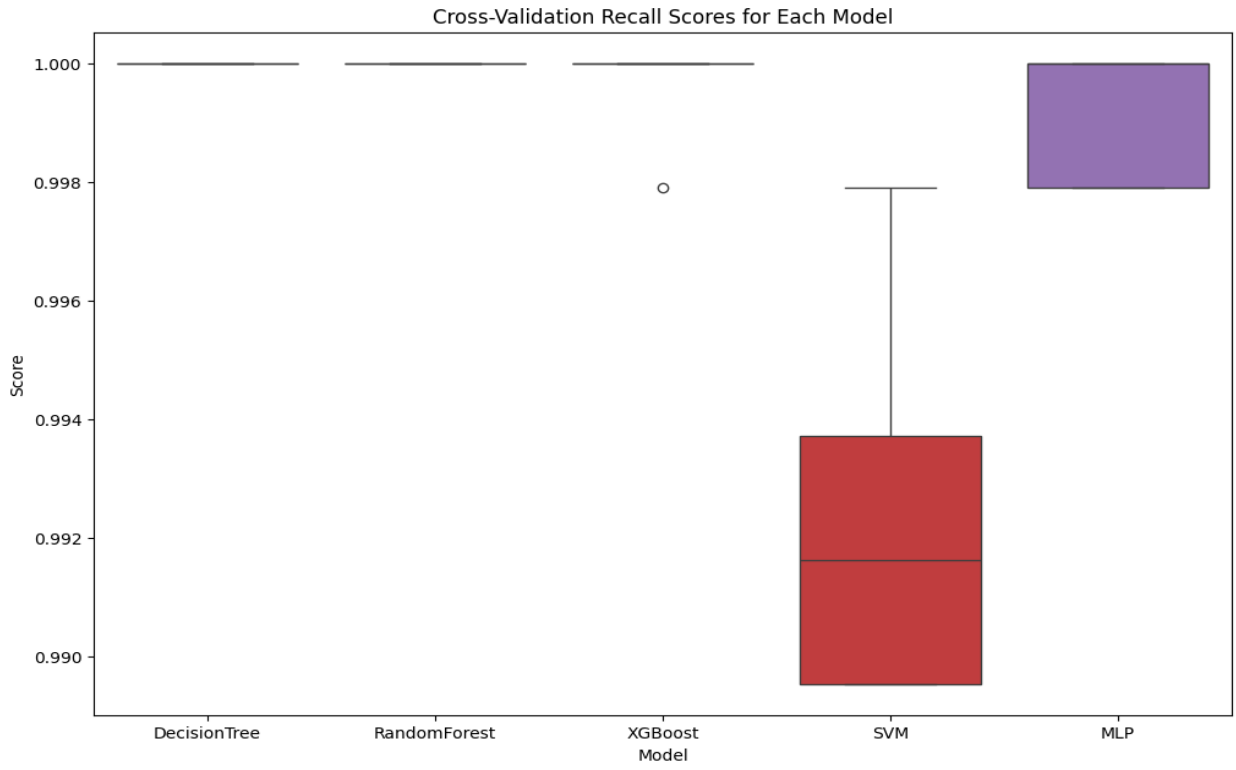
Decision Tree: 100% - Random Forest: 100% - XGBoost: 100% - SVM: 99.1% - MLP: 99.8%

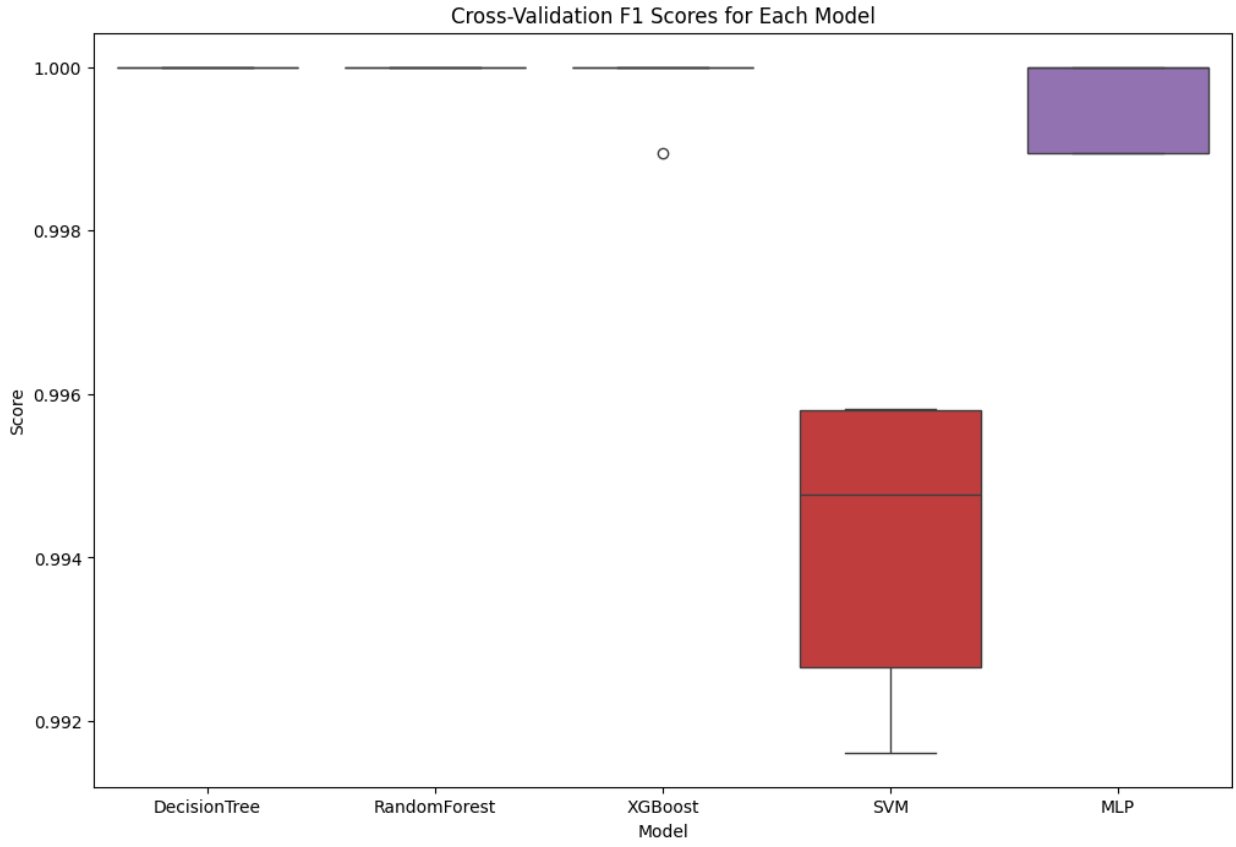
And now we are going to see plots of these data:

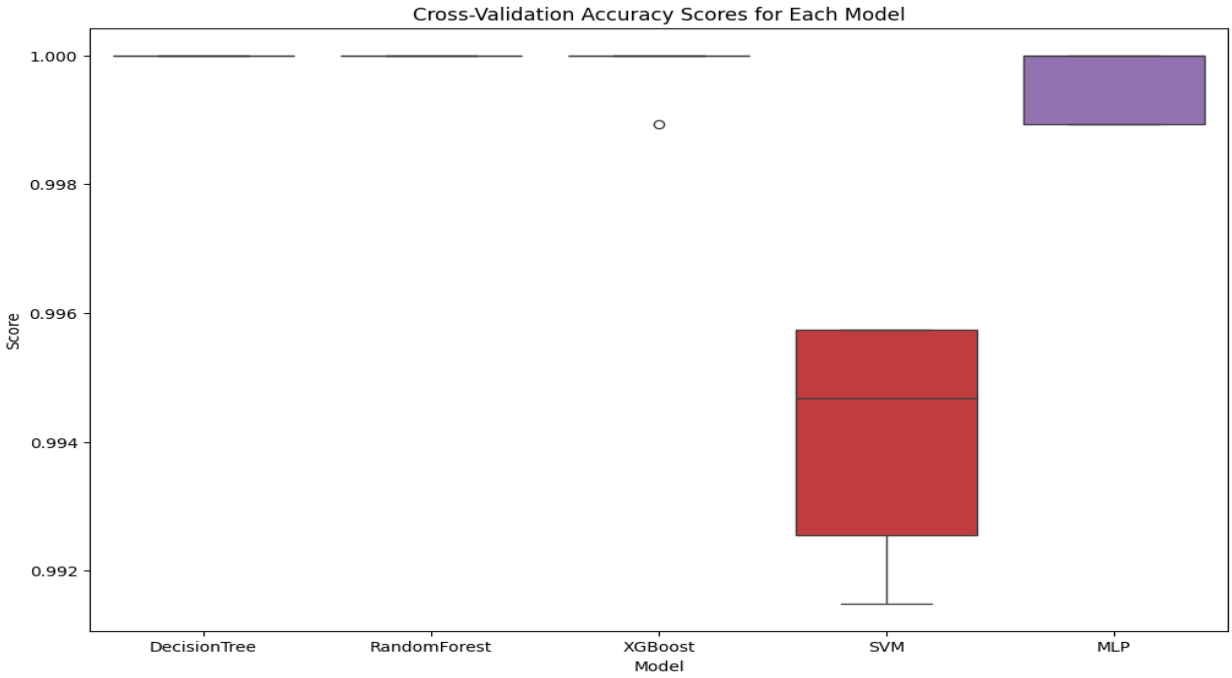
Confusion Matrix:

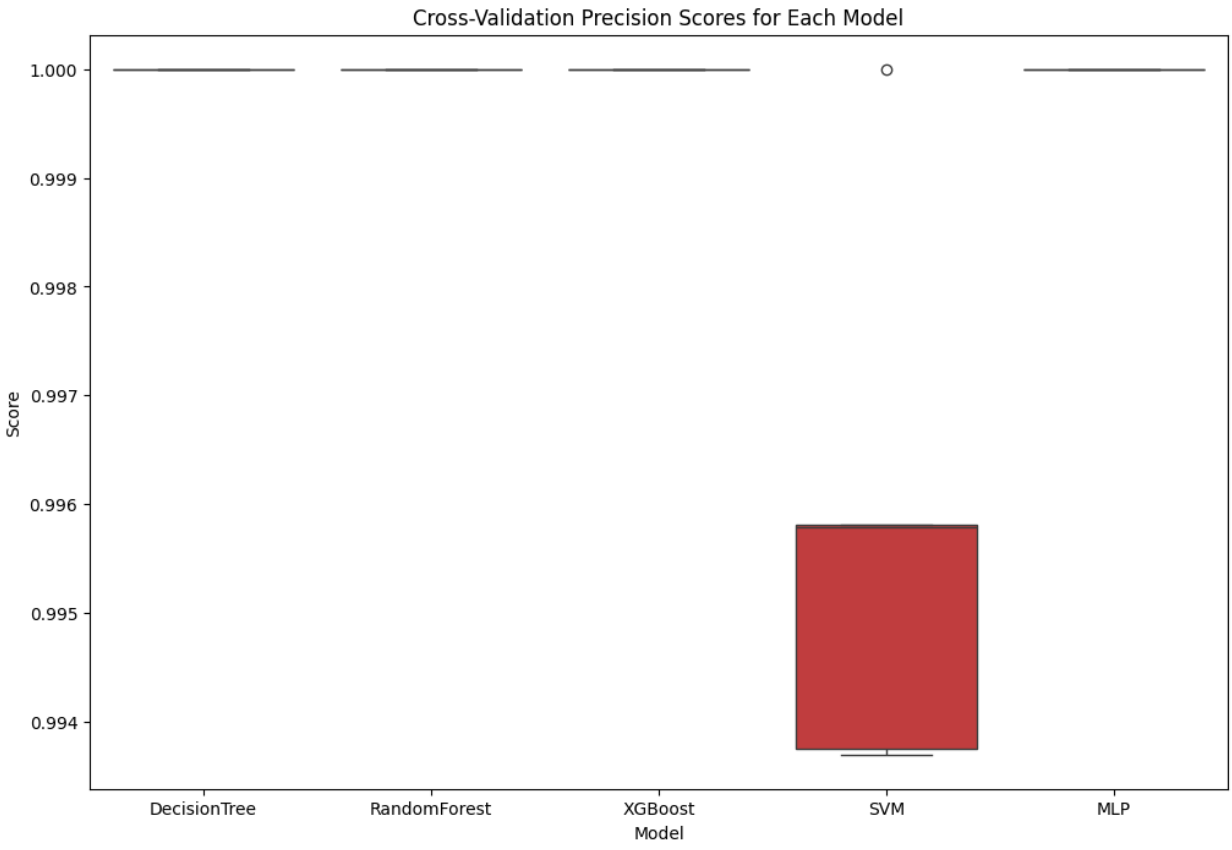

Confusion Matrix: DecisionTree
575
0
0
600
Actual
Predicted
Confusion Matrix: RandomForest
575
0
0
600
Actual
Predicted
Confusion Matrix: XGBoost
575
0
0
600
Actual
Predicted
Confusion Matrix: SVM
570
5
5
595
Actual
Predicted
Confusion Matrix: MLP
575
0
2
598
Actual
Predicted

Box Plots:


Cross-Validation Recall Scores for Each Model
1.000
0.998
0.996
0.994
0.992
0.990
Score
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation F1 Scores for Each Model
1.000
0.998
0.996
0.994
0.992
Score
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation Accuracy Scores for Each Model
1.000
0.998
0.996
0.994
0.992
Score
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation Precision Scores for Each Model
1.000
0.999
0.998
0.997
0.996
0.995
0.994
Score
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model

Mean Classify:

Now we will see results in this table and unfortunately we didn't get results as good as last one.

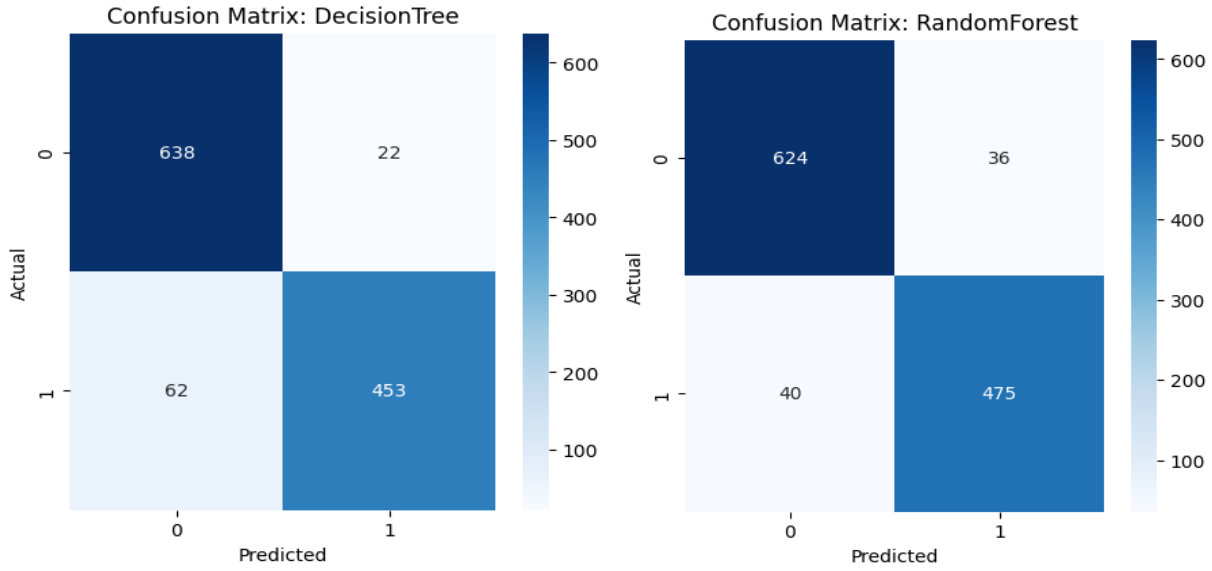
Method	Value Pameters	Accuracy	Precision	Recall	F1
Decision Tree	'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'	0.905	0.922	0.866	0.89
Random Forest	'criterion': 'entropy', 'max_depth': 20, 'min_samples_split': 5, 'n_estimators': 100	0.921	0.92	0.906	0.913
XGBoost	'colsample_bytree': 1.0, 'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100, 'subsample': 0.8	0.972	0.969	0.97	0.97
SVM	'C': 10, 'gamma': 1, 'kernel': 'rbf'	0.859	0.872	0.811	0.841
MLP	'activation': 'tanh', 'alpha': 0.001, 'hidden_layer_sizes': (200, 100), 'learning_rate': 'constant', 'max_iter': 500	0.92	0.914	0.911	0.913

Accuracies on predicting test data:

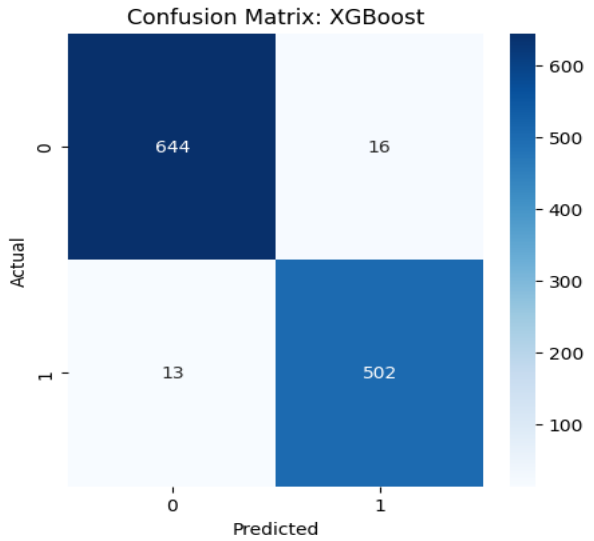
Decision Tree: 92.8% - Random Forest: 93.5% - XGBoost: 97.5% - SVM: 87.8% - MLP: 93.6%

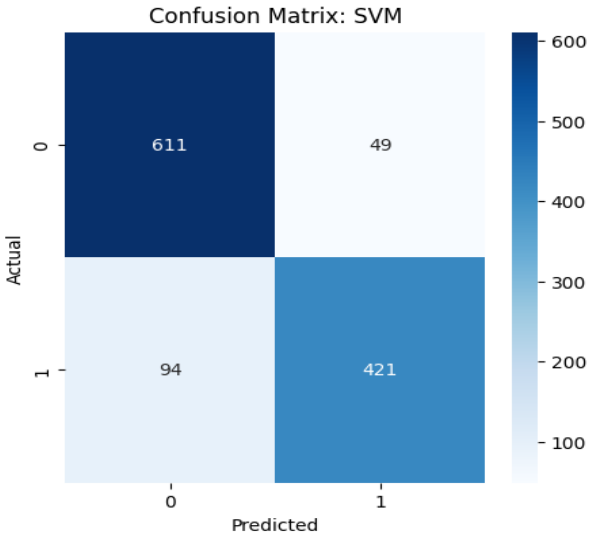
And now we are going to see plots of these data:

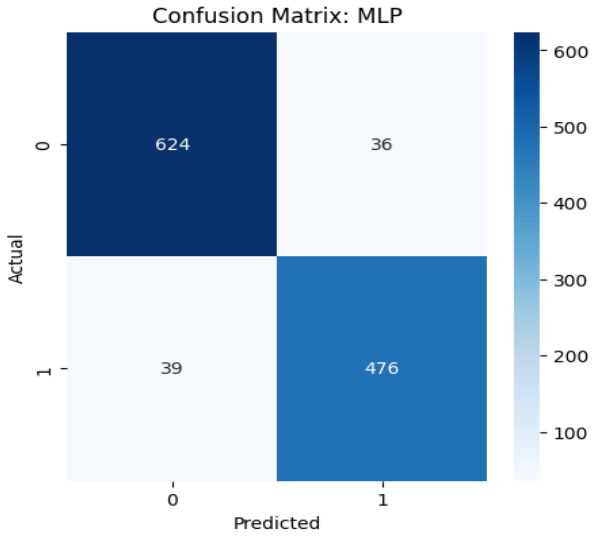
Confusion Matrix:


Confusion Matrix: DecisionTree
638
22
62
453
Actual
Predicted
0
1
600
500
400
300
200
100

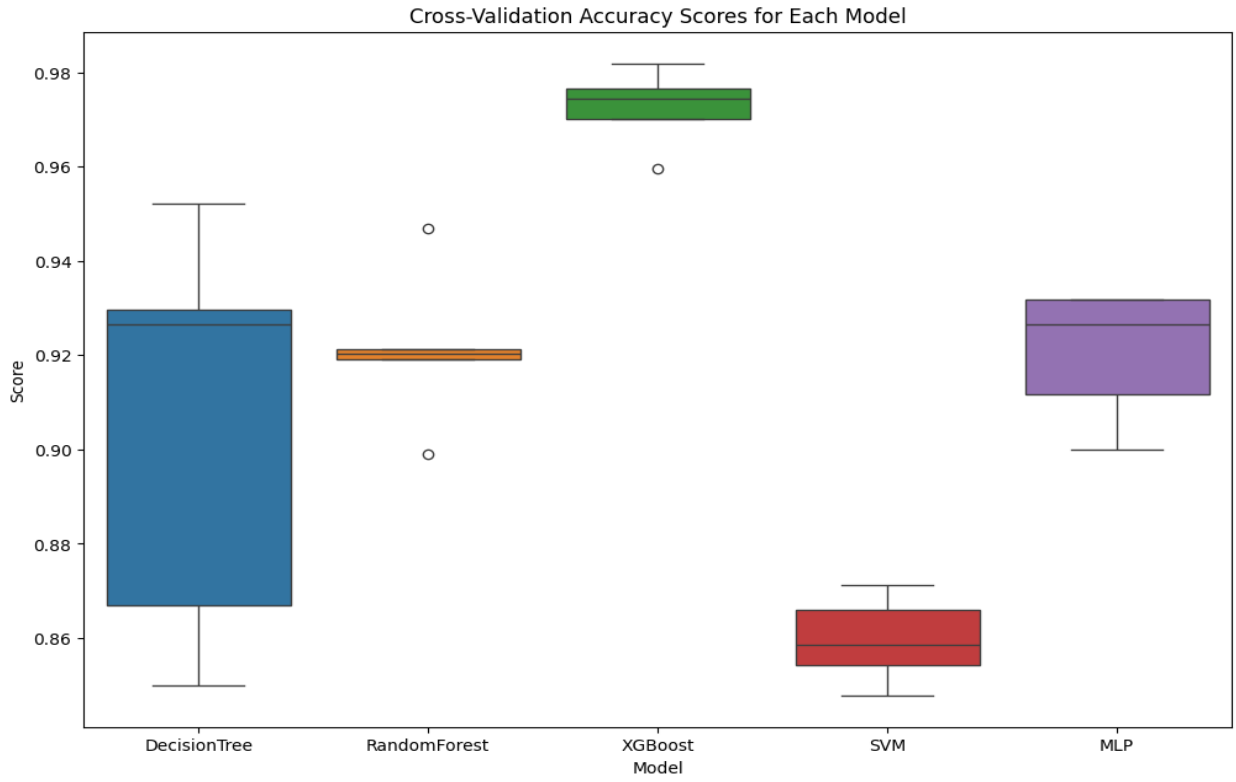
Confusion Matrix: RandomForest
624
36
40
475
Actual
Predicted
0
1
600
500
400
300
200
100

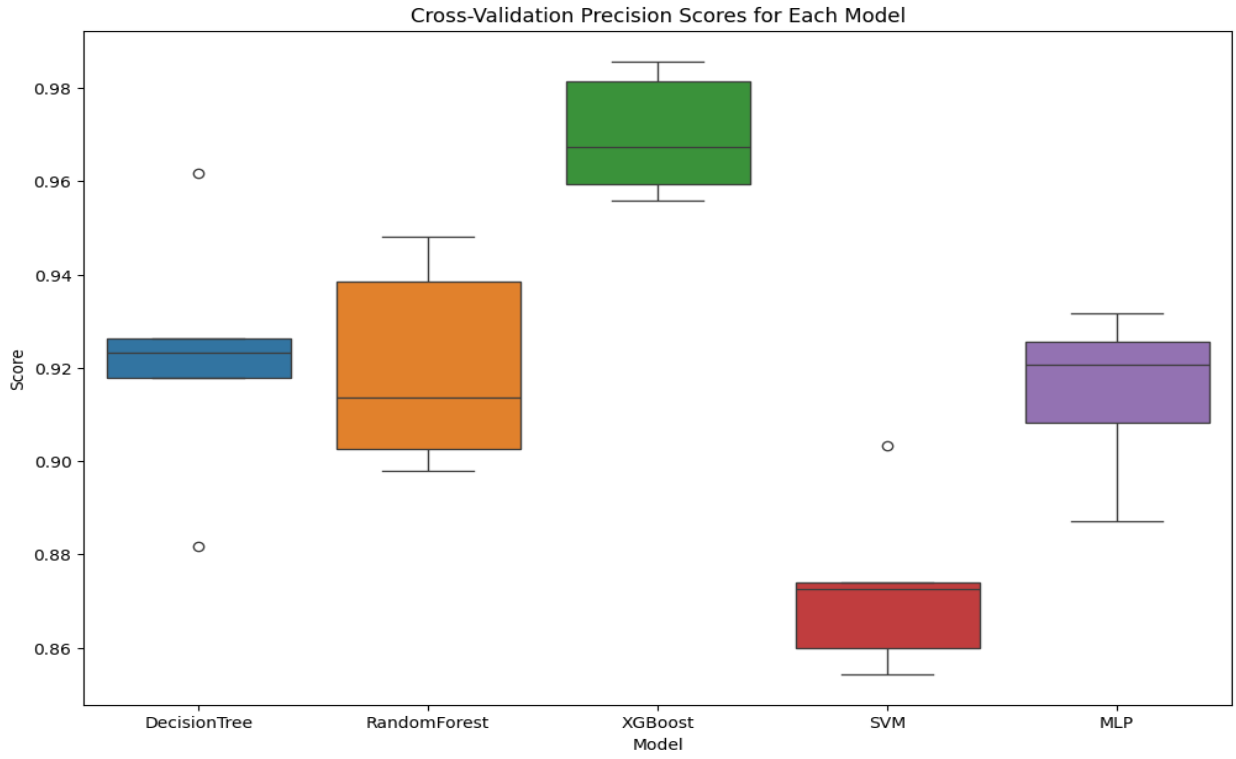

Confusion Matrix: XGBoost
644
16
13
502
Actual
Predicted
0
1
600
500
400
300
200
100

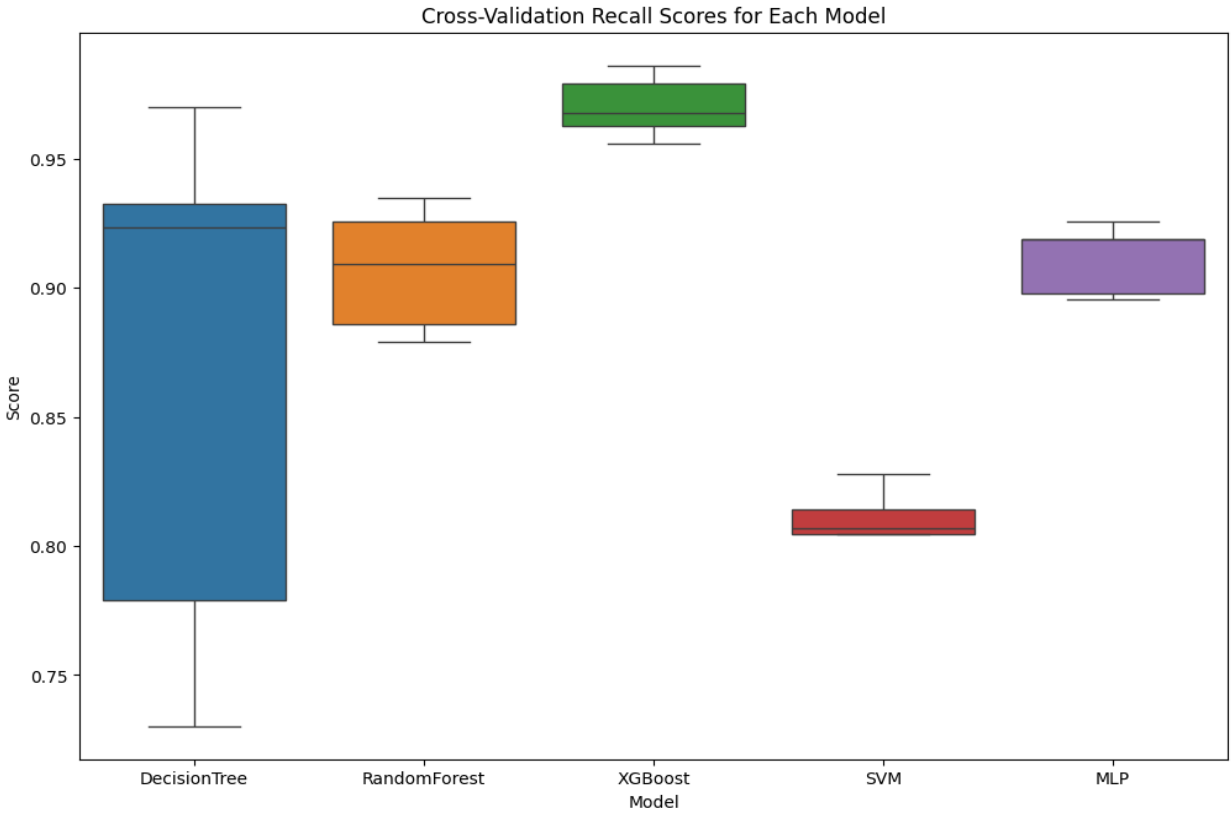

Confusion Matrix: SVM
611
49
94
421
Actual
Predicted
0
1
600
500
400
300
200
100

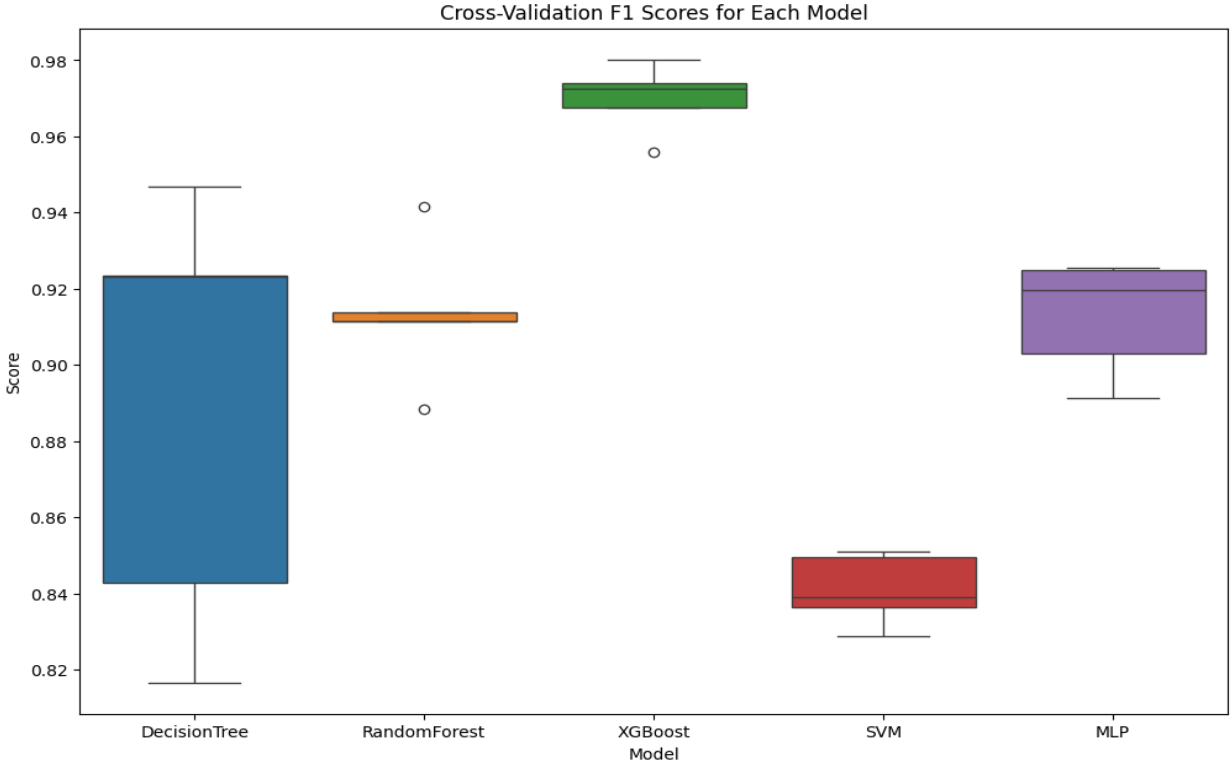

Confusion Matrix: MLP
624
36
39
476
Actual
Predicted
0
1
600
500
400
300
200
100

Box Plots:


Cross-Validation Accuracy Scores for Each Model
Score
0.98
0.96
0.94
0.92
0.90
0.88
0.86
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation Precision Scores for Each Model
Score
0.98
0.96
0.94
0.92
0.90
0.88
0.86
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation Recall Scores for Each Model
Score
0.95
0.90
0.85
0.80
0.75
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation F1 Scores for Each Model
Score
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0.82
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model

Median Classify:

After all, we are going to see the results of median in this table too.

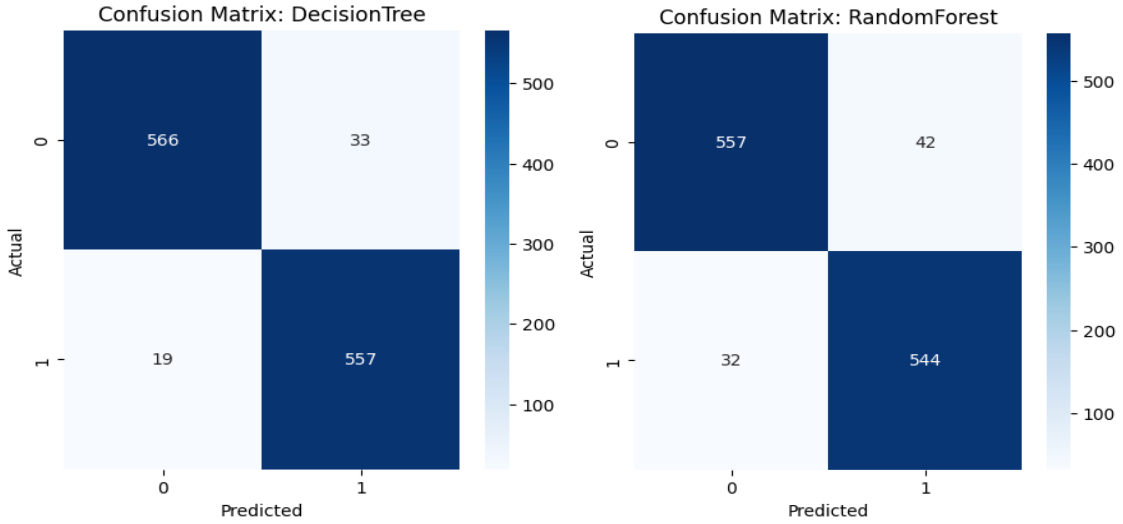
Method	Value Pameters	Accuracy	Precision	Recall	F1
Decision Tree	'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'best'	0.94	0.943	0.938	0.94
Random Forest	'criterion': 'entropy', 'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 100	0.921	0.91	0.94	0.923
XGBoost	'colsample_bytree': 1.0, 'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100, 'subsample': 0.8	0.97	0.97	0.965	0.966
SVM	'C': 10, 'gamma': 1, 'kernel': 'rbf'	0.873	0.85	0.91	0.88
MLP	'activation': 'tanh', 'alpha': 0.001, 'hidden_layer_sizes': (200, 100), 'learning_rate': 'constant', 'max_iter': 500	0.92	0.914	0.92	0.917

Accuracies on predicting test data:

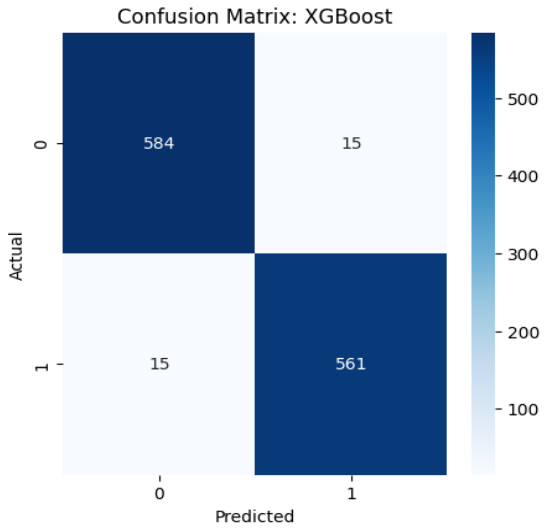
Decision Tree: 95.5% - Random Forest: 93.7% - XGBoost: 97.4% - SVM: 90% - MLP: 93.8%

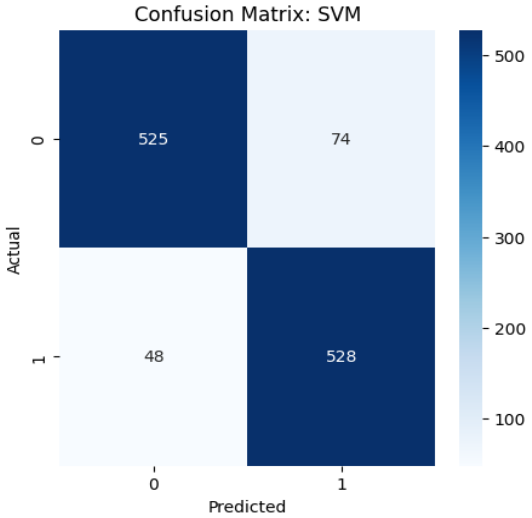
And now we are going to see plots of these data:

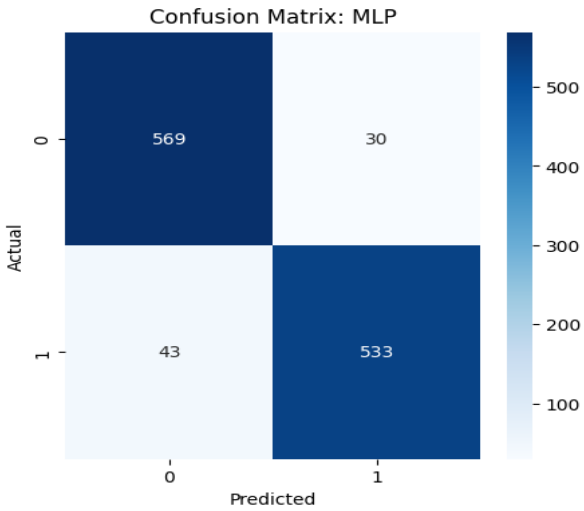
Confusion Matrix:


Confusion Matrix: DecisionTree
566
33
19
557
Actual
Predicted
0
1
500
400
300
200
100

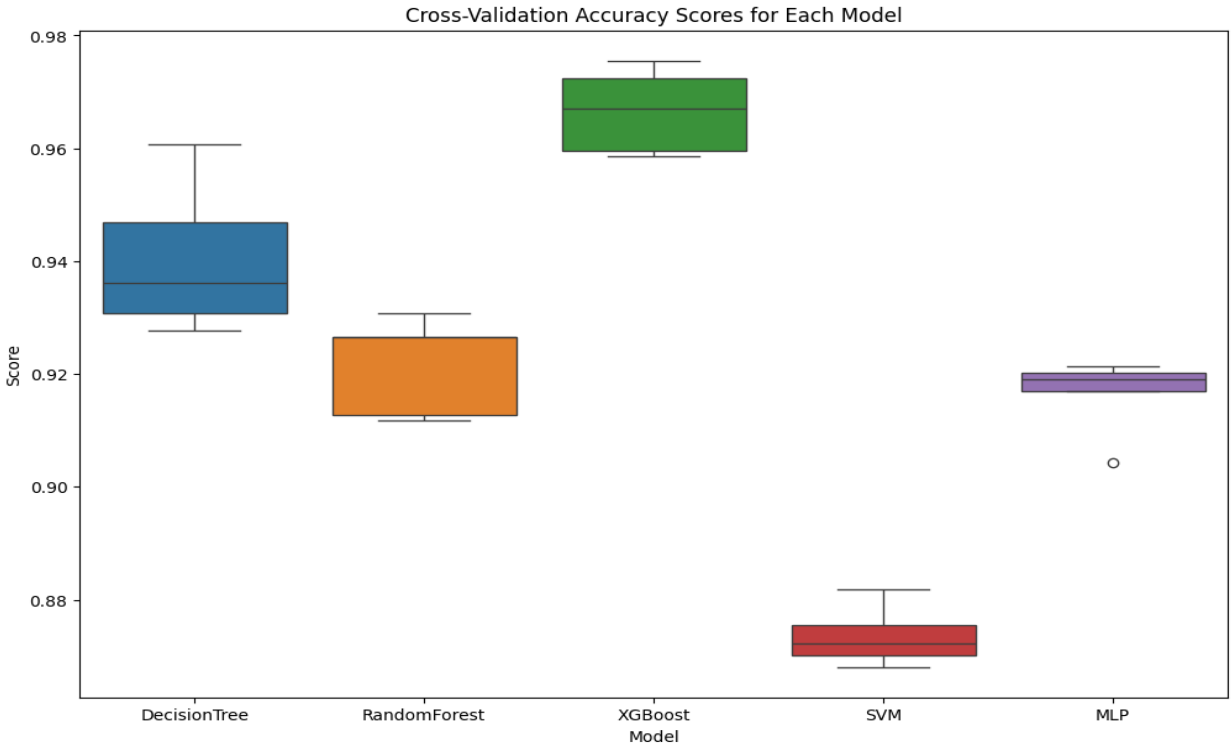
Confusion Matrix: RandomForest
557
42
32
544
Actual
Predicted
0
1
500
400
300
200
100

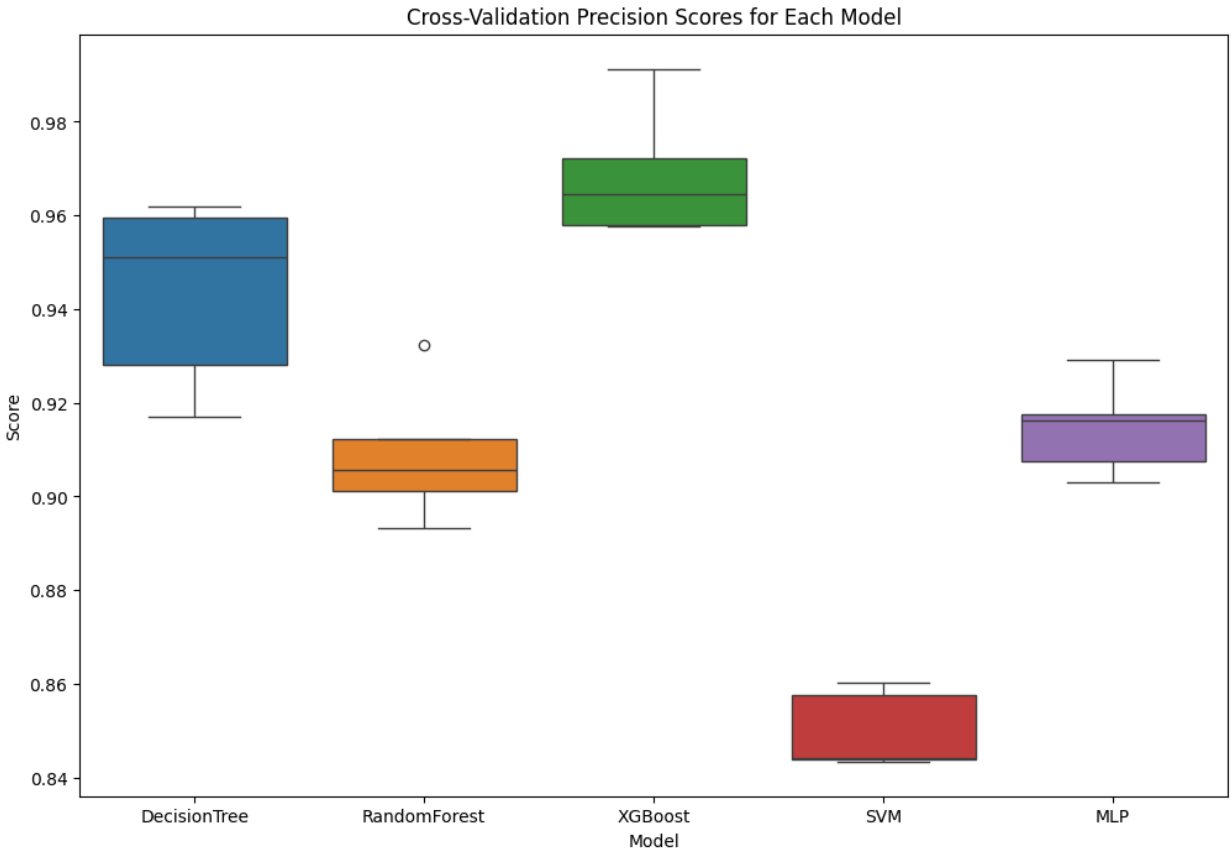

Confusion Matrix: XGBoost
584
15
15
561
Actual
Predicted
0
1
500
400
300
200
100

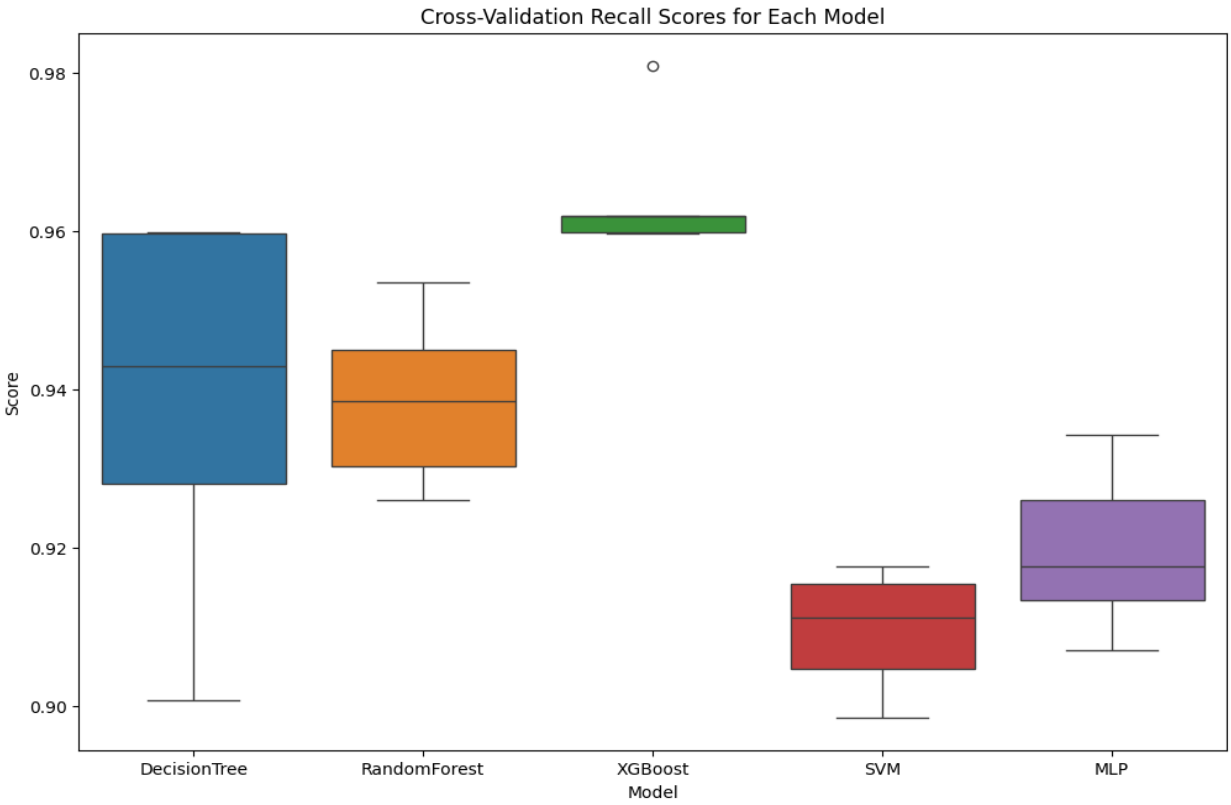

Confusion Matrix: SVM
525
74
48
528
Actual
Predicted
0
1
500
400
300
200
100

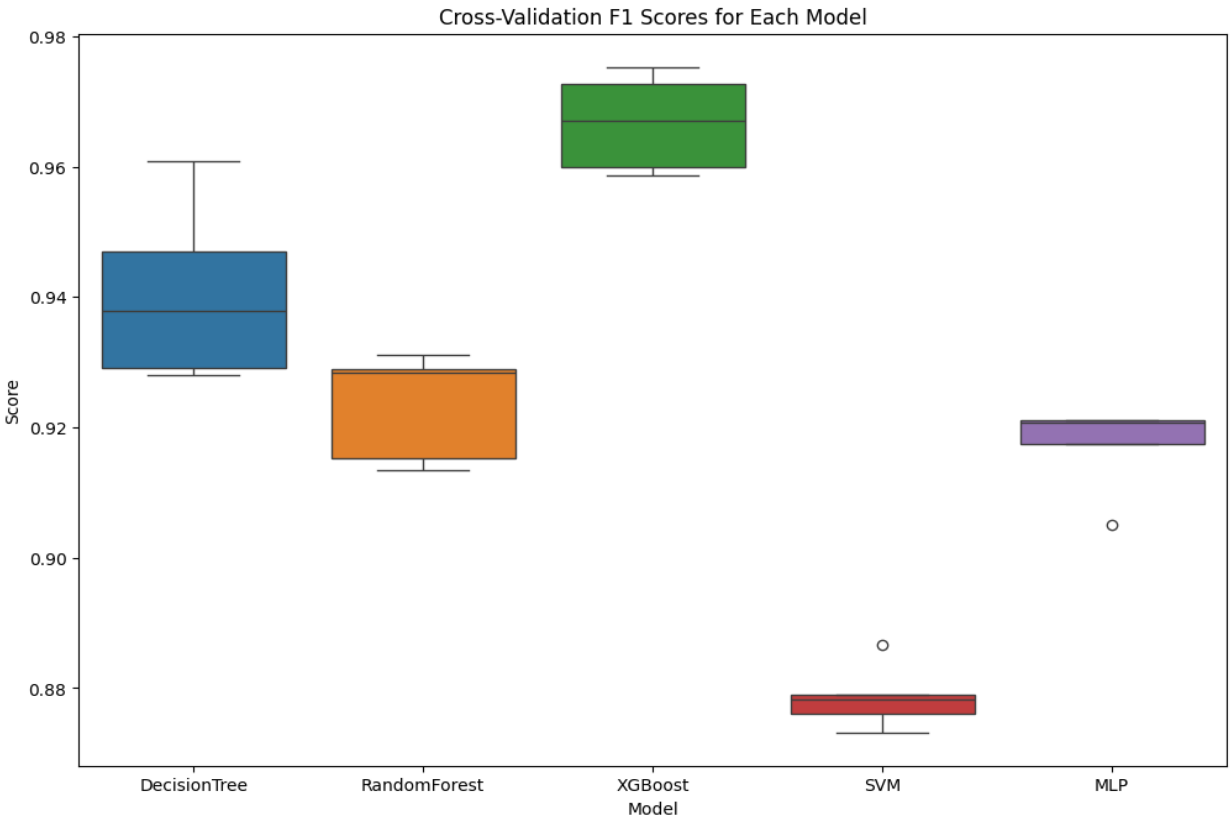

Confusion Matrix: MLP
569
30
43
533
Actual
Predicted
0
1
500
400
300
200
100

Box Plots:


Cross-Validation Accuracy Scores for Each Model
Score
0.98
0.96
0.94
0.92
0.90
0.88
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation Precision Scores for Each Model
Score
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation Recall Scores for Each Model
Score
0.98
0.96
0.94
0.92
0.90
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model


Cross-Validation F1 Scores for Each Model
Score
0.98
0.96
0.94
0.92
0.90
0.88
DecisionTree
RandomForest
XGBoost
SVM
MLP
Model