CardioViglant: Cardiovascular Decompensation Forecasting

PROJECT PHASE #2

Aishwarya Chand, Prajakta Jhade, Shreya Thakur

PROJECT OVERVIEW

*One person dies every **33 seconds** in the United States from cardiovascular disease. About 695,000 people in the United States died from heart disease (according to the data collected in 2021)—that's 1 in every 5 deaths. Globally, the scale of mortality due to this disease is even more surprising with a record of 20.5 million.*

Cardiovascular diseases are one of the major healthcare concerns and leading causes of mortality globally. Early intervention plays a crucial role here and enables healthcare providers to tailor treatment plans and adjust medications accordingly. Despite advancement in cardiovascular care, predicting and preventing cardiovascular decompensation remains a significant challenge.

Our Web Application: CardioVigilant aims to transform cardiovascular healthcare, equipping healthcare providers with an unparalleled tool for precision forecasting and redefine standards in patient care, ultimately saving lives and enhancing the quality of cardiovascular health globally.

PHASE-1 OVERVIEW

In the project phase one, we have performed data cleaning and exploratory data analysis (EDA) to prepare the data for the development of a Cardiovascular Decompensation Forecasting model. Initially, null values were removed from critical columns, including Sex, RestingBP, Cholesterol, MaxHR, and heart disease, to maintain the integrity of the data.

Categorical columns are expected as binary but containing more than the counted on the categories were corrected. Outliers on Age, especially the records with ages less than 20, were detected as deviations and removed. Gender variance was maintained by removing records with null values and correcting inconsistencies in naming. ChestPainType and ExerciseAngina columns were cleaned for consistency, also by performing renaming and grouping values. RestingBP and MaxHR Columns, we have imputed null values with mean to maintain dataset completeness. Label encoding was applied to convert categorical data into numerical which is suitable for modelling.

During EDA, the dataset disclosed insights as: males has higher incidence of heart disease as compared to females; some types of chest pain, like Asymptomatic (ASY), were more closely prone to heart disease and other factors like RestingECG and FastingBS showed higher susceptibility to get the heart issues. In Addition, a correlation matrix was used to identify the trends between various features and the targets.

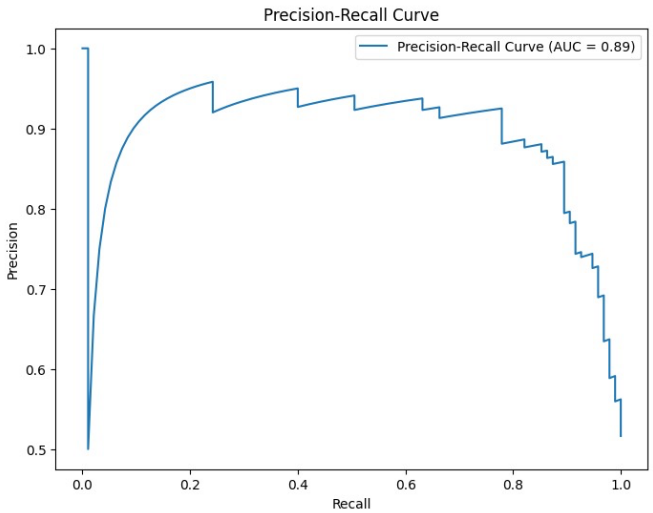
MODELS:

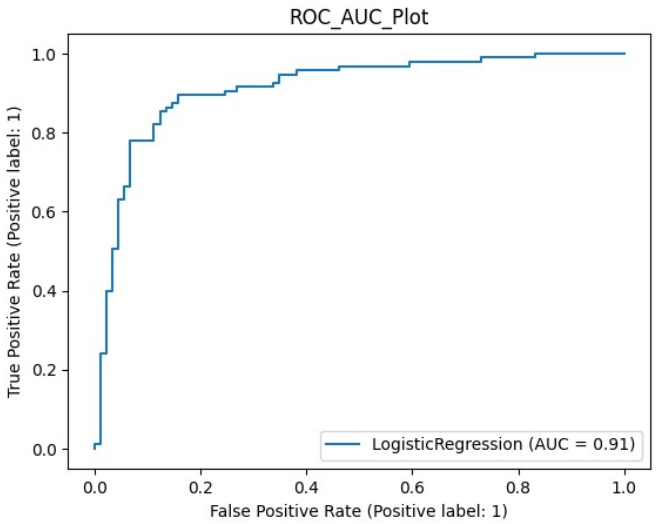
1. LOGISTIC REGRESSION:

EVALUATION MATRICS:

```
Accuracy :  86.96%
Cross Validation Score :  91.14%
ROC_AUC Score :  86.87%
```

	precision	recall	f1-score	support
0	0.88	0.84	0.86	89
1	0.86	0.89	0.88	95
accuracy			0.87	184
macro avg	0.87	0.87	0.87	184
weighted avg	0.87	0.87	0.87	184





1. **Why to choose the Logistic Regression Algorithm:**
   Logistic regression is used for binary classification such as whether the event will occur or not. This probabilistic interpretation of prediction is most suitable to identify if the person is susceptible to any heart disease or not.
   Efficiency – Healthcare system has abundant set of data and Logistic regression is suitable as handles large set of data effectively.
   In heart disease prediction, it is important to understand the factors contributing to the prediction, like age, cholesterol level, blood pressure, etc.
   Logistic regression measures the coefficient involved with each feature to specify which among them most strongly effect the prediction result.
   Moreover, the model handles both categorical and continuous data and it is less prone to overfitting as compared to other models.

2. **Work performed to tune or train the model:**
   For the Logistic Regression model, we have used scikit-learn library to train and test the model and used Confusion metrics, accuracy precision score to show the model accuracy value. The model's efficiency is represented by the library – RocCurveDisplay.
   The model is validated using RepeatedStartifiedFolds which ensures each fold contains equal set of data to eliminate biasness in the result.
   We used tried with multiple parameters for tuning and optimizing the accuracy. Out of them the most relevant parameters used in the model are -

3. **Effectiveness of the algorithm when applied to the data:**
   Handles large set of data - The model effectively works on – number of data.
   Effectiveness of the algorithm is evaluated based on several metrics parameters like accuracy score, precision, recall and confusion matrix.

4. **Relevant metrics used for demonstrating the model effectiveness:**
   Effectiveness of model is evaluated by the accuracy score, precision, recall and confusion matrix.
   For the model, Accuracy came out to be 87.50% while the cross-validation score is 91.12%
   ROC_AUC Score : 87.43%
   For the model, the analysis is as follows:
   1. Precision(class 0): 88%
      A higher precision indicates that the model effectively minimizes any false alarm for patients not having heart issues. This helps avoid unnecessary treatments and interventions in the medical field.
      Precision(Class 1): 87%
      A higher precision value indicates that if the model predicts a patient has heart disease, 87% of the time it is true. This is crucial for patients to receive early diagnosis and treatment.
   2. Recall(class 0): 85%
      This signifies that among 100 individuals without heart disease, the model determines 85 individuals. Therefore, the recall effectively identifies the truly healthy people not to be indicated as unhealthy.

