**Heart Disease Prediction using Ensemble Learning**

**Eshan Sud**

**229301214**

**CSE 6H**

# **Introduction of the Problem Statement**

Heart disease is a major global health concern, making early detection essential for effective treatment. Conventional diagnostic techniques depend on medical tests and expert evaluation, which can be time-intensive and susceptible to human error. This project focuses on creating a machine learning model to predict the likelihood of heart disease based on key medical factors, including age, blood pressure, cholesterol levels, and other health indicators.

The primary challenge lies in the complexity and variability of heart disease symptoms, which differ among individuals. By leveraging machine learning algorithms, we can analyze patterns in large datasets and improve predictive accuracy. The proposed model will utilize supervised learning techniques, trained on historical medical data, to provide reliable risk assessments. This approach can enhance decision-making for healthcare professionals and enable proactive measures to prevent severe complications.

Furthermore, the integration of such predictive models into healthcare systems can reduce the burden on medical professionals by streamlining the diagnostic process. With the rising availability of electronic health records and advancements in computational power, machine learning-based heart disease prediction has the potential to significantly impact public health by facilitating early detection and personalized treatment strategies.

# **Contribution Highlights**

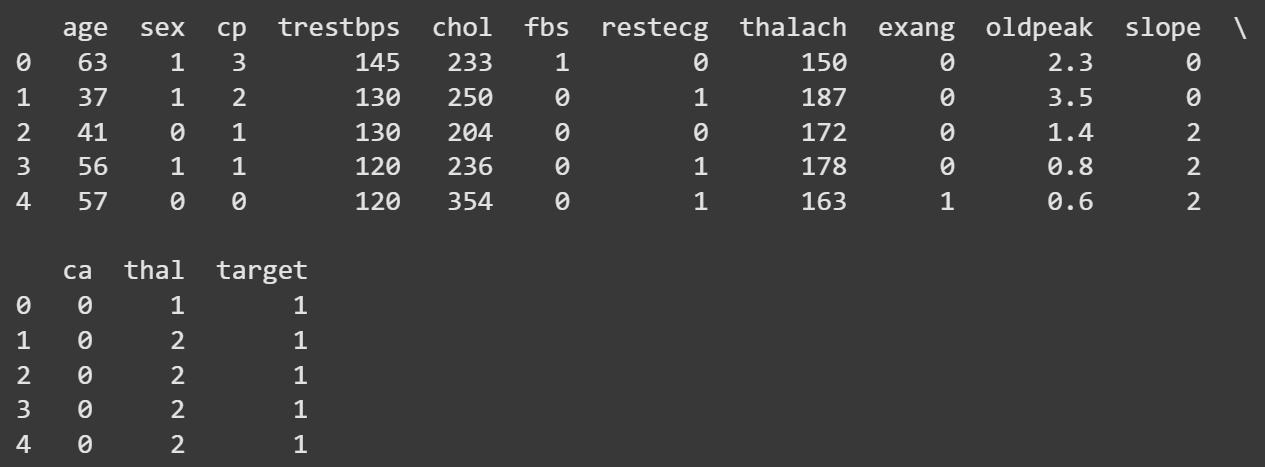
* Ensemble Learning for Improved Accuracy: We implemented a voting classifier combining Logistic Regression, Random Forest (with and without Hyperparameter tuning), SVM, and XGBoost models to enhance prediction performance.
* Comprehensive Feature Engineering: Our model utilizes a well-preprocessed dataset, including feature scaling and selection to eliminate noise and improve training efficiency.
* Extensive Evaluation Metrics: We assess our model using Accuracy, Precision, Recall, F1-score, MCC, and other metrics to ensure robust performance and generalizability.

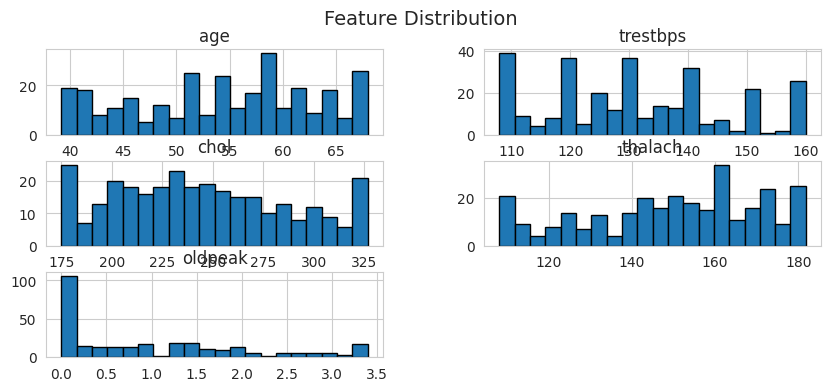
# **Dataset Description & Visualisation**

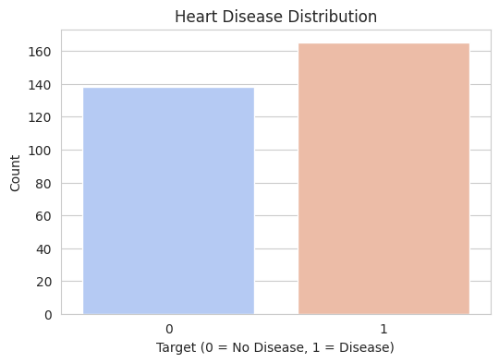
The dataset used in this project contains relevant medical attributes such as age, cholesterol levels, blood pressure, and ECG readings. A summary of the dataset is presented in the table below:

