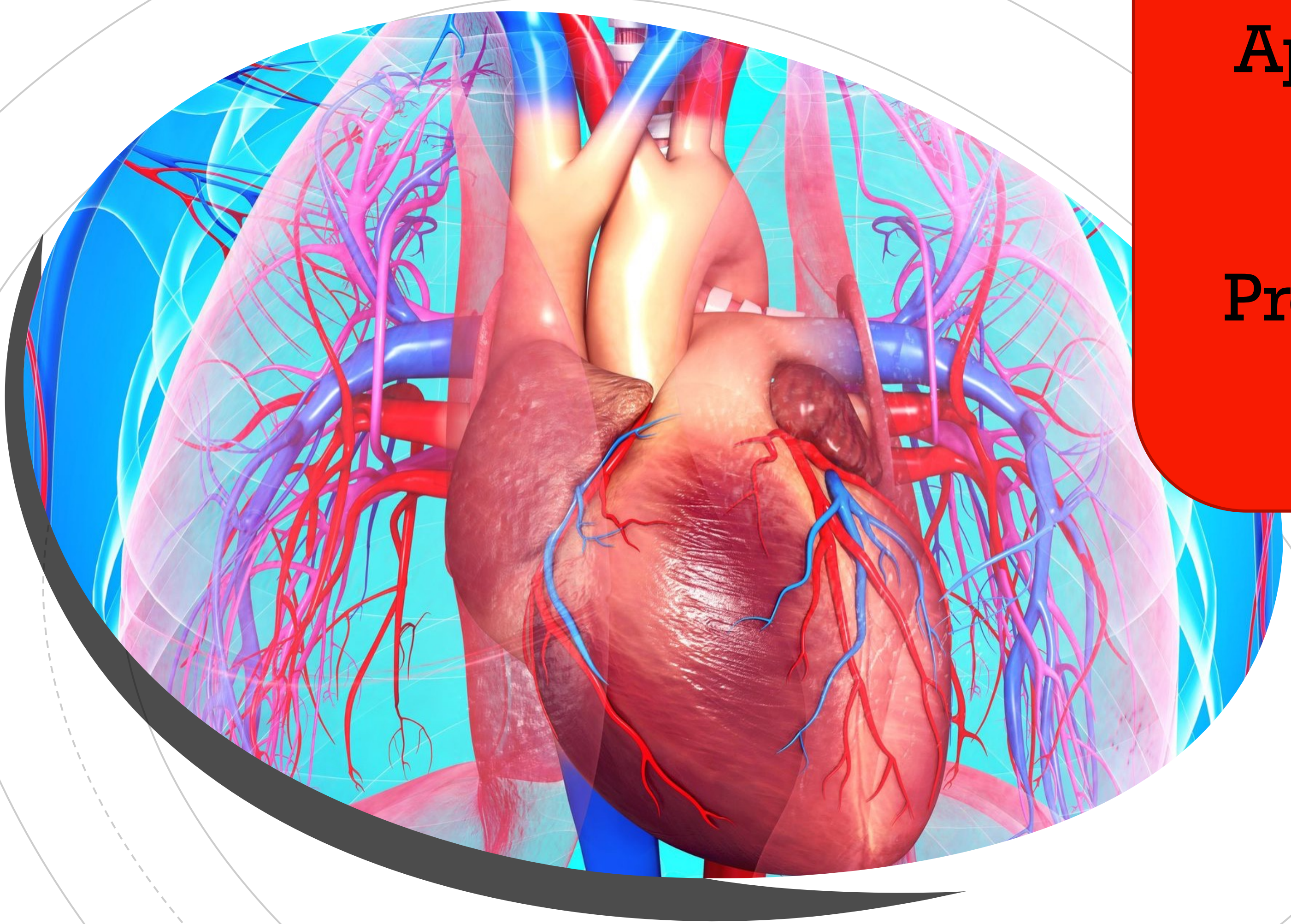


Applications of Machine Learning and Logistic Regression: Predicting Heart Disease

David Baron-Vega

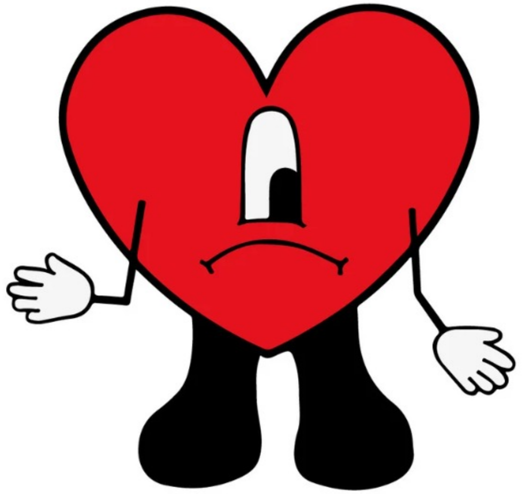
What is Heart Disease and why did we choose it?

- Despite continued medicinal research, cardiovascular disease is still the largest cause of death in the USA.
- Kills nearly one American every 34 seconds *each day.*
- *Leading* cause of death for most racial and ethnic groups across the globe.

Descriptive Analytics: Describing the Data

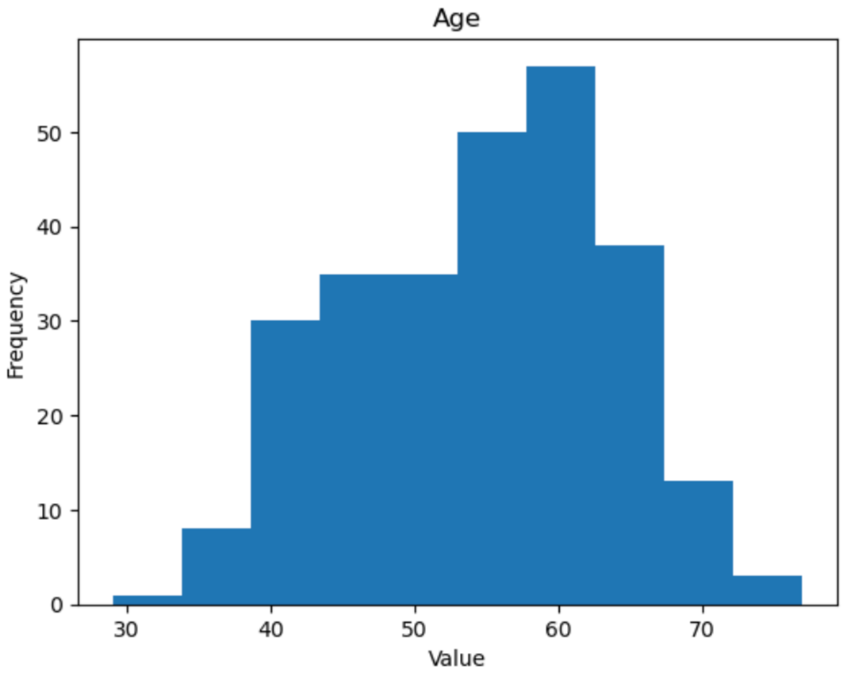
- 13 independent variables:
- Age, Sex, Chest Pain, Blood Pressure, Cholesterol, Blood Sugar, EKG Results, Max Hear Rate, Exercise Angina, ST Depression, Slope of ST, Visible Vessels, Thallium.
- 1 dependent variable: Heart Disease!