3. Recall(Class 1): 89%
   A high recall value signifies the truly unhealthy patients which is essential to minimize the chance of missing out the potential heart patients.
4. Accuracy: 87.50%
   A high accuracy signifies that the model effectively identifies individuals with heart disease or not.
5. Cross Validation: 91.12%
   Cross Validation score signifies the effectiveness of the model on different training and testing data sets. Therefore, higher score of 91.12% indicates that the model is likely to generate correct results for newer unknown data, thereby providing effective and accurate prediction of heart disease.

5. Intelligence gained from the algorithm about the model:

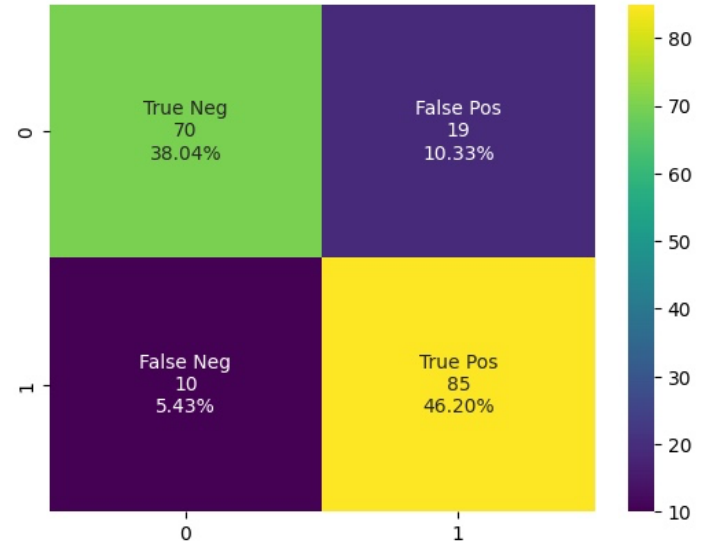
The model provides several insights about the model's performance, its capability and strengths. Here, the model is divided into 80-20 train-test data. The ROC-AUC score of 87.43% signifies that the model effectively able to distinguish between individuals susceptible to heart disease or not. The cross-validation value indicates the accuracy of the prediction for the newer unseen data. While the accuracy of 87.50% clearly shows the model's effectiveness towards the disease prediction.

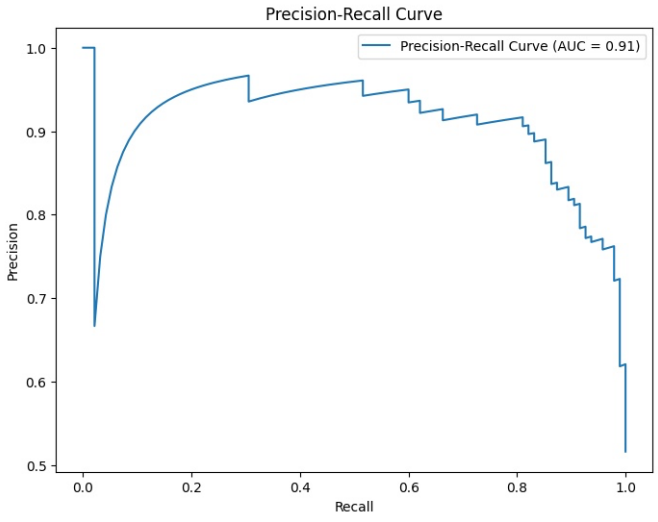
2. RANDOM FOREST:

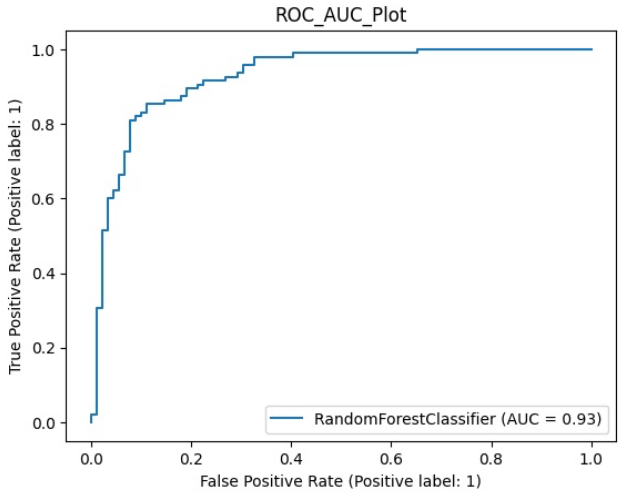
EVALUATION MATRICS:

```
Accuracy :  84.24%
Cross Validation Score :  93.19%
ROC_AUC Score :  84.06%
```

	precision	recall	f1-score	support
0	0.88	0.79	0.83	89
1	0.82	0.89	0.85	95
accuracy			0.84	184
macro avg	0.85	0.84	0.84	184
weighted avg	0.85	0.84	0.84	184







1. **Why to choose the Random Forest Algorithm:**
   Random Forest is an algorithm that is well suited for classification problems, making it a suitable choice for predicting Cardiovascular Decompensation. We have chosen Random Forest Classifier for its Versatility, as the dataset contains a mix of numerical and categorical data Random Forest can handle such heterogeneity in data.
   Random forest algorithms can capture enigmatic relationships between features and target variables particularly in cardiovascular disease dataset where interaction between features can be complex and not well understood.
   The cross validation shows that our model can work with new data it has not worked on before. This is important with medical aspects, because we have to be sure this model can make correct predictions for new patients' data.

2. **Work performed to tune or train the model:**
   Random Forest classifier is initialized with “max_depth = 4”, which always limits the depth of each tree in the forest. This parameter handles the complexities of the trees and prevents overfitting. In addition, “random_state = 0” which ensures reliability of results.

3. **Effectiveness of the algorithm when applied to the data:**
   The model's performance is evaluated using evaluation metrics with Accuracy, Recall, cross-validation score, ROC-AUC score, Precision, F1-score and Support.

4. **Relevant metrics used for demonstrating the model effectiveness:**
   The effectiveness of the algorithm when applied to the cardiovascular disease dataset are summarized by the below metrics:

   An accuracy as 84.24% indicates the model correctly predicts cardiovascular decomposition status for 84.24% of the samples in the test set.

   Cross-validation is used to calculate a model's performance on new data. The cross-validation score is 92.91% which indicates the model works well with new data and is not overfitting the training dataset.

   An ROC-AUC Score (Receiver Operating Characteristic) of 84.06% indicates that the model performs well in differentiating between patients with heart disease and without heart disease.

   The precision of 88% for normal class (0) and 82% for heart disease class (1), indicating correct predictions within each class. The recall is 79% for normal class (0) and 89% for heart disease class (1). The F1-score is 83% for class 0 and 85% for class 1, representing the balance between precision and recall.

