A SPONSORED PROJECT REPORT ON

"HEART DISEASE ANALYTICS"

Submitted in partial fulfillment of the requirement for the award of

Bachelor of Engineering

In

Computer Science and Engineering

Punyashlok Ahilyadevi Holkar Solapur University

By

Shreyas Deodhare TECSE-27
Kushal Gaikwad TECSE-28
Mitali Gangwal TECSE-29
Akarsha Jain TECSE-30

Under Guidance Of

Mentor: **Mr. Rajshekhar Bashetti, Associate Director at IQVIA INDIA, Bangalore, India**

Project Guide:

Mr. M.A. Mahant

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WALCHAND INSTITUE OF TECHNOLOGY
SOLAPUR - 413006
(2021-2022)

CERTIFICATE

This is to certify that the Mini-Project entitled

“ HEART DISEASE ANALYTICS”

Is

Submitted By

Shreyas Deodhare TECSE-27

Kushal Gaikwad TECSE-28

Mitali Gangwal TECSE-29

Akarsha Jain TECSE-30

(Prof. Mr. M.A Mahant)
Guide

(Dr. Mrs.A.M.Pujar)
***Head* Dept of Computer Sci. & Engg**

(Dr. V .A.Athavale)
Principal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WALCHAND INSTITUE OF TECHNOLOGY
SOLAPUR
(2021-2022)

INDEX

Abstract

According to the CDC, heart disease is one of the leading causes of death for people of most races in the US (African Americans, American Indians and Alaska Natives, and white people). About half of all Americans (47%) have at least 1 of 3 key risk factors for heart disease: high blood pressure, high cholesterol, and smoking. Other key indicator includes diabetic status, obesity (high BMI), not getting enough physical activity, or drinking too much alcohol. Detecting and preventing the factors that have the greatest impact on heart disease is very important in healthcare. Computational developments, in turn, allow the application of machine learning methods to detect "patterns" from the data that can predict a patient's condition. Extracting medical data is progressively becoming more and more necessary for the prediction and treatment of high death rates due to heart attacks. Terabytes of data are produced every day. Quality services are needed to avoid poor clinical decisions that lead to disastrous consequences. Hospitals can make use of appropriate decision support systems thus minimizing the cost of clinical tests. Now-a-day hospitals employ hospital information systems to manage patient data. An enormous amount of data generated by the health care industry is not effectively used. Some new approach is necessary to decrease the expense and predict the heart disease in an easy.

Today, cardiovascular diseases are the leading cause of death worldwide with 17.9 million deaths annually, as per the World Health Organization reports. Various unhealthy activities are the reason for the increase in the risk of heart disease like high cholesterol, obesity, increase in triglycerides levels, hypertension, etc. There are certain signs which the American Heart Association lists like the persons having sleep issues, a certain increase and decrease in heart rate (irregular heartbeat), swollen legs, and in some cases weight gain occurring quite fast; it can be 1-2 kg daily . All these symptoms resemble different diseases that also like it occurs in the aging persons, so it becomes a difficult task to get a correct diagnosis, which results in a fatality in near future. The objective of this paper is to analyze

various research works done on heart disease prediction and classification using various machine learning and deep learning techniques and to conclude which techniques are effective and accurate.

This Sponsored Project Report is based on the data analysis of heart disease work.

Introduction

The following were the objectives with which the project was started :

1) To build a dashboard for enabling a smart decision-making process.

2) Implement a cost-effective solution to reduce the impact of the disease.

Mapping with POs and PSOs :

Project Outcomes	
PO 1	Knowledge
PO 2	Problem Analysis
PO 3	Design
PO 4	Investigation
PO 5	Team
PO 6	Communication

Background

Heart disease is the leading cause of death in the United States, according to the Centres for Disease Control and Prevention (CDC)Trusted Source. In the United States, 1 in every 4 deaths is the result of heart disease. That's about 610,000 people who die from the condition each year. Heart disease doesn't discriminate. It's the leading cause of death for several populations, including white people, Hispanics, and Black people. Almost half of Americans are at risk for heart disease, and the numbers are rising. Learn more about the increase in heart disease rates. While heart disease can be deadly, it's also preventable in most people. By adopting healthy lifestyle habits early, you can potentially live longer with a healthier heart.

References:

A. *M. Amiri and G. Armano, "Early diagnosis of heart disease using classification and regression trees," The 2013 International Joint Conference on Neural Networks (IJCNN), 2013, pp. 1-4, DOI: 10.1109/IJCNN.2013.6707080.*

B. *T. P. Pushpavathi, S. Kumari and N. K. Kubra, "Heart Failure Prediction by Feature Ranking Analysis in Machine Learning," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 915-923, DOI: 10.1109/ICICT50816.2021.9358733.*

Technologies Used

1. Microsoft Excel

Excel is a spreadsheet program from Microsoft and a component of its Office product group for business applications. Microsoft Excel enables users to format, organize and calculate data in a spreadsheet. Excel uses a large collection of cells formatted to organize and manipulate data and solve mathematical functions. Users can arrange data in the spreadsheet using graphing tools, pivot tables, and formulas.

The best Features of Microsoft Excel are:

- Collection and verification of business data
- Business analysis
- Data entry and storage
- Data analysis
- Performance reporting
- Strategic analysis
- Accounting and budgeting
- Administrative and managerial management
- Account management
- Project management
- Office administration.

2. Python

Python is the third most popular programming language in the world, behind only Java and C. There are many reasons for the ubiquity of Python, including:

- Its ease of use. For those who are new to coding and programming, Python can be an excellent first step. It’s relatively easy to learn, making it a great way to start building your programming knowledge.
- Its simple syntax. Python is relatively easy to read and understand, as its syntax is more like English. Its straightforward layout means that you can work out what each line of code is doing.
- Its thriving community. As it’s an open-source language, anyone can use Python to code. What’s more, there is a community that supports and develops the ecosystem, adding their own contributions and libraries.
- Its versatility. As we’ll explore in more detail, there are many uses for Python. Whether you’re interested in data visualisation, artificial intelligence or web development, you can find a use for the language.

Description and Working of Project

Architecture & Design:

In Excel, we created visualizations for certain dimensions and got out various insights. Created a dashboard describing the relation between various entries such as relation between the heart analysis and gender ,kidney disease and age , sleep time and age and diabetes and so on.

Using Python we have built a prediction model on the heart dataset which predicts whether the person has heart disease or not.

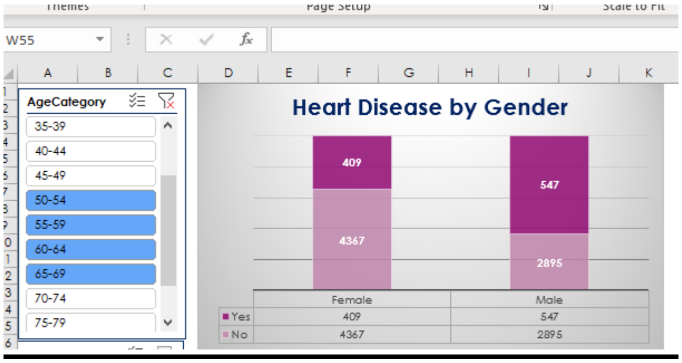
Data Source:

Originally, the dataset comes from the CDC and is a major part of the Behavioral Risk Factor Surveillance System (BRFSS), which conducts annual telephone surveys to gather data on the health status of U.S. residents. As the CDC describes: "Established in 1984 with 15 states, BRFSS now collects data in all 50 states as well as the District of Columbia and three U.S. territories. BRFSS completes more than 400,000 adult interviews each year, making it the largest continuously conducted health survey system in the world.". The most recent dataset (as of February 15, 2022) includes data from 2020. It consists of 401,958 rows and 279 columns. The vast majority of columns are questions asked to respondents about their health status, such as "Do you have serious difficulty walking or climbing stairs?" or "Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes]". In this dataset, I noticed many different factors (questions) that directly or indirectly influence heart disease, so I decided to select the most relevant variables from it and do some cleaning so that it would be usable for machine learning projects.

Analysis:

- What is the effect on sleep time of diabetic patients by age?
- What is the count of males and females who has Heart Disease?
- What is the total number of Heart Disease cases in different nations?
- What is the total number of Heart Disease cases in Asia?
- What is the total number of Recovered cases in India?
- Can we Forecast Trend-Line for Heart Disease Cases?
- Find the Highest Heart Disease Cases at International Level?
- Find the Age group with the Least Heart Disease Cases?
- Can we find the variation in Heart Disease Cases around the World
- Can we forecast the sheet for future reference?

Screenshots & Results



	Female	Male
Yes	409	547
No	4367	2895

Fig. Bar Chart of Heart Disease By Gender

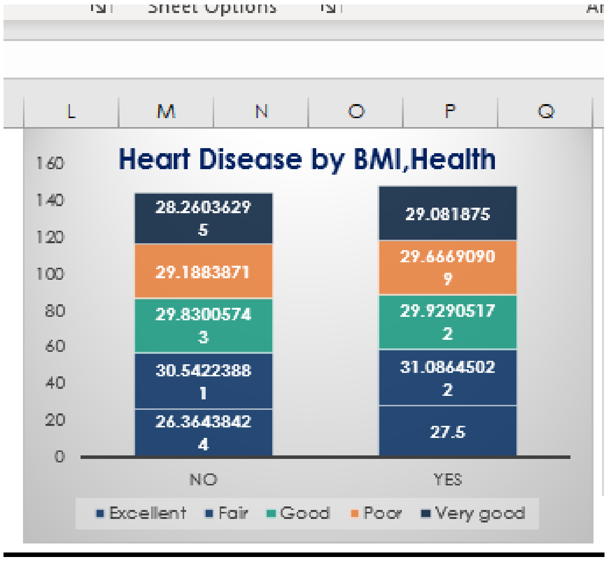


Fig. Stack wise representation of BMI and Health

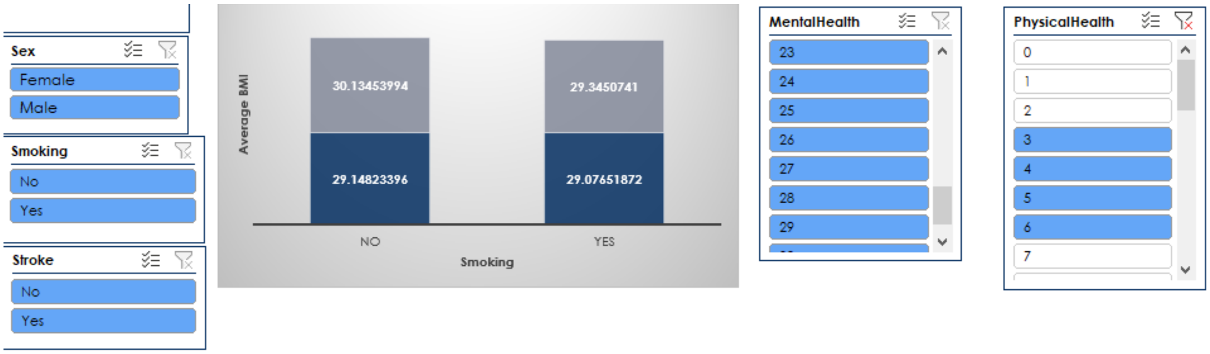


Fig. Smoking vs BMI

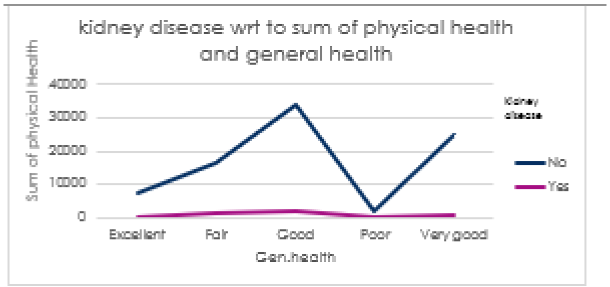
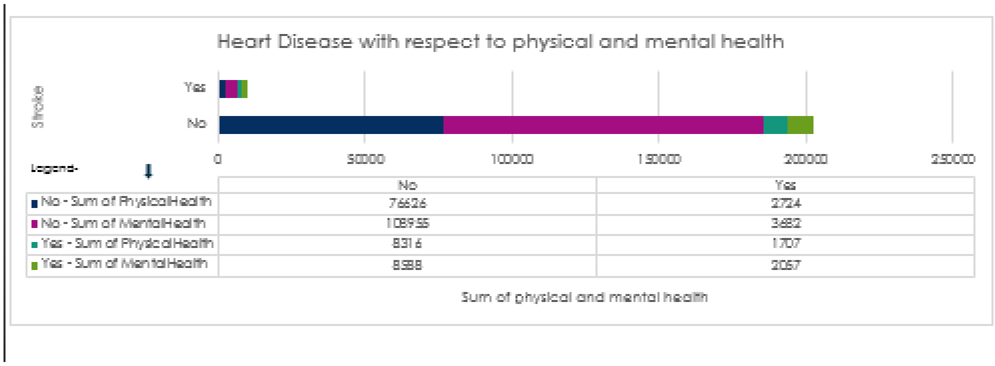


Fig. Physical and General Health

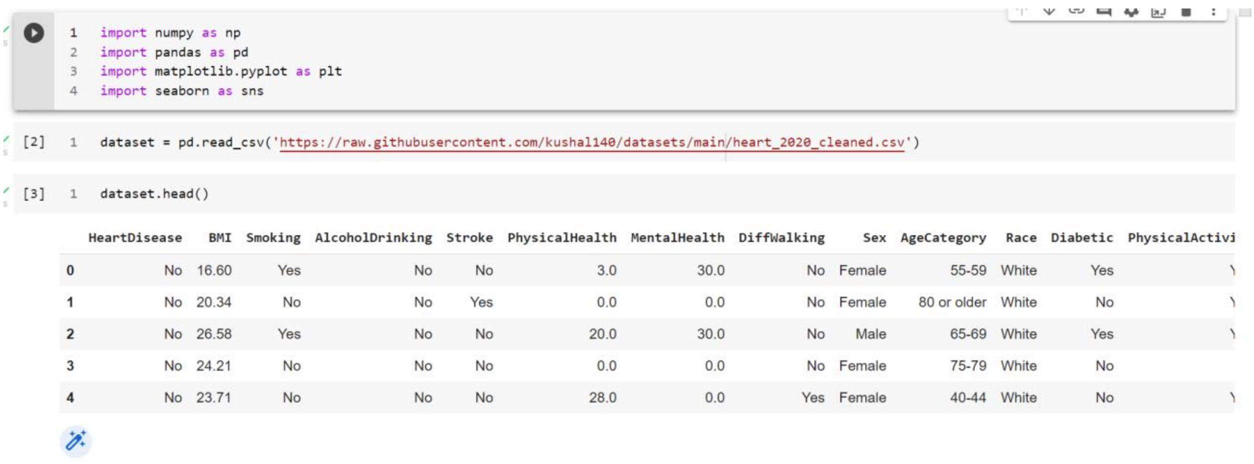


	No	Yes
■ No - Sum of PhysicalHealth	76626	2724
■ No - Sum of MentalHealth	108955	3682
■ Yes - Sum of PhysicalHealth	8316	1707
■ Yes - Sum of MentalHealth	8588	2067

Fig. Heart Disease w.r.t Physical and Mental Health

Prediction ::

Code::

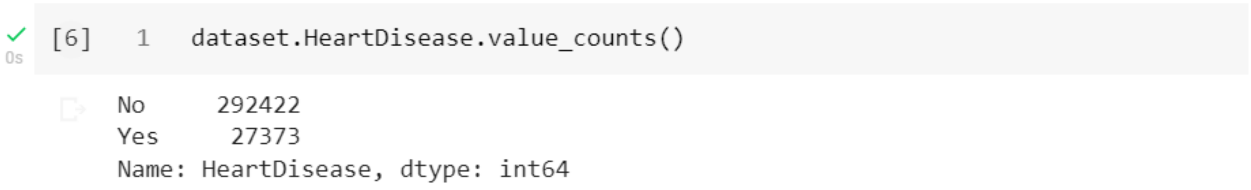


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]
dataset = pd.read_csv('https://raw.githubusercontent.com/kushal140/datasets/main/heart_2020_cleaned.csv')
```

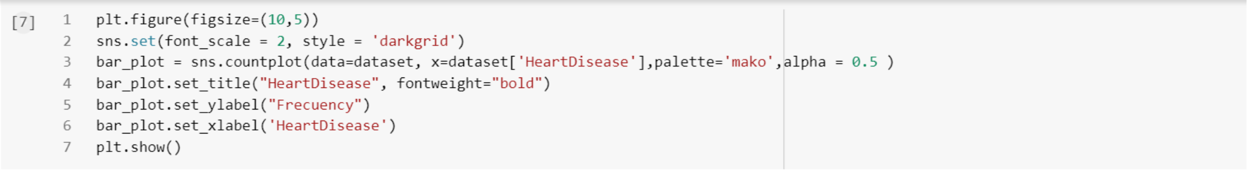
```
[3]
dataset.head()
```

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivi
0	No	16.60	Yes	No	No	3.0	30.0	No	Female	55-59	White	Yes	Y
1	No	20.34	No	No	Yes	0.0	0.0	No	Female	80 or older	White	No	Y
2	No	26.58	Yes	No	No	20.0	30.0	No	Male	65-69	White	Yes	Y
3	No	24.21	No	No	No	0.0	0.0	No	Female	75-79	White	No	
4	No	23.71	No	No	No	28.0	0.0	Yes	Female	40-44	White	No	Y



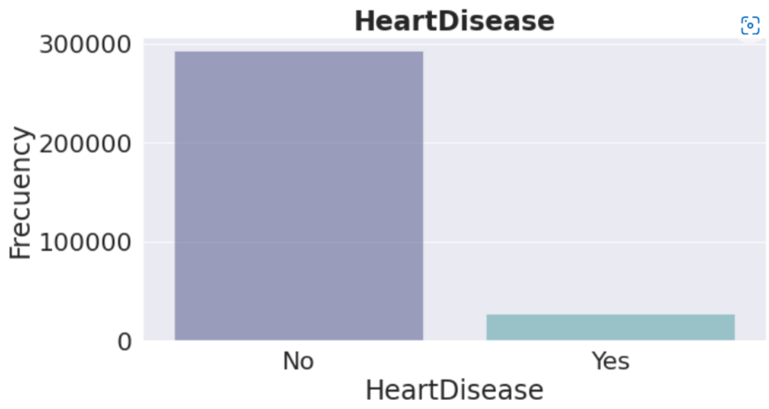
```
[6]
dataset.HeartDisease.value_counts()
```

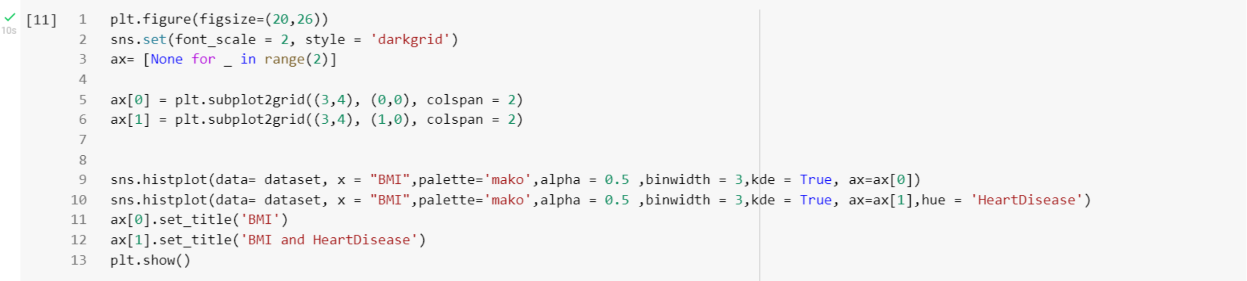
```
No     292422
Yes     27373
Name: HeartDisease, dtype: int64
```



[7]

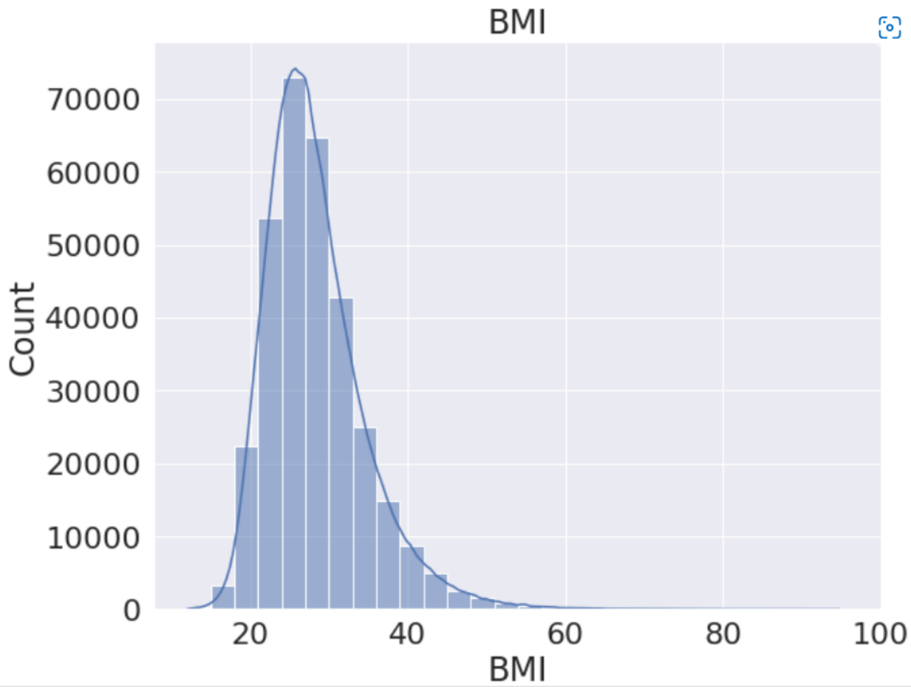
```
plt.figure(figsize=(10,5))
sns.set(font_scale = 2, style = 'darkgrid')
bar_plot = sns.countplot(data=dataset, x=dataset['HeartDisease'],palette='mako',alpha = 0.5 )
bar_plot.set_title("HeartDisease", fontweight="bold")
bar_plot.set_ylabel("Frecuency")
bar_plot.set_xlabel('HeartDisease')
plt.show()
```

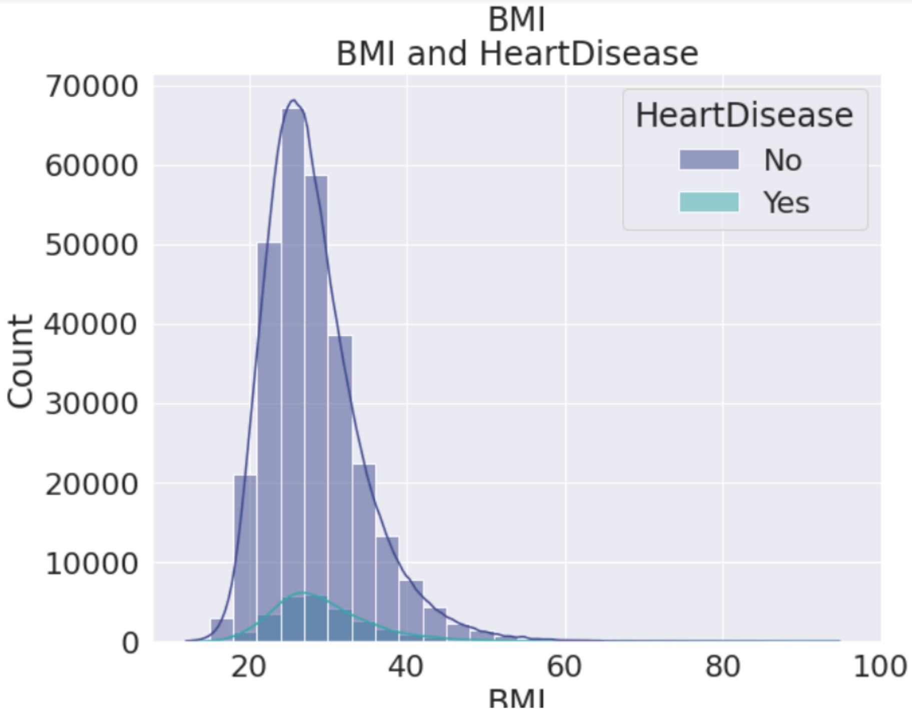


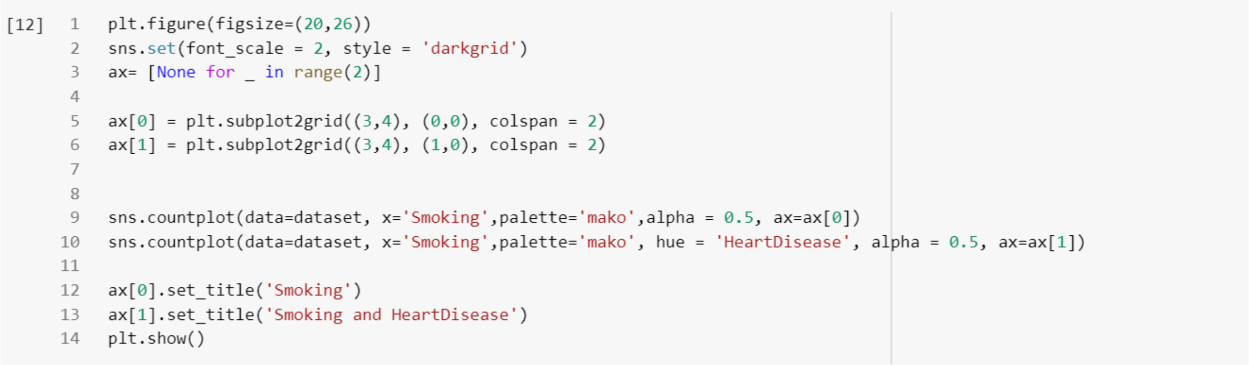


[11]
10s

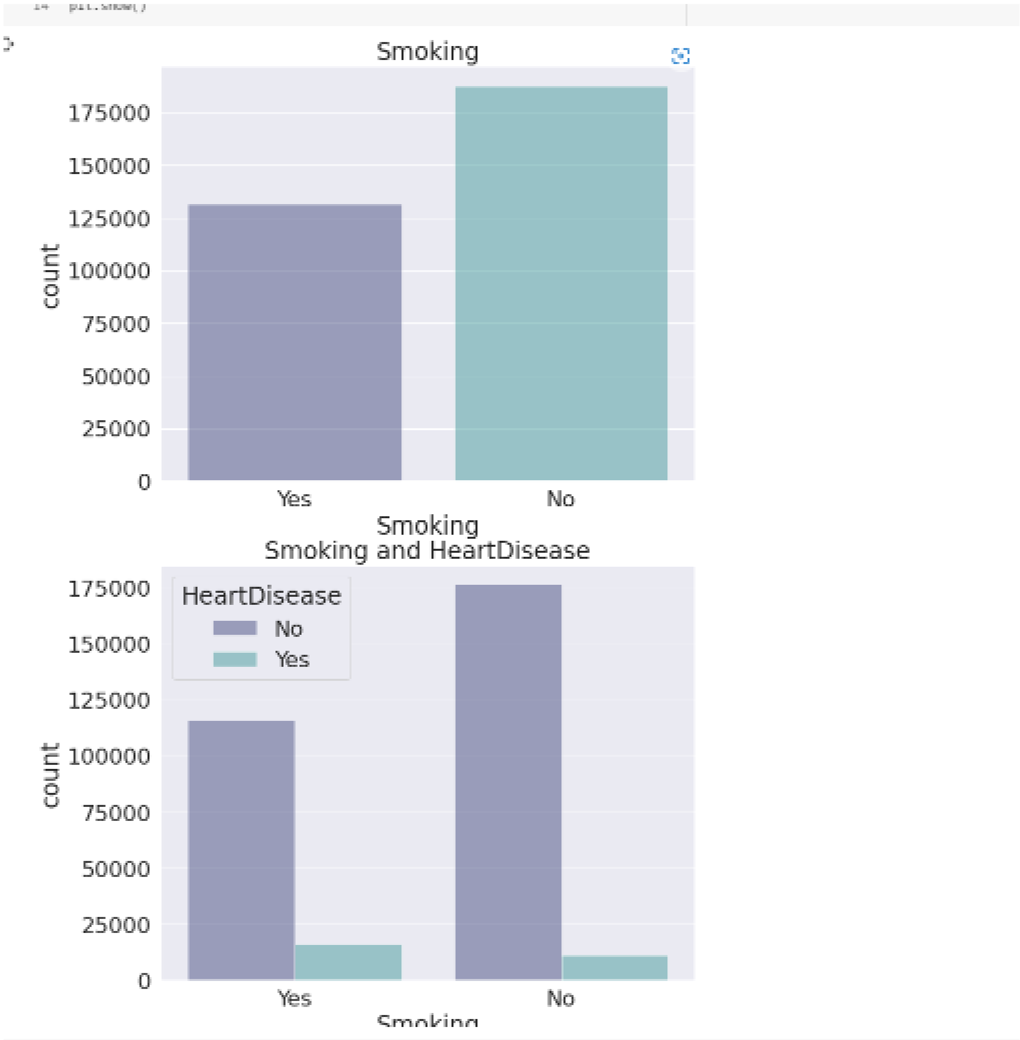
```
plt.figure(figsize=(20,26))
sns.set(font_scale = 2, style = 'darkgrid')
ax= [None for _ in range(2)]

ax[0] = plt.subplot2grid((3,4), (0,0), colspan = 2)
ax[1] = plt.subplot2grid((3,4), (1,0), colspan = 2)


sns.histplot(data= dataset, x = "BMI",palette='mako',alpha = 0.5 ,binwidth = 3,kde = True, ax=ax[0])
sns.histplot(data= dataset, x = "BMI",palette='mako',alpha = 0.5 ,binwidth = 3,kde = True, ax=ax[1],hue = 'HeartDisease')
ax[0].set_title('BMI')
ax[1].set_title('BMI and HeartDisease')
plt.show()
```



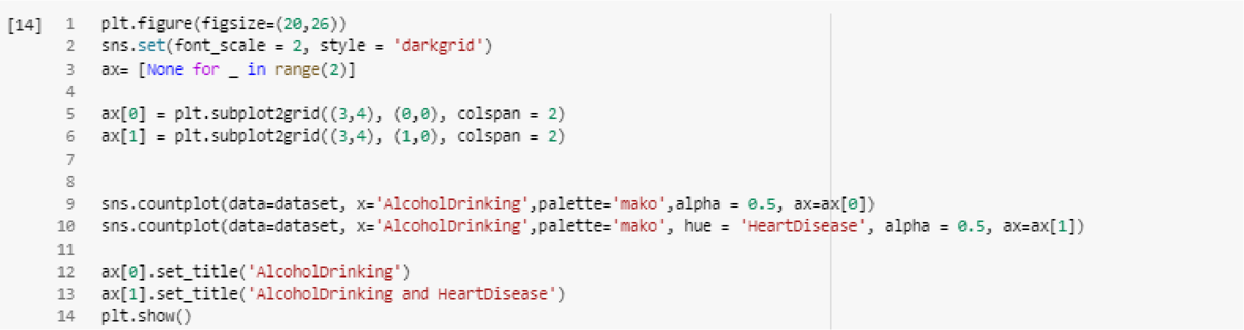



[12]

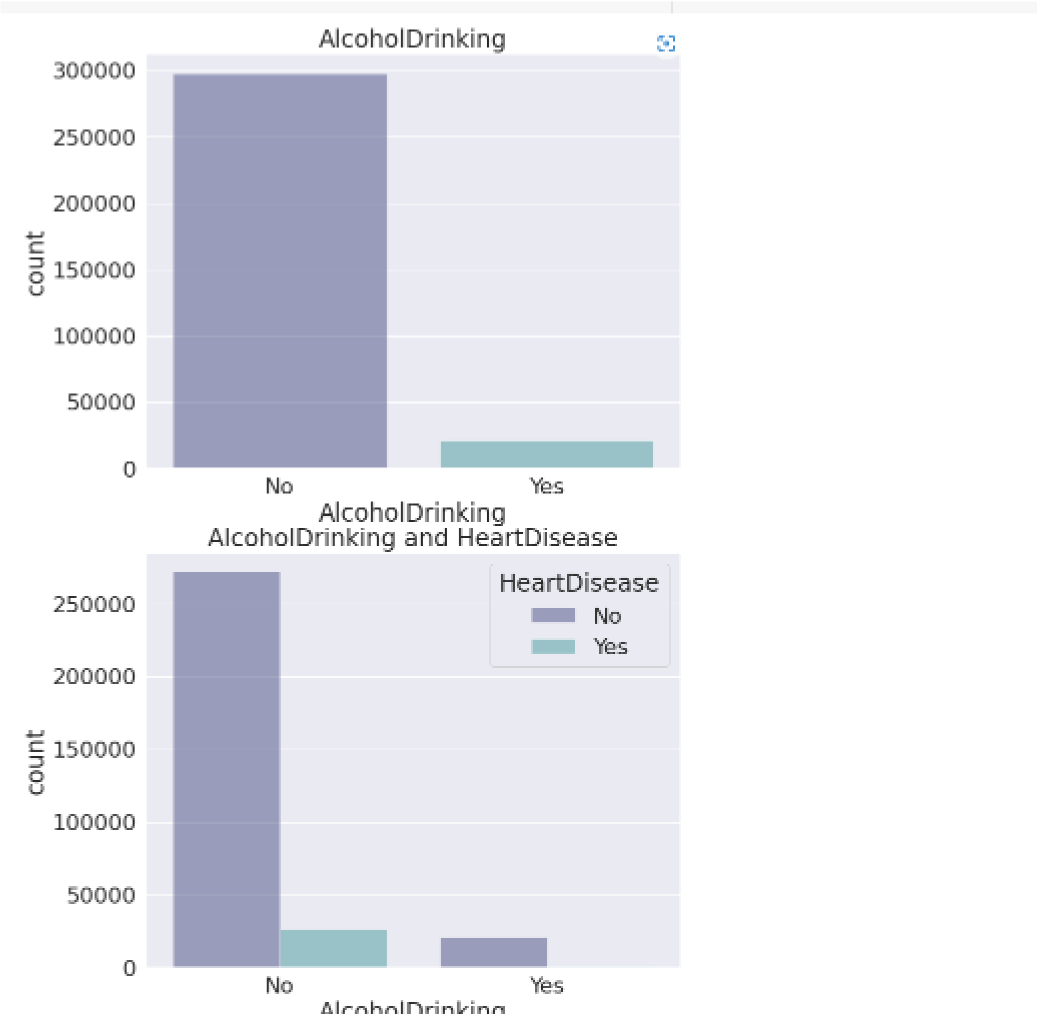
```
plt.figure(figsize=(20,26))
sns.set(font_scale = 2, style = 'darkgrid')
ax= [None for _ in range(2)]

ax[0] = plt.subplot2grid((3,4), (0,0), colspan = 2)
ax[1] = plt.subplot2grid((3,4), (1,0), colspan = 2)


sns.countplot(data=dataset, x='Smoking',palette='mako',alpha = 0.5, ax=ax[0])
sns.countplot(data=dataset, x='Smoking',palette='mako', hue = 'HeartDisease', alpha = 0.5, ax=ax[1])

ax[0].set_title('Smoking')
ax[1].set_title('Smoking and HeartDisease')
plt.show()
```

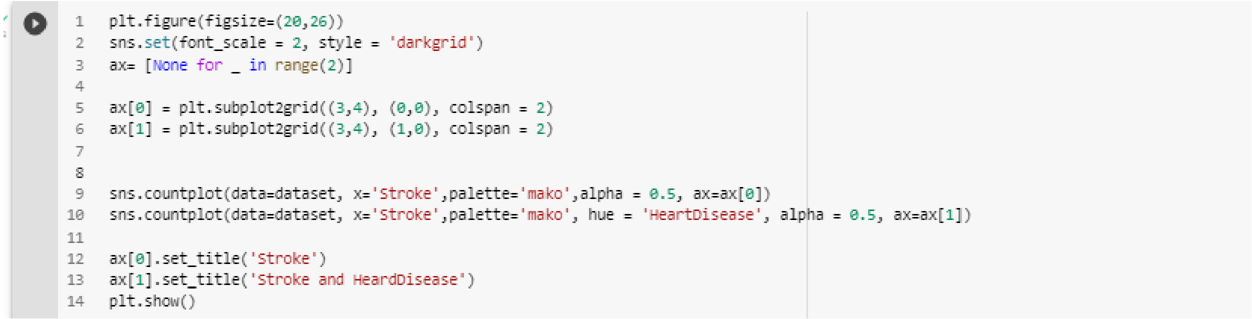


```
plt.show()
```

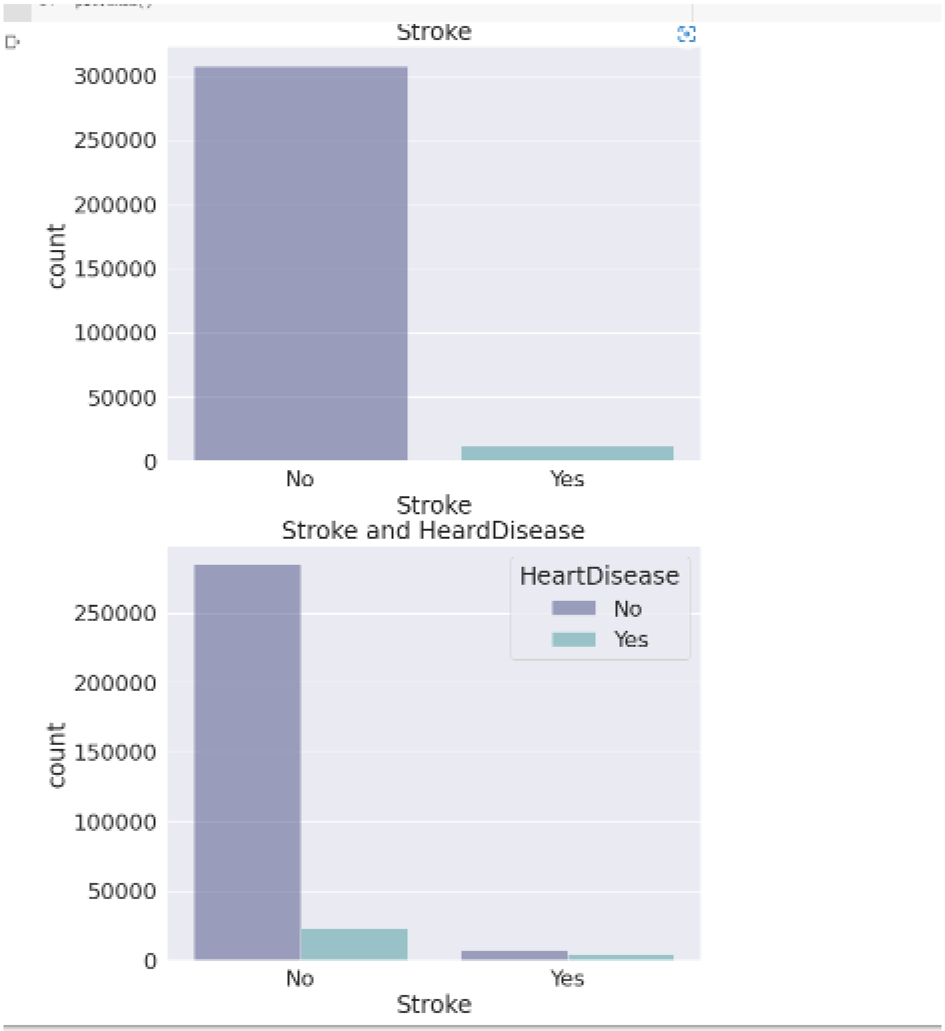


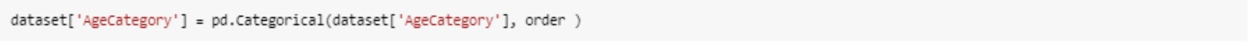
```
[14]  plt.figure(figsize=(20,26))
      sns.set(font_scale = 2, style = 'darkgrid')
      ax= [None for _ in range(2)]

      ax[0] = plt.subplot2grid((3,4), (0,0), colspan = 2)
      ax[1] = plt.subplot2grid((3,4), (1,0), colspan = 2)


      sns.countplot(data=dataset, x='AlcoholDrinking',palette='mako',alpha = 0.5, ax=ax[0])
      sns.countplot(data=dataset, x='AlcoholDrinking',palette='mako', hue = 'HeartDisease', alpha = 0.5, ax=ax[1])

      ax[0].set_title('AlcoholDrinking')
      ax[1].set_title('AlcoholDrinking and HeartDisease')
      plt.show()
```



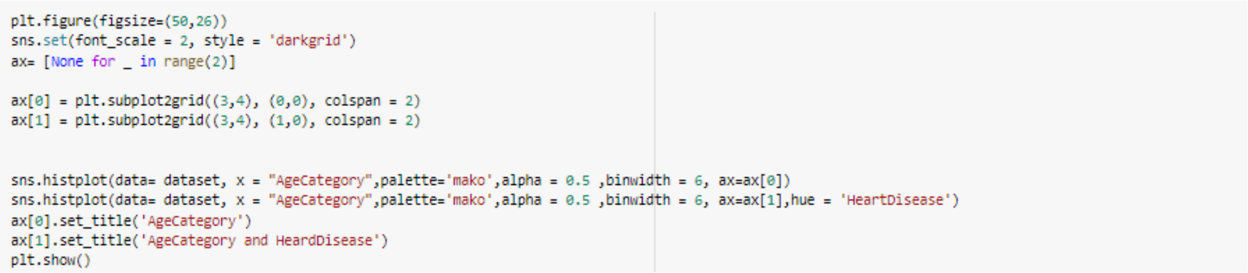


```
plt.figure(figsize=(20,26))
sns.set(font_scale = 2, style = 'darkgrid')
ax= [None for _ in range(2)]

ax[0] = plt.subplot2grid((3,4), (0,0), colspan = 2)
ax[1] = plt.subplot2grid((3,4), (1,0), colspan = 2)


sns.countplot(data=dataset, x='Stroke',palette='mako',alpha = 0.5, ax=ax[0])
sns.countplot(data=dataset, x='Stroke',palette='mako', hue = 'HeartDisease', alpha = 0.5, ax=ax[1])

ax[0].set_title('Stroke')
ax[1].set_title('Stroke and HeardDisease')
plt.show()
```

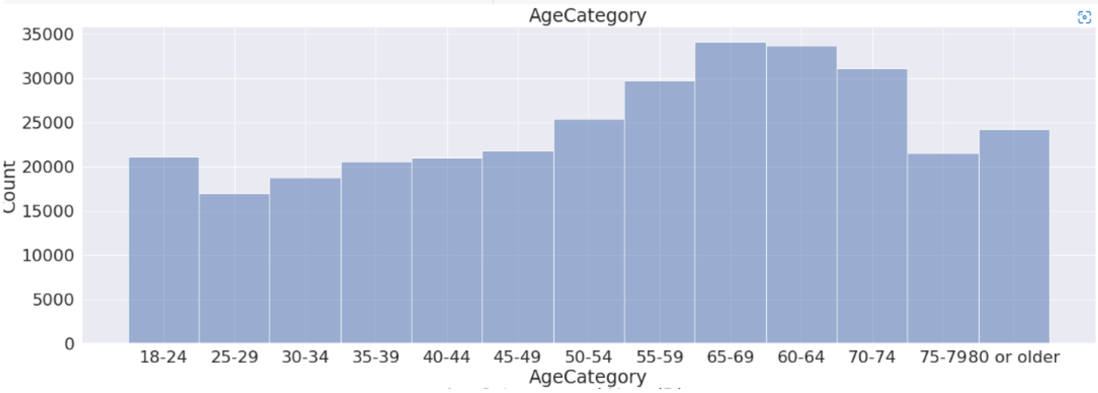


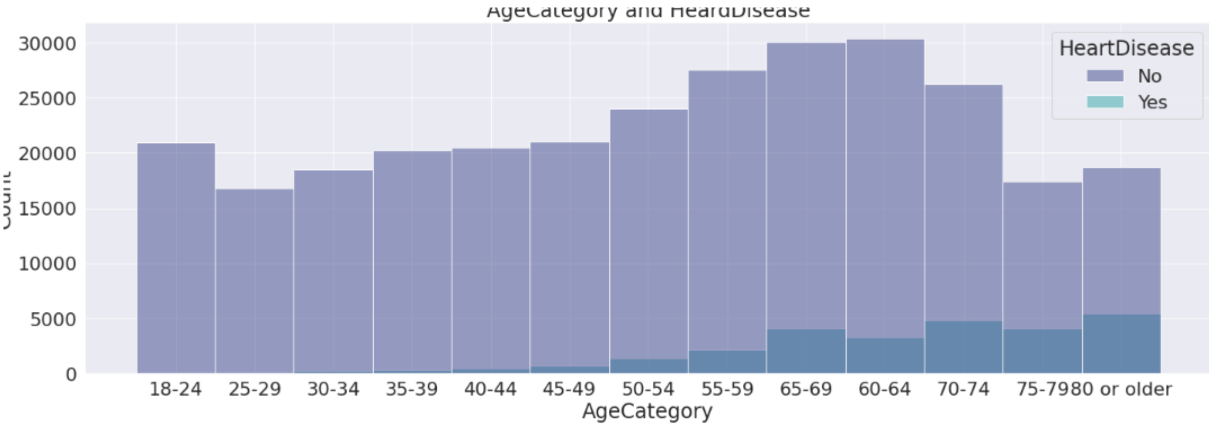


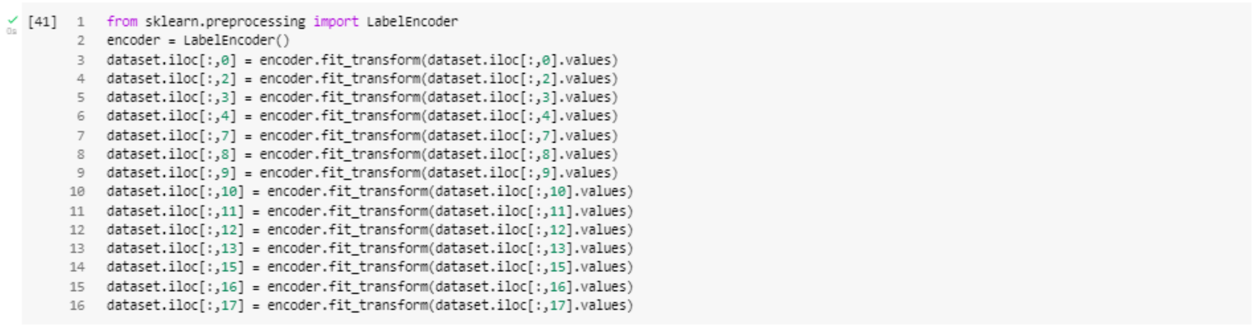
```
dataset['AgeCategory'] = pd.Categorical(dataset['AgeCategory'], order )
```



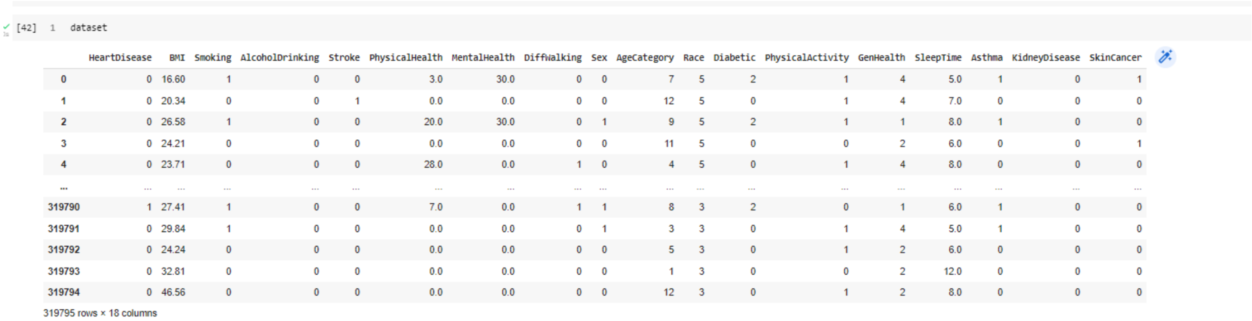
```
plt.figure(figsize=(50,26))
sns.set(font_scale = 2, style = 'darkgrid')
ax= [None for _ in range(2)]

ax[0] = plt.subplot2grid((3,4), (0,0), colspan = 2)
ax[1] = plt.subplot2grid((3,4), (1,0), colspan = 2)


sns.histplot(data= dataset, x = "AgeCategory",palette='mako',alpha = 0.5 ,binwidth = 6, ax=ax[0])
sns.histplot(data= dataset, x = "AgeCategory",palette='mako',alpha = 0.5 ,binwidth = 6, ax=ax[1],hue = 'HeartDisease')
ax[0].set_title('AgeCategory')
ax[1].set_title('AgeCategory and HeardDisease')
plt.show()
```







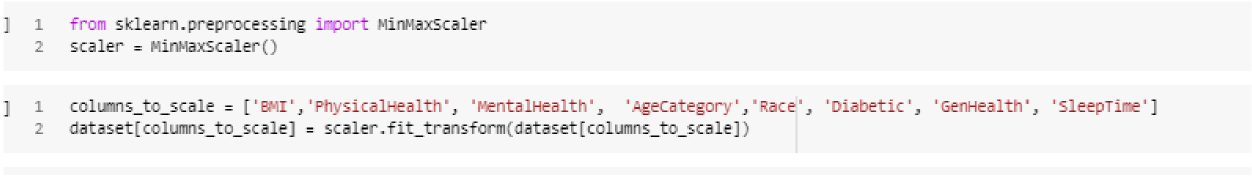
```python
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
dataset.iloc[:,0] = encoder.fit_transform(dataset.iloc[:,0].values)
dataset.iloc[:,2] = encoder.fit_transform(dataset.iloc[:,2].values)
dataset.iloc[:,3] = encoder.fit_transform(dataset.iloc[:,3].values)
dataset.iloc[:,4] = encoder.fit_transform(dataset.iloc[:,4].values)
dataset.iloc[:,7] = encoder.fit_transform(dataset.iloc[:,7].values)
dataset.iloc[:,8] = encoder.fit_transform(dataset.iloc[:,8].values)
dataset.iloc[:,9] = encoder.fit_transform(dataset.iloc[:,9].values)
dataset.iloc[:,10] = encoder.fit_transform(dataset.iloc[:,10].values)
dataset.iloc[:,11] = encoder.fit_transform(dataset.iloc[:,11].values)
dataset.iloc[:,12] = encoder.fit_transform(dataset.iloc[:,12].values)
dataset.iloc[:,13] = encoder.fit_transform(dataset.iloc[:,13].values)
dataset.iloc[:,15] = encoder.fit_transform(dataset.iloc[:,15].values)
dataset.iloc[:,16] = encoder.fit_transform(dataset.iloc[:,16].values)
dataset.iloc[:,17] = encoder.fit_transform(dataset.iloc[:,17].values)
```



```python
dataset
```

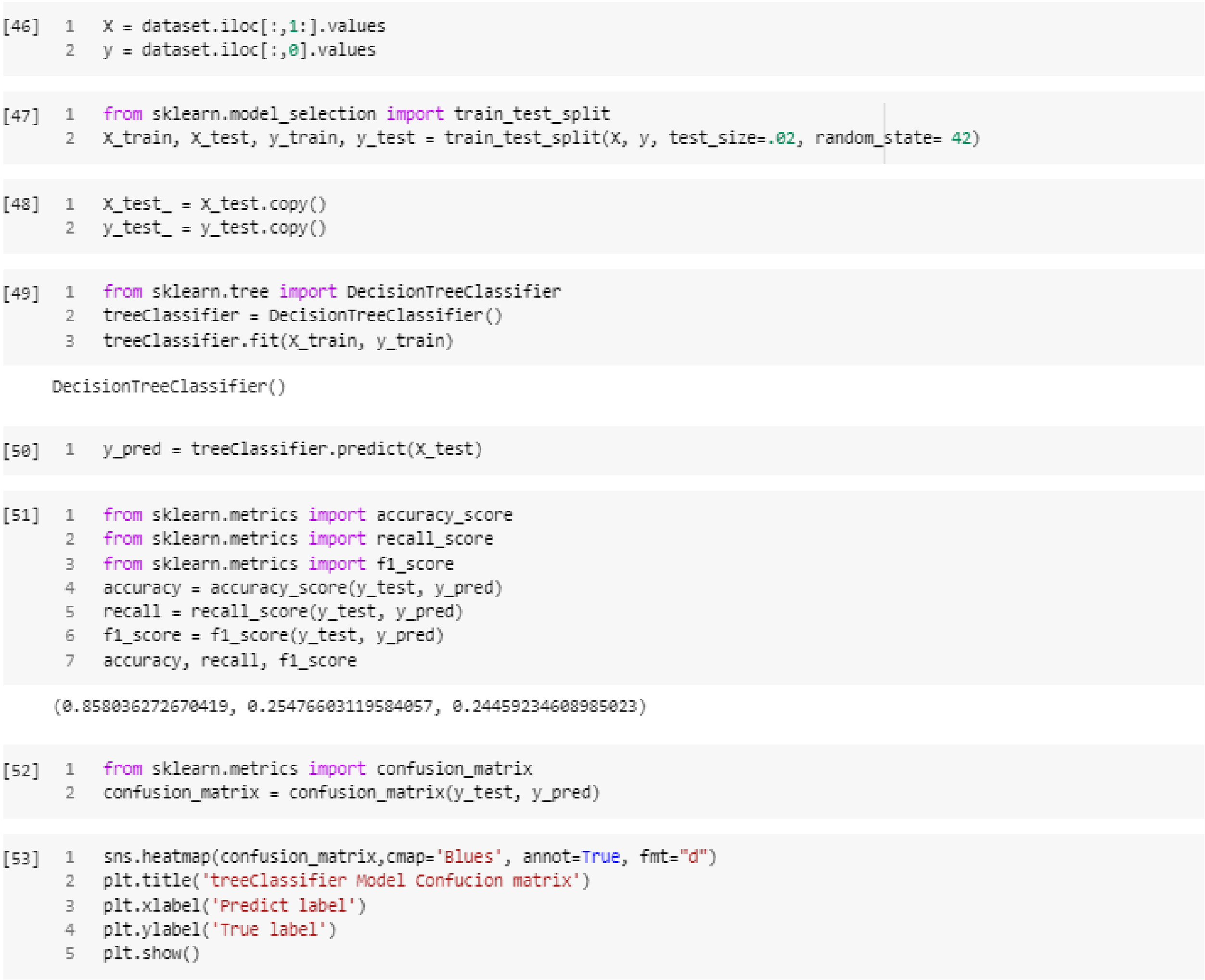
	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease	SkinCancer
0	0	16.60	1	0	0	3.0	30.0	0	0	7	5	2	1	4	5.0	1	0	1
1	0	20.34	0	0	1	0.0	0.0	0	0	12	5	0	1	4	7.0	0	0	0
2	0	26.58	1	0	0	20.0	30.0	0	1	9	5	2	1	1	8.0	1	0	0
3	0	24.21	0	0	0	0.0	0.0	0	0	11	5	0	0	2	6.0	0	0	1
4	0	23.71	0	0	0	28.0	0.0	1	0	4	5	0	1	4	8.0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
319790	1	27.41	1	0	0	7.0	0.0	1	1	8	3	2	0	1	6.0	1	0	0
319791	0	29.84	1	0	0	0.0	0.0	0	1	3	3	0	1	4	5.0	1	0	0
319792	0	24.24	0	0	0	0.0	0.0	0	0	5	3	0	1	2	6.0	0	0	0
319793	0	32.81	0	0	0	0.0	0.0	0	0	1	3	0	0	2	12.0	0	0	0
319794	0	46.56	0	0	0	0.0	0.0	0	0	12	3	0	1	2	8.0	0	0	0

319795 rows × 18 columns



```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

```
columns_to_scale = ['BMI','PhysicalHealth', 'MentalHealth',  'AgeCategory','Race', 'Diabetic', 'GenHealth', 'SleepTime']
dataset[columns_to_scale] = scaler.fit_transform(dataset[columns_to_scale])
```



[46]

```
X = dataset.iloc[:,1:].values
y = dataset.iloc[:,0].values
```

[47]

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.02, random_state= 42)
```

[48]

```
X_test_ = X_test.copy()
y_test_ = y_test.copy()
```

[49]

```
from sklearn.tree import DecisionTreeClassifier
treeClassifier = DecisionTreeClassifier()
treeClassifier.fit(X_train, y_train)
```

```
DecisionTreeClassifier()
```

[50]

```
y_pred = treeClassifier.predict(X_test)
```

[51]

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
accuracy = accuracy_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1_score = f1_score(y_test, y_pred)
accuracy, recall, f1_score
```

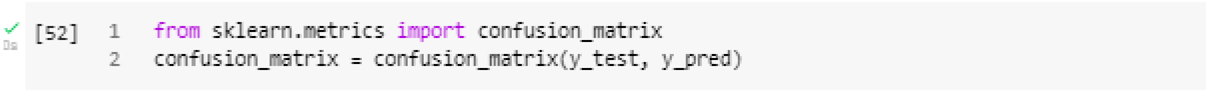
```
(0.858036272670419, 0.25476603119584057, 0.24459234608985023)
```

[52]

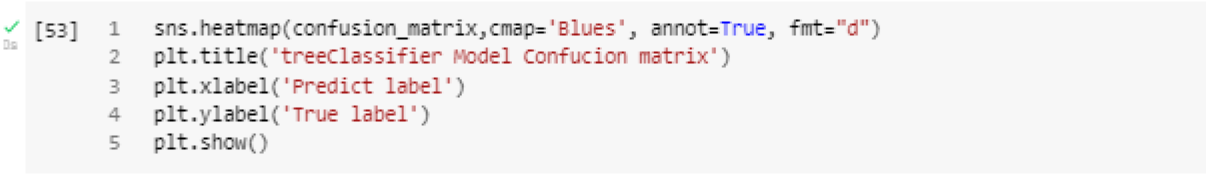
```
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
```

[53]

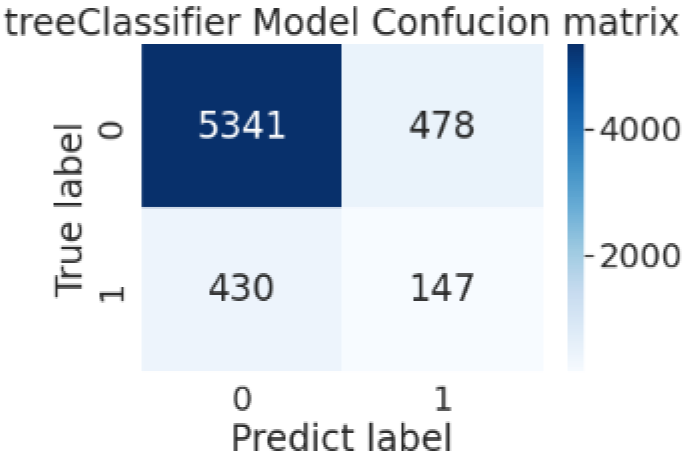
```
sns.heatmap(confusion_matrix,cmap='Blues', annot=True, fmt="d")
plt.title('treeClassifier Model Confucion matrix')
plt.xlabel('Predict label')
plt.ylabel('True label')
plt.show()
```

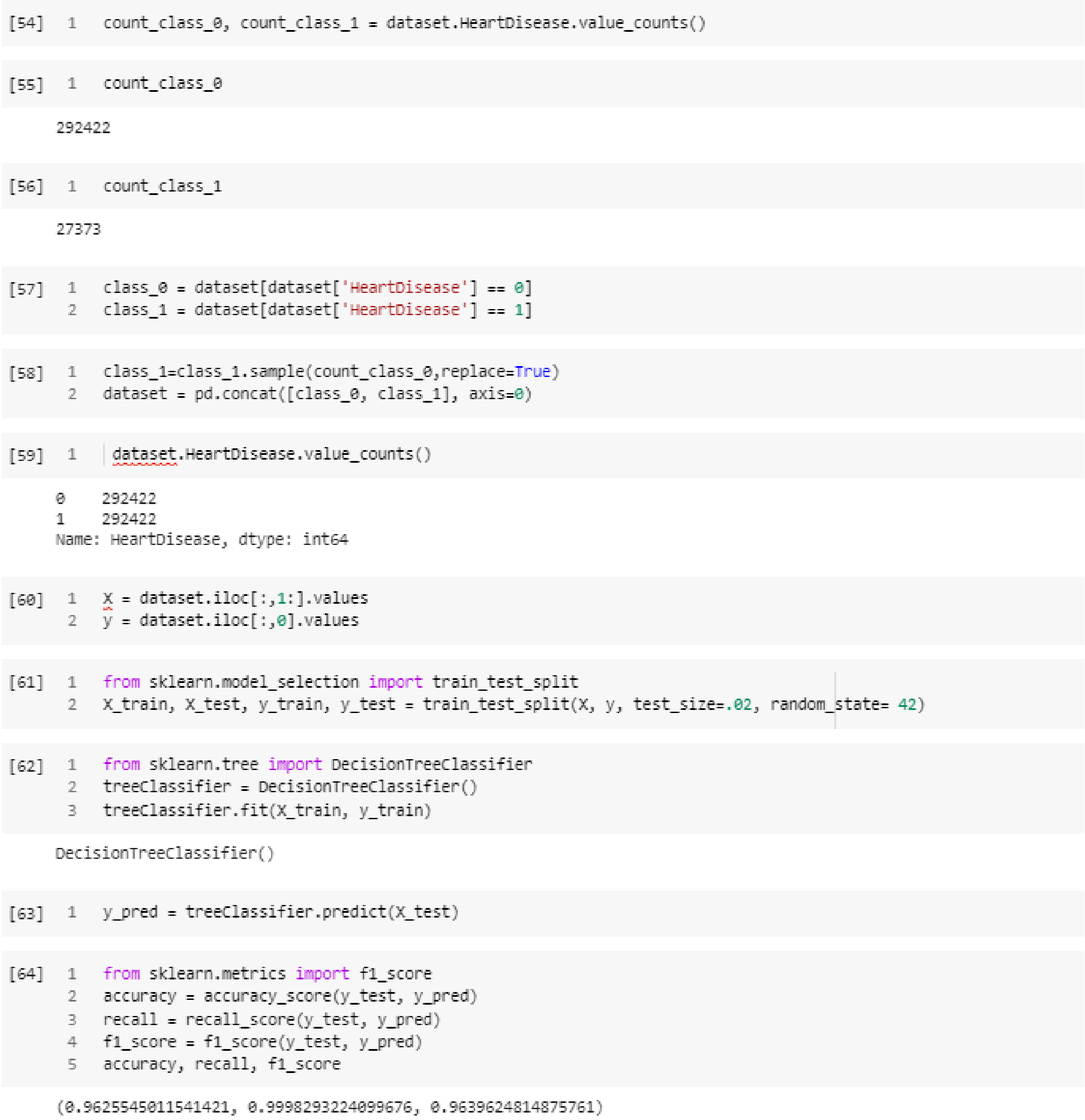


```
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
```



```
sns.heatmap(confusion_matrix,cmap='Blues', annot=True, fmt="d")
plt.title('treeClassifier Model Confucion matrix')
plt.xlabel('Predict label')
plt.ylabel('True label')
plt.show()
```





```
[54]  count_class_0, count_class_1 = dataset.HeartDisease.value_counts()
```

```
[55]  count_class_0
```

```
292422
```

```
[56]  count_class_1
```

```
27373
```

```
[57]  class_0 = dataset[dataset['HeartDisease'] == 0]
      class_1 = dataset[dataset['HeartDisease'] == 1]
```

```
[58]  class_1=class_1.sample(count_class_0,replace=True)
      dataset = pd.concat([class_0, class_1], axis=0)
```

```
[59]  dataset.HeartDisease.value_counts()
```

```
0    292422
1    292422
Name: HeartDisease, dtype: int64
```

```
[60]  X = dataset.iloc[:,1:].values
      y = dataset.iloc[:,0].values
```

```
[61]  from sklearn.model_selection import train_test_split
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.02, random_state= 42)
```

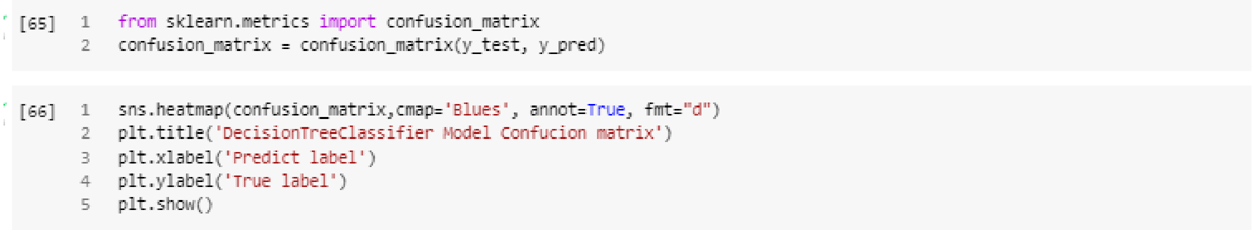
```
[62]  from sklearn.tree import DecisionTreeClassifier
      treeClassifier = DecisionTreeClassifier()
      treeClassifier.fit(X_train, y_train)
```

```
DecisionTreeClassifier()
```

```
[63]  y_pred = treeClassifier.predict(X_test)
```

```
[64]  from sklearn.metrics import f1_score
      accuracy = accuracy_score(y_test, y_pred)
      recall = recall_score(y_test, y_pred)
      f1_score = f1_score(y_test, y_pred)
      accuracy, recall, f1_score
```

```
(0.9625545011541421, 0.9998293224099676, 0.9639624814875761)
```

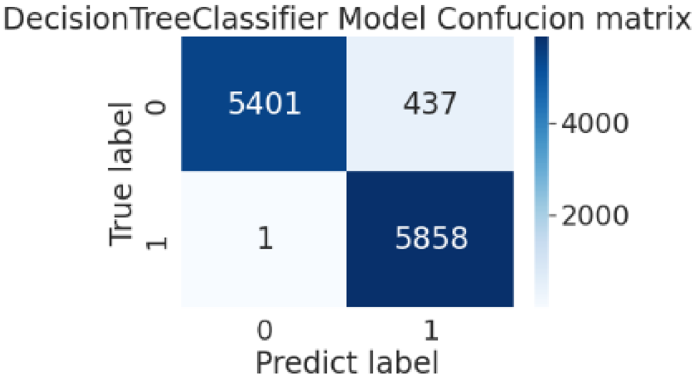


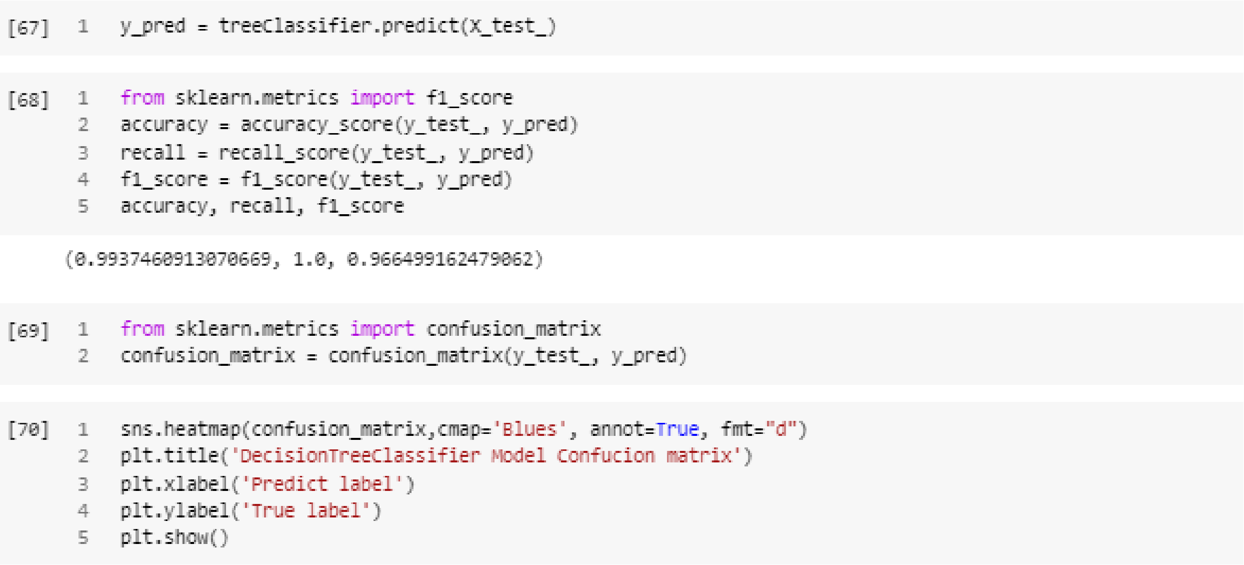
[65]

```python
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
```

[66]

```python
sns.heatmap(confusion_matrix,cmap='Blues', annot=True, fmt="d")
plt.title('DecisionTreeClassifier Model Confucion matrix')
plt.xlabel('Predict label')
plt.ylabel('True label')
plt.show()
```





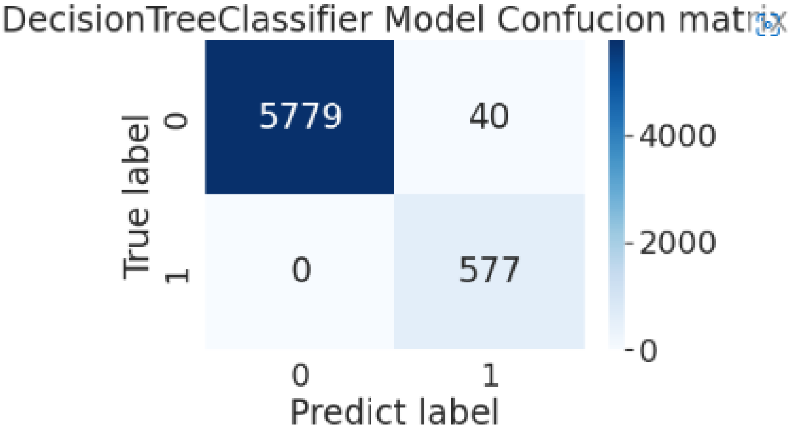
```
[67]  y_pred = treeClassifier.predict(X_test_)
```

```
[68]  from sklearn.metrics import f1_score
      accuracy = accuracy_score(y_test_, y_pred)
      recall = recall_score(y_test_, y_pred)
      f1_score = f1_score(y_test_, y_pred)
      accuracy, recall, f1_score
```

```
(0.9937460913070669, 1.0, 0.966499162479062)
```

```
[69]  from sklearn.metrics import confusion_matrix
      confusion_matrix = confusion_matrix(y_test_, y_pred)
```

```
[70]  sns.heatmap(confusion_matrix,cmap='Blues', annot=True, fmt="d")
      plt.title('DecisionTreeClassifier Model Confucion matrix')
      plt.xlabel('Predict label')
      plt.ylabel('True label')
      plt.show()
```



Advantages and Disadvantages or Applications

Advantages:-

Analysis

- It is easy to use and good for effective comparisons.
- Powerful analysis of a large amount of data can be done.

Prediction

- Easily identifies trends and patterns and gives predictions without human intervention.
- The accuracy of our prediction using this data set is 95%.
- Very useful in case of emergency.

Disadvantages:-

Analysis

- No forms.
- Less customizable Graphs.

Prediction

- The system is not fully automated, it needs data from the user for the full diagnosis.
- It’s not 100% accurate so the predictions can be wrong sometimes.

Applications:-

Analysis

- Timesheets.
- Plans and proposals.
- Reports
- Schedules
- Charts and diagrams
- Financial tools

Prediction

- Such information, if predicted well in advance, can provide important insights to doctors who can then adapt their diagnosis and treatment per-patient basis.
- One can check upon their health by entering their heart-related diagnosis report details.

Future scope

With the increasing number of deaths due to heart diseases, it has become mandatory to develop a system to predict heart diseases effectively and accurately. The motivation for the study was to find the most efficient ML algorithm for the detection of heart diseases. This study compares the accuracy score of Decision Tree, Logistic Regression, Random Forest, and Naive Bayes algorithms for predicting heart disease using a repository dataset. The result of this study indicates that the algorithm is the most efficient algorithm with an accuracy score of 95.16% for the prediction of heart disease. In the future, the work can be enhanced by developing a web application based on the Random Forest algorithm as well as using a larger dataset as compared to the one used in this analysis which will help to provide better results and help health professionals in predicting heart disease effectively and efficiently.

Overall Project Cost

(Table with a total cost estimation of the project)

Roll No	Team Members	Total no of hours	Cost per person (1hr = 1$)	Total Cost
				200$
27	Shreyas Deodhare	50	50$	
28	Kushal Gaikwad	50	50$	
29	Mitali Gangwal	50	50$	
30	Akarsha Jain	50	50$	

Conclusion/Summary

During the pandemic, we saw that people were afraid and had High Blood Pressure. Hence the Data sets related to the Heart test are increasing. We can predict from the dataset if the person is having some abnormal blood pressure or a malfunctioning heart. Before consulting a doctor we can check if we have an issue related to the heart. We can also determine the stage of damage to the heart by the construction of the graph.

References

1. https://youtu.be/lsf060bLH_Y

2. (110) Dashboard UI UX Design Best Practices & Tricks - YouTube3. (110) Most Excel users are not using full features of pivot table - YouTube

4. https://www.youtube.com/watch?v=KGnvCKiOLM0

5. https://youtu.be/QCusTU5iJJ8