HEART DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUES

NAME : MAHALAKSHMI S

UNIQUE ID : E7321012

SUPERVISOR NAME : ANAMIKA KUMARI

AGENDA

- ✓ INTRODUCTION
- ✓ OBJECTIVE
- ✓ LITERATURE REVIEW
- ✓ FUTURE WORK

INTRODUCTION

HEART DISEASE: Heart disease is a type of disease that affects the heart or blood vessels.

- ✓ Heart disease is considered as one of the major causes of death over the past decades.
- ✓ Human heart is the principal part of the human body. Basically, it regulates blood flow throughout our body.
- ✓ Several different symptoms are associated with heart disease, which makes it difficult to diagnose it quicker and better.
- ✓ This problem can be resolved by using machine learning techniques.

OBJECTIVE :

- ✓ The project aims to predict future Heart Disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning techniques.

- ✓ using machine learning models such as logistic regression, random forest and support vector machine and so on.

- ✓ Goal of this project is to develop a heart disease prediction model with improved and enhanced accuracy.

LITERATURE REVIEW

Literature review, going through on various paper and understanding the models being used in heart disease prediction using machine learning concepts.

LITERATURE REVIEW:

- ✓ Dataset splits into training and testing with the ratio of 80 : 20

MODEL ACCURACY:

EFFICIENT ACCURACY OF MODELS:

KNN - 86.885%

Random Forest – 81.967%

- ✓ **Apurv garg** , implemented KNN and Random forest machine learning algorithms in order to predict the heart diseases. The dataset obtained was the UCI dataset available at kaggle.
- ✓ After analyzing the data, correlation was found between different attributes and their effect on the target value.
- ✓ It was found that chest pain and maximum heart rate achieved had a positive correlation with the target attribute.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429

Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

LITERATURE REVIEW

- ✓ Dataset is splitting into training and testing with the ratio of 70 : 30.

MODEL ACCURACY :

EFFICIENT ACCURACY OF MODELS:

Logistic Regression – 83.83%

SVM – 83.17%

Decision Tree – 79.12%

Random Forest – 85.81%

- ✓ **Singh Yeshvendra** , proposed a various supervised machine learning algorithms such as Random Forest, SVM, Decision Tree using cross validation, Logistic and Linear Regression.
- ✓ Used Cleveland dataset which is available in kaggle.
- ✓ In the processing of data, removed the missing values and splitting the data.

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

ISSN (Online) 2581-9429; Impact Factor: 4.819

Volume 5, Issue 1, May 2021 Available at www.ijarsct.co.in

LITERATURE REVIEW :

- ✓ Dataset splitting into training and testing with the ratio of 80:20.

MODEL ACCURACY:

EFFICIENT ACCURACY OF MODELS:

Naïve Bayes	95.56%
Decision Tree	73.588%
SVM	73.588%

- ✓ S. Seema, focus on techniques that can predict the chronic disease using different models such as Naive Bayes, Decision Tree, Support Vector Machine.
- ✓ The dataset is available in UCI machine learning repository.
- ✓ After analyzing the data , data preprocessing and splitting the dataset.

International Journal of Computer Applications (0975 – 8887) Volume 181 – No. 18, September 2018.

Available at https://www.ijcaonline.org/

FUTURE WORK :

- ✓ DATASET
- ✓ METHODOLOGY
- ✓ CONCLUSION

DATASET :

- ✓ It is titled as **Heart Disease prediction using Machine Learning Techniques.** The dataset was obtained in UCI which is available at kaggle. It contains 1024 samples 14 Input Features and 1 output feature.
- ✓ The feature describe the patients age, chest pain, Resting Blood Pressure, cholesterol in mg/dl, maximum heart rate achieved and the output feature is the decision class which has value 0 – No disease, 1- affected by heart disease.
- ✓ The dataset doesn’t have any null values to be removed.

DATASET : HEART DISEASE PREDICTION USING MACHINE LEARNING TECHNIQUES

LINK : https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

DATA RESOURCE:

ATTRIBUTE	DESCRIPTION	RANGE
Age	Age of Person in years	29 - 79
Sex	Gender of Person(1-M, 0 – F)	0, 1
Cp	Chest Pain Type	1, 2, 3, 4
Trestbps	Resting Blood Pressure in mm Hg	94 - 200
Chol	Serum cholesterol in mg/dl	126 - 564
Fbs	Fasting Blood Sugar in mg/dl	0, 1
Restecg	Resting Electrocardiographic results	0, 1, 2
Thalach	Maximum Heart Rate achieved	71 - 202
Exang	Exercise Induced Angina	0, 1
Oldpeak	ST depression induced by exercise relative to rest	1 - 3
Slope	Slope of the Peak Exercise ST segment	1, 2, 3
Ca	Number of major vessels colored by fluoroscopy	0 - 3
Thal	0 – Normal, 1 – Fixed detect, 2 – Reversible defect	0, 1, 2
Result	Class attribute	0, 1

PROPOSED MODEL:

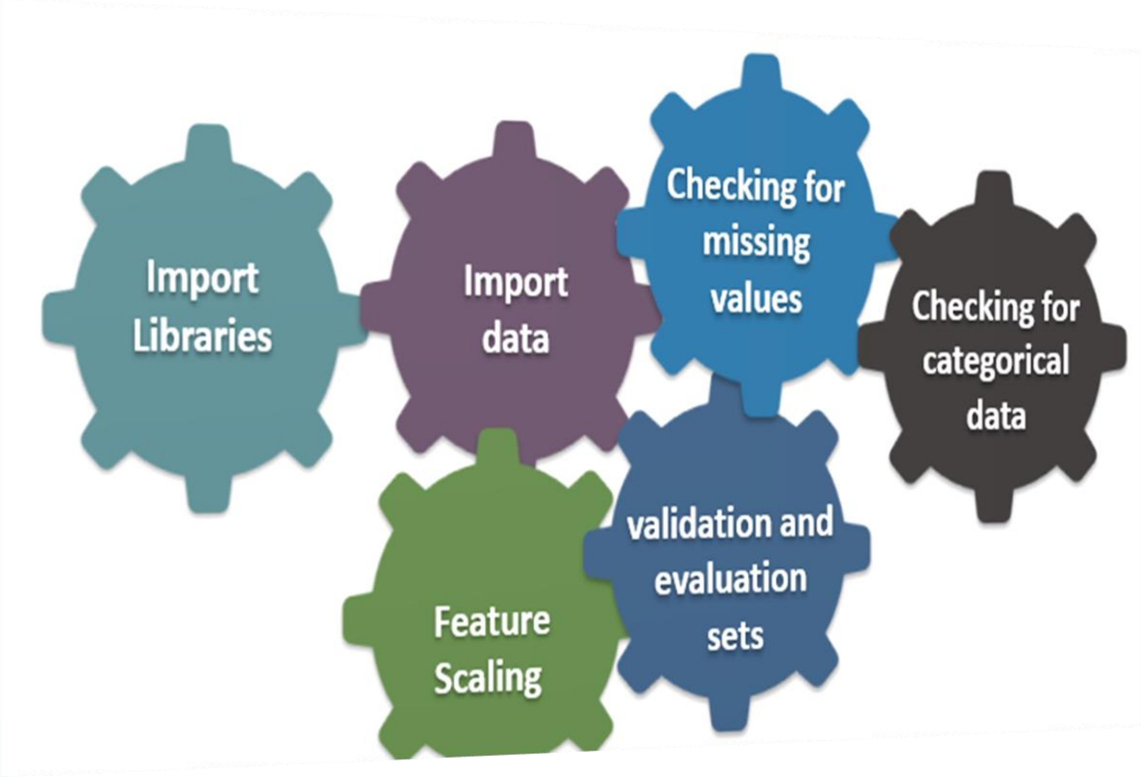
.

- ✓ The user inputs its specific medical details to get the prediction of heart disease for that user. The algorithm will calculate the probability of presence of heart disease.

MACHINE LEARNING ALGORITHMS:

- ✓ Logistic Regression
- ✓ Support Vector Machine
- ✓ K – Nearest Neighbor
- ✓ Decision Tree
- ✓ Random Forest

METHODOLOGY:


Import Libraries
Import data
Checking for missing values
Checking for categorical data
Feature Scaling
validation and evaluation sets

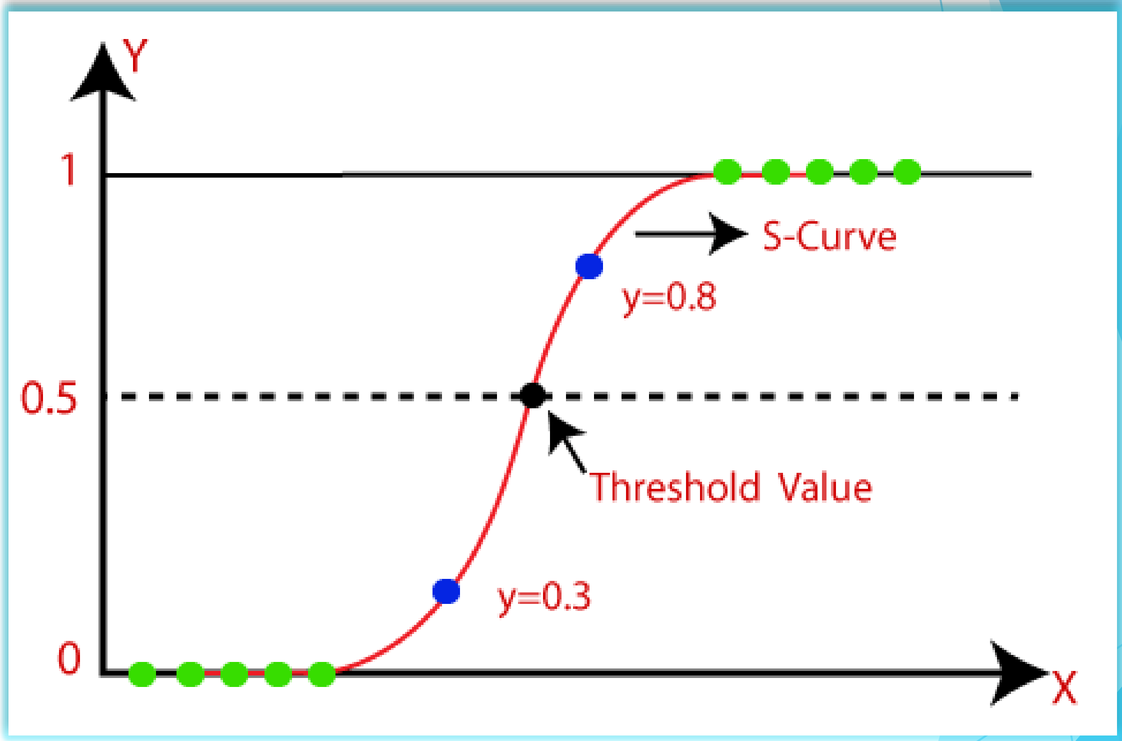
- **DATA PREPROCESSING:**

✓ Data Cleaning: NA values in the dataset were the major setback as it was reducing the accuracy of the predict. So, if there is any missing values present in the dataset it can be removed.

✓ Feature Scaling: Since the range of raw data varies widely in some machine learning algorithms, objective function will not work properly without feature scaling.

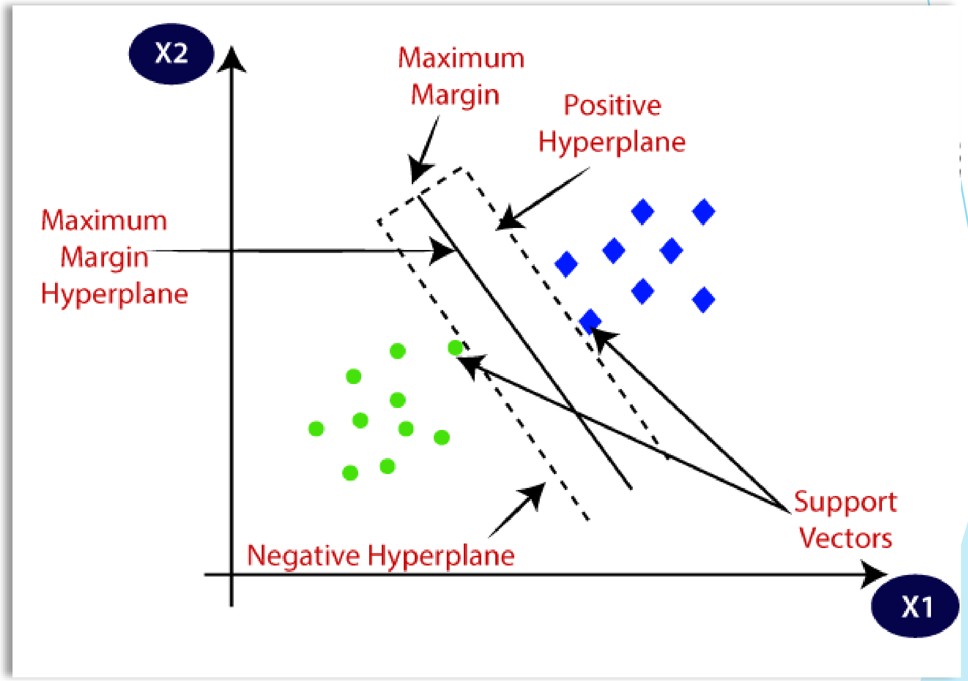
LOGISTIC REGRESSION:

- ✓ Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable.
- ✓ The nature of target or dependent variable is dichotomous, which means there would be only two possible classes 0 for failure and 1 for success



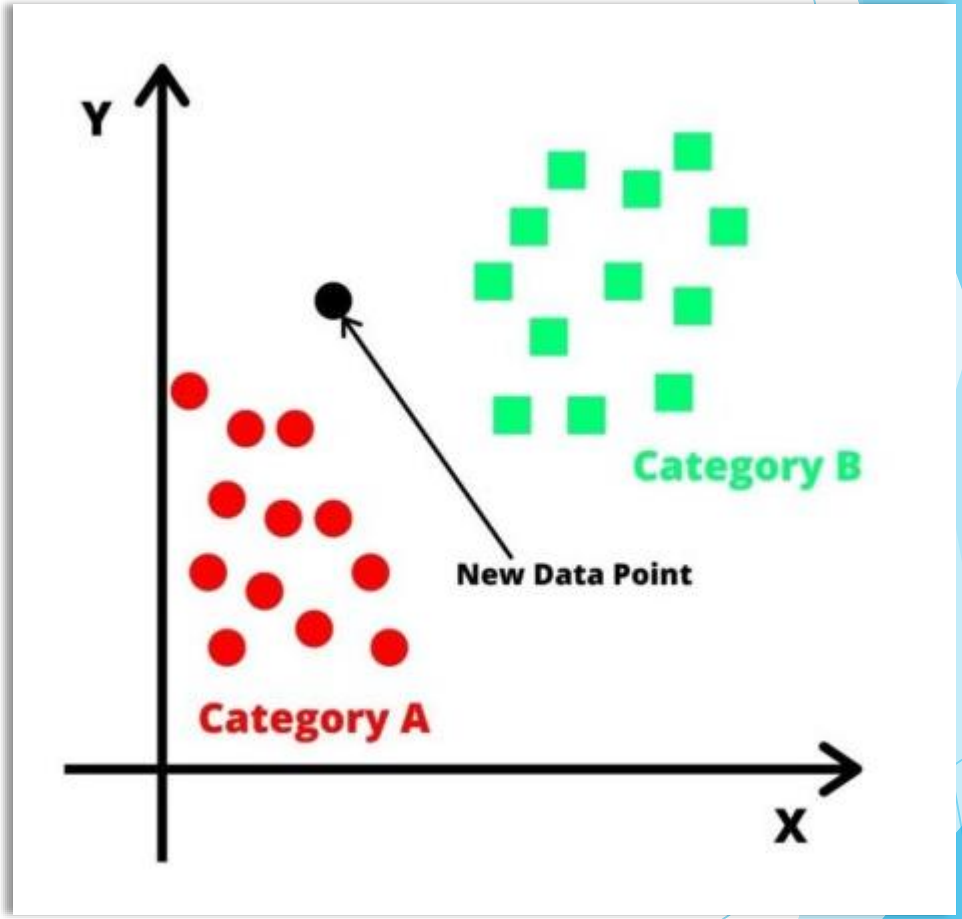
SUPPORT VECTOR MACHINE:

- ✓ The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.



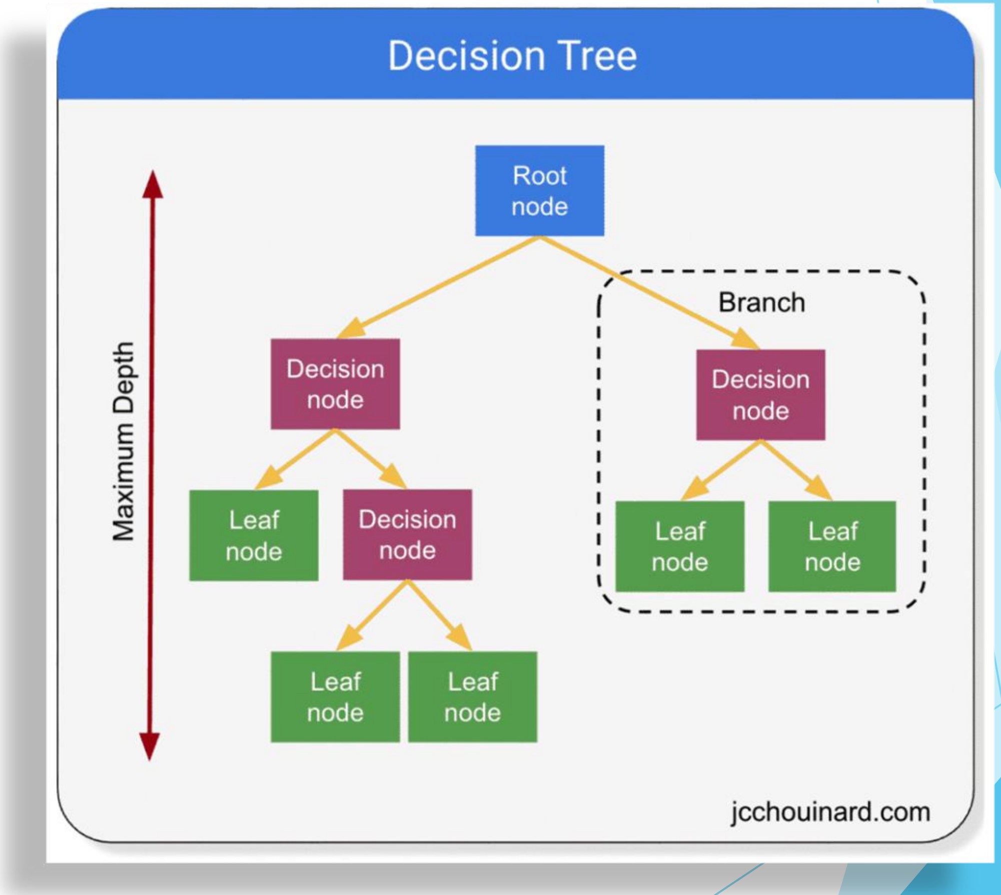
K – NEAREST NEIGHBOR:

- ✓ K-nearest neighbors (KNN) algorithm uses 'feature similarity' to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.

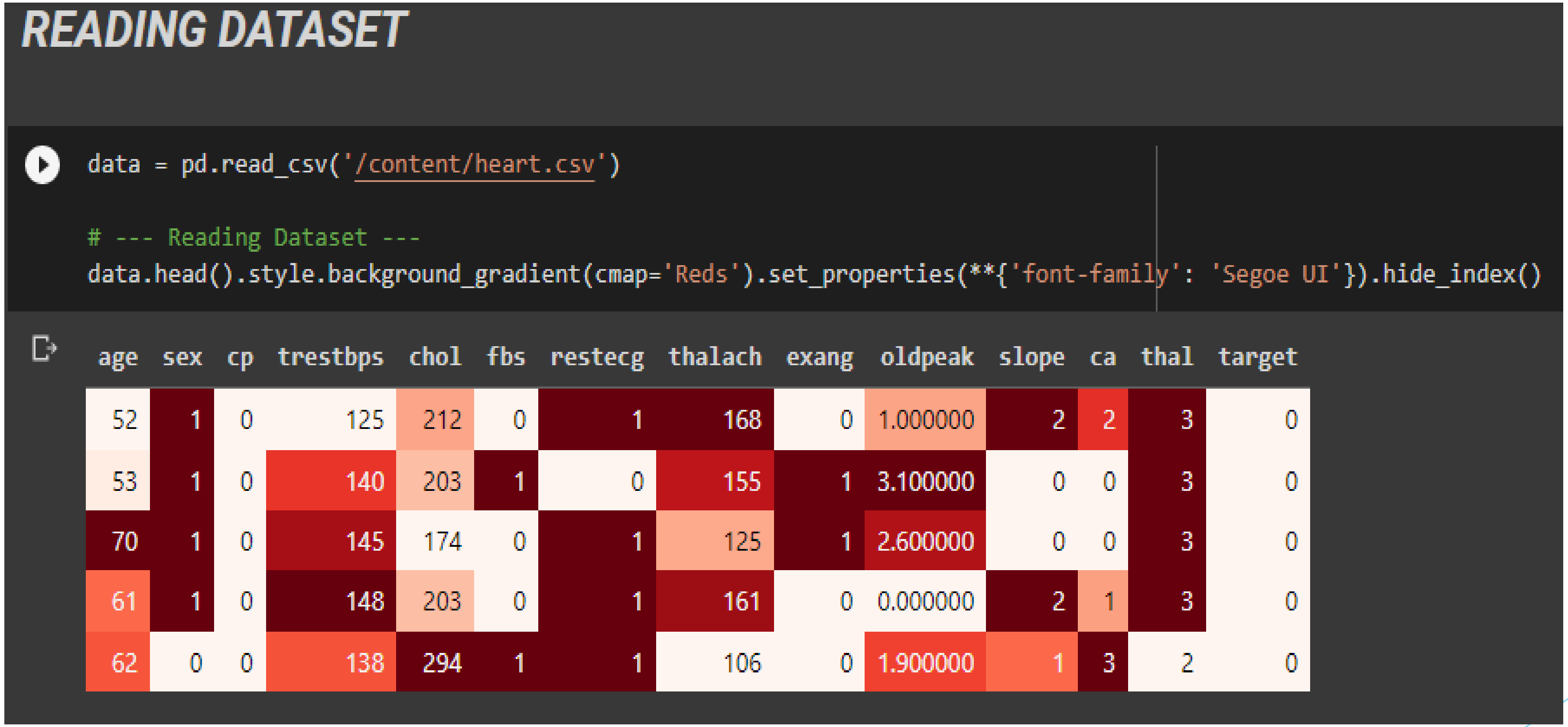


DECISION TREE:

- ✓ A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.



HEART DISEASE DATASET SOURCE CODE:

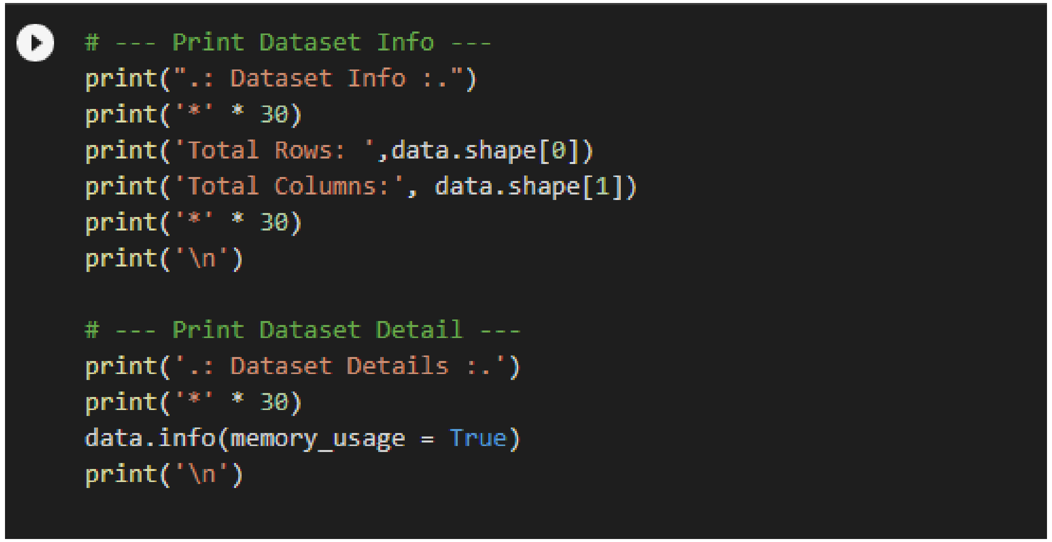


READING DATASET

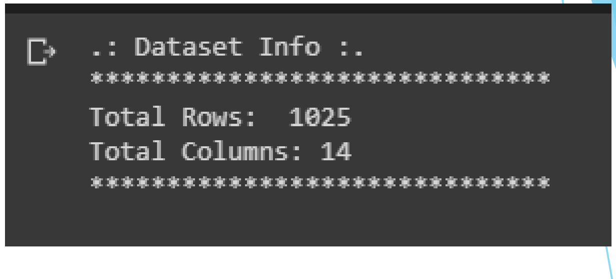
```
data = pd.read_csv('/content/heart.csv')

# --- Reading Dataset ---
data.head().style.background_gradient(cmap='Reds').set_properties(**{'font-family': 'Segoe UI'}).hide_index()
```

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
52	1	0	125	212	0	1	168	0	1.000000	2	2	3	0
53	1	0	140	203	1	0	155	1	3.100000	0	0	3	0
70	1	0	145	174	0	1	125	1	2.600000	0	0	3	0
61	1	0	148	203	0	1	161	0	0.000000	2	1	3	0
62	0	0	138	294	1	1	106	0	1.900000	1	3	2	0

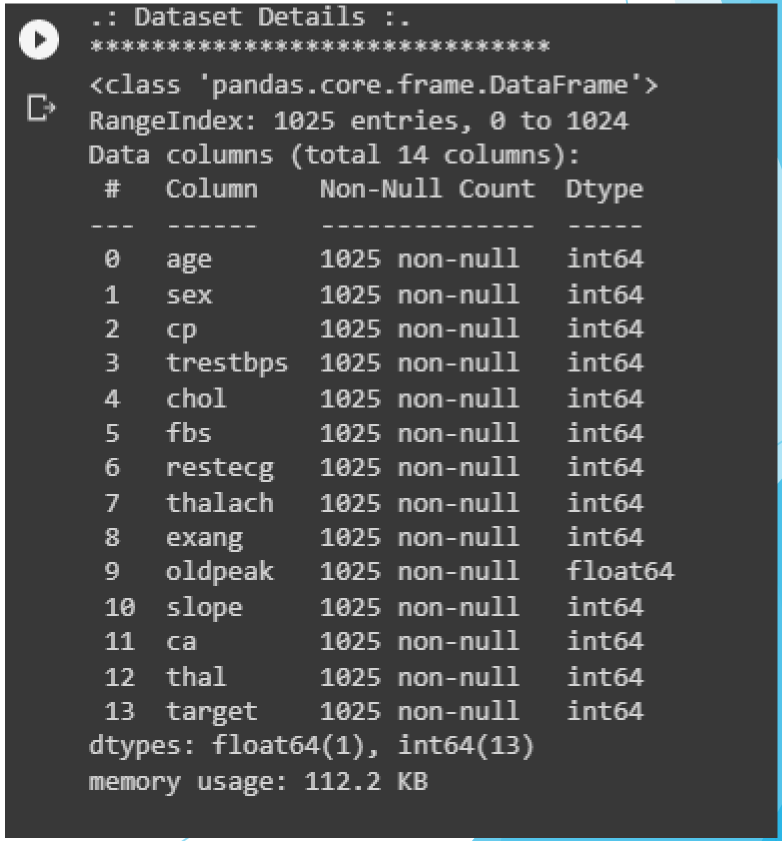
DATASET INFO:



```
# --- Print Dataset Info ---
print(".: Dataset Info :.")
print('*' * 30)
print('Total Rows: ',data.shape[0])
print('Total Columns:', data.shape[1])
print('*' * 30)
print('\n')

# --- Print Dataset Detail ---
print('.: Dataset Details :.')
print('*' * 30)
data.info(memory_usage = True)
print('\n')
```

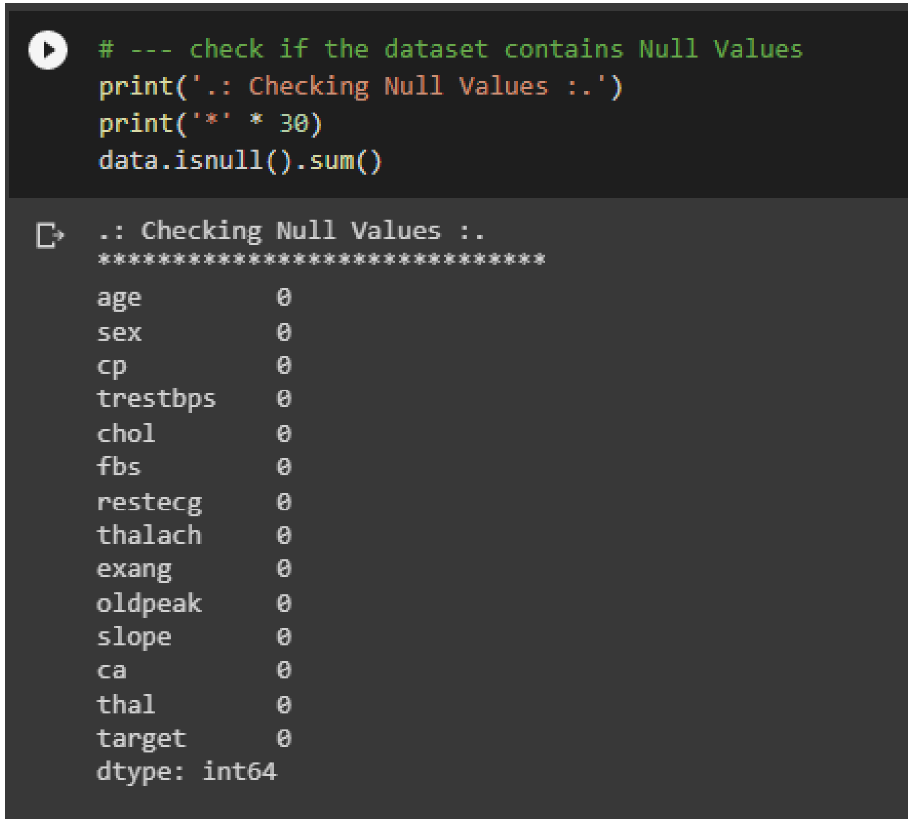


```
.: Dataset Info :.
******************************
Total Rows:  1025
Total Columns: 14
******************************
```



```
.: Dataset Details :.
******************************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

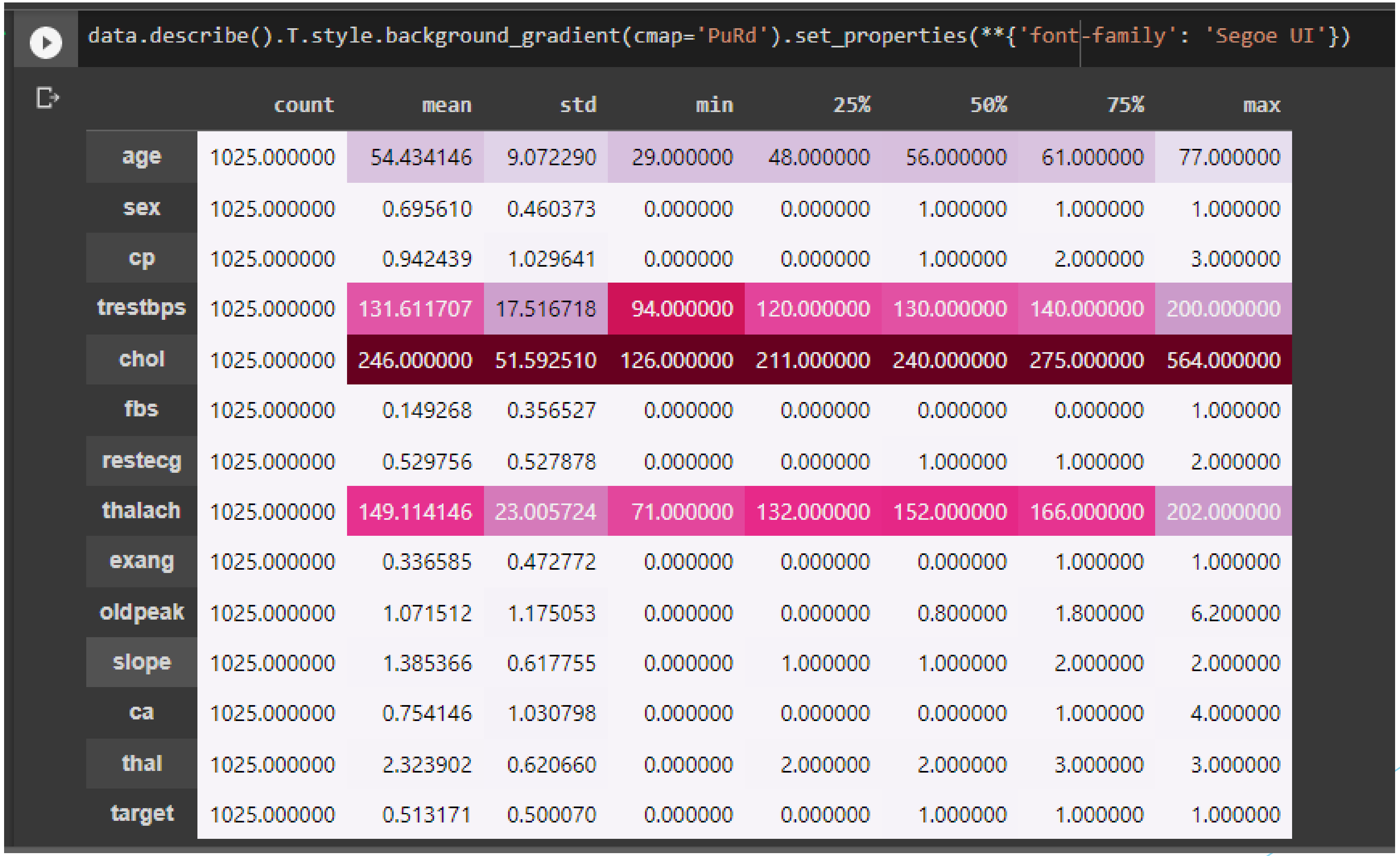
CHECKING NULL VALUES:



```
# --- check if the dataset contains Null Values
print('.: Checking Null Values :.')
print('*' * 30)
data.isnull().sum()
```

```
.: Checking Null Values :.
******************************
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

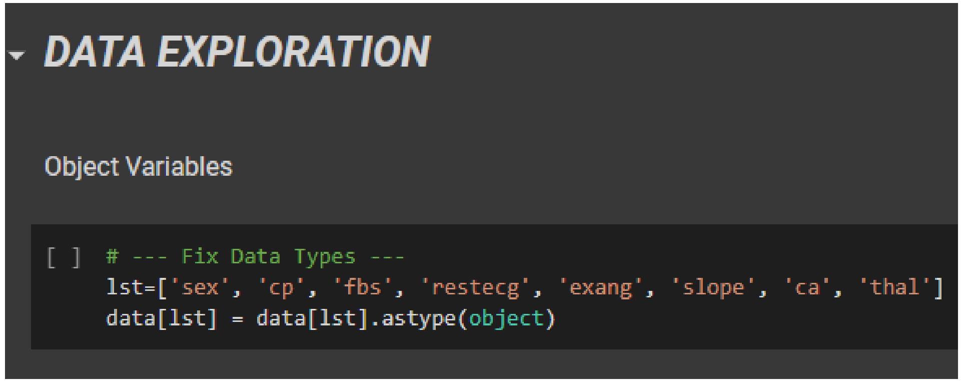
STATISTICAL INFORMATION OF DATA:



```
data.describe().T.style.background_gradient(cmap='PuRd').set_properties(**{'font-family': 'Segoe UI'})
```

	count	mean	std	min	25%	50%	75%	max
age	1025.000000	54.434146	9.072290	29.000000	48.000000	56.000000	61.000000	77.000000
sex	1025.000000	0.695610	0.460373	0.000000	0.000000	1.000000	1.000000	1.000000
cp	1025.000000	0.942439	1.029641	0.000000	0.000000	1.000000	2.000000	3.000000
trestbps	1025.000000	131.611707	17.516718	94.000000	120.000000	130.000000	140.000000	200.000000
chol	1025.000000	246.000000	51.592510	126.000000	211.000000	240.000000	275.000000	564.000000
fbs	1025.000000	0.149268	0.356527	0.000000	0.000000	0.000000	0.000000	1.000000
restecg	1025.000000	0.529756	0.527878	0.000000	0.000000	1.000000	1.000000	2.000000
thalach	1025.000000	149.114146	23.005724	71.000000	132.000000	152.000000	166.000000	202.000000
exang	1025.000000	0.336585	0.472772	0.000000	0.000000	0.000000	1.000000	1.000000
oldpeak	1025.000000	1.071512	1.175053	0.000000	0.000000	0.800000	1.800000	6.200000
slope	1025.000000	1.385366	0.617755	0.000000	1.000000	1.000000	2.000000	2.000000
ca	1025.000000	0.754146	1.030798	0.000000	0.000000	0.000000	1.000000	4.000000
thal	1025.000000	2.323902	0.620660	0.000000	2.000000	2.000000	3.000000	3.000000
target	1025.000000	0.513171	0.500070	0.000000	0.000000	1.000000	1.000000	1.000000

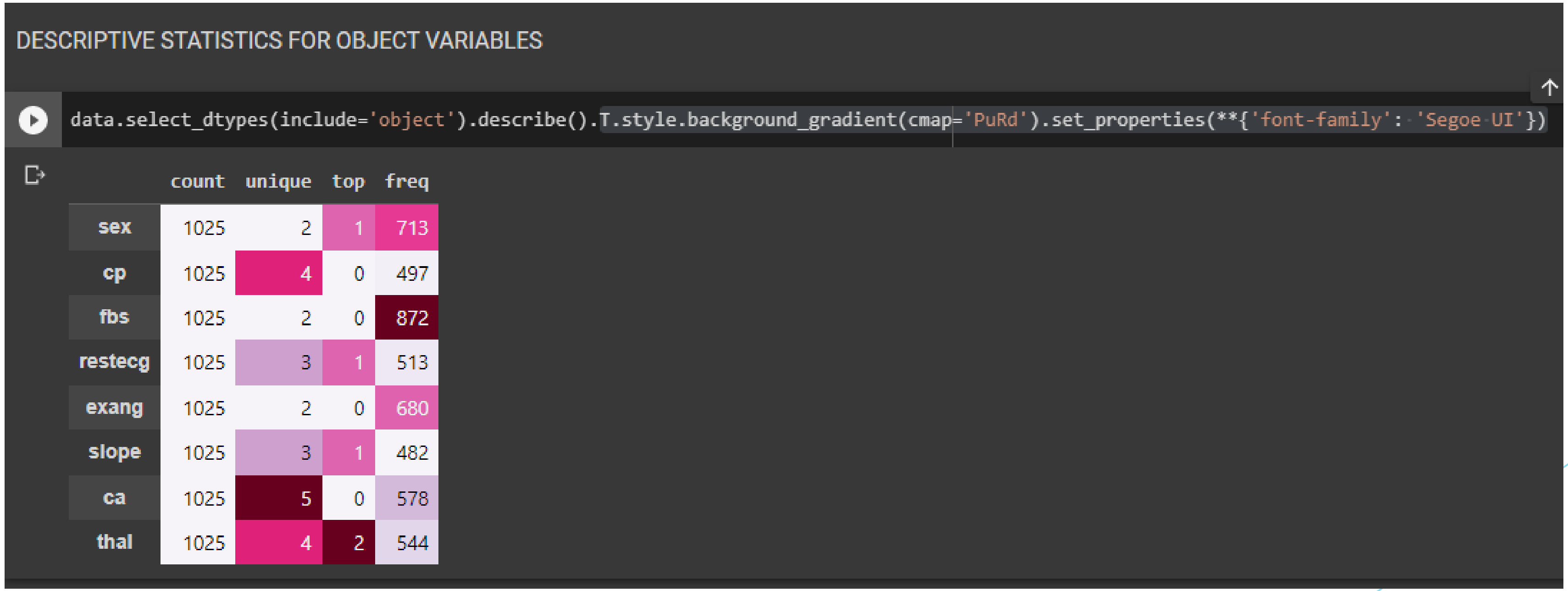
DATA EXPLORATION:



DATA EXPLORATION

Object Variables

```python
# --- Fix Data Types ---
lst=['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal']
data[lst] = data[lst].astype(object)
```

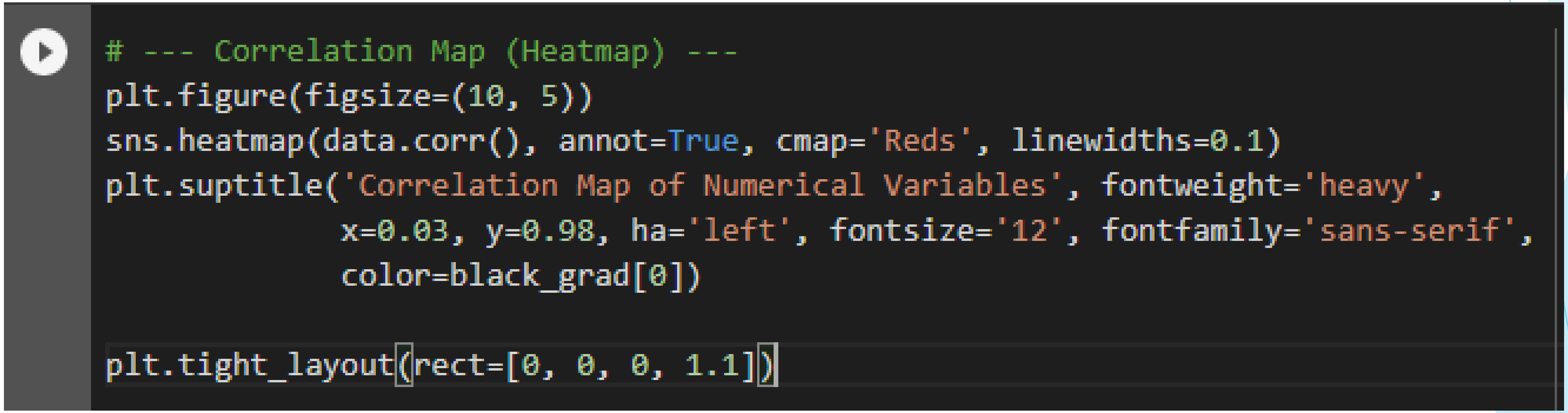


DESCRIPTIVE STATISTICS FOR OBJECT VARIABLES

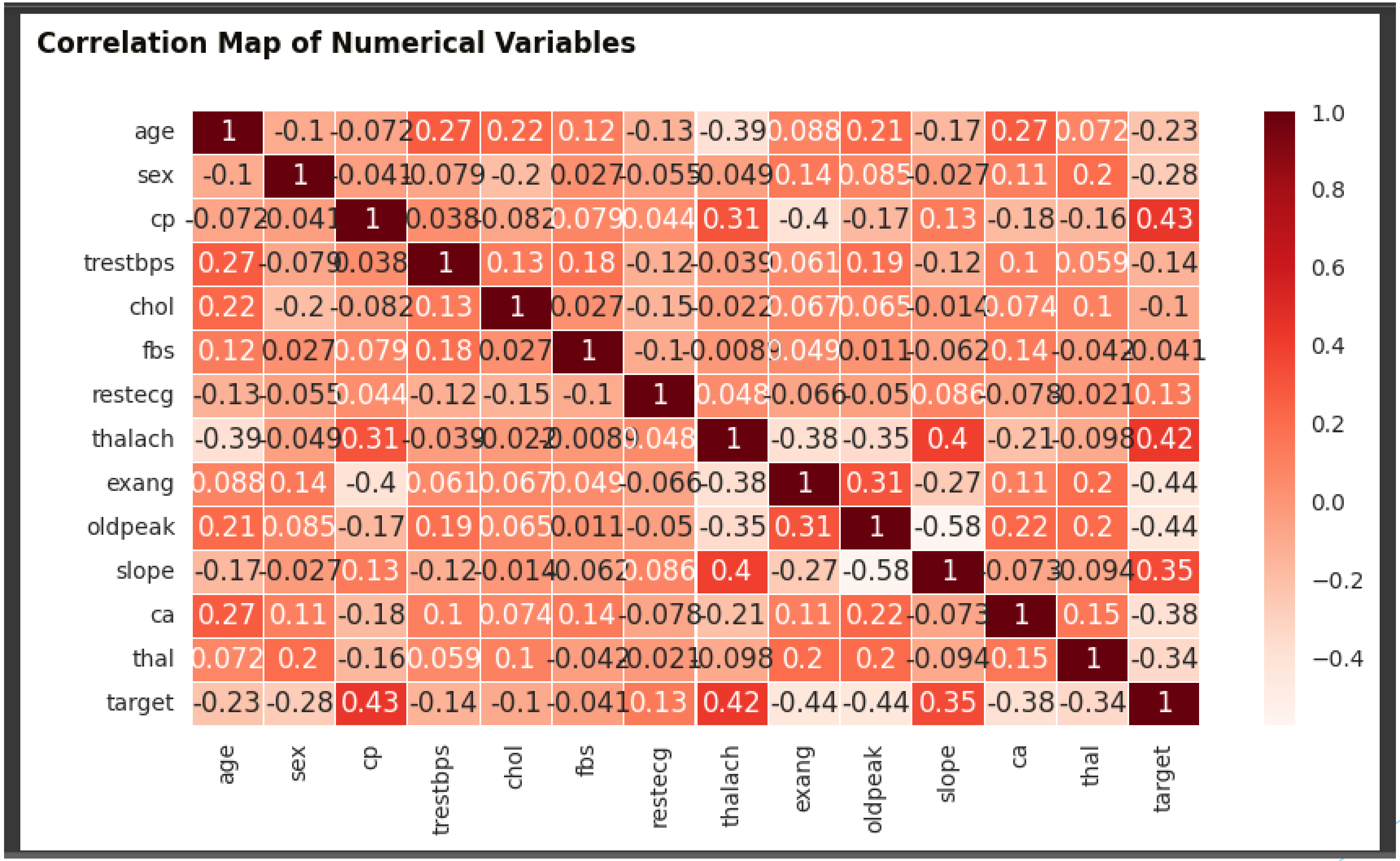
```python
data.select_dtypes(include='object').describe().T.style.background_gradient(cmap='PuRd').set_properties(**{'font-family': 'Segoe UI'})
```

	count	unique	top	freq
sex	1025	2	1	713
cp	1025	4	0	497
fbs	1025	2	0	872
restecg	1025	3	1	513
exang	1025	2	0	680
slope	1025	3	1	482
ca	1025	5	0	578
thal	1025	4	2	544

CORRELATION BETWEEN VARIOUS FEATURES:



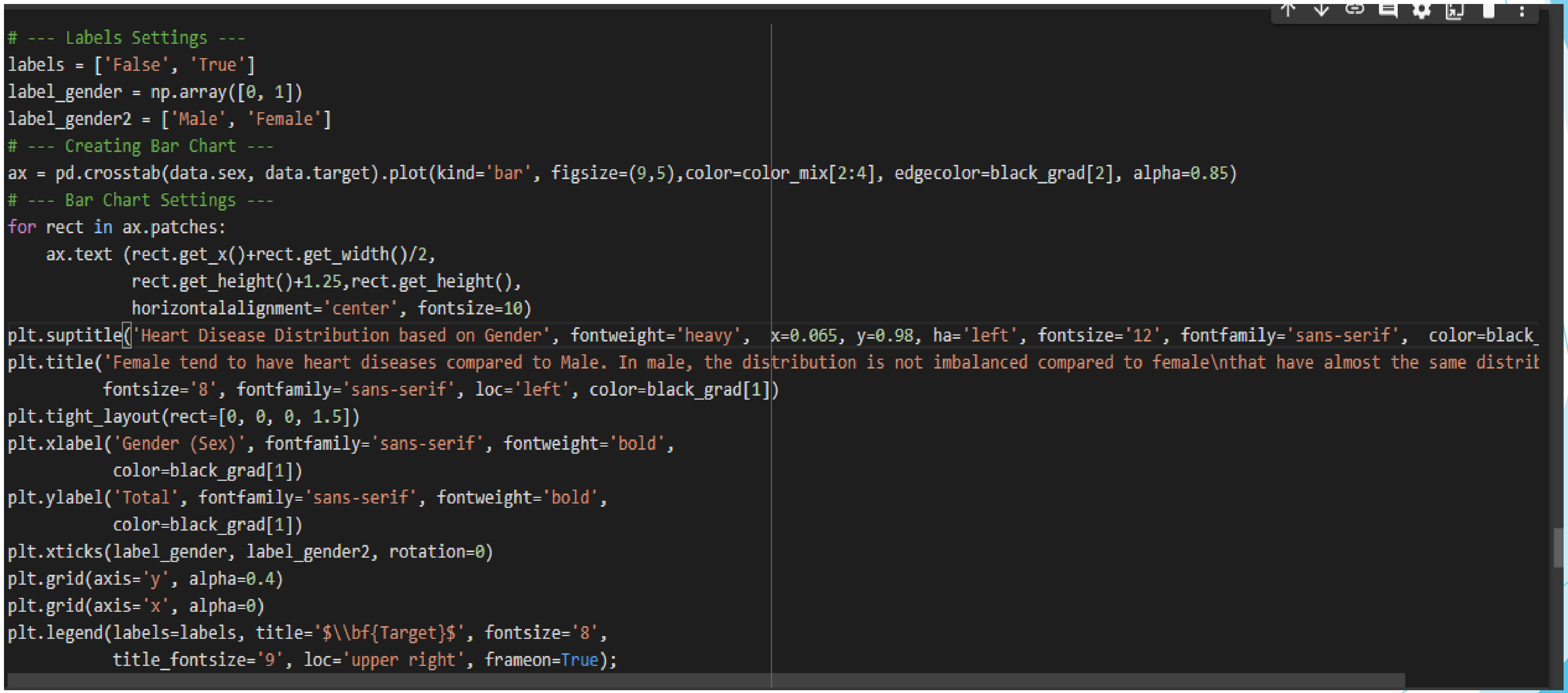
```
# --- Correlation Map (Heatmap) ---
plt.figure(figsize=(10, 5))
sns.heatmap(data.corr(), annot=True, cmap='Reds', linewidths=0.1)
plt.suptitle('Correlation Map of Numerical Variables', fontweight='heavy',
             x=0.03, y=0.98, ha='left', fontsize='12', fontfamily='sans-serif',
             color=black_grad[0])

plt.tight_layout(rect=[0, 0, 0, 1.1])
```



Correlation Map of Numerical Variables

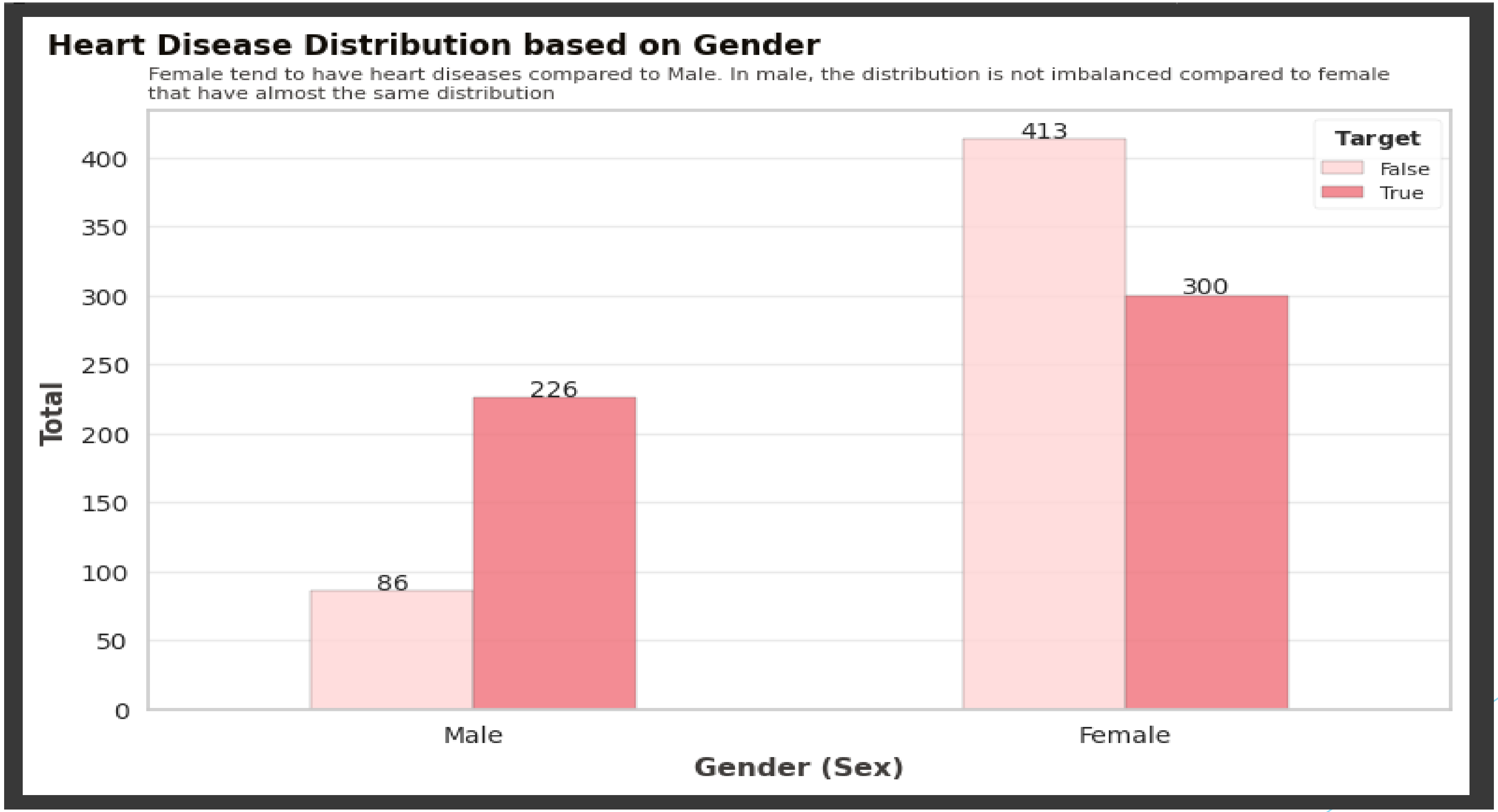
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.1	-0.072	0.27	0.22	0.12	-0.13	-0.39	0.088	0.21	-0.17	0.27	0.072	-0.23
sex	-0.1	1	-0.041	-0.079	-0.2	0.027	-0.055	-0.049	0.14	0.085	-0.027	0.11	0.2	-0.28
cp	-0.072	-0.041	1	0.038	-0.082	0.079	0.044	0.31	-0.4	-0.17	0.13	-0.18	-0.16	0.43
trestbps	0.27	-0.079	0.038	1	0.13	0.18	-0.12	-0.039	0.061	0.19	-0.12	0.1	0.059	-0.14
chol	0.22	-0.2	-0.082	0.13	1	0.027	-0.15	-0.022	0.067	0.065	-0.014	0.074	0.1	-0.1
fbs	0.12	0.027	0.079	0.18	0.027	1	-0.1	-0.0089	0.049	0.011	-0.062	0.14	-0.042	-0.041
restecg	-0.13	-0.055	0.044	-0.12	-0.15	-0.1	1	0.048	-0.066	-0.05	0.086	-0.078	0.021	0.13
thalach	-0.39	-0.049	0.31	-0.039	-0.022	-0.0089	0.048	1	-0.38	-0.35	0.4	-0.21	-0.098	0.42
exang	0.088	0.14	-0.4	0.061	0.067	0.049	-0.066	-0.38	1	0.31	-0.27	0.11	0.2	-0.44
oldpeak	0.21	0.085	-0.17	0.19	0.065	0.011	-0.05	-0.35	0.31	1	-0.58	0.22	0.2	-0.44
slope	-0.17	-0.027	0.13	-0.12	-0.014	-0.062	0.086	0.4	-0.27	-0.58	1	-0.073	-0.094	0.35
ca	0.27	0.11	-0.18	0.1	0.074	0.14	-0.078	-0.21	0.11	0.22	-0.073	1	0.15	-0.38
thal	0.072	0.2	-0.16	0.059	0.1	-0.042	0.021	-0.098	0.2	0.2	-0.094	0.15	1	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.1	-0.041	0.13	0.42	-0.44	-0.44	0.35	-0.38	-0.34	1

HEART DISEASE BASED ON THE GENDER:

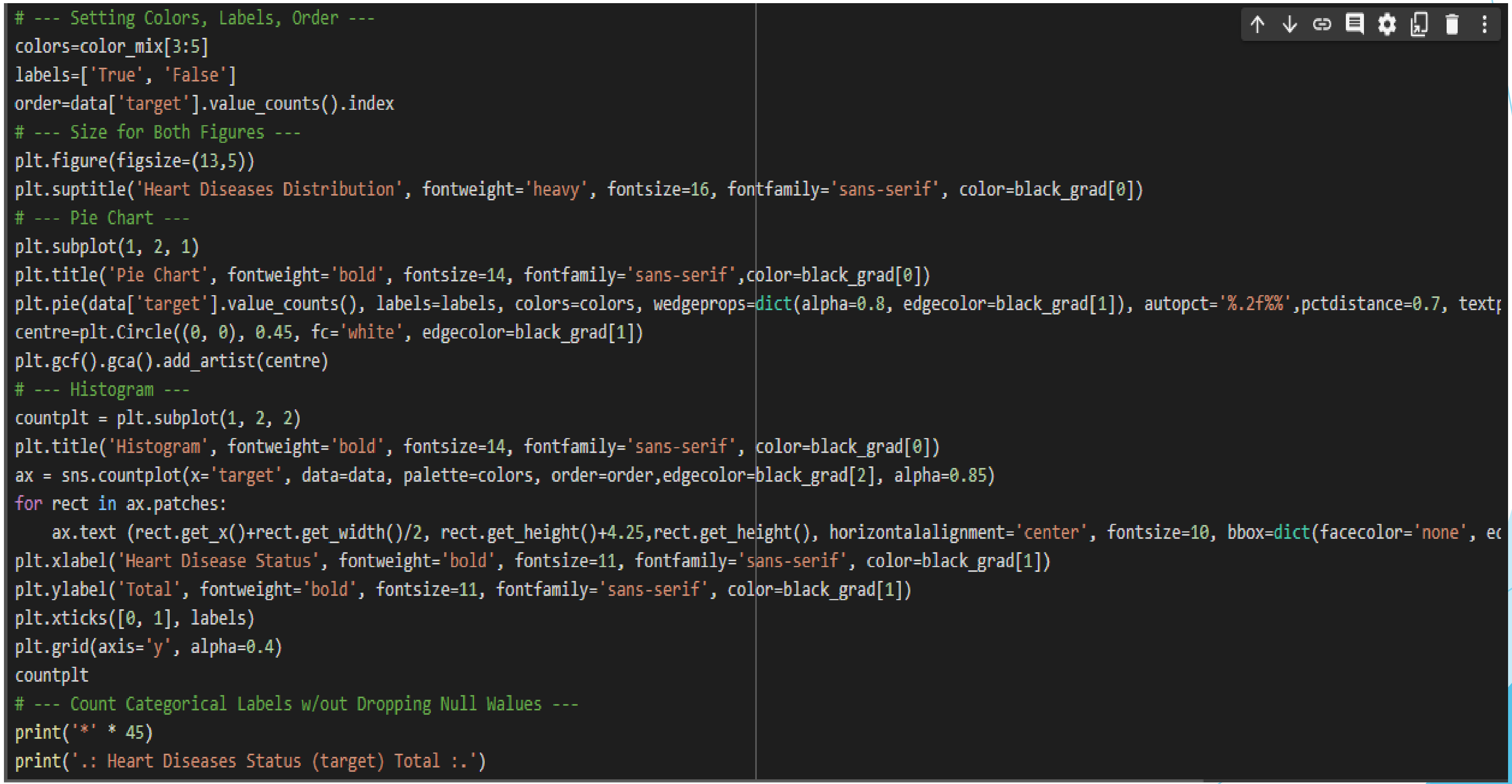


```
# --- Labels Settings ---
labels = ['False', 'True']
label_gender = np.array([0, 1])
label_gender2 = ['Male', 'Female']
# --- Creating Bar Chart ---
ax = pd.crosstab(data.sex, data.target).plot(kind='bar', figsize=(9,5),color=color_mix[2:4], edgecolor=black_grad[2], alpha=0.85)
# --- Bar Chart Settings ---
for rect in ax.patches:
    ax.text (rect.get_x()+rect.get_width()/2,
             rect.get_height()+1.25,rect.get_height(),
             horizontalalignment='center', fontsize=10)
plt.suptitle('Heart Disease Distribution based on Gender', fontweight='heavy',  x=0.065, y=0.98, ha='left', fontsize='12', fontfamily='sans-serif',  color=black_
plt.title('Female tend to have heart diseases compared to Male. In male, the distribution is not imbalanced compared to female\nthat have almost the same distrit
          fontsize='8', fontfamily='sans-serif', loc='left', color=black_grad[1])
plt.tight_layout(rect=[0, 0, 0, 1.5])
plt.xlabel('Gender (Sex)', fontfamily='sans-serif', fontweight='bold',
           color=black_grad[1])
plt.ylabel('Total', fontfamily='sans-serif', fontweight='bold',
           color=black_grad[1])
plt.xticks(label_gender, label_gender2, rotation=0)
plt.grid(axis='y', alpha=0.4)
plt.grid(axis='x', alpha=0)
plt.legend(labels=labels, title='$\\bf{Target}$', fontsize='8',
           title_fontsize='9', loc='upper right', frameon=True);
```

VISUALIZATION OF GENDER:

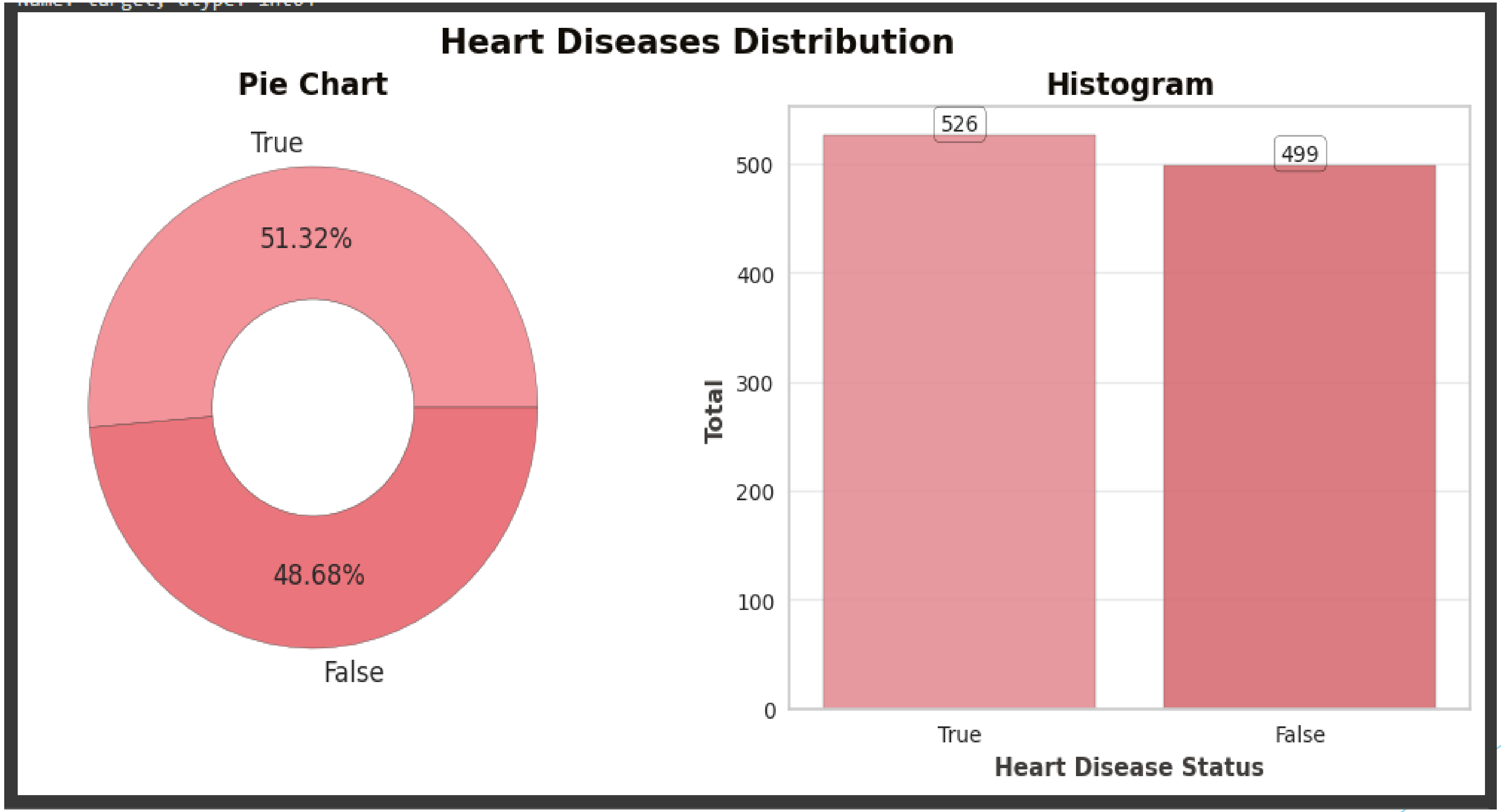


NUMBER OF PATIENTS AFFECTED BY HEART DISEASE:

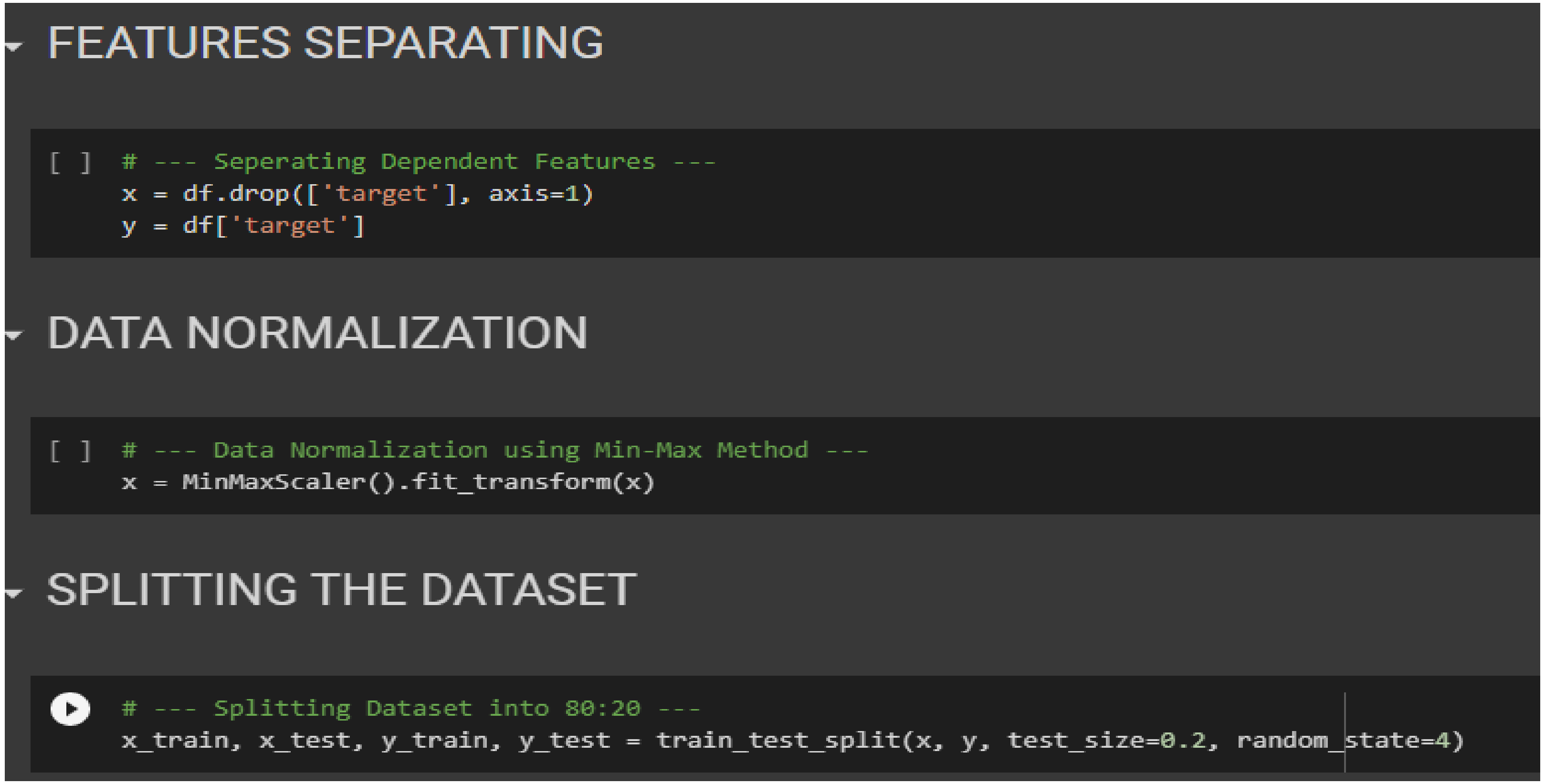


```
# --- Setting Colors, Labels, Order ---
colors=color_mix[3:5]
labels=['True', 'False']
order=data['target'].value_counts().index
# --- Size for Both Figures ---
plt.figure(figsize=(13,5))
plt.suptitle('Heart Diseases Distribution', fontweight='heavy', fontsize=16, fontfamily='sans-serif', color=black_grad[0])
# --- Pie Chart ---
plt.subplot(1, 2, 1)
plt.title('Pie Chart', fontweight='bold', fontsize=14, fontfamily='sans-serif',color=black_grad[0])
plt.pie(data['target'].value_counts(), labels=labels, colors=colors, wedgeprops=dict(alpha=0.8, edgecolor=black_grad[1]), autopct='%.2f%%',pctdistance=0.7, textp
centre=plt.Circle((0, 0), 0.45, fc='white', edgecolor=black_grad[1])
plt.gcf().gca().add_artist(centre)
# --- Histogram ---
countplt = plt.subplot(1, 2, 2)
plt.title('Histogram', fontweight='bold', fontsize=14, fontfamily='sans-serif', color=black_grad[0])
ax = sns.countplot(x='target', data=data, palette=colors, order=order,edgecolor=black_grad[2], alpha=0.85)
for rect in ax.patches:
    ax.text (rect.get_x()+rect.get_width()/2, rect.get_height()+4.25,rect.get_height(), horizontalalignment='center', fontsize=10, bbox=dict(facecolor='none', ec
plt.xlabel('Heart Disease Status', fontweight='bold', fontsize=11, fontfamily='sans-serif', color=black_grad[1])
plt.ylabel('Total', fontweight='bold', fontsize=11, fontfamily='sans-serif', color=black_grad[1])
plt.xticks([0, 1], labels)
plt.grid(axis='y', alpha=0.4)
countplt
# --- Count Categorical Labels w/out Dropping Null Walues ---
print('*' * 45)
print('.: Heart Diseases Status (target) Total :.')
```

VISUALIZATION TARGET VARIABLE:



DATA FEATURING,SPLITTING:



FEATURES SEPARATING

```
[ ]  # --- Seperating Dependent Features ---
     x = df.drop(['target'], axis=1)
     y = df['target']
```

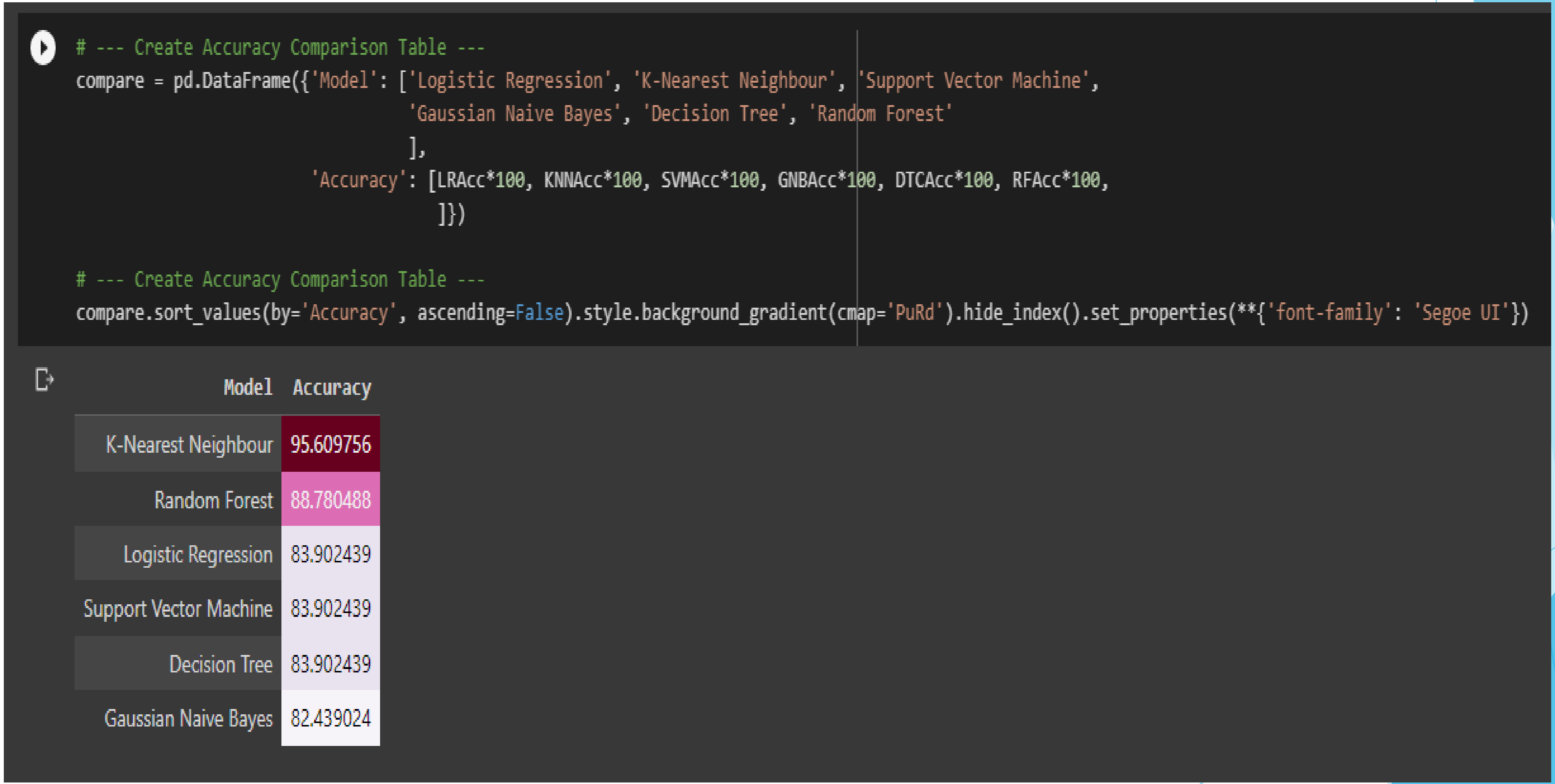
DATA NORMALIZATION

```
[ ]  # --- Data Normalization using Min-Max Method ---
     x = MinMaxScaler().fit_transform(x)
```

SPLITTING THE DATASET

```
     # --- Splitting Dataset into 80:20 ---
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=4)
```

ACCURACY OF THE DATASET:



```
# --- Create Accuracy Comparison Table ---
compare = pd.DataFrame({'Model': ['Logistic Regression', 'K-Nearest Neighbour', 'Support Vector Machine',
                                  'Gaussian Naive Bayes', 'Decision Tree', 'Random Forest'
                                  ],
                        'Accuracy': [LRAcc*100, KNNAcc*100, SVMAcc*100, GNBAcc*100, DTCAcc*100, RFAcc*100,
                                     ]})

# --- Create Accuracy Comparison Table ---
compare.sort_values(by='Accuracy', ascending=False).style.background_gradient(cmap='PuRd').hide_index().set_properties(**{'font-family': 'Segoe UI'})
```

Model	Accuracy
K-Nearest Neighbour	95.609756
Random Forest	88.780488
Logistic Regression	83.902439
Support Vector Machine	83.902439
Decision Tree	83.902439
Gaussian Naive Bayes	82.439024

CONCLUSION :

- ✓ The overall aim is to define various Machine Learning Techniques useful in effective heart disease prediction.

- ✓ Efficient and accurate prediction with a lesser number of attributes and tests is the goal of this research.

- ✓ There is a need to implement more complex and combinations of models to get higher accuracy for early prediction of heart disease.

REFERENCE:

[1]Author : S.SEEMA , “Heart disease prediction using machine learning techniques ”. International Journal of Computer Applications (0975 – 8887) Volume 181 - No.18,September 2018.Available at: https://www.ijcaonline.org

[2]Author : Singh Yeshvendra ,” International Journal of Advanced Research in Science, Communication and Technology (IJARSCT),ISSN (Online) 2581-9429; Impact Factor: 4.819,Volume 5, Issue 1, May 2021 Available at www.ijarsct.co.in

[3]Author: Apurv garg, “International Journal for Research in Applied Science & Engineering Technology (IJRASET) ,ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429 ,Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

[4]Authors : Ashwini Shetty A, Chandra Naik, “Different Data Mining Approaches for Predicting Heart Disease”, International Journal of Innovative in Science Engineering and Technology, Vol.5, May 2016, pp.277-281,Available at https://www.researchgate.net

[5]Author : Balakrishnan , S.,Syed Muzamil Basha , & Ravi Kumar Poluru , 2019. Heart Disease Prediction Using Machine Learning Algorithm. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-10.Available at https://bbrc.in