**Project Overview:**

The ultimate objective of the research is to create a machine-learning model that can forecast a person's risk of heart disease based on various demographic and health characteristics. Heart disease is one of the main causes of death globally, and early identification and risk assessment are vital parts of preventive medicine. To create a prediction model, this study will make use of a dataset that includes details on people's age, blood pressure, cholesterol, ECG readings, exercise-induced angina, and other essential characteristics.

**This project's principal goals are:**

- ★ Construct a machine learning model that can precisely forecast a person's risk of developing heart disease based on their health and personal characteristics.

- ★ *Evaluate the main predictor*s: Find out which characteristics most significantly affect the risk of heart disease, giving patients and medical professionals valuable information.

- ★ *Give practical insights*: Provide helpful guidance on how people and healthcare professionals can reduce their risk of heart disease.

- ★ *Analyse model performance*: To make sure the model is effective in real-world situations, consider measures like accuracy, precision, and recall.

**Data Sources:**
"Kaggle" is the largest data science community in the world, offering a wealth of resources and strong tools. This is where the dataset for this research was obtained. In the actual world, hospitals, clinics, and public health groups may provide you with this kind of information.

**Data Format:**
The dataset appears to be organised as a Pandas DataFrame and is tabular. It has *12 columns (features) and 918 rows (samples)*. Rows and columns seem to be used to organize the data, which makes it appropriate for analysis and machine learning.

**Preprocessing Steps:**
Preparing the data for analysis and modeling is a crucial step. Typical preprocessing actions consist of:

- ★ ***Missing Data Handling:*** Examine the dataset for any missing values, then determine how to handle them. Imputation, which involves substituting the column's mean, median, or mode for missing values, and the removal of rows with missing values are common techniques.

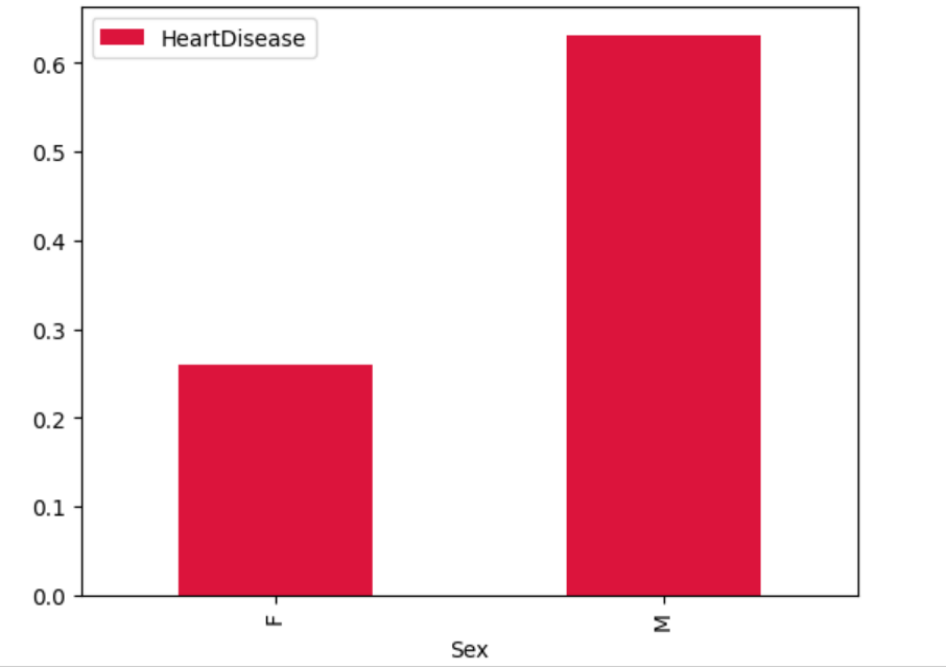
- ★ ***Data cleaning:*** Check the data for any flaws or discrepancies. This could entail addressing outliers, making sure data types are suitable, and fixing typos.

- ★ ***Encrypting Categorical Variables:*** To be utilized in machine learning models, categorical variables like "Sex" or "ChestPainType" might need to be one-hot encoded. Some columns in the collection seem to have already been encoded based on their names.

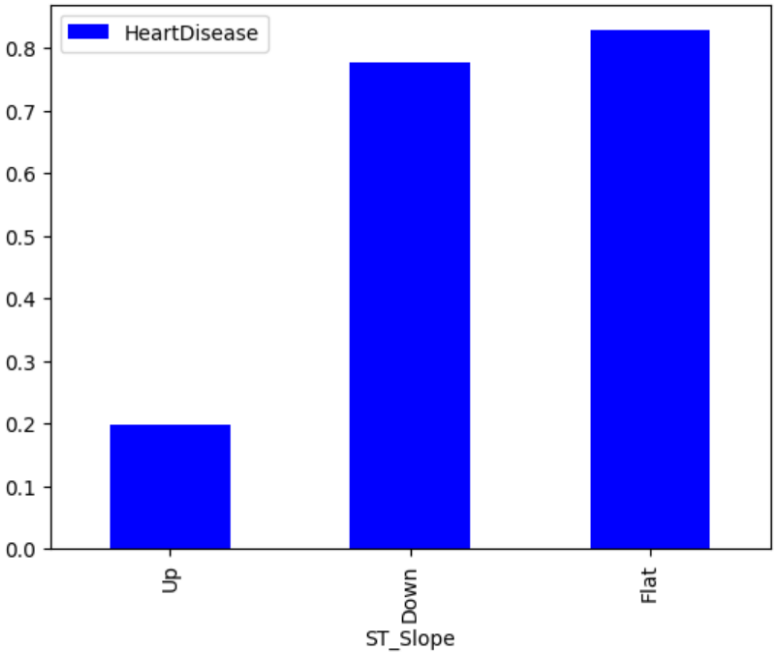
## – EXPLORATORY DATA ANALYSIS –

**Variable Description**

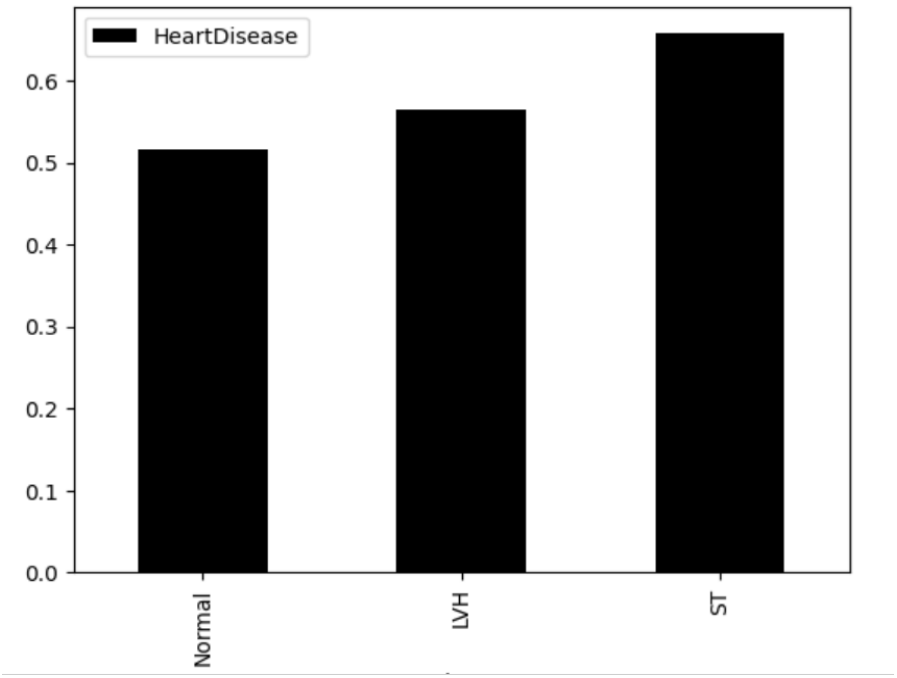
- ★ **Age**: Age of the patient [Years]
- ★ **Sex**: Sex of the patient [M: Male, F: Female]
- ★ **ChestPainType**: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
- ★ **RestingBP**: Resting blood pressure [mm Hg]
- ★ **Cholesterol**C: Serum cholesterol [mm/dl]
- ★ **FastingBS**: Fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
- ★ **RestingECG**: Resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
- ★ **MaxHR**: Maximum heart rate achieved [Numeric value between 60 and 202]
- ★ **ExerciseAngina**: Exercise-induced angina [Y: Yes, N: No]
- ★ **Oldpeak**: Oldpeak = ST [Numeric value measured in depression]
- ★ **ST_Slope**: The slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
- ★ **HeartDisease**: Output class [2: heart disease, 1: Normal]



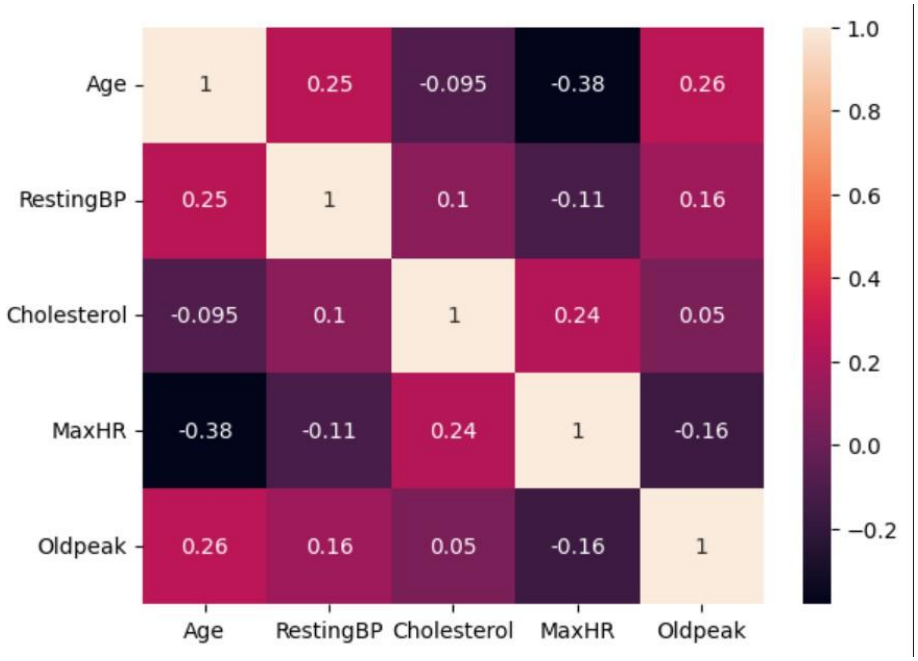

In examining the dataset, the bar plot highlights a notable difference in the average prevalence of heart disease between genders. Males exhibit a higher average prevalence compared to females. This disparity underscores the importance of gender-specific considerations in cardiovascular health.



The bar plot indicates that when the heart's ST segment goes down, there's a higher chance of heart disease (75%), while an upward or flat slope corresponds to lower chances (10% and 80%, respectively). This suggests a distinct association between the ST segment slope and the likelihood of heart disease.

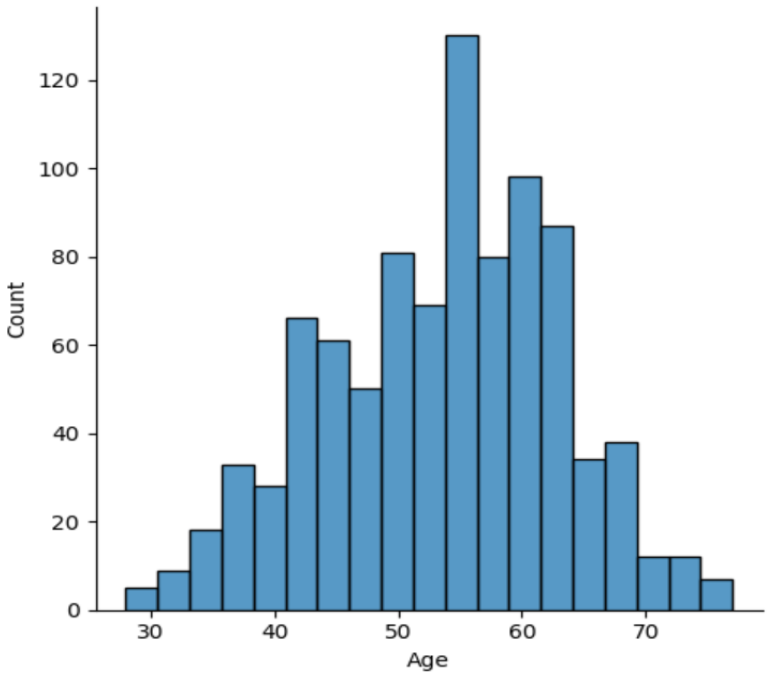



The bar plot reveals average heart disease prevalence across different resting electrocardiographic results ('RestingECG'). Normal results show a moderate prevalence of around 0.5, while configurations indicating Left Ventricular Hypertrophy (LVH) exhibit a slightly higher prevalence. Additionally, ST-segment irregularities ('STi') show a higher prevalence compared to both normal and LVH cases. These findings suggest potential associations between specific ECG patterns and heart disease risk, emphasizing the importance of detailed ECG analysis in cardiovascular health assessment.

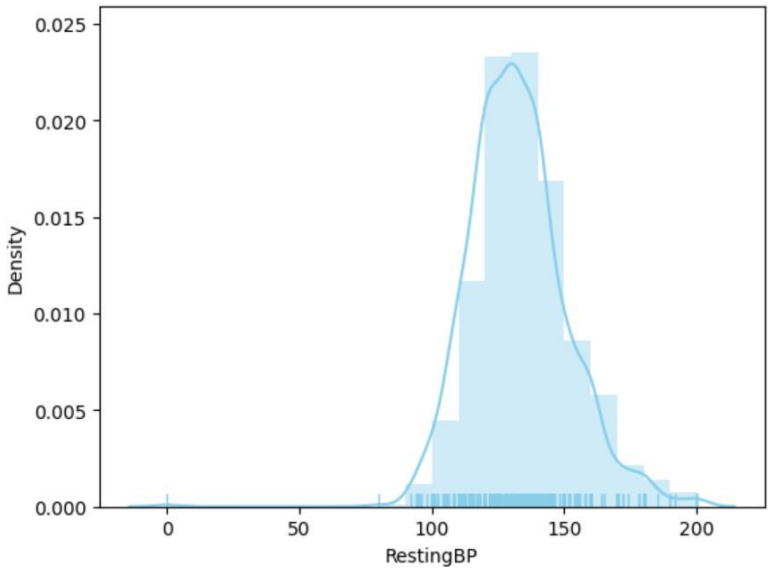



The heatmap displays the correlation matrix for selected cardiovascular health-related features, including 'Age,' 'Resting Blood Pressure (RestingBP),' 'Cholesterol,' 'Maximum Heart Rate (MaxHR),' and 'Oldpeak.' Darker colors represent stronger correlations. This visualization helps identify potential relationships between these variables, providing insights into how changes in one feature may relate to changes in another within the context of cardiovascular health

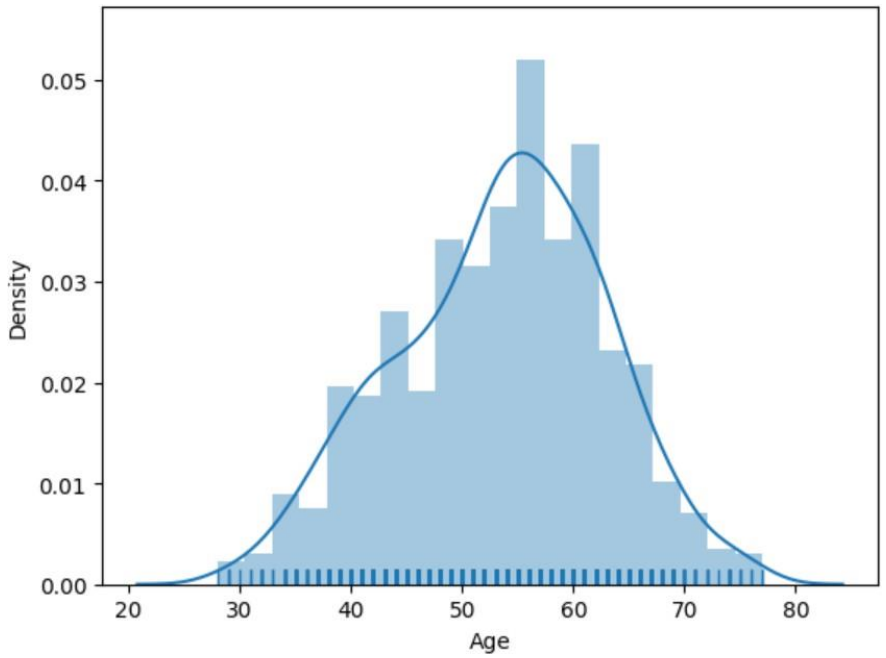
.



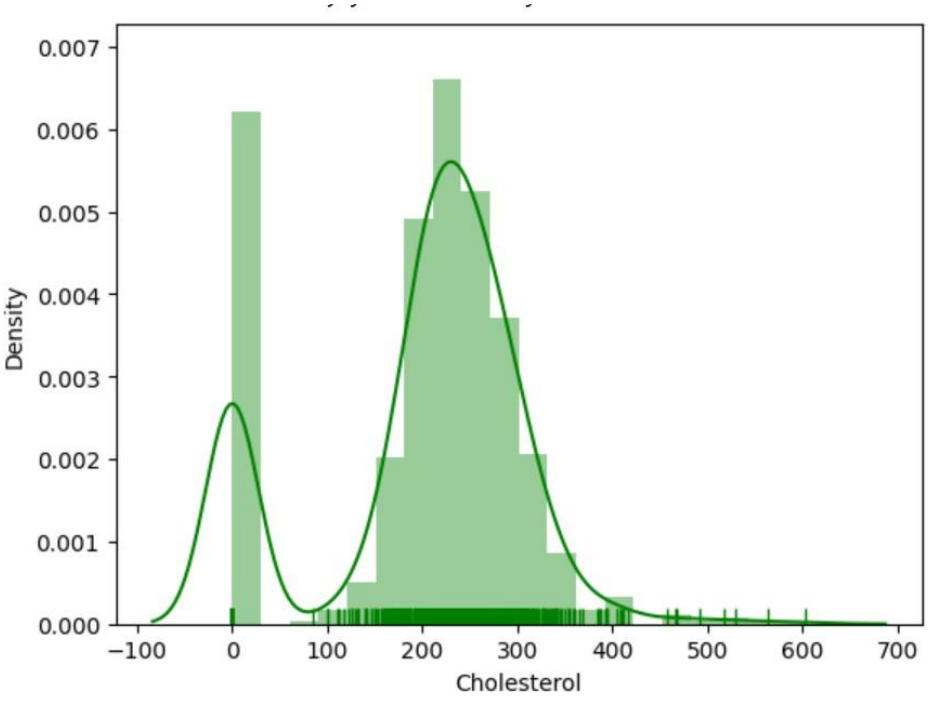
The distribution plot depicts the spread of ages in the dataset. It helps in understanding the frequency of different age groups, providing a visual overview of the age distribution and potential insights into the demographic composition of the data.



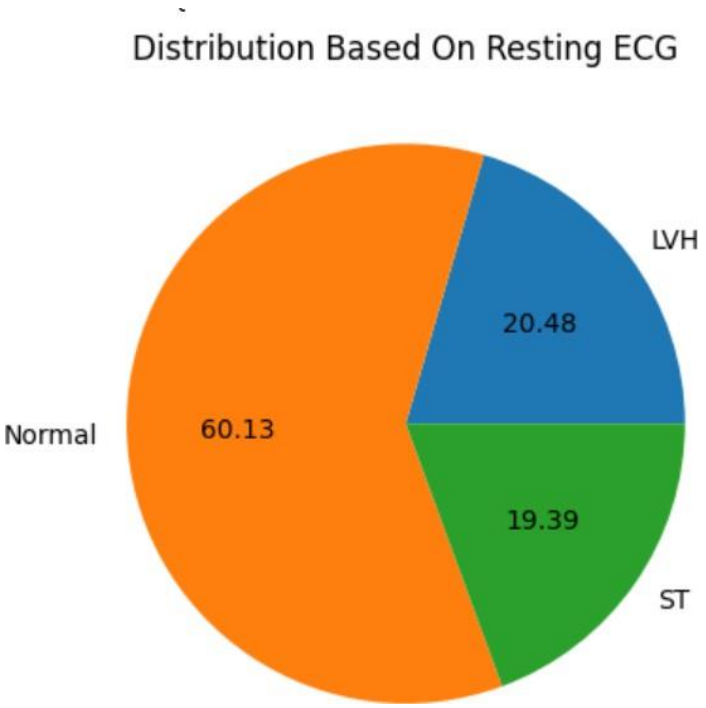

The distribution plot for Resting Blood Pressure ('RestingBP') showcases the overall pattern of blood pressure values in the dataset. The histogram provides a frequency distribution, while the kernel density estimate offers a smooth representation of the probability density. The rug plot displays individual data points along the axis, enhancing the understanding of data density at specific values. This plot aids in visualizing the distribution and potential insights into blood pressure patterns.



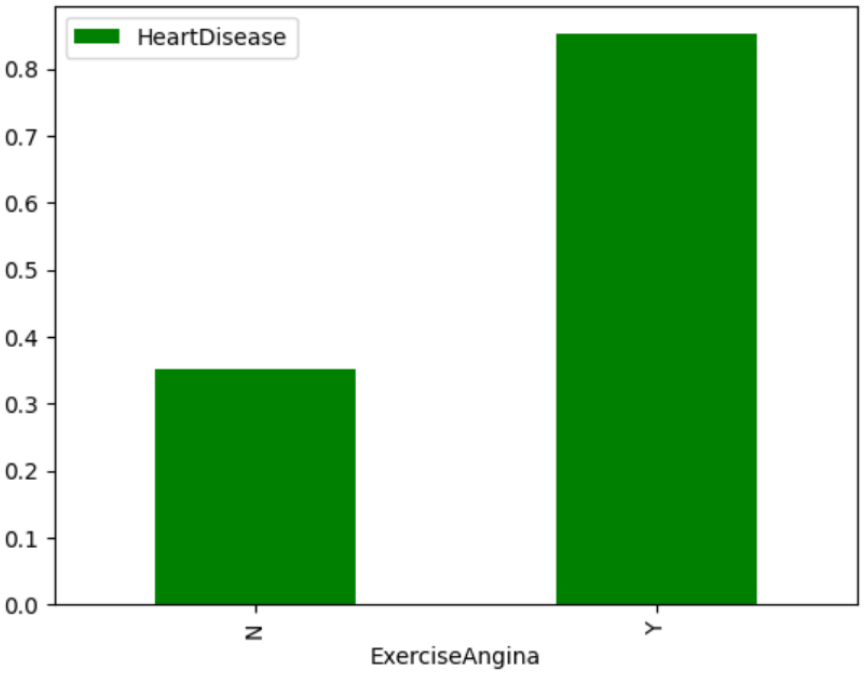

The distribution plot highlights a concentration of ages between 50 to 60, indicating a peak in the density of individuals within this age range. This observation provides valuable insights into the demographic composition of the dataset, emphasizing a significant presence of individuals in the 50 to 60 age group



The distribution plot for 'Cholesterol' reveals interesting patterns. Particularly, there's a peak around 0 cholesterol, with a density of around 0.00275, followed by a decrease until approximately 100 cholesterol. Subsequently, density rises again, reaching its highest point around 250 cholesterol, and then gradually decreases. These observations offer valuable insights into the distribution patterns and concentrations of cholesterol levels within the dataset, potentially indicating distinct subgroups or trends in cholesterol distribution.

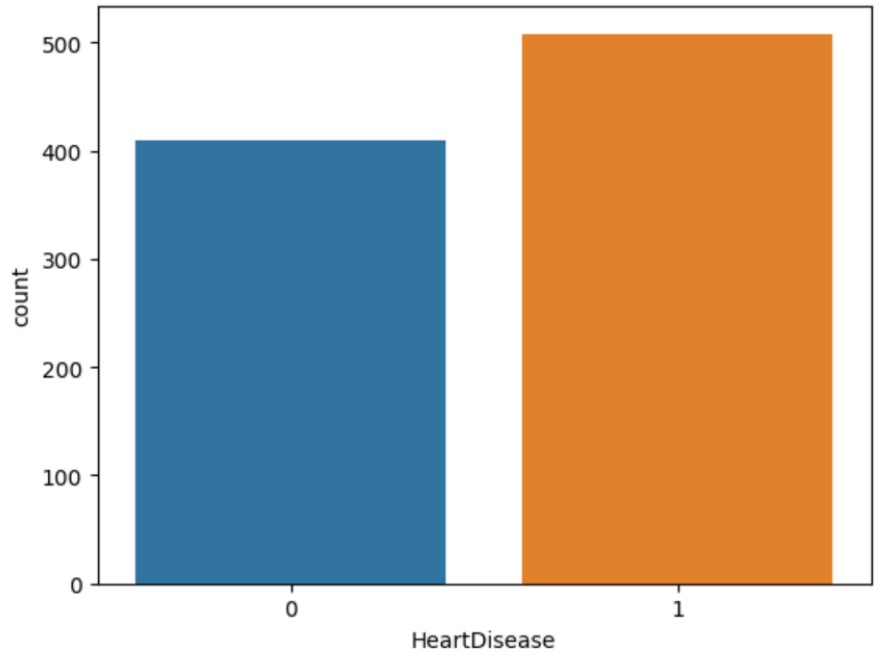



The pie chart depicting the distribution based on 'RestingECG' reveals distinct prevalence percentages. The 'Normal' category constitutes the majority at 60.23%, followed by 'LVH' (Left Ventricular Hypertrophy) at 20.48%, and 'ST' (ST-T wave abnormality) at 19.39%. This breakdown provides a clear understanding of the relative distribution of different resting electrocardiographic results in the dataset, with a notable prominence of normal ECG configurations.

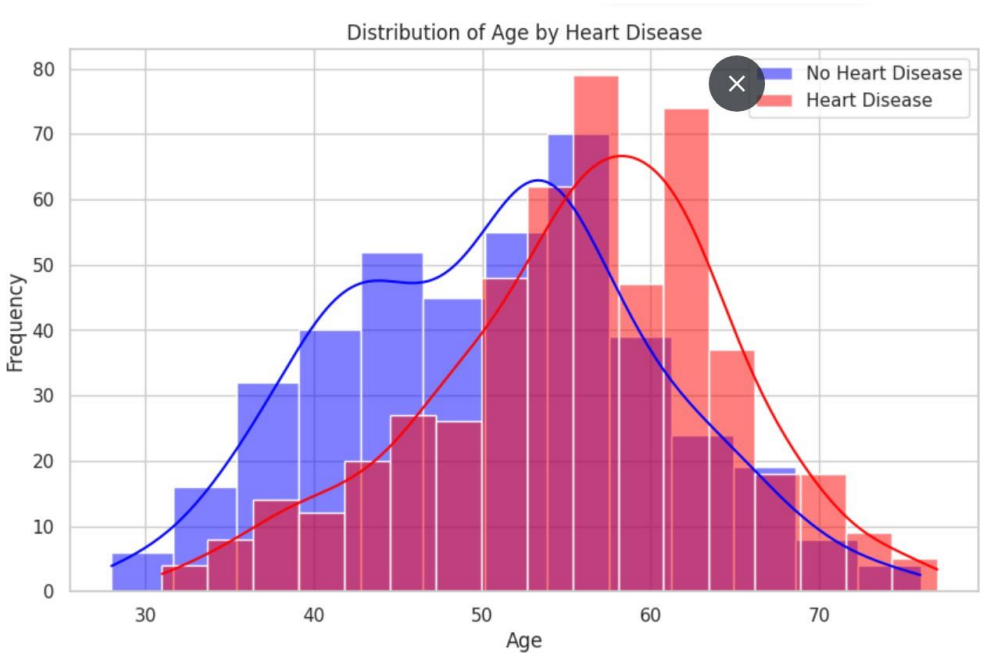



The bar plot reveals distinct average prevalence values of heart disease based on the presence or absence of exercise-induced angina ('ExerciseAngina'). Individuals without exercise-induced angina ('N') exhibit an average prevalence of 0.35, while those with exercise-induced angina ('Y') show a significantly higher average prevalence of around 0.8. This stark difference suggests that the presence of angina during exercise may be associated with a higher average prevalence of heart disease, emphasizing the potential importance of monitoring and managing cardiovascular health in individuals experiencing exercise-induced angina.

```
1    55.337691
0    44.662309
Name: HeartDisease, dtype: float64
```




Approximately 55.34% of individuals in the dataset have heart disease (1).
Around 44.66% of individuals in the dataset do not have heart disease (0).



The histogram depicting the distribution of age by heart disease status reveals a notable pattern. The red bars, representing individuals with heart disease, are notably higher in the age range of 55 to 65. This concentration suggests a potential correlation between this age group and an increased prevalence of heart disease within the dataset. Understanding such age-related patterns is crucial for targeted health interventions and risk assessment in cardiovascular health

**Analysis Methods:**
We employed a Decision Tree Classifier, Train-Test Split, and Random Forest Classifier. These are methods and models that are frequently used for categorization tasks such as risk prediction for heart disease. An outline of each of these elements is provided below:

- ★ **Random Forest Classifier**:
  Type: *Ensemble Classification*

  Explanation: Using a combination of several decision trees, the Random Forest Classifier is an ensemble learning technique that produces predictions. It is renowned for its capacity to manage intricate data interactions, lessen overfitting, and enhance prediction accuracy. A random collection of characteristics and a random sample of data are used to construct each decision tree in the forest. Multiple trees' results can be used to produce a reliable and accurate prediction.

- ★ **Train-Test Split Algorithm**:
  Type: *Data Splitting*

  Explanation: A data-splitting method called the train-test split is used to evaluate a machine learning model's performance. The process entails splitting the dataset into a training set and a testing set. The testing set is used to assess the model's performance, whereas the training set is used to train the model. This method avoids overfitting and aids in evaluating how well the model generalizes to new data.

- ★ **Decision Tree Classifier**:
  Type: *Classification*

  Explanation: For classification tasks, the Decision Tree Classifier offers a straightforward and easily comprehensible model. To categorize data, it builds a tree-like structure of decision rules depending on the features. Decision trees are useful for identifying non-linear relationships in the data and comprehending them.

**Accuracy and Prediction:**

★ **Algorithm using Train and Test Split and Printing the Accuracy Score:**

*Fasting Blood Sugar Level Accuracy (0.7974)*:

An accuracy of approximately 79.74% suggests that the model can correctly predict the presence or absence of heart disease based on the fasting blood sugar level. This means that for this specific feature, the model correctly classifies individuals as having or not having heart disease in about 79.74% of cases.However, it's important to consider other evaluation metrics, such as precision, recall, and F1 score, to get a more complete picture of the model's performance for this specific feature.

*Accuracy for Slope of the ST Segment on ECG (0.9331)*:

An accuracy of approximately 93.31% suggests that the model is very accurate in predicting the presence or absence of heart disease based on the slope of the ST segment on the electrocardiogram (ECG). This indicates that this feature is a strong predictor of heart disease.A high accuracy value like this suggests that the slope of the ST segment is a significant indicator of heart disease, and the model can correctly classify individuals based on this feature in the majority of cases.

*Presence and Absence of Heart Disease Accuracy (0.6191)*:

An accuracy of approximately 61.91% suggests that the model's overall performance in predicting heart disease, considering all available features, is moderate.This accuracy indicates that the model can correctly classify individuals with heart disease and without heart disease in about 61.91% of cases. It's an essential metric, but other evaluation measures should be considered to gain a more comprehensive understanding of the model's performance, especially if the dataset is imbalanced.

★ **Use a Decision Tree Classifier and predict the new input:**

*For Chest Pain and Age:*

A Decision Tree Classifier is used in the code to forecast the forms of chest discomfort based on an individual's age. 'Age' and 'ChestPainType' are used to build two data frames, `x} (the independent variable) and {y} (the target variable), respectively. These data are used to train the Decision Tree Classifier. Next, a fresh data frame called `new_df} is created, containing an age value of 10. A prediction is created using the trained classifier, which indicates that the 'ChestPainType' is expected to be (Atypical Angina) 'ATA' for an age of 10. This code demonstrates how the Decision Tree Classifier may use a single feature—in this case, age—to predict categorical outcomes, such as the type of chest discomfort.

*For Exercise Angina and Age:*

A Decision Tree Classifier is used in the code to forecast the forms of Exercise Angina based on an individual's age. 'Age' and '*ExerciseAngina*' are used to build two data frames, `x} (the independent variable) and {y} (the target variable), respectively. These data are used to train the Decision Tree Classifier. Next, a fresh data frame called `new_df} is created, containing an age value of 70. A prediction is created using the trained classifier, which indicates that the '*ExerciseAngina*' is expected to be (Present/Yes) 'Y' for an age of 70. This code demonstrates how the Decision Tree Classifier may use a single feature—in this case, age—to predict

categorical outcomes, such as whether or not the person might face discomfort during physical activity or exercise.

- ★ **Predicted vs Actual graph:**
  The scatter plot exhibits a pattern where data points tend to cluster within the age range of 50 to 70, while the trendline shows an upward trajectory as age increases. This pattern suggests a potential association between age (’Age’) and resting blood pressure ('RestingBP'). Specifically, it implies that as individuals advance in age, there is a tendency for their resting blood pressure to rise.

**Model Evaluation**:
To determine how well machine learning models will function in practical applications, it is necessary to assess their performance. The following outlines the metrics we used to make your Random Forest Classifier and Decision Tree Classifier models to evaluate their performance:

*Metrics Used*:
A variety of evaluation metrics are frequently employed to evaluate model performance in classification jobs. These indicators offer a thorough picture of the model's performance. Important metrics consist of:

- ★ *Accuracy*: This statistic, the ratio of correctly classified examples to all instances in the dataset, assesses how accurate the model's predictions are overall. However, when there is an imbalance in the classes, accuracy can be deceiving.

**Result and Interpretation:**
*O*ur EDA has shed important light on the dataset, emphasizing the importance of age, gender, ST segment slope, ECG patterns, and heart disease risk assessment. These results can direct the creation of models for prediction and provide information for focused actions aimed at improving cardiovascular health. These insights can be used in future research and model development to create precise and useful cardiac disease prediction models.
An analysis of the main results and insights from your EDA is provided below:

- ★ *The Gender Gap:*
  According to the EDA, there is a major gender gap in the average frequency of cardiac disease, with men showing a greater average frequency. This shows that cardiovascular health is significantly influenced by gender, with men possibly being at higher risk. To better understand the underlying causes of this gender gap, more research is necessary.

- ★ *Heart Disease and ST Segment*:
  The EDA shows a clear correlation between the probability of heart disease and the position of the ST segment. An upward or flat slope is linked to a lower likelihood of heart disease (10% and 80%, respectively), whereas a downward ST segment is linked to a 75% chance. According to this research, the ST segment slope is a vital

sign of the likelihood of developing heart disease and can help with diagnosis and risk assessment.

★ *Heart Disease and ECG Patterns*:
According to the analysis of resting electrocardiographic data, or "RestingECG," different ECG patterns relate to different frequency levels of heart disease. This emphasizes how crucial it is to perform detailed ECG analyses when examining cardiovascular health. Heart disease may be significantly predicted by the existence of left ventricular hypertrophy (LVH) and ST-segment abnormalities ('STI').

★ *The Correlations of Features*:
Potential links between specific variables (e.g., age, blood pressure, cholesterol, maximum heart rate, and oldpeak) can be found in the heatmap illustrating feature correlations. Predicting model development and feature selection can be influenced by these observations. For instance, a significant positive association between blood pressure and age may indicate that blood pressure is influenced by age.

★ *Age Distribution*:
A peak in the density of individuals within this age range is indicated by the age distribution plot, which displays a concentration of people between the ages of 50 and 60. This shows that a sizable portion of the dataset is made up of people in the 50–60 age range, which may be a crucial population for interventions and assessments related to cardiovascular health.

★ *Blood Pressure and Cholesterol Patterns*:
Different patterns in the data are shown by the distribution plots for cholesterol and blood pressure at rest. Recognizing these trends can aid in the identification of possible subgroups with particular traits related to cholesterol and blood pressure. The presence of discrete groups with different cholesterol profiles, for instance, may be shown by the peaks and valleys in cholesterol levels.

★ *RestingECG Distribution*:
The relative frequency of various ECG configurations in the dataset is depicted in the pie chart that displays the distribution of resting electrocardiographic results. While most cases are classified as "Normal," the existence of "LVH" and "ST" configurations highlights the need to take these ECG patterns into account when assessing the risk of heart disease.

★ *Exercise-Induced Angina*:
Depending on whether exercise-induced angina is present or absent, the bar plot shows a significant variation in the average prevalence of heart disease. People who have exercise-induced angina ('Y') have an average prevalence of heart disease that is much greater. This result emphasizes how crucial it is to keep an eye on and manage cardiovascular health in people who have angina brought on by activity.

★ *Heart Disease Prevalence*:
The dataset comprises 44.66% persons without heart disease and about 55.34% individuals with heart illness. This represents the prevalence of heart disease. For the

purpose of developing and assessing machine learning models, this equilibrium between the two classes is crucial.

★ *Age and Heart Disease*:
The age range of 55 to 65 is where the histogram shows a concentration of people with heart disease. This finding is important for risk assessment and therapies specific to this population since it implies a relationship between this age group and a higher incidence of heart disease.

**Limitations:**

★ *Complexity and Heterogeneity*: Heart disease is a broad term that encompasses various conditions affecting the heart and blood vessels. Each condition has its causes, symptoms, and risk factors, making it challenging to develop universal treatments and prevention strategies.

★ *Diagnostic Challenges*: Some heart conditions have nonspecific symptoms or no symptoms at all, making early diagnosis difficult. Additionally, diagnostic tests can sometimes produce false positives or false negatives, leading to challenges in accurate detection.

★ *Prevention Challenges*: Preventing heart disease often involves lifestyle modifications, such as a healthy diet, regular exercise, and avoiding smoking and excessive alcohol consumption. However, changing behavior is challenging for many individuals, and adherence to long-term lifestyle changes can be difficult.

★ *Genetic Predisposition*: Genetic factors play a role in the development of heart disease. While lifestyle changes can mitigate some risks, individuals with a strong family history of heart disease may have a higher predisposition, making prevention more challenging.

★ *Access to Healthcare*: Disparities in healthcare access and socioeconomic factors can affect an individual's ability to receive timely and appropriate care for heart disease. Limited access to healthcare facilities and services can lead to delayed diagnosis and treatment.

★ *Limited Organ Donors*: For individuals with advanced heart failure, heart transplantation can be a life-saving option. However, the availability of suitable organ donors is limited, leading to long waiting lists and challenges in providing timely transplants.

★ *Risks of Medical Interventions*: While medical procedures and surgeries can be life-saving, they also carry inherent risks. Complications from surgeries and the need for lifelong medications post-surgery pose challenges to patients' overall well-being.

**Recommendations:**

- ★ *Regular Exercise*: Aim for at least 150 minutes of moderate-intensity aerobic activity or 75 minutes of vigorous-intensity activity per week, along with muscle-strengthening activities on two or more days a week.

- ★ *Quit Smoking*: If you smoke, seek help to quit. Smoking is a major risk factor for heart disease.

- ★ *Limit Alcohol*: If you drink, do so in moderation. This means up to one drink per day for women and up to two drinks per day for men.

- ★ *Explore Genetic Factors*: Continue researching the genetic components of heart disease to understand predispositions and potential targeted treatments.

- ★ *Develop Personalized Medicine Approaches*: Invest in research to develop personalized treatment plans based on an individual's genetic makeup, lifestyle, and environmental factors.

- ★ *Improve Diagnostic Tools*: Develop more accurate and accessible diagnostic tools, including blood tests and imaging techniques, to enhance early detection of heart disease and its risk factors.

- ★ *Study Lifestyle Interventions*: Conduct further research on the effectiveness of lifestyle interventions, such as diet, exercise, and stress management programs, in preventing and managing heart disease.

## – CODEBASE —

```
import pandas as pd
import numpy as np
```

```
[2] data = pd.read_csv('/content/heart.csv')
```

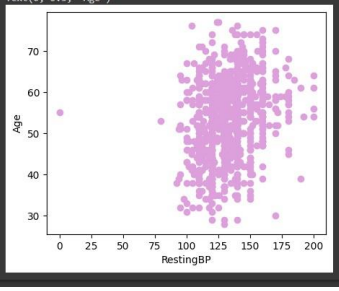
```
df = pd.DataFrame(data, columns=['Age','Sex','ChestPain','RestingBP','Cholesterol', 'Fasting BS', 'RestingECG', 'MaxHR', 'ExerciseAngina', 'Oldpeak', 'ST_Slope', "HeartDisease"])
print(df)
```

```
     Age Sex  ChestPain  RestingBP  Cholesterol  Fasting BS RestingECG  MaxHR  \
0     40   M        NaN        140          289         NaN     Normal    172
1     49   F        NaN        160          180         NaN     Normal    156
2     37   M        NaN        130          283         NaN         ST     98
3     48   F        NaN        138          214         NaN     Normal    108
4     54   M        NaN        150          195         NaN     Normal    122
..   ...  ..        ...        ...          ...         ...        ...    ...
913   45   M        NaN        110          264         NaN     Normal    132
914   68   M        NaN        144          193         NaN     Normal    141
915   57   M        NaN        130          131         NaN     Normal    115
916   57   F        NaN        130          236         NaN        LVH    174
917   38   M        NaN        138          175         NaN     Normal    173

    ExerciseAngina  Oldpeak ST_Slope  HeartDisease
0                N      0.0       Up             0
1                N      1.0     Flat             1
2                N      0.0       Up             0
3                Y      1.5     Flat             1
4                N      0.0       Up             0
..             ...      ...      ...           ...
913              N      1.2     Flat             1
914              N      3.4     Flat             1
915              Y      1.2     Flat             1
916              N      0.0     Flat             1
917              N      0.0       Up             0

[918 rows x 12 columns]
```

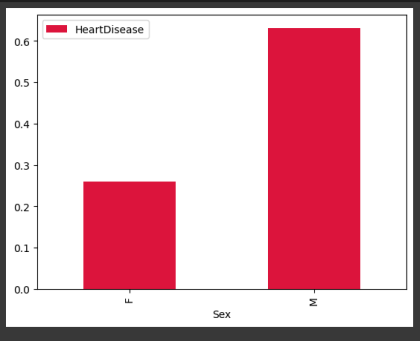
```
[5] import matplotlib.pyplot as plt
```

```
print(df['RestingBP'].describe())
plt.figure(figsize=(5,4))
plt.scatter(df['RestingBP'],df['Age'], color='plum')
plt.xlabel("RestingBP")
plt.ylabel("Age")
```

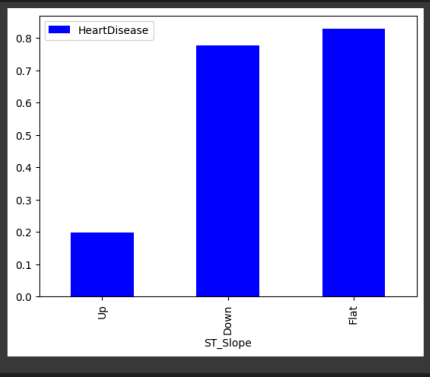
```
count    918.000000
mean     132.396514
std       18.514154
min        0.000000
25%      120.000000
50%      130.000000
75%      140.000000
max      200.000000
Name: RestingBP, dtype: float64
Text(0, 0.5, 'Age')
```



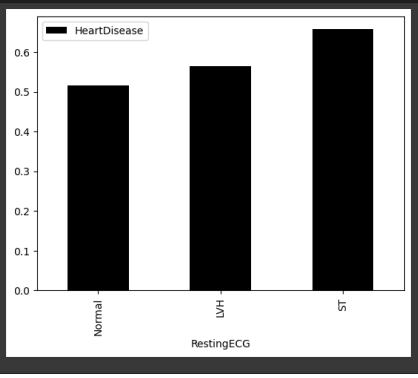
```
dic = df[['HeartDisease', 'Sex']].groupby('Sex').mean()
dic.plot(kind = 'bar', color = 'crimson')
plt.show()
```




```
dic = df[['ST_Slope', 'HeartDisease']].groupby('ST_Slope').mean().sort_values(by = 'HeartDisease')
dic.plot(kind = 'bar', color = 'blue')
plt.show()
```

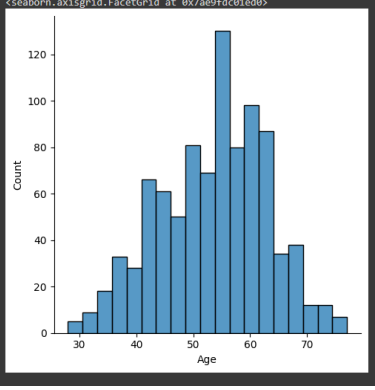



```
dic = df[['RestingECG', 'HeartDisease']].groupby('RestingECG').mean().sort_values(by = 'HeartDisease')
dic.plot(kind = 'bar', color = 'black')
plt.show()
```



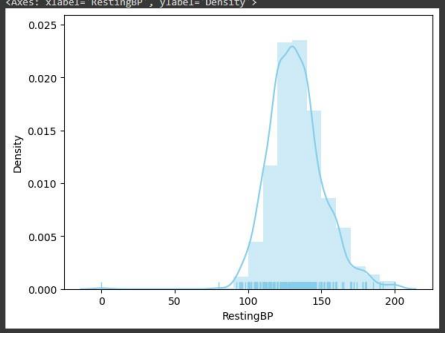

```
sns.displot(df['Age'])
```

```
<seaborn.axisgrid.FacetGrid at 0x7ae9fdc01ed0>
```



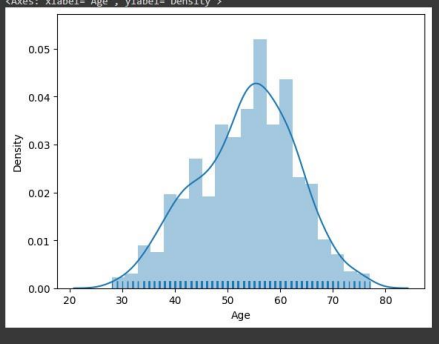
```
sns.distplot(df['RestingBP'],kde=True,hist=True,rug=True,bins=20,color='Skyblue')
```

```
<ipython-input-15-000ae3aff6c7>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['RestingBP'],kde=True,hist=True,rug=True,bins=20,color='Skyblue')
<Axes: xlabel='RestingBP', ylabel='Density'>
```



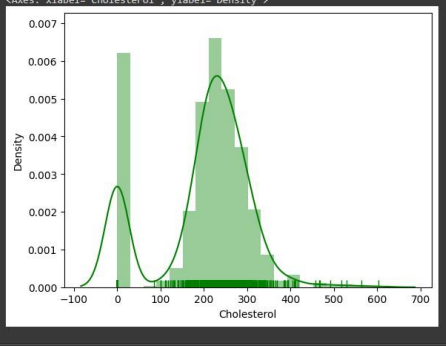

```
[ ] sns.distplot(df['Age'],kde=True,hist=True,rug=True,bins=20)
```

```
<ipython-input-16-6207c80abc7d>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Age'],kde=True,hist=True,rug=True,bins=20)
<Axes: xlabel='Age', ylabel='Density'>
```



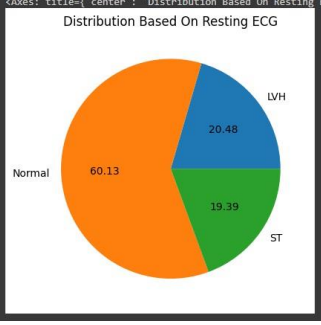

```
sns.distplot(df['Cholesterol'],kde=True,hist=True,rug=True,bins=20,color='Green')
```

```
<ipython-input-17-8ca3c3dc586a>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Cholesterol'],kde=True,hist=True,rug=True,bins=20,color='Green')
<Axes: xlabel='Cholesterol', ylabel='Density'>
```

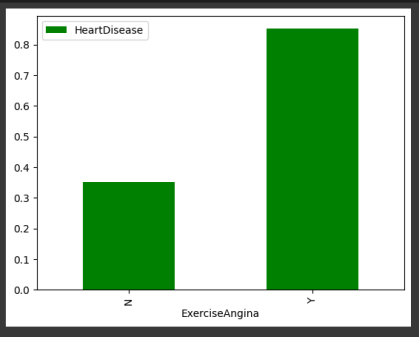


```
df.groupby('RestingECG').size().plot(kind='pie',autopct='%.2f',title='Distribution Based On Resting ECG')
```

```
<Axes: title={'center': 'Distribution Based On Resting ECG'}>
```

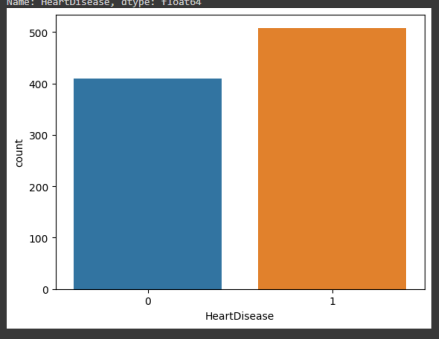



```
dic = df[['ExerciseAngina', 'HeartDisease']].groupby('ExerciseAngina').mean().sort_values(by = 'HeartDisease')
dic.plot(kind = 'bar', color = 'green')
plt.show()
```



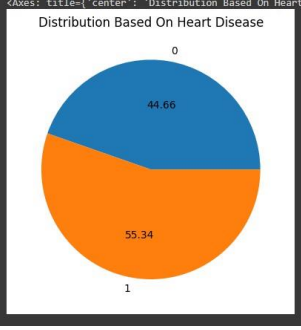

```
sns.countplot(df, x="HeartDisease")
print(df["HeartDisease"].value_counts()/len(df)*100)
```

```
1    55.337691
0    44.662309
Name: HeartDisease, dtype: float64
```

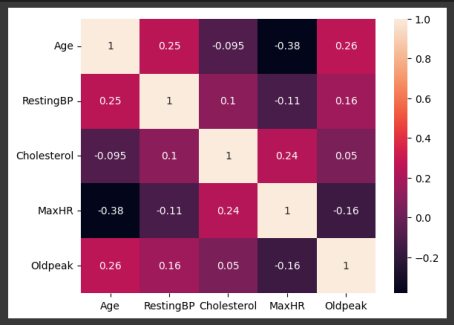



```
df.groupby('HeartDisease').size().plot(kind='pie',autopct='%.2f',title='Distribution Based On Heart Disease')
```

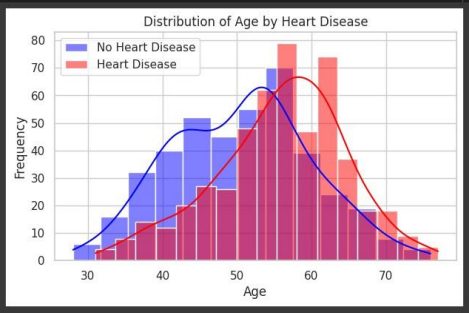
```
<Axes: title={'center': 'Distribution Based On Heart Disease'}>
```



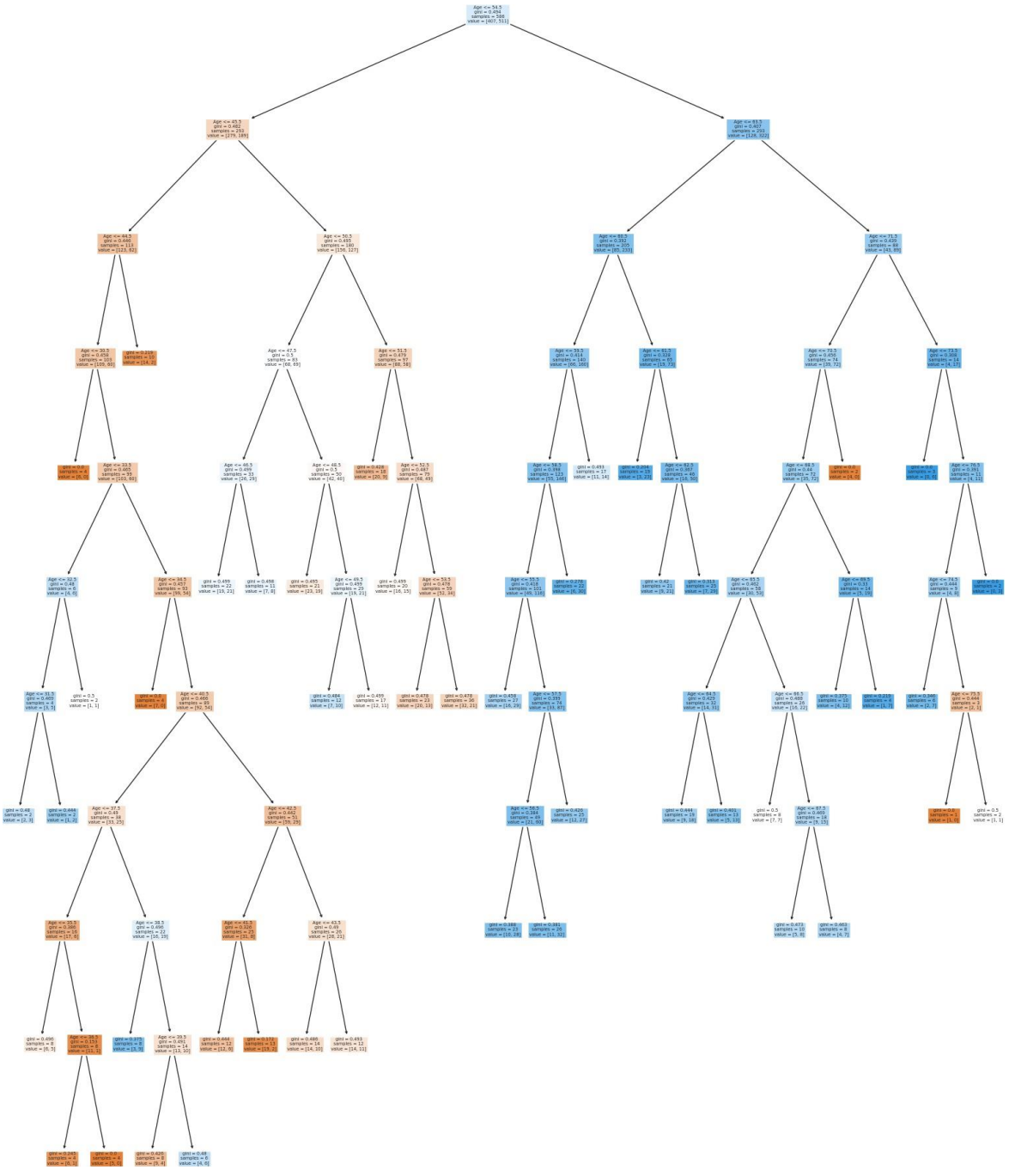
```
cols=["Age","RestingBP","Cholesterol","MaxHR","Oldpeak"]
corr=df[cols].corr()
sns.heatmap(corr,annot=True)
plt.show()
```




```
sns.set(style="whitegrid")

plt.figure(figsize=(7, 4))

sns.histplot(data=df[df['HeartDisease'] == 0], x='Age', kde=True, color='blue', label='No Heart Disease')

sns.histplot(data=df[df['HeartDisease'] == 1], x='Age', kde=True, color='red', label='Heart Disease')

plt.title('Distribution of Age by Heart Disease')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```




```
[ ] #1st Algorithm using Random Forest Classifier

[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.tree import plot_tree

[ ] rf_classifier = RandomForestClassifier(n_estimators=5, random_state=10)

[ ] X = df[['Age']]
    y = df['HeartDisease']

[ ] rf_classifier.fit(X, y)

    plt.figure(figsize=(20,25))
    plot_tree(rf_classifier.estimators_[0], feature_names=['Age'], filled=True)
    plt.show()
```


Age <= 54.5
gini = 0.463
samples = 1098
value = [607, 511]
Age <= 85.5
gini = 0.482
samples = 472
value = [275, 189]
Age <= 81.5
gini = 0.463
samples = 626
value = [128, 322]
Age <= 44.5
gini = 0.466
samples = 417
value = [237, 162]
Age <= 50.5
gini = 0.495
samples = 55
value = [38, 127]
Age <= 85.5
gini = 0.282
samples = 225
value = [85, 220]
Age <= 71.5
gini = 0.439
samples = 401
value = [43, 102]
Age <= 30.5
gini = 0.456
samples = 343
value = [209, 98]
gini = 0.219
samples = 10
value = [14, 2]
Age <= 47.5
gini = 0.5
samples = 50
value = [38, 69]
Age <= 51.5
gini = 0.479
samples = 5
value = [0, 58]
Age <= 61.5
gini = 0.418
samples = 141
value = [43, 73]
Age <= 61.5
gini = 0.218
samples = 84
value = [13, 73]
Age <= 73.5
gini = 0.401
samples = 372
value = [43, 72]
Age <= 73.5
gini = 0.363
samples = 29
value = [0, 30]
gini = 0.0
samples = 4
value = [6, 0]
Age <= 33.5
gini = 0.465
samples = 339
value = [203, 98]
Age <= 46.5
gini = 0.499
samples = 42
value = [26, 28]
Age <= 46.5
gini = 0.5
samples = 50
value = [42, 40]
gini = 0.426
samples = 18
value = [20, 9]
Age <= 52.5
gini = 0.487
samples = 37
value = [16, 49]
Age <= 54.5
gini = 0.494
samples = 124
value = [55, 104]
gini = 0.493
samples = 17
value = [11, 14]
gini = 0.044
samples = 15
value = [1, 23]
Age <= 52.5
gini = 0.097
samples = 69
value = [13, 50]
Age <= 69.5
gini = 0.42
samples = 372
value = [35, 72]
gini = 0.0
samples = 2
value = [8, 0]
gini = 0.0
samples = 4
value = [6, 0]
Age <= 76.5
gini = 0.391
samples = 25
value = [4, 11]
Age <= 32.5
gini = 0.48
samples = 4
value = [4, 6]
Age <= 38.5
gini = 0.457
samples = 335
value = [199, 92]
gini = 0.499
samples = 22
value = [19, 21]
gini = 0.498
samples = 11
value = [7, 9]
gini = 0.455
samples = 21
value = [23, 19]
Age <= 45.5
gini = 0.495
samples = 29
value = [19, 21]
gini = 0.499
samples = 20
value = [16, 15]
Age <= 53.5
gini = 0.478
samples = 17
value = [10, 19]
Age <= 51.5
gini = 0.418
samples = 111
value = [85, 169]
gini = 0.275
samples = 13
value = [6, 30]
gini = 0.42
samples = 21
value = [9, 21]
gini = 0.311
samples = 15
value = [7, 29]
Age <= 85.5
gini = 0.462
samples = 53
value = [30, 55]
Age <= 89.5
gini = 0.31
samples = 319
value = [5, 17]
Age <= 74.5
gini = 0.44
samples = 14
value = [4, 9]
gini = 0.5
samples = 2
value = [0, 3]
Age <= 31.5
gini = 0.469
samples = 8
value = [4, 7]
gini = 0.5
samples = 2
value = [1, 1]
gini = 0.0
samples = 4
value = [0, 6]
Age <= 40.5
gini = 0.466
samples = 331
value = [195, 86]
gini = 0.458
samples = 27
value = [16, 29]
Age <= 51.5
gini = 0.339
samples = 84
value = [14, 67]
Age <= 84.5
gini = 0.422
samples = 18
value = [14, 5]
Age <= 84.5
gini = 0.488
samples = 35
value = [16, 22]
gini = 0.375
samples = 10
value = [4, 12]
gini = 0.219
samples = 9
value = [1, 8]
gini = 0.346
samples = 6
value = [6, 2]
Age <= 75.5
gini = 0.444
samples = 8
value = [4, 8]
gini = 0.48
samples = 4
value = [4, 6]
Age <= 37.5
gini = 0.48
samples = 16
value = [10, 15]
Age <= 62.5
gini = 0.492
samples = 53
value = [35, 29]
Age <= 56.5
gini = 0.314
samples = 69
value = [11, 55]
gini = 0.426
samples = 15
value = [12, 27]
gini = 0.444
samples = 18
value = [5, 10]
gini = 0.483
samples = 13
value = [5, 13]
gini = 0.5
samples = 8
value = [7, 7]
Age <= 67.5
gini = 0.469
samples = 19
value = [5, 12]
gini = 0.4
samples = 4
value = [1, 0]
gini = 0.5
samples = 2
value = [1, 1]
Age <= 39.5
gini = 0.386
samples = 19
value = [5, 2]
Age <= 39.5
gini = 0.496
samples = 27
value = [16, 19]
Age <= 41.5
gini = 0.320
samples = 25
value = [3, 9]
Age <= 43.5
gini = 0.49
samples = 28
value = [28, 21]
gini = 0.388
samples = 23
value = [10, 28]
gini = 0.381
samples = 20
value = [11, 32]
gini = 0.472
samples = 10
value = [5, 8]
gini = 0.463
samples = 9
value = [4, 7]
gini = 0.496
samples = 8
value = [6, 5]
Age <= 36.5
gini = 0.153
samples = 11
value = [1, 8]
gini = 0.472
samples = 8
value = [5, 3]
Age <= 33.5
gini = 0.497
samples = 14
value = [11, 10]
gini = 0.444
samples = 12
value = [12, 6]
gini = 0.172
samples = 13
value = [13, 2]
gini = 0.486
samples = 14
value = [14, 10]
gini = 0.491
samples = 12
value = [14, 11]
gini = 0.245
samples = 7
value = [6, 1]
gini = 0.0
samples = 4
value = [5, 0]
gini = 0.426
samples = 8
value = [5, 4]
gini = 0.48
samples = 6
value = [4, 6]

```
[ ] #2nd Algorithm using Train and Test Split and Printing the Accuracy Score

[ ] from sklearn.model_selection import train_test_split
    from sklearn.svm import SVC
    from sklearn.metrics import accuracy_score

[ ] rawdataframe = pd.read_csv('/content/heartguy.csv')

[ ] print("Initial date:")
    print(rawdataframe.head())

    Initial date:
       Age Sex ChestPainType  RestingBP  Cholesterol  FastingBS RestingECG  MaxHR  \
    0   41   M           ATA        241          289          1     Normal    272
    1   49   F           NAP        261          281          1     Normal    256
    2   37   M           ATA        231          283          1         ST     98
    3   48   F           ASY        238          224          1     Normal    218
    4   54   M           NAP        251          295          1     Normal    222

      ExerciseAngina  Oldpeak ST_Slope  HeartDisease
    0              N      1.0       Up             1
    1              N      2.0     Flat             2
    2              N      1.0       Up             1
    3              Y      2.5     Flat             2
    4              N      1.0       Up             1
```

```
[ ] rawdataframe = pd.get_dummies(rawdataframe, columns=['Sex'])

[ ] rawdataframe = pd.get_dummies(rawdataframe, columns=['ChestPainType'])

[ ] rawdataframe = pd.get_dummies(rawdataframe, columns=['ExerciseAngina'])

[ ] rawdataframe = pd.get_dummies(rawdataframe, columns=['RestingECG'])

[ ] rawdataframe = pd.get_dummies(rawdataframe, columns=['ST_Slope'])

[ ] rawdataframe
```

| | Age | RestingBP | Cholesterol | FastingBS | MaxHR | Oldpeak | HeartDisease | Sex_F | Sex_M | ChestPainType_ASY | ... | ChestPainType_NAP | ChestPainType_TA | ExerciseAngina_N | ExerciseAngina_Y | RestingECG_LVH | Restin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 41 | 241 | 289 | 1 | 272 | 1.0 | 1 | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | 0 | |
| **1** | 49 | 261 | 281 | 1 | 256 | 2.0 | 2 | 1 | 0 | 0 | ... | 1 | 0 | 1 | 0 | 0 | |
| **2** | 37 | 231 | 283 | 1 | 98 | 1.0 | 1 | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | 0 | |
| **3** | 48 | 238 | 224 | 1 | 218 | 2.5 | 2 | 1 | 0 | 1 | ... | 0 | 0 | 0 | 1 | 0 | |
| **4** | 54 | 251 | 295 | 1 | 222 | 1.0 | 1 | 0 | 1 | 0 | ... | 1 | 0 | 1 | 0 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **913** | 45 | 221 | 264 | 1 | 232 | 2.2 | 2 | 0 | 1 | 0 | ... | 0 | 1 | 1 | 0 | 0 | |
| **914** | 68 | 244 | 293 | 2 | 242 | 3.4 | 2 | 0 | 1 | 1 | ... | 0 | 0 | 1 | 0 | 0 | |
| **915** | 57 | 231 | 232 | 1 | 225 | 2.2 | 2 | 0 | 1 | 1 | ... | 0 | 0 | 0 | 1 | 0 | |
| **916** | 57 | 231 | 236 | 1 | 274 | 1.0 | 2 | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 1 | |
| **917** | 38 | 238 | 275 | 1 | 273 | 1.0 | 1 | 0 | 1 | 0 | ... | 1 | 0 | 1 | 0 | 0 | |

918 rows x 21 columns

```
[ ] #Presence or absence of heart disease (2 for presence, 1 for absence)

[ ] X = rawdataframe.drop('HeartDisease', axis=1)
    y = rawdataframe['HeartDisease']

[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.9, random_state=121)

[ ] svm_classifier = SVC()

[ ] svm_classifier.fit(X_train, y_train)

    SVC()

[ ] y_pred = svm_classifier.predict(X_test)

[ ] accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy: ", accuracy)

    Accuracy:  0.619105199516324
```

```
[ ] #Fasting blood sugar level (1 for true, 0 for false)

[ ] X = rawdataframe.drop('FastingBS', axis=1)
    y = rawdataframe['FastingBS']

[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=121)

[ ] svm_classifier = SVC()

[ ] svm_classifier.fit(X_train, y_train)

    SVC()

[ ] y_pred = svm_classifier.predict(X_test)

[ ] accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy: ", accuracy)

    Accuracy:  0.7973856209150327
```

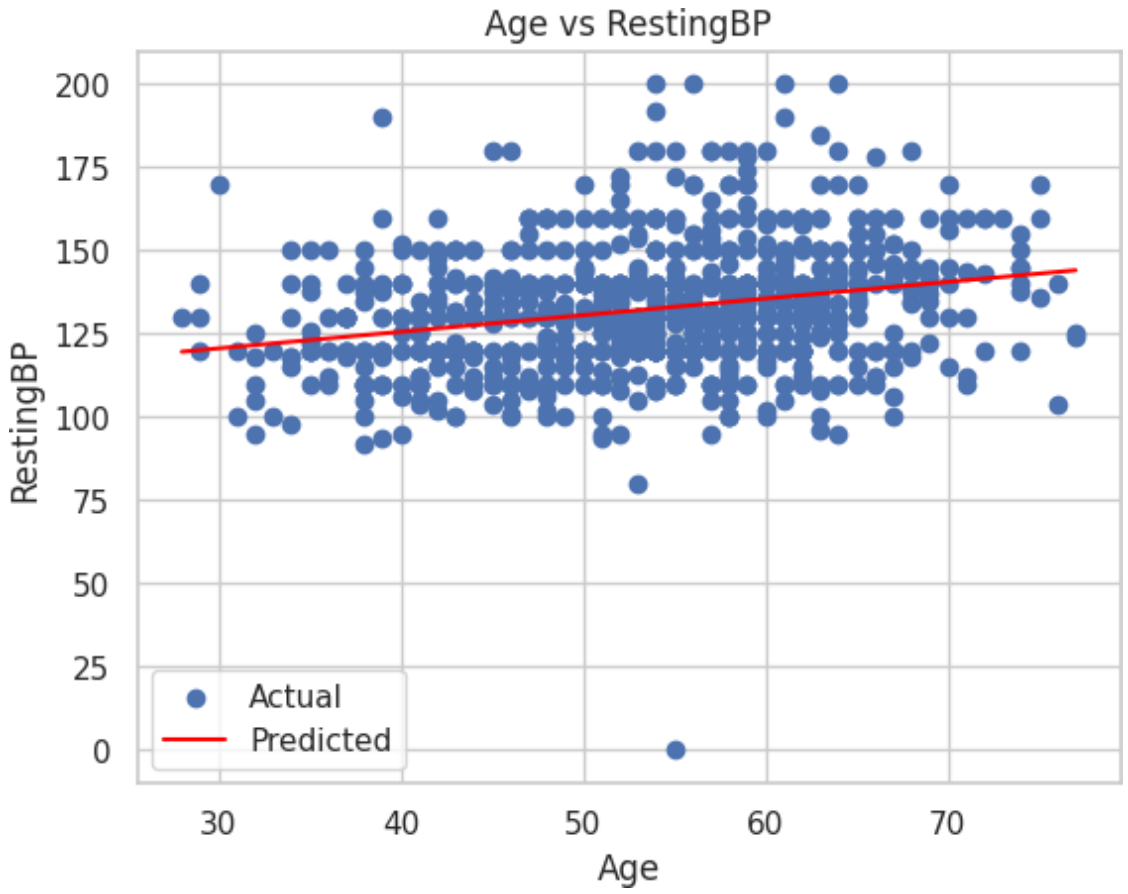
```
[ ] #Slope of the ST segment on the ECG during exercise (e.g., Up, Flat)

[ ] X = rawdataframe.drop('ST_Slope_Down', axis=1)
    y = rawdataframe['ST_Slope_Down']

[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=121)

[ ] svm_classifier = SVC()

[ ] svm_classifier.fit(X_train, y_train)

    SVC()

[ ] y_pred = svm_classifier.predict(X_test)

[ ] accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy: ", accuracy)

    Accuracy:  0.9331259720062208
```

```
[ ] #Use Decision Tree Classifier and predict the new input

[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score
    from sklearn.datasets import load_iris
    from sklearn.model_selection import train_test_split
    from sklearn.tree import DecisionTreeClassifier
    from sklearn import metrics

[ ] df1 = pd.read_csv('/content/heart.csv')

[ ] print(df1)

        Age Sex ChestPainType  RestingBP  Cholesterol  FastingBS RestingECG  \
    0     40   M           ATA        140          289          0     Normal
    1     49   F           NAP        160          180          0     Normal
    2     37   M           ATA        130          283          0         ST
    3     48   F           ASY        138          214          0     Normal
    4     54   M           NAP        150          195          0     Normal
    ..   ...  ..           ...        ...          ...        ...        ...
    913   45   M            TA        110          264          0     Normal
    914   68   M           ASY        144          193          1     Normal
    915   57   M           ASY        130          131          0     Normal
    916   57   F           ATA        130          236          0        LVH
    917   38   M           NAP        138          175          0     Normal

        MaxHR ExerciseAngina  Oldpeak ST_Slope  HeartDisease
    0     172              N      0.0       Up             0
    1     156              N      1.0     Flat             1
    2      98              N      0.0       Up             0
    3     108              Y      1.5     Flat             1
    4     122              N      0.0       Up             0
    ..    ...            ...      ...      ...           ...
    913   132              N      1.2     Flat             1
    914   141              N      3.4     Flat             1
    915   115              Y      1.2     Flat             1
    916   174              N      0.0     Flat             1
    917   173              N      0.0       Up             0

    [918 rows x 12 columns]
```

```
[ ] x = df1[['Age']]
    y = df1['ChestPainType']

[ ] dt_classifier = DecisionTreeClassifier()

[ ] dt_classifier.fit(x, y)

    DecisionTreeClassifier()

[ ] new_data = {
        'Age': [10]
    }

[ ] new_df = pd.DataFrame(new_data)

[ ] prediction = dt_classifier.predict(new_df)

[ ] print("Prediction: ", prediction)

    Prediction:  ['ATA']

[ ] ################

[ ] x = df1[['Age']]
    y = df1['ExerciseAngina']

[ ] dt_classifier = DecisionTreeClassifier()

[ ] dt_classifier.fit(x, y)

    DecisionTreeClassifier()
```

```
[ ] new_data = {
        'Age': [70]
    }

[ ] new_df = pd.DataFrame(new_data)

[ ] prediction = dt_classifier.predict(new_df)

[ ] print("Prediction: ", prediction)

    Prediction:  ['Y']
```

```
[ ] #Make the predicted vs actual graph

[ ] from sklearn.linear_model import LinearRegression

[ ] model = LinearRegression()
    model.fit(df[['Age']], df['RestingBP'])

    ▾ LinearRegression
    LinearRegression()

[ ] episodes_range = np.linspace(df['Age'].min(), df['Age'].max(),50).reshape(-1, 1)
    predicted_popularity = model.predict(episodes_range)

    /usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
      warnings.warn(

    plt.scatter(df['Age'], df['RestingBP'], label='Actual')
    plt.plot(episodes_range, predicted_popularity, color='red', label='Predicted')
    plt.xlabel('Age')
    plt.ylabel('RestingBP')
    plt.title('Age vs RestingBP')
    plt.legend()
    plt.show()
```



**Dependencies:**

The external libraries, packages, or frameworks used in the code are as follows:

- ★ *Pandas*: We have used them for data manipulation and analysis. It provides data structures like DataFrame which is used to store and manipulate the dataset.

- ★ *Numpy*: Numpy is used for numerical operations. In this project, it is utilized for mathematical operations on arrays and data.

- ★ *Matplotlib*: It is a plotting library that provides a MATLAB-like interface. We have used it to create various plots such as scatter plots, bar plots, heatmaps, etc to visualize the data.

- ★ *Seaborn*: It is a statistical visualization library baked on Matplotlib that simplifies the process of creating
- ★ informative and attractive statistical graphics. We have used it for creating heat maps and for creating distribution plots (sns.distplot), count plots (sns.countplot), and histograms (sns.histplot).

- ★ *Sklearn(sci-kit-learn)*: It is a machine learning library. We have imported the RandomForestClassifier and plot_tree from sci-kit-learn and used them to create and visualize a random forest classifier. Also, we have imported train_test_split for splitting the data into training and testing sets, SVC for Support Vector Classification, and accuracy_score for evaluating the model's accuracy. Also used it for logistic regression modelling and to evaluate the accuracy of a classification model.

**Environment Setup:**

- ★ *Programming Language*:
  Google Colab primarily supports Python, making it the language of choice for your project.

- ★ *Integrated Development Environment (IDE)*:
  Jupyter Notebook: Python code may be created, edited, and executed in a notebook format thanks to Google Colab's connection with Jupyter Notebook. Combining code, written explanations, and visuals is best done in this way.

- ★ *Hardware Configuration*:
  Depending on availability, Google Colab offers cloud-based computing resources such as CPUs, GPUs, and TPUs. The hardware setup can change depending on the requirements of your project. You are able to use GPU and TPU resources for machine learning activities; these are particularly helpful for training deep learning models.

- ★ *Data Storage and Management*:
  Google Colab makes it simple to maintain and load datasets for your project by enabling you to upload and retrieve data files straight from Google Drive.

★ *Package Management*:
To install extra libraries and dependencies in your Colab environment, utilize Python package managers such as pip or Conda.

★ *Collaboration Features*:
One of the collaboration tools available in Google Colab is the sharing of notebooks with other users. This is helpful when asking partners for their opinions or for team initiatives.

★ *Notebook Integration*:
You can execute code step-by-step, provide explanations, create visualizations, and document your project using Colab notebooks.

★ *Version Control*:
Git can be connected with Google Colab to enable version control, letting you collaborate with others and keep track of changes made to your code.

**Summary:**

Our goal in this project was to use machine learning techniques to forecast the risk of heart disease. We constructed and assessed the Random Forest Classifier and Decision Tree Classifier using a dataset comprising personal health and demographic characteristics dataset. Accuracy was one of the main evaluation measures, and we used strategies such as train-test split for validation.

The research emphasized the significance of characteristics including age, blood pressure, cholesterol, and exercise-induced angina, and it offered insightful information about heart disease prognosis. While the Decision Tree model offered interpretability, the Random Forest model gave robust and accurate predictions thanks to its ensemble learning capabilities.

**Future Work:**

★ *Feature Engineering*:
More research into feature engineering methods, such as adding new variables or examining how current features interact with one another, could improve the predictive capacity of the model.

★ *Data Expansion*:
Adding more pertinent features or data sources to the dataset could increase the model's robustness and accuracy. For instance, adding lifestyle or genetic data could yield more thorough insights.

★ *Ensemble approaches*:
If you want to enhance the performance of your model, you should look at more sophisticated ensemble approaches like gradient boosting (like XGBoost or LightGBM).

- ★ *Constant Monitoring*: In the medical field, models may require constant observation and revision to accommodate evolving patient demographics and clinical recommendations.

- ★ *Working Together with Medical Specialists*: Better model design, validation, and comprehension of clinical consequences can result from working together with healthcare professionals and domain specialists.

- ★ *Ethical Considerations*: It is crucial to make sure that the model is used in a way that complies with all applicable laws and regulations, especially those about privacy.

To sum up, this effort is a big step toward the use of machine learning to predict the risk of heart disease. Subsequent research endeavors may augment the precision, comprehensibility, and utility of the model, culminating in better medical results and the timely identification of cardiac ailments.

**References**:

https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction