UNIVERSITY OF CANBERRA
AUSTRALIA'S CAPITAL UNIVERSITY

Faculty of Science and Technology
Assignment Coversheet

Student ID number & Student Name	U3253332 Nafis Khan
Unit name	Software technology
Unit number	4483
Unit Tutor	Pranav Gupta
Assignment name	ST1 Capstone Project – Semester 2 2023
Due date	29/10/2023
Date submitted	29/10/2023

You must keep a photocopy or electronic copy of your assignment.

Student declaration

I certify that the attached assignment is my own work. Material drawn from other sources has been appropriately and fully acknowledged as to author/creator, source and other bibliographic details.

Signature of student: Nafis Khan **Date: 29/10/23**

Table of Contents

Introduction

This report delves into Heart Attacks and explores the varying factors that can make an individual more susceptible to this life-threatening medical emergency. To facilitate this exploration, the report will provide insights into heart attacks and provide various types of charts and explanations. To conduct this analysis, a dataset sourced from Kaggle will be utilized, offering a range of variables and parameters for examination. The methodology involves creating a Streamlit application and performing Exploratory Data Analysis (EDA) and Predictive Data Analysis (PDA) on the dataset. This approach aims to enhance individuals' understanding of the topic and prove the accuracy of the dataset.

By creating a user-friendly tool. the objective is to act as a preventive measure and raise awareness about heart attacks. Raising awareness can significantly reduce the risk of heart attacks by encouraging people to avoid leading causes and unhealthy habits. This approach allows for a prediction of an individual's likelihood of being diagnosed with a heart attack.

The report showcases a prototype software developed as a Python module and hosted in a Google Colab repository. The development process follows a data-driven scientific approach, incorporating Exploratory Data Analysis (EDA) and Predictive Analysis (PDA). The findings are implemented in the form of a desktop application using Streamlit.

Methodology

1. Design and Development

The project involves conducting Exploratory Data Analysis (EDA) and Predictive Data Analysis (PDA) to determine the most suitable artificial intelligence learning model. These analyses are crucial for identifying patterns, correlations, and predicting outcomes, all essential for solving real-world problems effectively.

2. Implementation

Once the optimal AI model is identified through PDA, it is integrated into the Streamlit Application as the primary tool for designing the interface. By implementing the AI model within the Streamlit, users can interact with the solution seamlessly.

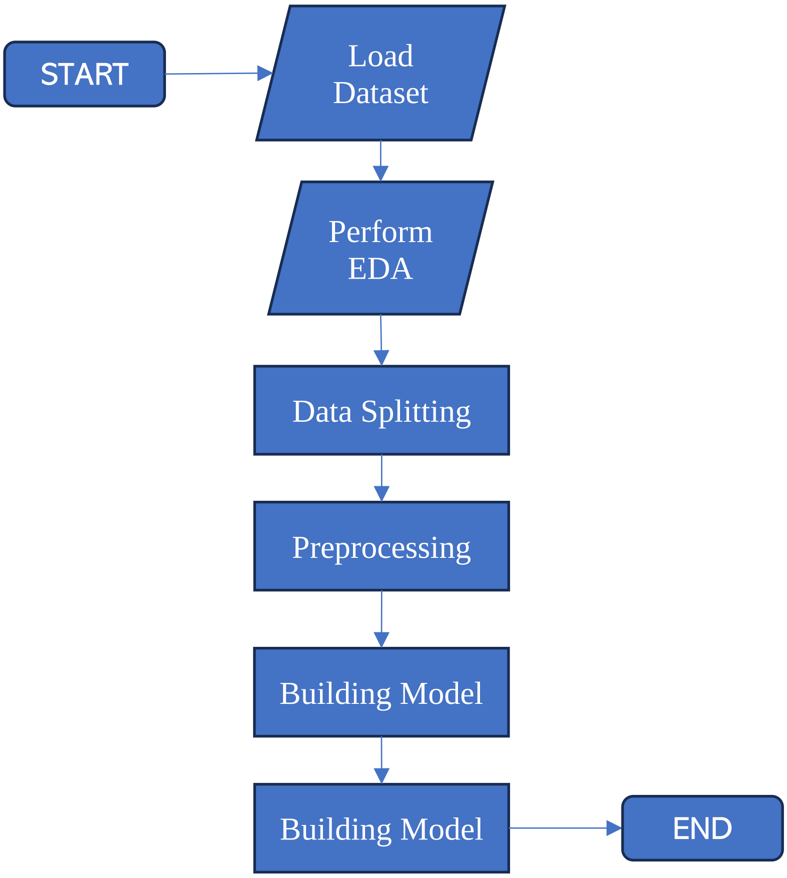
3. Deployment

After successful implementation deploy the tool as a web application using Streamlit.

Design and development

Algorithm for dataset:

Load Dataset
Perform EDA
Data Splitting
Preprocessing
Building Model
Performance Evaluation



Dataset description

In this project, a single dataset sourced from Kaggle is utilized, encompassing a comprehensive array of health-related attributes from 8763 patients worldwide. The dataset comprises 24 features, providing detailed insights into various aspects of patients' lives, including age, gender, cholesterol levels, blood pressure, lifestyle choices, medical history, socioeconomic status, and geographical location. One crucial binary target attribute is included, indicating the presence or absence of heart attack risk for each patient. The dataset's diverse nature allows for in-depth analysis and exploration, facilitating research into the complex interplay of these variables in determining the likelihood of a heart attack. The primary goal is to develop a predictive model and gain insights into the factors contributing to heart attack risk, thereby advancing proactive strategies for heart attack prevention. The task at hand is to develop a software tool to predict if a person is at risk of heart attack or not depending on their details.

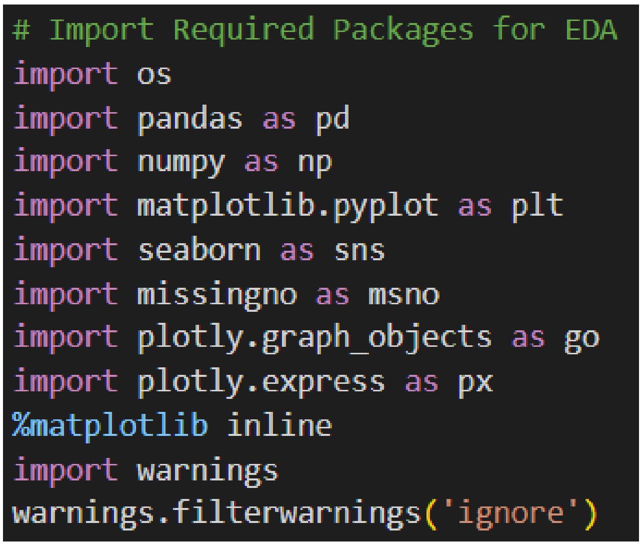
Exploratory Data Analysis

The EDA process was done using the google collab environment, since it is fast and very simple to run and also can be done from a browser. The programming language python was used to create the code.

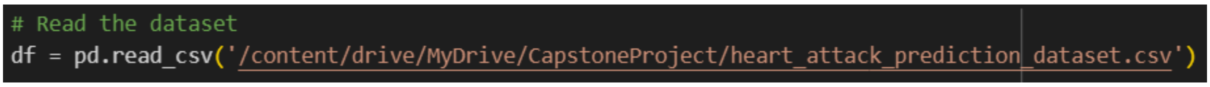
Questions to answer when analysing the dataset:

1. What percentage of the dataset is at risk of heart attack?
2. Does the age of a person contribute towards a heart attack?
3. Does having a high heart rate increase or decrease the risk of heart attack?
4. Does cholesterol level eventually contribute as a risk factor towards heart attack?
5. Is Gender a big factor in determining heart attack risk?

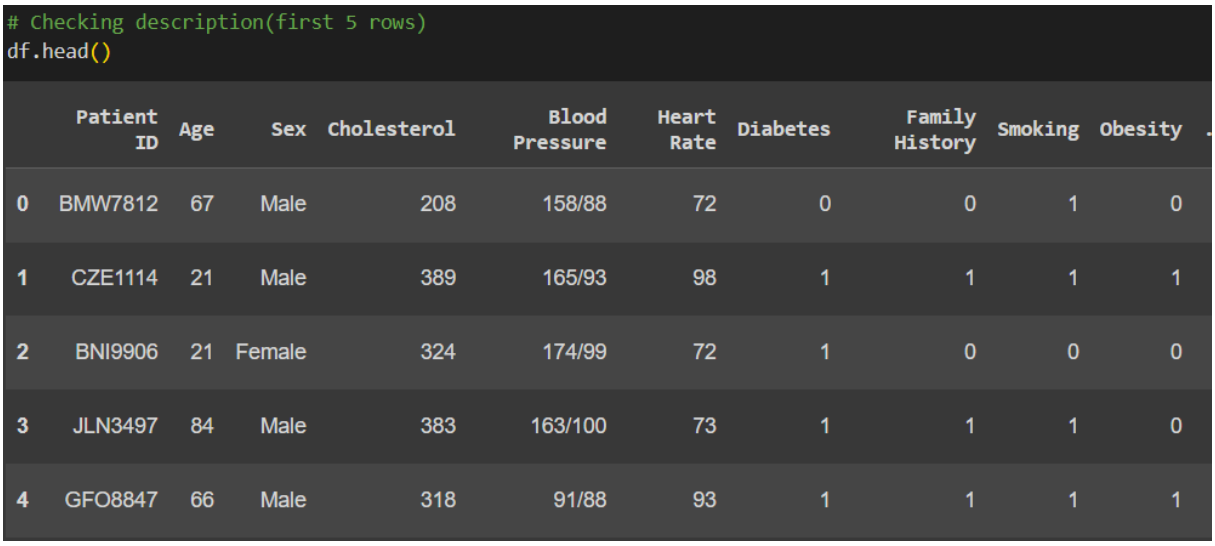
The first step to EDA is to understand the basic description of the data:



```
# Import Required Packages for EDA
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import plotly.graph_objects as go
import plotly.express as px
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

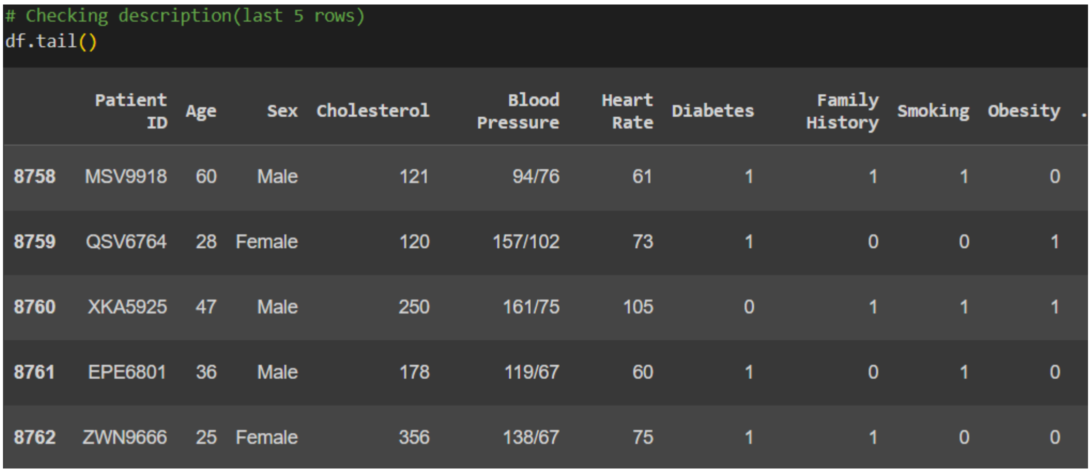


```
# Read the dataset
df = pd.read_csv('/content/drive/MyDrive/CapstoneProject/heart_attack_prediction_dataset.csv')
```



```
# Checking description(first 5 rows)
df.head()
```

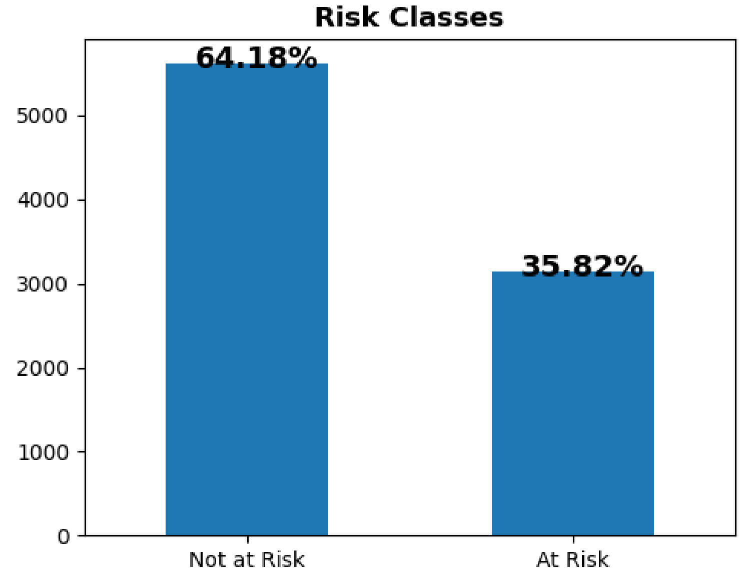
	Patient ID	Age	Sex	Cholesterol	Blood Pressure	Heart Rate	Diabetes	Family History	Smoking	Obesity
0	BMW7812	67	Male	208	158/88	72	0	0	1	0
1	CZE1114	21	Male	389	165/93	98	1	1	1	1
2	BNI9906	21	Female	324	174/99	72	1	0	0	0
3	JLN3497	84	Male	383	163/100	73	1	1	1	0
4	GFO8847	66	Male	318	91/88	93	1	1	1	1



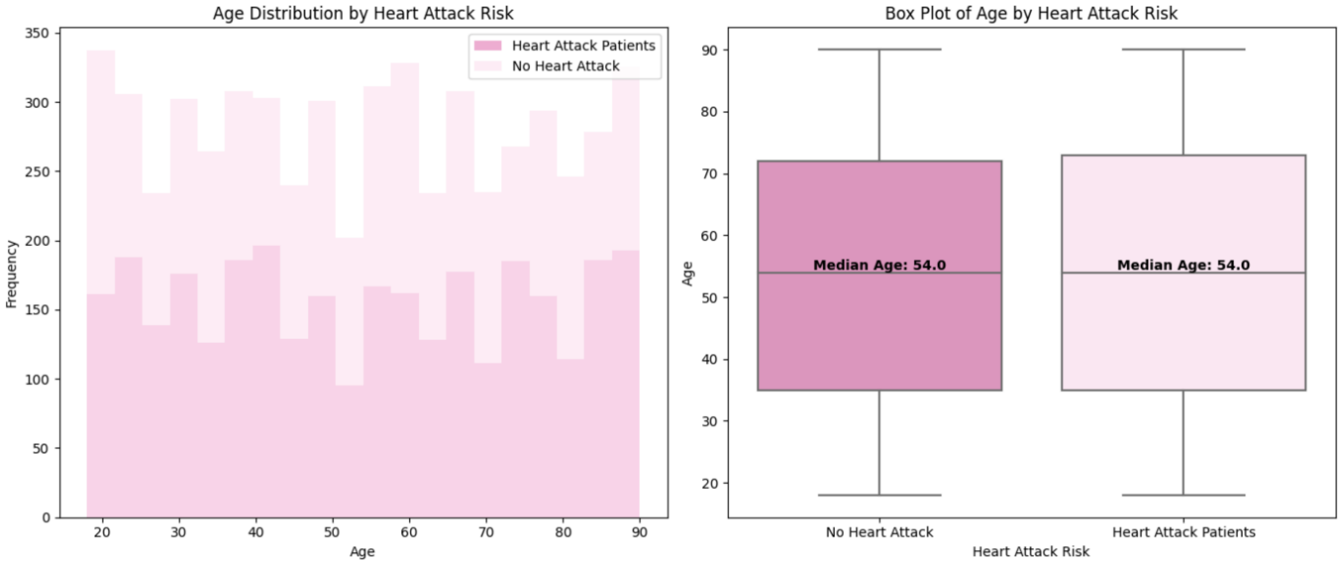
```
# Checking description(last 5 rows)
df.tail()
```

	Patient ID	Age	Sex	Cholesterol	Blood Pressure	Heart Rate	Diabetes	Family History	Smoking	Obesity
8758	MSV9918	60	Male	121	94/76	61	1	1	1	0
8759	QSV6764	28	Female	120	157/102	73	1	0	0	1
8760	XKA5925	47	Male	250	161/75	105	0	1	1	1
8761	EPE6801	36	Male	178	119/67	60	1	0	1	0
8762	ZWN9666	25	Female	356	138/67	75	1	1	0	0

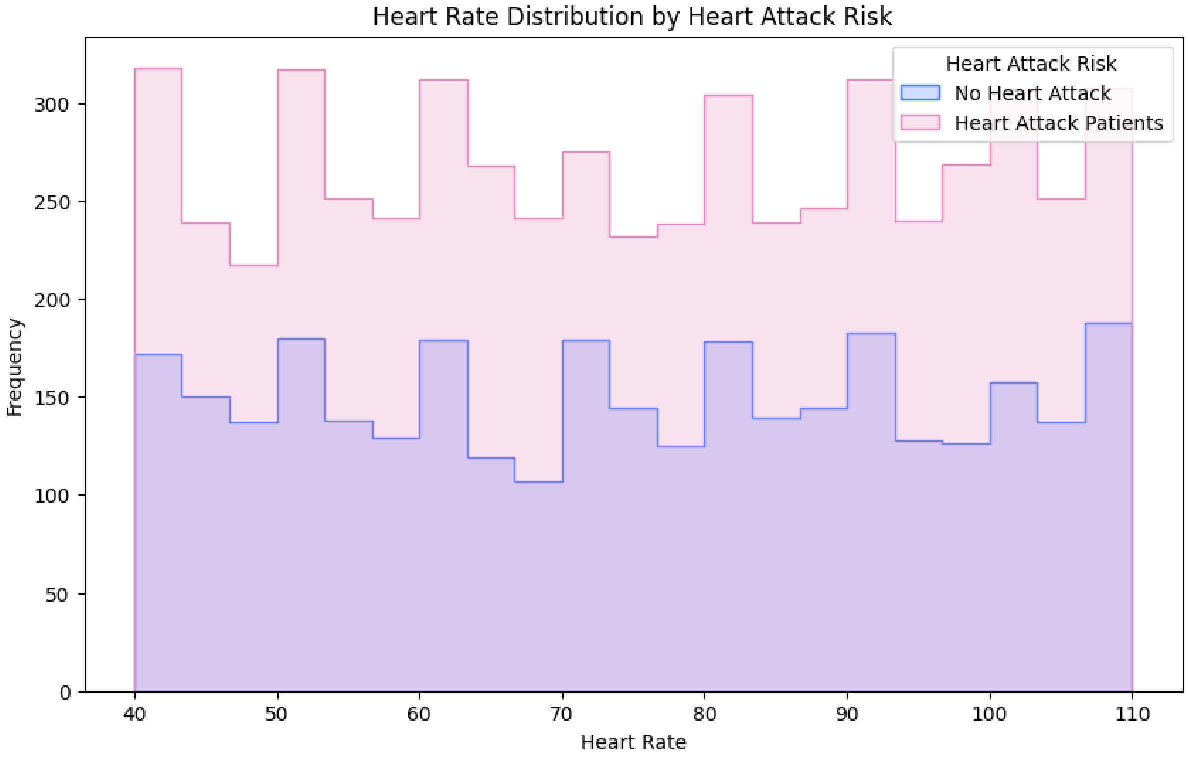
The next step is answering the questions:



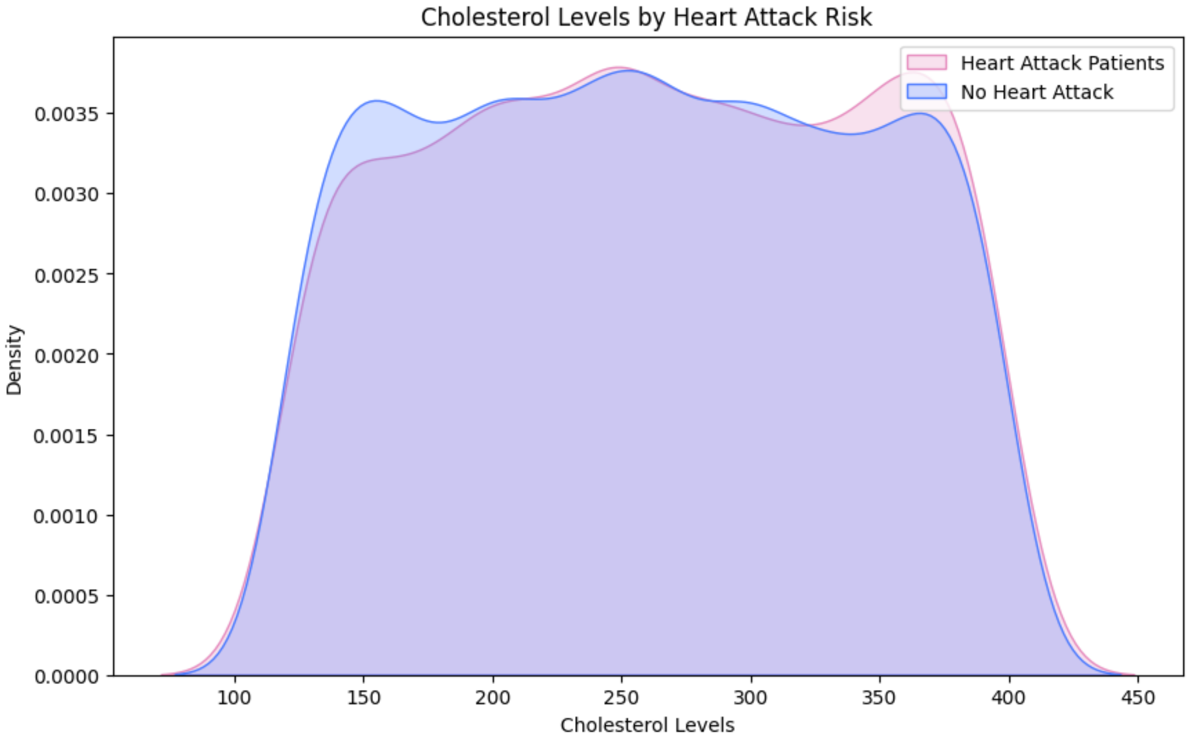
Observation: In this dataset 64.18% of patients are not at risk of heart attack, while 35.82% are at risk.



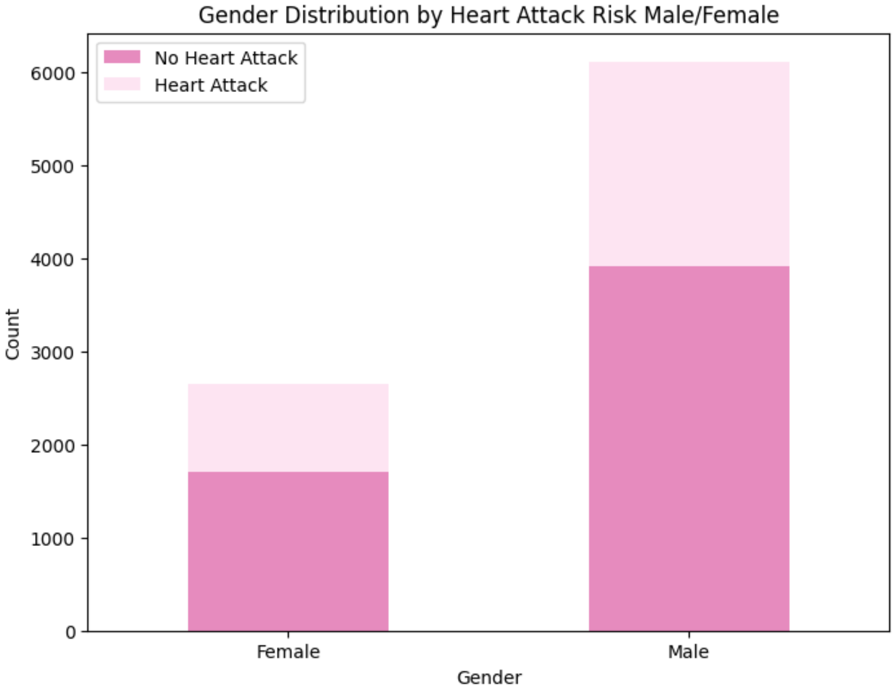
Observation: The age of a person does not affect the risk of a heart attack.



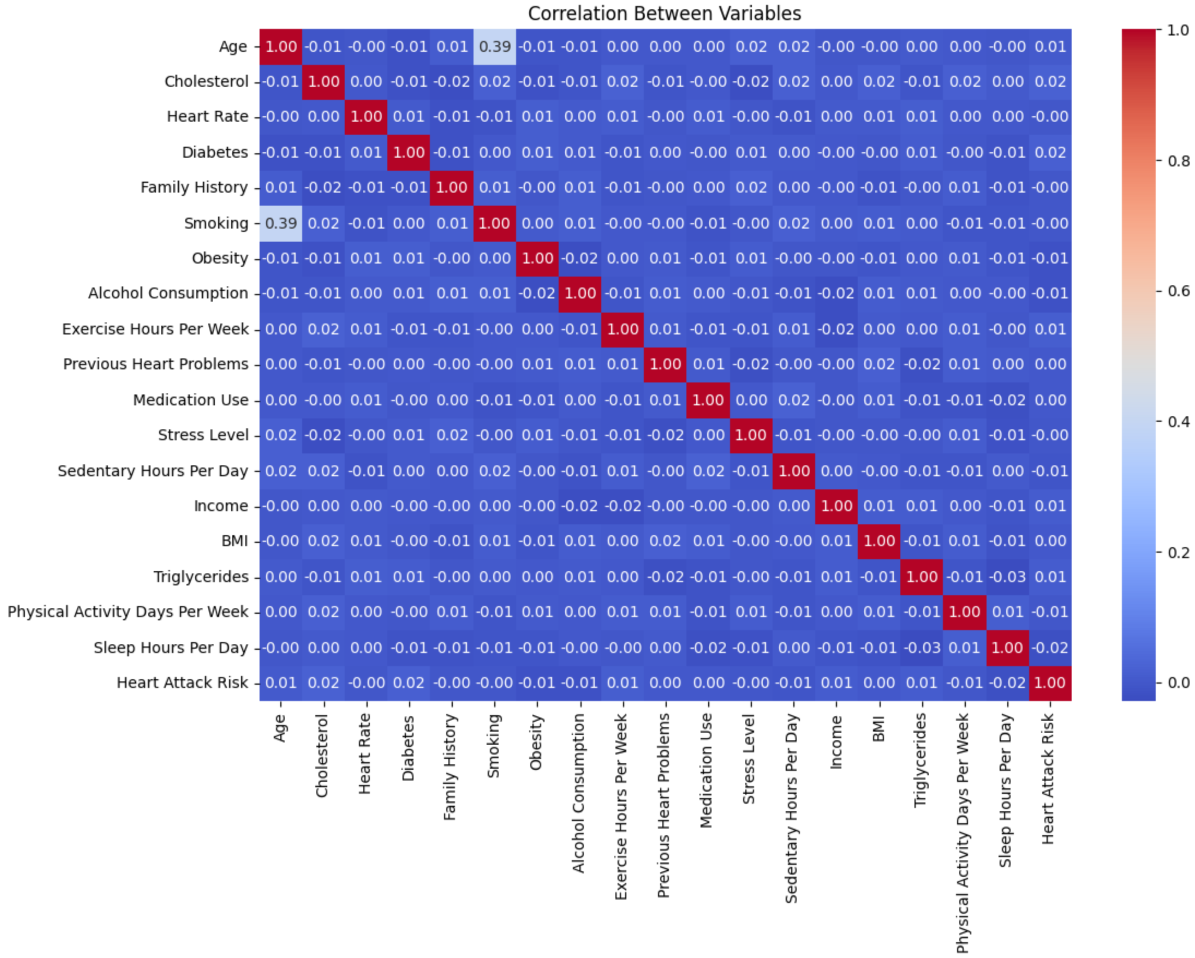
Observation: From the histogram above it can be deduced that having a higher heart rate does not increase the risk heart attack. As an increase/decrease in heart rate in the x-axis had similar frequency output in the y-axis.



Observation: Considering how the density plot of the 'No Heart Attack' is so similar to 'Heart Attack Patients', the Cholesterol levels of person does not contribute as a risk factor towards heart attack.



Observation: Men are more likely to be at risk of heart attack than woman.



Observation: From this correlation matrix it is observed that there aren't many variables that have high correlation to Heart Attack Risk.

Predictive Data Analytics:

To preform PDA several steps are required. This includes pre-processing, classifier comparison to identify the best machine learning classifier and performance evaluation with different objective metrics such as accuracy, classification report, confusion matrix, ROC_AUC curve and prediction report was obtained using python scikit-learn

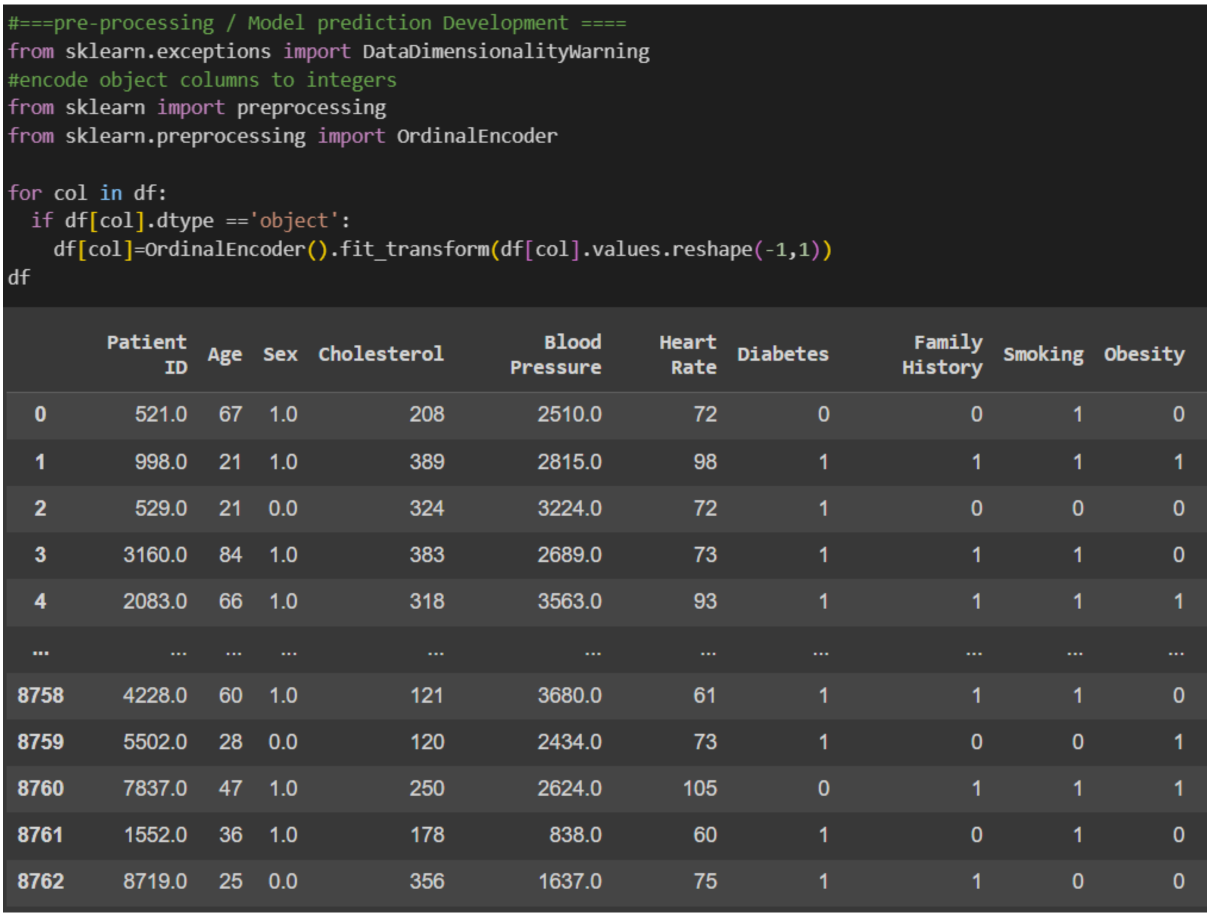
Steps:

Pre processing

- The data has both continuous and categorical attributes/values
- Attribute transformation, standardisation, normalisation (scikit-learn OrdinalEncoder() to perform attribute transformation

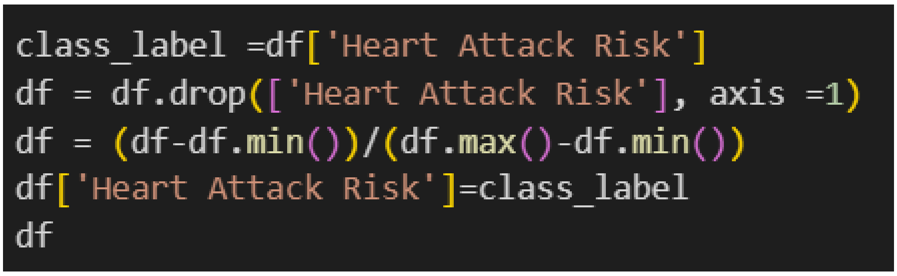
Normalisation

- Drop the target from the data frame, normalise it then reattach the target to the data frame

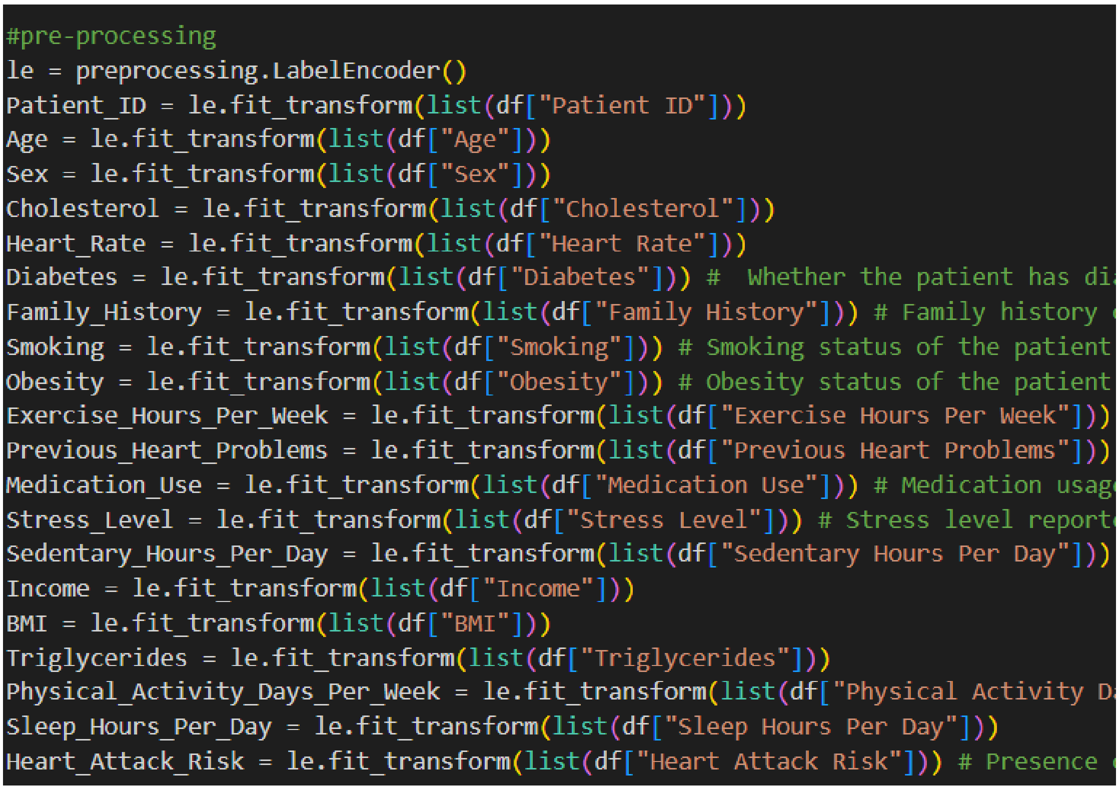


```
#===pre-processing / Model prediction Development ====
from sklearn.exceptions import DataDimensionalityWarning
#encode object columns to integers
from sklearn import preprocessing
from sklearn.preprocessing import OrdinalEncoder

for col in df:
  if df[col].dtype =='object':
    df[col]=OrdinalEncoder().fit_transform(df[col].values.reshape(-1,1))
df
```

	Patient ID	Age	Sex	Cholesterol	Blood Pressure	Heart Rate	Diabetes	Family History	Smoking	Obesity
0	521.0	67	1.0	208	2510.0	72	0	0	1	0
1	998.0	21	1.0	389	2815.0	98	1	1	1	1
2	529.0	21	0.0	324	3224.0	72	1	0	0	0
3	3160.0	84	1.0	383	2689.0	73	1	1	1	0
4	2083.0	66	1.0	318	3563.0	93	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...
8758	4228.0	60	1.0	121	3680.0	61	1	1	1	0
8759	5502.0	28	0.0	120	2434.0	73	1	0	0	1
8760	7837.0	47	1.0	250	2624.0	105	0	1	1	1
8761	1552.0	36	1.0	178	838.0	60	1	0	1	0
8762	8719.0	25	0.0	356	1637.0	75	1	1	0	0



```
class_label =df['Heart Attack Risk']
df = df.drop(['Heart Attack Risk'], axis =1)
df = (df-df.min())/(df.max()-df.min())
df['Heart Attack Risk']=class_label
df
```

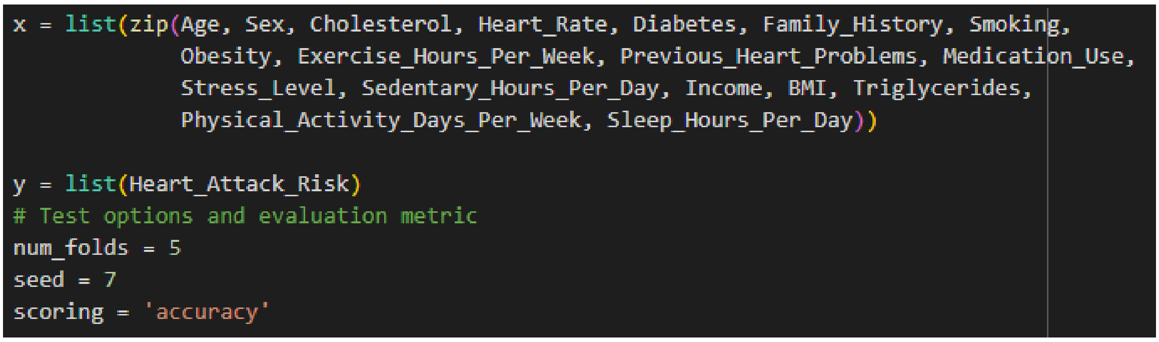


```
#pre-processing
le = preprocessing.LabelEncoder()
Patient_ID = le.fit_transform(list(df["Patient ID"]))
Age = le.fit_transform(list(df["Age"]))
Sex = le.fit_transform(list(df["Sex"]))
Cholesterol = le.fit_transform(list(df["Cholesterol"]))
Heart_Rate = le.fit_transform(list(df["Heart Rate"]))
Diabetes = le.fit_transform(list(df["Diabetes"])) #  Whether the patient has di
Family_History = le.fit_transform(list(df["Family History"])) # Family history
Smoking = le.fit_transform(list(df["Smoking"])) # Smoking status of the patient
Obesity = le.fit_transform(list(df["Obesity"])) # Obesity status of the patient
Exercise_Hours_Per_Week = le.fit_transform(list(df["Exercise Hours Per Week"]))
Previous_Heart_Problems = le.fit_transform(list(df["Previous Heart Problems"]))
Medication_Use = le.fit_transform(list(df["Medication Use"])) # Medication usag
Stress_Level = le.fit_transform(list(df["Stress Level"])) # Stress level report
Sedentary_Hours_Per_Day = le.fit_transform(list(df["Sedentary Hours Per Day"]))
Income = le.fit_transform(list(df["Income"]))
BMI = le.fit_transform(list(df["BMI"]))
Triglycerides = le.fit_transform(list(df["Triglycerides"]))
Physical_Activity_Days_Per_Week = le.fit_transform(list(df["Physical Activity Da
Sleep_Hours_Per_Day = le.fit_transform(list(df["Sleep Hours Per Day"]))
Heart_Attack_Risk = le.fit_transform(list(df["Heart Attack Risk"])) # Presence
```

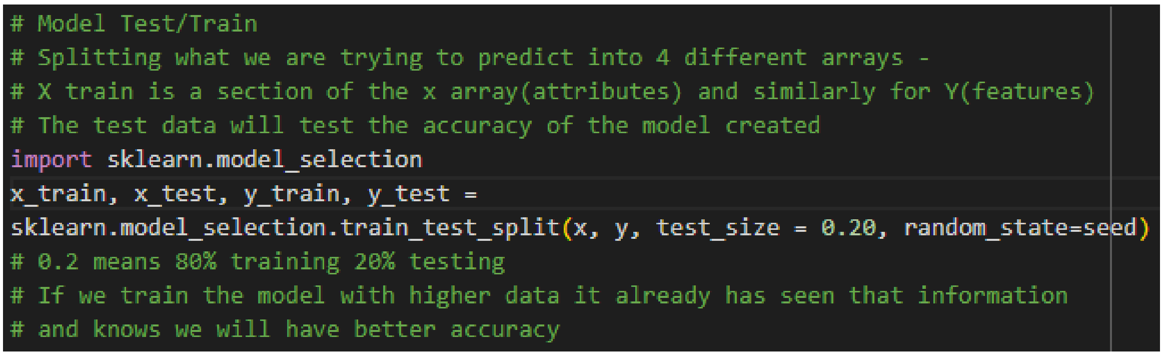
Model Preparation and Development

To prepare this model and develop it we need to convert the data frame into validation subsets by taking a random sample of 80% of the data and training the AI model on it then leaving the remaining 20% for testing.

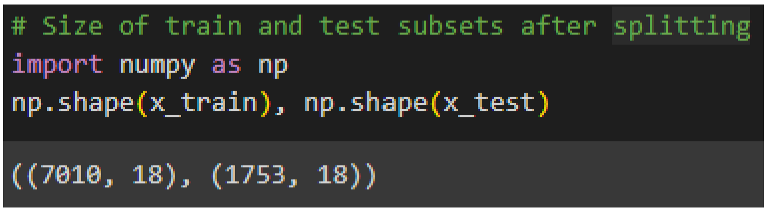
The development of this process is done by creating a test set by dropping all of the rows from the data frame then creating an x and y axis with x being everything but the last column and y being the target class (last column).



```
x = list(zip(Age, Sex, Cholesterol, Heart_Rate, Diabetes, Family_History, Smoking,
             Obesity, Exercise_Hours_Per_Week, Previous_Heart_Problems, Medication_Use,
             Stress_Level, Sedentary_Hours_Per_Day, Income, BMI, Triglycerides,
             Physical_Activity_Days_Per_Week, Sleep_Hours_Per_Day))

y = list(Heart_Attack_Risk)
# Test options and evaluation metric
num_folds = 5
seed = 7
scoring = 'accuracy'
```

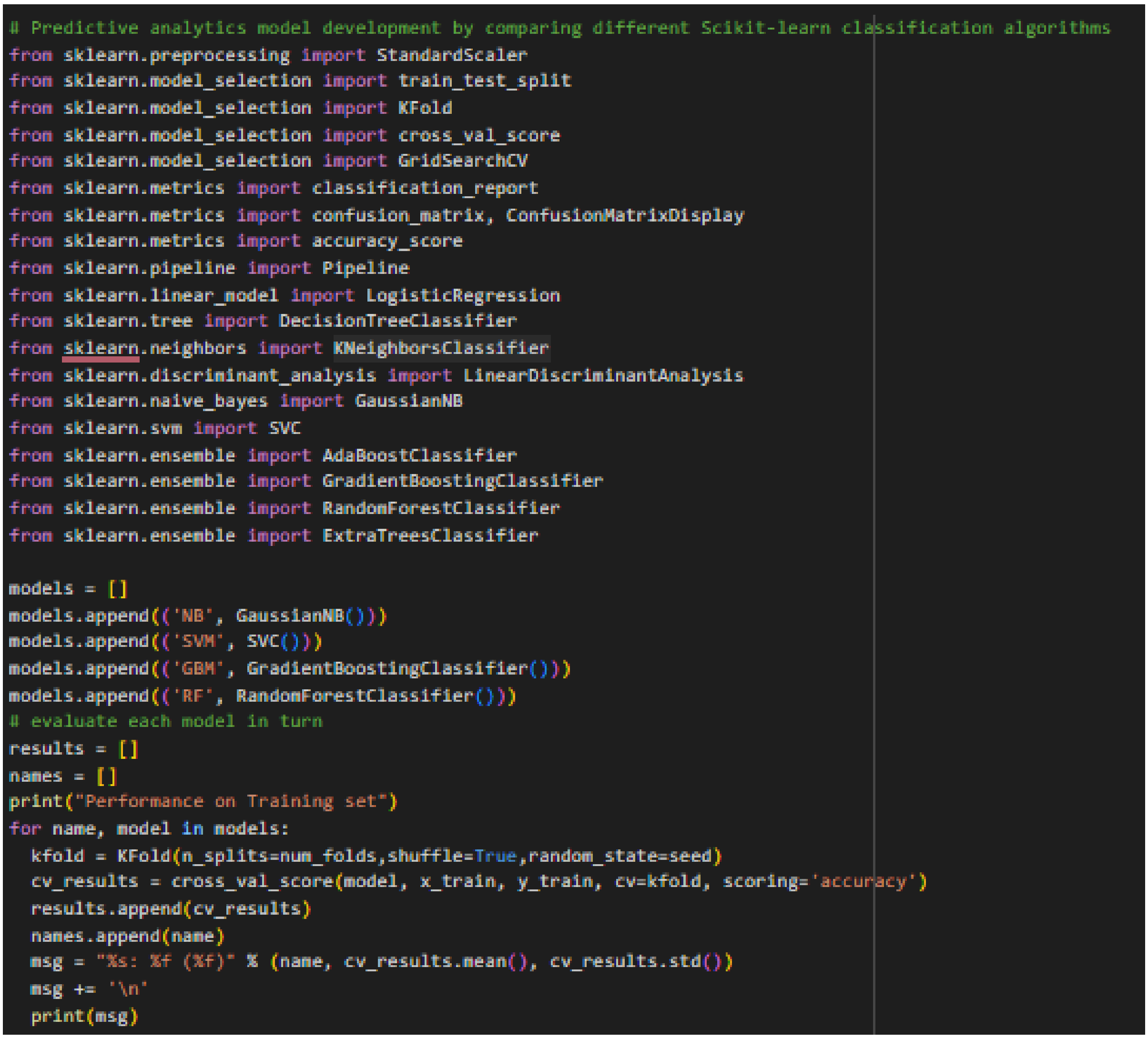


```
# Model Test/Train
# Splitting what we are trying to predict into 4 different arrays -
# X train is a section of the x array(attributes) and similarly for Y(features)
# The test data will test the accuracy of the model created
import sklearn.model_selection
x_train, x_test, y_train, y_test =
sklearn.model_selection.train_test_split(x, y, test_size = 0.20, random_state=seed)
# 0.2 means 80% training 20% testing
# If we train the model with higher data it already has seen that information
# and knows we will have better accuracy
```

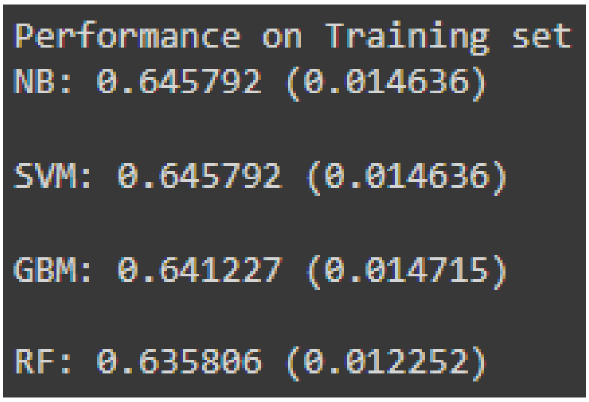


```
# Size of train and test subsets after splitting
import numpy as np
np.shape(x_train), np.shape(x_test)
```

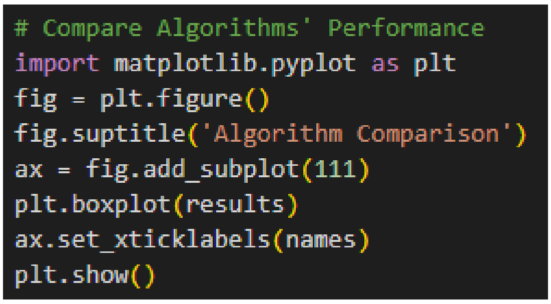
```
((7010, 18), (1753, 18))
```



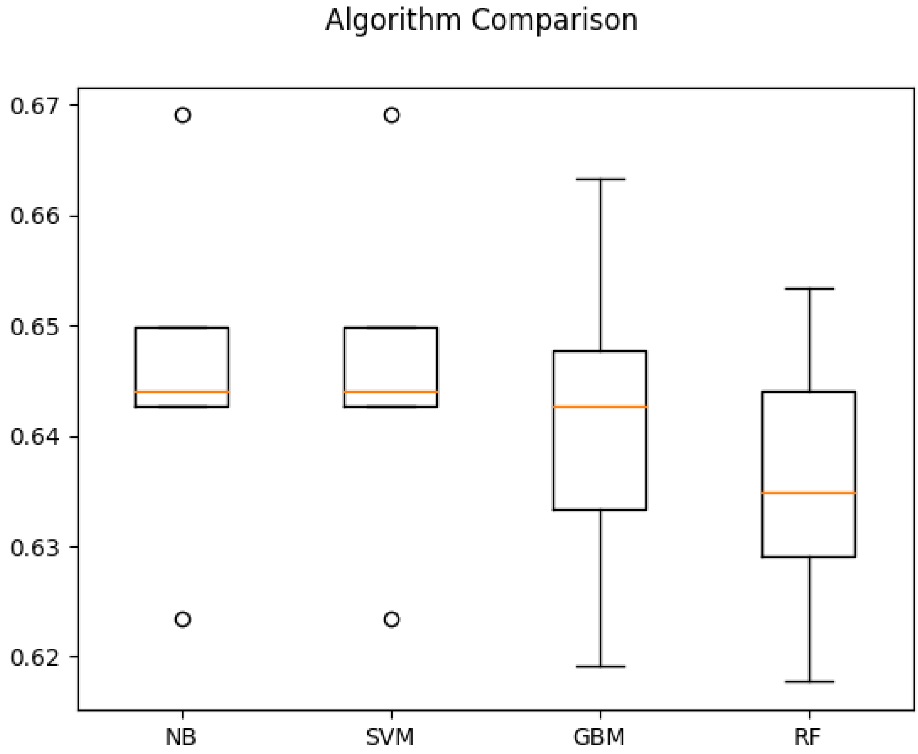
```
# Predictive analytics model development by comparing different Scikit-learn classification algorithms
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier

models = []
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
models.append(('GBM', GradientBoostingClassifier()))
models.append(('RF', RandomForestClassifier()))
# evaluate each model in turn
results = []
names = []
print("Performance on Training set")
for name, model in models:
  kfold = KFold(n_splits=num_folds,shuffle=True,random_state=seed)
  cv_results = cross_val_score(model, x_train, y_train, cv=kfold, scoring='accuracy')
  results.append(cv_results)
  names.append(name)
  msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
  msg += '\n'
  print(msg)
```

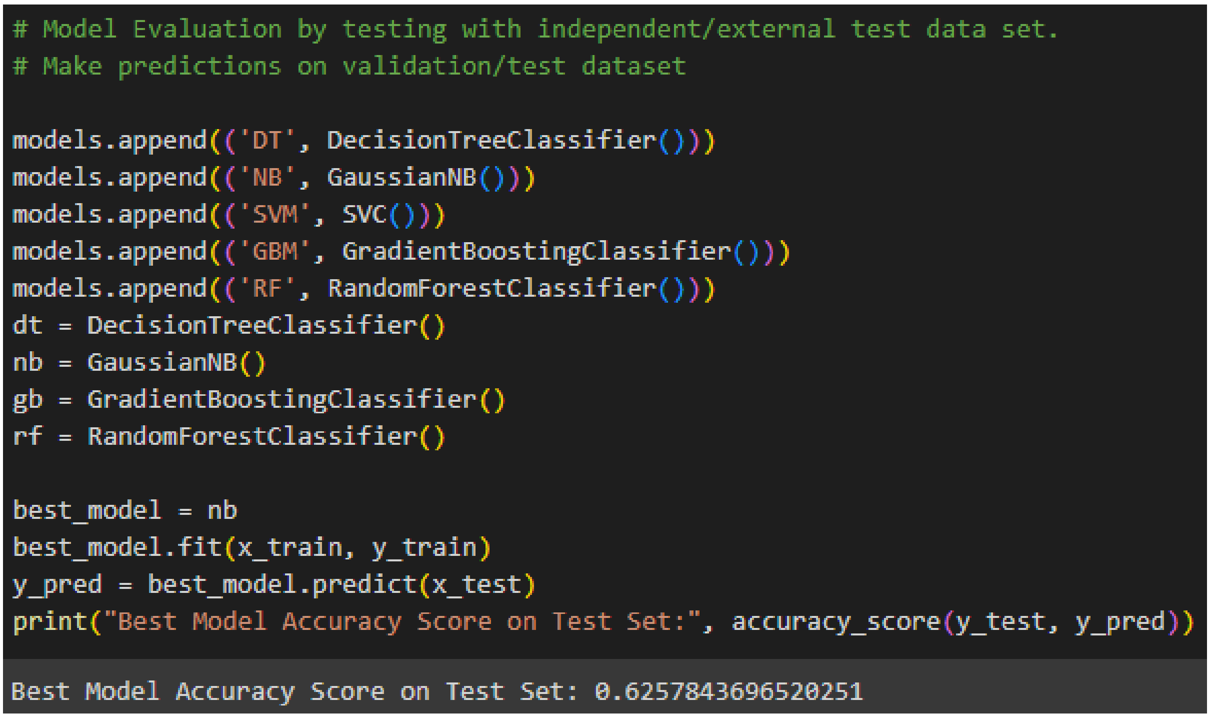


```
Performance on Training set
NB: 0.645792 (0.014636)

SVM: 0.645792 (0.014636)

GBM: 0.641227 (0.014715)

RF: 0.635806 (0.012252)
```



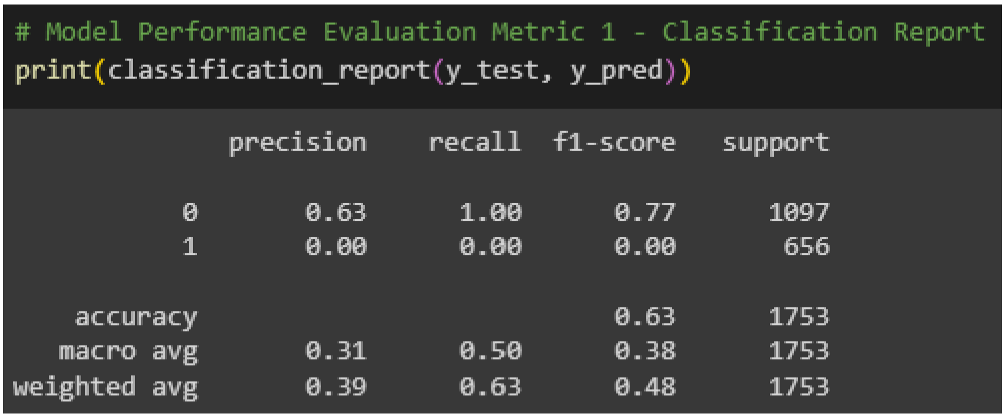
```
# Compare Algorithms' Performance
import matplotlib.pyplot as plt
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```





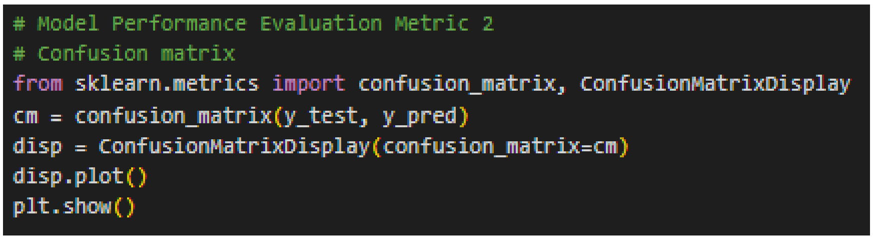
```
# Model Evaluation by testing with independent/external test data set.
# Make predictions on validation/test dataset

models.append(('DT', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
models.append(('GBM', GradientBoostingClassifier()))
models.append(('RF', RandomForestClassifier()))
dt = DecisionTreeClassifier()
nb = GaussianNB()
gb = GradientBoostingClassifier()
rf = RandomForestClassifier()

best_model = nb
best_model.fit(x_train, y_train)
y_pred = best_model.predict(x_test)
print("Best Model Accuracy Score on Test Set:", accuracy_score(y_test, y_pred))
```

```
Best Model Accuracy Score on Test Set: 0.6257843696520251
```

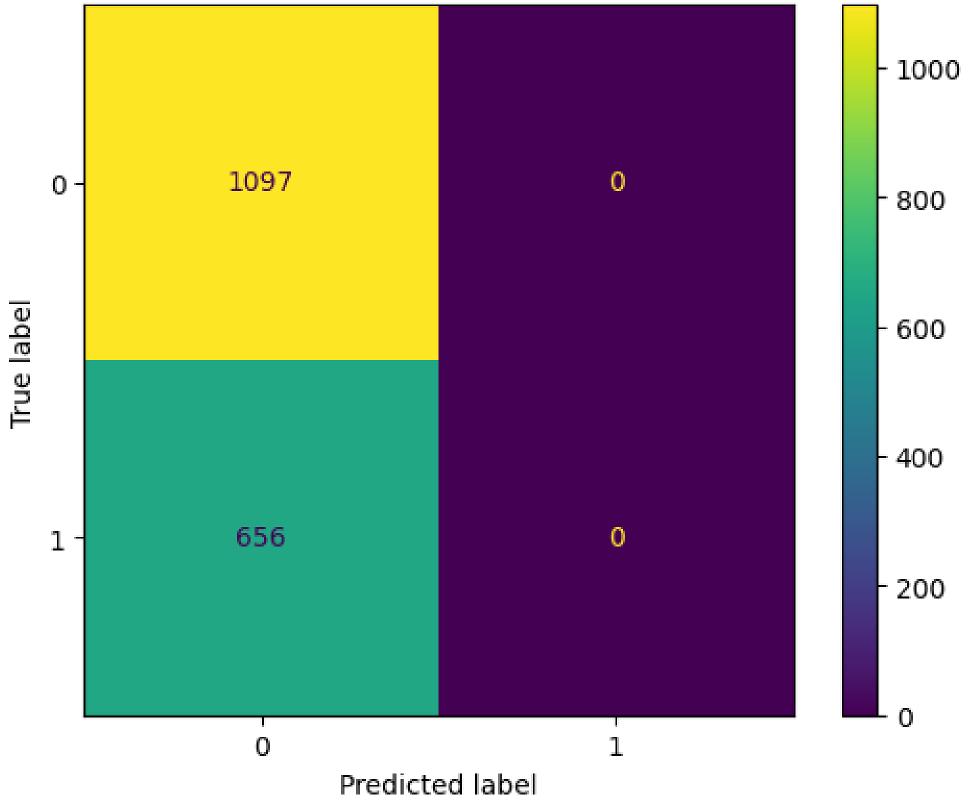


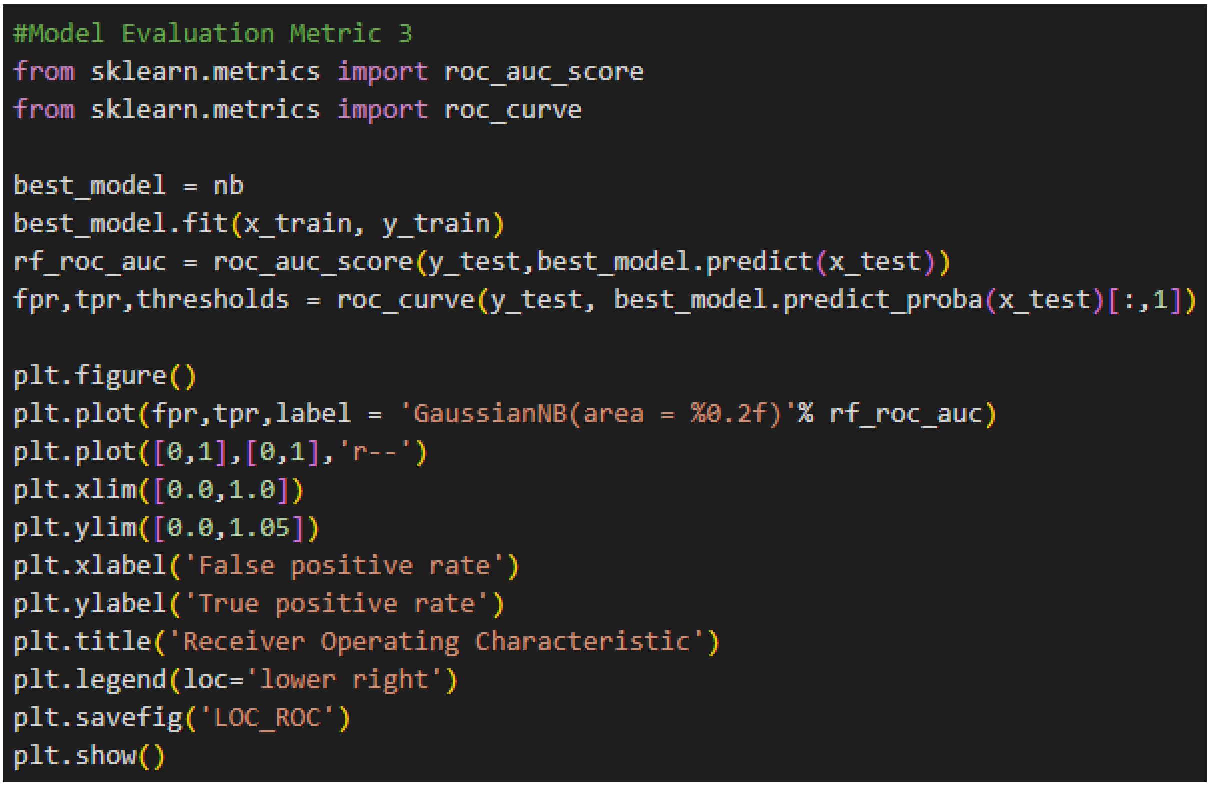
```
# Model Performance Evaluation Metric 1 - Classification Report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.63      1.00      0.77      1097
           1       0.00      0.00      0.00       656

    accuracy                           0.63      1753
   macro avg       0.31      0.50      0.38      1753
weighted avg       0.39      0.63      0.48      1753
```

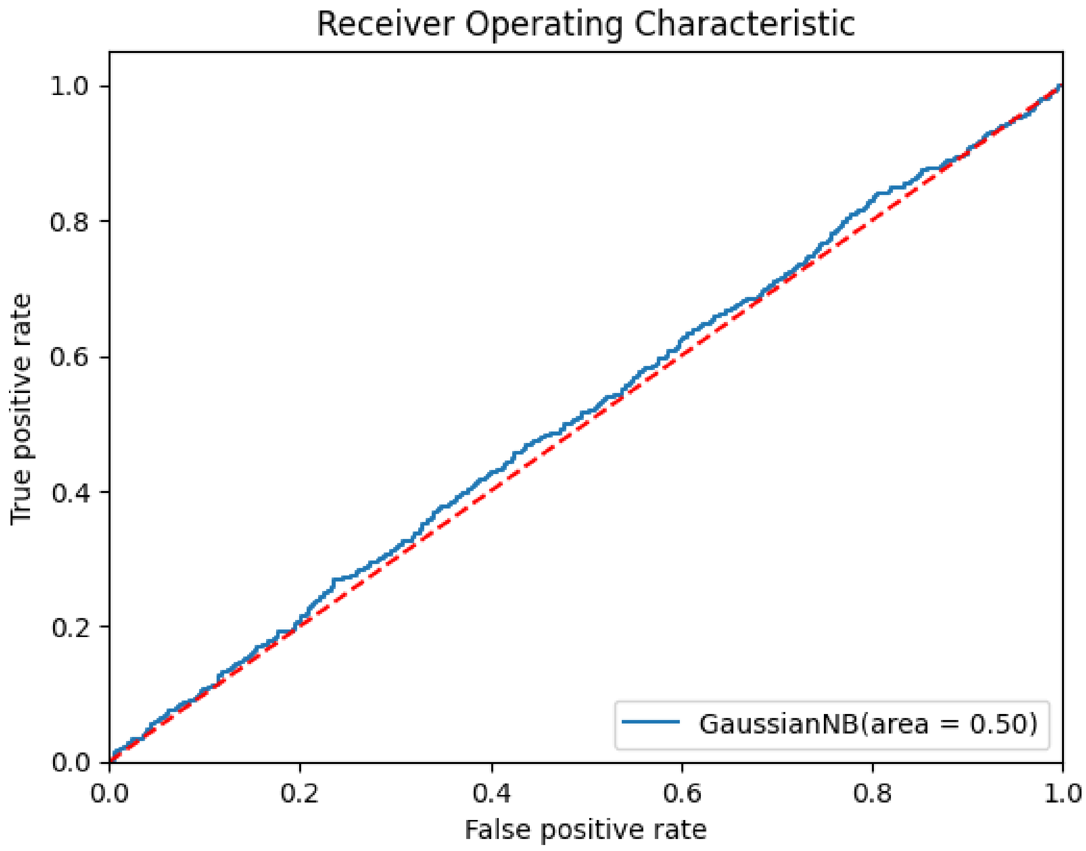


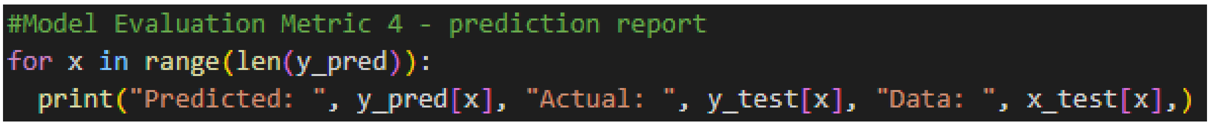
```
# Model Performance Evaluation Metric 2
# Confusion matrix
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.show()
```



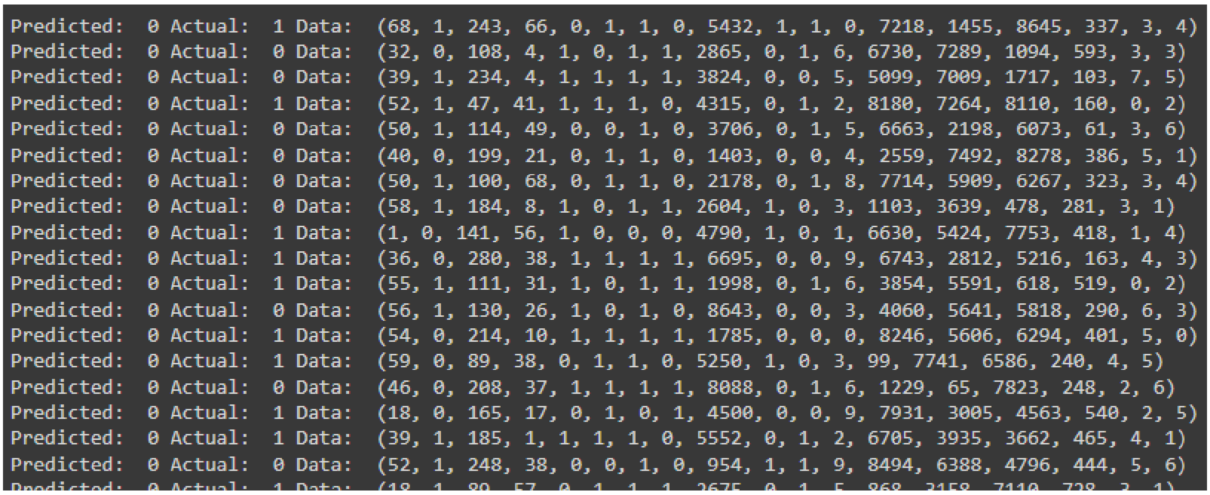


```
#Model Evaluation Metric 3
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve

best_model = nb
best_model.fit(x_train, y_train)
rf_roc_auc = roc_auc_score(y_test,best_model.predict(x_test))
fpr,tpr,thresholds = roc_curve(y_test, best_model.predict_proba(x_test)[:,1])

plt.figure()
plt.plot(fpr,tpr,label = 'GaussianNB(area = %0.2f)'% rf_roc_auc)
plt.plot([0,1],[0,1],'r--')
plt.xlim([0.0,1.0])
plt.ylim([0.0,1.05])
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc='lower right')
plt.savefig('LOC_ROC')
plt.show()
```





```
#Model Evaluation Metric 4 - prediction report
for x in range(len(y_pred)):
  print("Predicted: ", y_pred[x], "Actual: ", y_test[x], "Data: ", x_test[x],)
```



```
Predicted:  0 Actual:  1 Data:  (68, 1, 243, 66, 0, 1, 1, 0, 5432, 1, 1, 0, 7218, 1455, 8645, 337, 3, 4)
Predicted:  0 Actual:  0 Data:  (32, 0, 108, 4, 1, 0, 1, 1, 2865, 0, 1, 6, 6730, 7289, 1094, 593, 3, 3)
Predicted:  0 Actual:  0 Data:  (39, 1, 234, 4, 1, 1, 1, 1, 3824, 0, 0, 5, 5099, 7009, 1717, 103, 7, 5)
Predicted:  0 Actual:  1 Data:  (52, 1, 47, 41, 1, 1, 1, 0, 4315, 0, 1, 2, 8180, 7264, 8110, 160, 0, 2)
Predicted:  0 Actual:  0 Data:  (50, 1, 114, 49, 0, 0, 1, 0, 3706, 0, 1, 5, 6663, 2198, 6073, 61, 3, 6)
Predicted:  0 Actual:  0 Data:  (40, 0, 199, 21, 0, 1, 1, 0, 1403, 0, 0, 4, 2559, 7492, 8278, 386, 5, 1)
Predicted:  0 Actual:  0 Data:  (50, 1, 100, 68, 0, 1, 1, 0, 2178, 0, 1, 8, 7714, 5909, 6267, 323, 3, 4)
Predicted:  0 Actual:  0 Data:  (58, 1, 184, 8, 1, 0, 1, 1, 2604, 1, 0, 3, 1103, 3639, 478, 281, 3, 1)
Predicted:  0 Actual:  1 Data:  (1, 0, 141, 56, 1, 0, 0, 0, 4790, 1, 0, 1, 6630, 5424, 7753, 418, 1, 4)
Predicted:  0 Actual:  1 Data:  (36, 0, 280, 38, 1, 1, 1, 1, 6695, 0, 0, 9, 6743, 2812, 5216, 163, 4, 3)
Predicted:  0 Actual:  1 Data:  (55, 1, 111, 31, 1, 0, 1, 1, 1998, 0, 1, 6, 3854, 5591, 618, 519, 0, 2)
Predicted:  0 Actual:  0 Data:  (56, 1, 130, 26, 1, 0, 1, 0, 8643, 0, 0, 3, 4060, 5641, 5818, 290, 6, 3)
Predicted:  0 Actual:  1 Data:  (54, 0, 214, 10, 1, 1, 1, 1, 1785, 0, 0, 0, 8246, 5606, 6294, 401, 5, 0)
Predicted:  0 Actual:  1 Data:  (59, 0, 89, 38, 0, 1, 1, 0, 5250, 1, 0, 3, 99, 7741, 6586, 240, 4, 5)
Predicted:  0 Actual:  0 Data:  (46, 0, 208, 37, 1, 1, 1, 1, 8088, 0, 1, 6, 1229, 65, 7823, 248, 2, 6)
Predicted:  0 Actual:  1 Data:  (18, 0, 165, 17, 0, 1, 0, 1, 4500, 0, 0, 9, 7931, 3005, 4563, 540, 2, 5)
Predicted:  0 Actual:  1 Data:  (39, 1, 185, 1, 1, 1, 1, 0, 5552, 0, 1, 2, 6705, 3935, 3662, 465, 4, 1)
Predicted:  0 Actual:  0 Data:  (52, 1, 248, 38, 0, 0, 1, 0, 954, 1, 1, 9, 8494, 6388, 4796, 444, 5, 6)
Predicted:  0 Actual:  1 Data:  (18, 1, 89, 57, 0, 1, 1, 1, 2675, 0, 1, 5, 868, 3158, 7110, 728, 3, 1)
```

Implementation and Deployment

Once the best performing AI is chosen for predicting heart attack risk the program is then implemented and then deployed onto the web through streamlit.

1. The program is first implemented onto the web by importing the model and showcasing it using streamlit

2. Then after the program is then deployed onto the web using streamlit

As seen here: https://drive.google.com/drive/folders/1qDi4Y1BpMBY95XxwGvQw7bgQkUOMpLEa?usp=share_link

Conclusion

The completion of this project signifies a significant stride in cardiovascular research and predictive healthcare. By harnessing the power of data-driven predictive analytics, this program has provided a safe and controlled environment for exploring the complex dynamics of heart health and its associated risks. Through the analysis of the dataset, some patterns and correlations have been unearthed, shedding light on various factors contributing to heart attacks.

The successful development of the predictive model demonstrates the potential of artificial intelligence in foreseeing heart attack risks with remarkable accuracy. This achievement not only deepens our understanding of cardiovascular health but also equips healthcare professionals and researchers with a powerful tool for proactive interventions. By identifying high-risk individuals and understanding the nuanced factors that influence heart health, this program paves the way for heart attack prevention and management strategies.

In essence, this project shows the impact of data-driven technologies in the realm of healthcare. With this predictive platform in place, people in the field of cardiology and health now possess a sophisticated instrument to develop targeted initiatives, ultimately enhancing the overall well-being of individuals and communities. The strides made here underscore the potential of AI in reshaping the future of cardiovascular healthcare, offering a promising path toward a healthier and heart-safe society."

References

Banerjee, S. (2023). *Heart Attack Risk Prediction Dataset*. www.kaggle.com. Available at: https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset/data [Accessed 1 Oct. 2023].

Heart Foundation (2020). *Key Statistics: Heart attack | The Heart Foundation*. heartfoundation-prod.azurewebsites.net. Available at: https://www.heartfoundation.org.au/Bundles/For-Professionals/key-statistics-heart-attack.

Journal

Week 10:

I found a different dataset I wish to use for my capstone project, so I requested a change and it was approved, and I have also came up with 5 questions I want to answer when conducting my Exploratory Data Analysis.

Week 11:

I have completed my EDA, Predictive data analysis (PDA) and model preparation development. I have found that GaussianNB was the best AI learning model for my predictions and plan to use it for my implementation and deployment.

Week 12:

I have completed my implementation and deployment using streamlit and made my PowerPoint presentation slides and presented.

Week 13:

I have competed my deployment on streamlit and made a Git Repository with my work for this project.