**The six models utilized for predicting heart disease are:**

1. Logistic regression

2. Decision tree Classifier

3. Random forest Classifier

4. K Neighbour Classifier

5. XGboost Classifier

6. Ada Boost

**MODEL I: LOGISTIC REGRESSION**

Logistic regression is a widely used statistical model that allows for multivariate analysis and modeling of a binary outcome variable. It is a supervised machine learning algorithm that predicts the probability of an event occurring and, in our case, can predict if a person will have a heart attack or not.

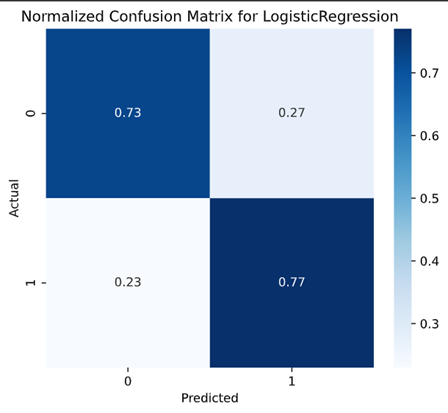
1. **Reasons for choosing the Logistic regression model: -**
2. Logistic regression is well-suited for binary classification problems, where the outcome variable has two possible values (e.g. heart disease or no heart disease).
3. It can handle multiple independent variables (risk factors like high blood pressure, smoking, obesity, etc.) and model their interactions to predict the outcome.
4. Logistic regression provides interpretable results in the form of odds ratios, which can help quantify the impact of different risk factors.
5. The model is relatively easy to implement, train, and interpret compared to more complex machine learning techniques.
6. Logistic regression makes fewer assumptions than other statistical models, making it a robust choice for analyzing complex real-world data.
7. **Evaluation Metrics: -**

The model is evaluated using several performance metrics, including accuracy score, F1 score, precision score, and recall score.

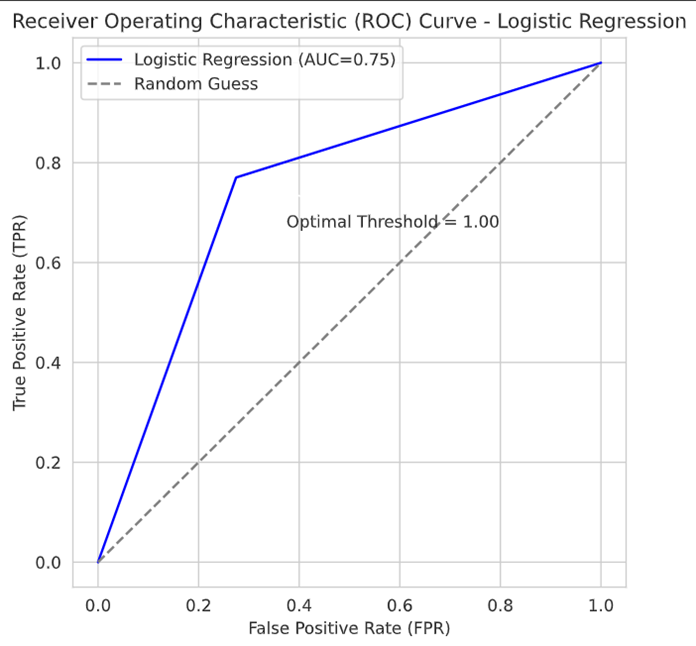
* Accuracy Score: 0.75
* F1 Score: 0.75
* Precision Score: 0.73
* Recall Score: 0.77

The logistic regression model has an accuracy, F1 score, precision, and recall of 0.75. It shows high recall (0.77), correctly identifying most positive cases (heart disease), but lower precision (0.73), indicating false positive predictions. This pattern is common in imbalanced datasets, where negative examples (no heart disease) dominate. Though the F1 score suggests reasonable overall performance, addressing dataset imbalance and fine-tuning the threshold could improve precision-recall balance.

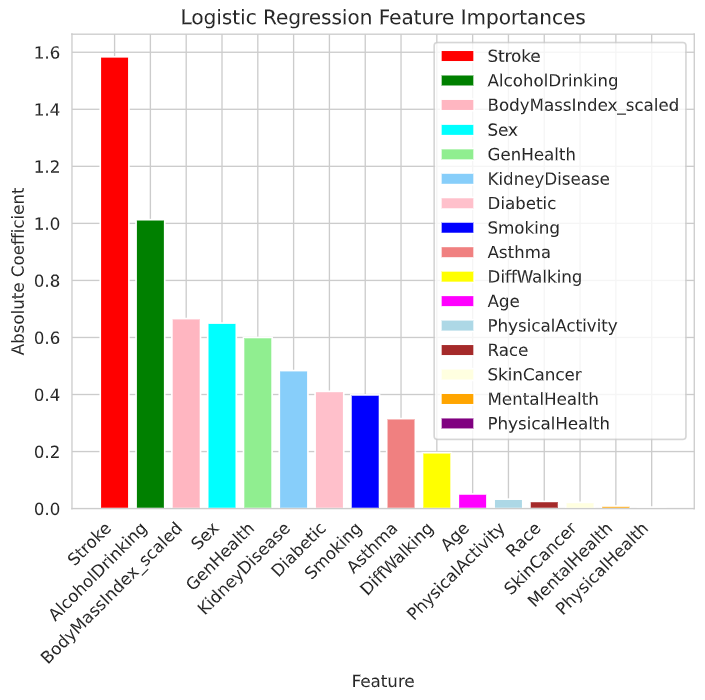
Overall, the model performs reasonably well, with strong recall but lower precision, suggesting potential for enhancement in identifying individuals at risk of heart disease. Further optimization and analysis of the model's errors could help enhance its performance in identifying individuals at risk of heart disease.



