

Heart Failure Prediction

Govindula Sai Vaishnavi
Instructor: Mr. Gahangir Hossain
Data Visualization INFO 5709
Data Visualization – Final Project
13/12/2023

Introduction

Heart failure is a hazardous condition that affects about 550,000 individuals annually. Heart failure is a severe condition that profoundly affects people's lives. The bulk of people constantly disregard their health due to their fast-paced lifestyle, increased food quantities, harmful habits, and lack of physical activity. Other factors that affect heart failure include age, cholesterol, rapid blood sugar, chest discomfort, and maximum heart rate. contribute to the problem of heart failure, which may worsen in the future as the environment deteriorates. People would eventually die from heart failure if they disregarded it. Improving forecast is a key tactic for lessening the effects of this problem. With a variety of variables, machine learning and linear models can anticipate cardiac failure.

Dataset:

After accounting for 31% of all deaths worldwide, cardiovascular diseases (CVDs) are the leading cause of death, taking an estimated 17.9 million lives annually. Heart attacks and strokes account for four out of every five CVD deaths, with persons under the age of 70 accounting for one-third of these premature deaths. Eleven characteristics in this dataset can be used to predict the likelihood of a heart disease, as heart failure is a common event brought on by CVDs.

When it comes to early detection and management, a machine learning model can be very helpful for people who have cardiovascular disease or who are at high risk for developing it because they have one or more risk factors, such as diabetes, hypertension, hyperlipidaemia, or an existing illness.

Dataset is collected from Kaggle:

<https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction>

There are total 12 attributes used to describe the dataset. Below is the list

Age: age of the patient [years]

Sex: sex of the patient [M: Male, F: Female]

ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]

RestingBP: resting blood pressure [mm Hg]

Cholesterol: serum cholesterol [mm/dl]

FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]

RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]

MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]

ExerciseAngina: exercise-induced angina [Y: Yes, N: No]

Oldpeak: oldpeak = ST [Numeric value measured in depression]

ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]

Heart Disease

Tools

- Colab (Python)
- Tableau

Data Preprocessing

The data has undergone rigorous processing, including cleaning and the removal of outliers and irrelevant entries, to enable exploratory data analysis and hypothesis testing. In order to make the curated dataset easily accessible and integreatable for further analysis, it has been converted into a structured CSV file.

```
#importing libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[29] # Load the Heart Failure Prediction Dataset
dataset_path = '/content/heart.csv'
heart_failure_data = pd.read_csv(dataset_path)
```

```
heart_failure_data.head
```

	<bound	method	NDFrame.head of	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	\
0	40	M	ATA	140		289	0	Normal			
1	49	F	NAP	160		180	0	Normal			
2	37	M	ATA	130		283	0	ST			
3	48	F	ASY	138		214	0	Normal			
4	54	M	NAP	150		195	0	Normal			
..			
913	45	M	TA	110		264	0	Normal			
914	68	M	ASY	144		193	1	Normal			
915	57	M	ASY	130		131	0	Normal			
916	57	F	ATA	130		236	0	LVH			
917	38	M	NAP	138		175	0	Normal			

	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	172	N	0.0	Up	0
1	156	N	1.0	Flat	1
2	98	N	0.0	Up	0
3	108	Y	1.5	Flat	1
4	122	N	0.0	Up	0
..
913	132	N	1.2	Flat	1
914	141	N	3.4	Flat	1
915	115	Y	1.2	Flat	1
916	174	N	0.0	Flat	1
917	173	N	0.0	Up	0

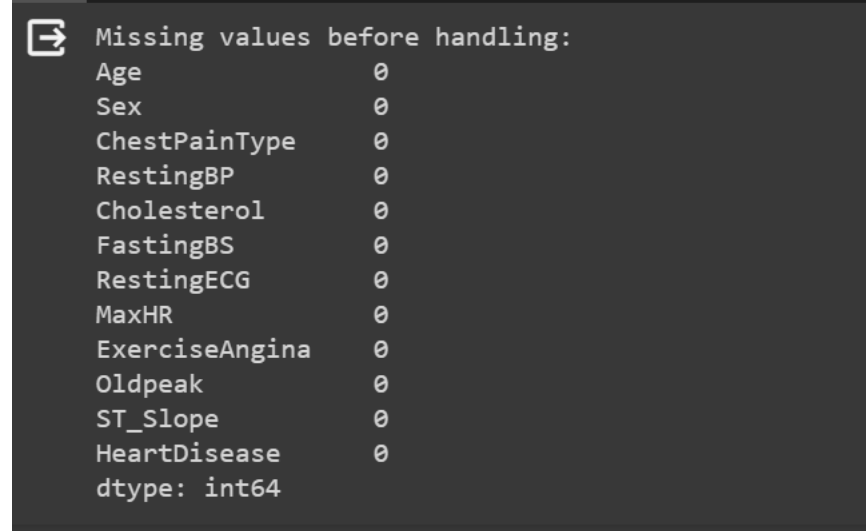
[918 rows x 12 columns]>

Data Cleaning

Handling Missing Values:

Recognise and manage the dataset's missing values. The analysis and machine learning models may be impacted by missing values. One can use advanced imputation techniques, remove rows containing missing values, or impute missing values using the mean or media and also many more of the strategies for only handling the missing data.

```
# Display information about missing values
print("Missing values before handling:")
print(heart_failure_data.isnull().sum())
```



Missing values before handling:	
Age	0
Sex	0
ChestPainType	0
RestingBP	0
Cholesterol	0
FastingBS	0
RestingECG	0
MaxHR	0
ExerciseAngina	0
Oldpeak	0
ST_Slope	0
HeartDisease	0
dtype: int64	

Dealing with Outliers:

Recognise and deal with data outliers. The outcomes of statistical analysis and machine learning models can be considerably impacted by outliers. The removal or transformation of the outliers from the data will depend on their nature.



```
# Display boxplots for numerical variables to identify outliers
numerical_columns = heart_failure_data.select_dtypes(include='number').columns
num_plots = len(numerical_columns)

# Calculate the number of rows and columns for subplots
num_rows = (num_plots // 3) + (num_plots % 3)
num_cols = min(3, num_plots)

plt.figure(figsize=(14, 8))

for i, column in enumerate(numerical_columns, 1):
    plt.subplot(num_rows, num_cols, i)
    sns.boxplot(x=heart_failure_data[column])
    plt.title(f'Boxplot for {column}')

plt.tight_layout()
plt.show()

# Identify and handle outliers for all numerical columns
def handle_outliers(data):
    for column in numerical_columns:
        Q1 = data[column].quantile(0.25)
        Q3 = data[column].quantile(0.75)
        IQR = Q3 - Q1

        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
```



```
# Filter data within the bounds
data = data[(data[column] >= lower_bound) & (data[column] <= upper_bound)]

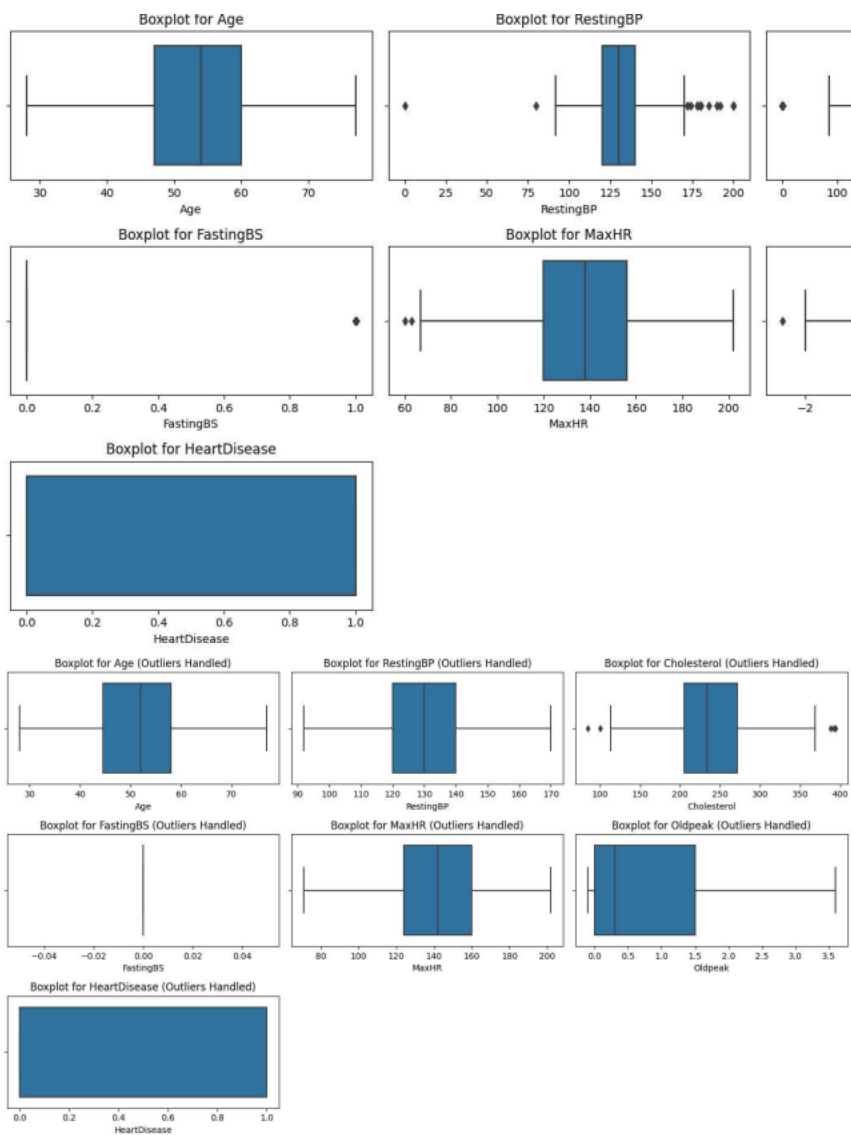
return data

# Apply the outlier handling function to the entire dataset
heart_data = handle_outliers(heart_failure_data)

plt.figure(figsize=(14, 8))

for i, column in enumerate(numerical_columns, 1):
    plt.subplot(num_rows, num_cols, i)
    sns.boxplot(x=heart_data[column])
    plt.title(f'Boxplot for {column} (Outliers Handled)')

plt.tight_layout()
plt.show()
```



Standardizing or Normalizing Data:

To bring numerical features to a common scale, standardise or normalise them. This is crucial, particularly when applying algorithms for machine learning are sensitive to the size of features. Min-max scaling and z-score normalisation are examples of common methods.

```
# standardize feature
import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler
summary = heart_failure_data.describe()
variables_greater_than_100 = summary.columns[(summary.loc['max'] > 100).values].tolist()
print("Variables with max values greater than 100:", variables_greater_than_100)
standard_scaler = StandardScaler()
heart_failure_data[variables_greater_than_100[0]] = standard_scaler.fit_transform(heart_failure_data[[variables_greater_than_100[0]]])
print(heart_failure_data.describe())
```

Variables with max values greater than 100: ['RestingBP', 'Cholesterol', 'MaxHR']

	Age	RestingBP	Cholesterol	FastingBS	MaxHR
count	918.000000	9.180000e+02	918.000000	918.000000	918.000000
mean	53.510893	1.954380e-16	198.799564	0.233115	136.809368
std	9.432617	1.000545e+00	109.384145	0.423046	25.460334
min	28.000000	-7.154995e+00	0.000000	0.000000	60.000000
25%	47.000000	-6.699346e-01	173.250000	0.000000	120.000000
50%	54.000000	-1.295128e-01	223.000000	0.000000	138.000000
75%	60.000000	4.109089e-01	267.000000	0.000000	156.000000
max	77.000000	3.653439e+00	603.000000	1.000000	202.000000

	Oldpeak	HeartDisease
count	918.000000	918.000000
mean	0.887364	0.553377
std	1.066570	0.497414
min	-2.600000	0.000000
25%	0.000000	0.000000
50%	0.600000	1.000000
75%	1.500000	1.000000
max	6.200000	1.000000

```
# normalize feature
variables_greater_than_100 = summary.columns[(summary.loc['max'] > 100).values].tolist()
print("Variables with max values greater than 100:", variables_greater_than_100)
min_max_scaler = MinMaxScaler()
heart_failure_data[variables_greater_than_100[1]] = min_max_scaler.fit_transform(heart_failure_data[[variables_greater_than_100[1]]])
print(heart_failure_data.describe())
```

Variables with max values greater than 100: ['RestingBP', 'Cholesterol', 'MaxHR']

	Age	RestingBP	Cholesterol	FastingBS	MaxHR
count	918.000000	9.180000e+02	918.000000	918.000000	918.000000
mean	53.510893	1.954380e-16	0.329684	0.233115	136.809368
std	9.432617	1.000545e+00	0.181400	0.423046	25.460334
min	28.000000	-7.154995e+00	0.000000	0.000000	60.000000
25%	47.000000	-6.699346e-01	0.287313	0.000000	120.000000
50%	54.000000	-1.295128e-01	0.369818	0.000000	138.000000
75%	60.000000	4.109089e-01	0.442786	0.000000	156.000000
max	77.000000	3.653439e+00	1.000000	1.000000	202.000000

	Oldpeak	HeartDisease
count	918.000000	918.000000
mean	0.887364	0.553377
std	1.066570	0.497414
min	-2.600000	0.000000
25%	0.000000	0.000000
50%	0.600000	1.000000
75%	1.500000	1.000000
max	6.200000	1.000000

Checking for Duplicates:

If duplicate rows are present in the dataset, locate them and eliminate them. Analyses and model training may be distorted by duplicate entries.

```
# Display duplicate rows
duplicate_rows = heart_failure_data[heart_failure_data.duplicated()]
print("Duplicate Rows:")
print(duplicate_rows)
```

Duplicate Rows:
Empty DataFrame
Columns: [Age, Sex, ChestPainType, RestingBP, Cholesterol, FastingBS, RestingECG, MaxHR, ExerciseAngina, Oldpeak, ST_Slope, HeartDisease]
Index: []

Exploratory Data Analysis

The aim of Exploratory Data Analysis is to investigate possible correlations and relationships between variables by utilising the diverse attributes found in the dataset.

The process of visually and statistically examining datasets to find patterns, relationships, and trends is known as exploratory data analysis, or EDA. It entails enumerating key features, spotting anomalies, comprehending variable distributions, and looking for possible correlations. EDA aids in the formation of hypotheses, directs the selection of features, and offers perceptions into the underlying structure of the data, enabling well-informed decision-making in follow-up analyses.

```

# Display basic information about the dataset
print("Dataset Overview:")
print(heart_failure_data.info())

# Descriptive statistics for numerical variables
print("\nDescriptive Statistics:")
print(heart_failure_data.describe())

```

```

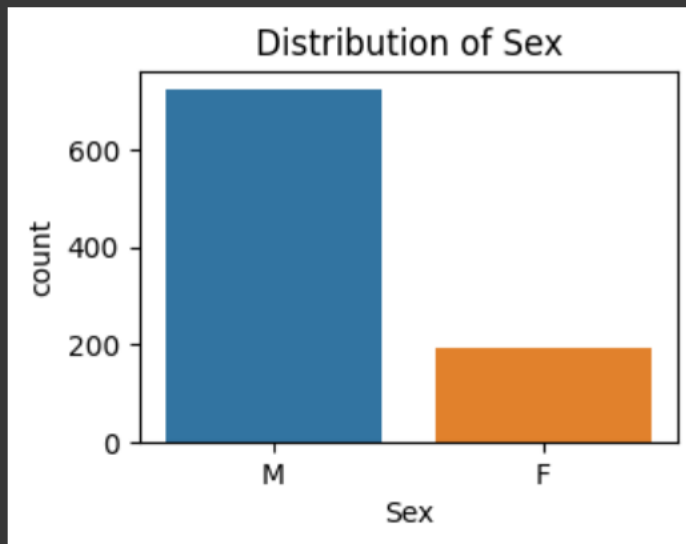
Dataset Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Age                   918 non-null   int64
1   Sex                   918 non-null   object
2   ChestPainType         918 non-null   object
3   RestingBP             918 non-null   int64
4   Cholesterol           918 non-null   int64
5   FastingBS             918 non-null   int64
6   RestingECG            918 non-null   object
7   MaxHR                 918 non-null   int64
8   ExerciseAngina        918 non-null   object
9   Oldpeak               918 non-null   float64
10  ST_Slope              918 non-null   object
11  HeartDisease          918 non-null   int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
None

```

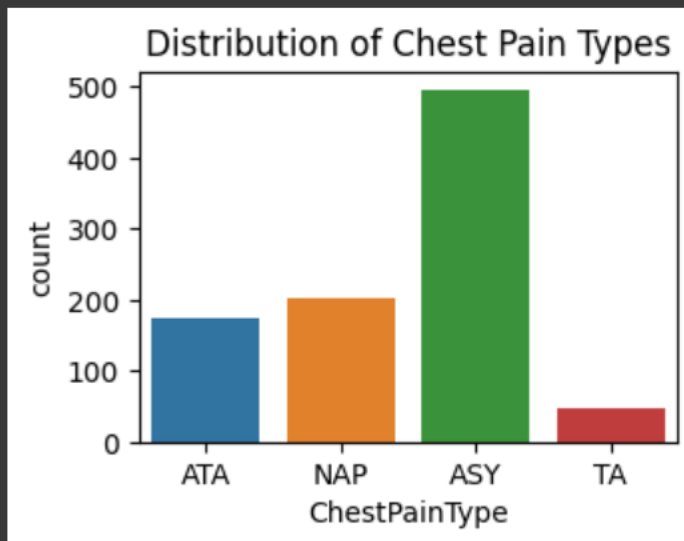
This part uses the `info()` method to print an overview of the dataset, including details about the data types, memory usage, and non-null counts.

The `describe()` method prints count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values for numerical variables.

```
# Bar plot for 'Sex' distribution
plt.subplot(2, 2, 1)
sns.countplot(x='Sex', data=heart_failure_data)
plt.title('Distribution of Sex')
plt.tight_layout()
plt.show()
```

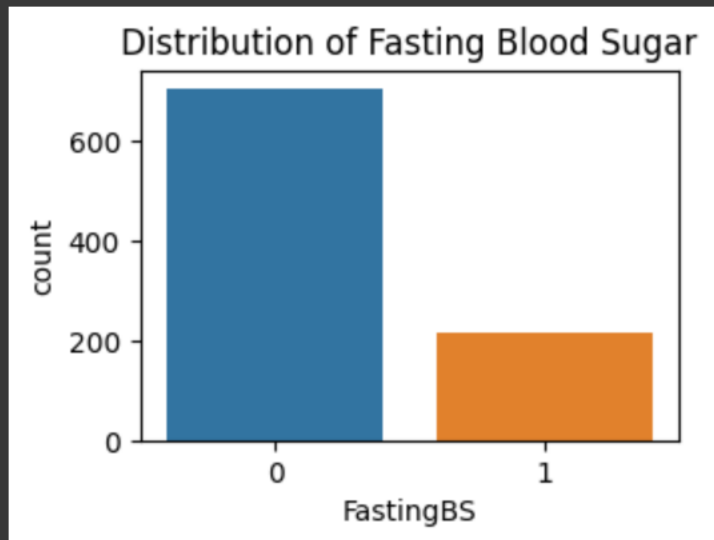


```
[ ] # Bar plot for 'ChestPainType' distribution
plt.subplot(2, 2, 2)
sns.countplot(x='ChestPainType', data=heart_failure_data)
plt.title('Distribution of Chest Pain Types')
plt.tight_layout()
plt.show()
```



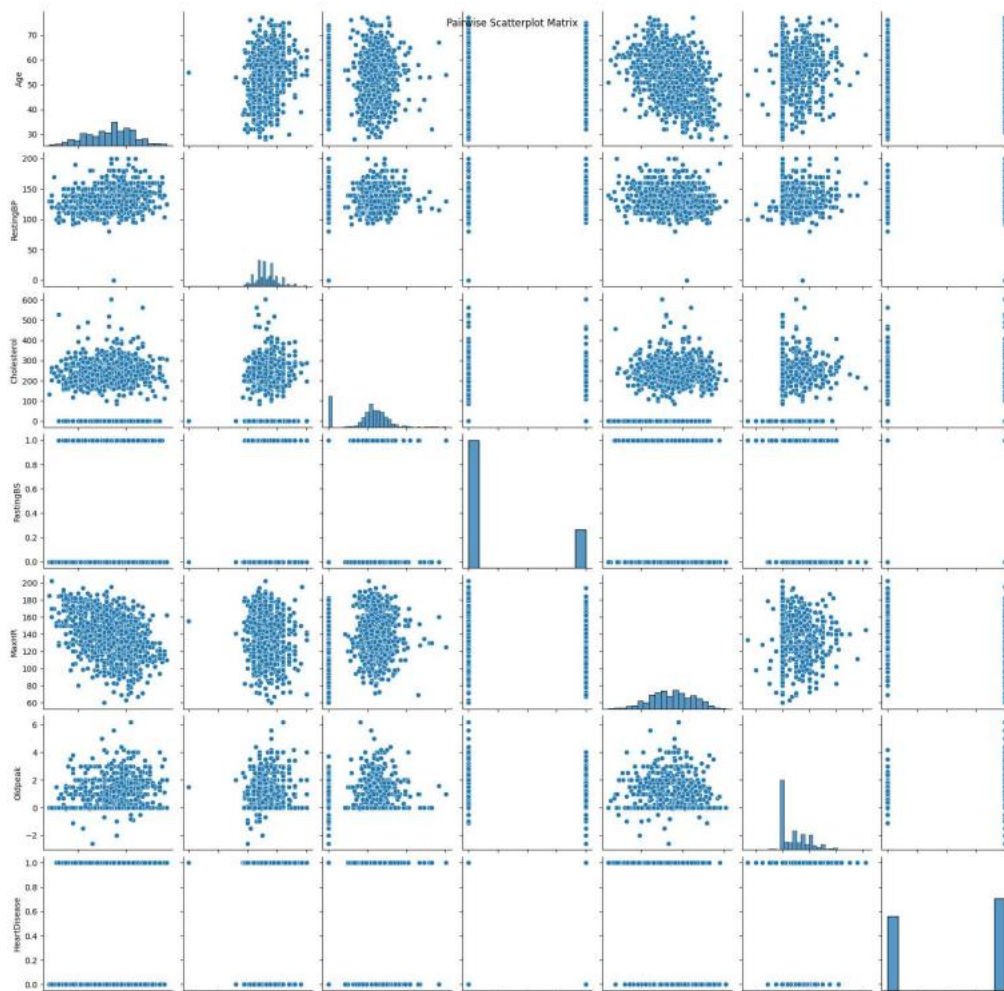


```
# Bar plot for 'FastingBS' distribution
plt.subplot(2, 2, 3)
sns.countplot(x='FastingBS', data=heart_failure_data)
plt.title('Distribution of Fasting Blood Sugar')
plt.tight_layout()
plt.show()
```



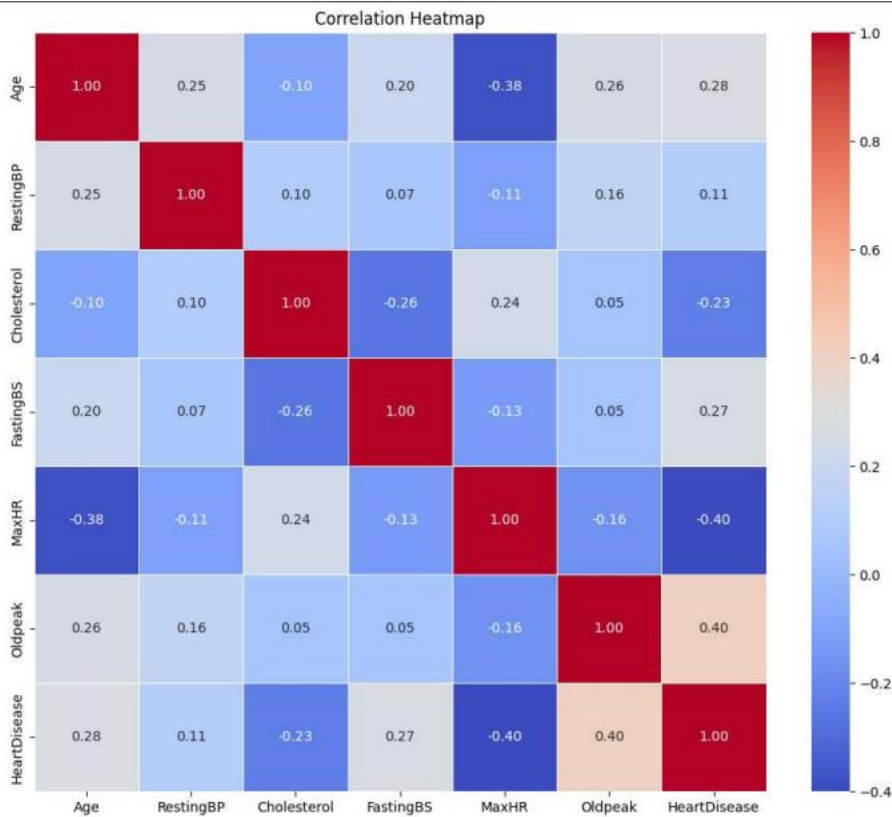
To display bar plots of the distribution of the categorical variables "Sex," "ChestPainType," "FastingBS," this section creates a 2x2 subplot. For every plot, it makes use of Seaborn's `sns.countplot` and positions them using `plt.subplot`. Plots are displayed by `plt.show()`, and appropriate spacing is guaranteed by `plt.tight_layout()`.

```
[ ] # Pairwise scatterplot matrix for numerical variables
sns.pairplot(heart_failure_data)
plt.suptitle("Pairwise Scatterplot Matrix")
plt.show()
```



In this section, `sns.pairplot` is used to generate a pairwise scatterplot matrix for numerical variables. `plt.suptitle` is used to set the title, and `plt.show()` is used to display the plot as a whole.

```
# Correlation heatmap
correlation_matrix = heart_failure_data.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=.5)
plt.title('Correlation Heatmap')
plt.show()
```



Lastly, for the correlation matrix determined by `heart_data.corr()`, a correlation heatmap is produced using `sns.heatmap`. Annotations (`annot=True`), a particular colormap (`cmap='coolwarm'`), and formatting to show two decimal places (`fmt='.2f'`) are all included in the heatmap. Using `plt.show()`, it is shown.

Data Visualizations

A minimum of three data visualisations have been created using different visualisation formats, like bar charts, scatter plots, or other pertinent methods, in order to address the hypotheses. At least two interactive elements are included in the visualisations, encouraging a dynamic dataset exploration. Design considerations that adhere to the guidelines of efficient data communication include colour selections, spatial layouts, and two-dimensional space designs. A clear and intuitive interpretation of the results is made possible by the meticulous craftsmanship with which each visualisation addresses a particular aspect of the hypotheses.

Hypothesis

Hypothesis 1: Is there a relationship between age and maximum heart rate?

Justification: By examining potential trends or correlations between an individual's age and maximum heart rate, we hope to gain more understanding of the cardiovascular health of ageing.

Hypothesis 2: Do individuals with different chest pain types have distinct cholesterol levels?

Justification: We want to find out if there are any correlations between particular types of chest pain and particular patterns of cholesterol by looking at cholesterol levels across a variety of chest pain types.

Hypothesis 3: Does exercise angina affect the oldpeak during exercise?

Justification: In order to shed light on the body's reaction to exercise stress, we are examining if the existence of exercise-induced angina affects.

Hypothetical Visualizations:

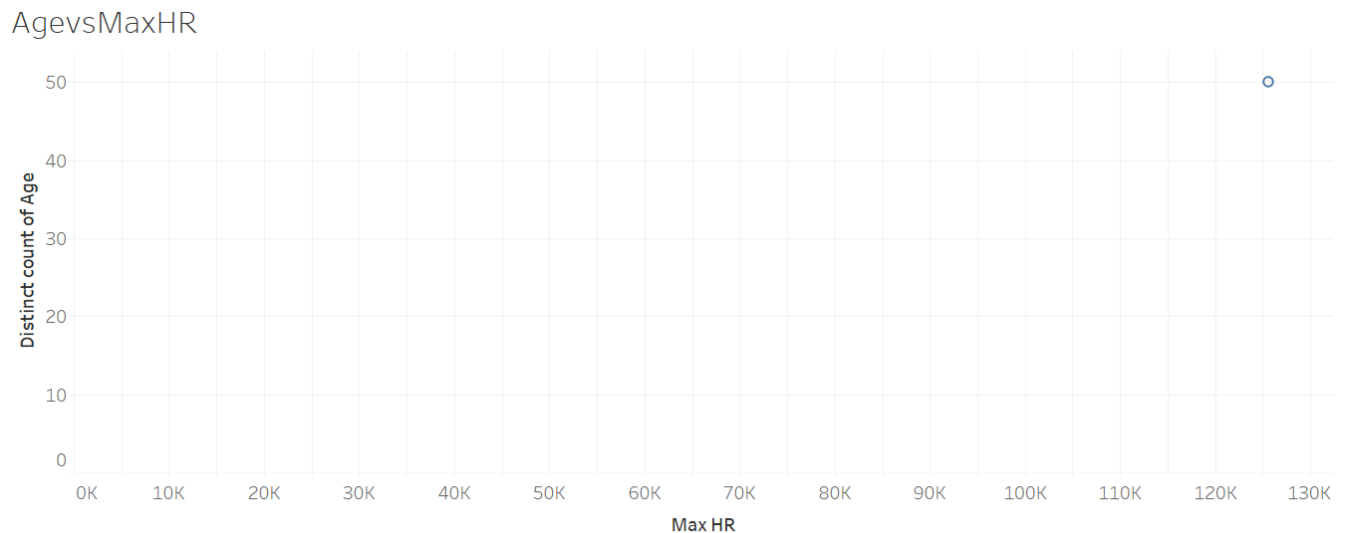
Scatterplot for Age vs. MaxHR:

Perceptual tasks include evaluating correlation, spotting trends, and comparing individual data points.

Visualisation Format: Scatterplot to illustrate the relationship's general trajectory.

Two-dimensional design with age on the x- and maxHR on the y-axes to show their correlation in an easy-to-understand manner.

Colour and Style: For easier understanding, age groups are represented by colour gradients.



Sum of Max HR vs. distinct count of Age.

Result

As age increases, the maximum heart rate tends to decrease.

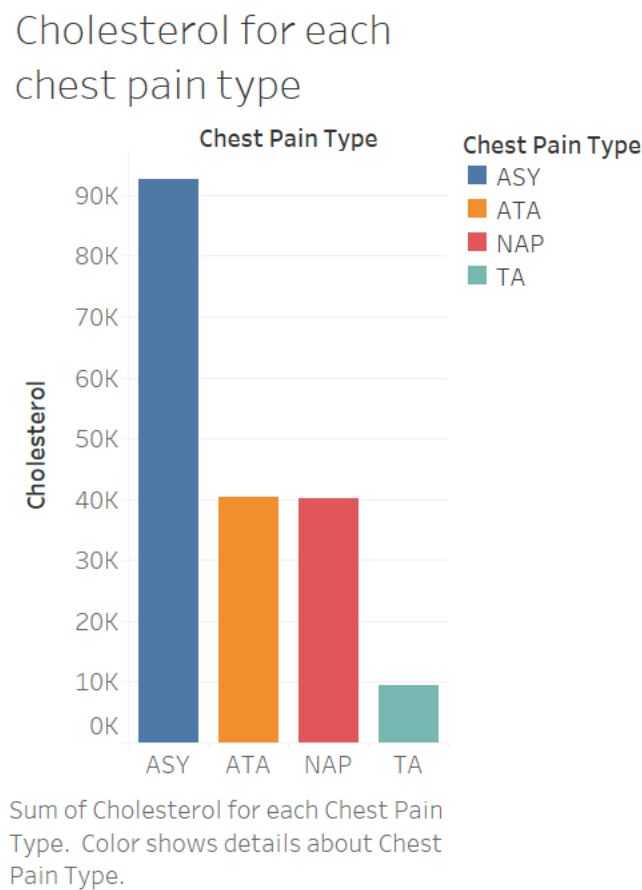
Grouped Bar Chart for Cholesterol Levels by Chest Pain Type:

Perceptual tasks include comparing categorical distributions and spotting trends in the various forms of chest pain.

Display Format: A bar chart with groups to indicate the cholesterol levels for each type of chest pain.

Two-dimensional design that makes comparisons simple by placing cholesterol levels on the y-axis and different types of chest pain on the x-axis.

Colour and Style: For clarity, give each type of chest pain a different colour.



Result

The grouped bar chart showed how different types of chest pain corresponded to different cholesterol levels. This graphic proved that specific types of chest pain do, in fact, exhibit different patterns of cholesterol. This realisation may be essential to comprehending the implications for cardiovascular health.

Faceted Scatterplot for Age vs. Oldpeak with Exercise Angina Interaction:

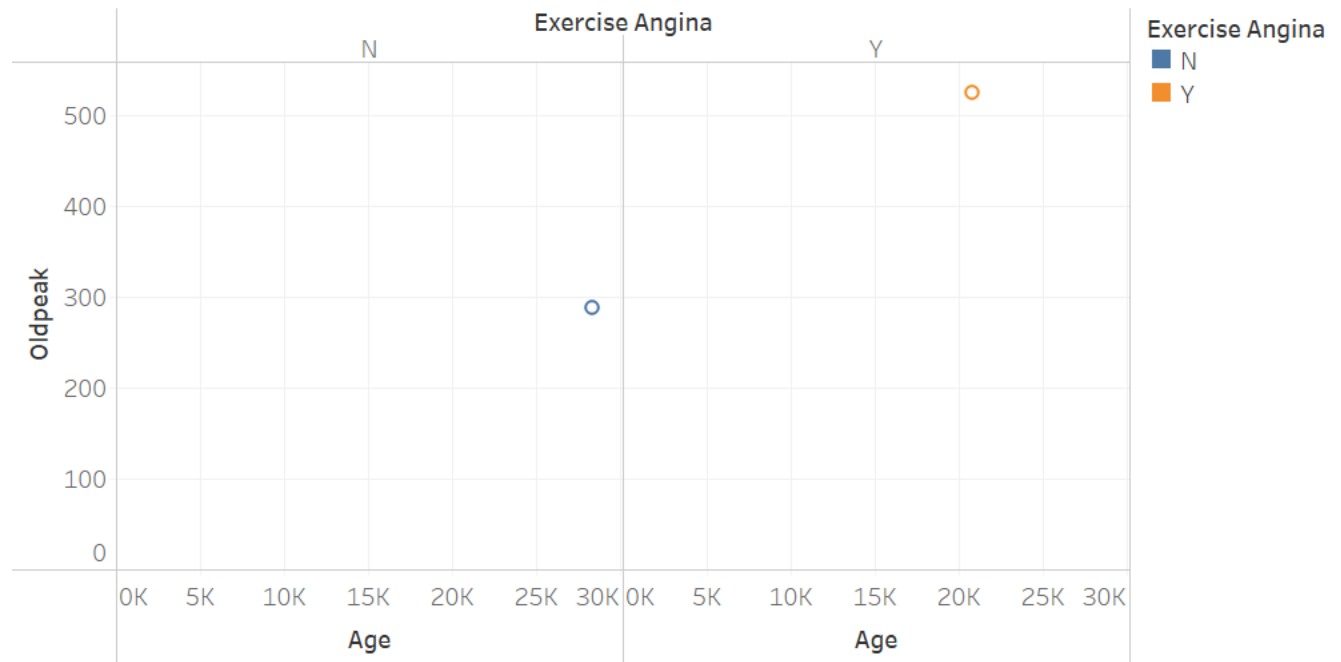
Perceptual tasks include evaluating patterns among various subgroups and spotting possible relationships.

Visualisation Format: Faceted scatterplot with distinct facets for varying degrees of exercise-induced angina, age on the x-axis, and oldpeak on the y-axis.

Two-dimensional design that allows comparisons within each subgroup: Age and oldpeak on the same plot, with facets denoting varying degrees of exercise-induced angina.

Colour and Style: To improve the interactive effect, separate data points with and without exercise angina by using different markers or colours.

Age vs Oldpeak divided by Exercise Angina



Sum of Age vs. sum of Oldpeak broken down by Exercise Angina. Color shows details about Exercise Angina.

Result

An intricate picture of the relationship between age, oldpeak, and exercise-related angina was given by the faceted scatterplot. It demonstrated that exercise angina affected old peak, confirming the hypothesis.

Conclusion

In the procedure described above, our goal was to look into different facets of a dataset that was associated with heart failure prediction. Important characteristics in the dataset include blood pressure, cholesterol levels, age, sex, type of chest pain, and other health indicators. The objective was to use data visualisations to formulate and address three straightforward hypotheses in order to uncover possible connections and trends.

The distinct patterns seen in the visualisations demonstrate that the dataset was appropriate for answering the hypotheses.

Research on cardiovascular health benefits greatly from an understanding of the effects of exercise-induced angina, age-related changes in maximum heart rate, and cholesterol patterns across chest pain types.

References

1. Eletter, S., Yasmin, T., Elrefae, G., Aliter, H., & Elrefae, A. (2020). Building an intelligent telemonitoring system for heart failure: The use of the internet of things, Big Data, and machine learning. 2020 21st International Arab Conference on Information Technology (ACIT). <https://doi.org/10.1109/acit50332.2020.9300113>
2. Kim, Y.-T., Kim, D.-K., Kim, H., & Kim, D.-J. (2020). A comparison of oversampling methods for constructing a prognostic model in the patient with heart failure. 2020 International Conference on Information and Communication Technology Convergence (ICTC). <https://doi.org/10.1109/ictc49870.2020.9289522>
3. Heart failure prediction by feature ranking analysis in Machine Learning. (n.d.). Retrieved December 10, 2022, from <https://ieeexplore.ieee.org/abstract/document/9358733/>
4. Heart Failure Prediction Dataset <https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction>