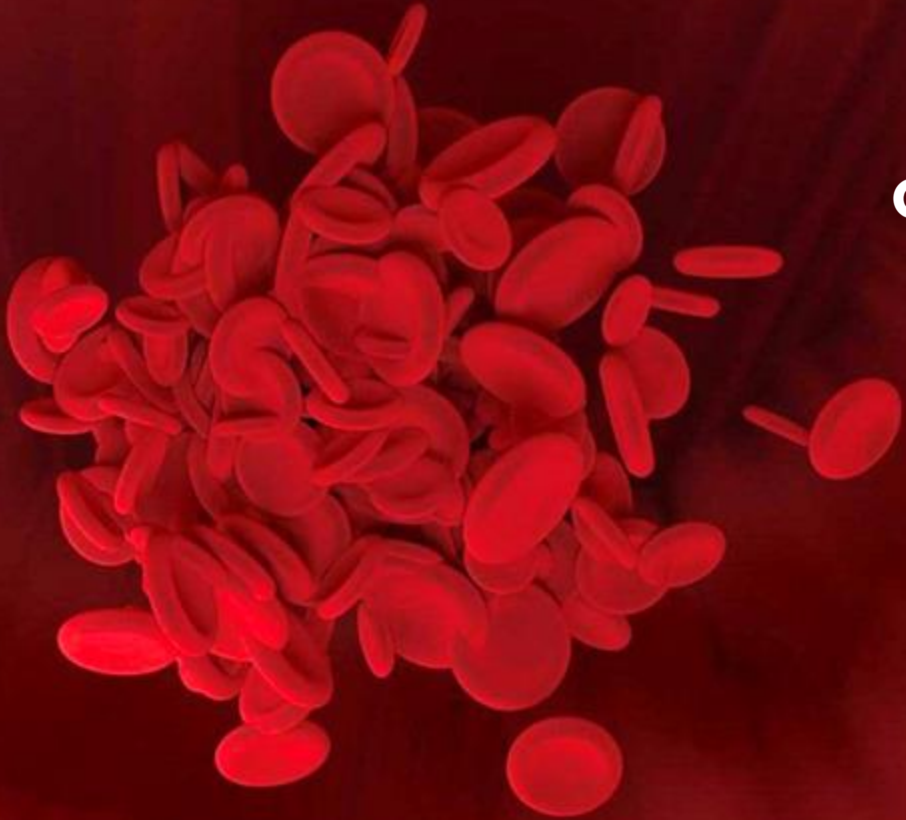
Data visualization & Predictive Analysis of Cardiovascular Disease

CIS 8695: Final Presentation

Abha Sharma, Ifeoluwa Kayode, Sindhuja Reddy, Avinesh Agrawal



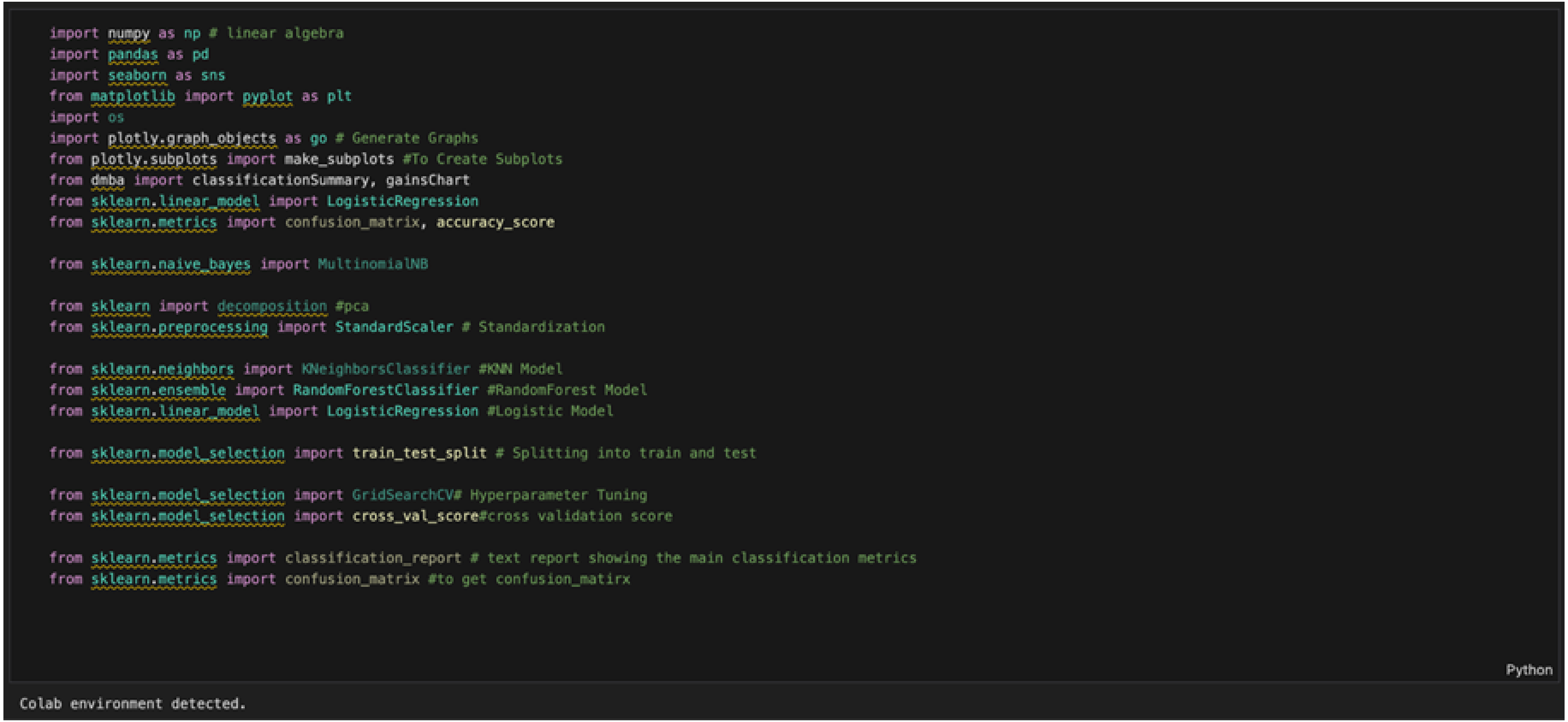
Background

- This dataset contains records of patient data concerning 12 features:
  - Age, Gender, Height, Weight, Systolic Blood Pressure, Diastolic Blood Pressure, Cholesterol, Glucose, Smoking, Alcohol Intake, Physical Activity, and the Presence or Absence of Cardiovascular Disease.
  - Dataset values were collected at the moment of medical examination.
  - Target class “cardio:”
    - 1 = presence of Cardiovascular Disease
    - 0 = absence of Cardiovascular Disease
  - 70,000 rows of cardio dataset
- Link to Kaggle Dataset: https://www.kaggle.com/code/sulianova/eda-cardiovascular-data/notebook
- Link to Collab File: https://colab.research.google.com/drive/1PBmy2mUm8_uS7ikWZhNY8tqKMM7PcaXE#scrollTo=GiXyDbCED2kM

Objective

- To uncover insights concerning factors that make an individual more likely to acquire cardiovascular disease by use of patient examination results.

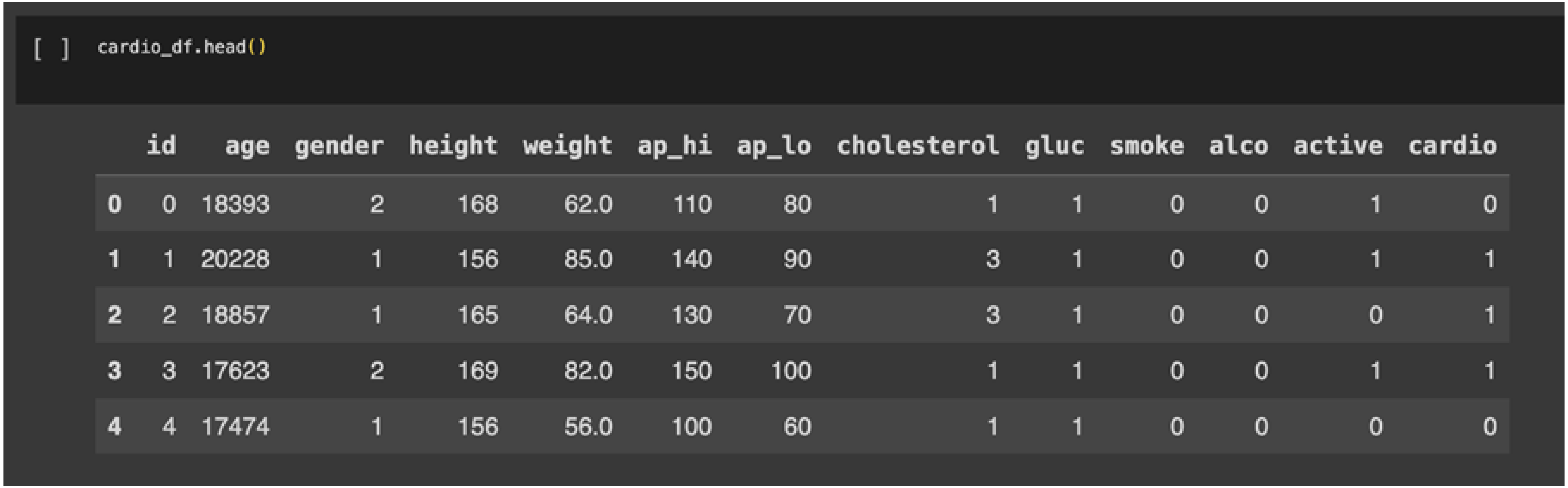
Applicable Libraries + Packages



```python
import numpy as np # linear algebra
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
import os
import plotly.graph_objects as go # Generate Graphs
from plotly.subplots import make_subplots #To Create Subplots
from dmba import classificationSummary, gainsChart
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score

from sklearn.naive_bayes import MultinomialNB

from sklearn import decomposition #pca
from sklearn.preprocessing import StandardScaler # Standardization

from sklearn.neighbors import KNeighborsClassifier #KNN Model
from sklearn.ensemble import RandomForestClassifier #RandomForest Model
from sklearn.linear_model import LogisticRegression #Logistic Model

from sklearn.model_selection import train_test_split # Splitting into train and test

from sklearn.model_selection import GridSearchCV# Hyperparameter Tuning
from sklearn.model_selection import cross_val_score#cross validation score

from sklearn.metrics import classification_report # text report showing the main classification metrics
from sklearn.metrics import confusion_matrix #to get confusion_matirx
```

```
Colab environment detected.
```

Data Preview



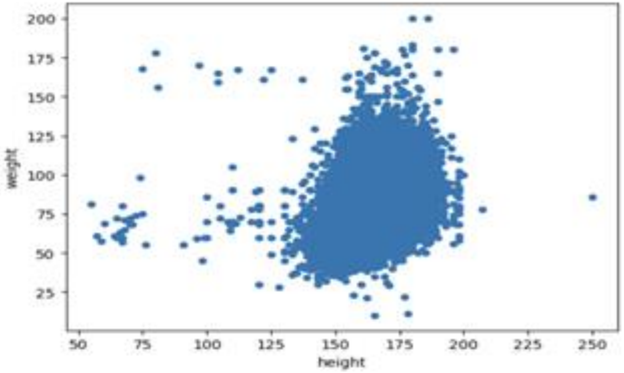
```
[ ]  cardio_df.head()
```

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0

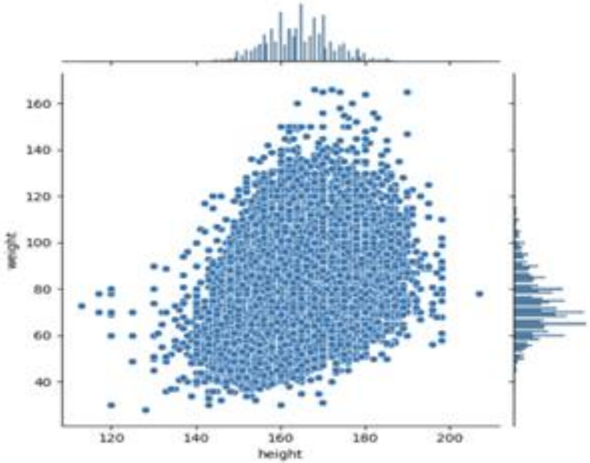
Data Cleaning Process

1. Handling missing values (no null values present)
2. Remove duplicate values (no duplicates present)
3. Removing outliers

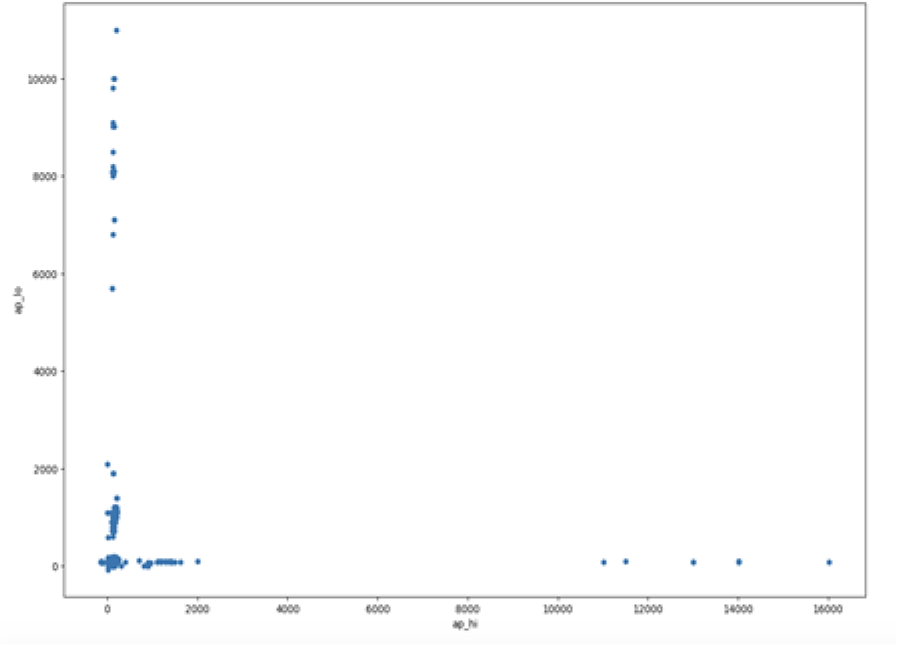
Data cleaning



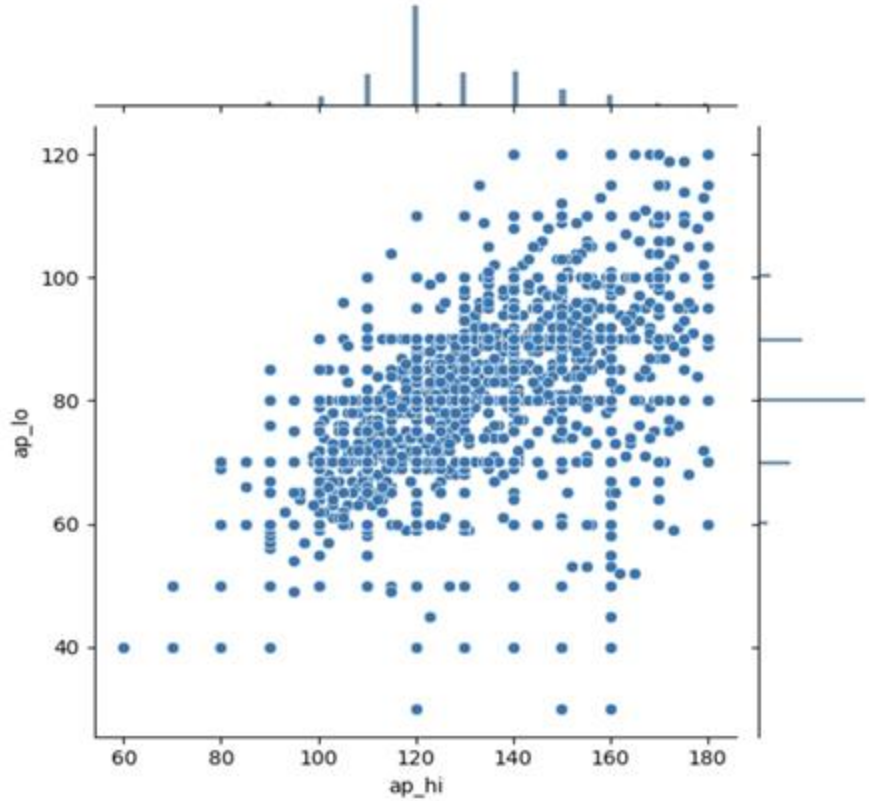
Scatterplot of Height and Weight with Outliers



Scatterplot of Height and Weight without Outliers

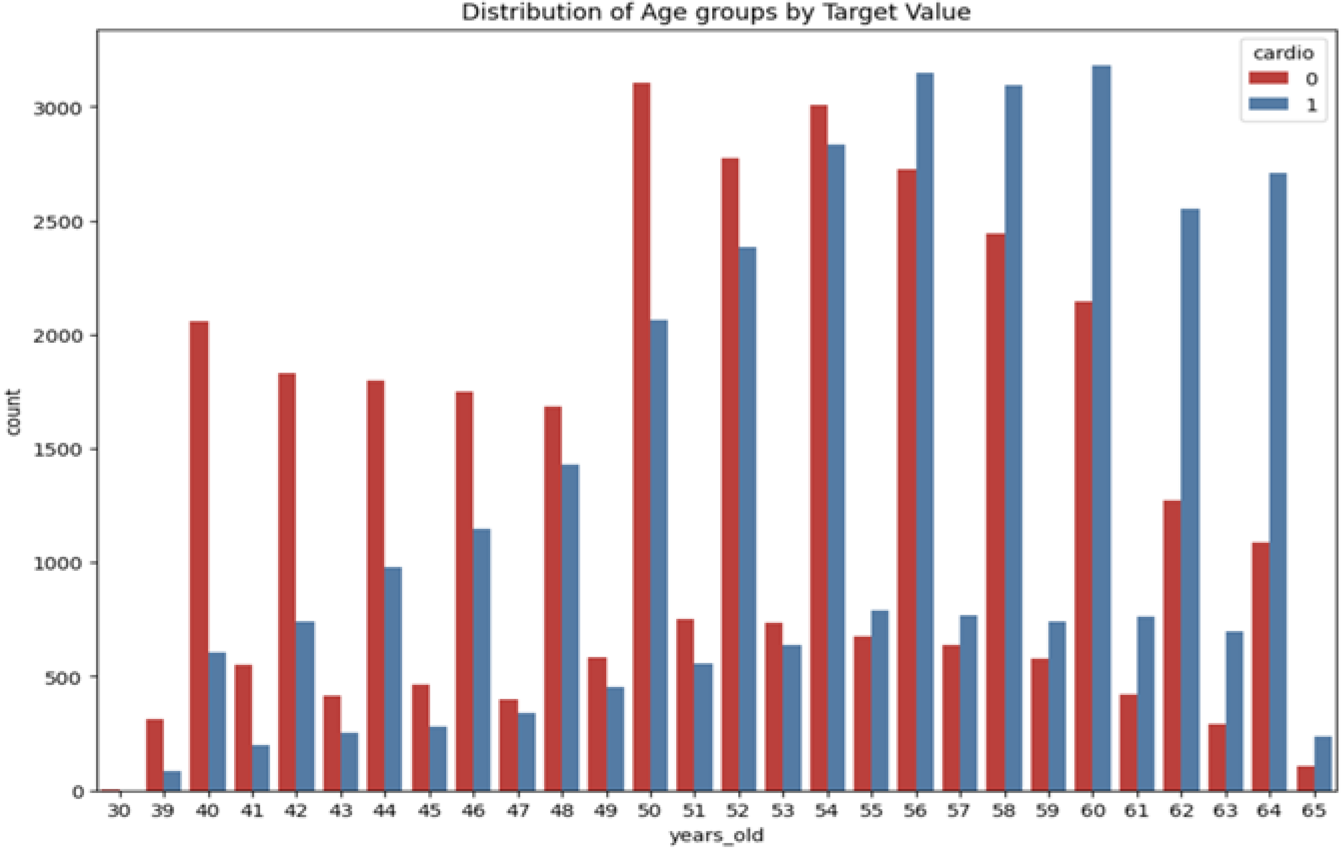


Outliers present among ap_hi (Systolic Blood Pressure) & ap_lo (Diastolic Blood Pressure)

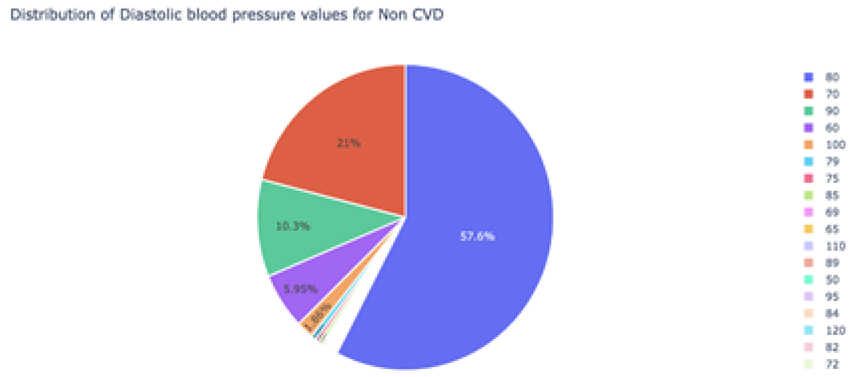


ap_hi (Systolic Blood Pressure) & ap_lo (Diastolic Blood Pressure) without outliers

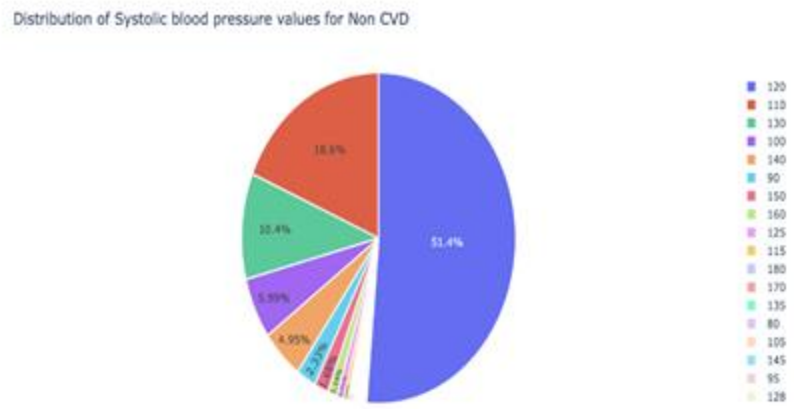
Data Visualization



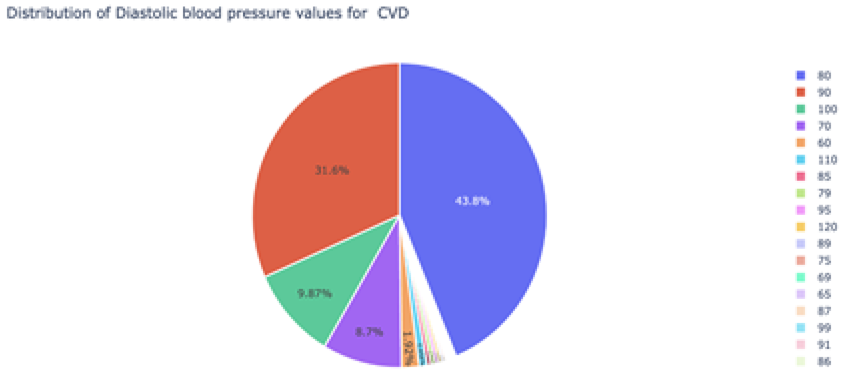
Higher age groups have a greater likelihood of having cardiovascular disease (CVD).



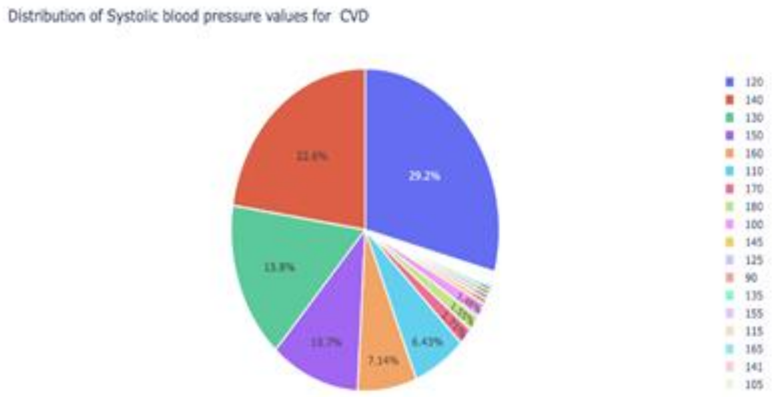
Distribution of Diastolic blood pressure values for Non CVD



Distribution of Systolic blood pressure values for Non CVD

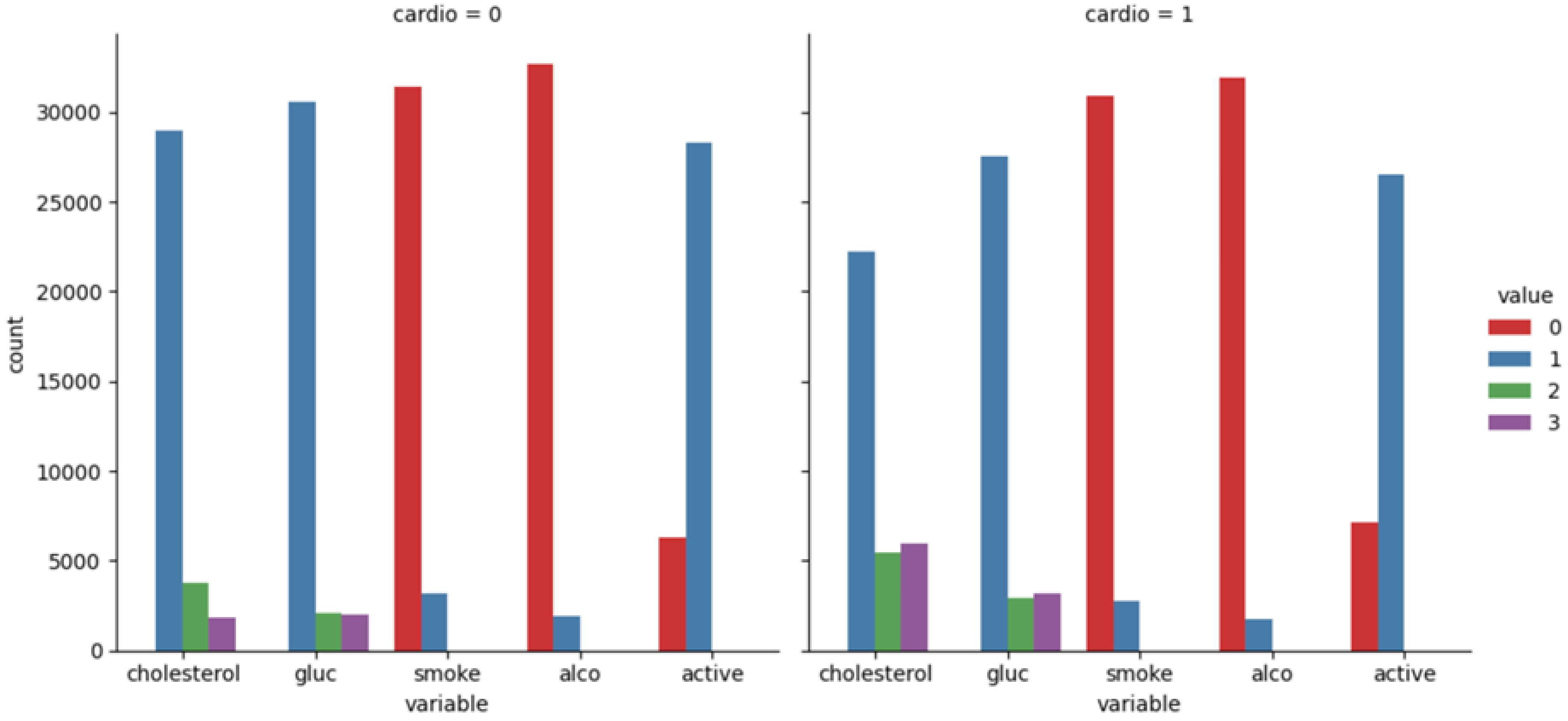


Distribution of Diastolic blood pressure values for CVD

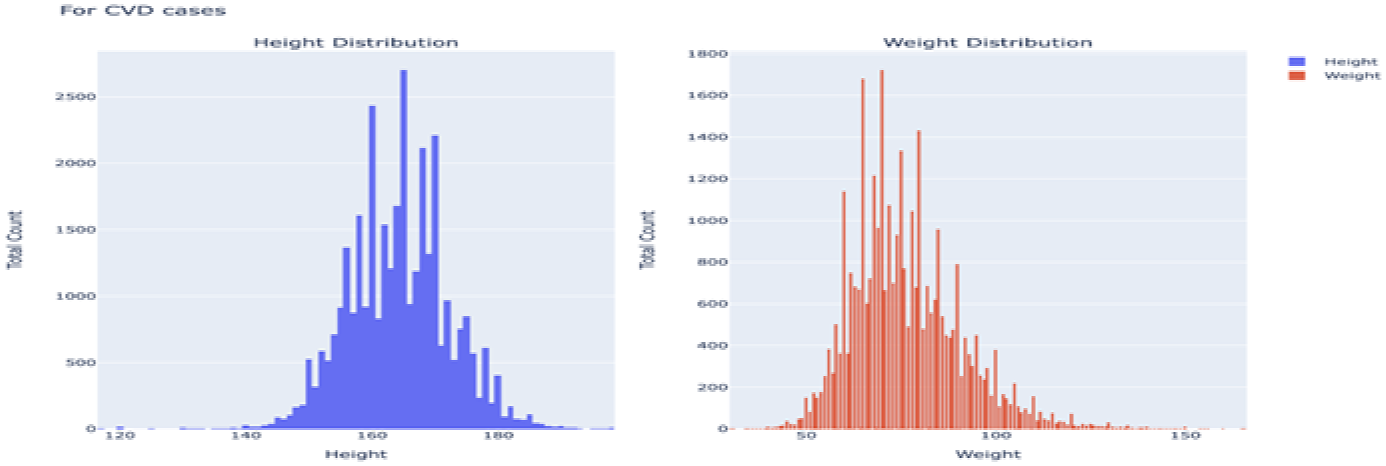


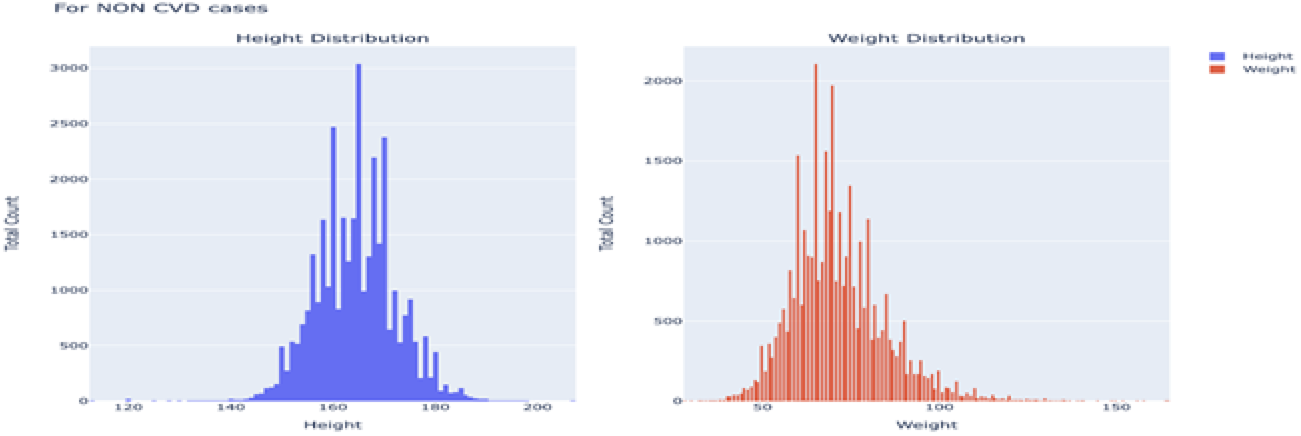
Distribution of Systolic blood pressure values for CVD

Maximum population has blood pressure level of 80 mmHg and 120 mmHg for diastolic and systolic respectively.

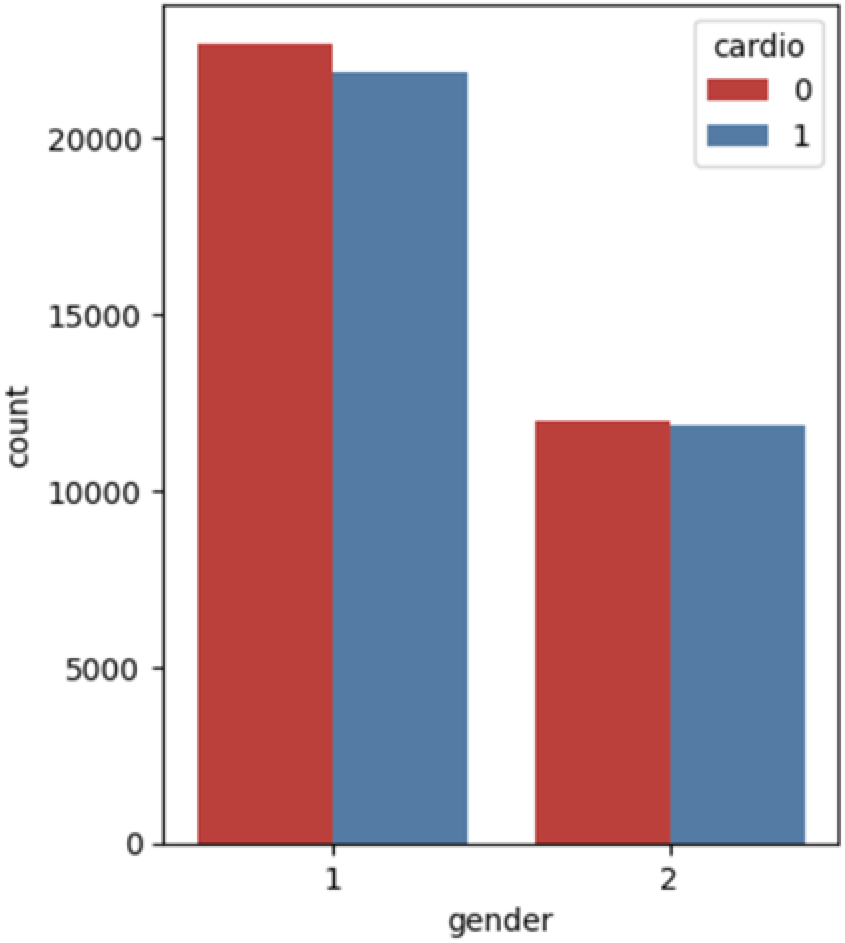


It can be clearly seen that patients with CVD have higher cholesterol and glucose level.





Features like Weight and Height are well distributed for Non - CVD and CVD Population.



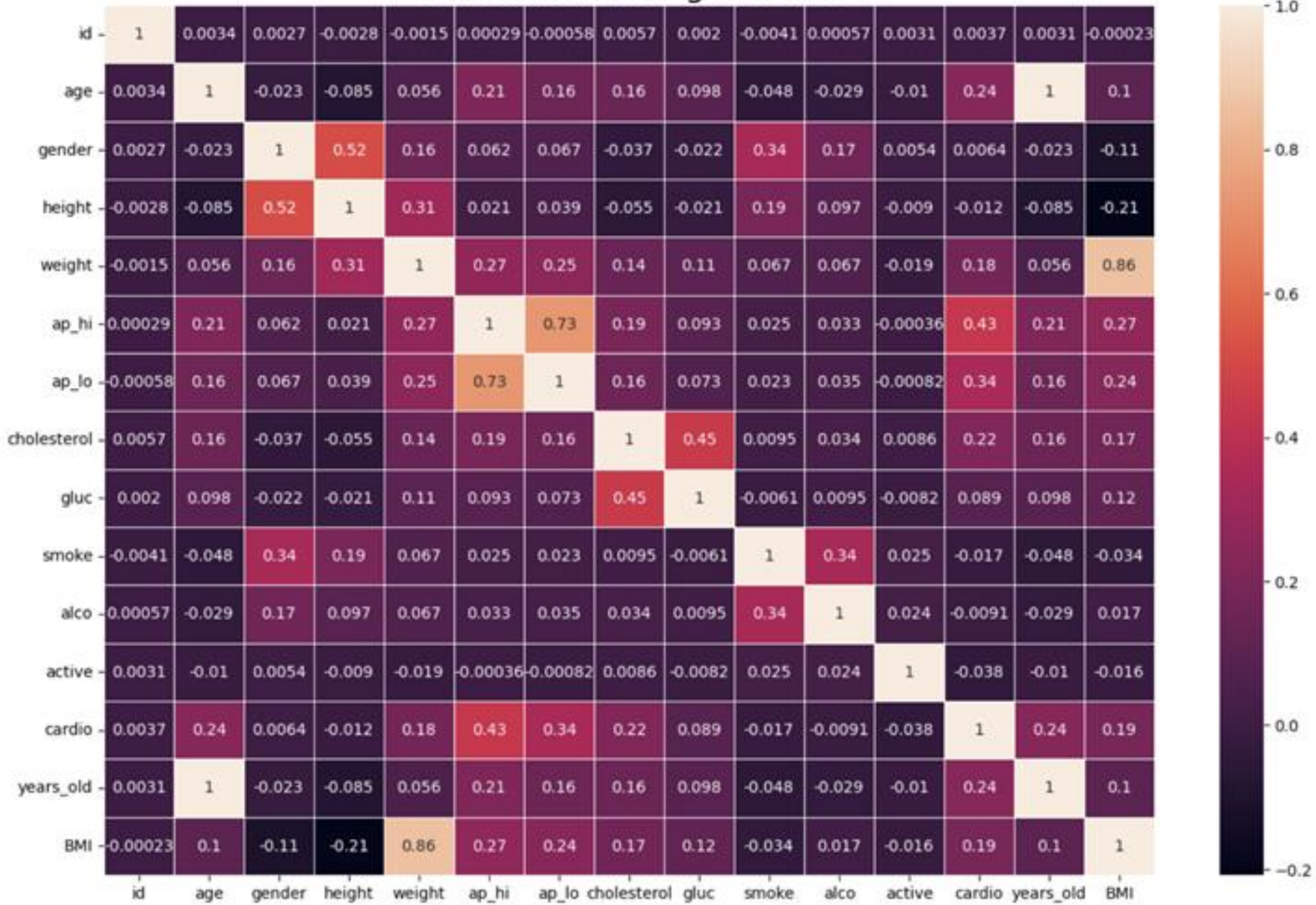
The dataset seems to be balanced for count of CVD and non CVD cases spread across both genders.

```
Note: 1 represents : Female
      2 represents : Male
```

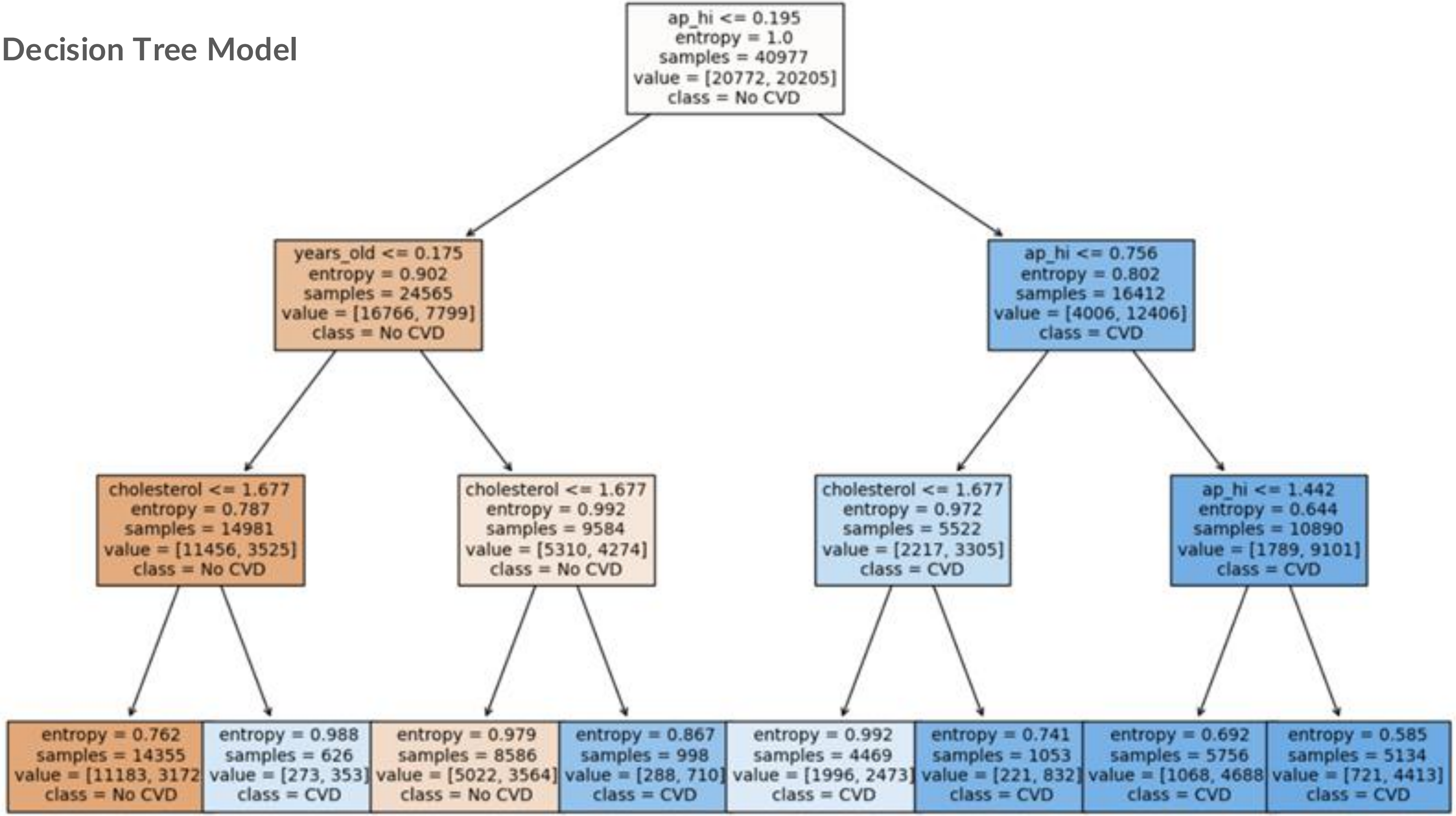
Predictive Analysis & Conclusion

- Models implemented:
    - Decision Tree
    - Random Forest Classifier
    - Logistic Regression
    - KNN

Correlation among features

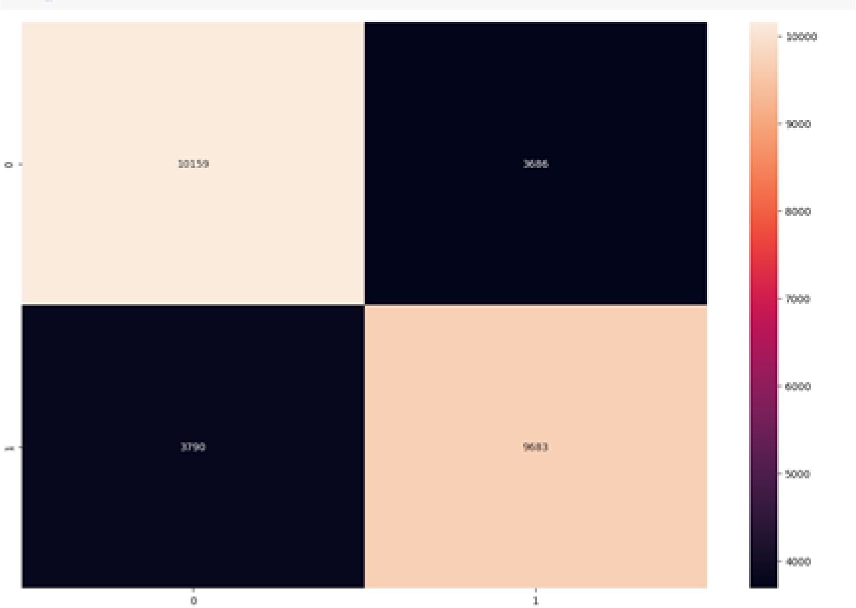


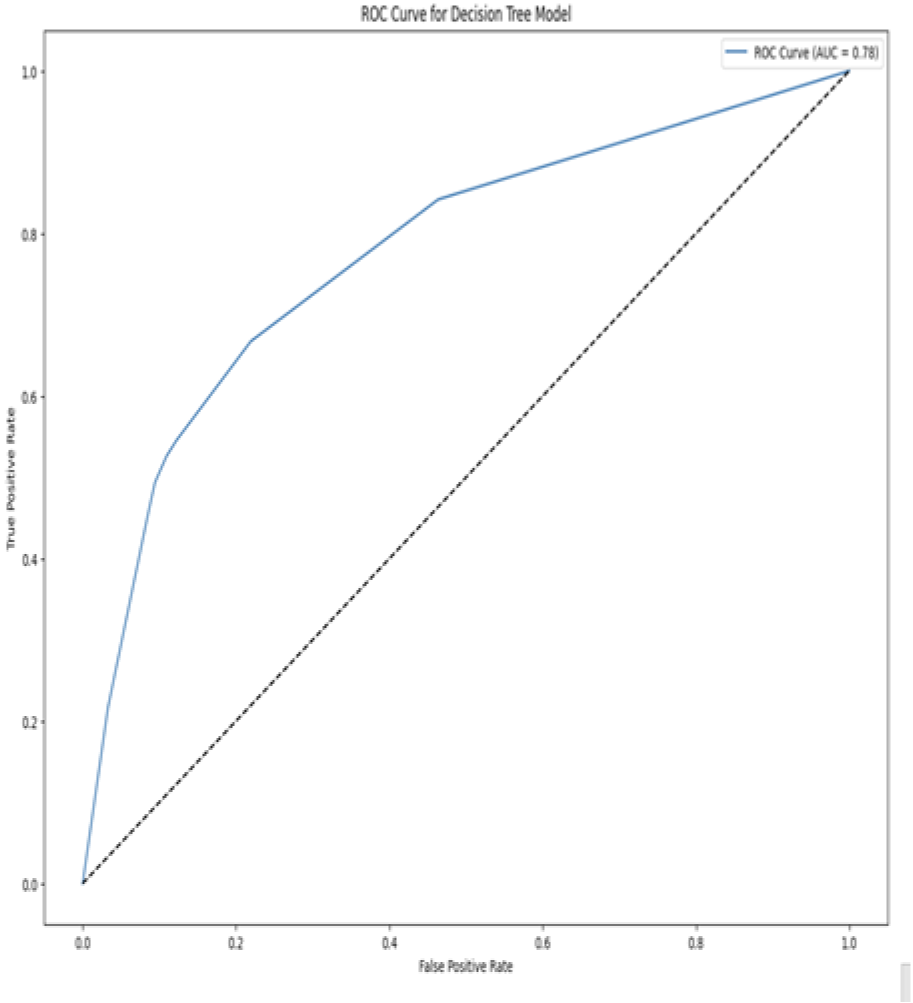
Decision Tree Model


ap_hi <= 0.195 entropy = 1.0 samples = 40977 value = [20772, 20205] class = No CVD
years_old <= 0.175 entropy = 0.902 samples = 24565 value = [16766, 7799] class = No CVD
ap_hi <= 0.756 entropy = 0.802 samples = 16412 value = [4006, 12406] class = CVD
cholesterol <= 1.677 entropy = 0.787 samples = 14981 value = [11456, 3525] class = No CVD
cholesterol <= 1.677 entropy = 0.992 samples = 9584 value = [5310, 4274] class = No CVD
cholesterol <= 1.677 entropy = 0.972 samples = 5522 value = [2217, 3305] class = CVD
ap_hi <= 1.442 entropy = 0.644 samples = 10890 value = [1789, 9101] class = CVD
entropy = 0.762 samples = 14355 value = [11183, 3172 class = No CVD
entropy = 0.988 samples = 626 value = [273, 353] class = CVD
entropy = 0.979 samples = 8586 value = [5022, 3564] class = No CVD
entropy = 0.867 samples = 998 value = [288, 710] class = CVD
entropy = 0.992 samples = 4469 value = [1996, 2473] class = CVD
entropy = 0.741 samples = 1053 value = [221, 832] class = CVD
entropy = 0.692 samples = 5756 value = [1068, 4688 class = CVD
entropy = 0.585 samples = 5134 value = [721, 4413] class = CVD

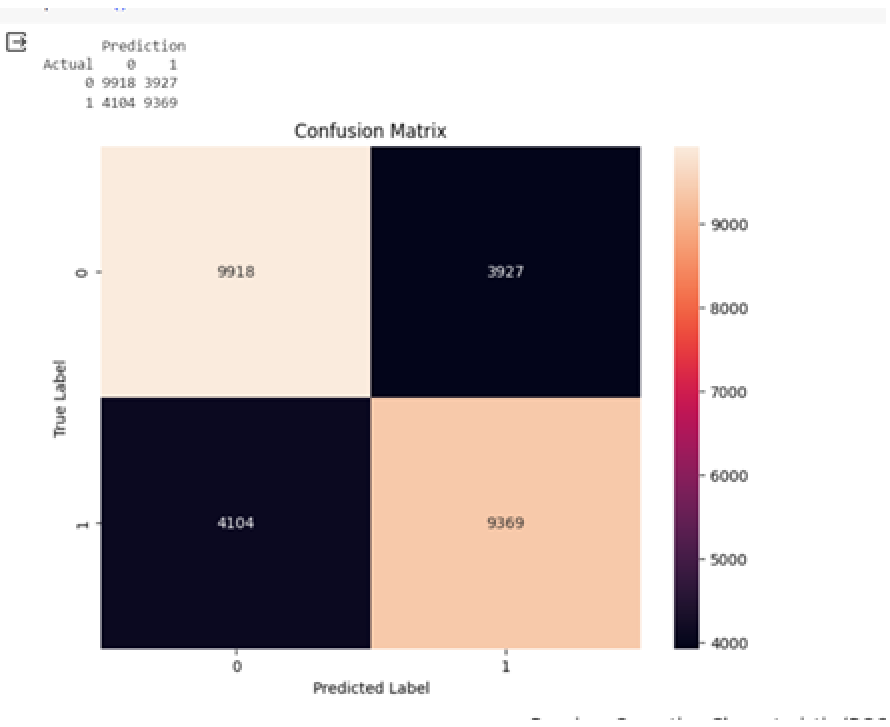
Decision Tree Model

```
Confusion Matrix (Accuracy 0.7263)

           Prediction
Actual         0      1
     0 10159   3686
     1  3790   9683
```

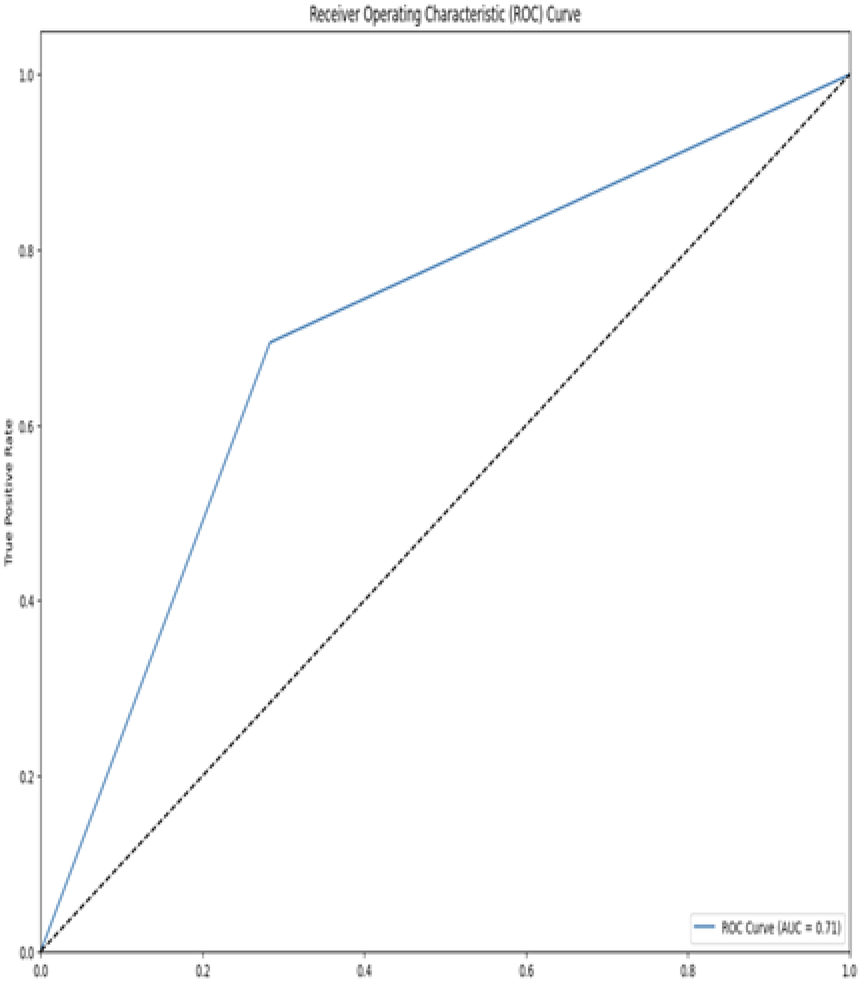




Random Forests

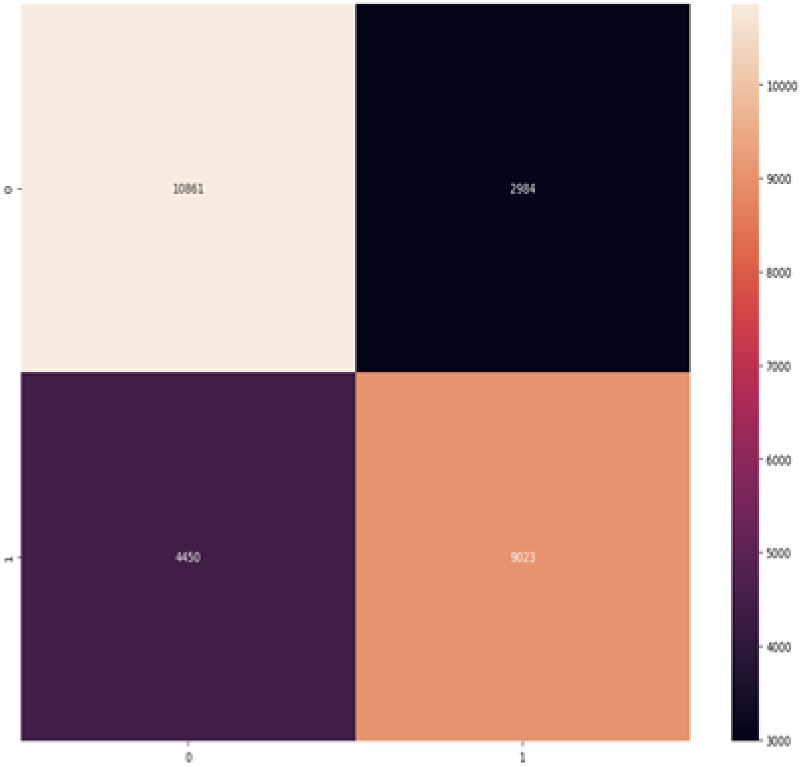


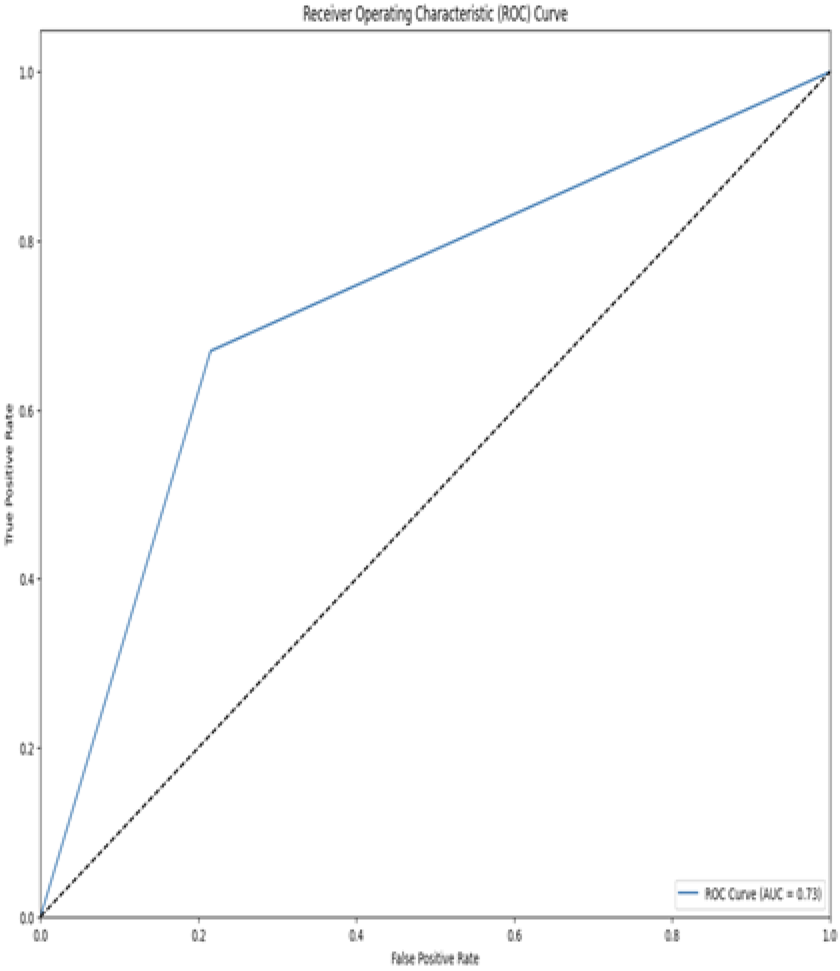
```
        Prediction
Actual     0    1
     0  9918 3927
     1  4104 9369
```



Logistic Regression

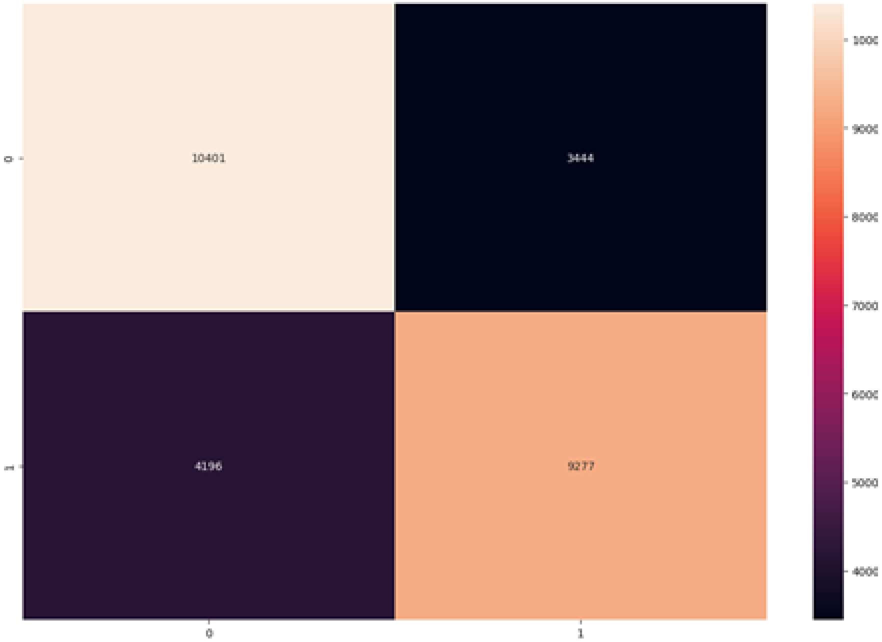
```
Confusion Matrix (Accuracy 0.7279)

       Prediction
Actual     0     1
     0 10861  2984
     1  4450  9023
```

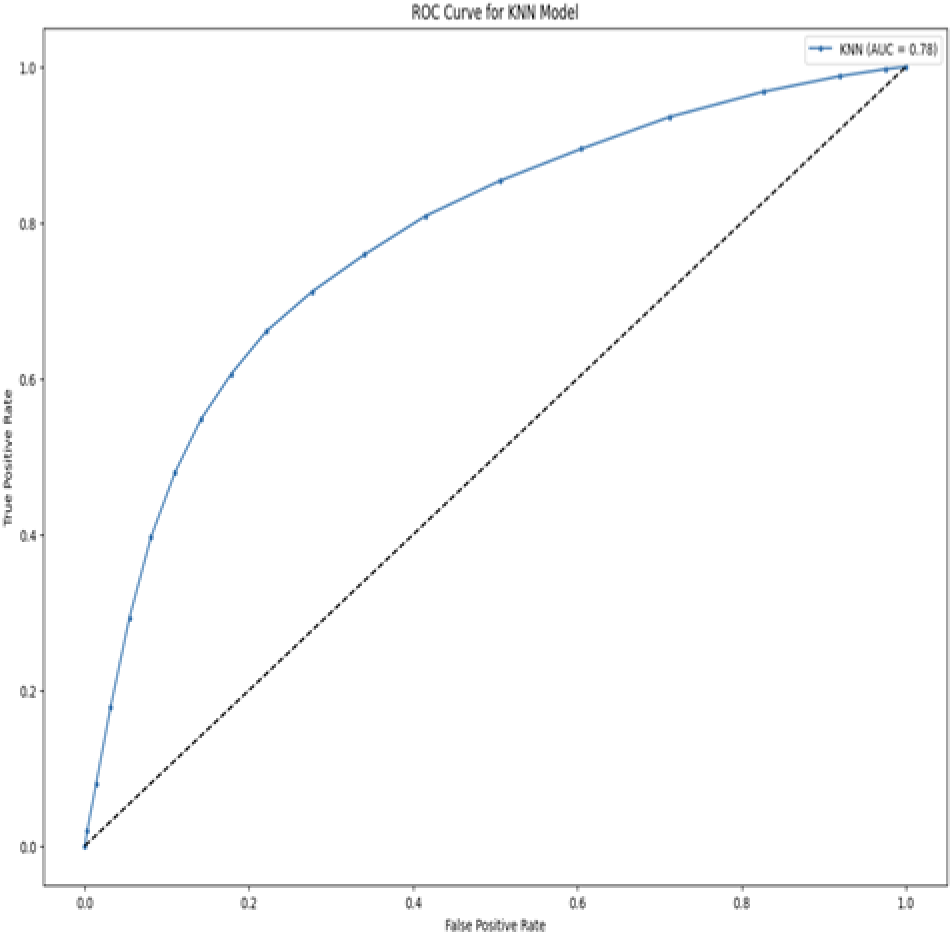




KNN Model

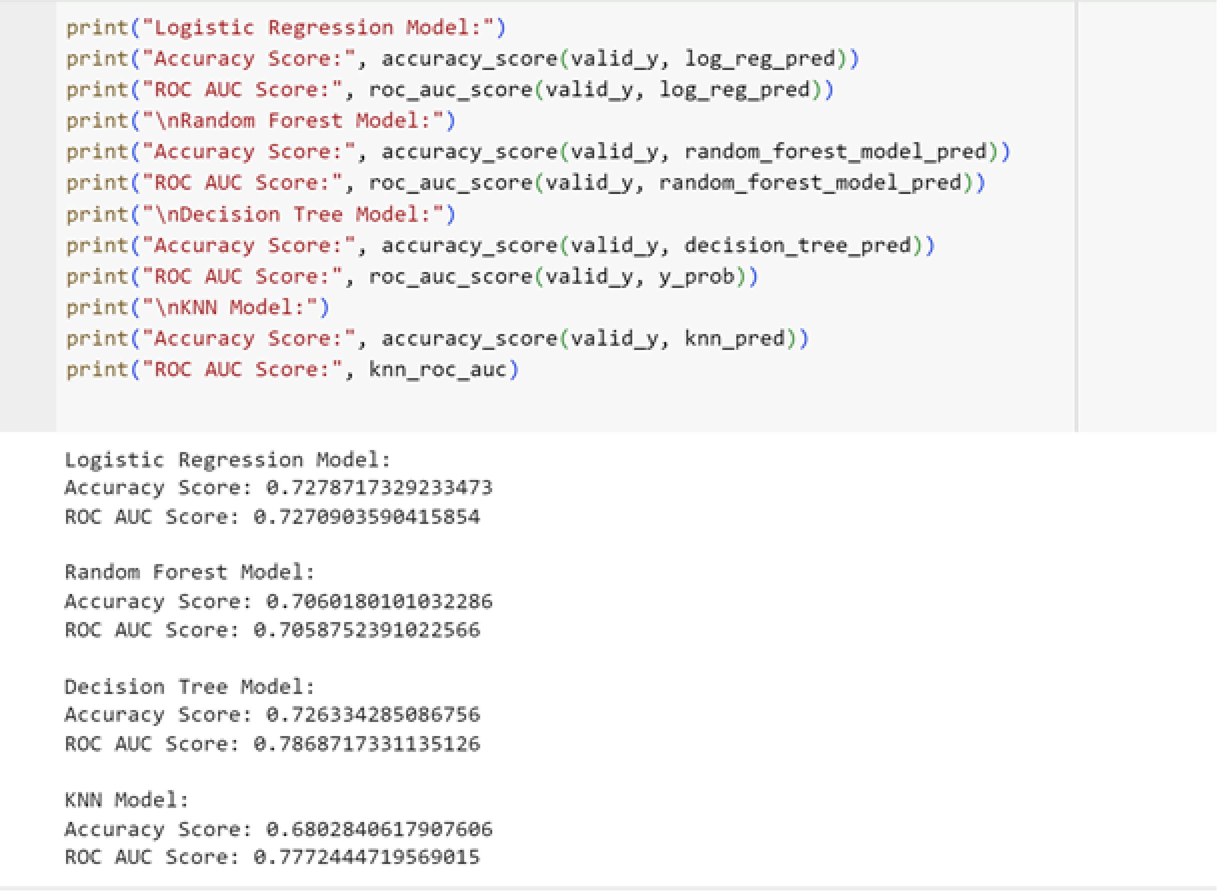
```
Confusion Matrix (Accuracy 0.7203)

       Prediction
Actual      0     1
     0 10401  3444
     1  4196  9277
```





Conclusion

- Models implemented:
  - Decision Tree
  - Random Forest Classifier
  - Logistic Regression
  - KNN
- As per our analysis Decision Tree Model best fits.



```python
print("Logistic Regression Model:")
print("Accuracy Score:", accuracy_score(valid_y, log_reg_pred))
print("ROC AUC Score:", roc_auc_score(valid_y, log_reg_pred))
print("\nRandom Forest Model:")
print("Accuracy Score:", accuracy_score(valid_y, random_forest_model_pred))
print("ROC AUC Score:", roc_auc_score(valid_y, random_forest_model_pred))
print("\nDecision Tree Model:")
print("Accuracy Score:", accuracy_score(valid_y, decision_tree_pred))
print("ROC AUC Score:", roc_auc_score(valid_y, y_prob))
print("\nKNN Model:")
print("Accuracy Score:", accuracy_score(valid_y, knn_pred))
print("ROC AUC Score:", knn_roc_auc)
```

```
Logistic Regression Model:
Accuracy Score: 0.7278717329233473
ROC AUC Score: 0.7270903590415854

Random Forest Model:
Accuracy Score: 0.7060180101032286
ROC AUC Score: 0.7058752391022566

Decision Tree Model:
Accuracy Score: 0.726334285086756
ROC AUC Score: 0.7868717331135126

KNN Model:
Accuracy Score: 0.6802840617907606
ROC AUC Score: 0.7772444719569015
```