Course Project – Final Paper
Predicting Treatment for Heart Attack Patients

Karla Reyes
Net ID: krr110
Section: 8
GitHub: https://github.com/karlarreyes/finalpaper

Objective: Create a predictive model for treatment based on clinical and demographic data from heart attack patients

Project Definition

The primary challenge addressed in this project is to predict an effective treatment strategy for heart attack patients based on the data set "Heart Attack Risk Factors" dataset from Kaggle. Throughout this project, I would be using python for statistical analysis and visualization to find correlations between variables and also see the distribution, like we learned in class. I would also be using Pandas to clean my data, analyze it and create models/graphs. I will also be using matplotlib.pyplot to plot my graphs, seaborn, sklearn.esemble, etc., for predictive models and data analysis.
Throughout the project we will see 7 different "predictors" or variables

- Age
- Gender (0 as Female, 1 as Male)
- Smoking Status (0 as Current, 1 as Former, 2 and Never)
- Chest Pain (0 is Asymptomatic, 1 as Atypical Angina, 2 as Non-anginal Pain, 3 as typical Angina)
- Treatment (Lifestyle Changes, Angioplasty, CABG, Medication)

Link: https://www.kaggle.com/datasets/waqi786/heart-attack-dataset/data

Novelty and Importance

This project is important because heart attacks is a leading cause of death globally. Predicting optimal treatment options based on patient data can help improve mortality rates and reduce costs. This project is exciting because I get to explore a health disparity and incorporate what I've learned throughout this course to help me analyze this health disparity. With data science, we are able to use data-driven methods to improve quality of life. It is a combination of machine learning and healthcare applications.
It is important to note some existing issues in current data management practice are unstructured data, privacy/legal concerns, and dealing with bias in algorithms. This however is not much of an issue when it comes to treatment prediction. One thing I will note is that it can be difficult to truly tackle the challenges of treatment optimization through prediction. The hopes of this project are to build on these foundations for further health disease analysis.

Revised Proposal

In my proposal, I first wanted to analyze heart attack risk factors and predict heart attack risks, however after more careful evaluation, creating prediction model for treatment fits best considering these are patient records about factors that can influence the risk of a heart attack and what treatments they are taking (based on several factors).

