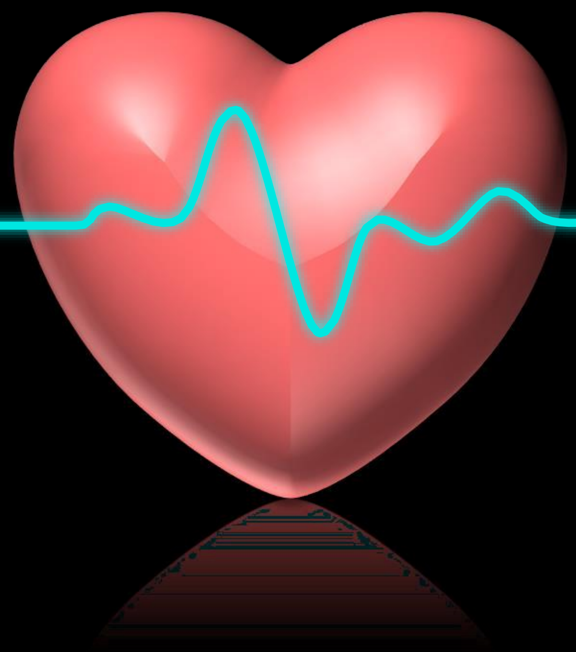
Cardiac Arrest Prediction



Data Mining (MGSC-5126 -10)

Arjun Thakur - 0271741

Introduction

Objective: To identify the variables that influence cardiovascular disease and use those variables to predict whether a patient will have any kind of cardiovascular disease.

Software used: Jupyter Notebook for Python

Method: Binary Classification (0/1)

Dataset Link : https://www.kaggle.com/code/sulianova/eda-cardiovascular-data/notebook?scriptVersionId=9722310

Input Variables:

ID
AGE
GENDER
HEIGHT
WEIGHT
SYSTOLIC_BP
DIASTOLIC_BP
CHOLESTEROL_LEVEL
GLUCOSE_LEVEL
SMOKER
ALCOHOL_CONSUMER
PHYSICAL_ACTIVITY

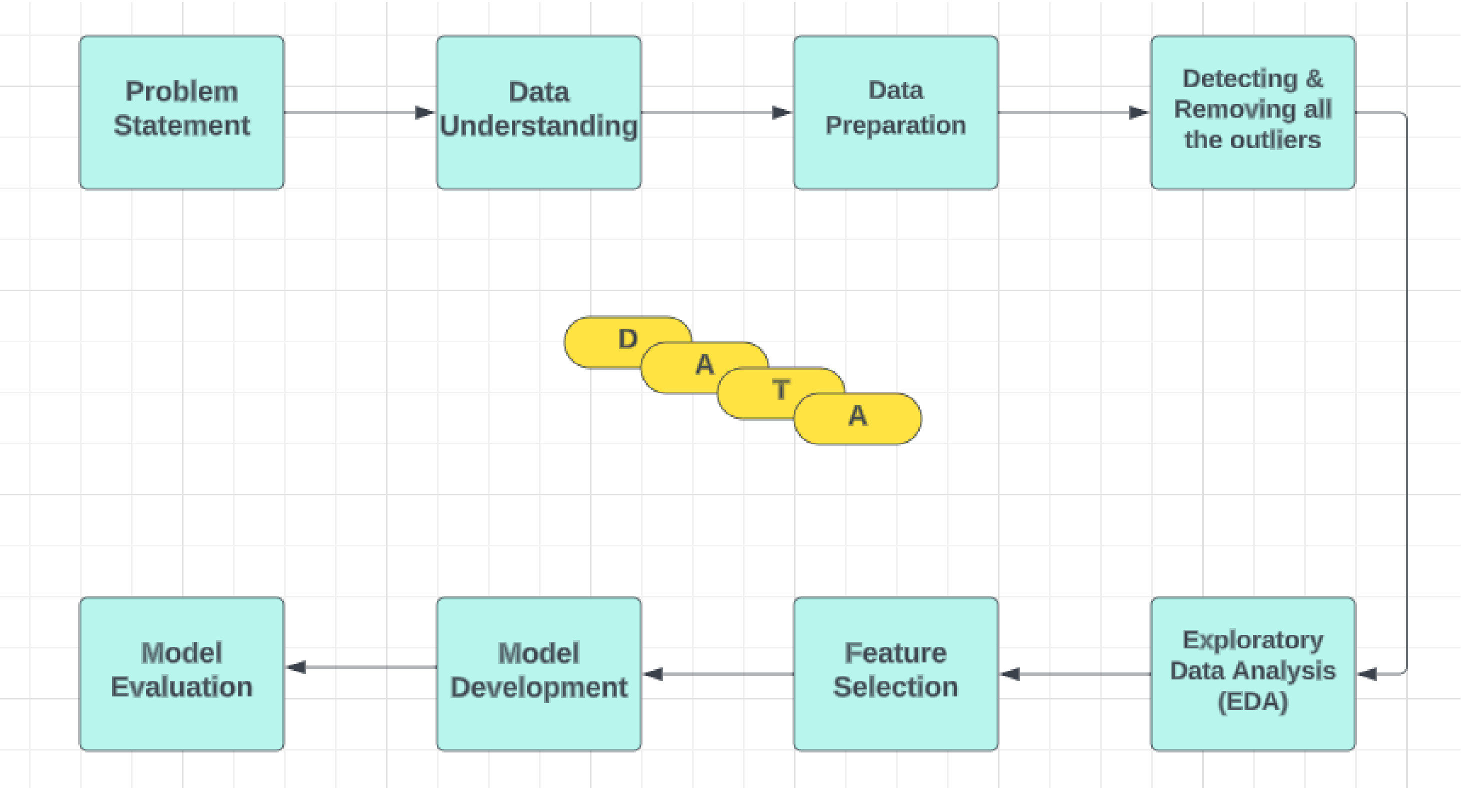
Target Variable : CARDIOVASCULAR DISEASE

Where class 0 : Absence of Cardiovascular Disease

class 1 : Presence of Cardiovascular Disease

Methodology

➢ Given below is the flowchart representing the 8 steps involved in the Data Mining Project.



Cardiovascular Disease Classification Steps Flowchart

Data Preparation

- ➢ Importing all the necessary libraries
- ➢ Loading the csv file dataset into the program (70000, 13)
- ➢ Data Pre-Processing & Cleaning along with Data Integration

```
data_raw.describe()
```

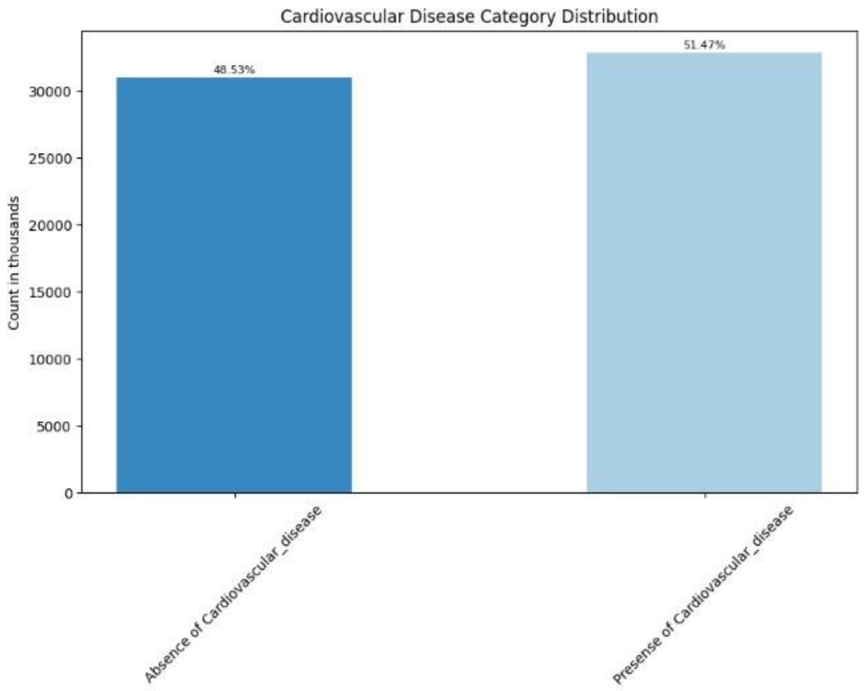
	AGE	GENDER	HEIGHT	WEIGHT	SYSTOLIC_BP	DIASTOLIC_BP	CHOLESTEROL_LEVEL	GLUCOSE_LEVEL	SMOKER	ALCOHOL_CONSUME
count	66801.000000	66801.000000	66801.000000	66801.000000	66801.000000	66801.000000	66801.000000	66801.000000	66801.000000	66801.00000
mean	52.825991	1.356252	164.343707	74.523705	129.255550	97.446415	1.382599	1.236134	0.092124	0.05628
std	6.798112	0.478895	8.333752	14.579585	157.618539	192.892442	0.690087	0.582109	0.289204	0.23047
min	29.000000	1.000000	55.000000	10.000000	1.000000	0.000000	1.000000	1.000000	0.000000	0.00000
25%	48.000000	1.000000	159.000000	65.000000	120.000000	80.000000	1.000000	1.000000	0.000000	0.00000
50%	53.000000	1.000000	165.000000	72.000000	120.000000	80.000000	1.000000	1.000000	0.000000	0.00000
75%	58.000000	2.000000	170.000000	83.000000	140.000000	90.000000	2.000000	1.000000	0.000000	0.00000
max	64.000000	2.000000	250.000000	200.000000	16020.000000	11000.000000	3.000000	3.000000	1.000000	1.00000

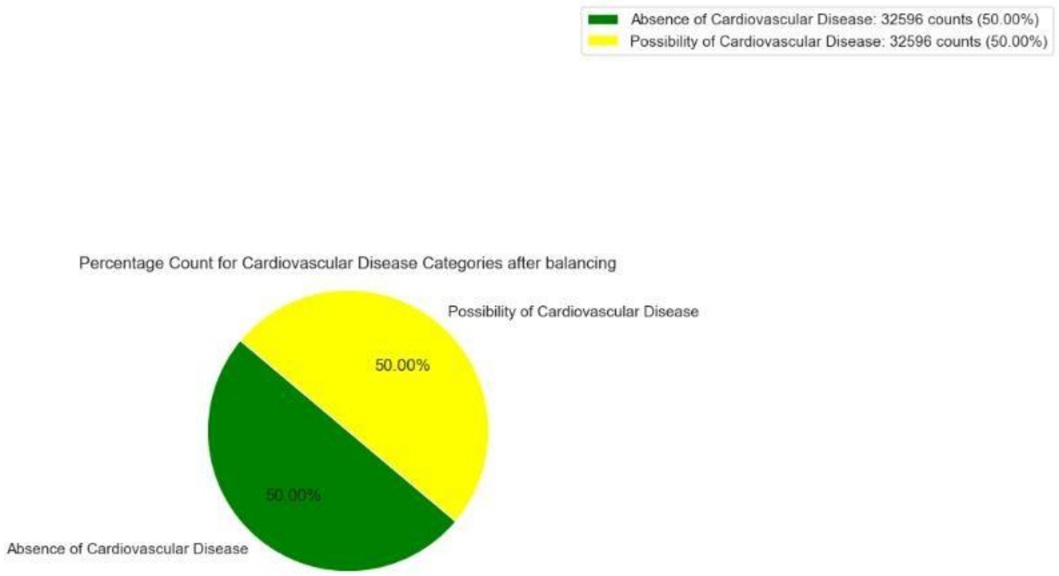
- Removing Irrelevant & Unwanted Redundant Features (1)
- Renaming the attribute names (13)
- Dropping duplicate cases in dataset (3191)
- Handling the missing values
- Removing columns with negative values for SYSTOLIC_BP and DIASTOLIC_BP (8)
- Finding Summary Statistics, Skewness, Kurtosis, Scatterplot to understand the data

```
Column: AGE
Kurtosis: -0.85
Kurtosis Classification: Fatter-than-normal distribution (platykurtic)

Column: WEIGHT
Kurtosis: 2.02
Kurtosis Classification: Fatter-than-normal distribution (platykurtic)

Column: SYSTOLIC_BP
Kurtosis: 5.09
Kurtosis Classification: Skinnier-than-normal distribution (leptokurtic)

Column: DIASTOLIC_BP
Kurtosis: 5.64
Kurtosis Classification: Skinnier-than-normal distribution (leptokurtic)

Column: CHOLESTEROL_LEVEL
Kurtosis: 0.78
Kurtosis Classification: Fatter-than-normal distribution (platykurtic)

Column: GLUCOSE_LEVEL
Kurtosis: 3.85
Kurtosis Classification: Skinnier-than-normal distribution (leptokurtic)

Column: PHYSICAL_ACTIVITY
Kurtosis: 0.20
Kurtosis Classification: Fatter-than-normal distribution (platykurtic)

Column: CARDIAC_ARREST
Kurtosis: -2.00
Kurtosis Classification: Fatter-than-normal distribution (platykurtic)
```

```
Column: AGE
Skewness: -0.27
Mean: 53.20
Median: 54.00
Mode: 55
Skewness Direction: Negatively Skewed

Column: WEIGHT
Skewness: 0.93
Mean: 74.51
Median: 72.00
Mode: 70
Skewness Direction: Positively Skewed

Column: SYSTOLIC_BP
Skewness: 0.08
Mean: 126.85
Median: 120.00
Mode: 120
Skewness Direction: Positively Skewed

Column: DIASTOLIC_BP
Skewness: 0.37
Mean: 81.53
Median: 80.00
Mode: 80
Skewness Direction: Positively Skewed

Column: CHOLESTEROL_LEVEL
Skewness: -0.03
Mean: 0.98
Median: 1.00
Mode: 1
Skewness Direction: Negatively Skewed
```

Balancing the Target Variable





Before Balancing

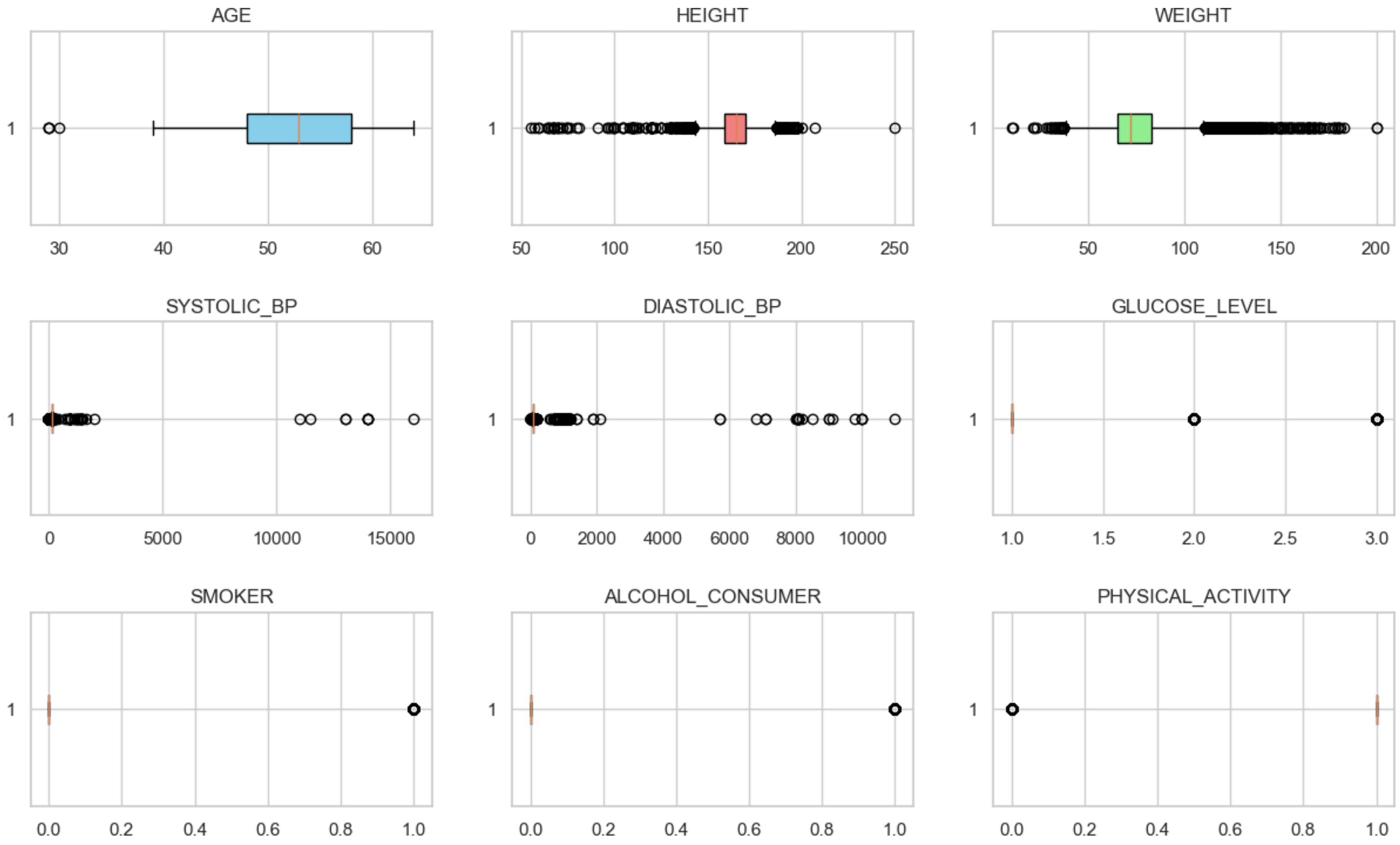
Undersampling

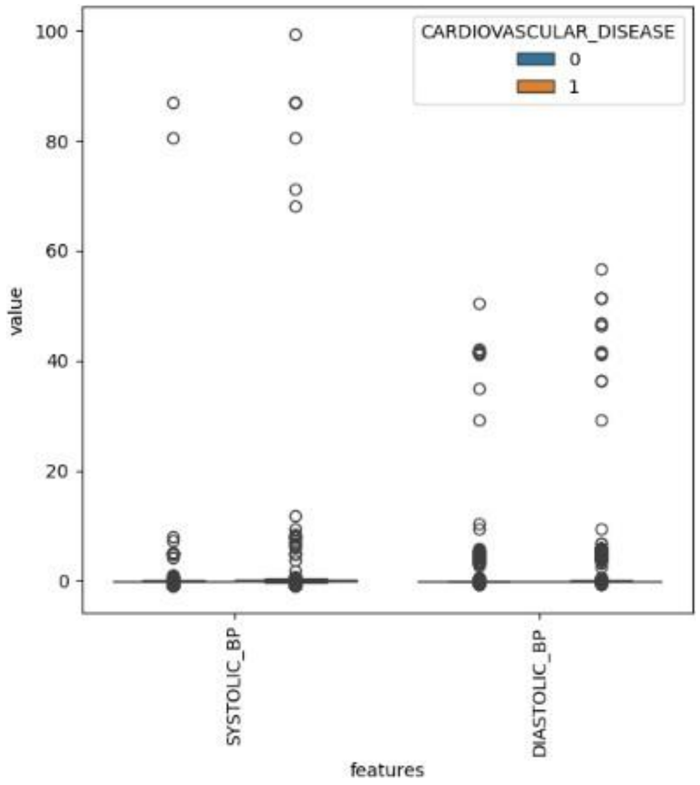
After Balancing

Detecting Outliers

Identify and handle outliers in the data.

- Initially, there were 66801 rows and 12 columns.
- After fixing the outliers there are 63828 rows and 12 columns



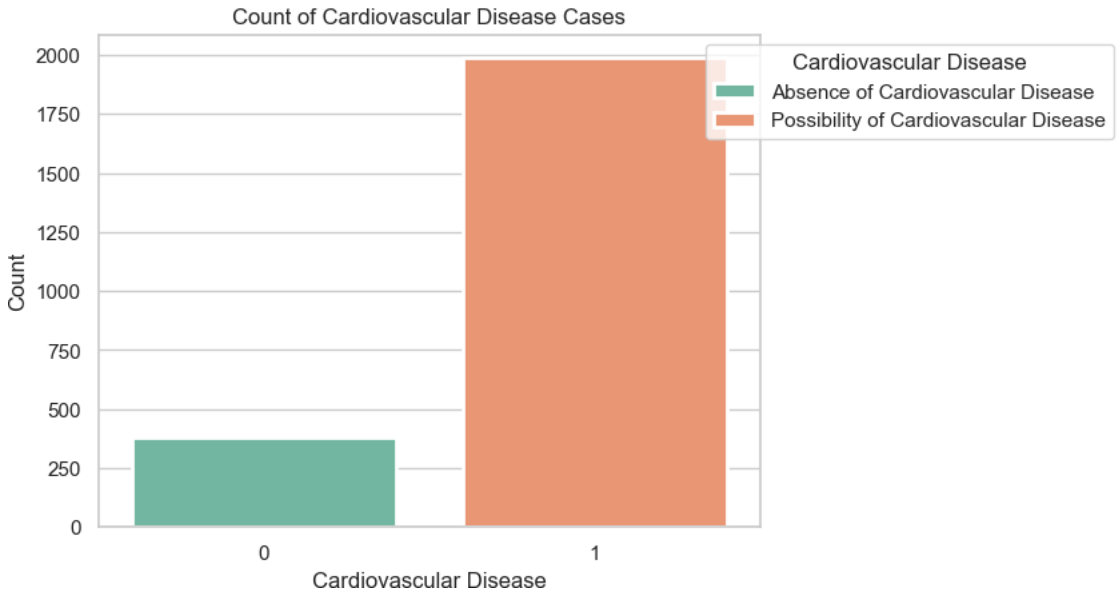


Fixing the Outliers

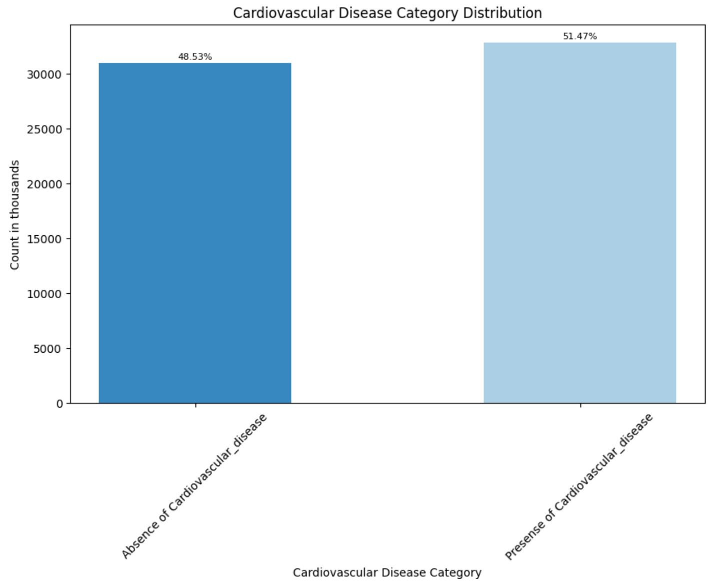
- AGE Attribute: Consider data only for ages between 40 and 65
- HEIGHT Attribute: Consider data only 4.5 ft to 6.5ft 'HEIGHT' that is between 140 cm and 200 cm
- WEIGHT Attribute: only between 40 and 180
- SYSTOLIC BP & DIASTOLIC BP Attribute
  - Found upper and lower acceptable limits and removed all the values that don’t satisfy the condition
  - Removed any other extreme outlier values

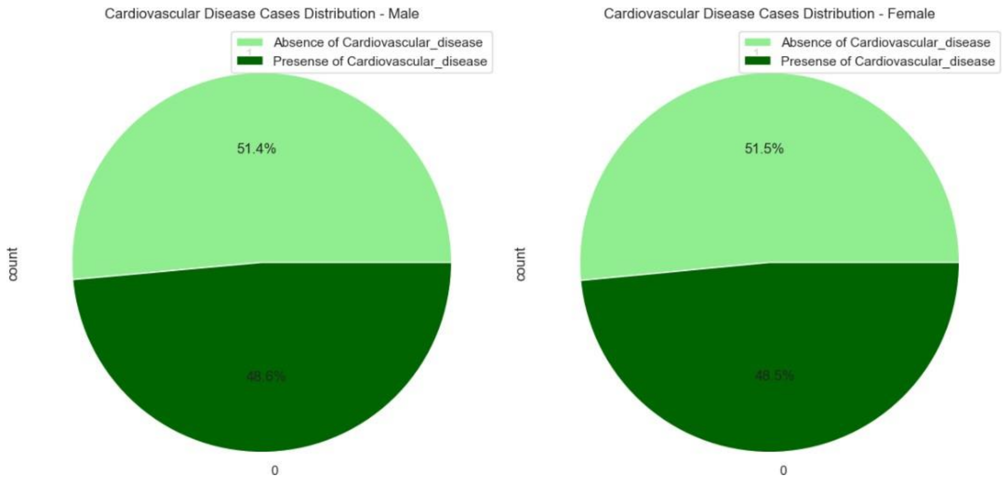
	SYSTOLIC_BP	DIASTOLIC_BP
lower_bound	90.0	65.0
upper_bound	170.0	105.0

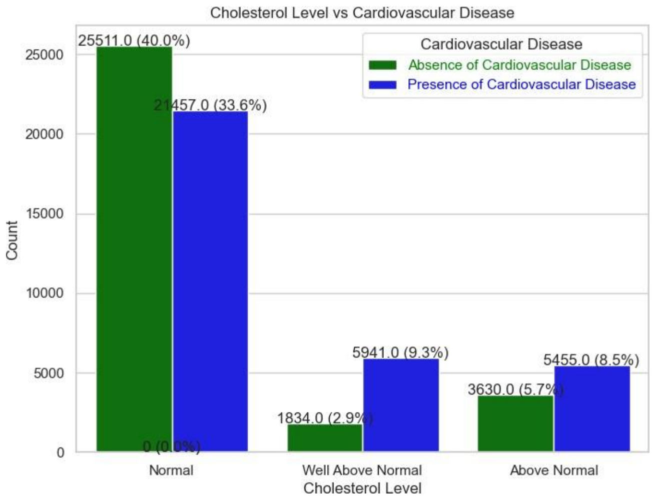
We can select the index of outlier data by using boundaries we calculated.

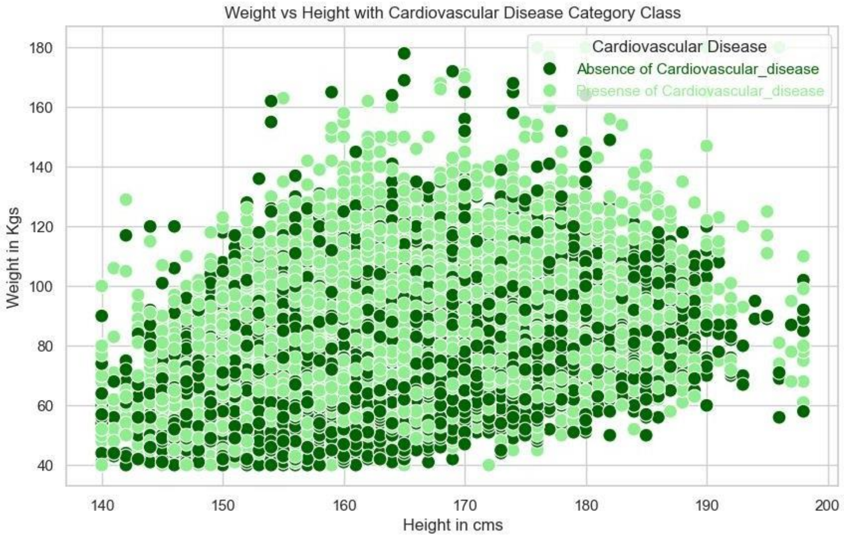


Exploratory Data Analysis



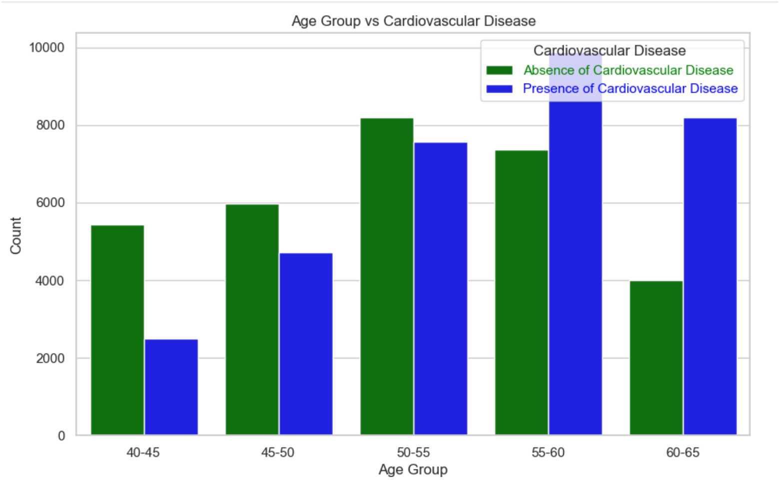


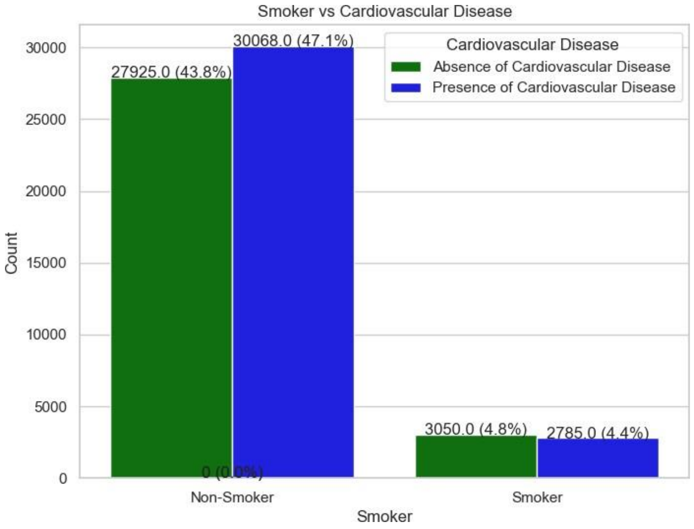


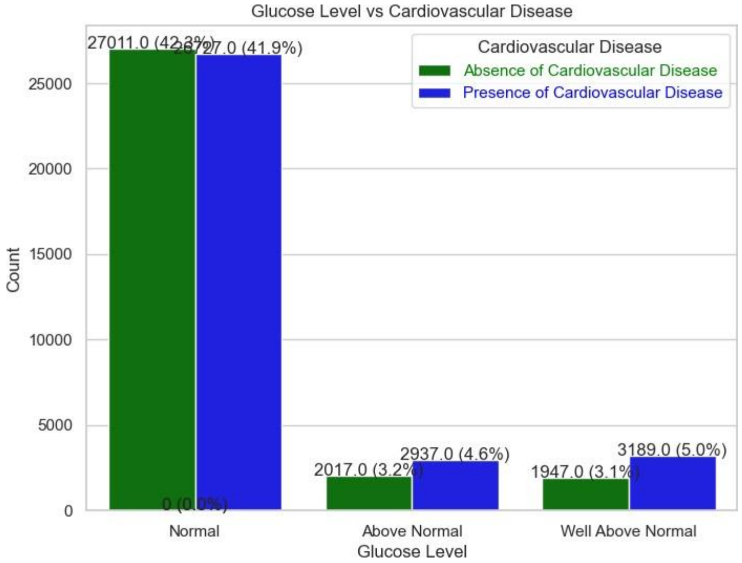


Exploratory Data Analysis

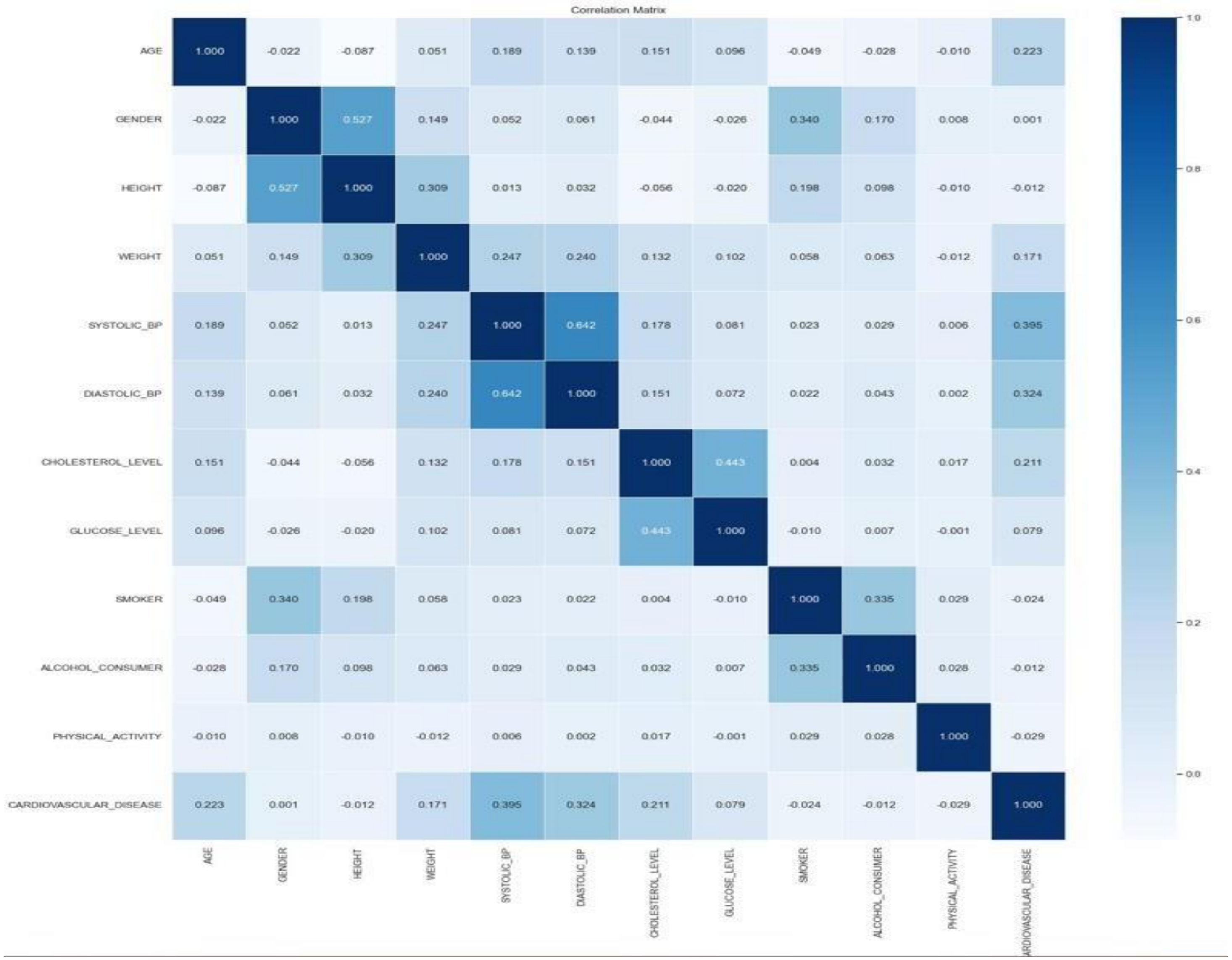
- We used EDA to to understand the structure and patterns in the data, identify outliers, and gain insights that can inform further analysis or modeling.
- Checking for the importance of AGE, SMOKER, GLUCOSE_LEVEL variable with the target variable
- CARDIOVASCULAR_DISEASE with graphical data analysis.







Correlation Matrix as part of Feature Selection



Finding the most important features using Decision Tree and Chi-Square Statistic

- Based on the correlation matrix, decision tree classifier and Chi-Square Statistic below methods, we were able to Drop 3 attributes namely GENDER, HEIGHT and ALCOHOL_CONSUMER.
- Thus, we are left with 8 input variables and 1 target variable for our binary classification of ‘0’ and ‘1’

	feature	importance
0	AGE	0.149973
1	GENDER	0.026827
2	HEIGHT	0.207259
3	WEIGHT	0.224206
4	SYSTOLIC_BP	0.232271
5	DIASTOLIC_BP	0.052895
6	CHOLESTEROL_LEVEL	0.037146
7	GLUCOSE_LEVEL	0.026205
8	SMOKER	0.012182
9	ALCOHOL_CONSUMER	0.010074
10	PHYSICAL_ACTIVITY	0.020962

Decision Tree Classifier

	Attribute	Score
4	SYSTOLIC_BP	25505.735627
5	DIASTOLIC_BP	8310.500762
3	WEIGHT	5182.963003
0	AGE	2509.920083
6	CHOLESTEROL_LEVEL	981.527952
7	GLUCOSE_LEVEL	110.627607
8	SMOKER	32.708920
10	PHYSICAL_ACTIVITY	11.160946
9	ALCOHOL_CONSUMER	8.879119
2	HEIGHT	3.462008
1	GENDER	0.005211

Chi-Square Statistic

Normalization

- **Standard Scaler Normalization –**
- To scale numerical features to a standard range.
- Used to ensure that all features have similar scales
- Helps algorithms converge faster and perform better, especially those sensitive to the scale of input features.
- With a mean of 0 and a standard deviation of 1.
- Range of -3 to 3
- Assuming normal distribution.
- To improve model performance by ensuring consistent feature scales.

- We used it for studying SYSTOLIC_BP and DIASTOLIC_BP variables to check for outliers.

	AGE	GENDER	HEIGHT	WEIGHT	SYSTOLIC_BP	DIASTOLIC_BP	CHOLESTEROL_LEVEL	GLUCOSE_LEVEL	SMOKER	ALCOHOL_CONSUMER	PHYSICAL_ACTIVITY	CARI
0	50	2	168	62	-0.122187	-0.091800	1	1	0	0	1	
1	55	1	156	85	0.065351	-0.039823	3	1	0	0	1	
2	51	1	165	64	0.002839	-0.143778	3	1	0	0	0	
3	48	2	169	82	0.127864	0.012155	1	1	0	0	1	
4	47	1	156	56	-0.184700	-0.195755	1	1	0	0	0	

- And for Standardizing the 8 input variables in 'X' while keeping the target variable 'Y' the same.

	AGE	WEIGHT	SYSTOLIC_BP	DIASTOLIC_BP	CHOLESTEROL_LEVEL	GLUCOSE_LEVEL	SMOKER	PHYSICAL_ACTIVITY
0	50	62	110	80	1	1	0	1
1	55	85	140	90	3	1	0	1
2	51	64	130	70	3	1	0	0
3	48	82	150	100	1	1	0	1
4	47	56	100	60	1	1	0	0

	AGE	WEIGHT	SYSTOLIC_BP	DIASTOLIC_BP	CHOLESTEROL_LEVEL	GLUCOSE_LEVEL	SMOKER	PHYSICAL_ACTIVITY
0	-0.493129	-0.869643	-0.934286	-0.152709	-0.556726	-0.407565	-0.3172	0.503117
1	0.277973	0.728595	0.729117	0.842587	2.328169	-0.407565	-0.3172	0.503117
2	-0.338909	-0.730666	0.174649	-1.148005	2.328169	-0.407565	-0.3172	-1.987610
3	-0.801570	0.520129	1.283584	1.837883	-0.556726	-0.407565	-0.3172	0.503117
4	-0.955791	-1.286575	-1.488753	-2.143301	-0.556726	-0.407565	-0.3172	-1.987610

Model Train-Test Split

- First we performed Data Partitioning using train_test_split from sklearn.model_selection to partition the dataset.
- Subsequently 60% of our data was allocated for Training Set, where input data is stored in X_train and target data in y_train that is major portion for model learning.
- The remaining 40 % data was divided, dedicating 20% to the Validation Set with X_val, y_val reserved for fine-tuning and optimization.
- Finally, the remaining 20% constituted the Testing Set with X_test, y_test, used for final evaluation of the model on unseen data.
- The Random State was set as 42 ensuring consistency with random state for reproducibility.

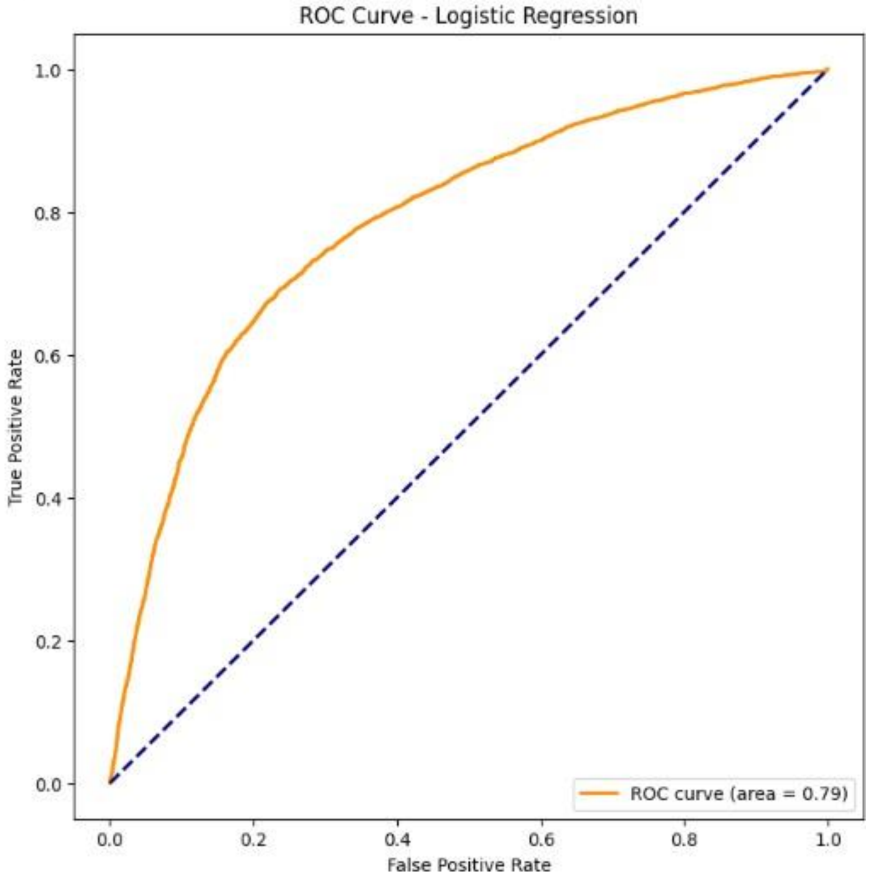
```
from sklearn.model_selection import train_test_split
```

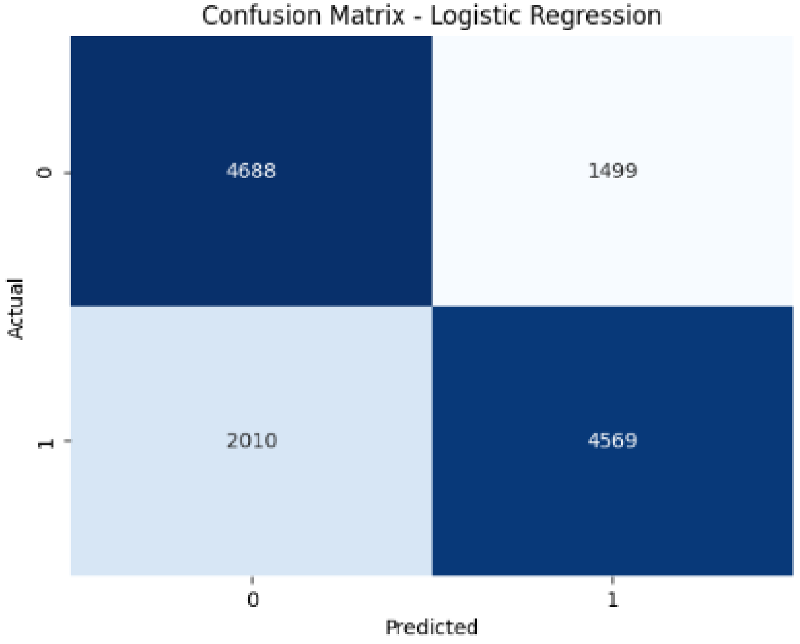
```
# Splitting the dataset into training and testing sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.4, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
# 60% of the data is used for training (X_train, y_train).
# 20% of the data is used for validation (X_val, y_val).
# 20% of the data is used for testing (X_test, y_test).
```

Model Evaluation & Assessment

1.Logistic Regression Model

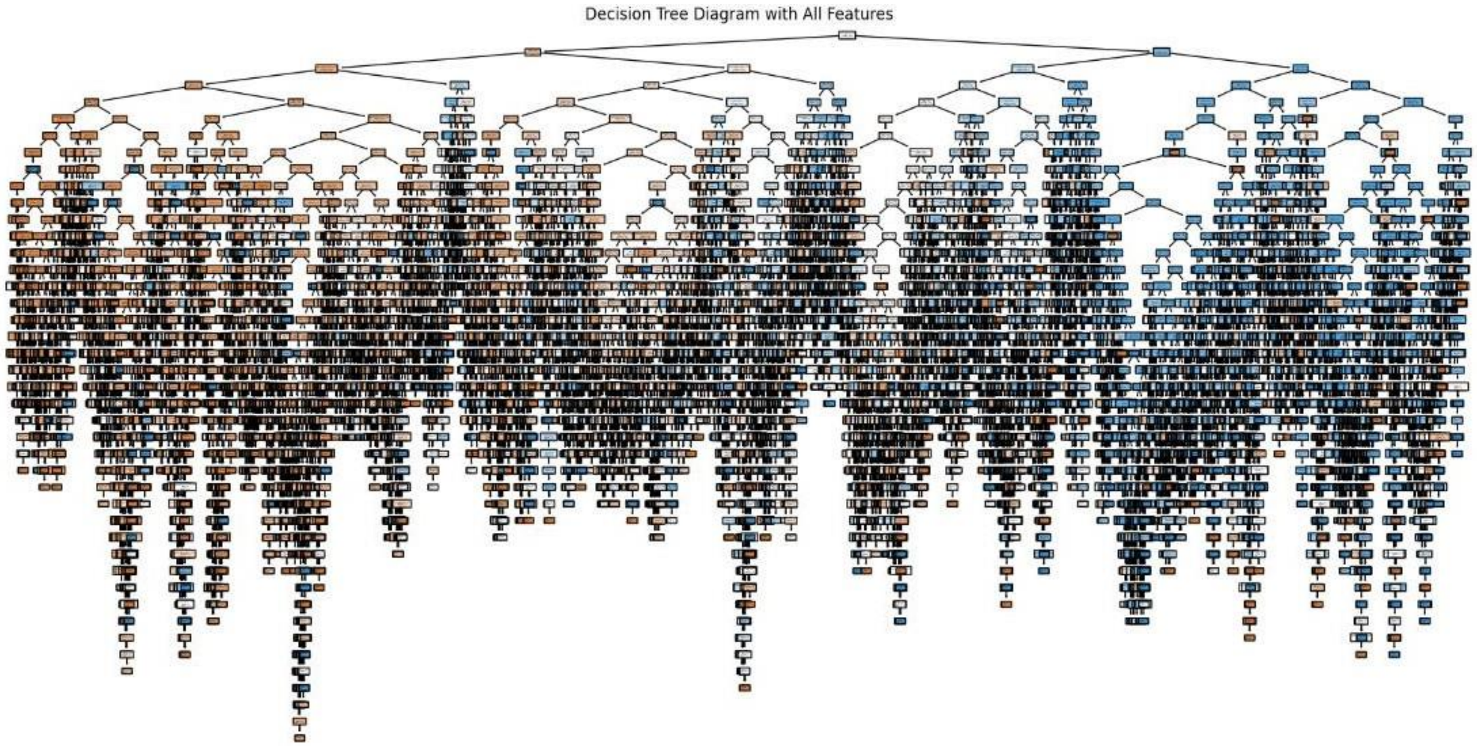
```
Evaluation metrics for Logistic Regression:
========================================
Accuracy: 0.7251
Precision: 0.7530
Recall: 0.6945
F1-Score: 0.7225
Cohen's Kappa: 0.4511
Log Loss: 0.5702
```

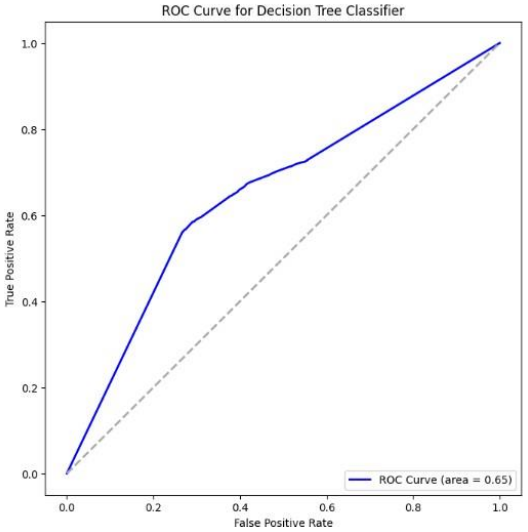


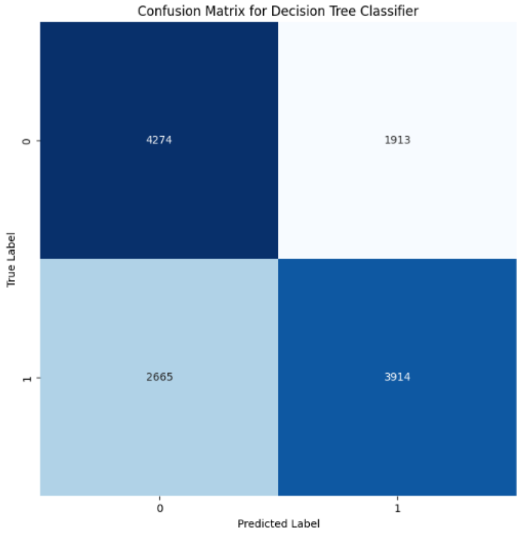


2. Decision Tree Classifier Model

Accuracy: 0.6414
Precision: 0.6717
Recall: 0.5949
F1-Score: 0.6310
Cohen's Kappa: 0.2847
Log Loss: 9.8178

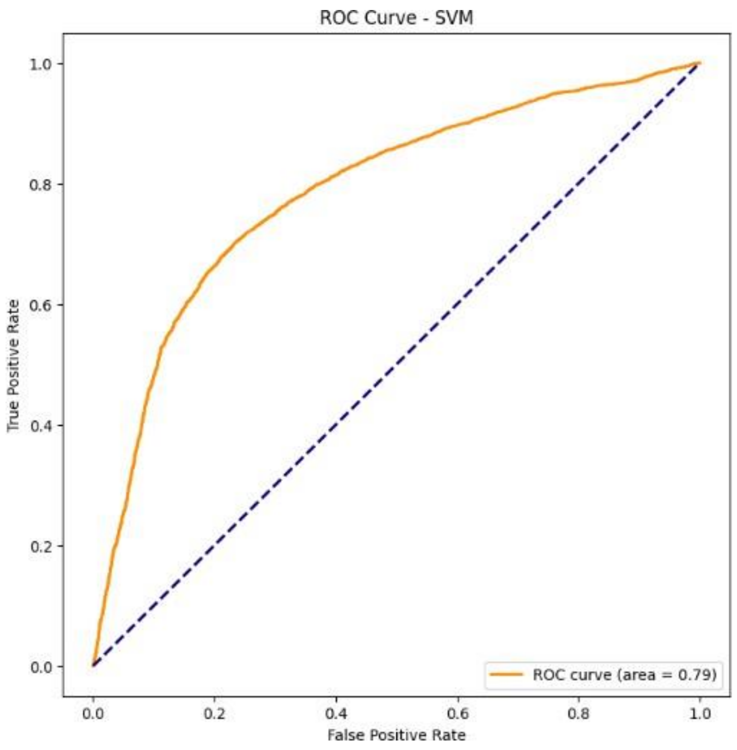


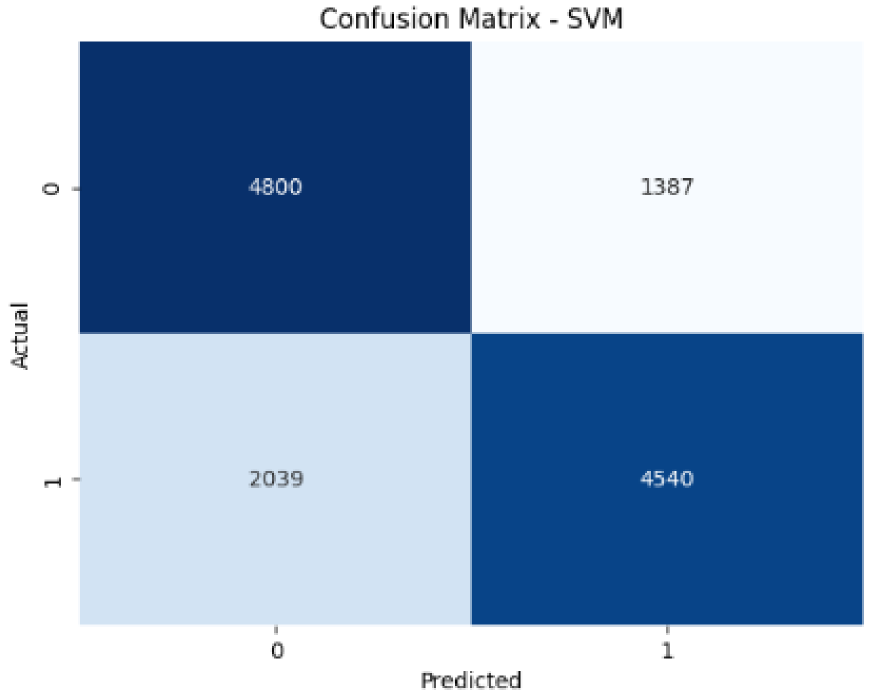




3. Support Vector Machine Model

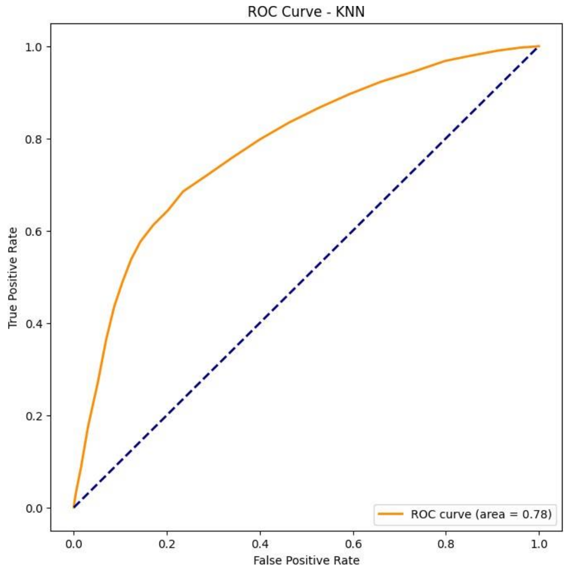
```
Number of support vectors: 22845

Evaluation metrics for SVM:
========================================
Accuracy: 0.7316
Precision: 0.7660
Recall: 0.6901
F1-Score: 0.7261
Cohen's Kappa: 0.4644
Log Loss: 0.5577
```

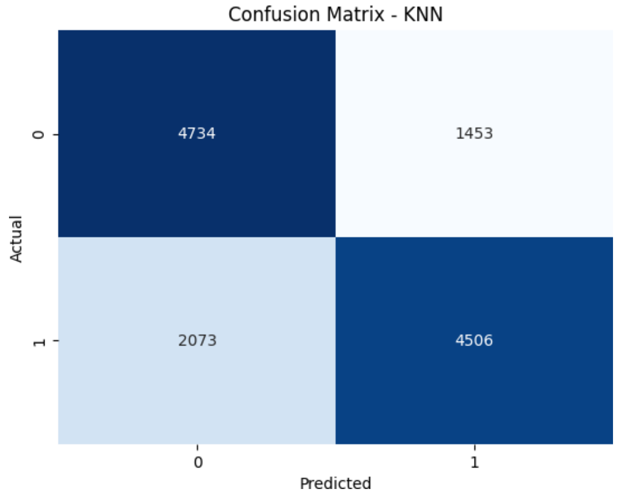


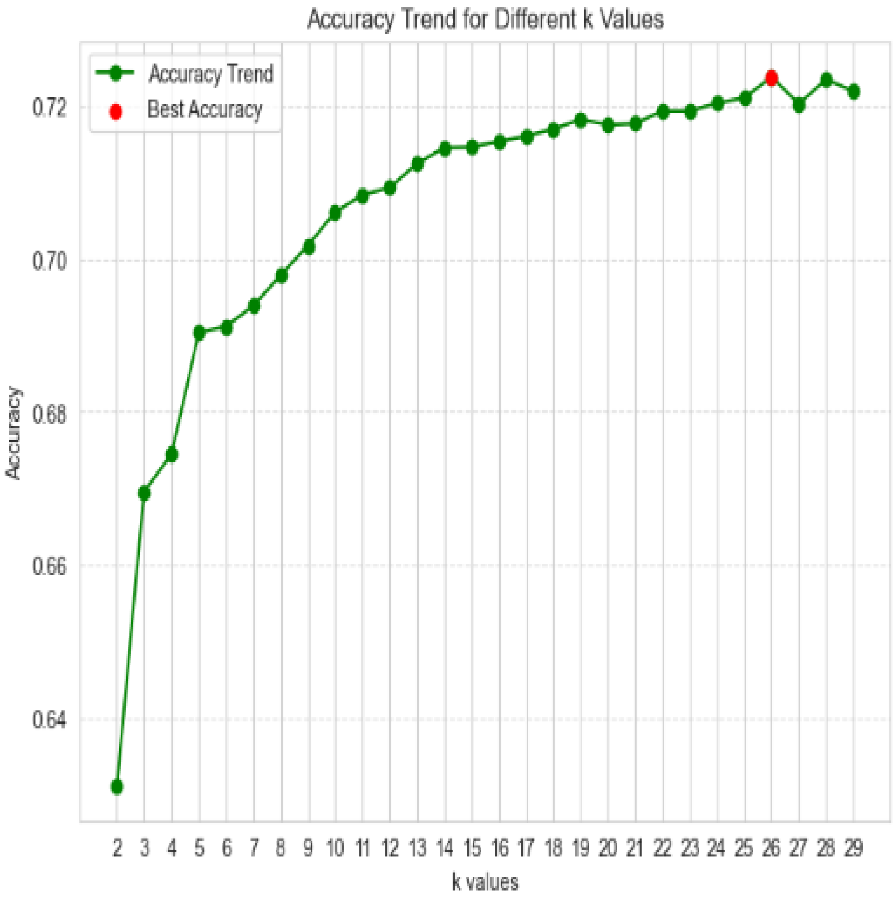


4. K- Nearest Neighbours Model

```
Accuracy for k = 2 is: 0.6312078959736801
Accuracy for k = 3 is: 0.6695911013629955
Accuracy for k = 4 is: 0.6745260849130503
Accuracy for k = 5 is: 0.690349365502115
Accuracy for k = 6 is: 0.691211029296569
Accuracy for k = 7 is: 0.6938743537521541
Accuracy for k = 8 is: 0.6978693404355318
Accuracy for k = 9 is: 0.7017859940466865
Accuracy for k = 10 is: 0.7060943130189566
Accuracy for k = 11 is: 0.7083659721134263
Accuracy for k = 12 is: 0.7093059689801035
Accuracy for k = 13 is: 0.7124392918690271
Accuracy for k = 14 is: 0.7145542848190506
Accuracy for k = 15 is: 0.7146326178912737
Accuracy for k = 16 is: 0.7153376155412815
Accuracy for k = 17 is: 0.7160426131912894
Accuracy for k = 18 is: 0.7169826100579665
Accuracy for k = 19 is: 0.7182359392135359
Accuracy for k = 20 is: 0.7175309415635281
Accuracy for k = 21 is: 0.7176876077079744
Accuracy for k = 22 is: 0.7193326022246592
Accuracy for k = 23 is: 0.7193326022246592
Accuracy for k = 24 is: 0.7203509321635595
Accuracy for k = 25 is: 0.7210559298135673
Accuracy for k = 26 is: 0.7237975873413756
Accuracy for k = 27 is: 0.7202725990913363
Accuracy for k = 28 is: 0.7234842550524831
Accuracy for k = 29 is: 0.7219175936080213
Best k: 26
Best accuracy: 0.7237975873413756

Evaluation metrics for KNN:
========================================
Accuracy: 0.7238
Precision: 0.7267
Recall: 0.7238
F1-Score: 0.7236
Cohen's Kappa: 0.4487
Log Loss: 0.5908
```