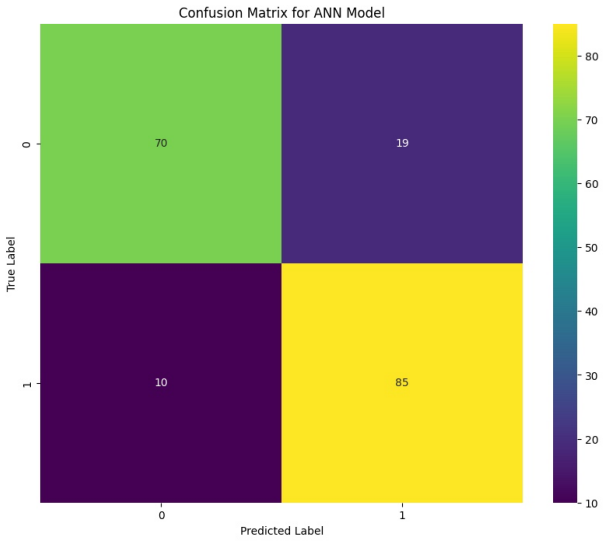
5. **Intelligence gained from the algorithm about the model:**
   The ROC-AUC plot gives a visual representation between true positive and false positive, allowing for an intuitive understanding of the model performance across different thresholds.

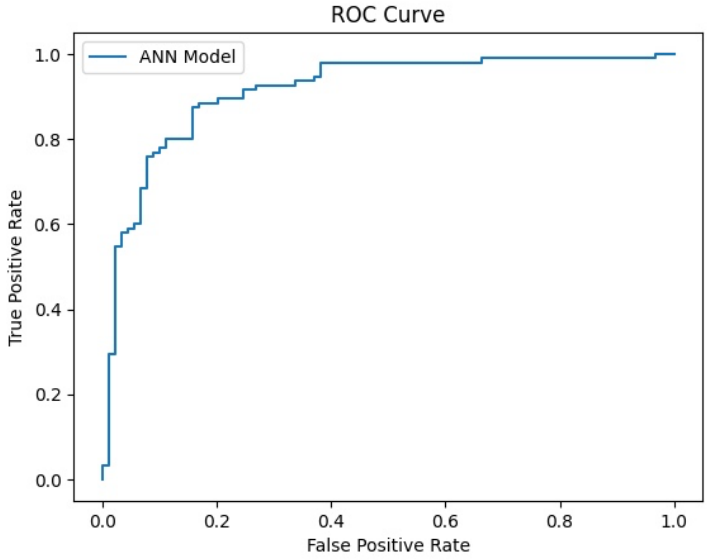
3. ARTIFICIAL NEURAL NETWORK (ANN):

EVALUATION MATRICS:

```
Epoch 98/100
23/23 [==============================] - 0s 4ms/step - loss: 0.2493 - accuracy: 0.8951 - val_loss: 0.3925 - val_accuracy: 0.8533
Epoch 99/100
23/23 [==============================] - 0s 3ms/step - loss: 0.2451 - accuracy: 0.9087 - val_loss: 0.3911 - val_accuracy: 0.8641
Epoch 100/100
23/23 [==============================] - 0s 4ms/step - loss: 0.2383 - accuracy: 0.9060 - val_loss: 0.3924 - val_accuracy: 0.8533
<keras.src.callbacks.History at 0x7bc7e2dc6f80>
```

	precision	recall	f1-score	support
0	0.88	0.79	0.83	89
1	0.82	0.89	0.85	95
accuracy			0.84	184
macro avg	0.85	0.84	0.84	184
weighted avg	0.85	0.84	0.84	184





1. **Why to choose the Artificial Neural Network (ANN) Algorithm:**
   Artificial neural network algorithm, a deep learning model considering dense layers (sequential architecture) with dropouts regularization was driven by several factors.
   The algorithm has the ability to learn hierarchical representations of features from raw data, which is very important when dealing with medical datasets.
   Deep learning models, particularly neural networks, are well-suited for capturing complex, non-linear relationships in high-dimensional data.

2. **Work performed to tune or train the model:**
   To train the model, we have taken the below steps:
   The model is defined with three dense layers, each layer using ReLu function to introduce the non-linearity.
   And we have added Dropout layer after the addition of the first two dense layers to prevent overfitting by dropping some fraction of neurons during training.
   The Adam optimizer is used to compile the model and binary cross entropy loss function and accuracy.
   We have considered 100 epochs with a batch size of 32 to perform the training and validation was provided on unseen data. Performance evaluated based on training and validation loss and accuracy.

3. **Effectiveness of the algorithm when applied to the data:**
   The model's performance is evaluated using evaluation metrics with Binary Cross Entropy, Loss function, Accuracy.

4. **Relevant metrics used for demonstrating the model effectiveness:**
   The effectiveness of the algorithm when applied to the cardiovascular disease dataset are summarized by the below metrics:
   The effectiveness of the algorithm when applied to the cardiovascular disease dataset are summarized by the below metrics:
   This model has an accuracy of 90.60% on the training data and 85.33% on the validation data. This indicates that the model can perform well on new patient's data, as the validation accuracy is nearly close to training accuracy.
   Loss value (binary_crossentropy) represents that the model aims to minimize the binary cross entropy loss.

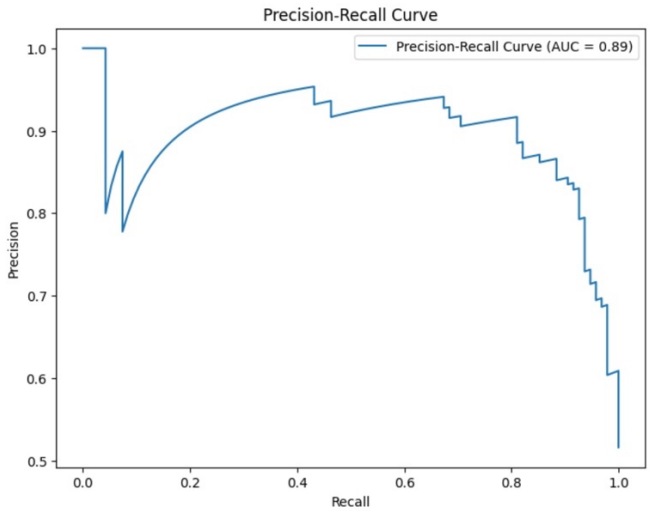
5. **Intelligence gained from the algorithm about the model:**
   The model's accuracy indicates that it has successfully learned patterns from given input features to predict the heart disease. The validation accuracy suggests that the model can generalize well to new patient's data, which is crucial for the practical utility.

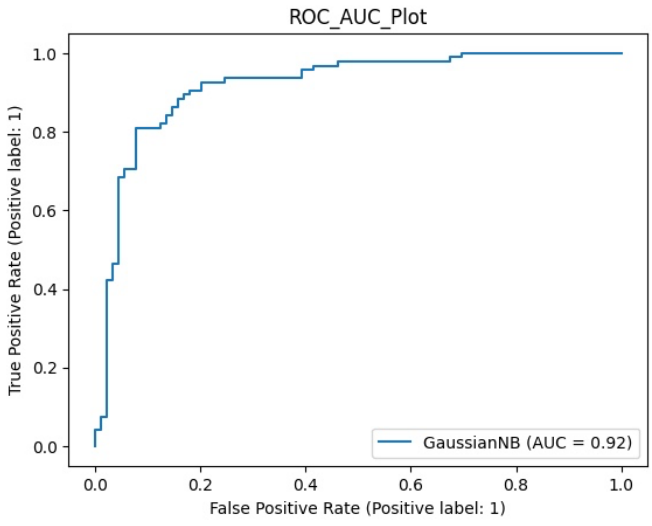
4. NAIVE BAYES:

EVALUATION MATRICS:

```
Accuracy :  86.41%
Cross Validation Score :  91.29%
ROC_AUC Score :  86.27%
```

	precision	recall	f1-score	support
0	0.89	0.82	0.85	89
1	0.84	0.91	0.87	95
accuracy			0.86	184
macro avg	0.87	0.86	0.86	184
weighted avg	0.87	0.86	0.86	184





1. **Why to choose the Naive Bayes Algorithm:**
   We have used the Gaussian Naive Bayes algorithm for this dataset because it's its simplicity, efficiency, and effectiveness in classification.

2. **Work performed to the tune or train the model:**
   As the algorithm Naive Bayes doesn't have so many hyperparameters to tune, the main focus is on preprocessing the data and making sure it fits the model. This involved removing missing values and encoding categorical variables and scaling numerical features. In addition, splitting the data into training and testing for evaluating the model's performance.

3. **Effectiveness of the algorithm when applied to the data:**
   We have achieved an accuracy of 85.87%. However, accuracy alone might not provide a complete picture, so other metrics such as the ROC-AUC score we have used and achieved the score of 85.75%.

4. **Relevant metrics used for demonstrating the model effectiveness:**
   We have used ROC-AUC metrics. This metric measures the area under positive and negative classes. Here, the ROC-AUC score of 85.75% suggests that the model has decent differentiating power.
   The precision of 88% for normal class (0) and 84% for heart disease class (1), indicating correct predictions within each class. The recall is 82% for normal class (0) and 89% for heart disease class (1). The F1-score is 85% for class 0 and 87% for class 1, representing the balance between precision and recall.

5. **Intelligence gained from the algorithm to your data about your model:**
   This model suggests that some features like age, cholesterol and maximum heart rate achieved are very important in predicting heart disease risk.
   It suggests that having specific symptoms like chest pain type, exercise induced angina and ST segment slope during exercise can be related to cardiovascular issues.

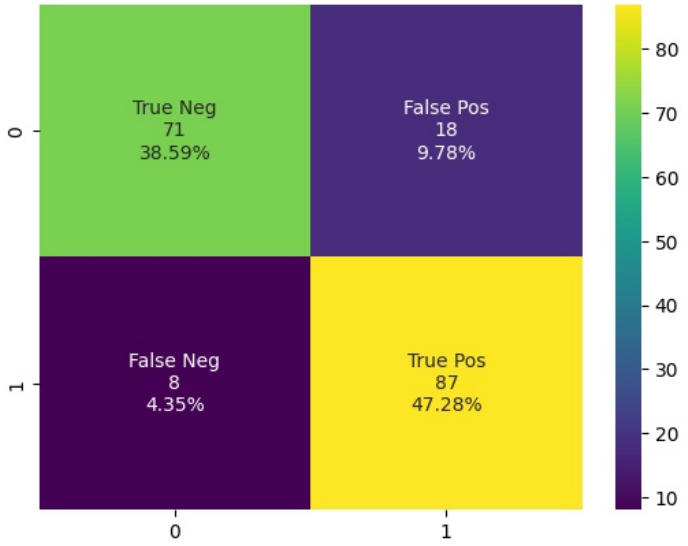
5. K-NEAREST NEIGHBOR:

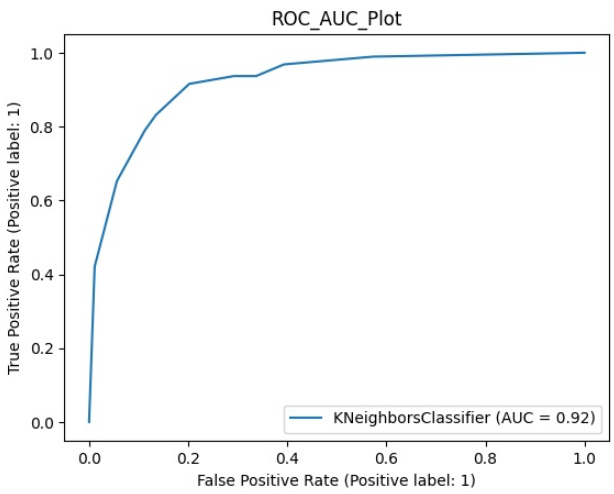
EVALUATION MATRICS:

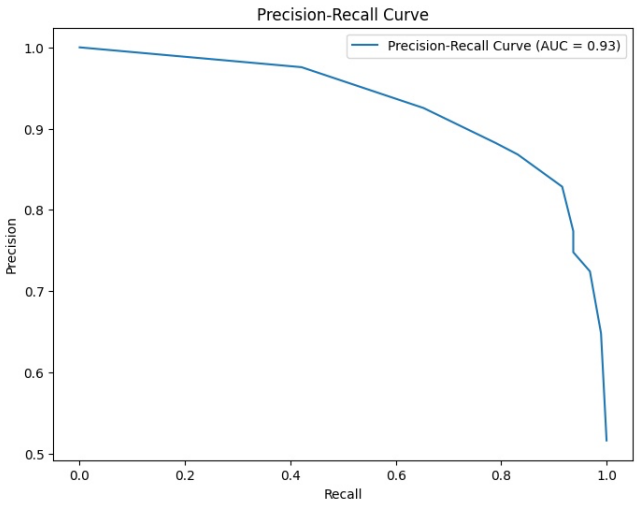
```
Accuracy :  85.87%
Cross Validation Score :  91.48%
ROC_AUC Score :  85.68%
```

	precision	recall	f1-score	support
0	0.90	0.80	0.85	89
1	0.83	0.92	0.87	95
accuracy			0.86	184
macro avg	0.86	0.86	0.86	184
weighted avg	0.86	0.86	0.86	184

	K	Distance	CV Score
13	9	manhattan	0.91
12	9	euclidean	0.91
14	9	minkowski	0.91
9	7	euclidean	0.91
11	7	minkowski	0.91







1. **Why to choose the K-Nearest Neighbor:**
   In our problem we are dealing with medical data of patients which involves nonlinear relationship between the patients characteristics(features) and the presence of heart disease. KNN is particularly capable of capturing non linearities without assuming a specific functional form, making it suitable for this type of problem. Further as KNN doesn't require a traditional training phase, we believe that it can be advantageous when working with medical data that is constantly evolving or being updated. This means the model can quickly adapt to new data without the need for retraining which would bring an edge to our model. Also, we found that some of our data was categorical and some of it numerical, KNN does have an ability to handle mix of such data. This is why we decided to try KNN model for our algorithm.

2. **Work performed to tune or train the model:**
   Repeated k-fold cross-validation provides a way to improve the estimated performance of a machine learning model. This involves simply repeating the cross-validation procedure multiple times and reporting the mean result across all folds from all runs.

   KNN does not require training however to obtain better accuracy we tried to use different parameters to fit the model. We tried to experiment with different number of K nearest neighbors - and different distance metrics. After trying with different K nearest neighbor values and distance metrics we found that K= 9 and distance metric = manhattan to be giving the best results. Further, we used RepeatedStratifiedKFold, which combines stratified sampling and repeated KFold cross-validation, ensuring balanced class representation and reducing variance. It's valuable for robustly evaluating model performance.

3. **Effectiveness of the algorithm when applied to the data:**
   Based on our classification problem and dataset, we found that predicting heart disease is often a case where false negatives (predicting no disease when the patient actually has it) are more critical than false positives (predicting disease when the patient doesn't have it). Therefore, apart from other evaluation metric earlier we found Recall to be a distinguished evaluation metric appropriate in this model. This is because we want to minimize the number of cases where the model fails to identify individuals who have the disease.

4. **Relevant metrics used for demonstrating the model effectiveness:**
   - Accuracy: Our model achieves an accuracy of 85.87%, It indicates that the model correctly predicts heart disease status for 85.87% of the cases.
   - Cross Validation Score: With a cross-validation score of 91.48%, our model demonstrates robustness and generalizability, performing consistently well across different subsets of the data.
   - ROC_AUC Score: The ROC AUC score of 85.68% indicates that our model suggests better discrimination ability.
   - Class 0 (No Heart Disease) has slightly higher precision, but lower recall compared to Class 1 (Heart Disease). This suggests that the model is better at correctly identifying instances without heart disease but may miss some actual cases.

   - Class 1 (Heart Disease) has higher recall, indicating that the model is better at capturing instances of heart disease but may have a slightly lower precision.
   - The F1-scores for both classes are relatively high, indicating a good balance between precision and recall for each class.
   - Further, we have also used a misclassification rate of approximately 0.141 or 14.1% means that about 14.1% of the instances in your dataset were misclassified by the model. Lower misclassification rates indicate better model performance, as they imply fewer incorrect predictions.

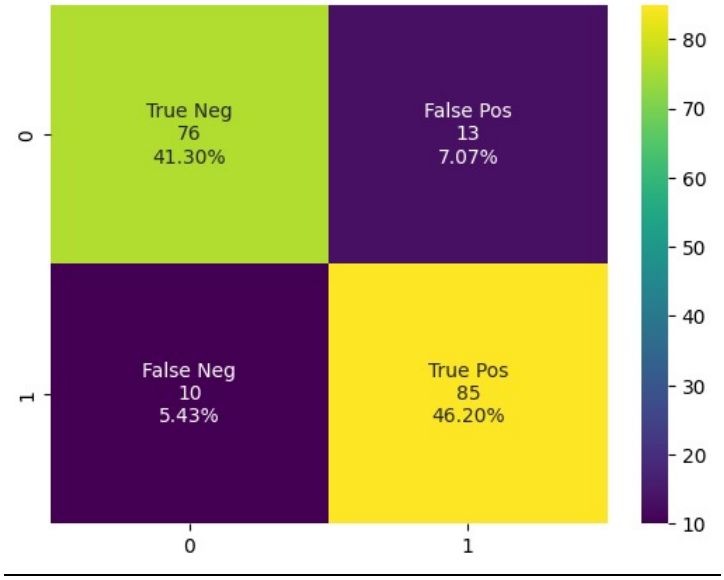
5. **Intelligence gained from the algorithm about the model:**
   In the model, the higher recall value over precision suggests that minimizing false negatives (missed cases) is prioritized, i.e., the patients with any heart disease is not misclassified as healthy. This is crucial for medical diagnosis. Additionally, low misclassification rate suggests good performance of the model.

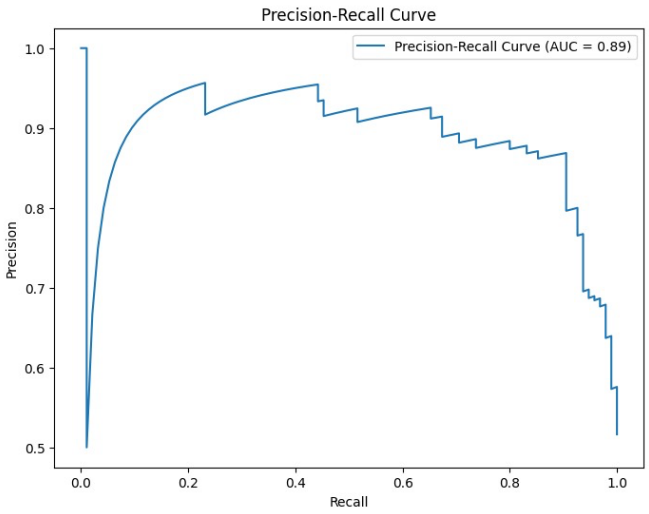
6. SUPPORT VECTOR MACHINE:

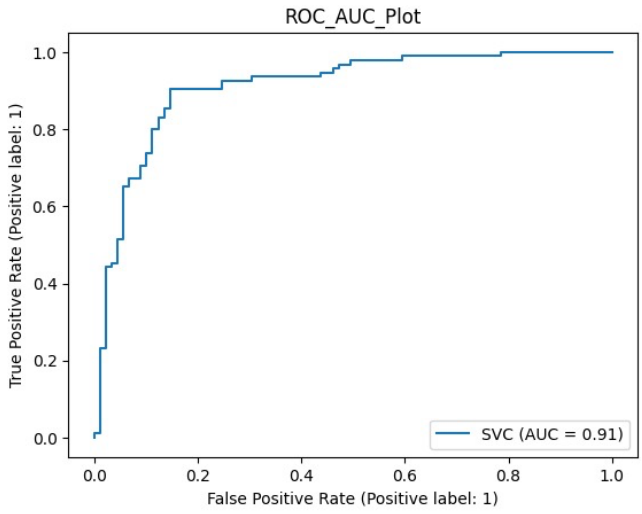
EVALUATION MATRICS:

```
Accuracy :  87.50%
Cross Validation Score :  90.46%
ROC_AUC Score :  87.43%
```

	precision	recall	f1-score	support
0	0.88	0.85	0.87	89
1	0.87	0.89	0.88	95
accuracy			0.88	184
macro avg	0.88	0.87	0.87	184
weighted avg	0.88	0.88	0.87	184







1. **Why to choose the Support Vector Machine Algorithm:**
   Due to their efficacy in high-dimensional spaces, which are typical in medical data, resilience against overfitting with small to medium-sized datasets, and capacity to handle non-linear relationships between risk factors and disease presence, Support Vector Machines (SVM) are well-suited for the prediction of heart disease.
   SVMs are also resilient to extraneous features, hence we believe this makes them even more appropriate for medical datasets where feature importance varies.

2. **Work performed to the tune or train the model:**
   After loading and splitting the data in the previous steps, we defined a parameter grid to specify different hyperparameters to be tuned during the grid search. For the SVM model, we focused on tuning the c parameter, which controls the regularization strength, and the gamma parameter which controls the kernel and finally compared with linear and rbf (radial basis function kernel) we instantiated our SVM model. Using GridSearchCV, we hyperparameters for the SVM classifier on the specified grid, maximizing performance through cross-validation. We fitted the grid search to the training data, training multiple SVM models with different hyperparameter combinations.
   Using mean accuracy as the evaluation metric by default, the top-performing model was chosen. This approach optimizes model performance by systematically exploring hyperparameter space, ensuring robustness and generalization to unseen data.

3. **Effectiveness of the algorithm when applied to the data:**
   In the model, higher recall for class 1 suggests that the model is effectively showing large proportion of actual instances of heart disease and correctly identifies patients who with heart disease.

4. **Relevant metrics used for demonstrating the model effectiveness:**
   Accuracy: The model achieved an accuracy of 87.50%, indicating that it correctly classified 87.50% of the instances in the test set.
   Cross Validation Score: The cross-validation score, which is slightly higher at 90.53%, suggests that the model's performance is consistent across different subsets of the data, indicating robustness.
   ROC_AUC Score: The ROC_AUC score of 87.43% indicates the model's ability to distinguish between positive and negative instances. A higher ROC_AUC score suggests better discrimination ability.

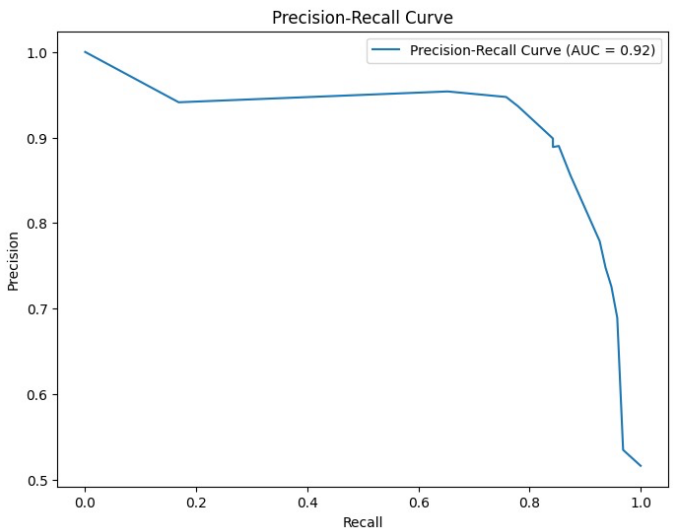
5. **Intelligence gained from the algorithm about the model:**
   `The model demonstrates good performance in predicting heart disease, with high accuracy, consistent cross-validation scores, and balanced precision and recall values for both classes. However, it's essential to consider the specific requirements and priorities of the application when interpreting these metrics. SVMs seek to identify the best decision boundary that optimizes the margin between classes, producing a well-defined boundary that facilitates comprehension of the connection between illness occurrence and input characteristic.

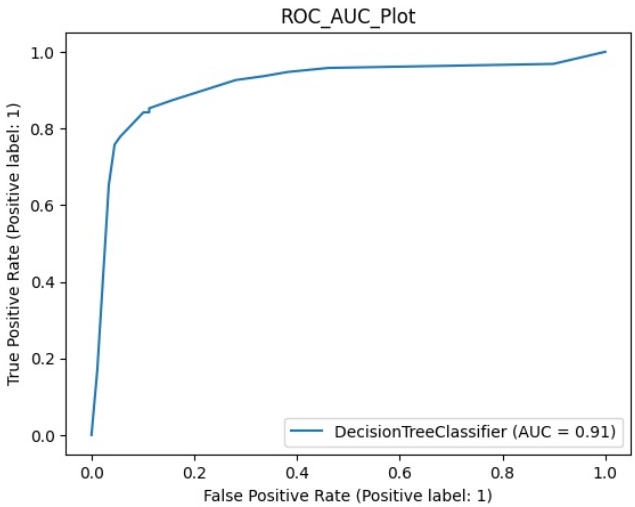
7. DECISION TREE:

EVALUATION MATRICS:

```
Accuracy :  85.87%
Cross Validation Score :  88.76%
ROC_AUC Score :  85.82%
```

	precision	recall	f1-score	support
0	0.86	0.84	0.85	89
1	0.86	0.87	0.86	95
accuracy			0.86	184
macro avg	0.86	0.86	0.86	184
weighted avg	0.86	0.86	0.86	184





1. **Why to choose the Decision Tree Algorithm:**
   Decision Trees offer interpretability and handle non-linear relationships well, making them suitable for heart disease prediction. They provide insights into significant risk factors and their interactions, aiding medical professionals in understanding and explaining diagnoses. However, they may suffer from overfitting and lack robustness compared to other models.

2. **Work performed to tune or train the model:**
   To tune and train the Decision Tree model, I iteratively experimented with different parameter combinations. I varied the max_depth, min_samples_split, and min_samples_leaf parameters to explore their effects on model performance. Each combination was trained on the training data, and its accuracy was evaluated on the test set. By systematically adjusting these parameters and evaluating the model's performance, I identified the combination that yielded the best accuracy. This iterative process allowed for fine-tuning the model's hyperparameters, optimizing its performance for predicting heart disease without overfitting.

3. **Effectiveness of the algorithm when applied to the data:**
   The Decision Tree algorithm demonstrates good effectiveness when applied to the heart disease prediction data, achieving an accuracy of 85.87%. With a cross-validation score of 88.06% and an ROC_AUC score of 85.78%, the model showcases robustness and discriminative capability in distinguishing between positive and negative instances of heart disease.

4. **Relevant metrics used for demonstrating the model effectiveness:**
   Relevant metrics such as precision, recall, and F1-score provide insight into the Decision Tree model's effectiveness. With precision values of 0.87 for class 0 and 0.85 for class 1, and recall values of 0.83 for class 0 and 0.88 for class 1, the model demonstrates balanced performance in correctly classifying instances for both classes. Additionally, F1-scores of 0.85 and 0.87 further validate the model's effectiveness in capturing the trade-off between precision and recall.

8. **Intelligence gained from the algorithm to your data about your model:**
   Examining precision, recall, and F1-score, we find balanced performance across both classes, with values around 0.85 to 0.87. This suggests that the model is adept at correctly classifying instances for both individuals with and without heart disease. Notably, the model highlights the importance of features captured within a maximum depth of 5 levels, with minimum samples required for splitting and leaf nodes set to 10 and 2, respectively. This indicates that complex interactions between features are captured effectively within these parameters, aiding in understanding significant risk factors contributing to heart disease prediction.

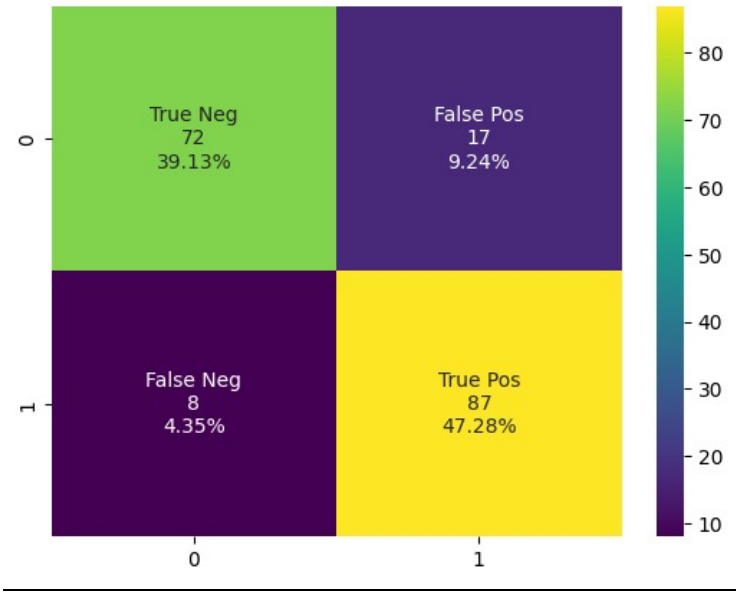
8. XGBOOST:

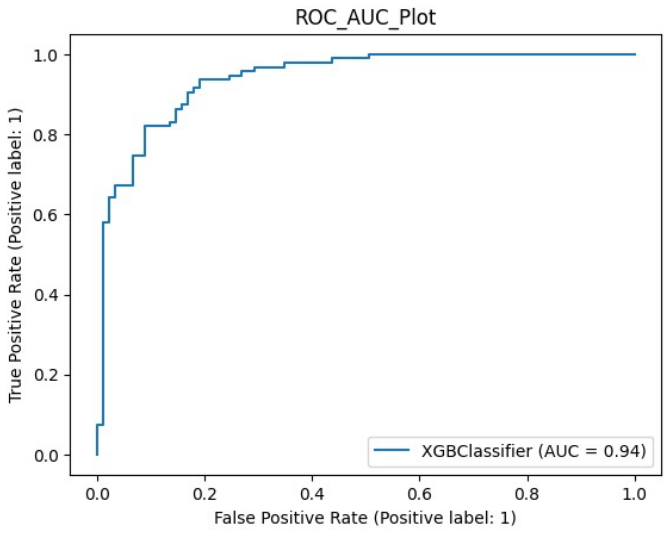
EVALUATION MATRICS:

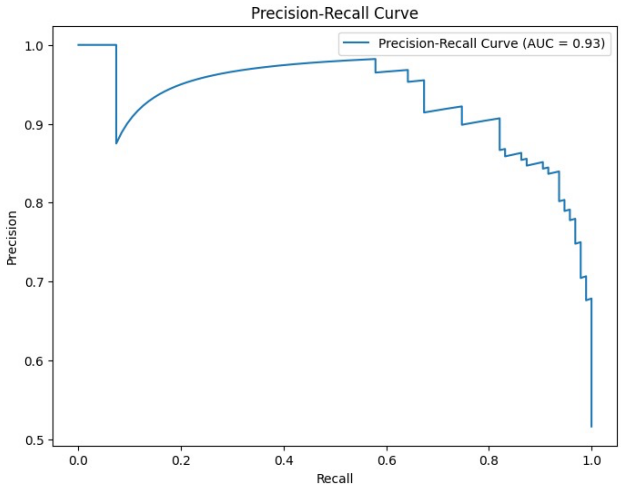
```
Accuracy of XGBoost alongside desicion tree:  86.41%
XGBoost with decision tree Cross-Validation Accuracy: 89.11%
              precision    recall  f1-score   support

           0       0.90      0.81      0.85        89
           1       0.84      0.92      0.87        95

    accuracy                           0.86       184
   macro avg       0.87      0.86      0.86       184
weighted avg       0.87      0.86      0.86       184
```







1. **Why to choose the XGBoost Algorithm:**
   Combining XGBoost with Random Forest and Decision tree makes the algorithm more powerful, robust and flexible. The efficiency and speed of Decision Tree increases when combined with XGBoost.
   XGBoost can handle missing data values more efficiently and internally. Thereby eliminating the requirement of imputation. XGBoost can be implemented in a parallel manner with distributed system and multiple cores. This allows for faster modelling of the datasets.
   Also, the overfitting can be avoided by using L1 and L2 regularization and thereby enhancing. the performance. Moreover, it allows hyperparameter tuning like tree depth, regularization parameter, which improves the overall performance of the model.
   The model is based on gradient boosting framework that train the decision tree for reduced loss function and optimized result.

2. **Work performed to tune or train the model:**
   Used GridSearchCV to tune data. This works on search in the hyperparameter space to find the best suitable combination of hyperparameters that results into better optimization of the model.
   Using GridSearchCV with XGBoost resulted into an increased accuracy of 86% from 85%.
   The regularization parameter 'gamma' used here represents minimum loss reduction. For hyperparameter tuning process, GridSearchCV examines the performance for the classifier with different 'gamma' values and finds the one that results into model balancing and generalization.

3. **Effectiveness of the algorithm when applied to the data:**
   XGBoost works on the prediction made by the weak learners-decision tree- and assemble a more powerful and accurate prediction.
   Overfitting is one of the issues faced by Decision Tree model. It can be overcome by using XGBoost by iteratively optimizing the loss function and regularization the hyperparameters like controlling depth of the tree.
   Therefore, XGBoost when applied to data, provides improved performance and accuracy.

   By combining the XGBoost with Decision Tree, the accuracy increased by 86.97% from 84.78%.

4. **Relevant metrics used for demonstrating the model effectiveness:**
   Metrics such as precision, recall, and F1-score provide insight into the XGboost effectiveness.
   The model provides Precision values of 0.89 for class 0 and 0.85 for class 1 and recall values of 0.83 for class 0 and 0.91 for class 1.
   The model demonstrates balanced performance in correctly classifying instances for both classes.
   He model shows an F1-scores of 0.8 and 0.88.
   Additionally, the model gives an accuracy of 86.96% when XGBoost is applied alongside decision tree and improves the performance.

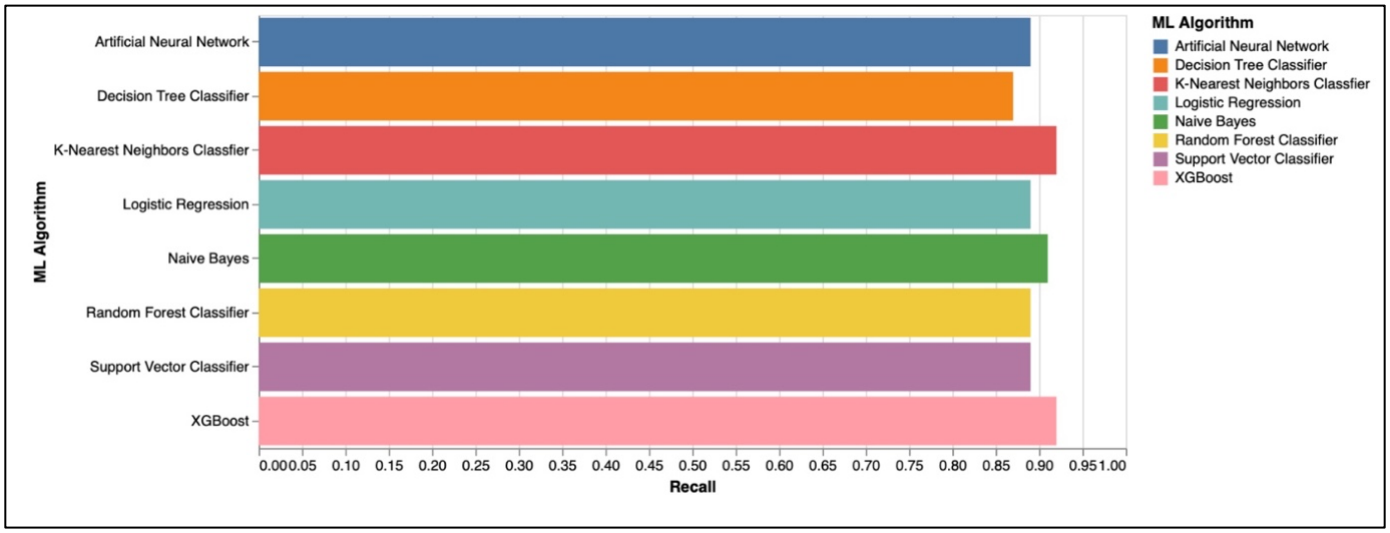
5. **Intelligence gained from the algorithm to your data about your model:**
   XGBoost with decision tree provides several insights about our model:
   Improved Prediction: higher accuracy enables health professionals to identify patients with high risk of heart disease.
   Health professionals can seek to the feature importance scores of XGBoost to identify the factors associated the heart disease and targeted interventions can be done.

