

## **Exploratory Data Analysis - Titanic Dataset**

Soniya Khadka

Westcliff University

TECH405: Artificial Neural Network and Deep Learning

Professor Acharya

November 3, 2024

### **Abstract**

This report explores the Titanic dataset to predict the survival rates of passengers through various survival factors and prepare data for neural network implementation. Exploratory Data Analysis (EDA) and feature engineering is performed by identifying patterns through visualization, addressing missing values, and encoding categorical variables. By performing these steps, it sets the stage for predictive modeling in future assignments.

## Table of Contents

<b>Introduction</b>	<b>5</b>
<b>Methodology and Findings</b>	<b>6</b>
<b>Discussion</b>	<b>16</b>
<b>References</b>	<b>17</b>

## List of Figures

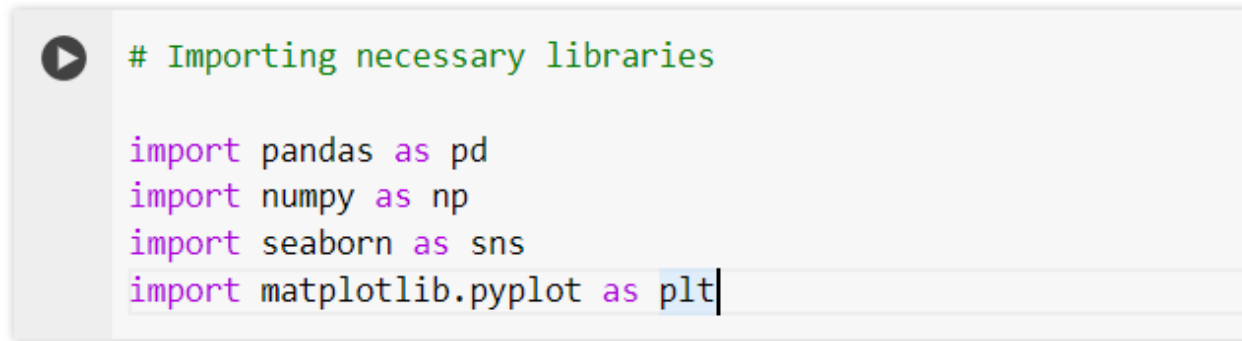
Fig i. Importing necessary libraries	5
Fig ii. Loading the dataset and displaying head	5
Fig iii. Information of dataset	6
Fig iv. Statistical summary of numerical columns	7
Fig v. Checking missing values	7
Fig vi. Column chart to visualize the distribution of survival	8
Fig vii. Distribution of Age	9
Fig viii. Survival Rate by Gender	10
Fig ix. Survival Rate by Passenger Class	11
Fig x. Filling missing values of “Age” with median	12
Fig xi. Dropping Cabin, Name, and Ticket column	13
Fig xii. Feature Correlation Heatmap	14
Fig xiii. Checking the cleaned dataset	14
Fig xiv. Saving the cleaned dataset	14

## **Exploratory Data Analysis - Titanic Dataset**

### **Introduction**

For this report, the Titanic dataset is used which is widely used for binary classification tasks. The objective is to predict passenger survival based on features like age, gender, and ticket class. In this part, Exploratory Data Analysis (EDA) and feature engineering is performed to prepare the dataset for further tasks regarding neural networks. EDA is important as it helps in analyzing and investigating datasets for summarizing their characteristics through data visualizations (IBM, n.d.).

## Methodology and Findings

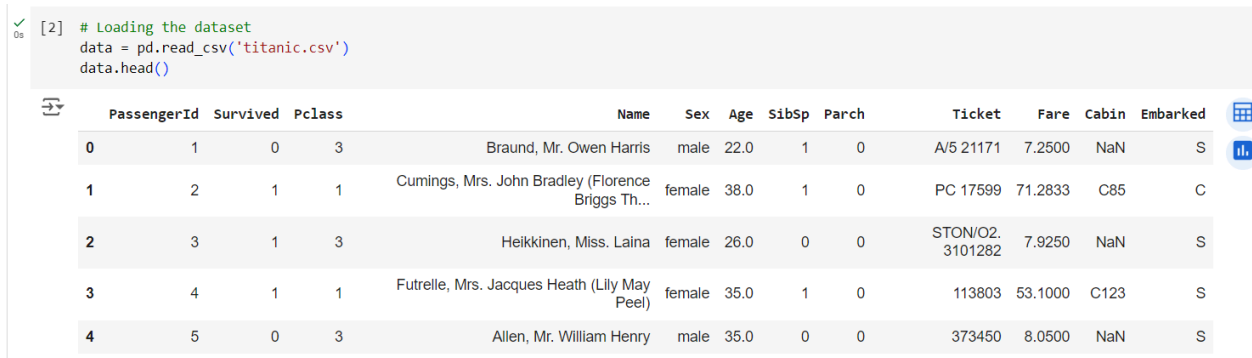


```
# Importing necessary libraries

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

*Fig i. Importing necessary libraries*

Initially, important libraries necessary for the EDA are imported.



```
[2] # Loading the dataset
data = pd.read_csv('titanic.csv')
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

*Fig ii. Loading the dataset and displaying head*

Then, the “titanic.csv” file is read through the use of pandas and the first 5 rows of the dataset are displayed. In the output, we can see the information of passengers like Survived or Survival Status (0 means not survived and 1 means survived), Pclass or Passenger Class (1, 2, and 3), Name, Sex, Age, SibSp (Number of siblings/spouses aboard), Parch (Number of parents/children aboard), Ticket, Fare, Cabin, and Embarked (Port of embarkation).

```

data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   PassengerId   891 non-null    int64  
 1   Survived      891 non-null    int64  
 2   Pclass        891 non-null    int64  
 3   Name          891 non-null    object  
 4   Sex           891 non-null    object  
 5   Age           714 non-null    float64 
 6   SibSp         891 non-null    int64  
 7   Parch         891 non-null    int64  
 8   Ticket        891 non-null    object  
 9   Fare          891 non-null    float64 
10   Cabin         204 non-null    object  
11   Embarked      889 non-null    object  
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB

```

*Fig iii. Information of dataset*

In this part, the summary of the dataset is displayed. It contains 891 records in 12 columns. The data types include integers, floats, and objects.

[4] data.describe()

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

*Fig iv. Statistical summary of numerical columns*

Then, the statistical summary of numerical data in the Titanic dataset is displayed. For example, around 38% of passengers survived (mean of Survived is 0.383838) and the mean age is 30 years.

```

✓ [5] # Performing Exploratory Data Analysis (EDA)

# Checking missing values
data.isnull().sum()

```

	0
<b>PassengerId</b>	0
<b>Survived</b>	0
<b>Pclass</b>	0
<b>Name</b>	0
<b>Sex</b>	0
<b>Age</b>	177
<b>SibSp</b>	0
<b>Parch</b>	0
<b>Ticket</b>	0
<b>Fare</b>	0
<b>Cabin</b>	687
<b>Embarked</b>	2

dtype: int64

*Fig v. Checking missing values*

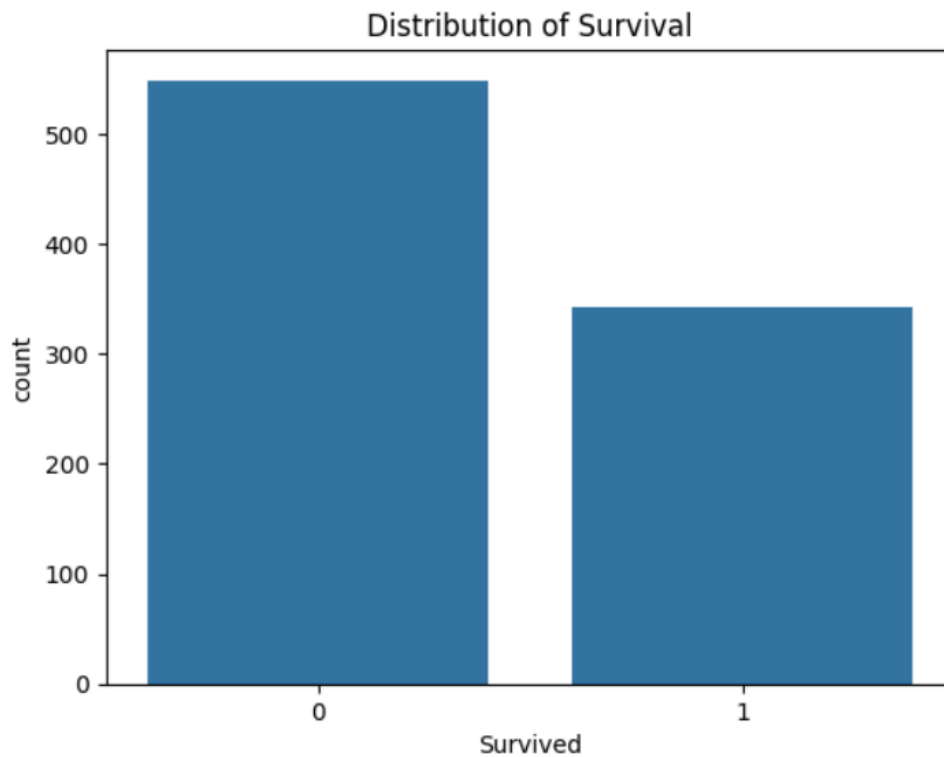
After that, missing values are checked in all columns of the dataset. Most of the columns are complete, however, Age (177), Cabin (687), and Embarked (2) have missing values.



✓  
0s

# Plotting the distribution of the target variable "Survived"

```
sns.countplot(x='Survived', data=data)  
plt.title('Distribution of Survival')  
plt.show()
```

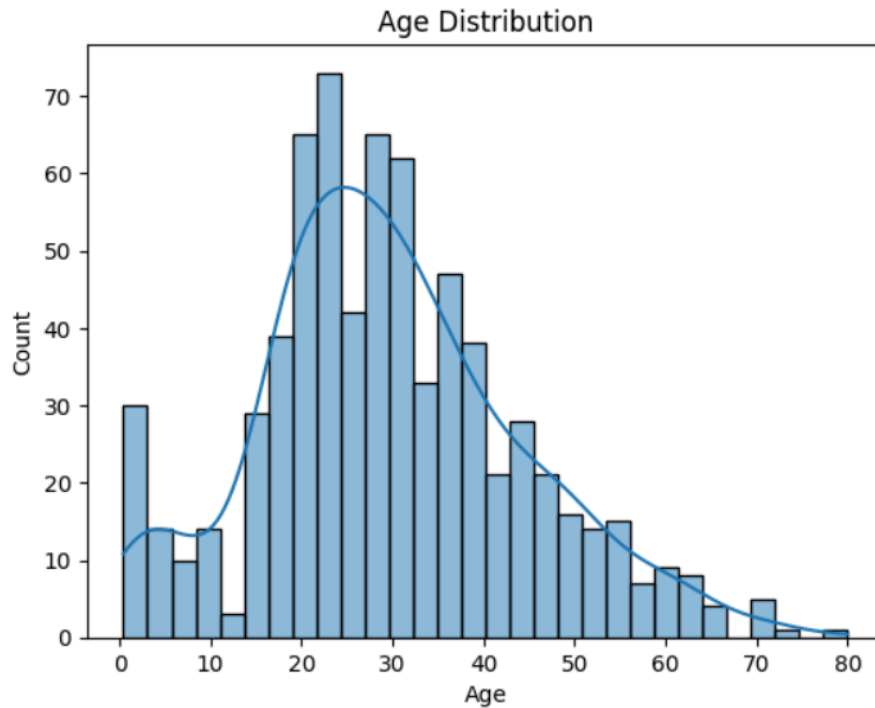


*Fig vi. Column chart to visualize the distribution of survival*

Moreover, the column chart is used to visualize the distribution of survival outcomes on the Titanic. We can see that more people (around 550) did not survive (0) and around 340 passengers survived showing the tragic nature of the accident where most passengers did not survive.

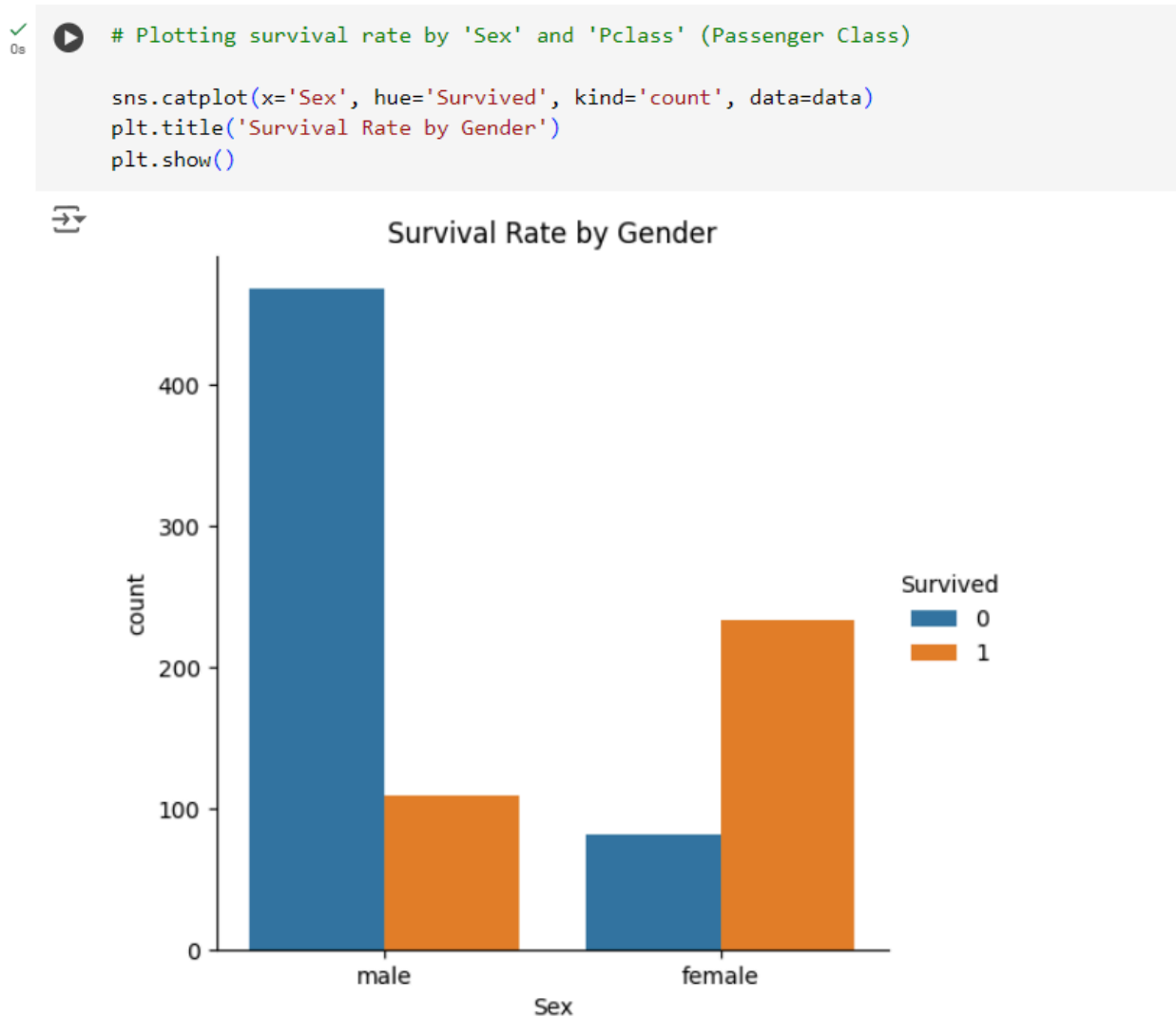
✓  
0s [7] # Plotting the distribution of 'Age' to understand its spread

```
sns.histplot(data['Age'], kde=True, bins=30)
plt.title('Age Distribution')
plt.show()
```



*Fig vii. Distribution of Age*

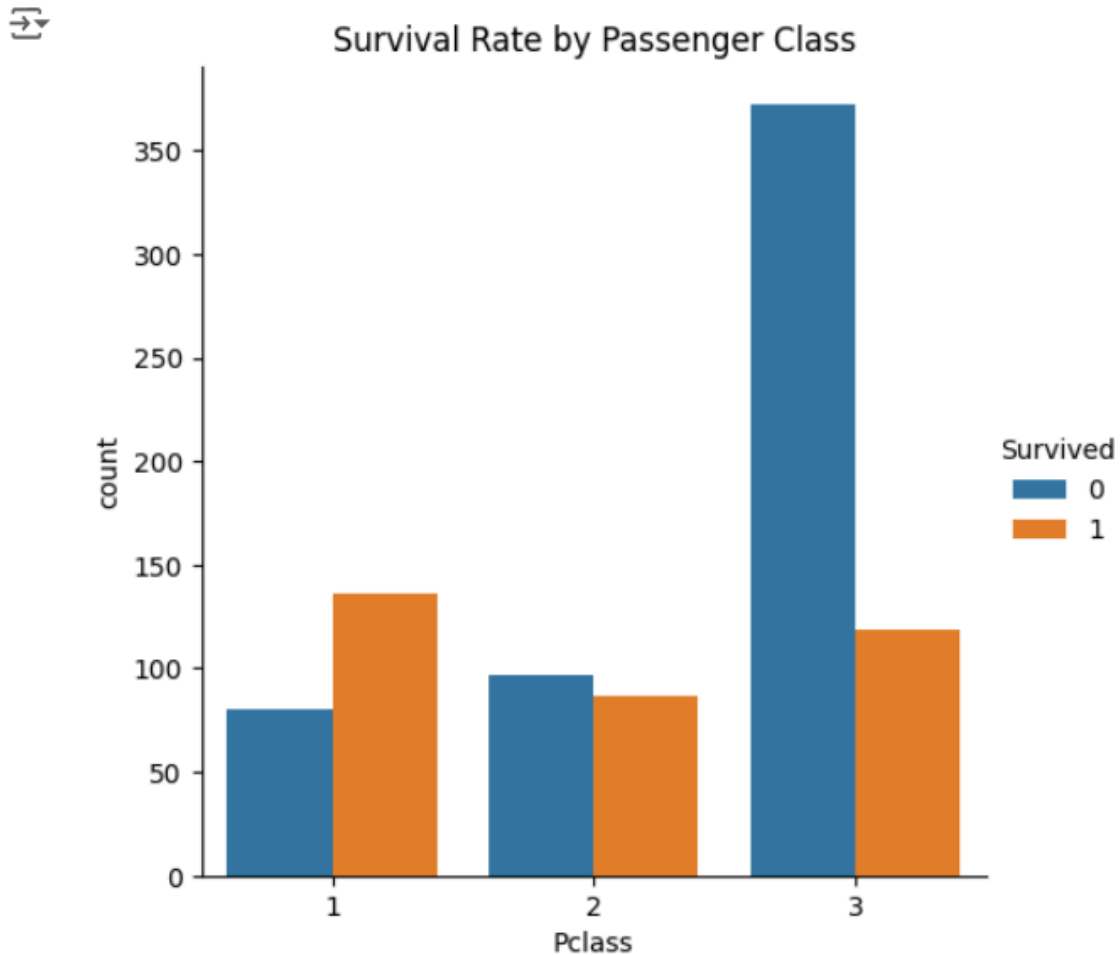
In this part, Age Distribution is visualized through the histogram with a KDE overlay using `sns.histplot()`. Through the visualization, we can see that most of the passengers were young adults from age 20–40 years and peak was from 20–30 years. There were also children from age 0–10 years. The distribution is slightly right skewed with less elderly passengers.



*Fig viii. Survival Rate by Gender*

In this visualization, we can see that most of the male passengers did not survive while female passengers had higher survival rate. Specifically, around 450 male passengers died, around 100 surviving and around 230 female passengers survived with around 80 dying. This shows the “women and children first” protocol which was followed in the accident.

```
[9] sns.catplot(x='Pclass', hue='Survived', kind='count', data=data)
plt.title('Survival Rate by Passenger Class')
plt.show()
```



*Fig ix. Survival Rate by Passenger Class*

Furthermore, we can observe that most of the 1st class passengers survived (around 160) compared to the casualties, 2nd class passengers had roughly equal survival and death rates, and 3rd class passengers had the lowest survival rates with around 370 passengers perishing and only about 120 surviving.

```

# Step 3: Feature Engineering

# Filling missing values for 'Age' with the median

data['Age'].fillna(data['Age'].median(), inplace=True)
data.isnull().sum()

```

<ipython-input-10-0a0e1af957aa>:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or 'df[col] = df[col].method(value)' instead, to perform the operation on the original Dataframe.

```

data['Age'].fillna(data['Age'].median(), inplace=True)

```

PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

dtype: int64

*Fig x. Filling missing values of “Age” with median*

The missing values of Age are filled using median and on checking, we spot no missing values in Age. Median is used because it is less sensitive to outliers than mean (Firdose, 2023).

```

[12] # Dropping 'Cabin' column due to high percentage of missing values and 'Name' and 'Ticket' for simplicity

data = data.drop(['Cabin', 'Name', 'Ticket'], axis=1)

```

```

[13] data

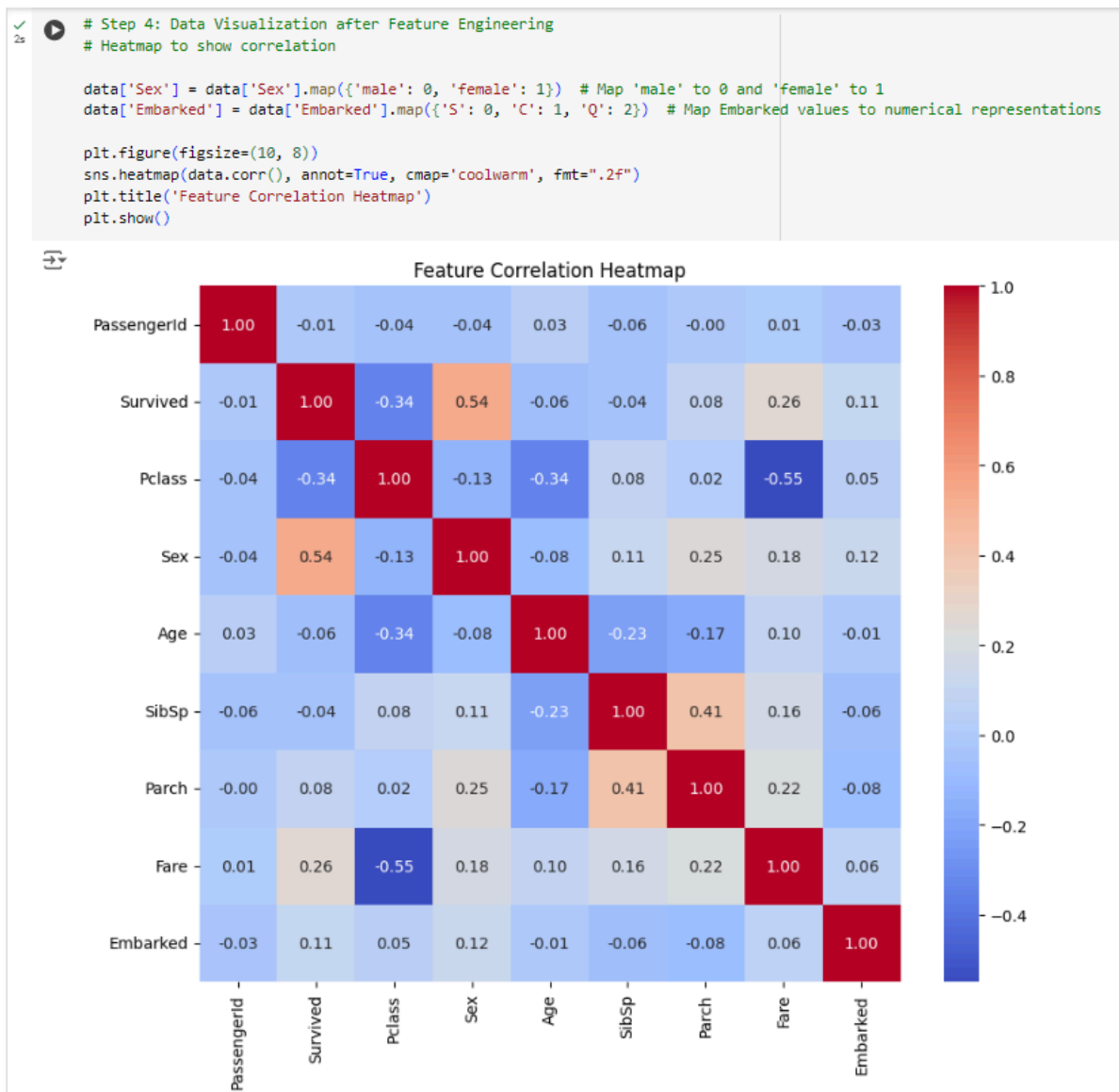
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	S
1	2	1	1	female	38.0	1	0	71.2833	C
2	3	1	3	female	26.0	0	0	7.9250	S
3	4	1	1	female	35.0	1	0	53.1000	S
4	5	0	3	male	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...	...
886	887	0	2	male	27.0	0	0	13.0000	S
887	888	1	1	female	19.0	0	0	30.0000	S
888	889	0	3	female	28.0	1	2	23.4500	S
889	890	1	1	male	26.0	0	0	30.0000	C
890	891	0	3	male	32.0	0	0	7.7500	Q

891 rows × 9 columns

*Fig xi. Dropping Cabin, Name, and Ticket column*

In this part, three columns Cabin (dropped due to too many missing values), Name (dropped for simplicity), and Ticket (dropped for simplicity) are dropped. This makes the dataset cleaner and now 9 columns are left.



*Fig xii. Feature Correlation Heatmap*

In the first two lines of code, the categorical variables Sex and Embarked are encoded into numerical values. Then, the heatmap visualizes how different features in the dataset

correlate with each other. Dark red indicates strong positive correlation (+1.0) while dark blue indicates strong negative correlation (-1.0). For example, “Survived” has a positive correlation with “Sex” (0.54) which suggests that women were more likely to survive.

```

[15] # Checking the cleaned dataset
print(data.head())
print(data.info())

```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	0	22.0	1	0	7.2500	0
1	2	1	1	1	38.0	1	0	71.2833	1
2	3	1	3	1	26.0	0	0	7.9250	0
3	4	1	1	1	35.0	1	0	53.1000	0
4	5	0	3	0	35.0	0	0	8.0500	0

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Sex          891 non-null    int64
4   Age         891 non-null    float64
5   SibSp        891 non-null    int64
6   Parch        891 non-null    int64
7   Fare         891 non-null    float64
8   Embarked     891 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 62.8 KB
None

```

*Fig xiii. Checking the cleaned dataset*

Then, the cleaned dataset is checked with no missing values and dropped columns.

```

[15] data.to_csv('titanic_cleaned.csv', index=False)

```

*Fig xiv. Saving the cleaned dataset*

Finally, the cleaned dataset is saved to “titanic\_cleaned.csv” for further analysis in the upcoming assignment.

**Discussion**

From the EDA, we spot that gender and class had a strong influence on survival rates which shows that socio-economic factors played a huge role in survival of passengers in Titanic. Additionally, imputing missing values preserved the quality of data. Moreover, encoding categorical variables and removing unnecessary columns streamlined the dataset which would be convenient in performing various machine learning models.

**GitHub's Code Link**

[https://github.com/Soniyaa123/EDA\\_Titanic](https://github.com/Soniyaa123/EDA_Titanic)



## References

Firdose, T. (2023, May 28). *Filling missing values with Mean and Median* | by Tahera Firdose | *Medium*. Tahera Firdose.  
<https://tahera-firdose.medium.com/filling-missing-values-with-mean-and-median-76635d55c1bc>

IBM. (n.d.). *What is Exploratory Data Analysis?* IBM.  
<https://www.ibm.com/topics/exploratory-data-analysis>