**Heart Disease Classification Using ML**

**Abstract**

Day by day the cases of heart diseases are increasing at a rapid rate and it’s very Important and concerning to predict any such diseases beforehand. This diagnosis is a difficult task i.e. it should be performed precisely and efficiently. The research paper mainly focuses on 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 the patient is likely to be diagnosed with a heart disease or not using the medical history of the patient. We used different algorithms of machine learning such as logistic regression, SVM, Random Forest to predict and classify the patient with heart disease. A quite Helpful approach was used to regulate how the model can be used to improve the accuracy of prediction of Heart Attack in any individual. The strength of the proposed model was quiet satisfying and was able to predict evidence of having a heart disease in a particular individual by using Random Forest, SVM and Logistic Regression which showed a good accuracy. So, a quiet significant amount of pressure has been lifted off by using the given model in finding the probability of the classifier to identify the heart disease correctly and accurately. The Given heart disease prediction system enhances medical care and reduces the cost. This project gives us significant knowledge that can help us predict the patients with heart disease It is implemented on the. ipynb format.

**Objective**

The primary objective of this project is to develop and implement a classification algorithm capable of predicting the condition of heart disease based on angiographic findings with the ultimate goal of assisting healthcare professionals in making informed decisions for patient care.

**Dataset Source**

**We used the heart disease that is available in the UCI Machine Learning Respiratory Disease dataset, contain several pieces of data information on the disease instances. These are provided by the following for Clinical Shoshan Level and Clinical Foundation, Hungarian Tissue Cardiology, Long Beach Medical Center and University Hospital is Switzerland.**

Number of instances: 303

No. of columns: 14 continuous columns.

* age: Age in years
* sex: Sex (1= male, 0= female)
* cp: Chest pain type (Value 1: typical angina, Value 2: atypical angina, Value3: non-anginal pain, Value 4: asymptomatic)
* trestbps: Resting blood pressure (in mmHg on admission to the hospital)
* Chol: Serum Cholesterol in mg/dl
* fbs: fast blood sugar > 120 mg/dl (1= true, 0=false)
* restecg: Resting electrocardiographic results (0: normal, 1: having ST-T wave abnormality (T wave inversions and/or St elevation or depression of > 0.05mV, 2: showing probable or definite left ventricular hypertrophy.)
* thalach: Maximum heart rate achieved
* exang: Exercise included angina (1= yes, 0=no)
* old peak: ST depression induced by exercise relative to rest
* slope: the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: downsloping)
* ca: Number of major vessels (0-3) colored by fluoroscopy
* thal: A blood disorder called thalassemia.

3= normal, 6= fixed defect, 7= reversable defect

* Heart Disease: Diagnosis of heart disease- Angiprahic disease status

Value 0: No Heart Disease

Value 1: Heart Disease.

**Important Libraries**

The following libraries were used in this project for various tasks:

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings("ignore")

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import GridSearchCV

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

import pickle

After importing the libraries, we need to load the dataset which is in excel format into dataframe So, we used this code:

df = pd.read\_excel(r"Ch3.ClevelandData.xlsx")

**Data Cleaning & Preprocessing**

We checked the information of our dataset using code df.info ():

Range Index: 303 entries, 0 to 302

Data columns (total 14 columns):

# Column Non-Null Count Dtype

0 age 303 non-null int64

1 sex 303 non-null int64

2 cp 303 non-null int64

3 trestbps 303 non-null int64

4 chol 303 non-null int64

5 fbs 303 non-null int64

6 restecg 303 non-null int64

7 thalach 303 non-null int64

8 exang 303 non-null int64

9 oldpeak 303 non-null float64

10 slope 303 non-null int64

11 ca 303 non-null object

12 thal 303 non-null object

13 HeartDisease 303 non-null int64

dtypes: float64(1), int64(11), object(2)

As we can see there is 2 columns which have datatype “object” so we need to convert that into numeric datatype (i.e., int64 or float64) for further analysis.

While we were converting, we got to know there is some special characters like “?”. So, we replaced it with NaN values and then we filled it with Mode values.

df['ca']= df['ca'].replace(“?”, np.NaN)

df['thal']= df['thal'].replace(“?”, np.NaN)

# Replace NaN with 0 in the 'ca' column

df['ca'] = df['ca'].fillna(0)

# Replace NaN with 0 in the 'ca' column

df['thal'] = df['thal'].fillna(3)

# Convert the 'ca' column to np.float64

df['ca'] = df['ca'].astype(np.float64)

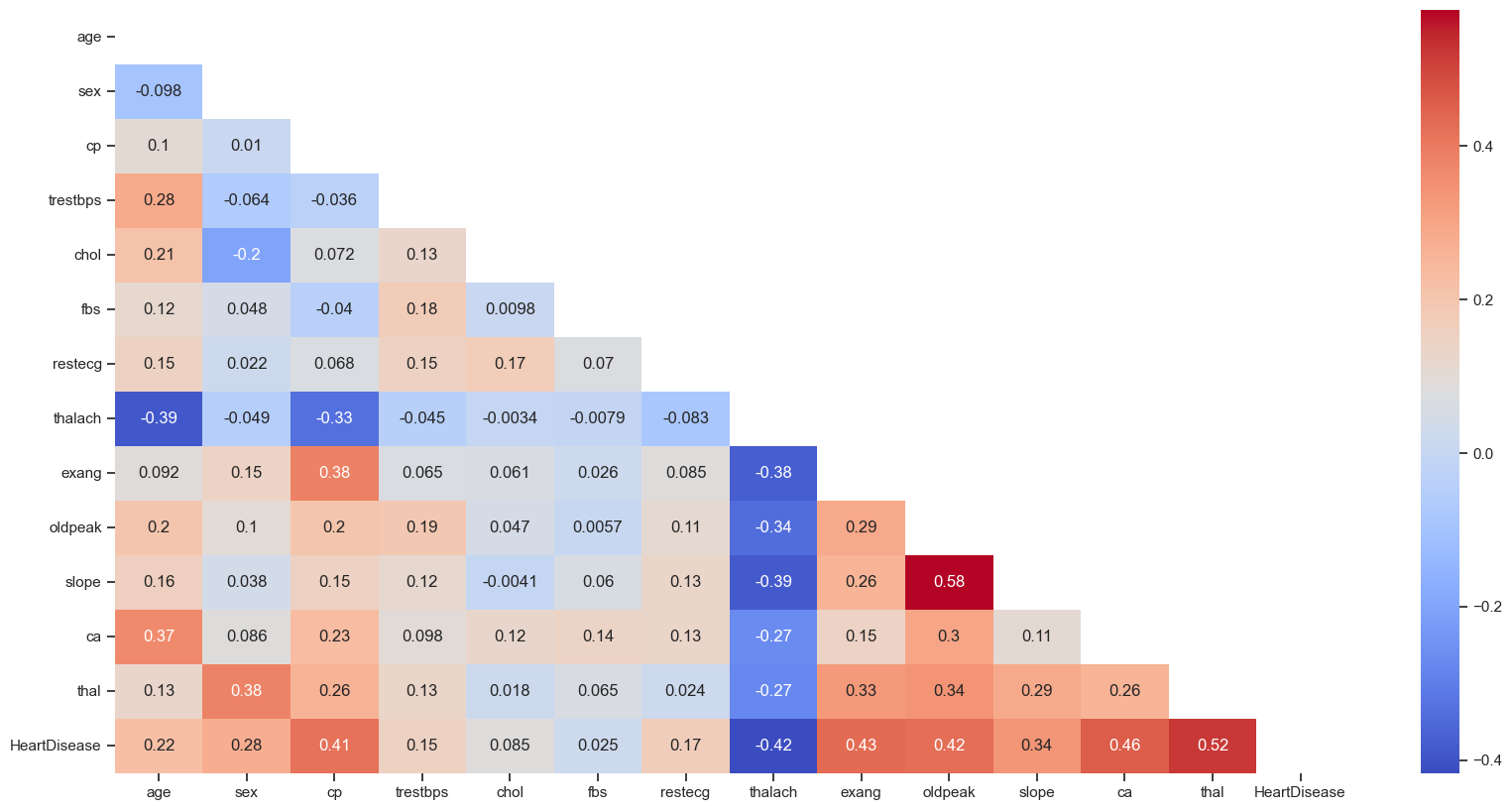
# Convert the 'ca' column to np.float64

df['thal'] = df['thal'].astype(np.float64)

Now, we changed the datatype also.

**Correlation using Heatmap**

The heatmap represents the correlation coefficients between various health-related variables and heart disease. Here are the key insights:



1. **Strong Positive Correlations**:

* + Thalassemia(thal) and Heart Disease: Both are highly correlated with each other.
  + Oldpeak (ST depression) and Slope: Greater ST depression is linked to a steeper slope of the peak exercise ST segment.

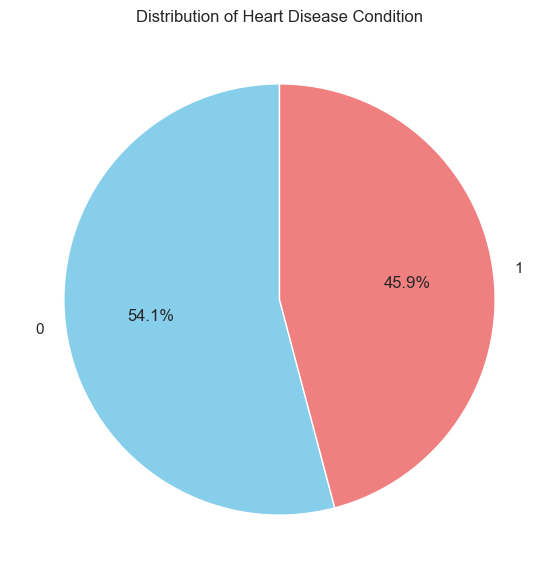
1. **Strong Negative Correlations**:
   * Exercise-Induced Angina (exang) and Maximum Heart Rate (thalach): Presence of angina during exercise is associated with a lower maximum heart rate.
   * Number of Major Vessels (ca) and Maximum Heart Rate (thalach): More major vessels colored by fluoroscopy correlate with a lower maximum heart rate.
   * Maximum Heart Rate(thalach) & Heart Disease are strongly negatively correlated.
2. **No Significant Direct Correlations**:
   * Age, Sex, trestbps, fbs, restecg and Cholesterol (chol) do not show strong direct correlations with heart disease in this dataset.

**Exploratory Data Analysis**

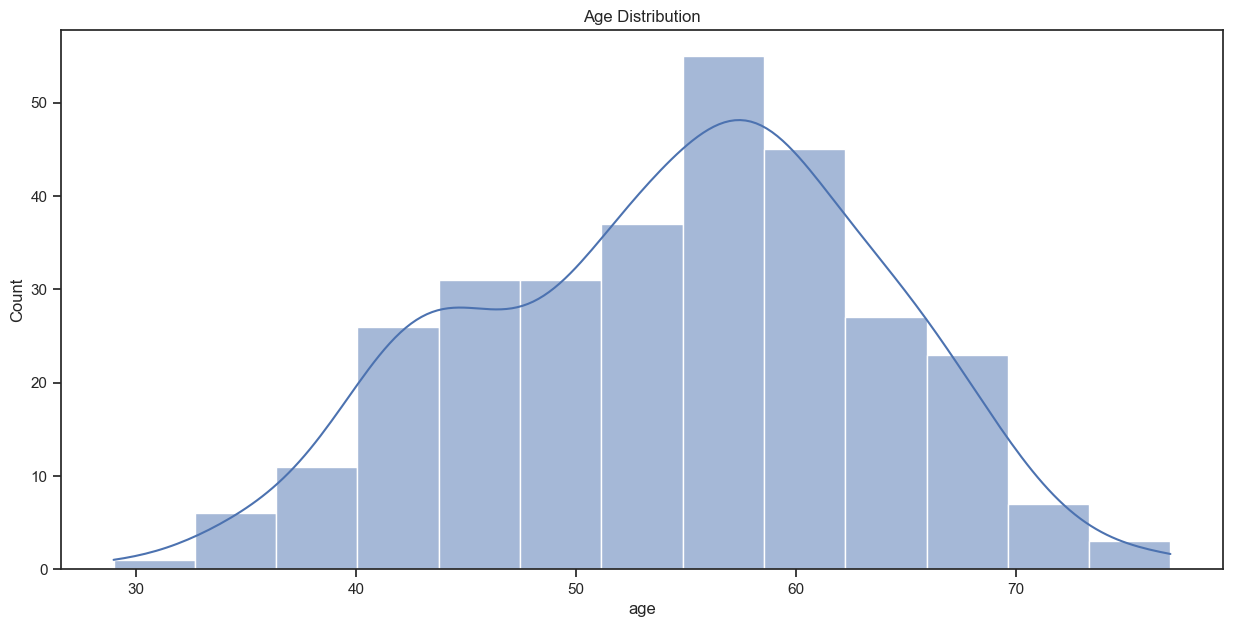
Exploratory Data Analysis (EDA) is a crucial step in understanding and summarizing the main characteristics of a dataset. Here are some steps you can take for EDA on heart disease dataset:

**1.Univariate Analysis:**

* **Distribution of Heart Disease Condition:**

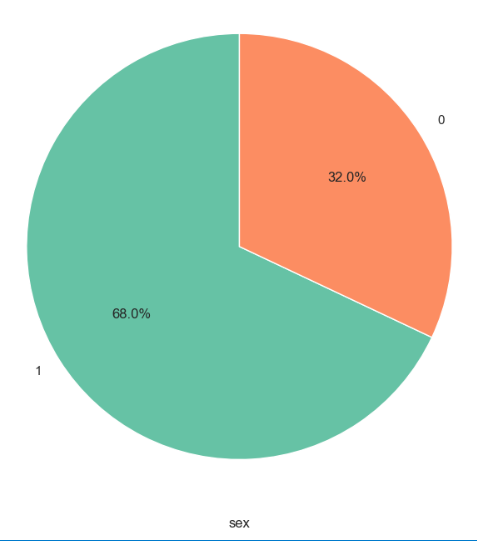
****

* **Without Heart Disease**: Approximately **54.1%** of the population does not have heart disease.
* **With Heart Disease**: About **45.9%** of the population has heart disease.
* **Age Distribution:**

****

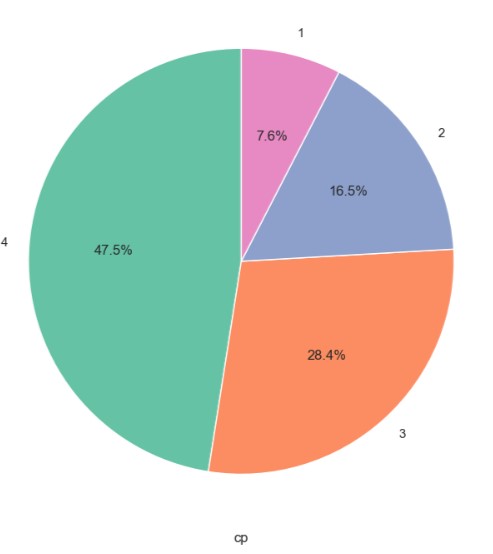
The x-axis represents age, ranging approximately from 30 to 70 years, and the y-axis represents the count of individuals within each age bin. The histogram bars show the frequency of individuals within specific age ranges, with the tallest bar around the 50-55 age range, indicating this is the most common age group in the dataset.

* **Distribution of sex:**



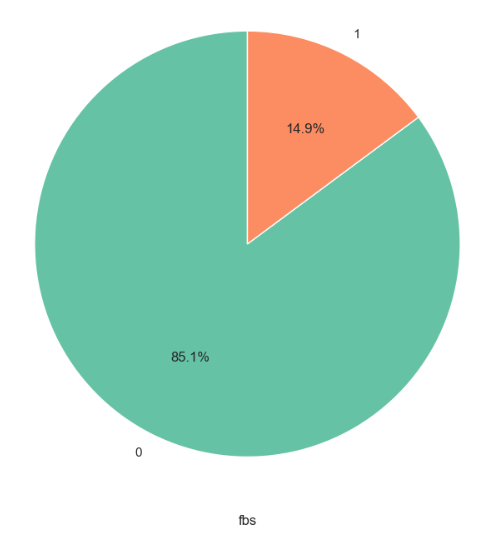
The pie chart shows the distribution of sex in a dataset, with 68% of the individuals being male (denoted by 1) and 32% being female (denoted by 0). This indicates a higher proportion of males in the dataset.

* **Distribution of the Chest pain type:**



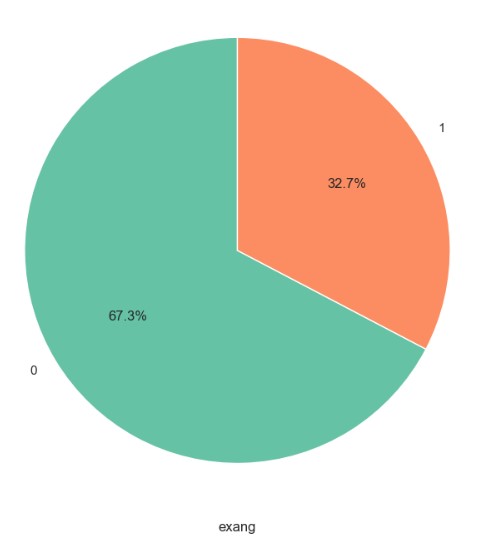
The pie chart represents the distribution of the 'cp' variable, which stands for chest pain type, in a dataset. The largest segment is type 4 (asymptomatic), making up 47.5% of the cases, followed by type 3 (non-anginal pain) at 28.4%, type 2 (atypical angina) at 16.5%, and the smallest being type 1 (typical angina) at 7.6%.

* **Distribution of Fasting Blood Sugar:**



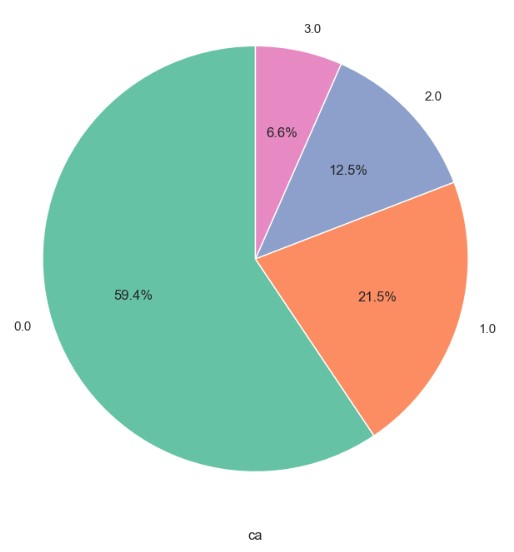
The pie chart shows the distribution of the 'fbs' variable, which stands for fasting blood sugar, in a dataset. The chart indicates that 85.1% of individuals have fasting blood sugar below 120 mg/dl (denoted by 0), and 14.9% have fasting blood sugar above 120 mg/dl (denoted by 1), suggesting that most of the individuals in this dataset do not have elevated fasting blood sugar levels.

* **Distribution of Exang (Exercise induced Angina):**



The chart is used to represent the distribution of a binary variable 'exang' in a dataset, likely related to heart disease prediction, given the context of the window title. The larger portion (67.3%) represents cases where 'exang' is 0, and the smaller portion (32.7%) represents cases where 'exang' is 1.

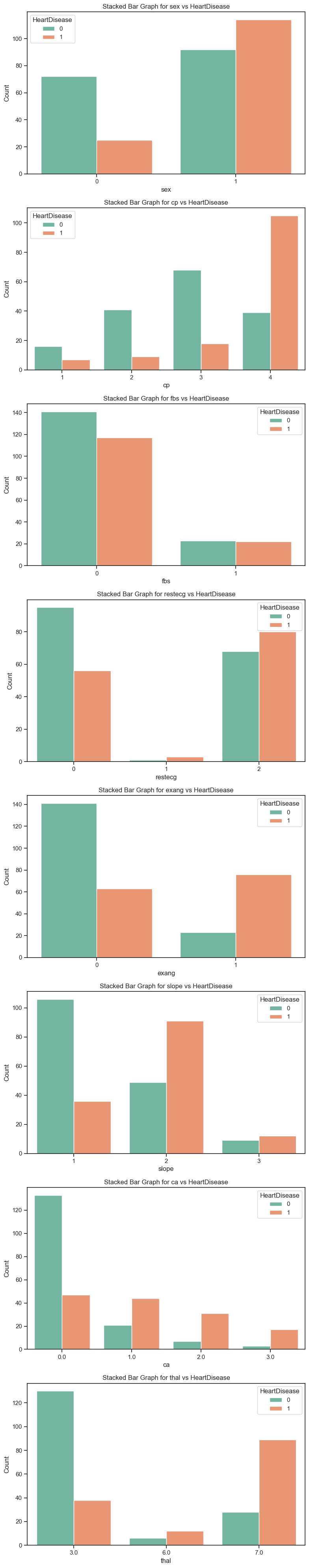
* **Distribution of ca:**



In this chart, where 'ca' could represent a categorical variable with four levels. The largest segment of the chart is for category 0.0, which constitutes just under three-fifths of the data.

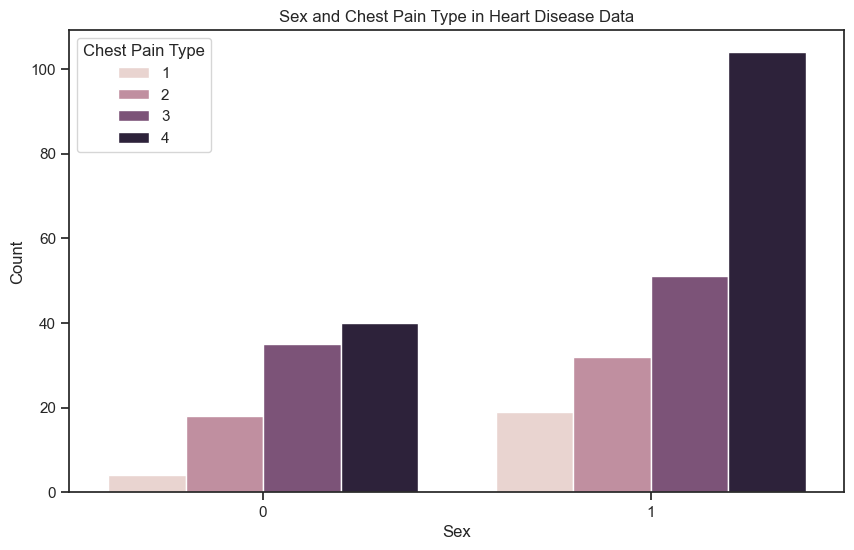
* **Bivariate Analysis:**

The image contains multiple stacked bar graphs, each representing the count of individuals with and without heart disease across different categorical variables. Here are the details for each graph:

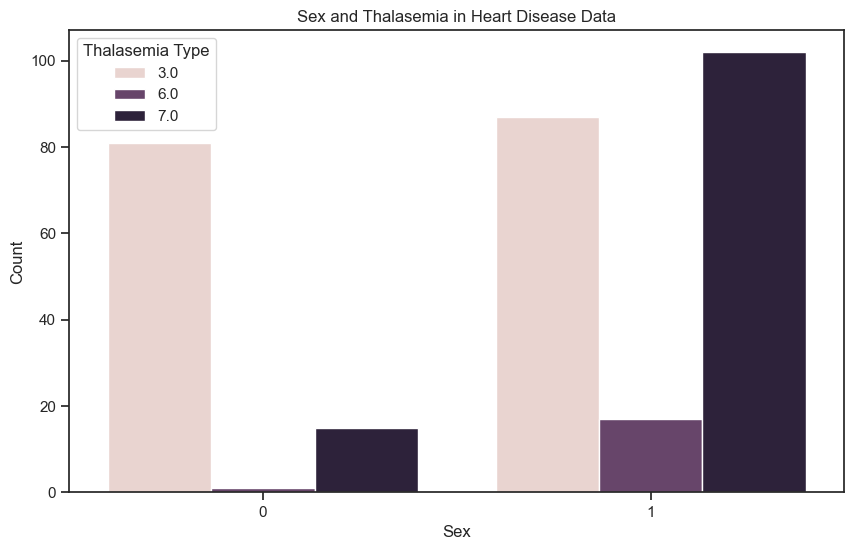


1. **Stacked Bar Graph for sex vs HeartDisease**:
   * The x-axis represents sex with two categories (0 and 1).
   * The y-axis represents the count of individuals.
   * Two colours represent the presence (1) or absence (0) of heart disease.
   * Category 0 has a higher count of individuals without heart disease than with it.
   * Category 1 has a higher count of individuals with heart disease than without it.
2. **Stacked Bar Graph for cp vs HeartDisease**:
   * The x-axis represents chest pain type (cp) with four categories (1, 2, 3, 4).
   * The y-axis represents the count of individuals.
   * Category 1 has a higher count of individuals without heart disease.
   * Categories 2 and 3 have a higher count of individuals with heart disease.
   * Category 4 has a relatively equal distribution of individuals with and without heart disease.
3. **Stacked Bar Graph for fbs vs HeartDisease:**
   * The x-axis represents fasting blood sugar (fbs) with two categories (0 and 1).
   * The y-axis represents the count of individuals.
   * Category 0 has a higher count of individuals without heart disease.
   * Category 1 has fewer individuals overall, with more individuals without heart disease than with it.
4. **Stacked Bar Graph for restecg vs HeartDisease:**
   * The x-axis represents resting electrocardiographic results (restecg) with three categories (0, 1, 2).
   * The y-axis represents the count of individuals.
   * Category 0 has a higher count of individuals without heart disease.
   * Category 1 has a higher count of individuals with heart disease.
   * Category 2 has a very low count of individuals, mostly without heart disease.
5. **Stacked Bar Graph for exang vs HeartDisease:**
   * The x-axis represents exercise-induced angina (exang) with two categories (0 and 1).
   * The y-axis represents the count of individuals.
   * Category 0 has a higher count of individuals without heart disease.
   * Category 1 has a higher count of individuals with heart disease.
6. **Stacked Bar Graph for slope vs HeartDisease:**
   * The x-axis represents the slope of the peak exercise ST segment (slope) with three categories (1, 2, 3).
   * The y-axis represents the count of individuals.
   * Category 1 has a higher count of individuals without heart disease.
   * Category 2 has a higher count of individuals with heart disease.
   * Category 3 has fewer individuals, with more having heart disease.
7. **Stacked Bar Graph for ca vs HeartDisease:**
   * The x-axis represents the number of major vessels colored by fluoroscopy (ca) with four categories (0.0, 1.0, 2.0, 3.0).
   * The y-axis represents the count of individuals.
   * Category 0.0 has a higher count of individuals without heart disease.
   * Categories 1.0, 2.0, and 3.0 have a higher count of individuals with heart disease.
8. **Stacked Bar Graph for thal vs HeartDisease:**
   * The x-axis represents thalassemia (thal) with four categories (3.0, 6.0, 7.0).
   * The y-axis represents the count of individuals.
   * Category 3.0 has a higher count of individuals without heart disease.
   * Category 7.0 has a higher count of individuals with heart disease.
   * Category 6.0 has a very low count of individuals, with more having heart disease.

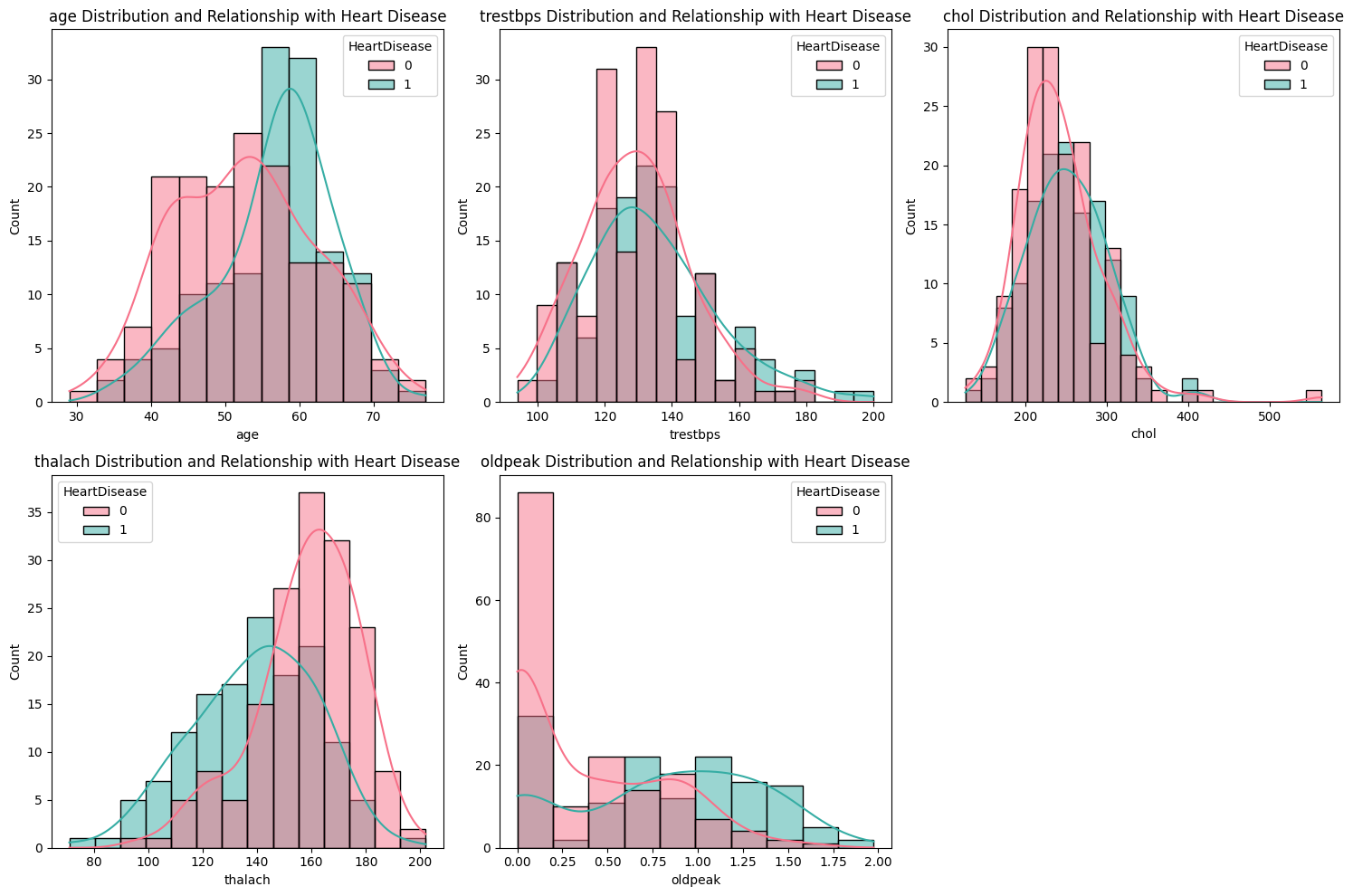
* **Bar Graph for 'Sex and Chest Pain Type:**



* This bar graph shows the count of individuals by sex (0 or 1) and chest pain type (1, 2, 3, 4).
* The x-axis represents sex, and the y-axis represents the count of individuals.
* There are four bars for each sex category, each corresponding to a different chest pain type.
* For sex category 0, the counts for chest pain types 1, 2, 3, and 4 are relatively lower compared to sex category 1.
* In sex category 1, chest pain type 4 is the most common, with the count significantly higher than the other types.
* **Bar Graph for Sex and Thalassemia:**



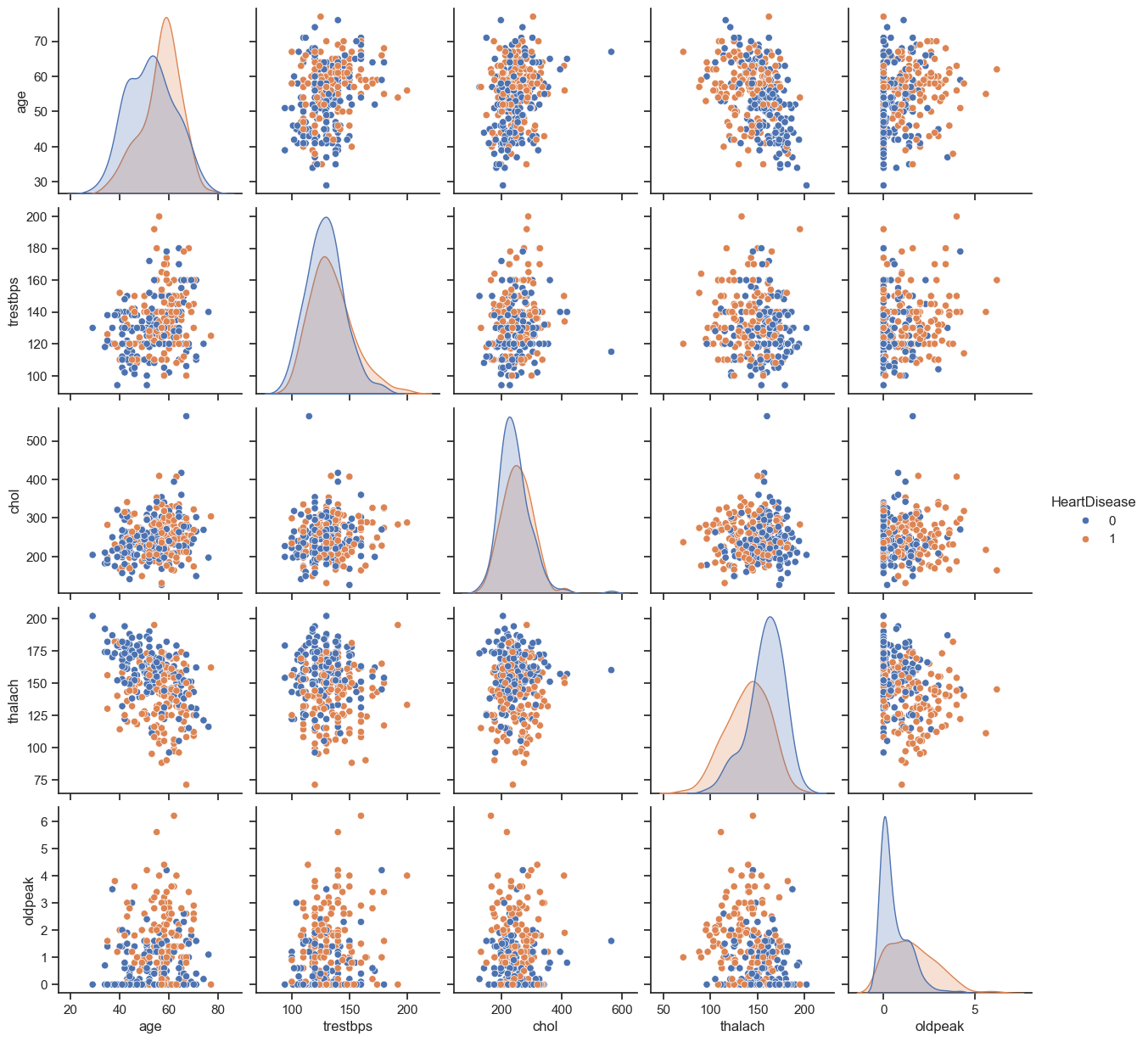
* There are three sets of bars for each sex category, corresponding to different types of thalassemia: 3.0, 6.0, and 7.0.
* For sex category 0, the count of individuals with thalassemia type 3.0 is the highest, followed by a much smaller count for types 6.0 and 7.0.
* For sex category 1, the count of individuals with thalassemia type 3.0 is also the highest, with type 7.0 being the second-highest and type 6.0 having the least count.
* The count of individuals with thalassemia type 3.0 is higher in sex category 1 compared to category 0, indicating a possible higher prevalence of this type of thalassemia in sex category 1.
* **Numerical Features Analysis:**



1. **Age Distribution and Relationship with Heart Disease:**
   * This plot shows the age distribution of individuals with and without heart disease. Two distinct colours represent the presence (1) or absence (0) of heart disease. It appears that people in their 50s and 60s are more likely to have heart disease.
2. **Trestbps Distribution and Relationship with Heart Disease:**
   * This plot illustrates the distribution of resting blood pressure (trestbps) among individuals. There is a notable overlap, indicating that resting blood pressure alone might not be a strong indicator of heart disease.
3. **Chol Distribution and Relationship with Heart Disease:**
   * The cholesterol level (chol) distribution is depicted here. Like trestbps, there is an overlap between individuals with and without heart disease, suggesting cholesterol levels also might not be a decisive factor for heart disease on its own.
4. **Thalach Distribution and Relationship with Heart Disease:**
   * This plot represents the maximum heart rate achieved (thalach) distribution. There seems to be a pattern where higher maximum heart rates are associated with the absence of heart disease.
5. **Oldpeak Distribution and Relationship with Heart Disease:**
   * Oldpeak refers to depression induced by exercise relative to rest. The plot indicates that lower oldpeak values are associated more frequently with the absence of heart disease.
6. **Slope Distribution and Relationship with Heart Disease:**
   * Slope refers to the slope of peak exercise ST segment, an ECG measure related to stress testing on hearts. Different categories indicate different risk levels associated with ST depression observed on exertion.

* **Multivariate Analysis:**

The pairplot visually represents the relationships between different health-related variables and heart disease.

****

In summary, age, maximum heart rate achieved, and ST depression appear to be significant factors in heart disease risk. However, resting blood pressure and serum cholesterol alone do not strongly predict heart disease.

As we can see oldpeak is right skewed So, we applied log transform to make it normal distributed. The code is:

# Apply log transformation

df['oldpeak'] = np.log1p(df['oldpeak'])

**Model Building and Evaluation**

We prepared the data for machine learning by splitting it into features (**X**) and the target variable (**Y**). The subsequent steps involve training a predictive model, such as logistic regression or decision trees, on the training set, making predictions on the testing set, and evaluating the model's performance. This iterative process allows for the optimization of the model to predict heart disease based on the dataset's features. The choice of algorithm and continuous refinement are crucial in building an effective and accurate predictive model.

X = df. drop(columns=['HeartDisease'], axis=1)

Y = df['HeartDisease']

We efficiently splits the dataset into training and testing sets using the train\_test\_split function, allocating 80% of the data for training (X\_train and y\_train) and 20% for testing (X\_test and y\_test).

X\_train, X\_test, y\_train, y\_test = train\_test\_split (X, Y, test\_size=0.2, random\_state=0)

The **StandardScaler** is a preprocessing technique used in machine learning to standardize the features of a dataset. Standardization involves transforming the data such that it has a mean of 0 and a standard deviation of 1.

scaler = StandardScaler ()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

X\_train\_scaled = pd. DataFrame (X\_train\_scaled, columns=X\_train. columns)

X\_test\_scaled = pd. DataFrame (X\_test\_scaled, columns=X\_test. Columns)

* **Logistic Regression Model:**

We checked the performance of the logistic regression model on both the training and testing data, helping to assess its effectiveness in classifying heart disease.

**Accuracy on training Data => 0.8677**

**Accuracy on test Data => 0.7868**

**Classification Report:**

**precision recall f1-score support**

**0 0.78 0.89 0.83 35**

**1 0.81 0.65 0.72 26**

**accuracy 0.79 61**

**macro avg 0.79 0.77 0.78 61**

**weighted avg 0.79 0.79 0.78 61**

The model achieved an accuracy of 86.77% on the training data and 78.68% on the test data. It performed well in classifying class 0 with 78% precision and 89% recall, while class 1 had 81% precision and 65% recall. The overall performance is satisfactory, with a weighted average F1-score of 0.78 on the test data.Top of Form

* **Hyperparameter Tuning on Logistic Regression Model:**

**Best Hyperparameters: {'C': 0.1, 'penalty': 'l2'}**

**Accuracy on training Data => 0.8636**

**Accuracy on test Data => 0.8196**

**Classification Report:**

**precision recall f1-score support**

**0 0.80 0.91 0.85 35**

**1 0.86 0.69 0.77 26**

**accuracy 0.82 61**

**macro avg 0.83 0.80 0.81 61**

**weighted avg 0.82 0.82 0.82 61**

The model, trained with hyperparameters {'C': 0.1, 'penalty': 'l2'}, achieved an accuracy of 86.36% on the training data and 81.97% on the test data. It performed well in classifying both classes, with precision of 80% for class 0 and 86% for class 1. The model demonstrated good balance between precision and recall, resulting in a weighted average F1-score of 0.82 on the test data. Overall, the hyperparameter-tuned model shows improved performance on the test set compared to the previous model.

* **Support Vector Classifier (SVC):**

We checked the performance of the SVC model on both the training and testing data, helping to assess its effectiveness in classifying heart disease.

**Accuracy on training Data => 0.9256**

**Accuracy on test Data => 0.8032**

**Classification Report:**

**precision recall f1-score support**

**0 0.78 0.91 0.84 35**

**1 0.85 0.65 0.74 26**

**accuracy 0.80 61**

**macro avg 0.82 0.78 0.79 61**

**weighted avg 0.81 0.80 0.80 61**

The model, with an accuracy of 92.56% on the training data and 80.32% on the test data, demonstrates strong generalization. It shows good precision and recall for both classes, with a weighted average F1-score of 0.80 on the test set. The model strikes a balance between classifying class 0 (precision of 78% and recall of 91%) and class 1 (precision of 85% and recall of 65%). Overall, it performs well on both training and test sets.

* **Hyperparameter Tuning on SVC model:**

**Best Hyperparameters: {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}**

**Accuracy on training Data => 0.9256**

**Accuracy on test Data => 0.8032**

**Classification Report:**

**precision recall f1-score support**

**0 0.78 0.91 0.84 35**

**1 0.85 0.65 0.74 26**

**accuracy 0.80 61**

**macro avg 0.82 0.78 0.79 61**

**weighted avg 0.81 0.80 0.80 61**

The model, trained with hyperparameters {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}, achieved an accuracy of 92.56% on the training data and 80.33% on the test data. It shows consistent performance with the previous model, maintaining good precision and recall for both classes. The hyperparameter tuning did not significantly impact the model's performance, suggesting that the initial set of hyperparameters already provided an effective configuration for this dataset.

* + Top of Form
* **RANDOM FOREST MODEL:**

We checked the performance of the Random Forest model with on both the training and testing data, helping to assess its effectiveness in classifying heart disease.

**Best Hyperparameters: {'max\_depth': 10, 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 50}**

**Accuracy on training Data => 0.9214**

**Accuracy on test Data => 0.8032**

**Classification Report:**

**precision recall f1-score support**

**0 0.78 0.91 0.84 35**

**1 0.85 0.65 0.74 26**

**accuracy 0.80 61**

**macro avg 0.82 0.78 0.79 61**

**weighted avg 0.81 0.80 0.80 61**

The model, trained with hyperparameters {'max\_depth': 10, 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 50}, achieved an accuracy of 92.15% on the training data and 80.33% on the test data. It shows consistent performance with the previous models, maintaining good precision and recall for both classes. The hyperparameter tuning did not significantly impact the model's performance, and the initial set of hyperparameters already provided an effective configuration for this dataset.

**Comparison of models**

Let's compare the performance of the Logistic Regression, Support Vector Classifier (SVC), and Random Forest models, both in their initial form and after hyperparameter tuning:

**1. Logistic Regression:**

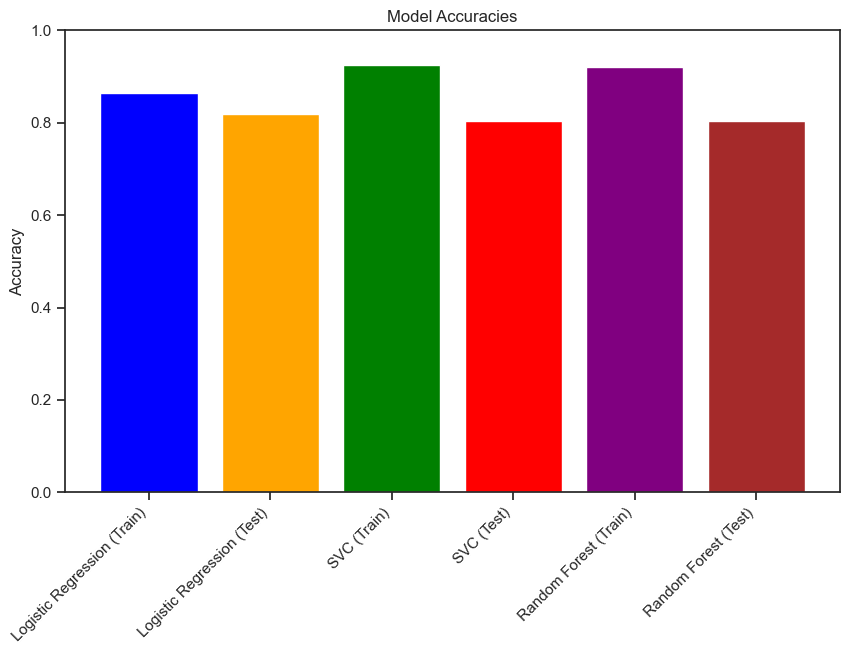
* Initial Model:
  + Accuracy (Test): 78.68%
  + Weighted Avg F1-score: 0.78
* Hyperparameter-Tuned Model:
  + Accuracy (Test): 81.97%
  + Weighted Avg F1-score: 0.82
* **Conclusion:** Hyperparameter tuning improved the model's performance, resulting in higher accuracy and F1-score.

**2. Support Vector Classifier (SVC):**

* Initial Model:
  + Accuracy (Test): 80.32%
  + Weighted Avg F1-score: 0.80
* Hyperparameter-Tuned Model:
  + Accuracy (Test): 80.33%
  + Weighted Avg F1-score: 0.80
* **Conclusion:** Hyperparameter tuning did not significantly impact the performance, suggesting that the initial configuration was effective.

**3. Random Forest:**

* Initial Model:
  + Accuracy (Test): 80.33%
  + Weighted Avg F1-score: 0.80
* Hyperparameter-Tuned Model:
  + Accuracy (Test): 80.33%
  + Weighted Avg F1-score: 0.80
* **Conclusion:** Hyperparameter tuning did not significantly impact the performance, and the initial set of hyperparameters was already effective.



**Overall Conclusion**

* The Logistic Regression model benefited from hyperparameter tuning, showing improved accuracy and F1-score on the test set.
* Support Vector Classifier (SVC) maintained strong performance, and hyperparameter tuning did not result in a significant difference.
* Random Forest showed consistent performance, with hyperparameter tuning having minimal impact.
* Considering these results, the choice of the best model may depend on factors such as interpretability, computational complexity, and specific requirements of the problem at hand. The Logistic Regression model, after hyperparameter tuning, may be a favourable choice due to its improved performance.

**Reference**

1. <https://archive.ics.uci.edu/dataset/45/heart+disease>
2. <https://www.kaggle.com/code/ahmadpk/heart-disease-classification>
3. <https://towardsdatascience.com/heart-disease-uci-diagnosis-prediction-b1943ee835a7>
4. <https://www.kaggle.com/code/desalegngeb/heart-disease-predictions>