The confusion matrix can be used to assess how well a logistic regression model is performing at classifying instances. In this case, the model seems to be performing well, as the majority of the instances are classified correctly



The trade-off between the true positive rate (TPR) and false positive rate (FPR) will be displayed via the ROC curve.The area under the curve (AUC) of 0.75 indicates fair performance.



The above Feature Importance bar chart will display the relative weights assigned to each feature used in the Logistic Regression model. The model's predictions will be more influenced by features with higher significance values, whereas features with lower values will have less of an impact. From the above graph we can see that ‘Stroke’, ‘Alcohol Drinking’ and ‘BMI’ features have a larger effect on the model compared to others.

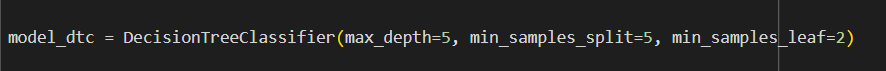
**MODEL II: DECISION TREE CLASSIFIER**

Decision trees are a popular and widely used supervised machine learning algorithm, suitable for both classification and regression problems. This flexibility makes them a strong candidate for the heart disease prediction task.

1. **Reasons for choosing the Decision Tree model: -**
2. Decision trees are non-parametric and do not make any assumptions about the underlying data distribution, making them robust to outliers and noise. This is particularly useful when working with real-world healthcare data.
3. Decision trees have built-in feature selection capabilities, automatically identifying the most important predictors. This can help uncover the key risk factors driving heart disease in the dataset.
4. While decision trees can be prone to overfitting, this can be addressed through techniques like pruning and ensemble methods like Random Forests. These techniques can help improve the model's generalization performance.

Overall, the decision tree classifier is a strong choice for this heart disease prediction project, as it offers interpretability, flexibility, and robustness - key requirements for developing effective and actionable predictive models in the healthcare domain.

1. **Preventing Overfitting using Parameters**



The DecisionTreeClassifier model is instantiated with specific parameters to control its complexity and prevent overfitting. By setting `max\_depth=5`, we limit the depth of the decision tree to five levels, which helps ensure the model generalizes well to unseen data. Additionally, `min\_samples\_split=5` specifies that a node must have at least five samples to be split further, reducing the risk of creating overly complex decision boundaries. With `min\_samples\_leaf=2`, we enforce a minimum number of samples required at a leaf node, promoting better generalization and preventing the model from being overly specific to the training data. Together, these parameter choices aim to strike a balance between model complexity and generalization performance, optimizing the DecisionTreeClassifier for effective classification tasks on the given dataset.

1. **K-fold Cross Validation**

When working on decision trees, we noticed a high variance between accuracy and F1 score. This indicates that your model performs inconsistently or unstable across various data subsets. By splitting the dataset up into several subsets (folds) and iteratively training and testing the model on various combinations of these subsets, cross-validation aids in the acquisition of a more stable estimate of your model's performance. It is imperative that you use 5-fold cross-validation in your Decision Tree model, particularly if you see a substantial difference between accuracy and F1 score.

By dividing the dataset into multiple folds and iteratively training and testing the model on various fold combinations, cross-validation provides a more stable estimate of performance and reduces the risk of overfitting inherent in Decision Trees. This approach not only enhances reliability but also aids in assessing generalization ability and facilitates hyperparameter tuning. By evaluating different parameter combinations, we can optimize the model's performance and ensure its robustness. Overall, integrating 5-fold cross-validation enhances the model's reliability and effectiveness, particularly when encountering significant variability between accuracy and F1 score.

1. **Evaluation Metrics: -**

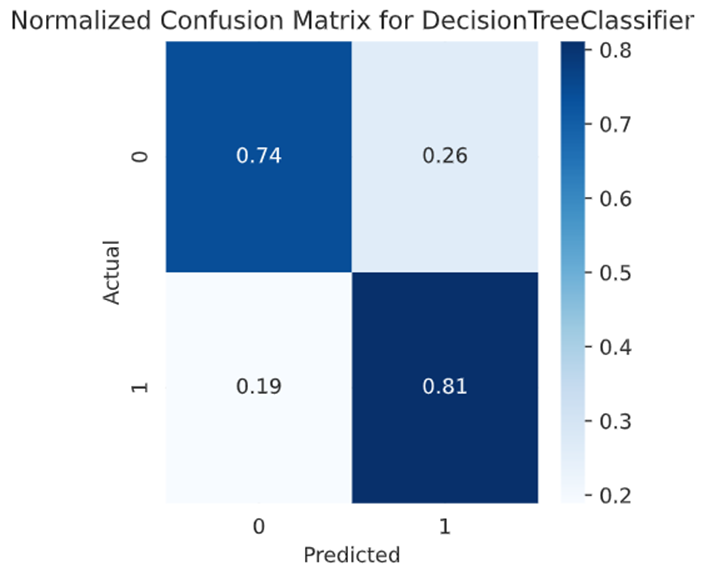
These performance metrics indicate how well the Decision Tree Classifier model is able to predict the target variable based on the given credit record and application record data.

* Accuracy Score: 0.78998
* F1 Score: 0.796
* Precision Score: 0.773
* Recall Score: 0.82

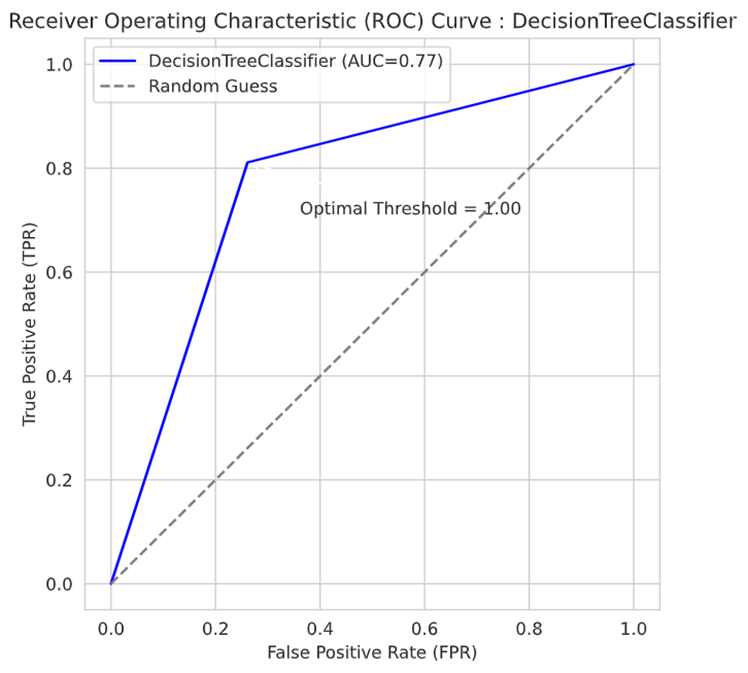
The Decision Tree Classifier model achieves an accuracy, F1 score, precision, and recall of 0.78, 0.796, 0.773, and 0.82 respectively. Its high recall (0.82) indicates effective identification of positive cases (heart disease) with low false negatives. However, the precision (0.77) suggests a notable rate of false positives, resulting in a higher false positive rate compared to false negatives.

Despite this, the F1 score of 0.796 demonstrates good overall performance, balancing precision and recall. With an accuracy of 78.9%, the model shows reasonable classification ability but leaves room for enhancement.

Overall, while the model excels in recall, further optimization is needed to improve precision without sacrificing recall.



In this case also, the model seems to be performing well, as the majority of the instances are classified correctly comparing the actual and predicted value.



The trade-off between the true positive rate (TPR) and false positive rate (FPR) will be displayed via the ROC curve.The area under the curve (AUC) of 0.77 indicates fair performance.

**MODEL III: RANDOM FOREST CLASSIFIER**

The Random Forest Classifier is a robust and versatile ensemble learning algorithm widely used for classification tasks in machine learning. It operates by constructing a multitude of decision trees during training, each trained on a random subset of the training data and features. This randomness introduces diversity among the trees, making the model less prone to overfitting and more robust to noise and outliers in the data. One of the key advantages of Random Forests is their ability to achieve high accuracy by aggregating the predictions of multiple trees through majority voting. This ensemble approach not only enhances predictive performance but also provides insights into feature importance, aiding in feature selection and understanding the underlying data patterns.

During training, Random Forests utilize hyperparameters such as the number of trees (n\_estimators), maximum depth of trees (max\_depth), and minimum samples required for node splitting (min\_samples\_split). These hyperparameters allow fine-tuning of the model's complexity and generalization capabilities, making Random Forests adaptable to different types of datasets and scenarios. Additionally, the Random Forest algorithm is parallelizable, making it suitable for handling large datasets efficiently.

1. **Reasons for Selecting Random Forest Classifier**
2. High Accuracy: Random Forests typically provide high accuracy in predicting outcomes. By aggregating predictions from multiple decision trees, the model can reduce variance and overfitting, leading to better generalization on unseen data.
3. Robustness to Overfitting: Random Forests are less susceptible to overfitting compared to individual decision trees. The random sampling of data and features at each tree node helps create diverse trees that collectively improve model robustness.
4. Handles Nonlinearity: Random Forests can capture nonlinear relationships between features and target variables effectively. This makes them suitable for datasets with complex interactions and patterns.
5. Feature Importance: Random Forests provide a measure of feature importance, indicating which features contribute most to the model's predictions. This information is valuable for feature selection, dimensionality reduction, and gaining insights into the data.

**II. EVALUATION METRICS**

Accuracy Score (0.92451):

* 1. The Accuracy Score measures the overall correctness of the model's predictions.
  2. In this case, an accuracy of approximately 0.92451 means that the model correctly predicted the class for about 92.45% of the samples in the test set.

F1 Score (0.923):

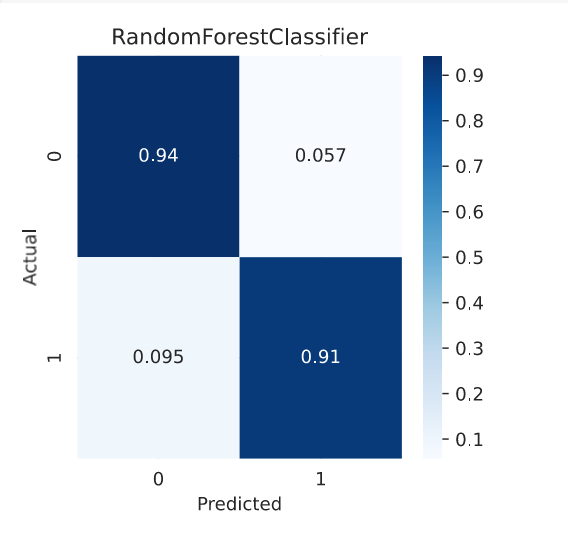
* 1. The F1 Score is the harmonic mean of precision and recall, providing a balance between false positives and false negatives.
  2. A high F1 Score, around 0.923 in this case, indicates a good balance between precision and recall, which is important for binary classification tasks where the classes are imbalanced or have different costs associated with false positives and false negatives.

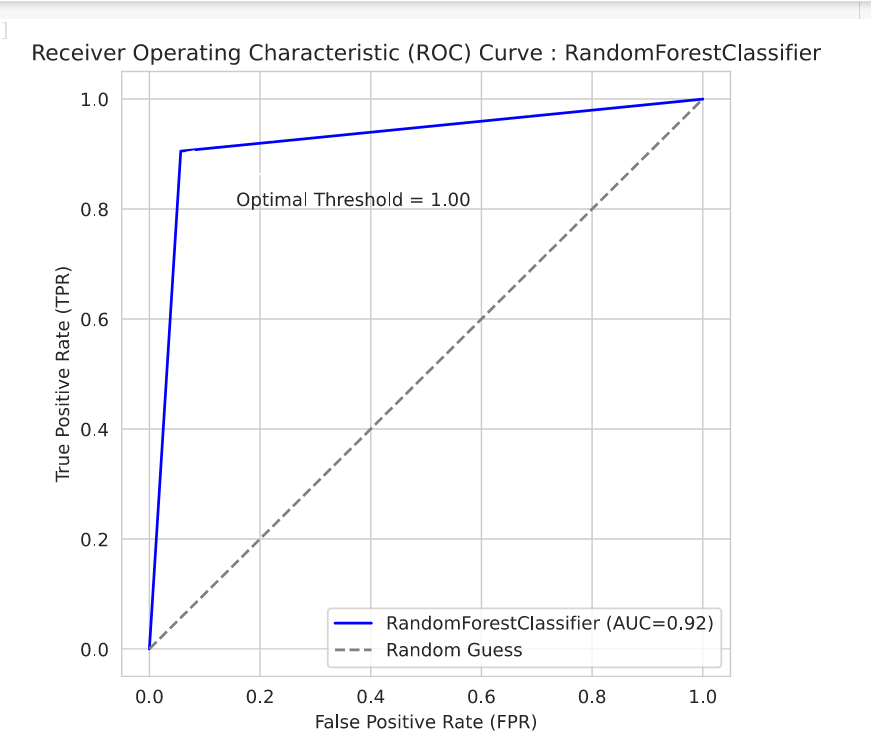
Precision Score (0.941):

* 1. Precision measures the proportion of true positive predictions out of all positive predictions made by the model.
  2. A high Precision Score of approximately 0.941 indicates that out of all the samples predicted as positive by the model, about 94.1% are actually positive. In other words, the model has a low false positive rate.

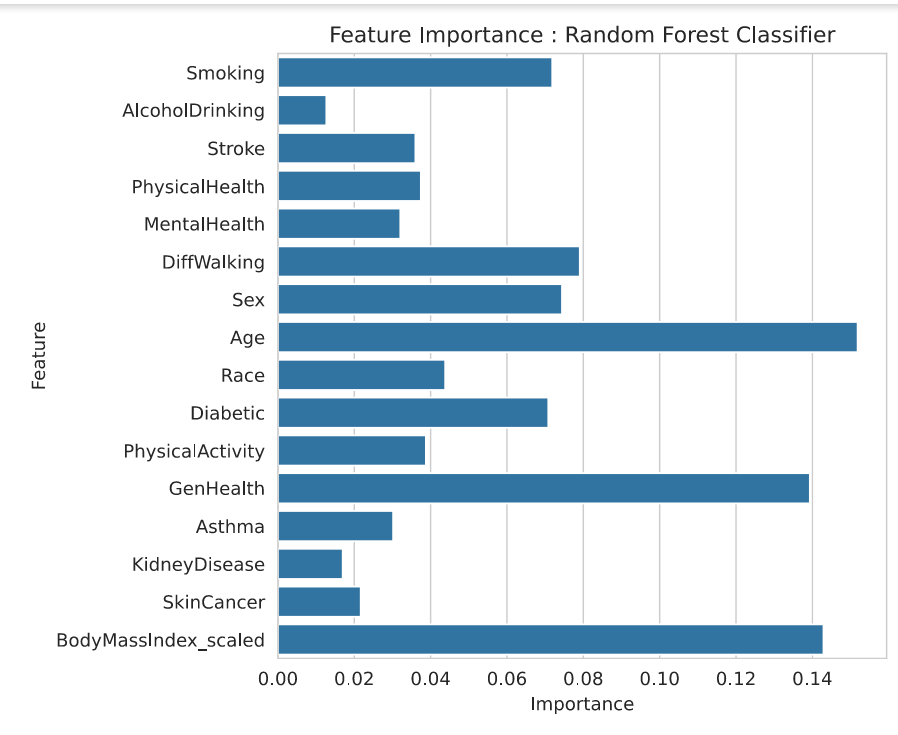
Recall Score (0.905):

* 1. Recall, also known as sensitivity or true positive rate, measures the proportion of actual positives that were correctly predicted by the model.
  2. A Recall Score of approximately 0.905 means that the model correctly identified about 90.5% of the actual positive samples. In other words, the model has a low false negative rate.





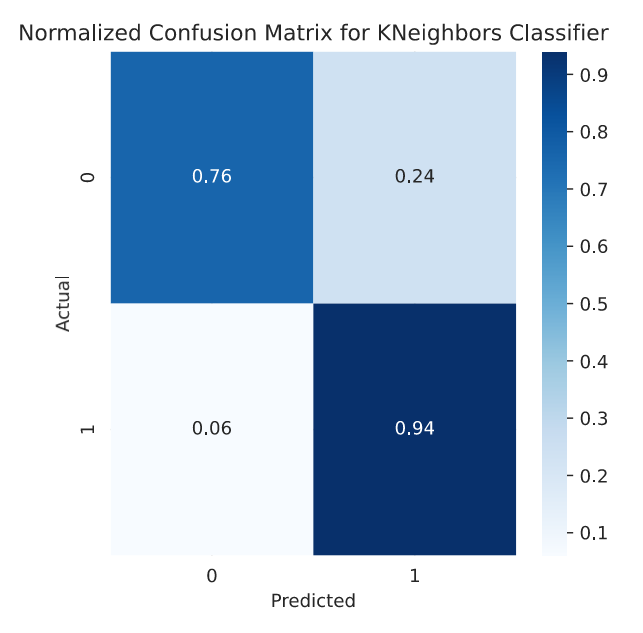
We can see that age plays as an important feature in the feature importance

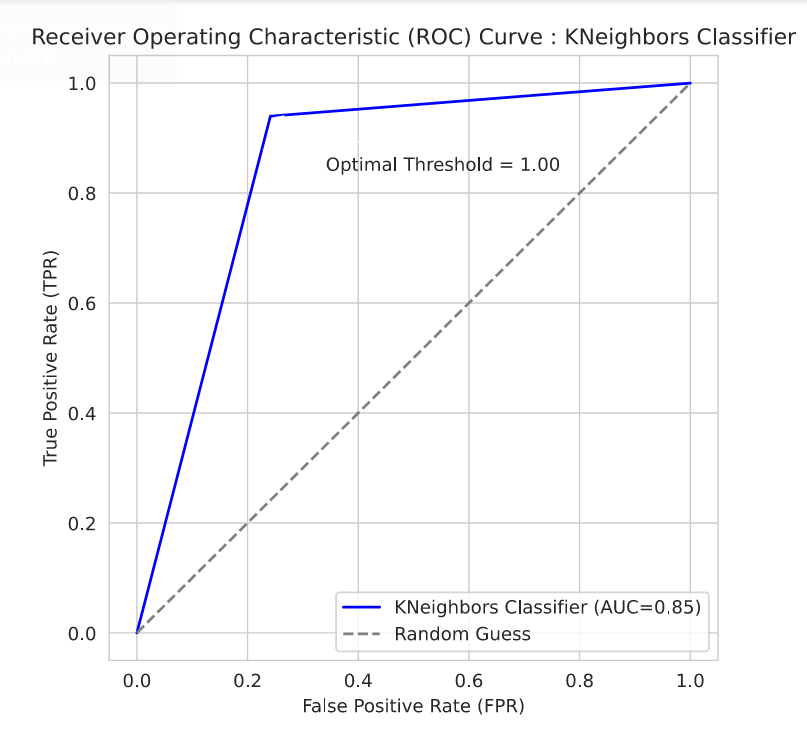


**MODEL IV: KNN CLASSIFIER**

K-Nearest Neighbors (KNN) Classifier is a simple yet effective supervised machine learning algorithm used for both classification and regression tasks. It's a type of instance-based learning, where the model memorizes the entire training dataset and makes predictions based on the similarity between new data points and the existing data points in the training set.

1. **Reasons for Selecting Random Forest Classifier**
2. **Simple Implementation**: KNN is relatively easy to understand and implement compared to some other classification algorithms. It doesn't require assumptions about the underlying data distribution, making it a good starting point for beginners in machine learning.
3. **Instance-Based Learning**: KNN is an instance-based learning algorithm, which means it memorizes the training data and makes predictions based on the similarity of new data points to the existing data. This can be advantageous when the decision boundaries are not linear or when the data exhibits complex patterns that are hard to capture with parametric models.
4. **Flexibility in Feature Space**: KNN can handle both numerical and categorical data without the need for extensive preprocessing. It's robust to noisy data and outliers and can work well in high-dimensional feature spaces.
5. **Interpretability**: KNN provides transparent and interpretable results. Predictions are based on the nearest neighbors, which can be visualized and understood, making it easier to explain to stakeholders or domain experts.
6. **EVALUATION METRICS**
7. Accuracy Score (0.849):
   1. The Accuracy Score measures the overall correctness of the model's predictions.
   2. An accuracy of approximately 0.849 means that the KNN Classifier correctly predicted the class for about 84.9% of the samples in the test set.
   3. While accuracy is an important metric, it's essential to consider other scores, especially in the presence of imbalanced classes.
8. F1 Score (0.862):
   1. The F1 Score is the harmonic mean of precision and recall, providing a balance between false positives and false negatives.
   2. A high F1 Score of approximately 0.862 indicates a good balance between precision and recall, which is crucial in binary classification tasks with imbalanced classes.
   3. It suggests that the KNN Classifier is effective in correctly identifying both positive and negative instances while minimizing false positives and false negatives.
9. Precision Score (0.796):
   1. Precision measures the proportion of true positive predictions out of all positive predictions made by the model.
   2. A Precision Score of approximately 0.796 means that out of all the samples predicted as positive by the KNN Classifier, about 79.6% are actually positive.
   3. A higher precision score indicates a lower false positive rate, which is desirable in scenarios where false positives are costly or undesirable.
10. Recall Score (0.940):
    1. Recall, also known as sensitivity or true positive rate, measures the proportion of actual positives that were correctly predicted by the model.
    2. A Recall Score of approximately 0.940 means that the KNN Classifier correctly identified about 94.0% of the actual positive samples.





**MODEL V: AdaBoost CLASSIFIER**

AdaBoost is designed to improve the accuracy of binary classification models by combining the predictions of multiple "weak" classifiers, such as decision trees. This can lead to a more robust and accurate heart disease prediction model.

1. **Reasons for Selecting Random Forest Classifier**
2. Handling Weak Learners: AdaBoost is well-suited for working with weak learners, such as shallow decision trees, that may individually have relatively poor performance. By iteratively training and combining these weak models, AdaBoost can create a strong overall classifier.
3. Adaptability: AdaBoost is an "adaptive" boosting algorithm, meaning it can adjust the weights of the training examples at each iteration to focus on the instances that were previously misclassified. This makes it effective at handling complex, real-world datasets like the CDC heart disease data.
4. Interpretability: When using decision trees as the base learners, the AdaBoost model maintains the interpretability of the individual decision trees, which aligns with the project's goal of providing healthcare professionals with transparent and explainable predictive models.
5. Robustness to Overfitting: AdaBoost is generally less prone to overfitting compared to other machine learning algorithms, making it a reliable choice for this heart disease prediction task.
6. Ease of Implementation: AdaBoost is relatively simple to implement and use compared to more complex ensemble methods, which is an advantage when working with large, real-world datasets.

By leveraging the strengths of AdaBoost, such as its ability to improve accuracy, handle weak learners, and maintain interpretability, the project can develop a highly effective and transparent heart disease prediction model to support healthcare professionals in early detection and intervention.

1. **EVALUATION METRICS**

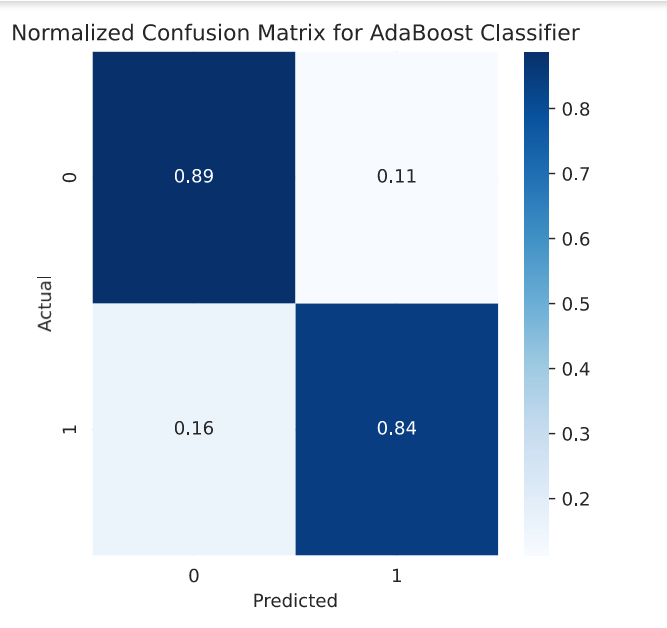
These performance metrics indicate how well the AdaBoost Classifier model is able to predict the target variable based on the given credit record and application record data.

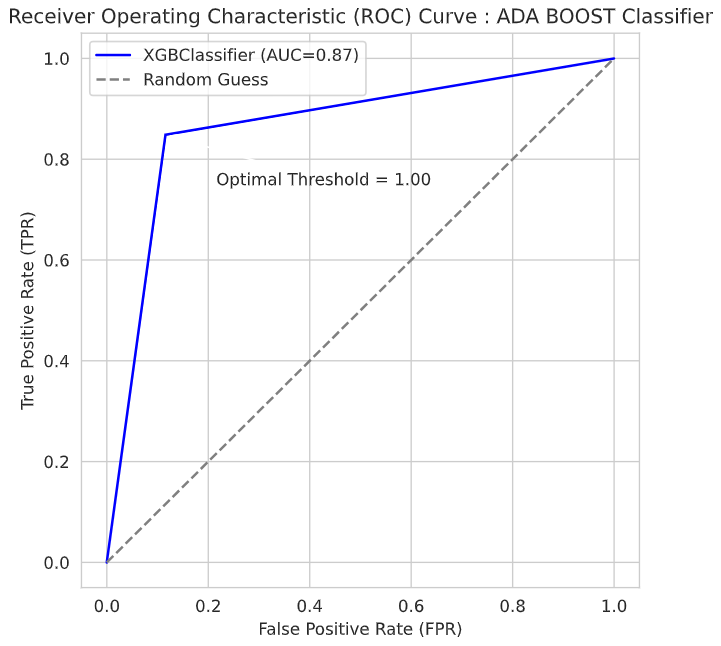
* Accuracy Score: 0.86
* F1 Score: 0.864
* Precision Score: 0.88
* Recall Score: 0.849

The AdaBoost Classifier model achieves an accuracy, F1 score, precision, and recall of 0.86, 0.864, 0.88, and 0.849 respectively. Its high recall (0.849) indicates effective identification of positive cases (heart disease) with low false negatives. However, the precision (0.849) suggests a notable rate of false positives, resulting in a higher false positive rate compared to false negatives.

Despite this, the F1 score of 0.864 demonstrates good overall performance, balancing precision and recall. With an accuracy of 86%, the model shows excellent classification ability but leaves room for enhancement.

Overall, while the model excels in recall and accuracy, further optimization is needed to improve precision without sacrificing recall.





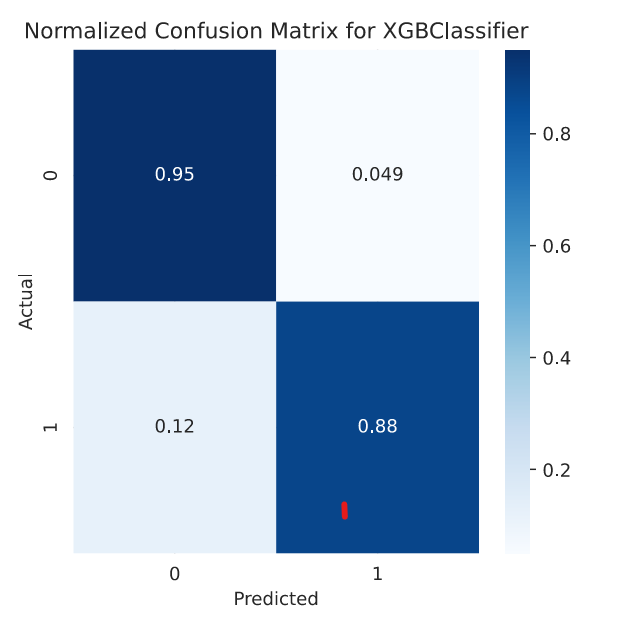
**MODEL VI: XGB CLASSIFIER**

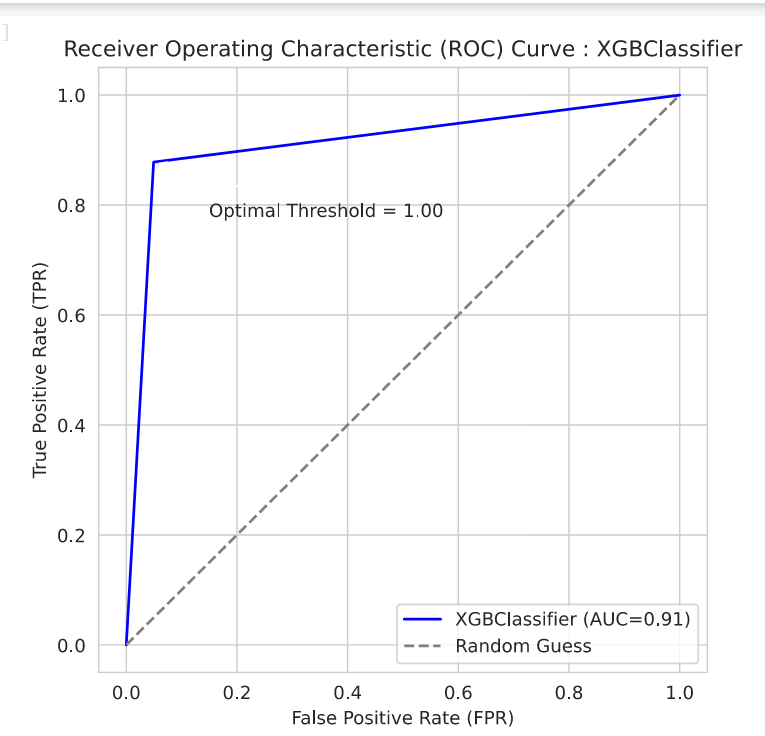
XGBoost (eXtreme Gradient Boosting) is a powerful and popular machine learning algorithm that belongs to the boosting family of algorithms. It is known for its efficiency, scalability, and high performance in a wide range of machine learning tasks, particularly in structured/tabular data settings.

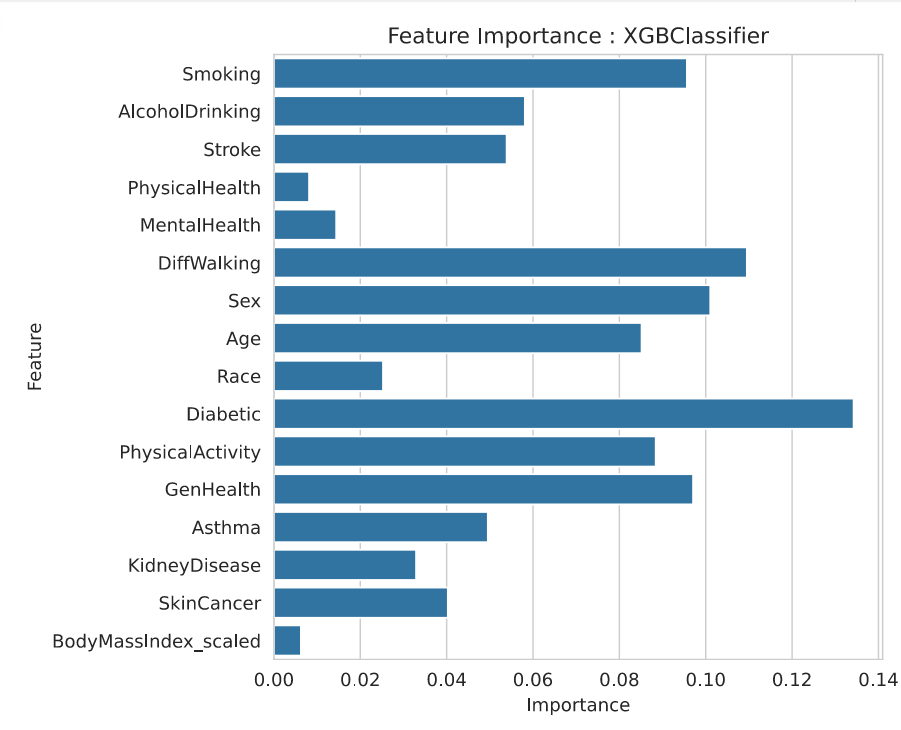
1. **Reasons for Selecting Random Forest Classifier**
2. **High Performance**:XGBoost is known for its high performance and accuracy in a wide range of machine learning tasks.
3. **Efficiency**:XGBoost is highly optimized for speed and memory usage, making it efficient for training on large datasets with many features.
4. **Robustness to Overfitting**: XGBoost includes regularization techniques such as L1 and L2 regularization (Ridge and Lasso) to prevent overfitting.
5. **Feature Importance Analysis**: XGBoost provides feature importance scores based on the average gain or coverage attributed to each feature across all trees in the ensemble.

**EVALUATION METRICS**

1. Accuracy Score (0.91438):
   1. The Accuracy Score measures the overall correctness of the model's predictions.
   2. An accuracy of approximately 0.91438 means that the model correctly predicted the class for about 91.44% of the samples in the test set.
   3. Accuracy is a useful metric but may not provide a complete picture, especially if the classes are imbalanced.
2. F1 Score (0.911):
   1. The F1 Score is the harmonic mean of precision and recall, providing a balance between false positives and false negatives.
   2. A high F1 Score of approximately 0.911 indicates a good balance between precision and recall, which is crucial in binary classification tasks with imbalanced classes or uneven costs of false positives and false negatives.
3. Precision Score (0.947):
   1. Precision measures the proportion of true positive predictions out of all positive predictions made by the model.
   2. A Precision Score of approximately 0.947 means that out of all the samples predicted as positive by the model, about 94.7% are actually positive.
   3. A higher precision score indicates a lower false positive rate, which is desirable when minimizing false alarms or incorrect positive predictions is important.
4. Recall Score (0.878):
   1. Recall, also known as sensitivity or true positive rate, measures the proportion of actual positives that were correctly predicted by the model.
   2. A Recall Score of approximately 0.878 means that the model correctly identified about 87.8% of the actual positive samples.
   3. A higher recall score indicates a lower false negative rate, which is important when capturing all positive instances is critical and false negatives are costly.

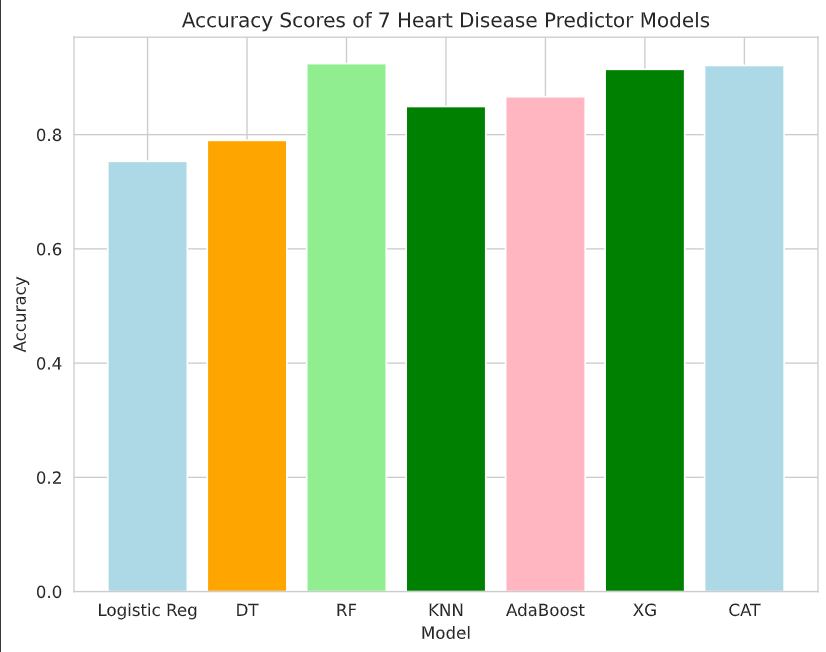






Diabetic plays an important feature.

**COMPARISON BETWEEN MODELS**

****

**As we can see Random forest has the highest accuracy and is performing well when compared to the other models.**