   Also, By providing XGBoost alongside decision tree, accuracy slightly improved While the Cross Validation value remained consistent with different set of data.
   An improved performance is observed for other metrics as well, like – precision, recall, ROC_AUC score and F1 score. Thereby showing an improved performance collectively.

CONCLUSION:

COMPARISIONS OF ALL THE MODELS:

Sr. No.	ML Algorithm	Accuracy	Precision	Recall
1	Logistic Regression	0.87	0.86	0.89
2	Support Vector Classifier	0.88	0.87	0.89
3	Decision Tree Classifier	0.86	0.86	0.87
4	Random Forest Classifier	0.84	0.82	0.89
5	K-Nearest Neighbors Classfier	0.86	0.83	0.92
6	Artificial Neural Network	0.91	0.82	0.89
7	Naïve Byes	0.86	0.84	0.91
8	XGBoost	0.86	0.84	0.92



FINAL CHOICE OF MODEL FOR PHASE 3:

K-Nearest Neighbor

In this phase of our model, we have used **8 different classification algorithms** to classify whether a patient is prone to cardiovascular disease or not depending on multiple attributes.
We have performed training, tuning of hyperparameters and have further evaluated each model using various evaluation metrics such as accuracy, precision, recall, f1_score, and confusion matrix.

After evaluation we believe **K-Nearest Neighbor** would be an appropriate algorithm to classify whether a patient is prone to cardiovascular disease or not depending on multiple attributes. **KNN has performed with a recall of 0.92.**

In our problem we are dealing with medical data of patients which involves a nonlinear relationship between the patient characteristics(features) and the presence of cardiovascular disease. KNN is particularly capable of capturing non linearities without assuming a specific functional form, making it suitable for this type of problem.

The KNN algorithm's approach, which classifies a sample based on the majority class among its k nearest neighbors, aligns well with the underlying principle of identifying similar patterns in patients' data to predict their heart disease status accurately. Moreover, KNN is non-parametric and inherently adaptive to complex decision boundaries, **making it suitable for handling potentially nonlinear relationships between predictors and the target variable in heart disease prediction.**

Model Tuning and Training

KNN does not require training however to obtain better accuracy we tried to use different parameters to fit the model. We tried to experiment with different number of K nearest neighbors -and different distance metrics. After trying with different K nearest neighbor values and distance metrics we found that K= 9 and distance metric = manhattan to be giving the best results. Further, we used RepeatedStratifiedKFold, which combines stratified sampling and repeated KFold cross-validation, ensuring balanced class representation and reducing variance. It's valuable for robustly evaluating model performance.

Effectiveness of The Algorithm

In our project focused on detecting heart disease in patients, we have evaluated the effectiveness of various machine learning algorithms using multiple metrics, including accuracy, precision, recall, and ROC AUC Curve. After careful analysis, we have determined that recall is the most suitable evaluation metric for our problem and dataset.
It is evident that the K-Nearest Neighbors (KNN) Classifier achieved the highest recall score of 0.92, indicating its strong capability to correctly identify patients with heart disease among all actual positive cases. **Recall, for this problem and dataset, signifies the model's ability to correctly identify individuals with heart disease among all actual positive cases.** This high recall score signifies that the model has a lower tendency to miss identifying patients with heart disease, which is crucial for ensuring timely medical intervention.

Given the critical nature of accurately identifying patients with heart conditions to initiate timely medical interventions, a high recall rate is paramount.

Metrics for the Model Effectiveness.

The most relevant metric for detecting heart disease in patients would be recall, which is focused on correctly identifying those with the disease. While we have also performed other metrics such as accuracy, precision, and ROC AUC curve and they have helped us provide valuable insights into the model's performance, recall is particularly important in situations where identifying positive cases is of the most importance, such as in medical diagnosis.

Class 1 (Heart Disease) has higher recall, indicating that the model is better at capturing instances of heart disease but may have a slightly lower precision.

Further, we have also used a misclassification rate of approximately 0.141 or 14.1% means that about 14.1% of the instances in your dataset were misclassified by the model. Lower misclassification rates indicate better model performance, as they imply fewer incorrect predictions. We achieve an accuracy of 85.87% with our model. According to the results, 85.87% of the cases were correctly predicted by the model.

Cross Validation Score: With a cross-validation score of 91.48%, our model demonstrates robustness and generalizability, performing consistently well across different subsets of the data. The F1-scores for both classes are relatively high, indicating a good balance between precision and recall for each class.

Intelligence Gained from the model:

The K-Nearest Neighbors (KNN) model that we have used has provided in depth insights into the prediction of cardiovascular disease, contributing to a deeper understanding of the underlying patterns and risk factors associated with the condition:

By analyzing the KNN model, we can identify the most influential predictors of heart disease. KNN's nearest neighbor prospect shows the importance of similar patient profiles in predicting the presence or absence of heart disease. This insight can help us to prioritize interventions and screenings for individuals with high-risk factors such as elevated blood pressure, cholesterol levels, or age.

Further, KNN model inherently captures local data patterns and relationships by classifying samples based on their proximity to other data points. This visualization will help us to understand the relation between different risk factors and their impact on disease outcome which could be sometimes complex to understand.

By evaluating the KNN model's performance metrics, such as accuracy, precision, recall, and ROC AUC score, we can assess its effectiveness in predicting heart disease. By analyzing using these metrics, we can help validate the model's predictive capabilities in future.

Through the discovery of underlying data patterns, the identification of key predictors, and providing a framework for understanding the complex relationships between patient characteristics and disease outcome, the KNN model has contributed significantly to gaining insight into heart disease prediction.

REFERENCES

ANN Model – https://www.tensorflow.org/tutorials/quickstart/beginner

https://scikit-learn.org/stable/

Decision Tree – https://developers.google.com/machine-learning/decision-forests/decision-trees

Random Forest - https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

XGBoost – https://xgboost.readthedocs.io/en/stable/