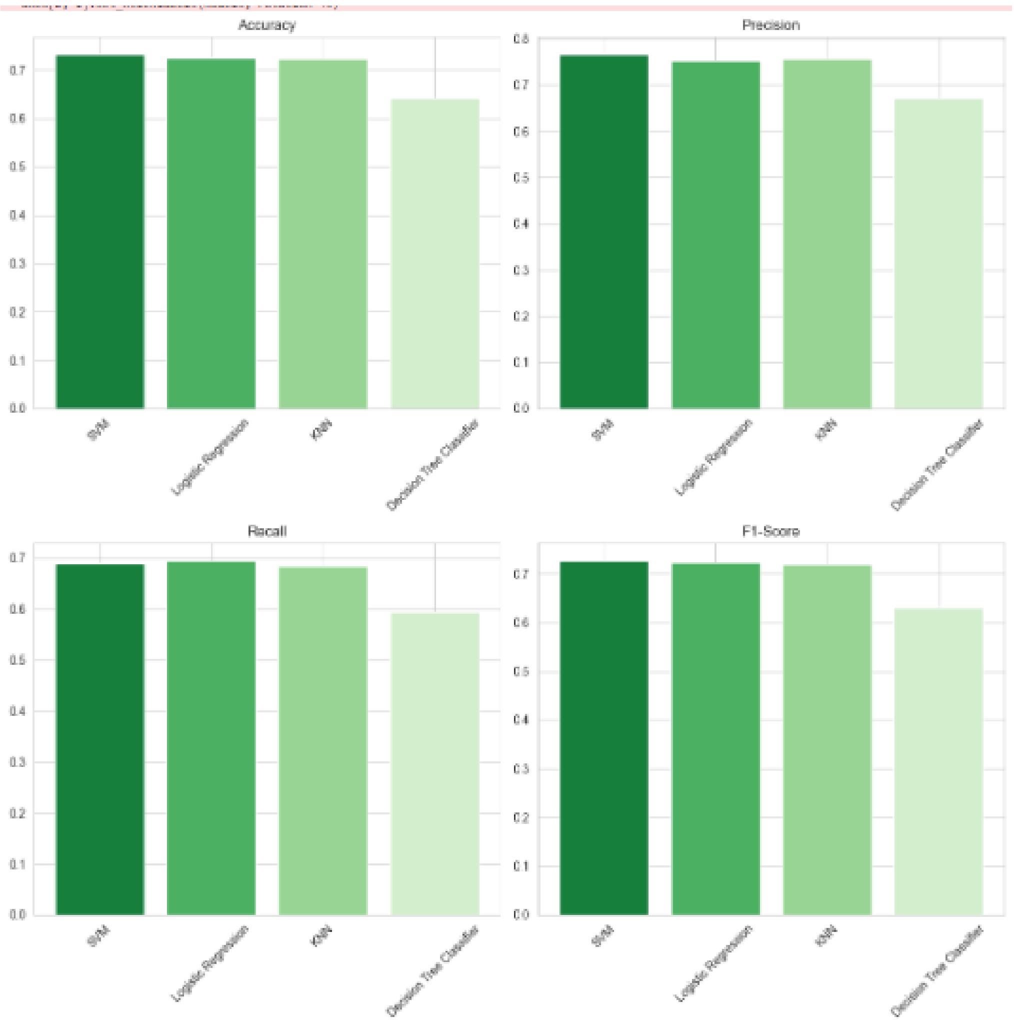


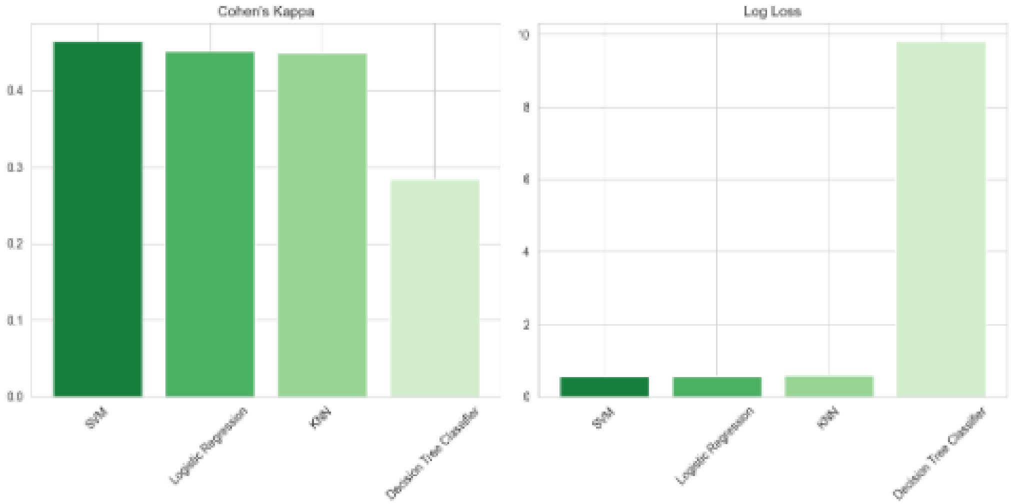


Model Comparison & Visualization

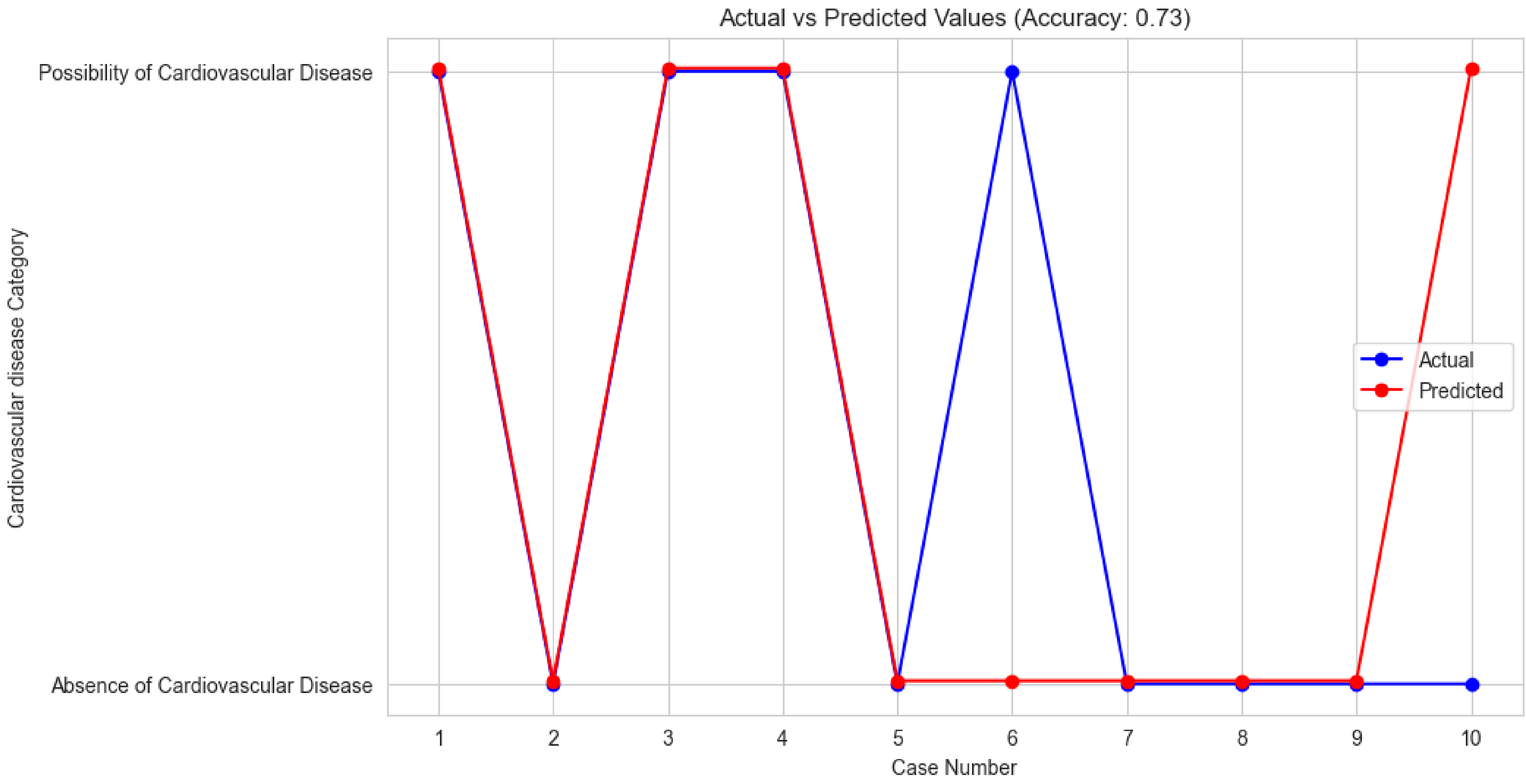
	Model	Accuracy	Precision	Recall	F1-Score	\
1	SVM	0.7316	0.7660	0.6901	0.7261	
2	Logistic Regression	0.7251	0.7530	0.6945	0.7225	
3	KNN	0.7238	0.7562	0.6849	0.7188	
4	Decision Tree Classifier	0.6414	0.6717	0.5949	0.6310	

	Cohen's Kappa	Log Loss
1	0.4644	0.5577
2	0.4511	0.5702
3	0.4487	0.5908
4	0.2847	9.8178





Prediction on unseen data using SVM Model



Thank You & Q&A