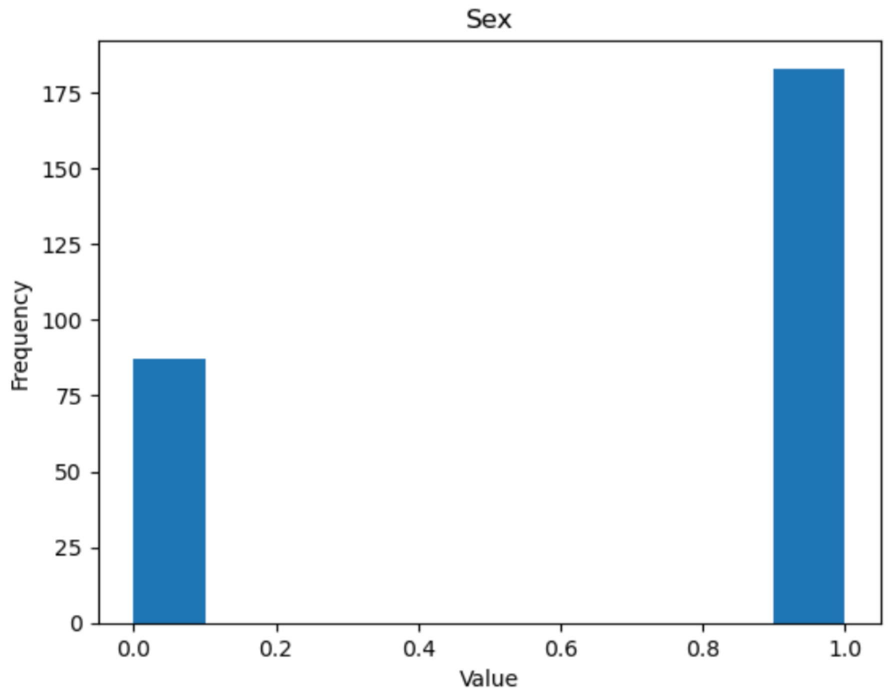
Size of sample (n):
270 Patients

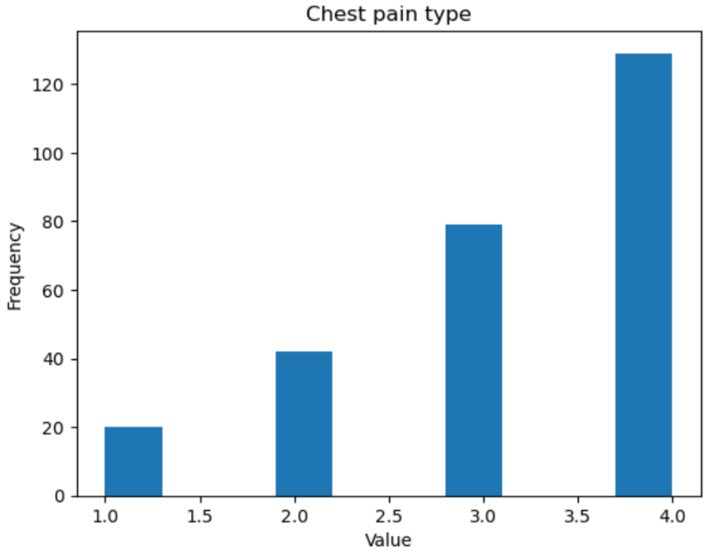


Age: Age of patient in years

Sex: 0 = female, 1 = male

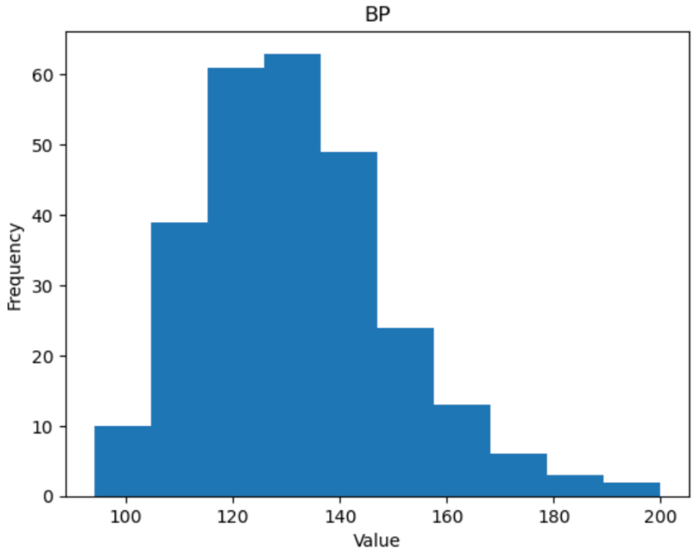






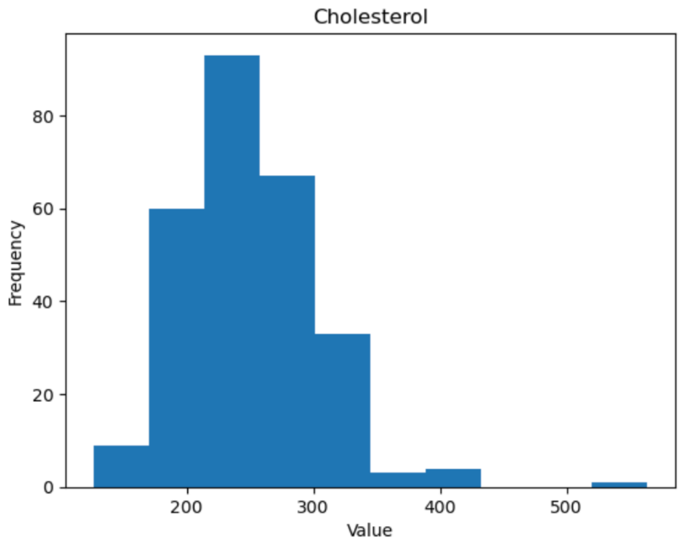
Chest Pain Type :

Type 1 : Typical angina
Type 2 : Atypical angina
Type 3: Non-anginal pain
Type 4: Asymptomatic



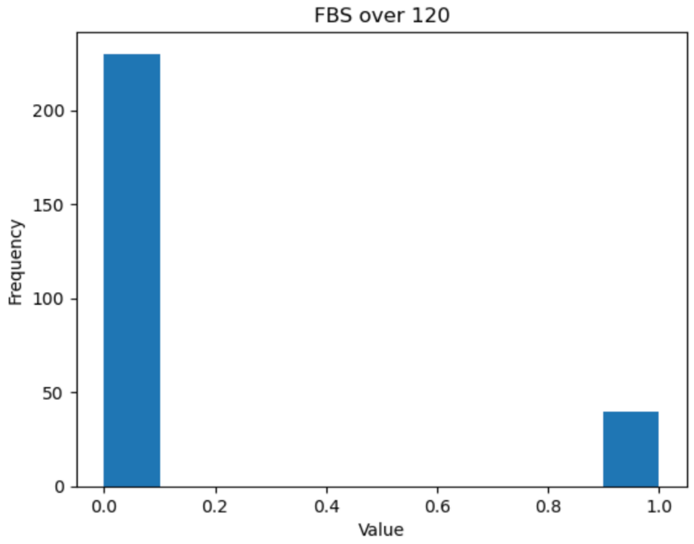
Blood Pressure:

Resting blood pressure (in mm Hg on admission to the hospital)



Cholesterol :

Serum cholesterol in mg/dl

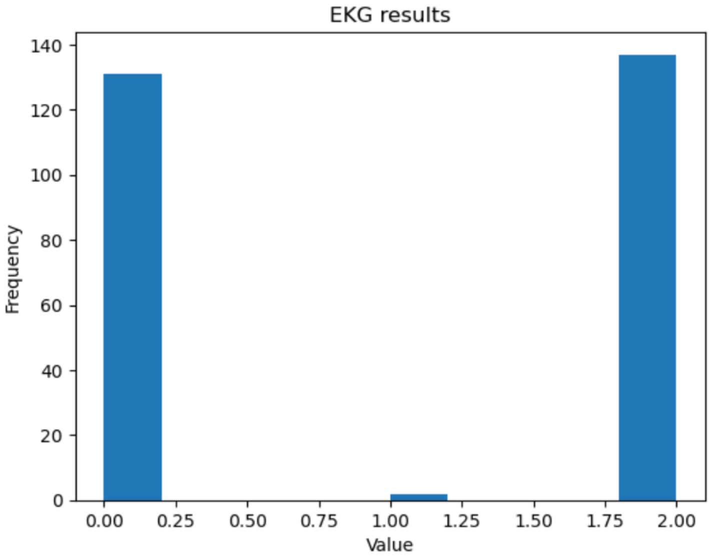


Fasting Blood Sugar :

FBS > 120 ml/dl

0 = false

1 = true



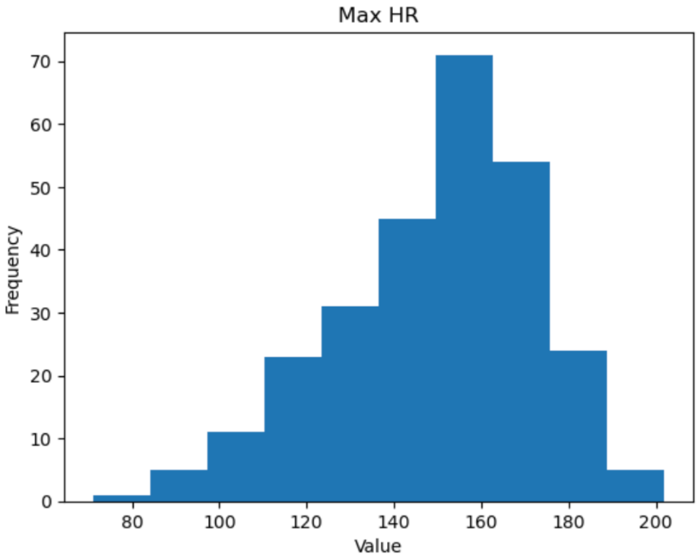
EKG Results :

Resting electrocardiographic results

0 = Normal

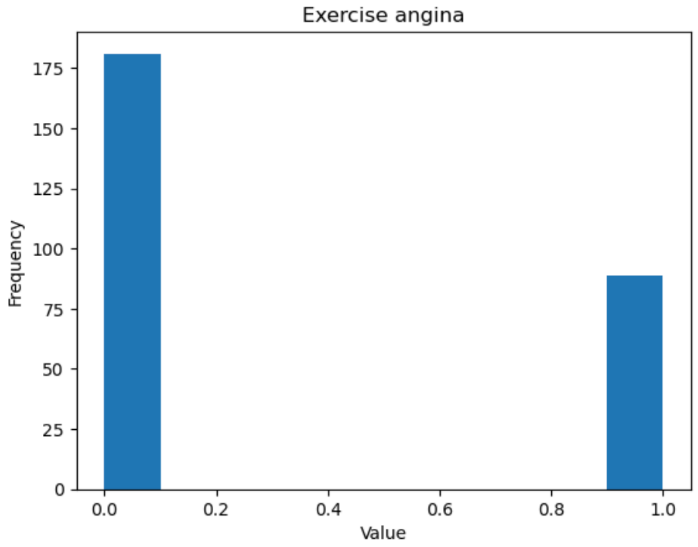
1 = ST-T wave abnormality

2 = Probable or definite left ventricular hypertrophy



Max Heart Rate :

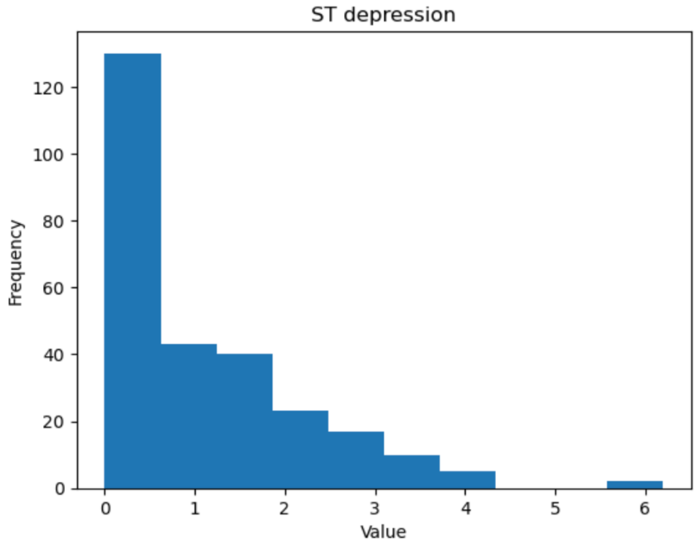
Maximum heart rate achieved



Exercise Induced Angina :

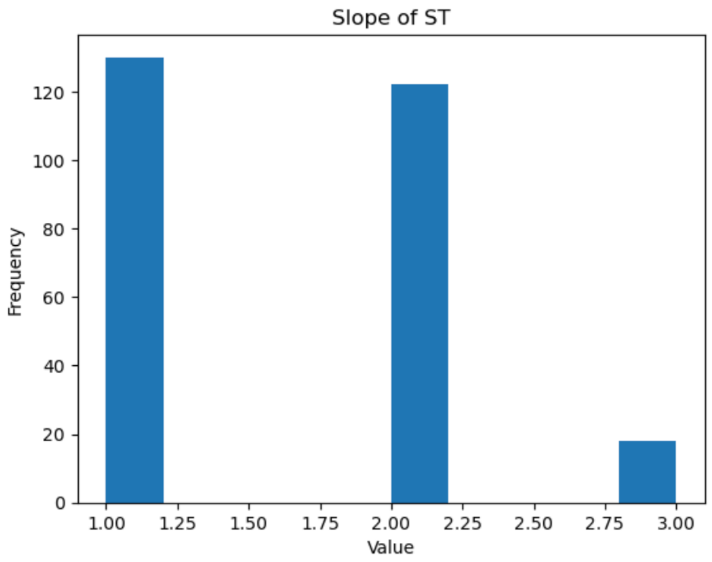
0 = No

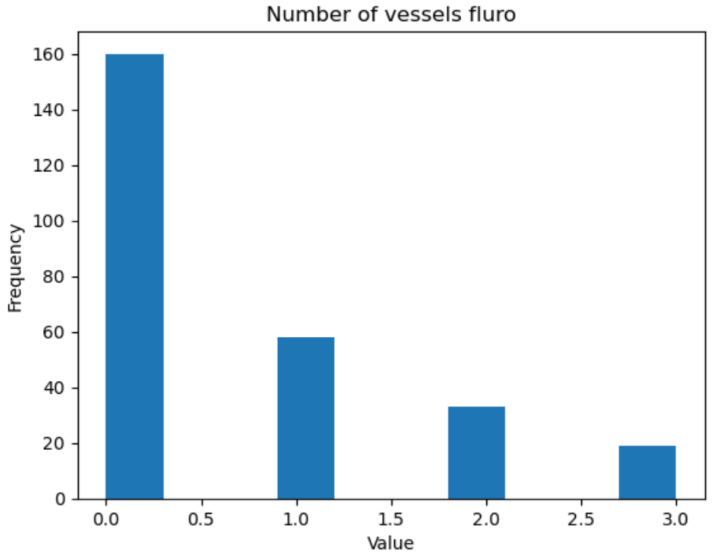
1 = Yes



ST Depression :

Time when ST measure depression was noted





Slope of ST :

Slope of the peak exercise ST segment

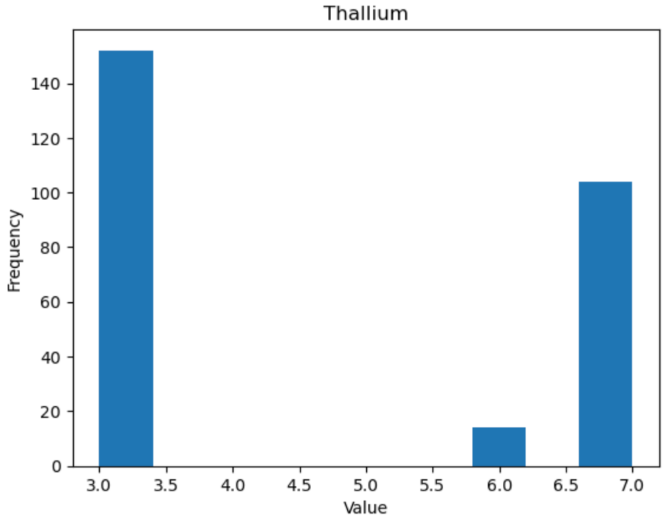
1 = Up-sloping

2 = Flat

3 = Down-sloping

Number of Vessels :

Number of major vessels ranging from 0 to 3 and colored by flouroscopy

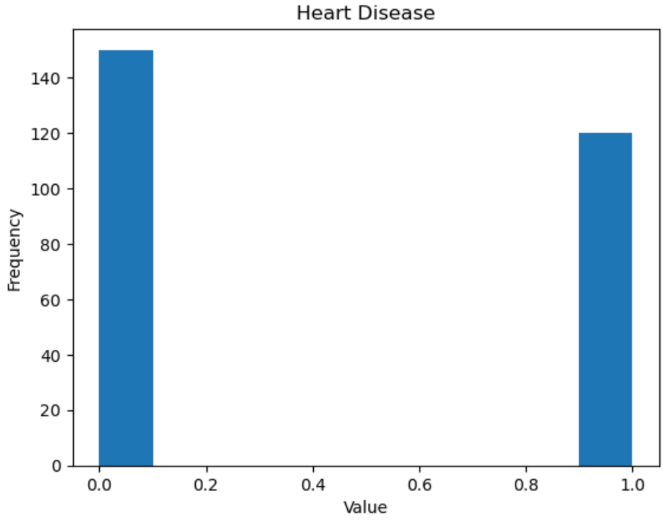


Thallium visibility :

3 = normal

6 = fixed defect

7 = reversable defect



Heart Disease:

0 = Absent

1 = Present

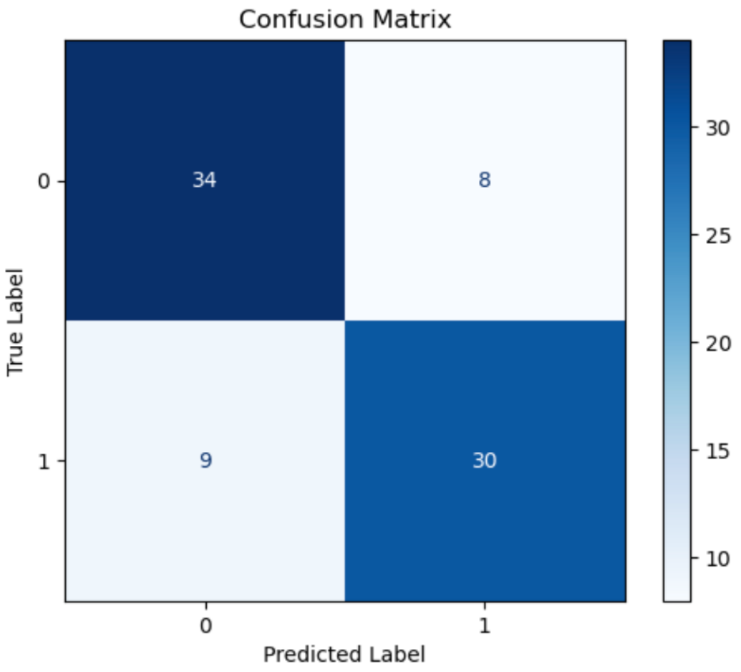
Is the Data Balanced?

- # of participants diagnosed with heart disease vs. not diagnosed: ≈
- *Significant difference between # of men and # of women observed!
- Age is fairly distributed between the ranges of 30-70 (slightly right-skewed)
- All other variables approximate either an expected normal distribution or an expected binomial distribution
- Conclusion:
    - The dataset is fairly balanced, but the models may likely improve by implementing a broader study and by performing some level of variable treatment.

Predicting Heart Disease

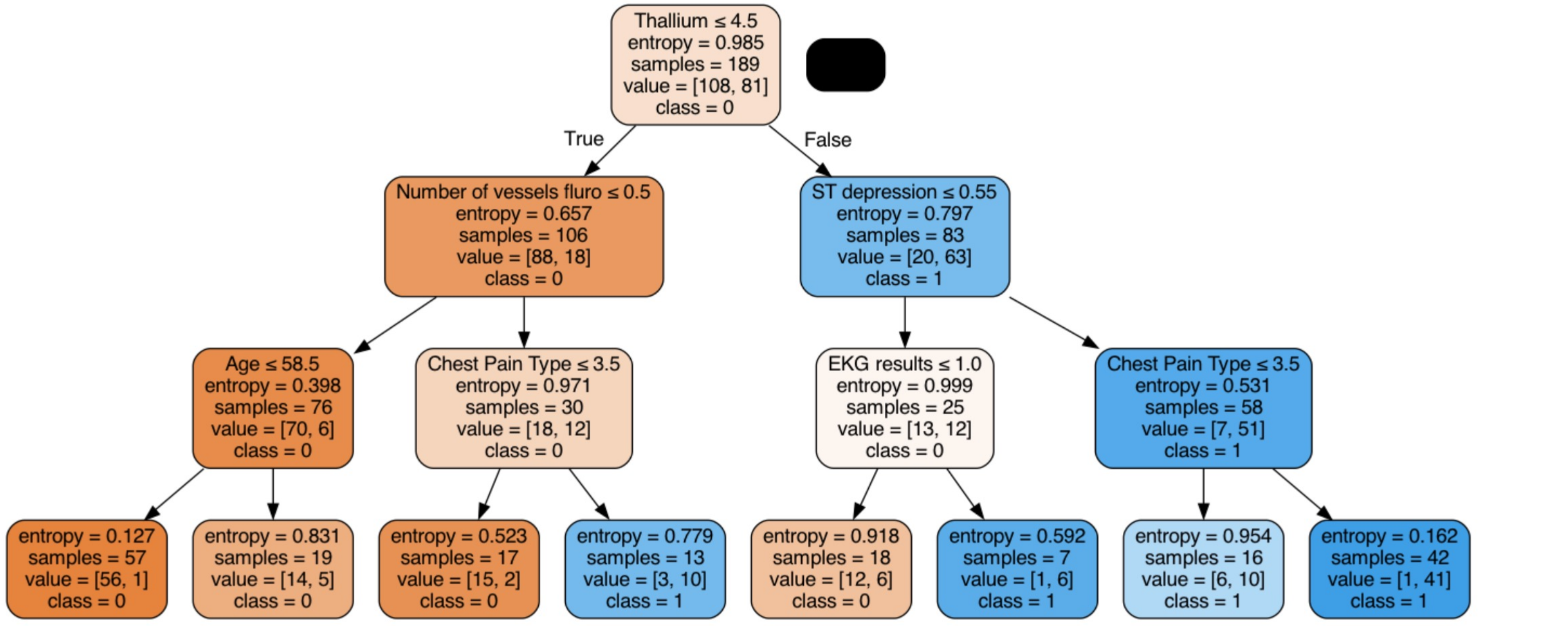
- Machine Learning Model: Single Decision Tree ANN
- Logisitic Regression Modeling

Results: The Machine Learning Model:

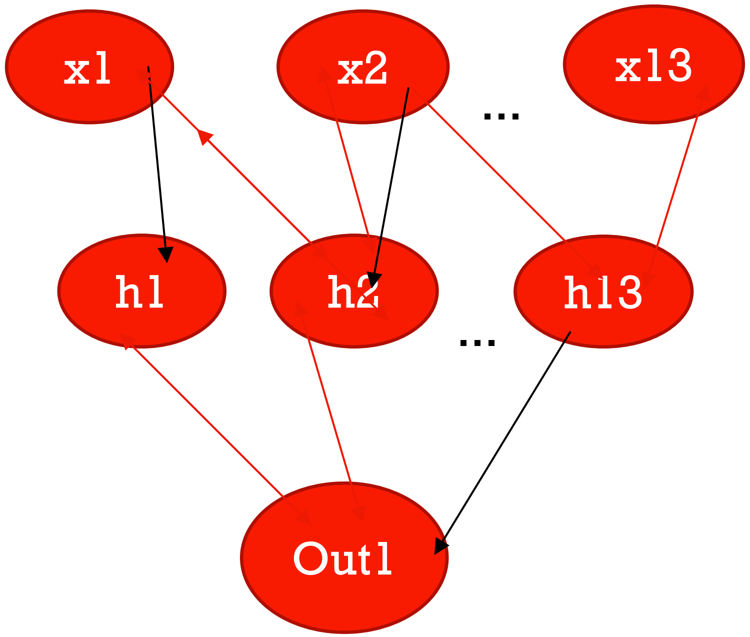


Accuracy: 0.7703703703703704
Precision: 0.7636363636363637
Recall: 0.7
F1 score: 0.7304347826086957

[67]:



What kind of Neural Network is this?



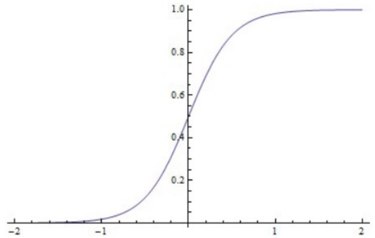
- The hidden layer is where all the magic is!
- Assigning weights and iteratively improves optimization.

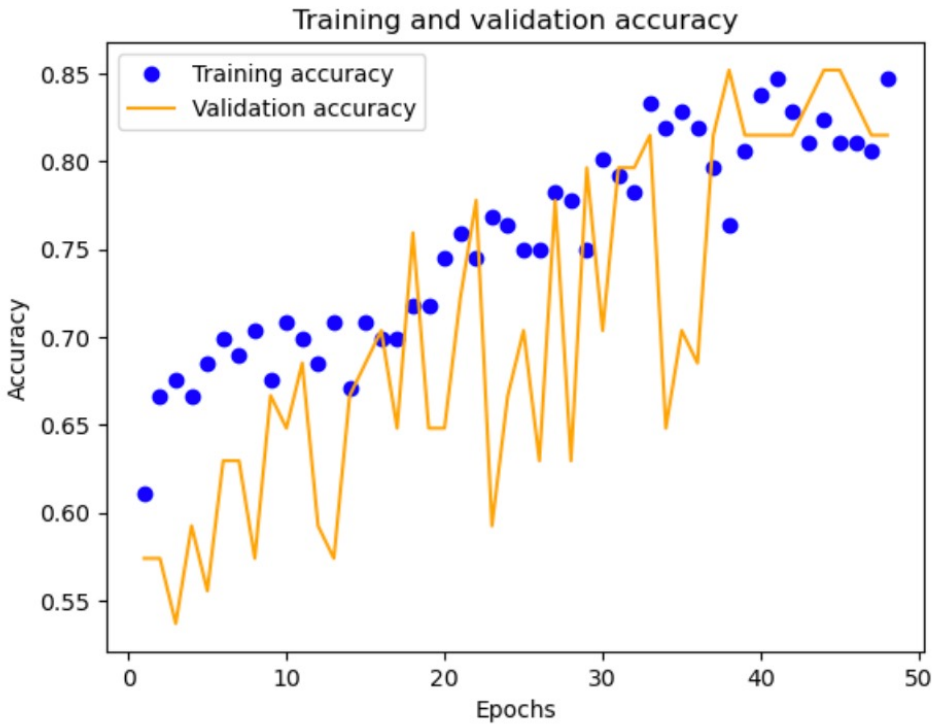
- How to decide if a node should fire
  - Usually done with a sigmoid function

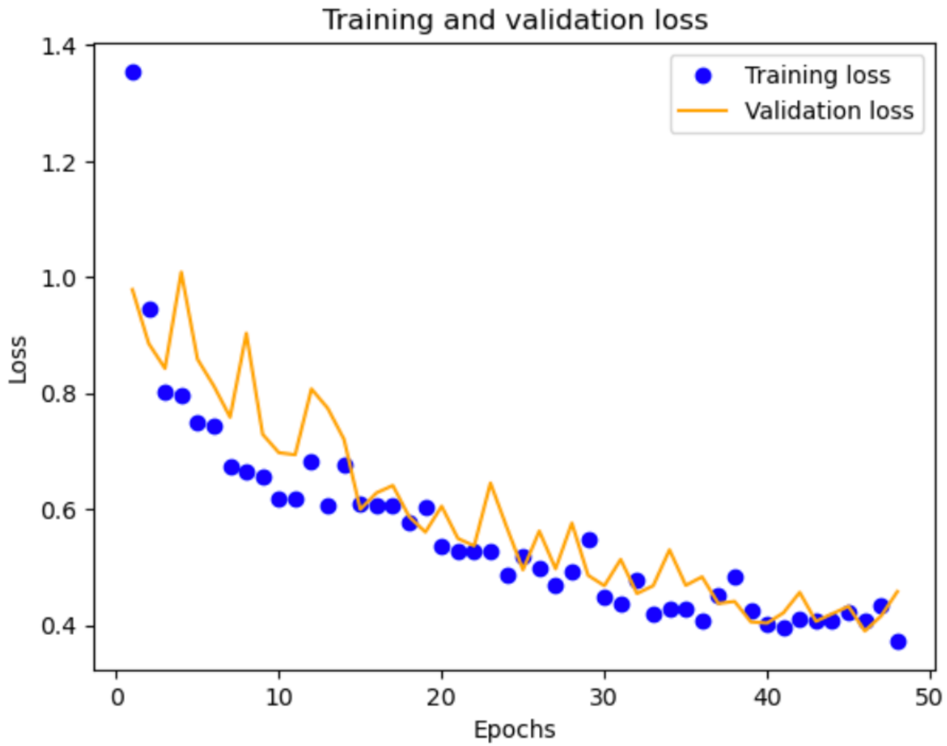
$$f(x) = \frac{1}{1 + e^{-inpu\ \ sum}}$$

$$h_j = \frac{1}{1 + e^{-\sum_{i=0}^{A} w1_{ij} x_i}}$$

$$o_j = \frac{1}{1 + e^{-\sum_{i=0}^{B} w2_{ij} h_i}}$$



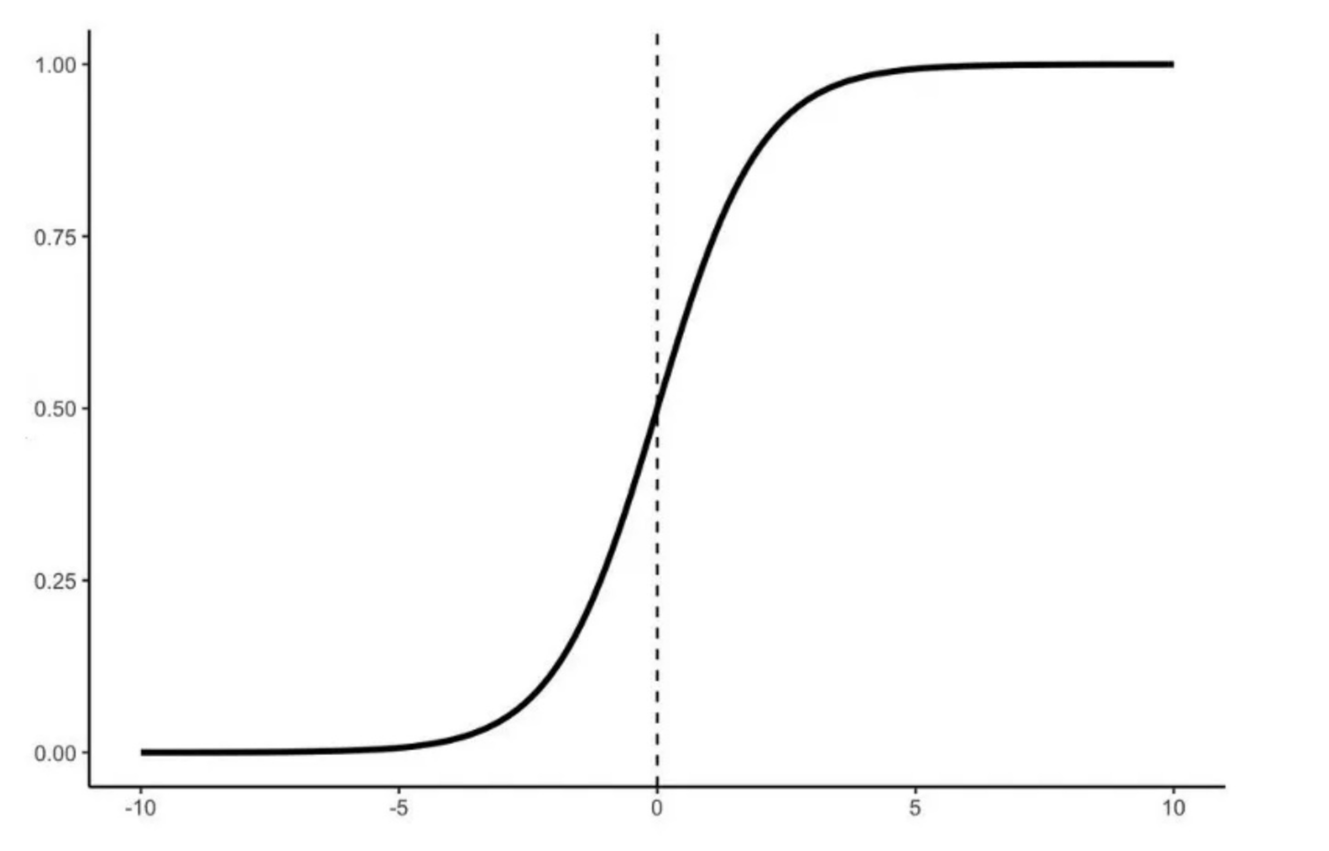




0.8518518805503845

Validating Results from ANN Model

Logistic Regression



Sigmoid Function: $$S(x) = \frac{1}{1+e^{-x}}$$

$$P(X=1) = F(g(x)) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

- Linear Regression is used when we want to model the relationship between one discrete, dependent variable whose outcome is binary (0,1), given 1 or more continuous or discrete independent variables.
- The logit function is a transformation of the linear regression function that maps the predicted values of the linear regression equation (R:[-∞,∞]) onto the log-odds scale (R:[0,1]).
- The logit function is defined as: logit(p) = ln(p / (1-p))

Further Comparison:

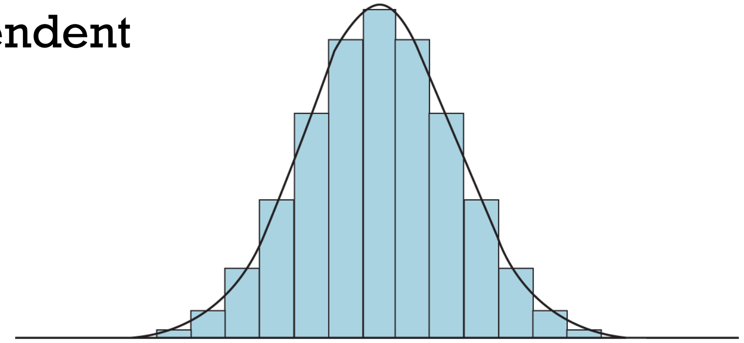
Linear Regression:

- Used to solve numerical problems
- (Y) = continuous
- Range(Y) = [-∞,∞]
- Error term distribution: assumed to be normal
- Beta1: change in the dependent variable for each one-unit change in the corresponding independent variable.

$$Y_i = \beta_0 + \beta_1 X_i$$

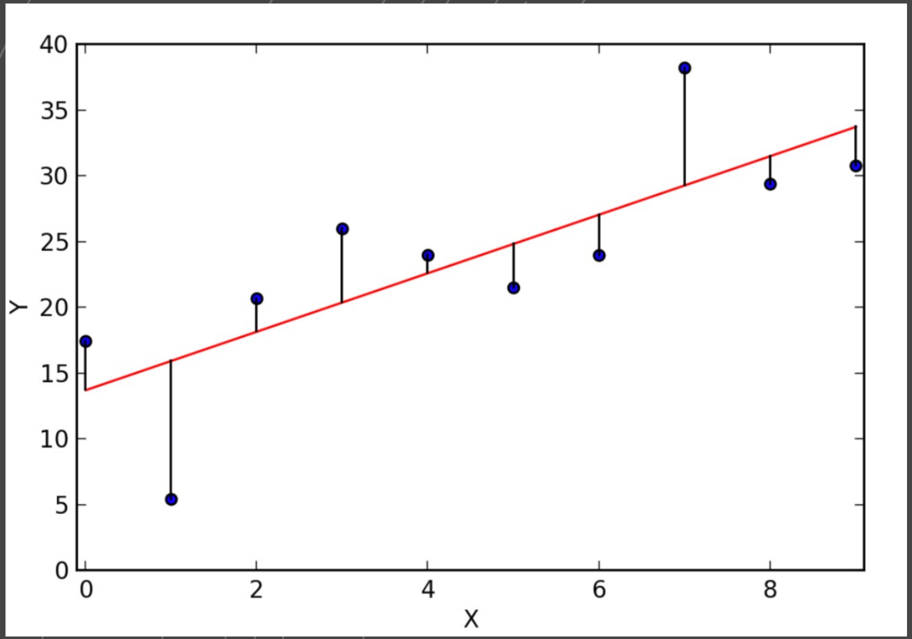
Logistic Regression:

- Used to solve classification problems
- (Y) = discrete, binary
- Range(Y) = [0,1]
- Error term distribution is assumed to be binomial
- Beta1: change in the log odds of the binary outcome for each one-unit change in the corresponding independent variable