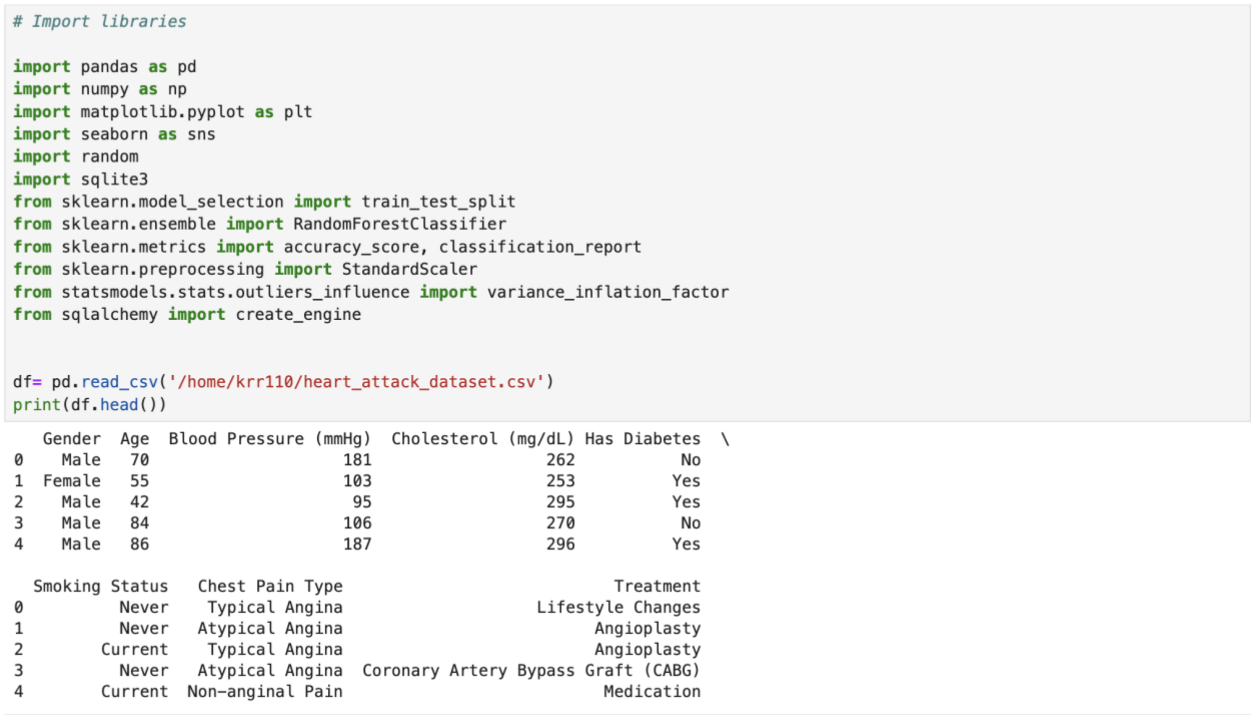
Progress and Contribution

The data I will be using in this project was retrieved from Kaggle, which contains clinical information; age, gender, blood pressure, cholesterol, smoking habits, diabetes status, chest pain type, and treatment. This dataset ensures diversity and an accurate representation of heart attack risk factors across different demographics. This dataset is clean and easy to work with.

- https://www.kaggle.com/datasets/waqi786/heart-attack-dataset/data

Models/Techniques/Algorithms

In this project, the first step consisted of importing libraries I thought were important to the project. Then, I uploaded my dataset and used the code print(df.head()) to see the first couple of rows in each column to ensure my dataset was properly uploaded and get a better look at my data.

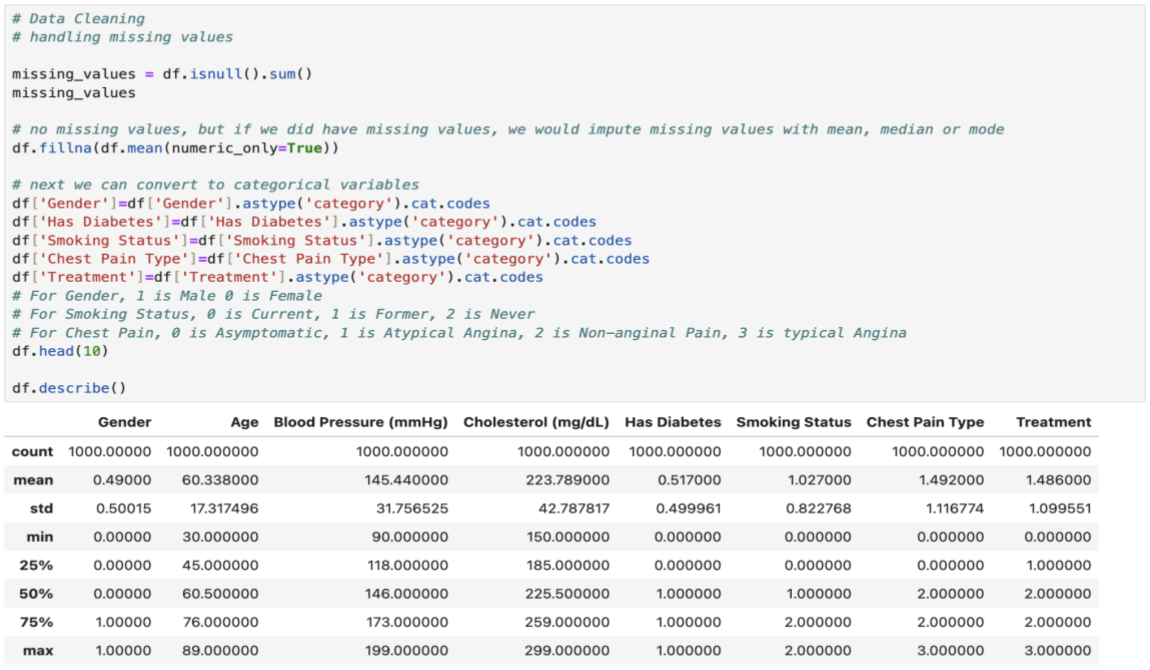


```python
# Import libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import random
import sqlite3
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import StandardScaler
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sqlalchemy import create_engine


df= pd.read_csv('/home/krr110/heart_attack_dataset.csv')
print(df.head())
```

```
   Gender  Age  Blood Pressure (mmHg)  Cholesterol (mg/dL) Has Diabetes  \
0    Male   70                    181                  262           No
1  Female   55                    103                  253          Yes
2    Male   42                     95                  295          Yes
3    Male   84                    106                  270           No
4    Male   86                    187                  296          Yes

  Smoking Status   Chest Pain Type                         Treatment
0          Never    Typical Angina                 Lifestyle Changes
1          Never   Atypical Angina                       Angioplasty
2        Current    Typical Angina                       Angioplasty
3          Never   Atypical Angina  Coronary Artery Bypass Graft (CABG)
4        Current  Non-anginal Pain                        Medication
```

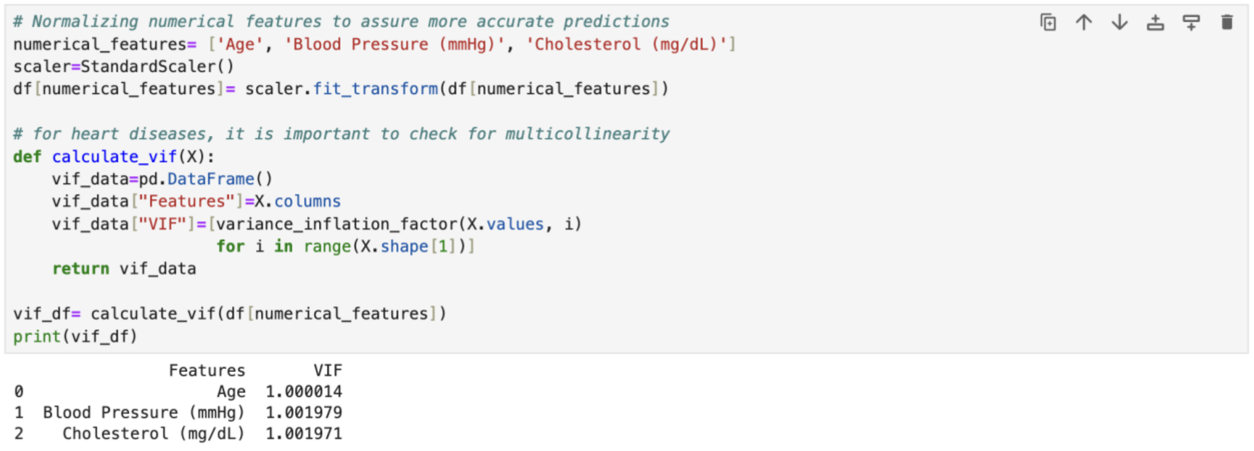
The second step, I proceeded to clean my data. In this case, the dataset from Kaggle was already clean and easy to use, however I included the steps I would have taken if my dataset wasn’t already clean. These steps include handling missing numbers where I would evaluate how many missing numbers I have per column and then insert the mean for my numerical variables. Then, I proceeded to convert specific columns such as ‘Gender’, ‘Has Diabetes’, ‘Smoking Status’, ‘Chest Pain Type’, and ‘Treatment’ as categorical variables for easy analysis. Then I proceeded to use the describe function to examine minimums, maximums, means, etc.

Then, I proceeded to normalize my numerical features to assure more accurate predictions and ensuring each features contributes equally in my model.



```python
# Data Cleaning
# handling missing values

missing_values = df.isnull().sum()
missing_values

# no missing values, but if we did have missing values, we would impute missing values with mean, median or mode
df.fillna(df.mean(numeric_only=True))

# next we can convert to categorical variables
df['Gender']=df['Gender'].astype('category').cat.codes
df['Has Diabetes']=df['Has Diabetes'].astype('category').cat.codes
df['Smoking Status']=df['Smoking Status'].astype('category').cat.codes
df['Chest Pain Type']=df['Chest Pain Type'].astype('category').cat.codes
df['Treatment']=df['Treatment'].astype('category').cat.codes
# For Gender, 1 is Male 0 is Female
# For Smoking Status, 0 is Current, 1 is Former, 2 is Never
# For Chest Pain, 0 is Asymptomatic, 1 is Atypical Angina, 2 is Non-anginal Pain, 3 is typical Angina
df.head(10)

df.describe()
```

	Gender	Age	Blood Pressure (mmHg)	Cholesterol (mg/dL)	Has Diabetes	Smoking Status	Chest Pain Type	Treatment
count	1000.00000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.49000	60.338000	145.440000	223.789000	0.517000	1.027000	1.492000	1.486000
std	0.50015	17.317496	31.756525	42.787817	0.499961	0.822768	1.116774	1.099551
min	0.00000	30.000000	90.000000	150.000000	0.000000	0.000000	0.000000	0.000000
25%	0.00000	45.000000	118.000000	185.000000	0.000000	0.000000	0.000000	1.000000
50%	0.00000	60.500000	146.000000	225.500000	1.000000	1.000000	2.000000	2.000000
75%	1.00000	76.000000	173.000000	259.000000	1.000000	2.000000	2.000000	2.000000
max	1.00000	89.000000	199.000000	299.000000	1.000000	2.000000	3.000000	3.000000

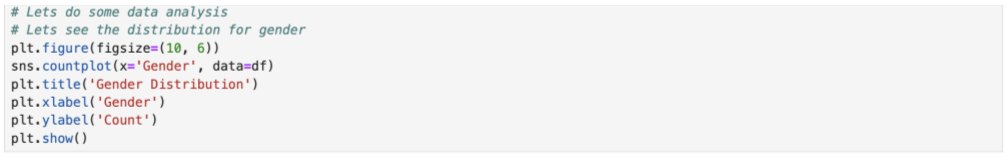


```python
# Normalizing numerical features to assure more accurate predictions
numerical_features= ['Age', 'Blood Pressure (mmHg)', 'Cholesterol (mg/dL)']
scaler=StandardScaler()
df[numerical_features]= scaler.fit_transform(df[numerical_features])

# for heart diseases, it is important to check for multicollinearity
def calculate_vif(X):
    vif_data=pd.DataFrame()
    vif_data["Features"]=X.columns
    vif_data["VIF"]=[variance_inflation_factor(X.values, i)
                    for i in range(X.shape[1])]
    return vif_data

vif_df= calculate_vif(df[numerical_features])
print(vif_df)
```

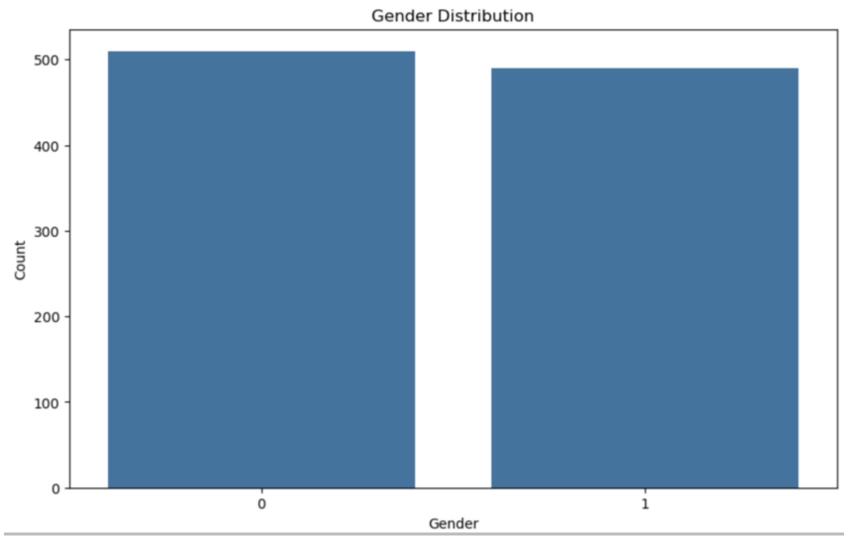
```
               Features       VIF
0                   Age  1.000014
1  Blood Pressure (mmHg)  1.001979
2    Cholesterol (mg/dL)  1.001971
```

Data Analysis

After my data has been cleaned and prepped to start predictions, I need to do some data analysis and visualizations first to certain distributions. The first distribution I decided to plot was the distribution for gender. It is important to analyze distribution to see if there are even distributions for accurate analysis and predictions. The distribution of gender is fairly even, implying a relatively even amount of male and females in this data set.

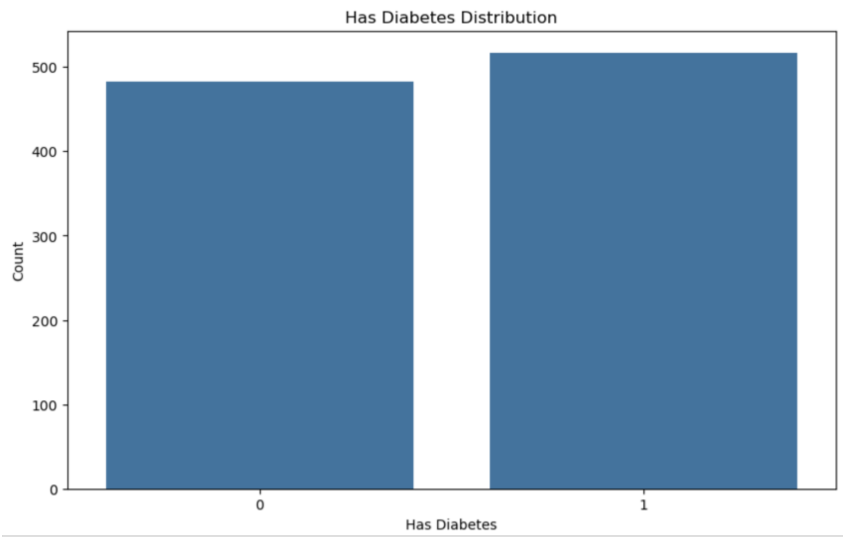


```
# Lets do some data analysis
# Lets see the distribution for gender
plt.figure(figsize=(10, 6))
sns.countplot(x='Gender', data=df)
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```

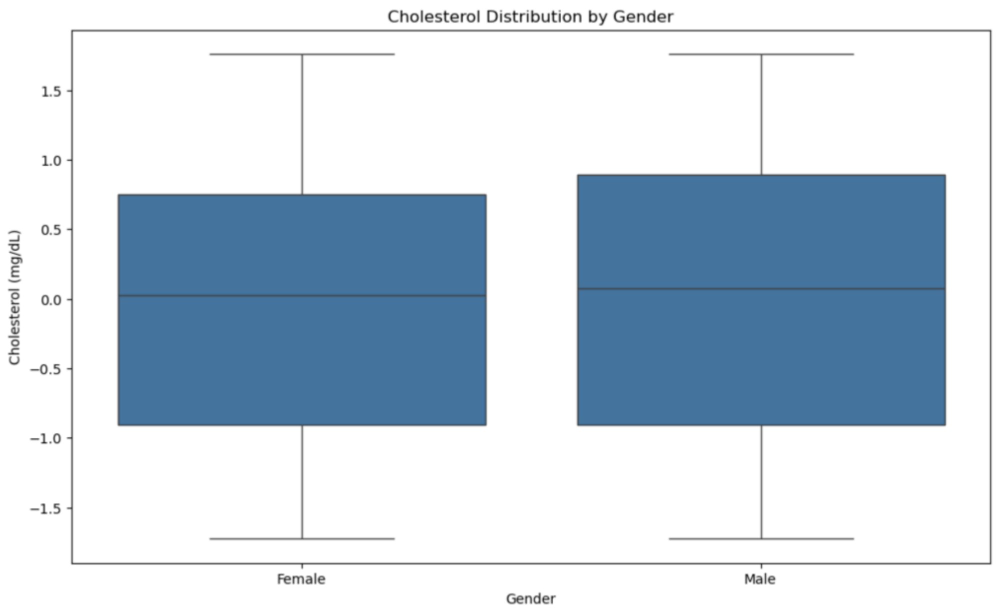


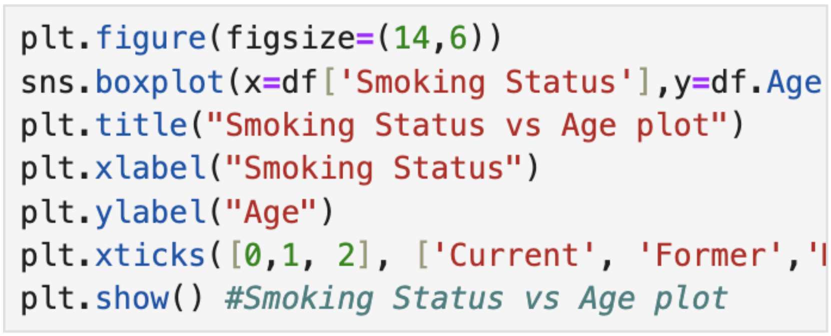
The next distribution I analyzed was the distribution of diabetes, which appears to also be fairly balanced between those who have diabetes and those who do not. There is a relatively even distribution between these two categories.

```
# Distribution of Diabetes
plt.figure(figsize=(10, 6))
sns.countplot(x='Has Diabetes', data=df)
plt.title('Has Diabetes Distribution')
plt.xlabel('Has Diabetes')
plt.ylabel('Count')
plt.show()
```

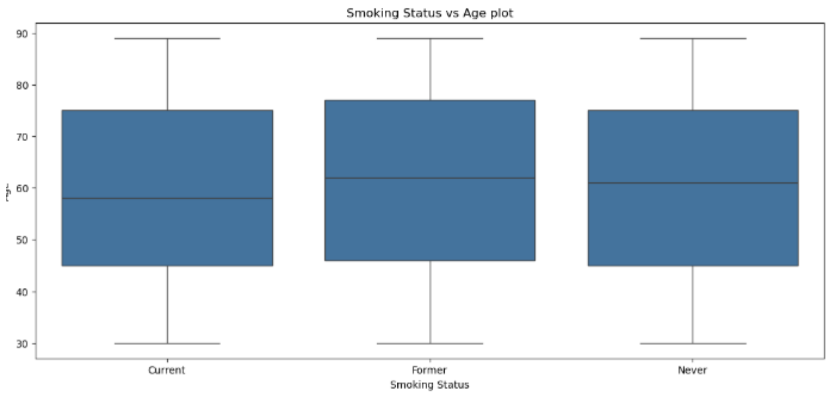


Next, I decided to visualize the distribution of cholesterol by gender and another distribution of Smoking Status vs Age. For both plots, there is also a relatively even distribution between each category however it is important to note in the distribution for smoking where former and current smokers take up about 66% or 2/3 of the data which implies that smoking is a very important and very common health attack risk factor.



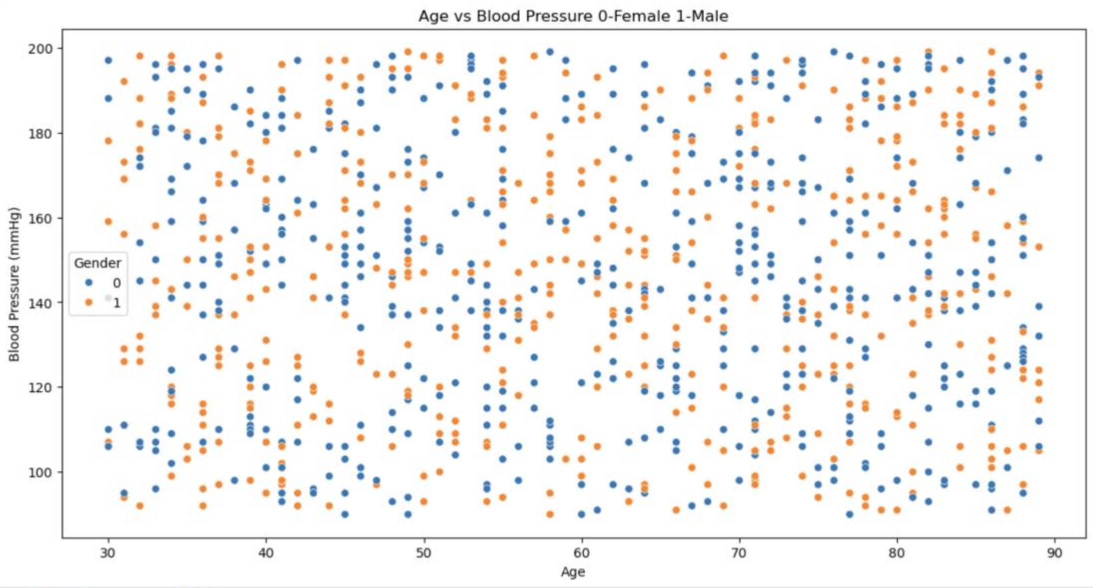


```
plt.figure(figsize=(14,6))
sns.boxplot(x=df['Smoking Status'],y=df.Age
plt.title("Smoking Status vs Age plot")
plt.xlabel("Smoking Status")
plt.ylabel("Age")
plt.xticks([0,1, 2], ['Current', 'Former','
plt.show() #Smoking Status vs Age plot
```

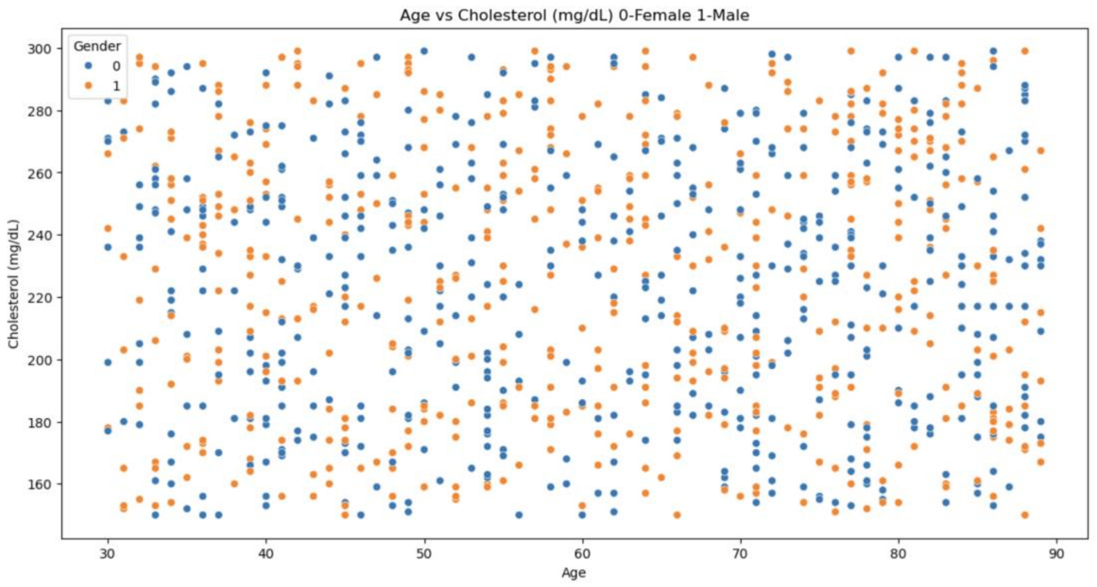


Then I decided to use scatter plots to visualize distributions deeper. I created a scatter plot between Age vs Blood Pressure between male and female and Age vs Cholesterol between male and female. Cholesterol levels and Blood Pressure seem to vary between age and there's a similar pattern for both genders. These graph suggest there is variation but there isn't a very clear trend with age.

```
plt.figure(figsize=(14,7))
sns.scatterplot(x=df['Age'],y=df['Blood Pressure (mmHg)'],hue='Gender',data=df)
plt.title("Age vs Blood Pressure 0-Female 1-Male")
plt.xlabel("Age")
plt.ylabel("Blood Pressure (mmHg)")
plt.show()
```

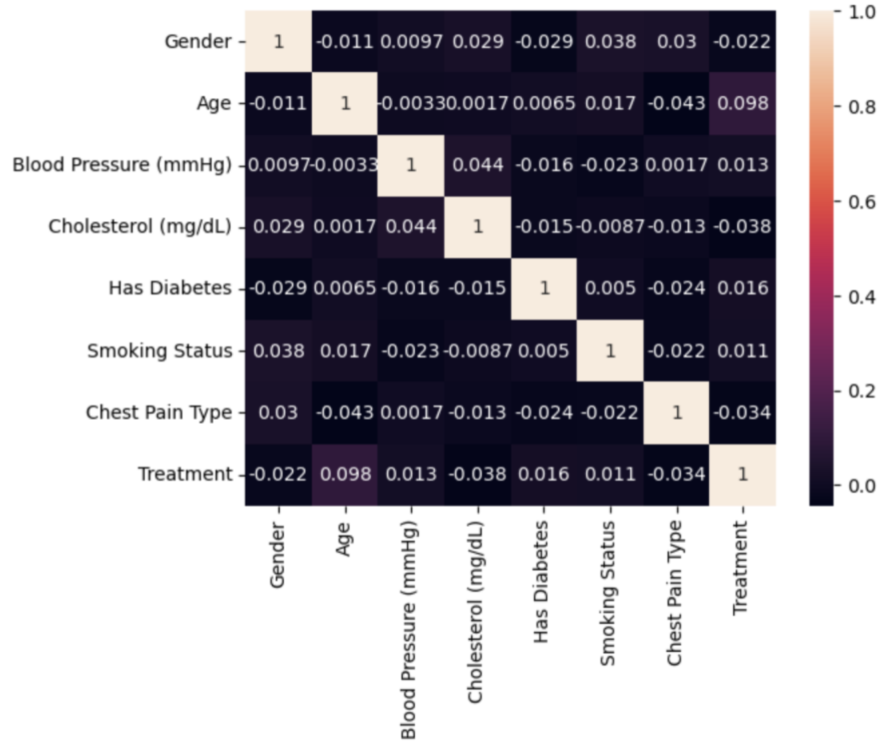


```
plt.figure(figsize=(14,7))
sns.scatterplot(x=df['Age'],y=df['Cholesterol (mg/dL)'],hue='Gender',data=df)
plt.title("Age vs Cholesterol (mg/dL) 0-Female 1-Male")
plt.xlabel("Age")
plt.ylabel("Cholesterol (mg/dL)")
plt.show()
```



Finally, before my prediction model, I created a correlation heatmap, where we visualize the correlations between each variable. This allows me to potentially identify any issues such as multicollinearity where there is high correlation between variables in data. This allows me to build a more accurate predictive model by selecting relevant features and avoiding redundancy.

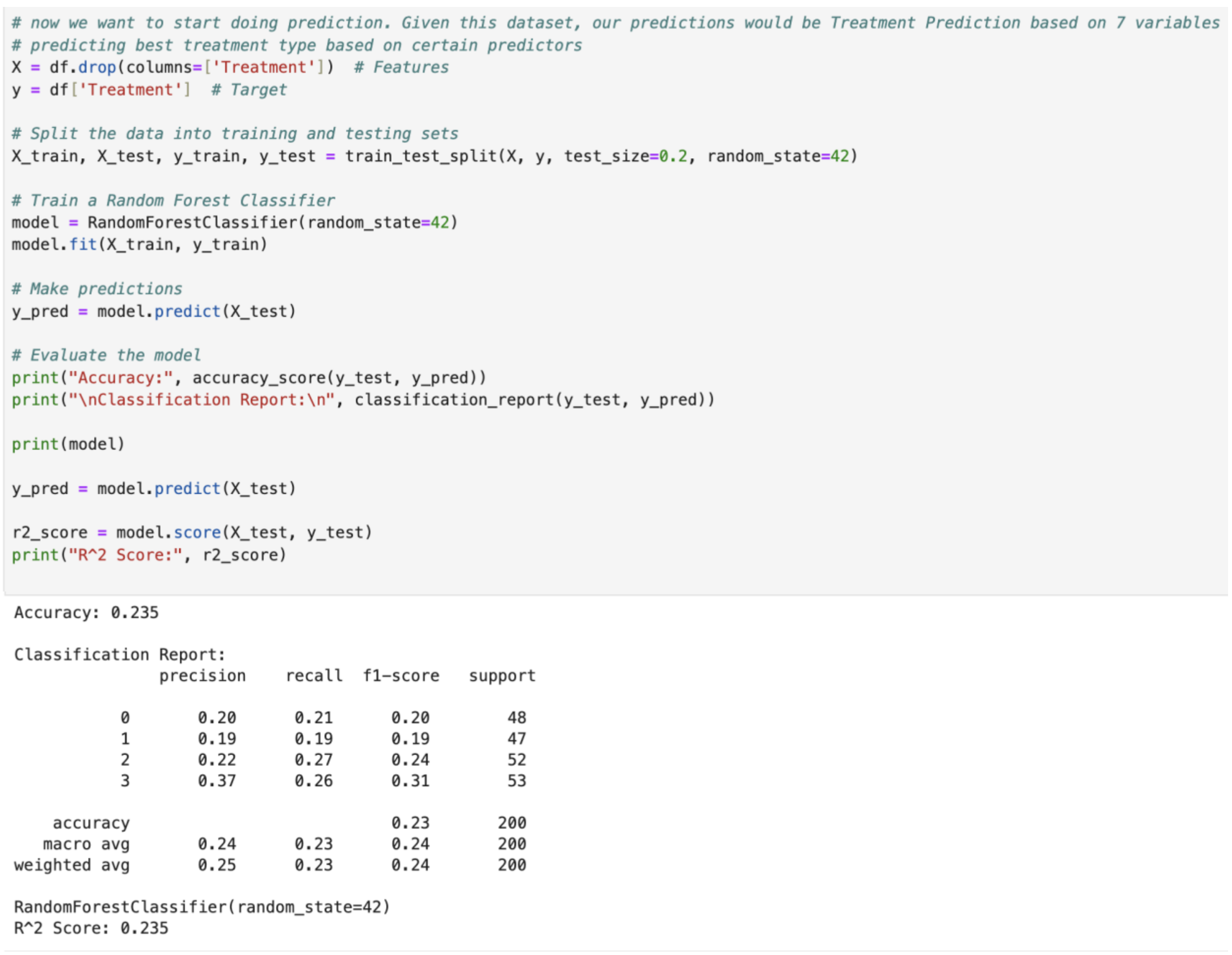
```python
# Lets create a heatmap to portray the correlation between each column
sns.heatmap(df.corr(),annot=True)
plt.show()
```



The results of my heatmap portray very little positive and negative correlations between each variable, so a weak correlation. This indicates that these health factors are independent of each other and the variables that do have a very weak factor still suggest that a variable cannot reliably predict another variable.

Results/ Key findings

Using Random Forest Classifier, I created my Prediction table and analyzed it's accuracy. The model was able to correctly classify about 24% of the samples in the test set. This is very low, as the proportion of true positive predictions is very low. Overall, there was a poor performance of the model which implies we should explore more additional features and other external data to improve the data or experiment with other machine learning algorithms.



```python
# now we want to start doing prediction. Given this dataset, our predictions would be Treatment Prediction based on 7 variables
# predicting best treatment type based on certain predictors
X = df.drop(columns=['Treatment'])  # Features
y = df['Treatment']  # Target

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

# Train a Random Forest Classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

print(model)

y_pred = model.predict(X_test)

r2_score = model.score(X_test, y_test)
print("R^2 Score:", r2_score)
```

```
Accuracy: 0.235

Classification Report:
               precision    recall  f1-score   support

           0       0.20      0.21      0.20        48
           1       0.19      0.19      0.19        47
           2       0.22      0.27      0.24        52
           3       0.37      0.26      0.31        53

    accuracy                           0.23       200
   macro avg       0.24      0.23      0.24       200
weighted avg       0.25      0.23      0.24       200

RandomForestClassifier(random_state=42)
R^2 Score: 0.235
```

- The numbers 0-3 imply the different type of treatments
  - 0 – Angioplasty
  - 1- CABG
  - 2 – Lifestyle changes
  - 3- Medication

Advantages/Limitations of my approach

There are several advantages/Limitations to my approach. Throughout this project, I was able to explore and analyze an important global health disparity. The advantage was exploring important heart risk factors and analyzing the distribution between each one. However, the main goal of the project was the predict what treatment will be best for a patient considering the state of these predictors. However, the model did not perform well which implies there's other factors and research we need to explore more. It is a complex issue to truly analyze correctly, but it is something that brings us closer to finding real-life solutions.