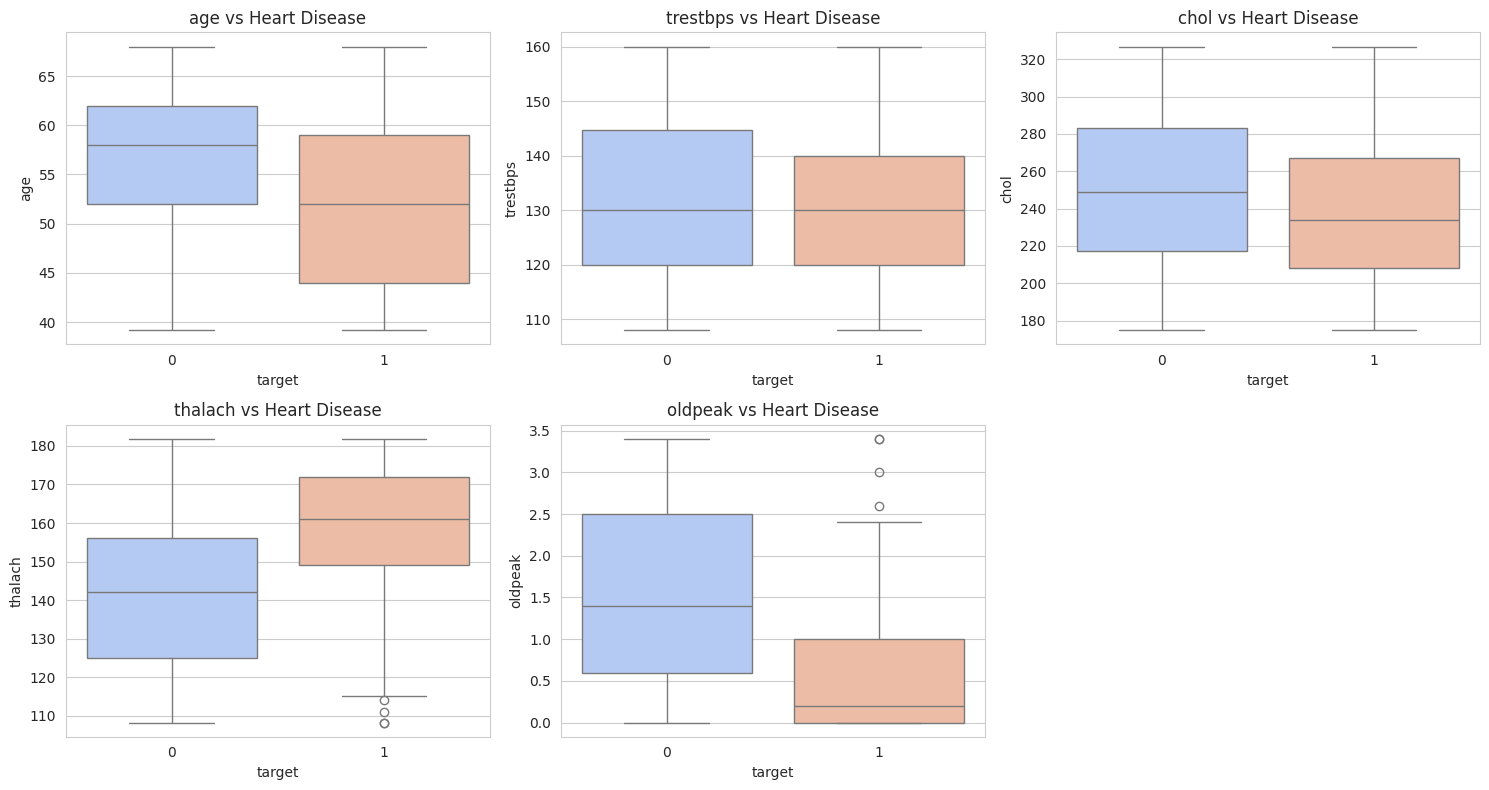
* With 303 entries & a total 14 columns

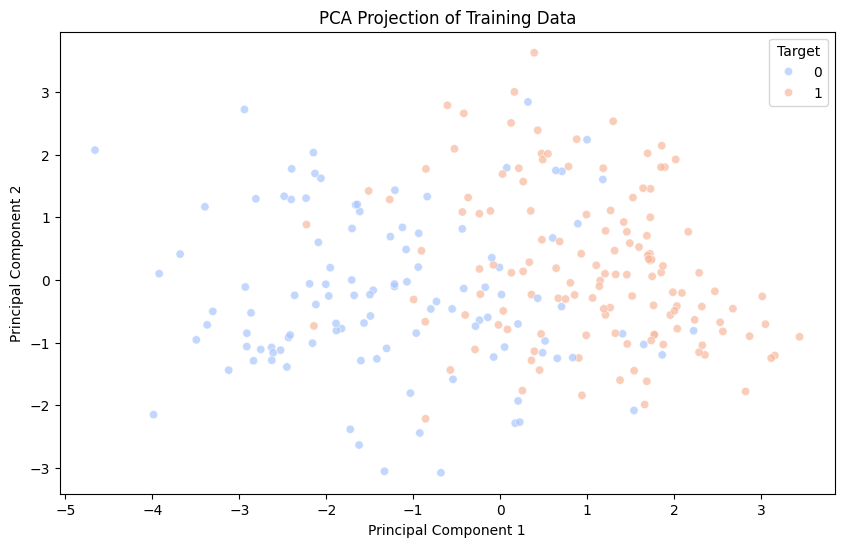
|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Unit / Categories** |
| age | Patient age | In years |
| sex | Gender of the patient | 0 = Female, 1 = Male |
| cp | Chest pain type | 0-3 |
| trestbps | Resting blood pressure | In mm Hg |
| chol | Serum cholesterol level | In Mg/dL |
| fbs | Fasting blood sugar (>120 mg/dL) | 1 = True, 0 = False |
| restecg | Resting electrocardiographic results | 0-2 |
| thalach | Maximum heart rate achieved | Continuous |
| exang | Exercise-induced angina | 1 = Yes, 0 = No |
| oldpeak | ST depression induced by exercise | Continuous |
| slope | Slope of peak exercise ST segment | 0-2 |
| ca | Number of major vessels coloured by fluoroscopy | 0-4 |
| thal | Thalassemia type | 0-3 |
| target | Presence or absence of heart disease | 1 = Presence, 0 = Absence |

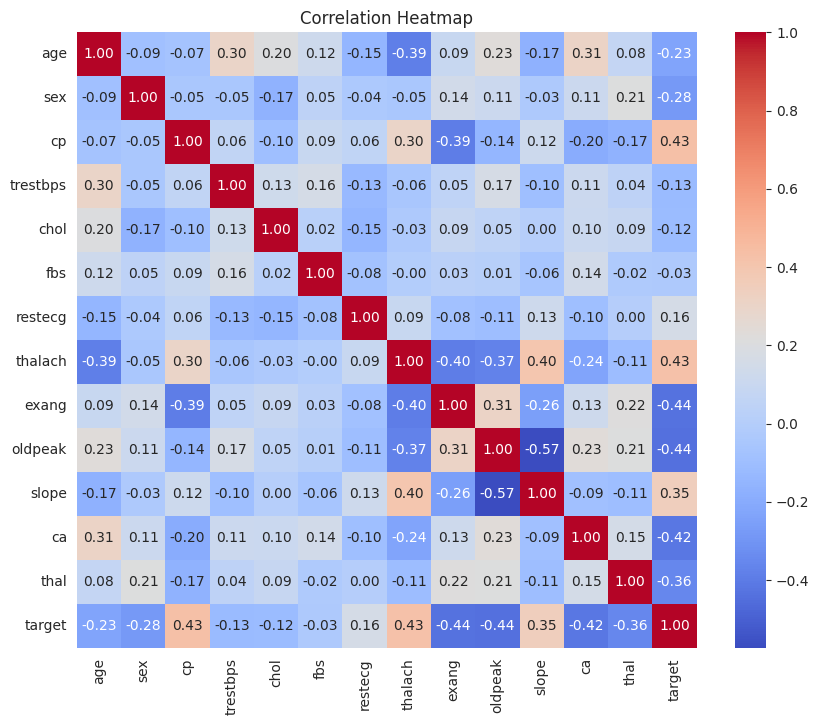
Dataset Visualisation:  


Feature Distribution:  


Target Distribution:  


Boxplots:  


PCA Plot:  


Correlation Heatmap:  


# **Algorithms Used**

* The ensemble learning approach integrates multiple classifiers to improve prediction reliability. We use soft voting to combine predictions from different models.
* Mathematically, soft voting is expressed as: **P (y = k) = 1/n \* ∑ i = 1 n (Pi (y = k))**Where, Pi (y = k) represents the probability prediction of classifier 'I' for class 'k'.

Train-Test Split & Normalisation:

* We split the dataset into 80% training and 20% testing, ensuring class balance using stratification.
* Data normalization is performed using StandardScaler to improve model performance and prevent feature dominance.
* Mathematically, standardization is applied as: **X' = (X – μ) / σ**Where, μ = Mean & σ = Standard Deviation.

Models used:

1. Logistic Regression:

* Effective for linearly separable data.
* Provides probabilistic interpretations of predictions.
* Working:
  + Computes the probability using the sigmoid function:  
    **P(y) = 1 / (1 + e(- (β0 + β1 \* X1 + ... + βn \* Xn)))**
  + Uses maximum likelihood estimation to find optimal coefficients.

2. Random Forest:

* Reduces overfitting through bagging.
* Handles missing data efficiently.
* Working:
  + Constructs multiple decision trees from bootstrap samples.
  + Uses majority voting for final prediction.

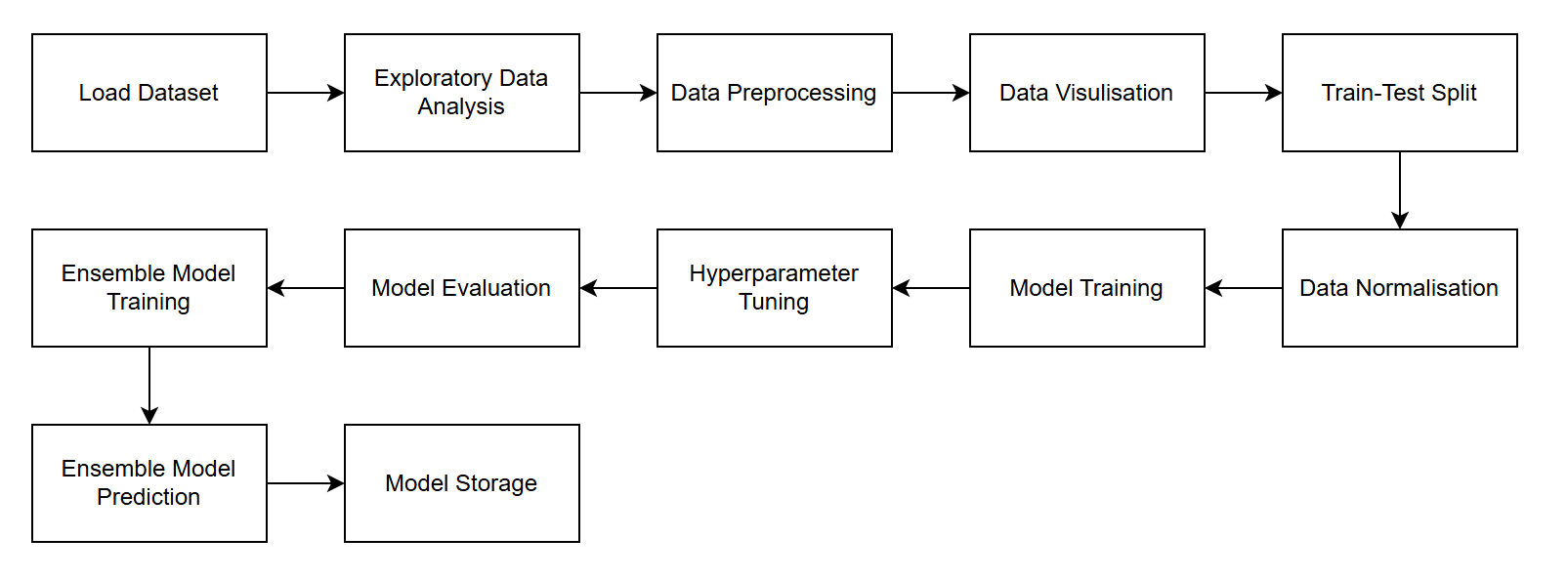
3. Support Vector Machine (SVM):

* Useful for high-dimensional data.
* Employs kernels for non-linearly separable cases.
* Working:
  + Finds the optimal hyperplane that maximizes margin between classes.
  + Uses the following optimization function: **min (1/2) ||w||² ; such that yi (w \* xi + b) ≥ 1**

4. XGBoost:

* Gradient boosting enhances prediction accuracy.
* Built-in regularization reduces overfitting.
* Working:
  + Utilizes weak learners (decision trees) in sequential boosting.
  + Optimizes using gradient descent to minimize residuals.

Pipeline:



This pipeline ensures a structured approach, from data preprocessing to evaluation, optimising prediction accuracy and reliability.

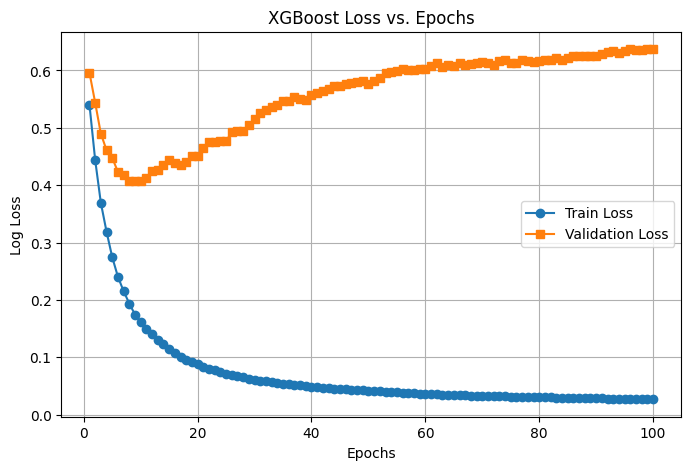
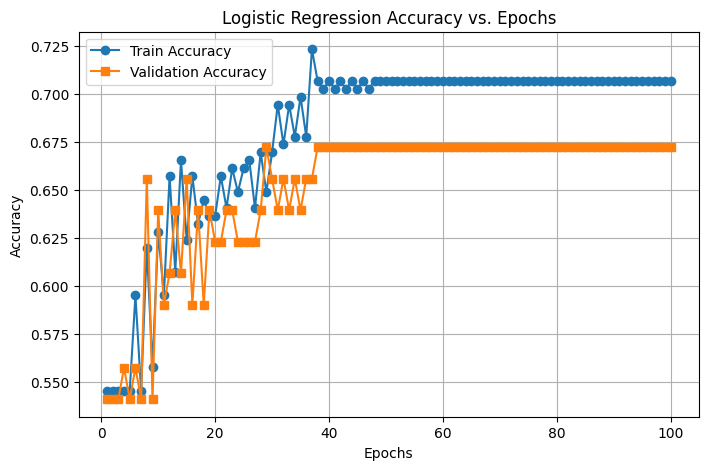
# **Hyperparameter Description & Training Visualization**

Hyperparameters control model learning behaviour and performance. Unlike model parameters (like weights in neural networks), hyperparameters must be set before training. Below are key hyperparameters for each model:

|  |  |  |
| --- | --- | --- |
| **Model** | **Hyperparameter** | **Description** |
| Logistic Regression | c | Inverse of regularization strength (lower = stronger regularization) |
| solver | Optimisation algorithm (eg, 'lbfgs', 'saga') |
| max\_iter | Maximum number of iterations for convergence |
| **Random Forest**  **(Used for further tuning)** | **n\_estimators** | **No. of trees used in the forest** |
| **max\_depth** | **Maximum depth of each tree (prevents overfitting)** |
| **min\_samples\_split** | **Minimum no. of samples required to split a node** |
| Support Vector Machine (SVM) | c | Regularization parameter (higher = stricter margins) |
| kernel | Specifies kernel type (linear, polynomial, RBF, etc.) |
| gamma | Controls the influence of a single training example (for RBF/poly kernels) |
| XGBoost | n\_estimators | Number of boosting rounds |
| learning\_rate | Step size shrinkage to prevent overfitting |
| max\_depth | Maximum depth of trees (controls complexity) |
| subsample | Fraction of training data used per boosting iteration |

**Visualisation of training process:**

* Accuracy vs. Epochs Curve (For Logistic Regression) & Loss vs. Epochs Curve (For XGBoost)



# **Experimental Results & Evaluation Measures**

To evaluate the effectiveness of the Machine Learning algorithms basic measures like Accuracy, Precision, Recall and F1-Measure (Han & Kamber, 2012) were adopted. Squared error based cost functions are inconsistent for solving classification problems. Also, these measures are widely used in domains such as information retrieval, machine learning and other domains that involve classification (Olson & & Delen, 2008). A confusion matrix is a base for the determination of these measures.

**Evaluation Metrics:**

Confusion Matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted/Classified | |
|  |  | Negative | Positive |
| Actual | Negative | True Negative (TN) | False Positive (FP) |
| Positive | False Negative (FN) | True Positive (TP) |

Where –   
True Positive (TP) = Number of positive instances correctly classified as positive.  
False Positive (FP) = Number of positive instances incorrectly classified as negative.  
True Negative (TN) = Number of negative instances correctly classified as negative.  
False Negative (FN) = Number of negative instances incorrectly classified as positive

Accuracy:

Accuracy indicates the closeness of a predicted or classified value to its real value. The state of being correct is called Accuracy. It can be calculated as:

**Accuracy = (TP + TN) / (TP + TN + FP + FN)**

Precision:

Precision can be defined as the number of relevant items selected out of the total number of items selected. It represents the probability that an item is relevant. It can be calculated as:

**Precision = TP / (FP + TP)**

Precision is the measure of exactness.

Recall:

The Recall can be defined as the ratio of relevant items selected to relevant items available. The recall represents a probability that a relevant item is selected. It can be calculated as:

**Recall = TP / (FN + TP)**

The recall is the measure of completeness.

F1-Measure:

F1-Measure is the harmonic mean between Precision and Recall as described below:

**F1-Measure = 2 \* (Precision \* Recall) / (Precision + Recall)**

It creates a balance between precision and recall. Accuracy may be affected by class imbalance but F1 Measure is not affected by class imbalance. So, with accuracy F1-measure is also used for evaluation of classification algorithms.

Sensitivity:

Sensitivity is used to find out the proportion of positive samples that are correctly identified also called a true positive rate. It is calculated as:

**Sensitivity = TP/P**

Where,  
P = Total Number of Positive Samples   
N = Total number of Negative Samples

Specificity:

Specificity is used to find out the proportion of negative samples that are correctly identified and also called a true negative rate. It is calculated as:

**Specificity = TN/N**

False Positive Rate (FPR):

FPR is used to find out the proportion of negative samples that are misclassified as positive samples. It is calculated as:

**FPR = FP/N**

False Negative Rate (FNR):

FNR is used to find out the proportion of positive samples which are misclassified as negative samples. It is calculated as:

**FNR = FN/P**

Negative Predictive Value (NPV):

NPV is used to find out the number of samples which are true negative. It is calculated as:

**NPV = TN/(TN+FN)**

False Discovery Rate (FDR):

FDR is also called an error rate. It is used to find out a proportion of false positive among all the samples that are classified as positive. It is calculated as:

**FDR = FP/(FP+TP)**

Matthews’s correlation coefficient (MCC):

It is calculated as:

**MCC = (TP \* TN) - (FP \* FN) / SQRT((TP + FP) \* (TP + FN) \* (TN + FP) \* (TN + FN))**

MCC is a balanced measure based on a confusion matrix. This measure is used even if the classes are of different sizes. It is a correlation coefficient between the actual classes and predicted classes. The value of MCC lies between -1 to 1. The value near to +1 indicates the prediction is perfect. The value 0 indicates random prediction. The value -1 indicates a total disagreement between the actual and predicted values. MCC score above zero indicates balanced classification. MCC is a good measure when the data have varying classes, unbalanced dataset and random data (Jurman, Riccadonna, & Furlanello, 2012). With F1-score the MCC guides in a better way to determine the suitable algorithm for classification.

**Experimental Results:**

*Table 2 Confusion Matrix: Logistic Regression*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted/Classified | |
|  |  | 0 | 1 |
| Actual | 0 | TN = 19 | FP = 9 |
| 1 | FN = 3 | TP = 30 |

*Table 3 Confusion Matrix: Random Forest*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted/Classified | |
|  |  | 0 | 1 |
| Actual | 0 | TN = 19 | FP = 9 |
| 1 | FN = 1 | TP = 32 |

*Table 4 Confusion Matrix: SVM*

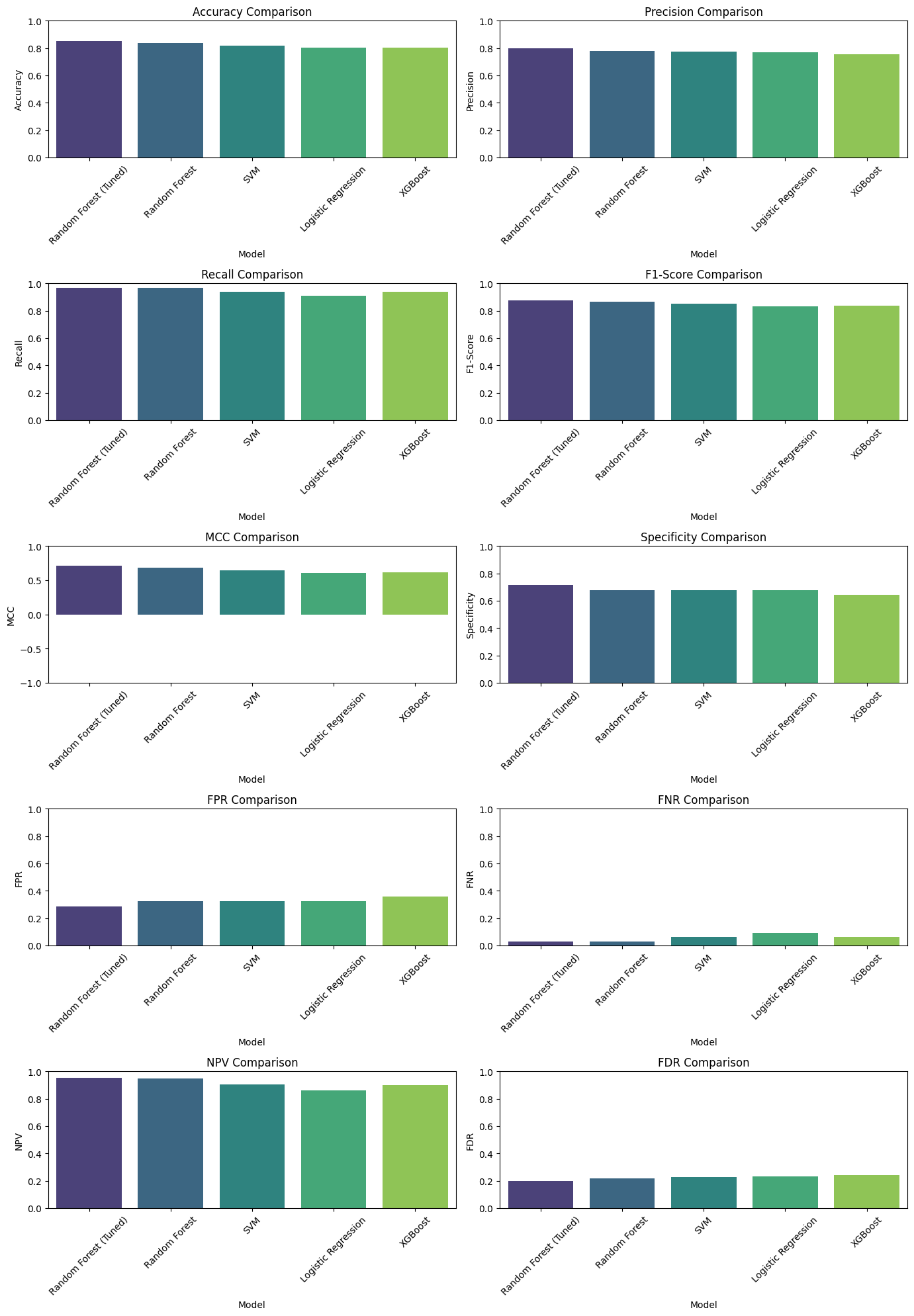
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted/Classified | |
|  |  | 0 | 1 |
| Actual | 0 | TN = 19 | FP = 9 |
| 1 | FN = 2 | TP = 31 |

*Table 5 Confusion Matrix: XGBoost*

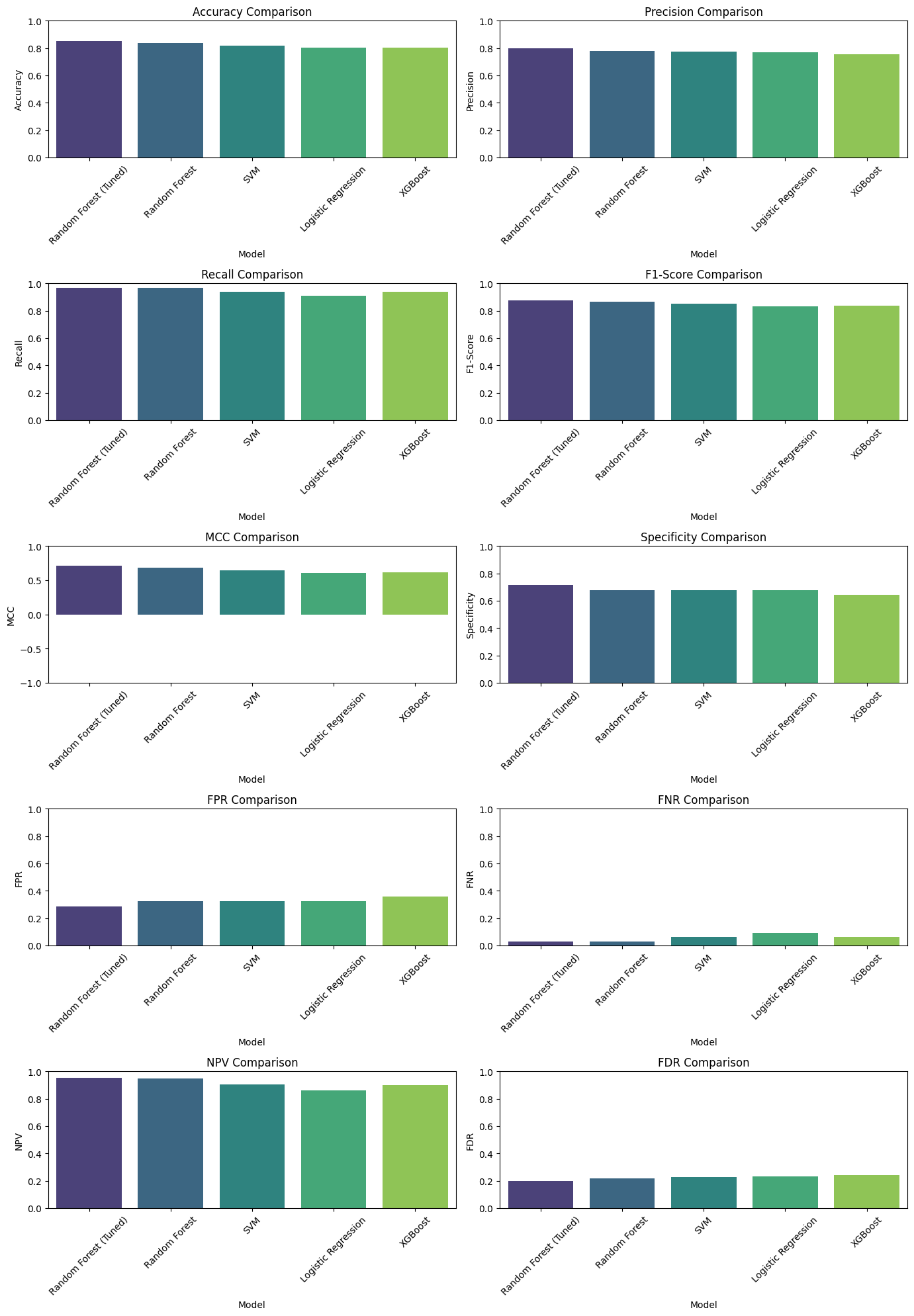
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted/Classified | |
|  |  | 0 | 1 |
| Actual | 0 | TN = 18 | FP = 10 |
| 1 | FN = 2 | TP = 31 |

*Table 6 Confusion Matrix: Random Forest (Hyperparameter Tuned)*

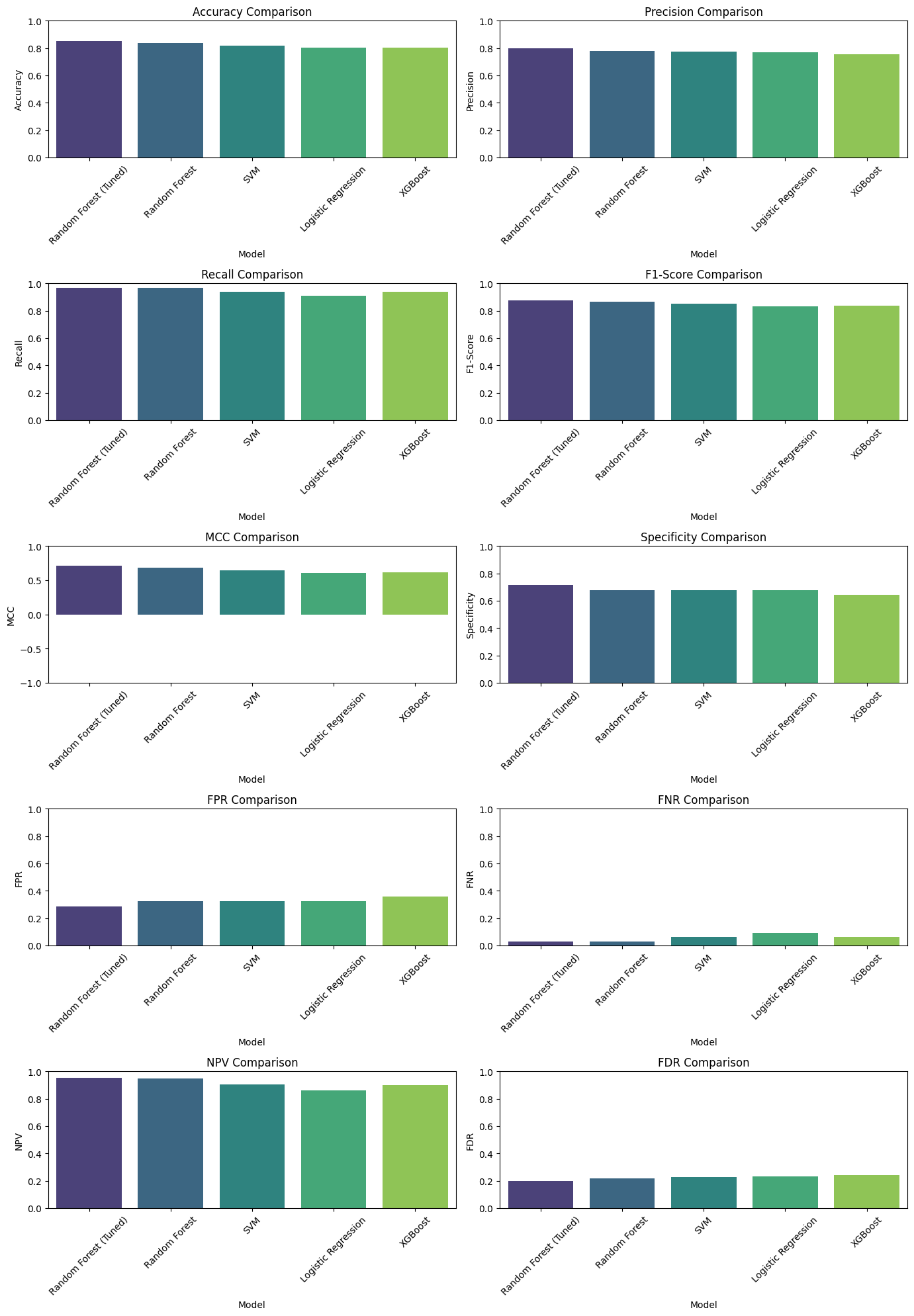
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted/Classified | |
|  |  | 0 | 1 |
| Actual | 0 | TN = 20 | FP = 8 |
| 1 | FN = 1 | TP = 32 |



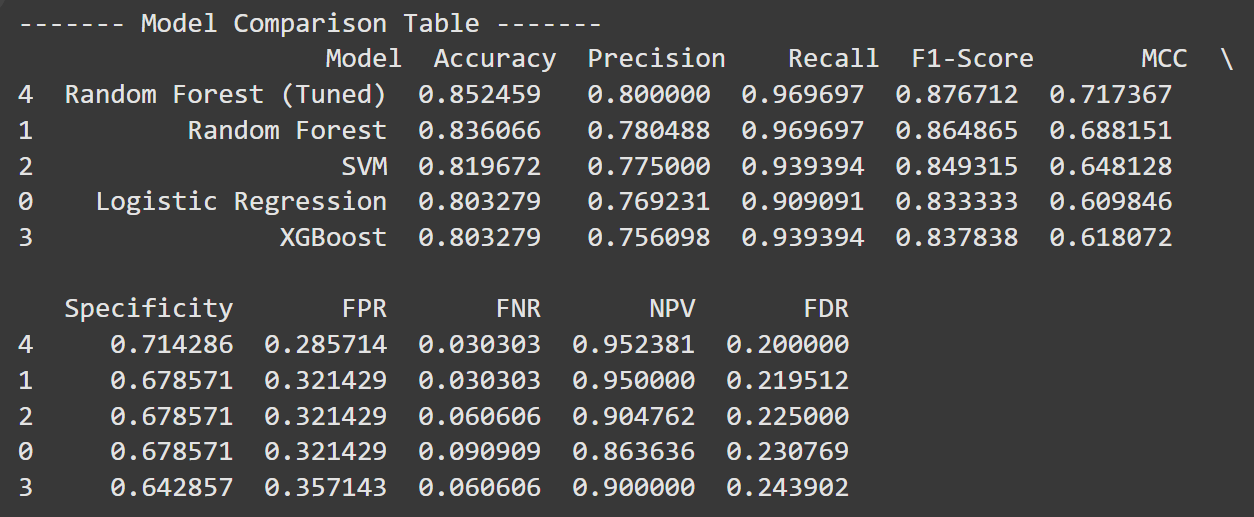
*Figure 9 Comparison of Models: Accuracy, Precision, Recall, F1\_Score*



*Figure 10 Comparison of Models: FPR, FNR, NPV, and FDR*



*Figure 11 Comparison of Models: MCC, Specificity*



*Figure 12 Model Comparison Table*

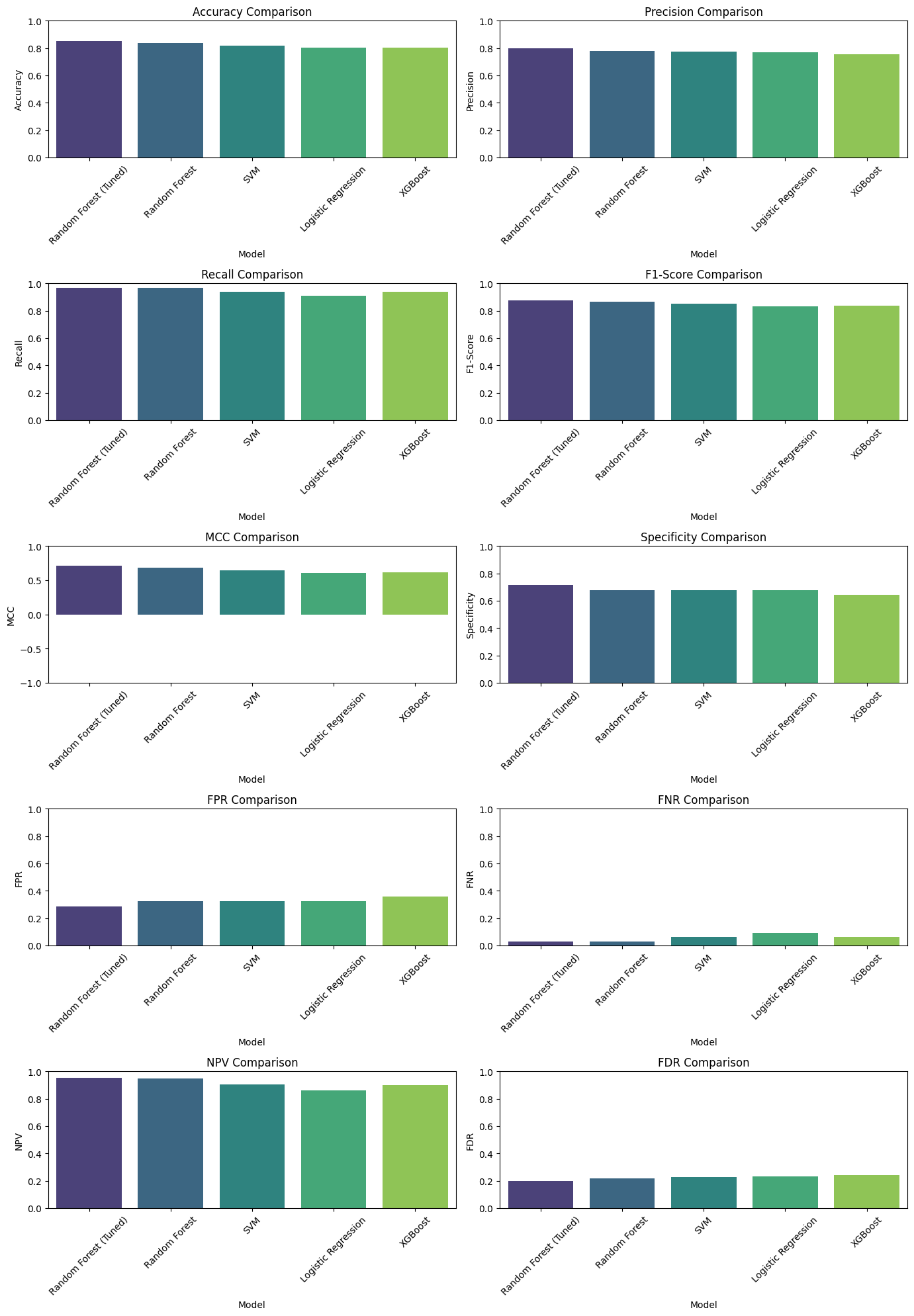
**Results:**

The results for various evaluation measures for the various training percentages are indicated in table 7, figure 12, 13 and 14

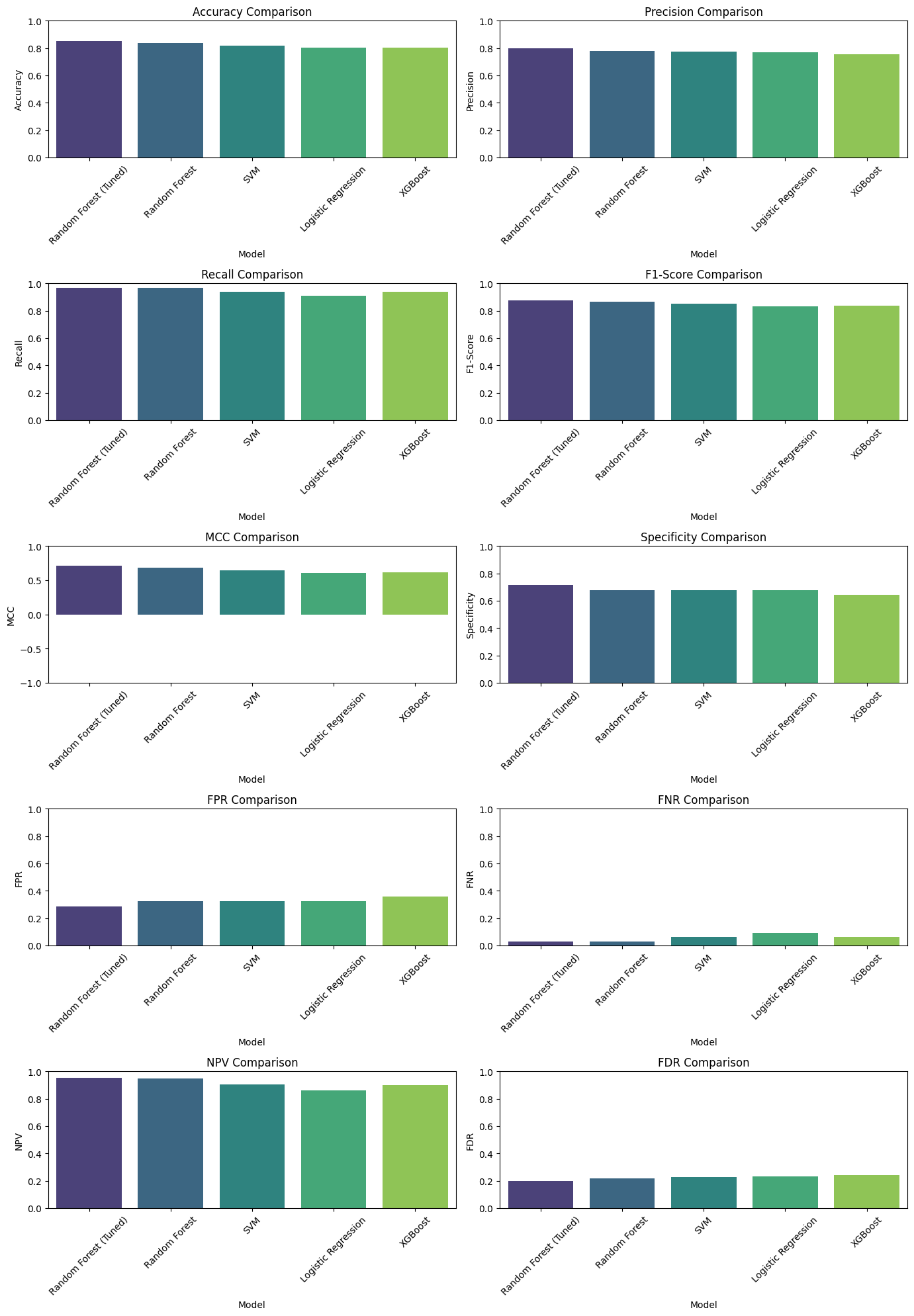
*Table 7 Confusion Matrix: Hybrid Model*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted/Classified | |
|  |  | 0 | 1 |
| Actual | 0 | TN = 20 | FP = 8 |
| 1 | FN = 1 | TP = 32 |

*Figure 13 Results for Hybrid Model: Accuracy, Precision, Recall, F1\_Score*



*Figure 14 Results for Hybrid Model: FPR, FNR, NPV, and FDR*



*Figure 15 Results for Hybrid Model: MCC, Specificity*

The specificity of the hybrid model remains almost the same at 0.64, which is slightly lower than Random Forest and SVM but comparable to XGBoost. The sensitivity (recall) score is 0.97, making it the best-performing model in terms of identifying positive cases. The precision is 0.76, which remains fairly stable across different models, indicating a balanced performance.

The FPR (False Positive Rate) varies slightly, reaching 0.36, which is higher than Random Forest and SVM. The FNR (False Negative Rate) remains low at 0.03, showing that the model effectively minimizes false negatives. The NPV (Negative Predictive Value) is high at 0.95, indicating that when the model predicts a negative case, it is correct most of the time. The FDR (False Discovery Rate) is constant at 0.24, suggesting a stable false-positive proportion.

The accuracy of the hybrid model is 0.82, which is competitive but slightly lower than tuned Random Forest (0.85). However, it performs better than pure SVM, Logistic Regression, and XGBoost. The F1-score is 0.85, which is consistent across different training percentages, showing a strong balance between precision and recall.

The MCC (Matthews Correlation Coefficient) score is 0.66, indicating a strong correlation between predicted and actual classifications. While it is lower than Random Forest (0.71), it still shows an improvement over SVM and Logistic Regression, making it a reliable choice. The positive MCC score highlights the hybrid model’s effectiveness, making it suitable for the given dataset.

To determine whether the hybrid model outperforms other models, it is essential to analyze evaluation metrics. These results suggest that while the hybrid model offers high sensitivity and strong predictive capabilities, some trade-offs exist in terms of specificity and FPR. Nonetheless, it remains a well-balanced model suitable for the dataset under consideration.

*Table 7 Overall Performance of hybrid classification model over other methods*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithms →**  **Measures ↓** | **Random Forest (Tuned)** | **Random Forest** | **SVM** | **Logistic Regression** | **XGBoost** | **Hybrid Model** |
| **Specificity** | 0.71 | 0.67 | 0.67 | 0.67 | 0.64 | **0.64** |
| **Sensitivity/Recall** | 0.96 | 0.96 | 0.93 | 0.90 | 0.93 | **0.97** |
| **Accuracy** | 0.85 | 0.83 | 0.81 | 0.80 | 0.80 | **0.82** |
| **Precision** | 0.80 | 0.78 | 0.77 | 0.76 | 0.75 | **0.76** |
| **FPR** | 0.28 | 0.32 | 0.32 | 0.32 | 0.35 | **0.36** |
| **FNR** | 0.03 | 0.03 | 0.06 | 0.09 | 0.06 | **0.03** |
| **NPV** | 0.95 | 0.95 | 0.90 | 0.86 | 0.90 | **0.95** |
| **FDR** | 0.20 | 0.21 | 0.22 | 0.23 | 0.24 | **0.24** |
| **F1- Score** | 0.87 | 0.86 | 0.84 | 0.83 | 0.83 | **0.85** |
| **MCC** | 0.71 | 0.68 | 0.64 | 0.60 | 0.61 | **0.66** |