Calcworkshop.com

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Least-Squares Method and Algorithm



$$S = \sum_{i=1}^{n} {d_i}^2$$

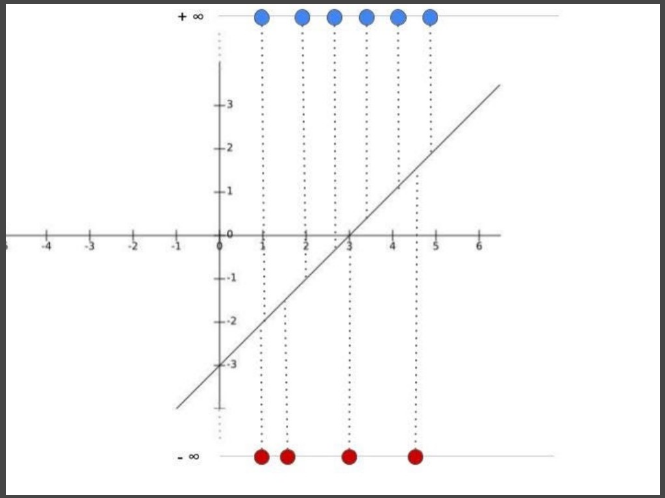
$$S = \sum_{i=1}^{n} [y_i - f_{x_i}]^2$$

$$S = {d_1}^2 + {d_2}^2 + {d_3}^2 + \cdots + {d_n}^2$$

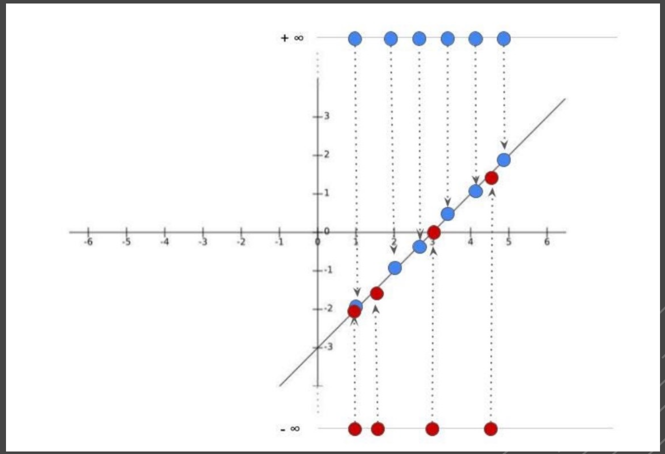
Visual Explanation of Maximum-Likelihood Estimation:

$$log(odds) = log\left(\frac{p}{1-p}\right) = logit(p)$$

1.) The log(odds) line is plotted and returns values between [-∞,∞].



2.) These values are mapped onto the log(odds) line:



$$p = \frac{e^{log(odds)}}{1 + e^{log(odds)}}$$

- 3.) The log(odds) values are transformed into probability values (between 0-1):

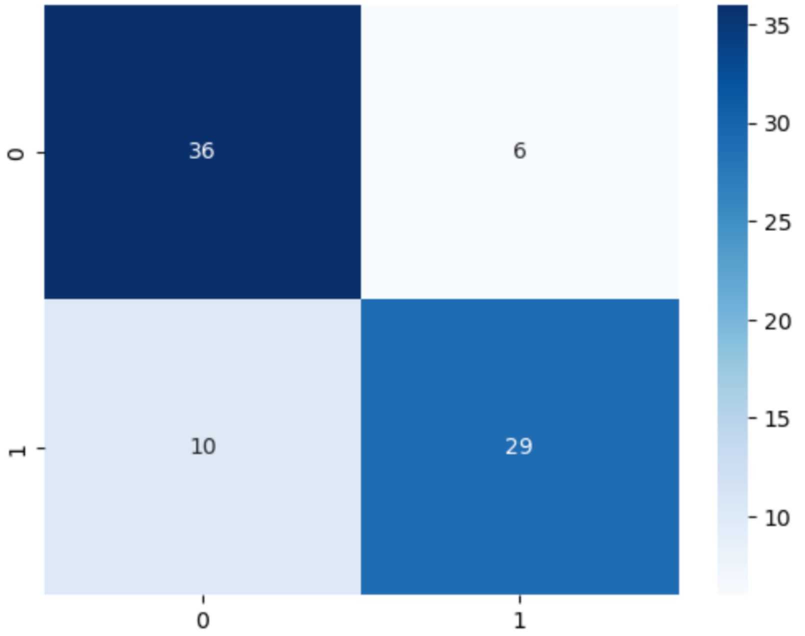
- 4.) The probability values are mapped onto the sigmoid curve (1st iteration)
- 5.) We repeat steps 1-4, but each time we change/rotate the slope of the log(odds) line (360 degrees), until we have maximized the log-likelihood of the values on the sigmoid curve.

  - Any value in the range of 0 to 0.5 is classified as 0 and 0.5 to 1 is classified as 1.
  - (The above is true when our threshold value along the sigmoid curve is equal to .5)

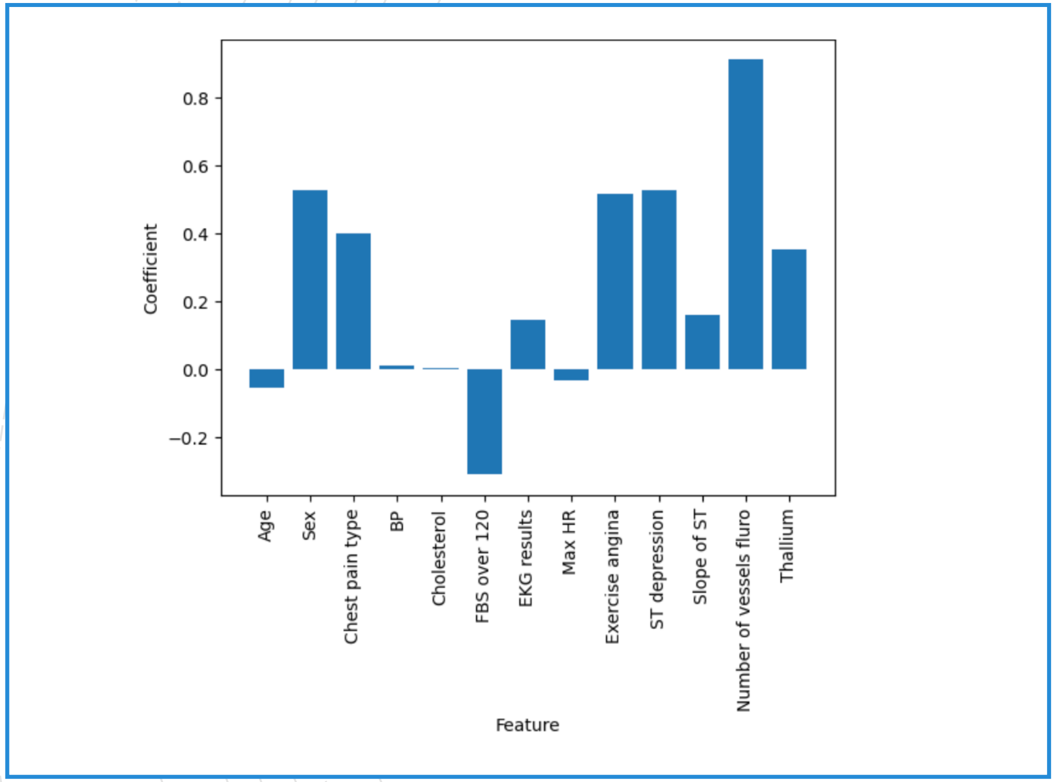
Results and Performance of Logistic Regression Model

```
[7]: import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

# read data
df = pd.read_csv("/Users/david/Desktop/Heart_Disease_Prediction.csv")

# set variable names
col_names = ['Age', 'Sex', 'Chest Pain Type', 'Blood Pressure', 'Cholesterol', 'FBS over 120', 'EKG results', 'Max HR',
             'Exercise angina', 'ST depression', 'Slope of ST', 'Number of vessels fluro', 'Thallium', 'Heart Disease']
xcol_names = ['Age', 'Sex', 'Chest Pain Type', 'Blood Pressure', 'Cholesterol', 'FBS over 120', 'EKG results', 'Max HR',
             'Exercise angina', 'ST depression', 'Slope of ST', 'Number of vessels fluro', 'Thallium']

# create a dictionary to map the "yes" and "no" values to 1's and 0's respectively
mapping = {'Presence': 1, 'Absence': 0}

# use the map() function to apply the mapping to the desired column
df['Heart Disease'] = df['Heart Disease'].map(mapping)

#Selecting the independent variables and the dependent variable (H.D.):
x = df.iloc[:, [1,2,3,4,5,6,7,8,9,10,11,12,13]]
Y = df.iloc[:,[14]]

# split data into training and testing sets
X = x
y = Y
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# instantiate logistic regression model and fit to training data
lr = LogisticRegression()
lr.fit(X_train, y_train)

# make predictions on test data
y_pred = lr.predict(X_test)

# evaluate performance of the model
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
print('Accuracy:', accuracy_score(y_test, y_pred))
print('Precision:', precision_score(y_test, y_pred))
print('Recall:', recall_score(y_test, y_pred))

Accuracy: 0.9074074074074074
Precision: 0.9444444444444444
Recall: 0.8095238095238095
F1-score: 0.8717948717948718
```

Accuracy: 0.9074074074074074
Precision: 0.9444444444444444
Recall: 0.8095238095238095
F1-score: 0.8717948717948718

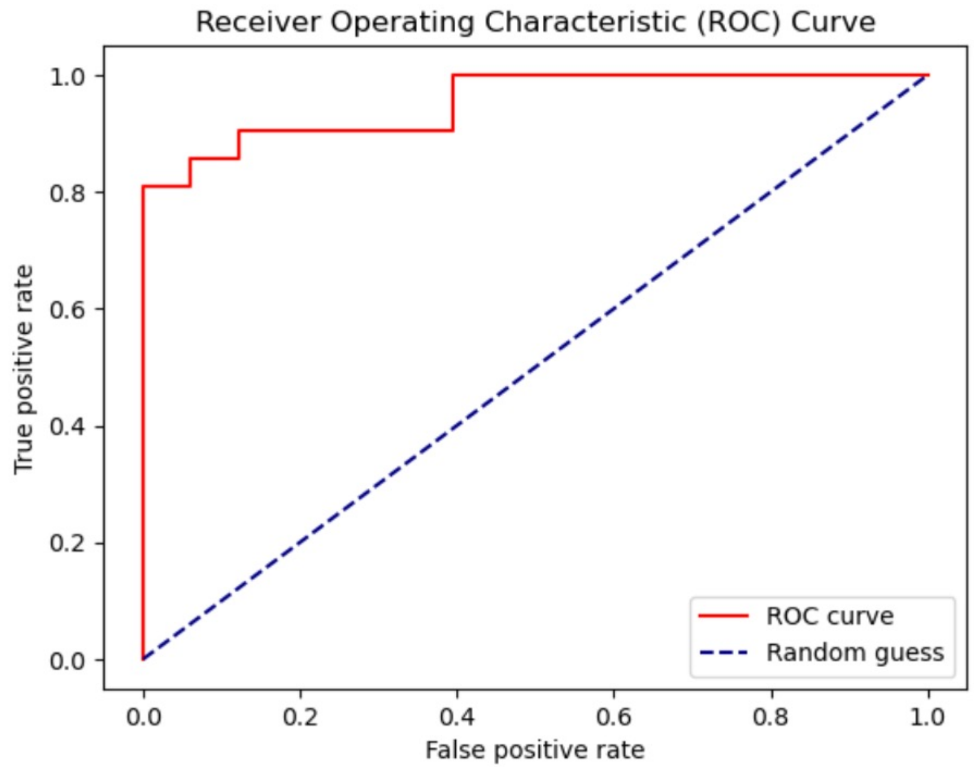


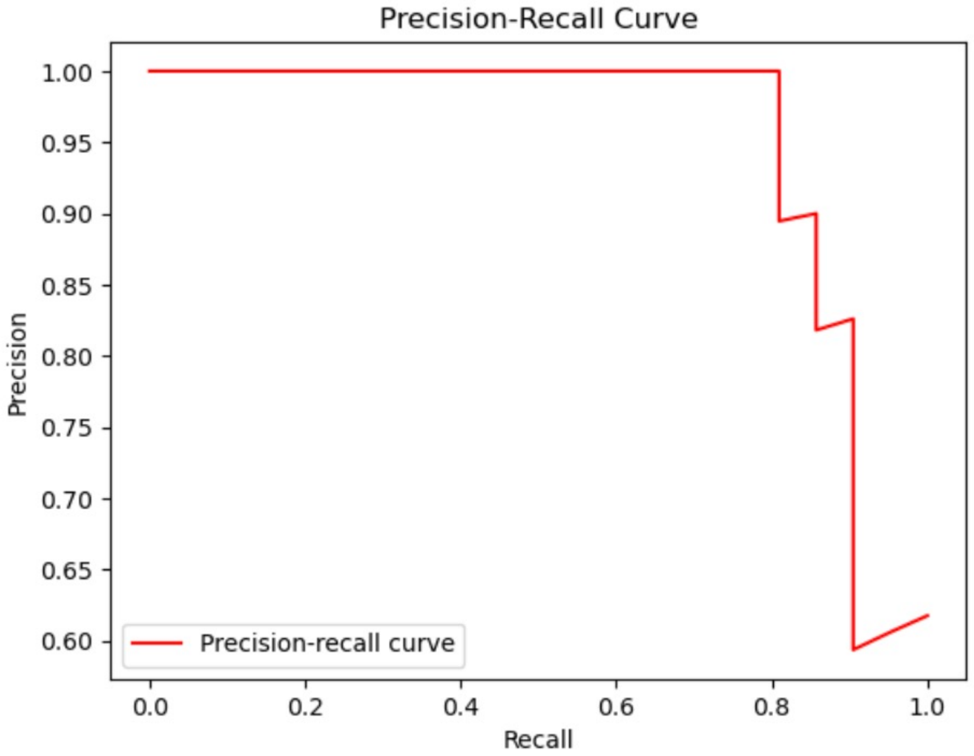
Validating the Logistic Regression Model: Feature Importance Plot



- provide a way to visualize the importance of each feature
- helps identify which features are most significant predictors

Validating the Logistic Regression Model: ROC (Receiving Operating Characteristic) Curve and Precision-Recall Curve






```
ROC AUC: 0.9538239538239539
Precision-Recall AUC: 0.9499902200669466
```

ANN Model Final Results:

```
Accuracy: 0.7703703703703704
Precision: 0.7636363636363637
Recall: 0.7
F1 score: 0.7304347826086957
```

Logistic Regression Model Final Results:

```
Accuracy: 0.9074074074074074
Precision: 0.9444444444444444
Recall: 0.8095238095238095
F1-score: 0.8717948717948718
```

Comparing results of the ANN vs. Logistic Regression: