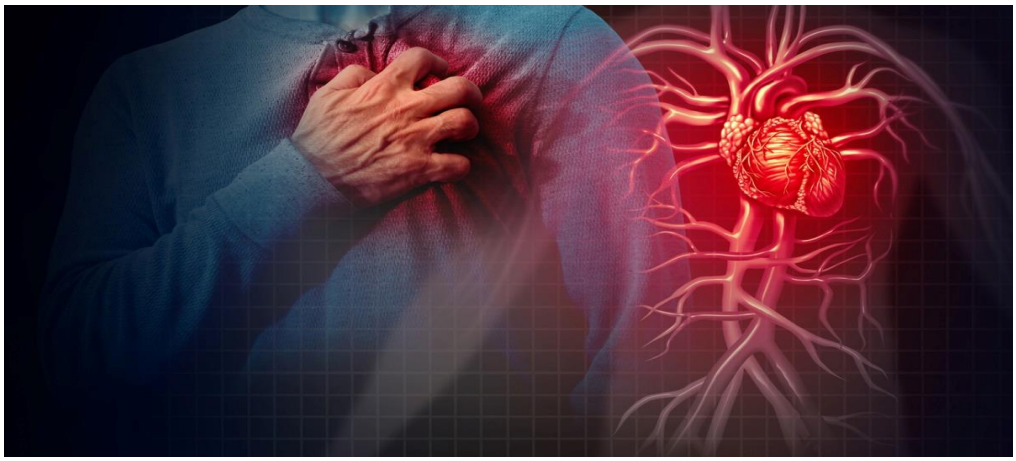
Heart Disease Detection



Abstract:

Day by day, the cases of heart diseases are increasing at a rapid rate, and it's very important and concerning to predict such diseases beforehand. This diagnosis is a difficult task and should be performed precisely and efficiently. The research paper mainly focuses on predicting which patient is more likely to have a heart disease based on various medical attributes. We prepared a heart disease prediction system to predict whether a patient is likely to be diagnosed with a heart disease or not using their medical history. We used different machine learning algorithms such as Logistic Regression, Support Vector Classifier (SVC), and Random Forest to predict and classify patients with heart disease. After evaluation, Logistic Regression and SVC with PCA (Principal Component Analysis) data were found to be the best-performing models for predicting heart disease with high accuracy. A helpful approach was employed to regulate how the model can be used to improve the accuracy of predicting heart disease in an individual. The strength of the proposed model was quite satisfying and was able to predict the likelihood of heart disease in an individual using Logistic Regression and SVC with PCA, which showed better accuracy compared to previously used classifiers such as Naive Bayes. A significant amount of pressure has been lifted off by using the given model in determining the probability of correctly and accurately identifying heart disease. The proposed heart disease prediction system enhances medical care and reduces costs. This project provides significant knowledge that can help predict patients at risk for heart disease.

Introduction:

Cardiovascular diseases (CVDs) are becoming increasingly prevalent, representing a broad spectrum of conditions that can impact the heart. According to the World Health Organization (WHO), approximately 17.9 million global deaths are attributed to CVDs annually, making them the leading cause of death among adults.

Our project aims to predict individuals at high risk for heart disease by analyzing their medical history. By identifying symptoms such as chest pain or high blood pressure, the system can assist in diagnosing potential heart disease cases with fewer medical tests, leading to more effective treatments and timely interventions.

This project focuses on three primary ML techniques to predict the likelihood of heart disease:

1. Logistic Regression
2. Support Vector Classifier (SVC)
3. Random Forest Classifier

The objective of this work is to determine whether a patient is likely to be diagnosed with cardiovascular diseases based on their medical attributes. This project is inspired by a growing body of research related to cardiovascular disease diagnosis using machine learning algorithms. A literature survey has been conducted to explore various models and their efficiencies in predicting heart disease.

Among the various algorithms explored, **Logistic Regression** and **SVC with PCA-transformed data** have shown promising results. Each algorithm demonstrated distinct strengths in achieving the defined objectives. Conversely, models such as **Random Forest** and **SVC with all data** frequently exhibited overfitting, limiting their generalizability and practical application.

While some models incorporate crucial factors, like family history, in the decision-making process, their accuracy tends to be lower compared to newer machine learning and deep learning models, which offer enhanced precision in predicting coronary heart disease. The data used for this analysis was sourced from the **Framingham Heart Study dataset**, available on Kaggle (https://www.kaggle.com/datasets/navink25/framingham).

Data Attributes:

The dataset comprises several key attributes categorized into demographics, lifestyle factors, medical history, health metrics, and the target variable. Understanding these attributes is essential for analyzing the risk factors associated with coronary heart disease (CHD).

Demographics

- **Sex**: This attribute indicates the gender of the individual, which can influence heart disease risk due to biological and lifestyle differences.
- **Age**: Age is a significant risk factor for CHD, with older individuals generally at higher risk due to cumulative exposure to various risk factors over time.

Lifestyle

- **Current Smoker**: This binary attribute denotes whether the individual currently smokes tobacco, a known risk factor for cardiovascular diseases.
- **Cigs Per Day**: This variable quantifies the number of cigarettes smoked daily, allowing for a more nuanced understanding of smoking's impact on heart health.

Medical History

- **BP Meds**: This indicates whether the individual is on blood pressure medications, reflecting their history of hypertension, which is a significant contributor to heart disease.
- **Prevalent Stroke**: A binary indicator of whether the individual has a history of stroke, which can correlate with increased CHD risk.
- **Prevalent Hyp**: This attribute indicates whether the individual has a history of hypertension.
- **Diabetes**: This attribute identifies whether the individual has been diagnosed with diabetes, another major risk factor for cardiovascular conditions.

Health Metrics

- **Tot Chol**: Total cholesterol levels can indicate the risk of heart disease, with higher levels often associated with increased risk.
- **Sys BP**: Systolic blood pressure is a critical measure of cardiovascular health, with elevated levels indicating hypertension.
- **Dia BP**: Diastolic blood pressure, like systolic, contributes to the overall assessment of blood pressure health.
- **BMI**: Body Mass Index is a measure of body fat based on weight and height, serving as an important indicator of obesity-related heart disease risk.
- **Heart Rate**: This metric reflects the number of heartbeats per minute, with abnormal rates potentially signaling cardiovascular issues.
- **Glucose**: Blood glucose levels can indicate diabetes risk and metabolic health, influencing overall cardiovascular risk.

Target Variable

- **10-Year CHD Risk**: This is the primary outcome of interest in the dataset, representing the estimated risk of developing coronary heart disease within the next ten years based on the combined influence of the aforementioned attributes. This variable is essential for assessing the effectiveness of risk factors in predicting long-term heart health outcomes.

Importing Libraries and Data Preparation

To effectively analyze the dataset on coronary heart disease risk factors, several essential Python libraries will be utilized. These libraries provide powerful tools for data manipulation, visualization, and machine learning, facilitating a comprehensive understanding of the dataset.

1. **NumPy**: A fundamental library for scientific computing in Python, NumPy provides support for large, multi-dimensional arrays and matrices. It also offers a wide variety of mathematical functions to operate on these arrays, including linear algebra, statistical, and random number operations.
2. **Pandas**: Built on top of NumPy, Pandas is a powerful library for data manipulation and analysis. It provides two main data structures: Series (1D) and DataFrame (2D), which make it easy to manipulate, clean, and analyze structured data, like CSV files or SQL tables.
3. **Matplotlib**: Matplotlib is a plotting library for creating static, animated, and interactive visualizations in Python. It provides a wide range of plotting options such as line charts, bar graphs, scatter plots, and histograms, allowing for high-quality visual representations of data.
4. **Seaborn**: Built on top of Matplotlib, Seaborn is a statistical data visualization library that simplifies creating informative and attractive visualizations. It includes features for creating complex visualizations like heatmaps, violin plots, and pair plots, with less code compared to Matplotlib.
5. **Scikit-learn (sklearn)**: A key library for machine learning in Python, Scikit-learn provides tools for data preprocessing, classification, regression, clustering, and dimensionality reduction. It also includes a wide range of models and algorithms, making it easy to train, test, and validate machine learning models.
6. **Pickle**: Pickle is a module in Python used to serialize and deserialize Python objects, allowing them to be saved to a file and later restored. It's commonly used to save machine learning models after training so they can be reused without retraining.
7. **Streamlit**: Streamlit is an open-source library that allows the creation of web applications for data science and machine learning projects with minimal effort. It enables rapid prototyping of interactive dashboards and applications, directly from Python scripts, without the need for web development skills.

Overview of Logistic Regression, SVC, and Random Forest Classifier

1. **Logistic Regression**: Logistic Regression is a statistical method for binary classification, where the outcome is a probability that is mapped to a binary class (0 or 1). It uses a logistic function (sigmoid) to model the relationship between the input features and the probability of the class. The model finds the best-fit line (hyperplane in higher dimensions) that separates the two classes based on the training data.
2. **Support Vector Classifier (SVC)**: SVC is a powerful classification algorithm that aims to find the optimal hyperplane that maximizes the margin between classes. The margin is the distance between the hyperplane and the closest data points from either class, known as support vectors. SVC works well for both linear and non-linear classification tasks by using kernel tricks to

transform data into higher-dimensional spaces, where a hyperplane can better separate the classes.

3. **Random Forest Classifier**: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the majority vote from all the individual trees. It introduces randomness by selecting random subsets of features and data points to build each tree, which helps improve model generalization and reduce overfitting. Random Forest is particularly useful for handling large datasets with complex relationships and provides good accuracy with less tuning.

These three models represent different approaches to classification tasks, with Logistic Regression focusing on a linear decision boundary, SVC looking for the optimal separation in feature space, and Random Forest leveraging the power of multiple decision trees for robust and flexible classification.

First I making three fuction that I used :

1. **information(data)**: This function provides a summary of the dataset. It returns the data types of each feature, the number of unique values in each column, and the number of missing values (nulls). Additionally, it counts and prints the number of duplicate records in the dataset, as well as the shape of the dataset (i.e., the number of records and features). This function is useful for performing initial data exploration and understanding the dataset's structure.
2. **train_classifier(model, x_train, y_train, x_test, y_test)**: This function is used to train a machine learning model and evaluate its performance. It takes a model (e.g., Logistic Regression, SVC, or Random Forest) and training data (x_train, y_train) along with test data (x_test, y_test). The function fits the model to the training data, then makes predictions on both the training and test datasets. It calculates several evaluation metrics, including accuracy, recall, classification report, and confusion matrix for both training and testing. It also plots the ROC curve to visualize the model's performance in distinguishing between classes.
3. **plot_decision_boundary(model, X, y)**: This function visualizes the decision boundary of a classifier on a 2D dataset. It generates a mesh grid covering the feature space and predicts class labels for each point in the grid. The decision boundary is then plotted, along with the actual data points, showing how the classifier separates the classes. This is especially useful for understanding how the model classifies data and the nature of its decision-making process, particularly for models trained on 2D datasets.

These functions collectively serve the purpose of dataset inspection, model training and evaluation, and visualizing the decision boundaries of classifiers.

Data Information Overview:

The dataset consists of both continuous and categorical columns.

Continuous Columns: These include variables such as age, cigsPerDay, totChol, sysBP, diaBP, BMI, heartRate, and glucose, which contain numerical values that can take any value within a range.

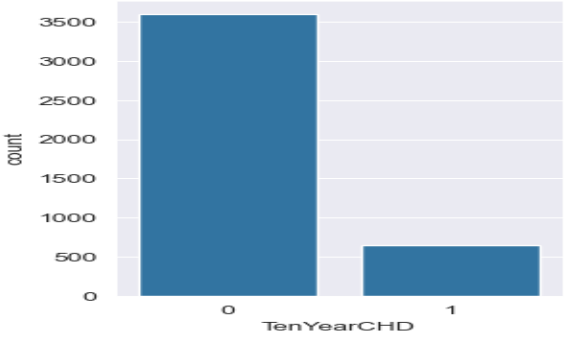
Categorical Columns: This group includes variables like BPMeds, prevalentStroke, prevalentHyp, Sex, and diabetes, which represent distinct categories or classes.

Irrelevant and Redundant Columns: The education column was considered irrelevant, and currentSmoker was redundant with the cigsPerDay column. These columns have been removed from the dataset to ensure a cleaner and more relevant dataset.

Column Types

- con_cols: Refers to the continuous columns, which contain numerical values that can vary over a range.
- nom_numcols: Refers to nominal columns encoded as numerical values, where the numbers do not imply any specific order or ranking.
- cat_col: Refers to the categorical columns, which represent distinct categories or groups in the data.

Target Column Imbalance:



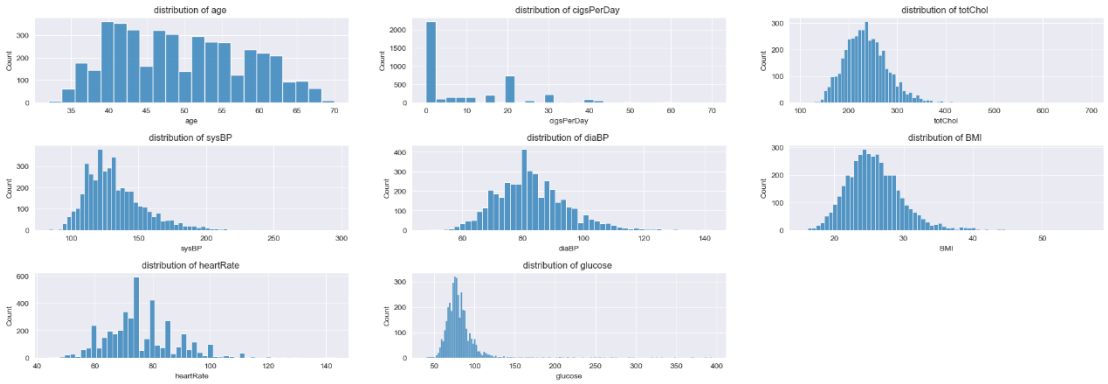
The target column in the dataset exhibits imbalanced data, where the number of individuals without heart disease significantly outnumbers those with heart disease. This imbalance means that the distribution of values in the target column is skewed, which can affect the performance of machine learning models.

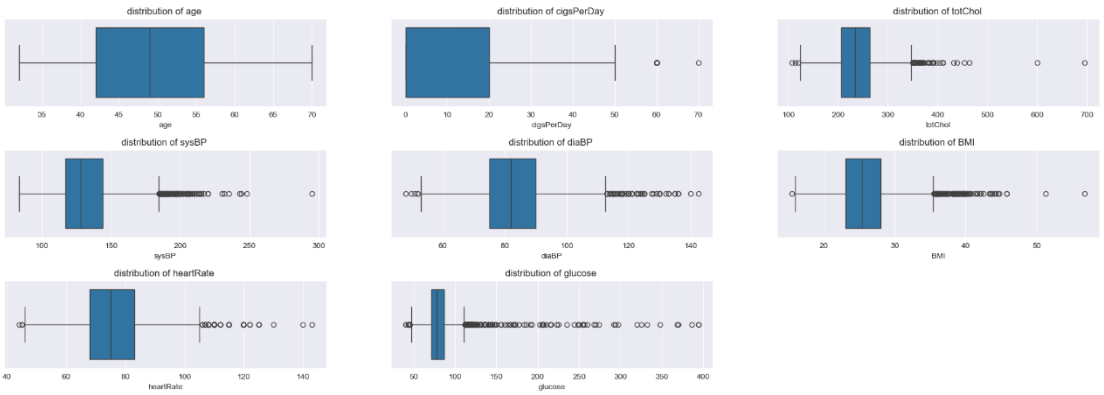
Outliers in Continuous Columns

After examining the continuous columns in the dataset, no values were identified as outliers. The data appears to be within expected ranges, with no extreme deviations that would indicate outliers.

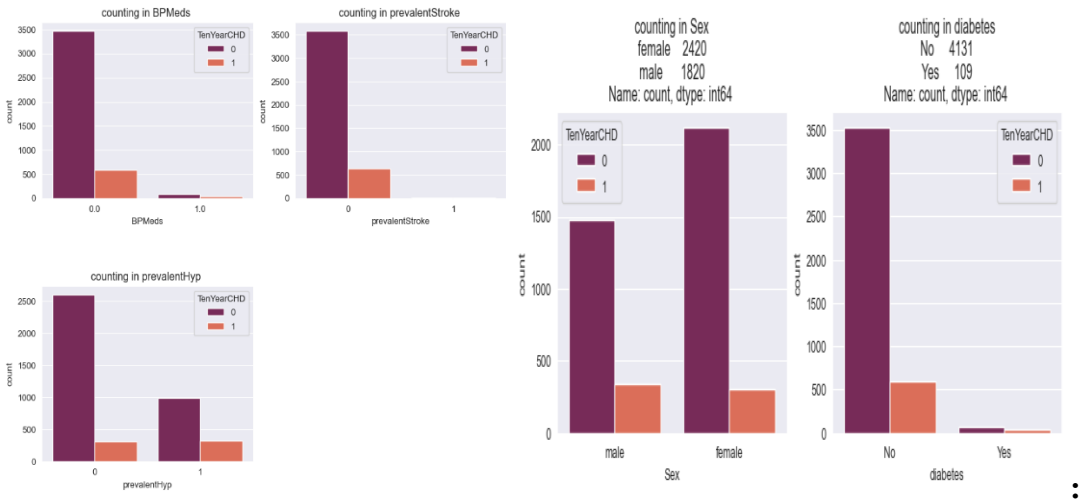
Data Distribution Summary

- **Age**: The distribution of age follows a normal distribution, suggesting symmetrical variation across different age intervals.
- **Total Cholesterol (totChol)**, **Systolic Blood Pressure (sysBP)**, **Diastolic Blood Pressure (diaBP)**, and **BMI (Body Mass Index)**: These variables also follow a normal distribution, indicating balanced and symmetrical differences in the data.
- **Heart Rate** and **Glucose Levels**: Both of these variables exhibit a normal distribution, with values symmetrically spread around the mean.
- **Cigarettes Per Day (CigsPerDay)**: This variable shows a positive skew, meaning the majority of the data points are concentrated on the lower end, with fewer high values.





Key Observations and Data Insights:



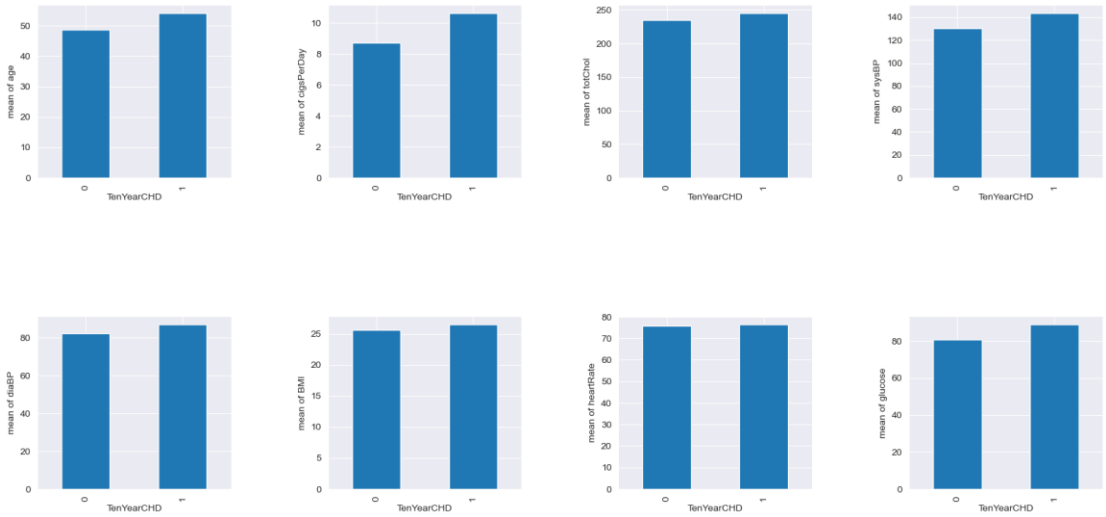
:

- **Blood Pressure Medication [BPMeds]:**
  People taking blood pressure medications are more likely to develop heart disease compared to those who do not. ***(Accompanied by a visualization showing heart disease prevalence by BPMeds status.)***
- **Stroke [PrevalentStroke]:**
  Individuals who have suffered a stroke are also more likely to develop heart disease, though they represent a very small minority. ***(Visualization highlighting the relationship between stroke history and heart disease.)***
- **High Blood Pressure [PrevalentHyp]:**
  A clear relationship is evident among those who suffer from high blood pressure and their likelihood of developing heart disease. ***(Plot showing heart disease prevalence among individuals with and without high blood pressure.)***
- **Gender Differences:**
  While the number of males is smaller than the number of females, males are more likely to develop heart disease compared to females. ***(Graph comparing heart disease prevalence by gender.)***
- **Diabetes and Heart Disease:**
  A significant percentage of individuals with diabetes are at a high risk of developing heart disease. ***(Visualization demonstrating the proportion of individuals with diabetes who have heart disease.)***

Correlation Analysis of Continuous Variables:

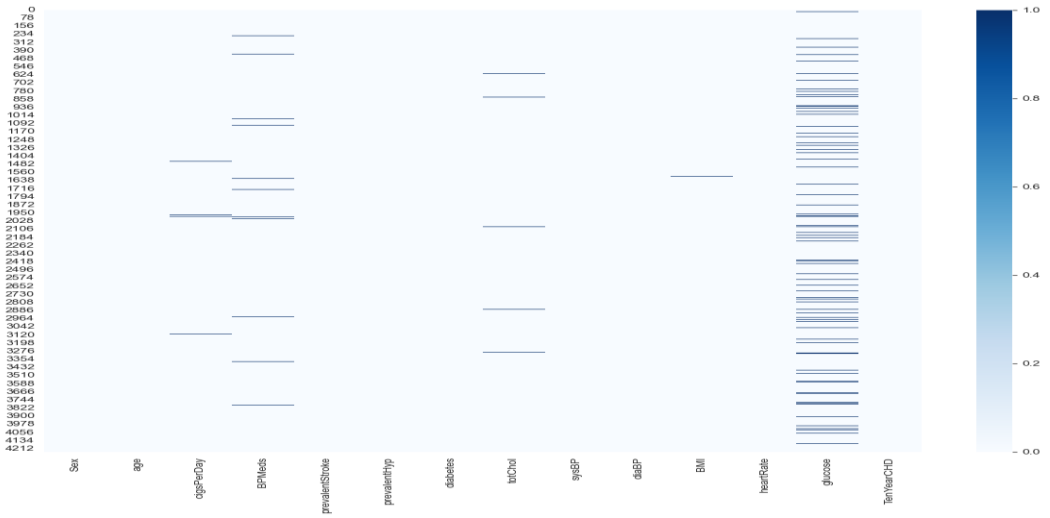
To analyze the correlation between continuous variables, we examine how these variables relate to each other to identify patterns or associations. ***(Heatmap or pair plot showing correlations among continuous variables.)***

Plot Description: Average of Continuous Columns by Target Values:



This plot illustrates the average values of all continuous columns, grouped by the two categories in the target column. It provides insights into how the continuous variables differ based on the target classification. ***(Bar plot showing the mean values of continuous columns for each target class.)***

Distribution of Nulls in the Dataset:



This section shows the distribution of null values in the dataset. ***(Graph displaying the count or percentage of null values for each column.)***

Handling Missing Values

Since no outliers were identified and the majority of the continuous columns follow a normal distribution, the next step is to handle the missing (null) values. For most continuous columns, missing values will be imputed using the **median** or **mean** of the respective columns.

- **Median**: The median will be used for columns with a skewed distribution (e.g., CigsPerDay), as it is less sensitive to extreme values.
- **Mean**: The mean will be used for columns that follow a normal distribution (e.g., Age, totChol, sysBP, diaBP, BMI, heartRate, glucose), as it provides a good representation of the central tendency.

Data Preprocessing

After cleaning the data and handling missing values, the next step is data preprocessing. The key steps in this phase are:

1. **Splitting Data**:
   The dataset is split into **training** and **testing** sets, typically with 70-80% of the data used for training and 20-30% for testing. This ensures that we can train the model on one subset and evaluate its performance on another, unseen subset, helping prevent overfitting.

2. **Encoding Categorical Variables**:
   For any categorical columns (such as Sex, BPMeds, diabetes, etc.), we apply encoding techniques like **Label Encoding** to convert these categorical values into numerical representations that machine learning models can process.
3. **Feature Scaling**:
   Continuous variables (such as age, totChol, sysBP, etc.) will be scaled using techniques like **Standardization (Z-score scaling)** This ensures that all features are on the same scale, which is particularly important for models sensitive to feature magnitudes, like SVM or logistic regretion

By following these preprocessing steps—splitting the data, encoding categorical variables, and scaling features—we ensure the dataset is properly prepared for training machine learning models.

After splitting the data into training and testing sets, we create a copy of the dataset to apply feature scaling. This results in two versions of the data:

1. **Unscaled Data for Random Forest**:
   The original dataset (i.e., x_train and x_test) remains **unscaled**. This version is used for algorithms like **Random Forest**, which are not sensitive to feature scaling.
2. **Scaled Data for Logistic Regression and SVM**:
   A copy of the dataset is created (i.e., x_train1 and x_test1) and then **scaled** using techniques like **Standardization (Z-score scaling)** or **Min-Max Scaling**. This scaled version is used for models like **Logistic Regression** and **Support Vector Machines (SVM)**, which perform better when features are on the same scale.

By maintaining both scaled and unscaled versions, we can apply the appropriate preprocessing for different machine learning algorithms to optimize their performance.

Evaluation Metrics Overview

1. **Accuracy**:
   Accuracy measures the overall correctness of a model by calculating the proportion of correct predictions (both positive and negative) out of the total predictions. It’s a general indicator of performance, though it may be misleading in imbalanced datasets.
2. **Precision**:
   Precision focuses on the quality of positive predictions, showing how many of the predicted positive instances are actually positive. It is important when minimizing false positives is critical.
3. **Recall (Sensitivity or True Positive Rate)**:
   Recall measures the model's ability to correctly identify all actual positive instances. It is crucial when minimizing false negatives is important, ensuring that true positives are not missed.
4. **ROC Curve**:
   The ROC curve plots the True Positive Rate (Recall) against the False Positive

Rate at various classification thresholds. The area under the ROC curve (AUC) gives a comprehensive measure of the model's ability to distinguish between classes, with a good model having an AUC closer to 1.

Summary:

- **Accuracy** gives a general performance metric but may not reflect model performance in imbalanced datasets.
- **Precision** is key when false positives need to be minimized.
- **Recall** is essential when the cost of missing true positives is high.
- **ROC Curve** helps assess a model's overall performance, with an ideal curve near the top-left corner, indicating high sensitivity and low false positives.

In the context of predicting heart disease, it is crucial to prioritize the following:

- ****Maximizing True Positives (Recall)****:

Identifying individuals who truly have heart disease is critical for early intervention and treatment. Missing these cases (false negatives) could lead to severe consequences.

- ****Minimizing False Negatives****:

A false negative occurs when the model predicts that a person does not have heart disease when, in fact, they do. Reducing false negatives is essential to ensure no high-risk individual is overlooked

Why Recall Matters:

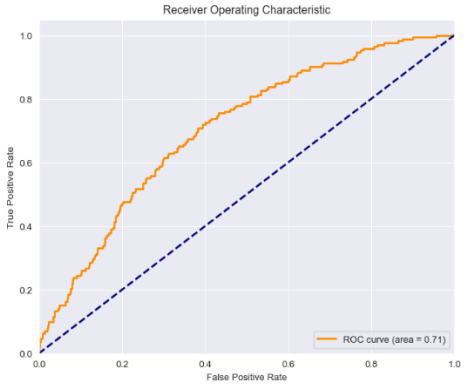
Given the life-threatening nature of heart disease, ****Recall**** (True Positive Rate) is the most important metric. It ensures that the model is effective in identifying as many individuals with heart disease as possible, even if it means tolerating some false positives (which can be verified through additional tests).

Applying ML Models:

Model Selection: Logistic Regression as the Best Model

After evaluating three different models — **Logistic Regression**, **SVC**, and **Random Forest** — based on accuracy, recall, and performance on both training and testing datasets, the following conclusions were drawn:

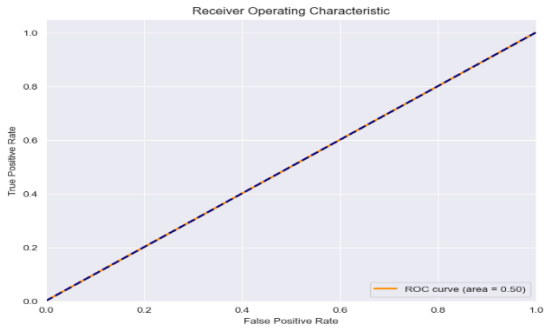
Logistic Regression:



- **Training Accuracy**: 86%
- **Testing Accuracy**: 84%
- **Training Recall**: 8.5%
- **Testing Recall**: 4.6%

Logistic Regression provides a reasonable balance between training and testing accuracy, with relatively stable performance across both sets. Despite the lower recall values, it performs consistently without signs of overfitting, making it suitable for this dataset.

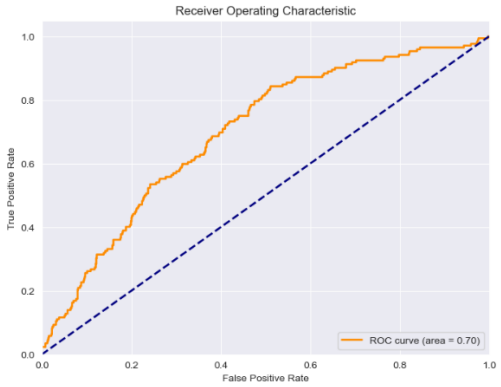
SVC (Support Vector Classification):



- **Training Accuracy**: 94%
- **Testing Accuracy**: 84%
- **Training Recall**: 61%
- **Testing Recall**: 0%

SVC showed high training accuracy and recall, but it failed to generalize well to the testing set, where the recall for the positive class dropped to 0%. This indicates significant overfitting, making SVC less ideal for this analysis.

Random Forest:



- **Training Accuracy**: 95%
- **Testing Accuracy**: 83%
- **Training Recall**: 64%
- **Testing Recall**: 2%

Similar to SVC, Random Forest exhibited high training accuracy and recall but suffered from overfitting, with a sharp decline in performance on the testing set, especially for recall. This suggests the model learned specific patterns from the training data that did not generalize well.

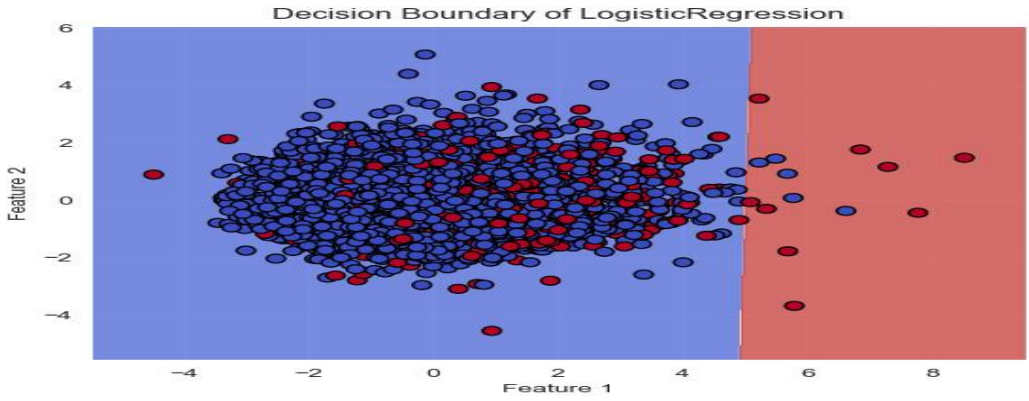
Conclusion:

While **Random Forest** and **SVC** achieved higher training performance, they both showed signs of overfitting and did not generalize well to unseen data. **Logistic Regression** provided the most consistent and reliable performance across both training and testing sets, making it the best choice for this dataset. Despite the lower recall, the stable accuracy and lack of overfitting make it the preferred model for this analysis.

Dimensionality Reduction with Principal Component Analysis (PCA):

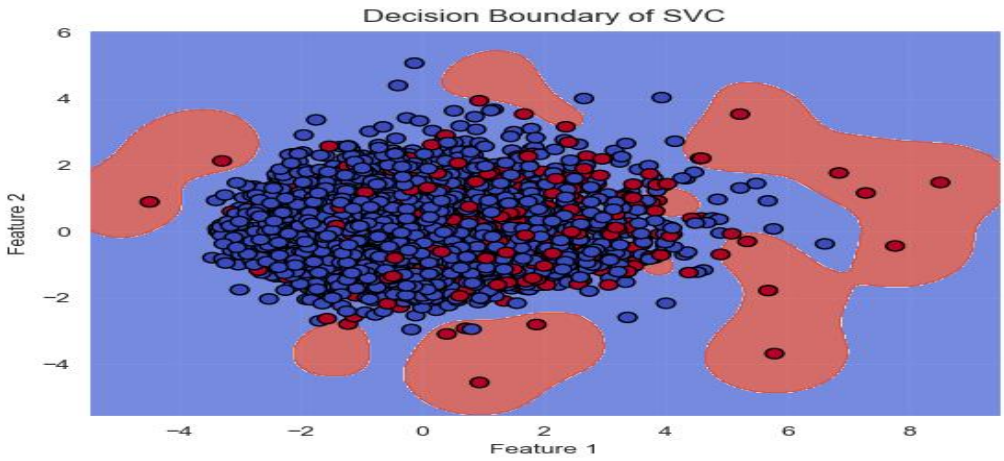
Decision Boundary Plot for Logistic Regression with Two Features

To visualize the decision boundary for **Logistic Regression**, we use only two features from the dataset. This helps us understand how the model makes predictions based on the features and allows us to visualize the boundary between the classes.



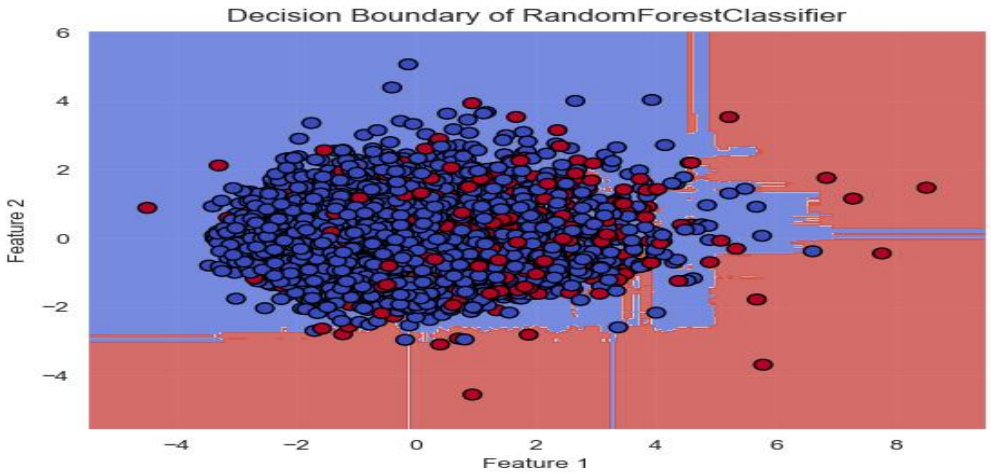
Decision Boundary Plot for SVC with Two Features

Similarly, we plot the decision boundary for **SVC (Support Vector Classification)** using two features. This allows us to examine the model's behavior and the way it separates the classes in a 2D space.

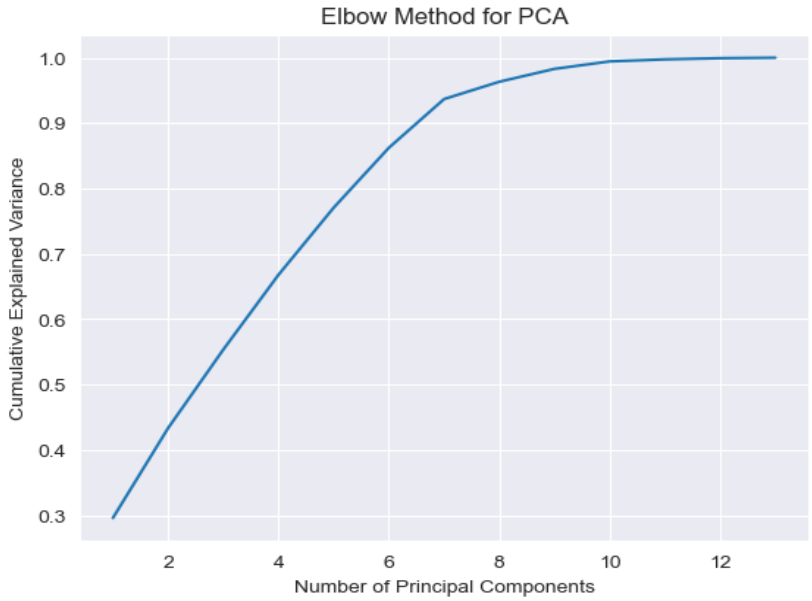


Decision Boundary Plot for Random Forest with Two Features

For **Random Forest**, we also visualize the decision boundary using two features. The plot gives us insight into how Random Forest distinguishes between different classes.



Using the Elbow Technique to Find the Best Number of Dimensions:



To reduce the dimensionality of the dataset, we apply the **Elbow Technique** in conjunction with **Principal Component Analysis (PCA)**. This method involves plotting the **explained variance** for each principal component and identifying the "elbow point." The elbow point represents the optimal number of dimensions, where the explained variance begins to plateau. This helps in selecting the best number of dimensions to retain for further analysis while preserving most of the information from the original data.

New Data After PCA Transformation

After applying **Principal Component Analysis (PCA)**, the original data is transformed into a lower-dimensional space. The following datasets are obtained after applying PCA:

- **x_train_pca**: The training data after PCA transformation.
- **x_test_pca**: The testing data after PCA transformation.

These transformed datasets now consist of **7 features**, which represent the optimal number of dimensions for the analysis. This reduction helps simplify the models while retaining the most important information for classification tasks.

Final Model Selection and Implementation Plan

After thorough training and evaluation of various classification models, we have selected **Logistic Regression** and **Support Vector Classifier (SVC)** with **Principal Component Analysis (PCA)** as our final models for predicting coronary heart disease (CHD) risk. The decision was based on performance metrics such as accuracy, precision, and recall, which indicated that both models effectively captured the underlying patterns in the dataset while maintaining a balance between sensitivity and specificity.

Logistic Regression is favored for its simplicity and interpretability. It provides a clear understanding of how each feature contributes to the prediction, making it an excellent choice for medical professionals who require transparency in decision-making processes. On the other hand, SVC, particularly when combined with PCA, excels in handling high-dimensional data. Its ability to create non-linear boundaries allows it to model complex relationships that may exist between risk factors and CHD, which can be crucial in a dataset with numerous interrelated variables.

To implement these models, we propose using **Streamlit**, an open-source app framework that enables the creation of interactive web applications. This tool allows users to visualize model predictions and compare the performance of Logistic Regression and SVC effectively. The implementation plan involves several key steps:

Model Deployment: Both trained models will be saved using pickle for easy deployment within the Streamlit app. This process will ensure that we can load the models quickly and efficiently when users make predictions.

User Interface Development: The Streamlit app will feature an intuitive interface where users can input patient data, such as age, cholesterol levels, and lifestyle factors. The app will then display the predicted CHD risk based on both models, allowing for side-by-side comparison.

By following this implementation plan, we will ensure that users can leverage the predictive power of our models to better assess heart disease risk, ultimately contributing to improved healthcare outcomes.

Thank You For Reading My Project Documentation , And Take care And Be Careful For Your Health 😊