TASK – 1

Data Cleaning & Preprocessing

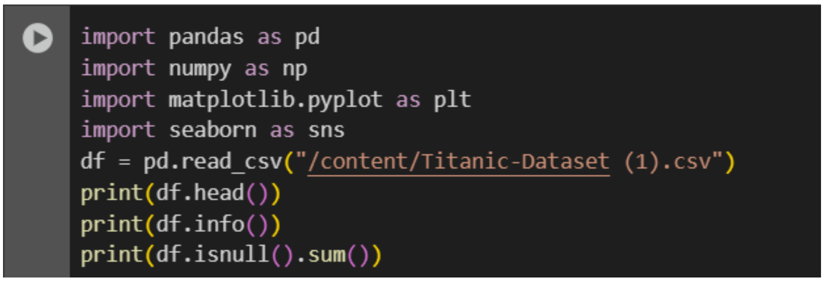
Objective: Learn how to clean and prepare raw data for ML

Tools used: Python, Pandas, NumPy, Matplotlib/Seaborn

Dataset used: Titanic dataset

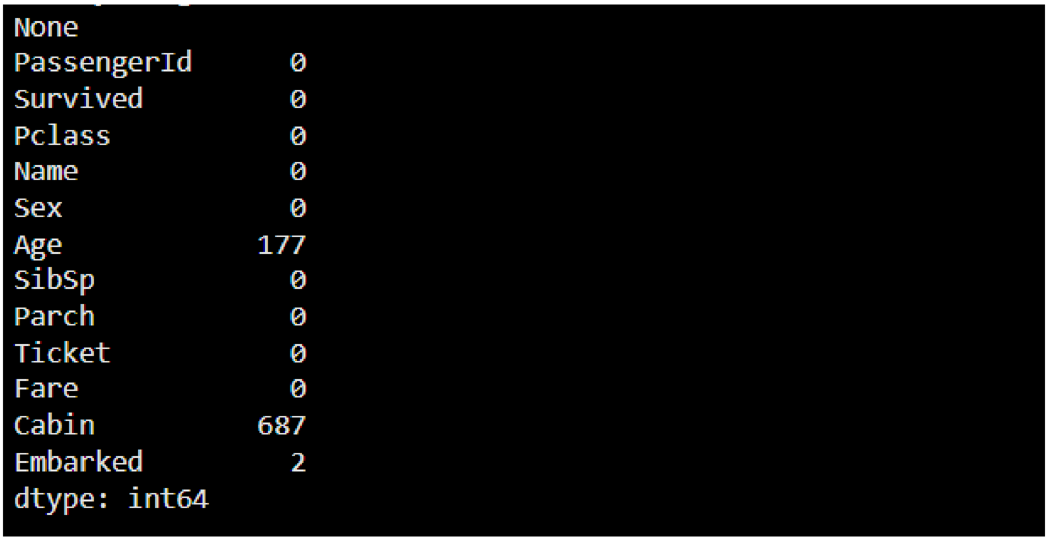
Solution :

Step : 1 – To import the dataset

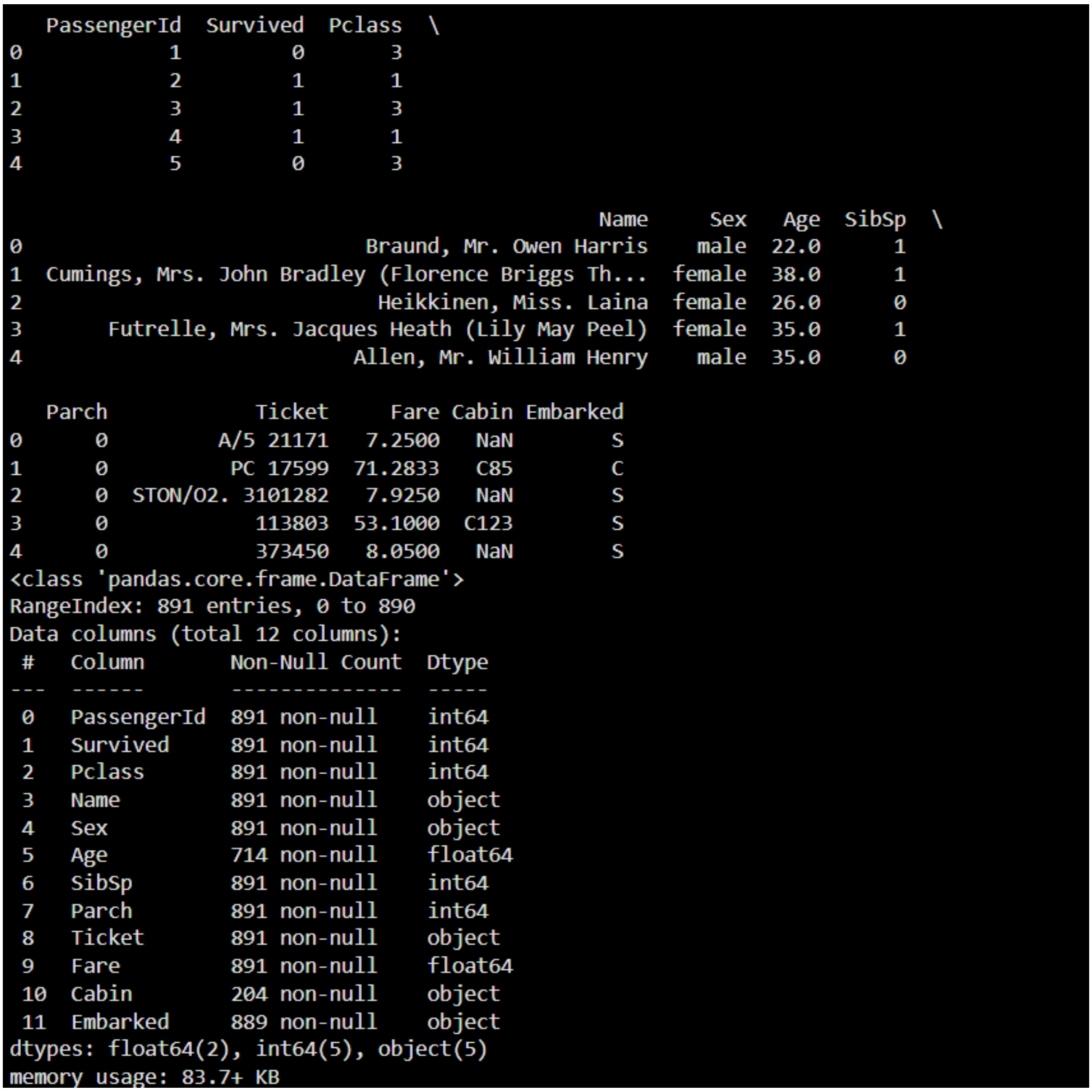


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("/content/Titanic-Dataset (1).csv")
print(df.head())
print(df.info())
print(df.isnull().sum())
```

Output :

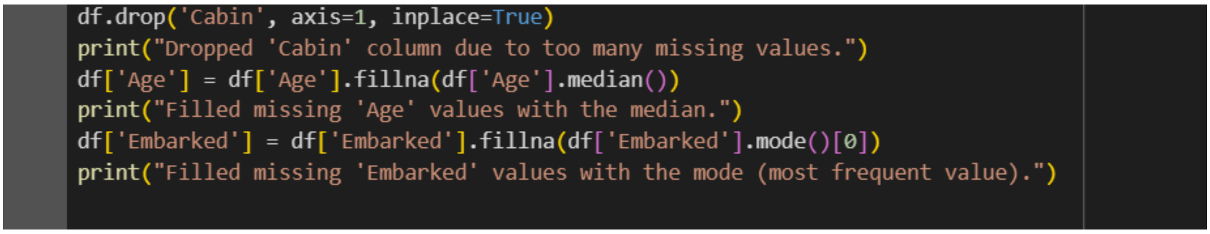


```
None
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```



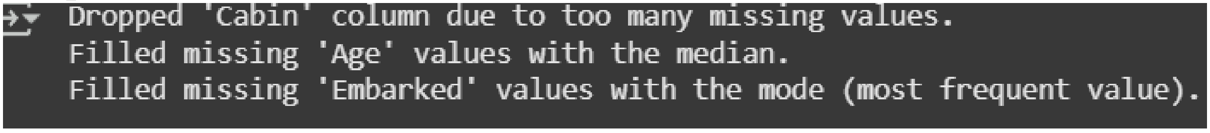
```
   PassengerId  Survived  Pclass  \
0            1         0       3   
1            2         1       1   
2            3         1       3   
3            4         1       1   
4            5         0       3   

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1   
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1   
2                             Heikkinen, Miss. Laina  female  26.0      0   
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1   
4                           Allen, Mr. William Henry    male  35.0      0   

   Parch            Ticket     Fare Cabin Embarked  
0      0         A/5 21171   7.2500   NaN        S  
1      0          PC 17599  71.2833   C85        C  
2      0  STON/O2. 3101282   7.9250   NaN        S  
3      0            113803  53.1000  C123        S  
4      0            373450   8.0500   NaN        S  
<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
```

Step:2 – Handling missing data



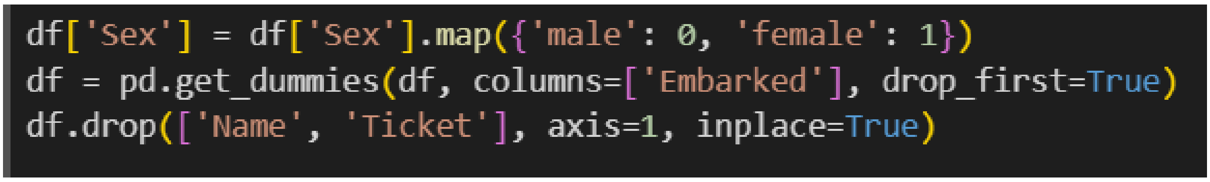
```python
df.drop('Cabin', axis=1, inplace=True)
print("Dropped 'Cabin' column due to too many missing values.")
df['Age'] = df['Age'].fillna(df['Age'].median())
print("Filled missing 'Age' values with the median.")
df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])
print("Filled missing 'Embarked' values with the mode (most frequent value).")
```

Output:



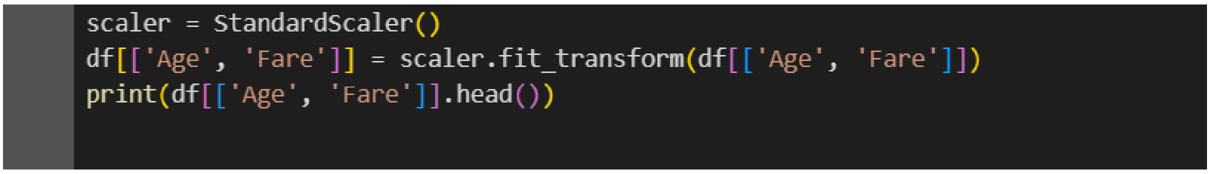
```
Dropped 'Cabin' column due to too many missing values.
Filled missing 'Age' values with the median.
Filled missing 'Embarked' values with the mode (most frequent value).
```

Step:3-Converting categorical features into numerical



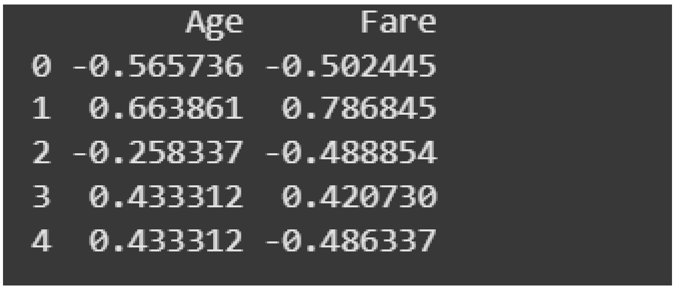
```
df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})
df = pd.get_dummies(df, columns=['Embarked'], drop_first=True)
df.drop(['Name', 'Ticket'], axis=1, inplace=True)
```

Step: 4 – Normalising Numerical feature



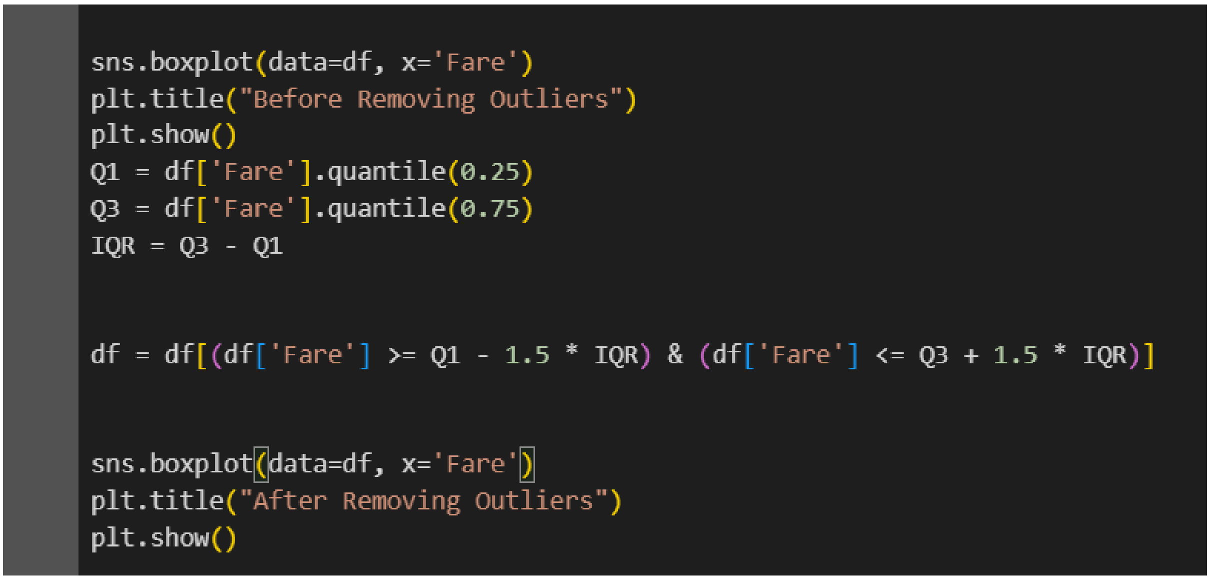
```
scaler = StandardScaler()
df[['Age', 'Fare']] = scaler.fit_transform(df[['Age', 'Fare']])
print(df[['Age', 'Fare']].head())
```

Output :



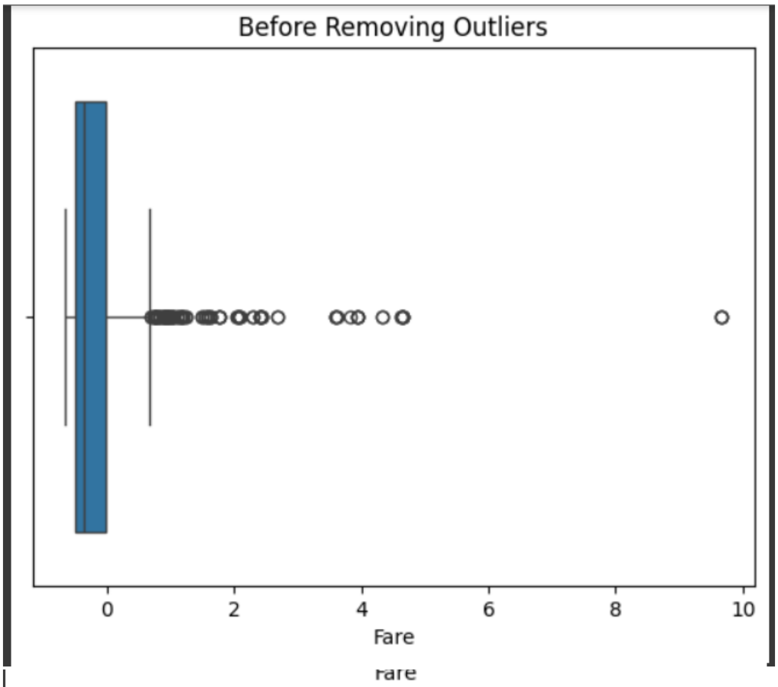
```
        Age      Fare
0 -0.565736 -0.502445
1  0.663861  0.786845
2 -0.258337 -0.488854
3  0.433312  0.420730
4  0.433312 -0.486337
```

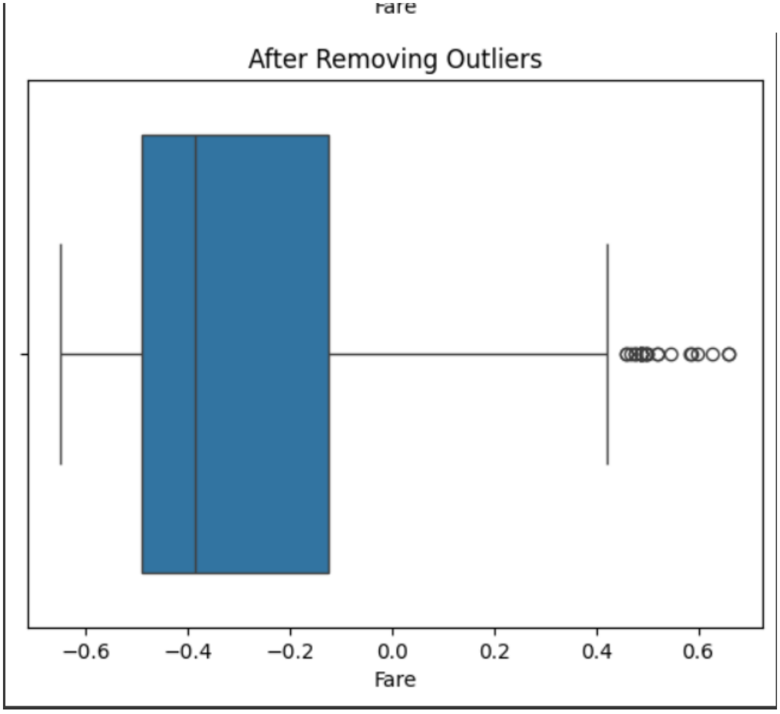
Step: 5 – Visualising



```
sns.boxplot(data=df, x='Fare')
plt.title("Before Removing Outliers")
plt.show()
Q1 = df['Fare'].quantile(0.25)
Q3 = df['Fare'].quantile(0.75)
IQR = Q3 - Q1


df = df[(df['Fare'] >= Q1 - 1.5 * IQR) & (df['Fare'] <= Q3 + 1.5 * IQR)]


sns.boxplot(data=df, x='Fare')
plt.title("After Removing Outliers")
plt.show()
```

Output :


Before Removing Outliers
0
2
4
6
8
10
Fare
Fare


After Removing Outliers
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
Fare

CONCLUSION : Things I learnt in this task :

1. How to read a dataset and check what's missing or wrong in it.
2. How to fix missing data using smart methods like:
   - Filling numbers with the average or middle value
   - Filling categories with the most common option
3. How to change text into numbers, because machines can't read words:
   - Example: "male" → 0, "female" → 1
4. How to scale numbers so they're in the same range (important for ML).
5. How to find and remove outliers (extreme values) using boxplots and the IQR method.