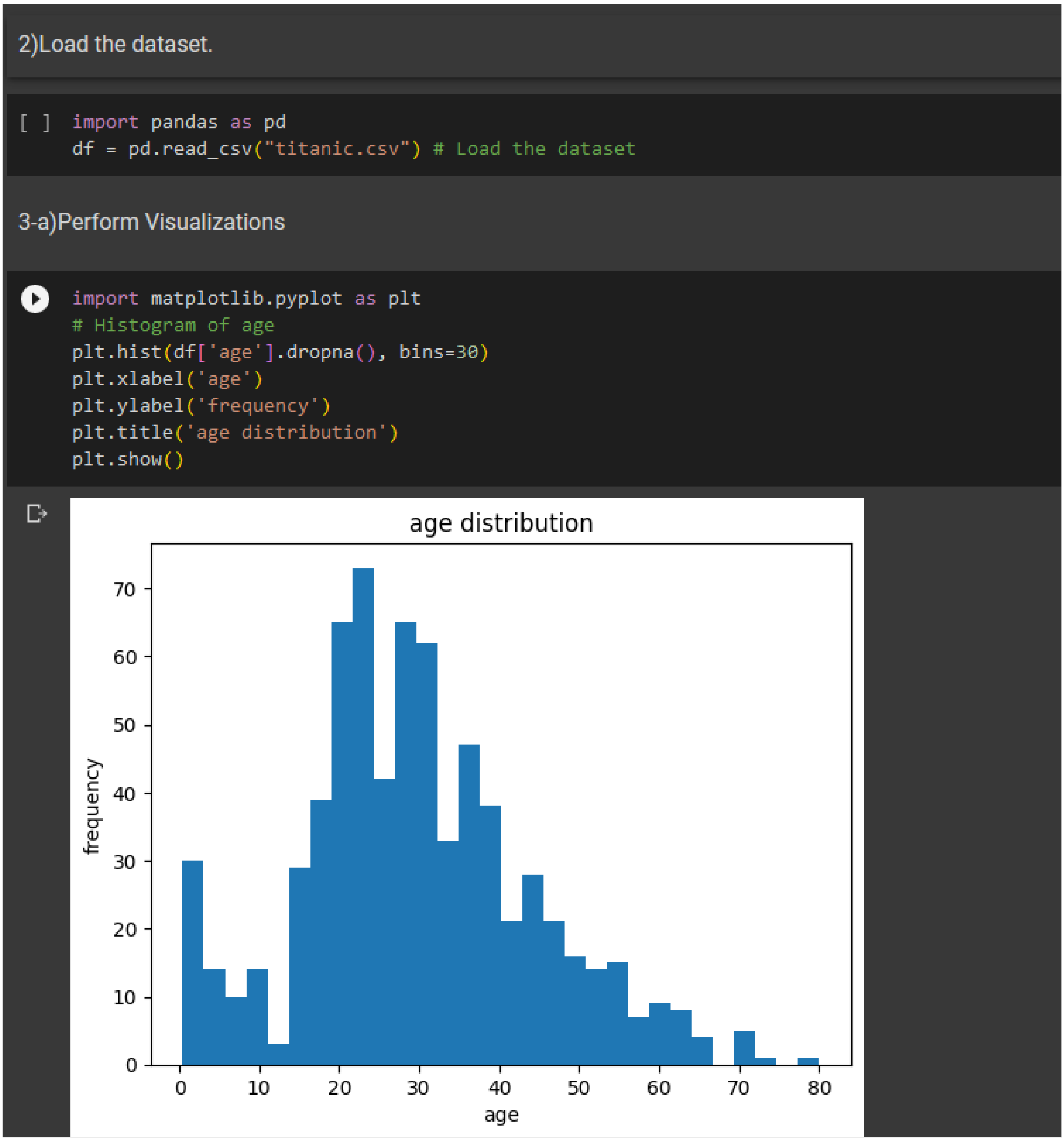
ASSIGNMENT-2

G.SATHWIK

20BCE7644

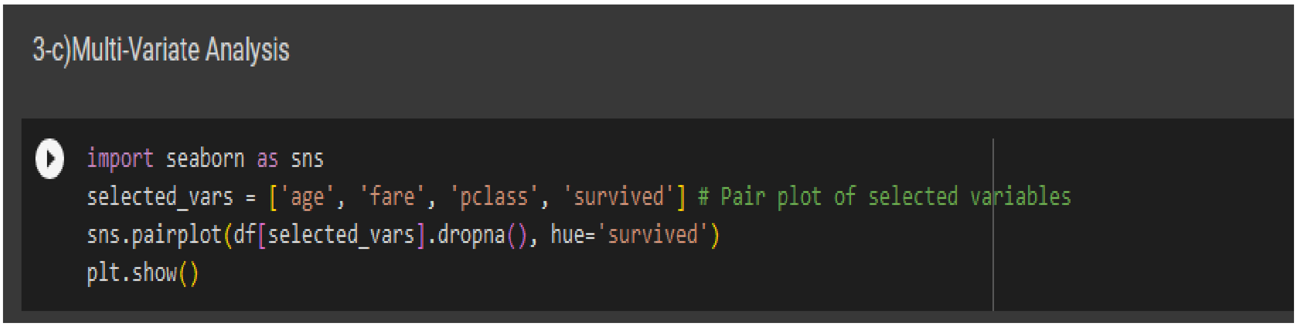


2)Load the dataset.

```
[ ] import pandas as pd
    df = pd.read_csv("titanic.csv") # Load the dataset
```

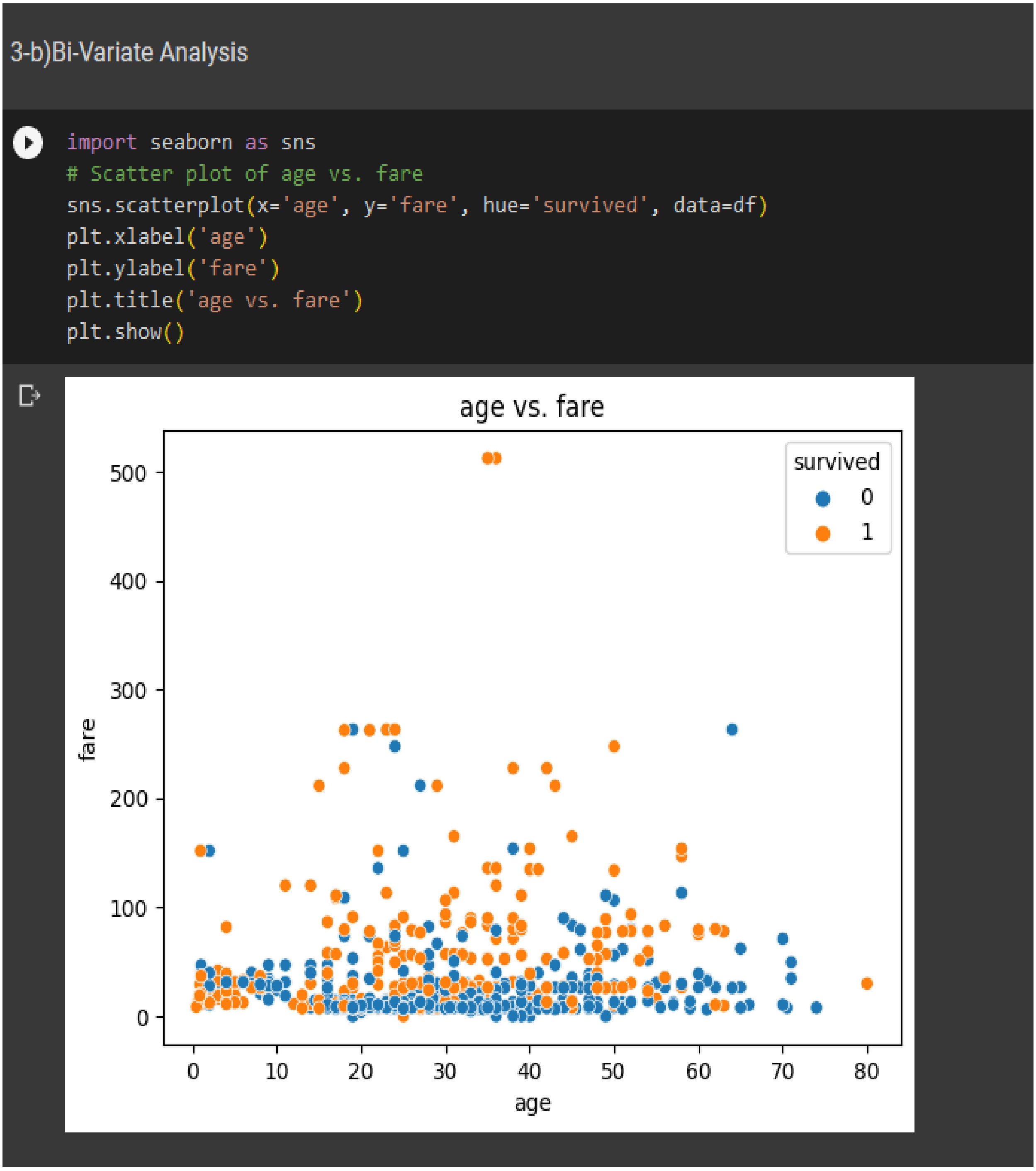
3-a)Perform Visualizations

```
import matplotlib.pyplot as plt
# Histogram of age
plt.hist(df['age'].dropna(), bins=30)
plt.xlabel('age')
plt.ylabel('frequency')
plt.title('age distribution')
plt.show()
```



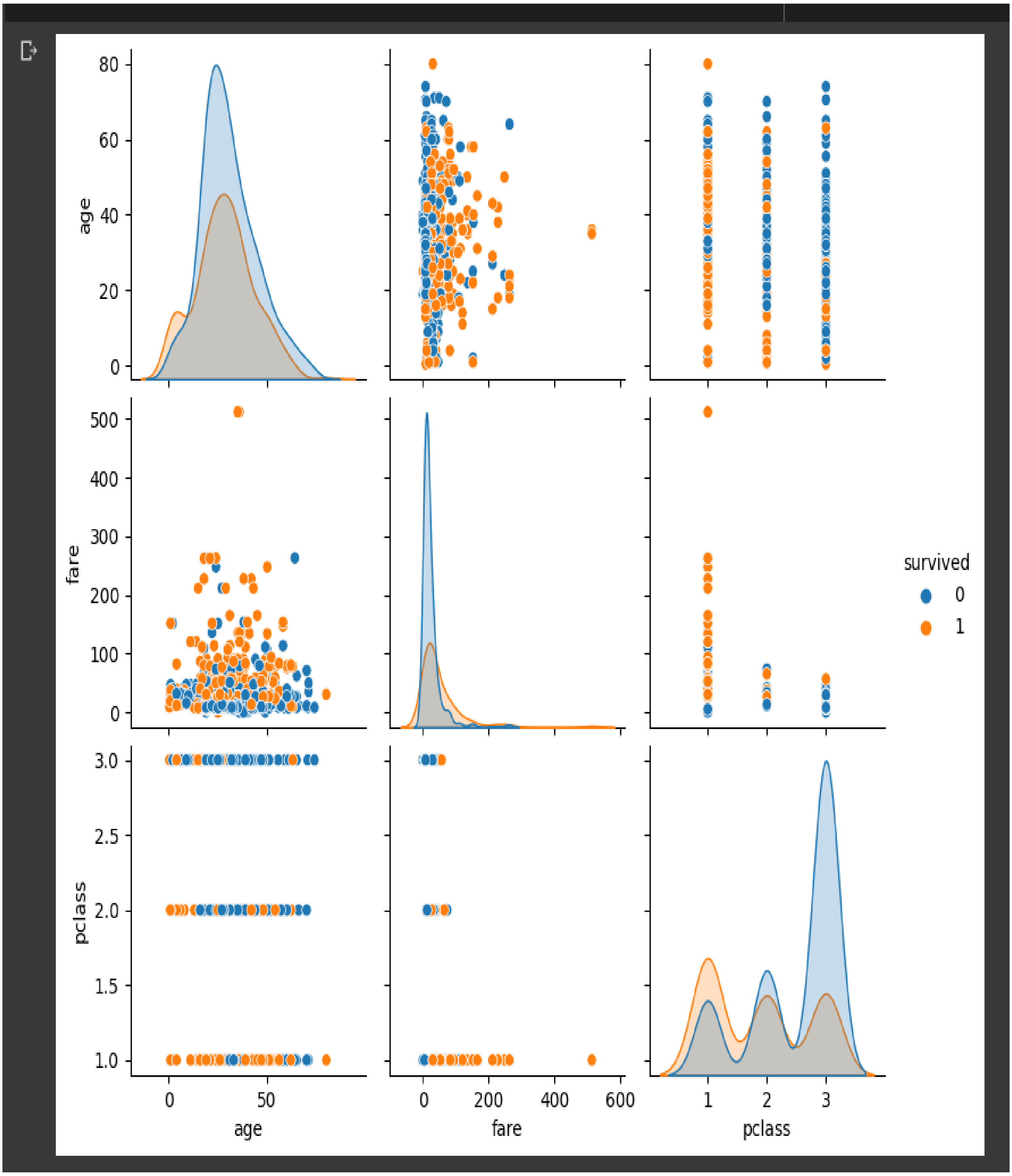
3-c)Multi-Variate Analysis

```
import seaborn as sns
selected_vars = ['age', 'fare', 'pclass', 'survived'] # Pair plot of selected variables
sns.pairplot(df[selected_vars].dropna(), hue='survived')
plt.show()
```

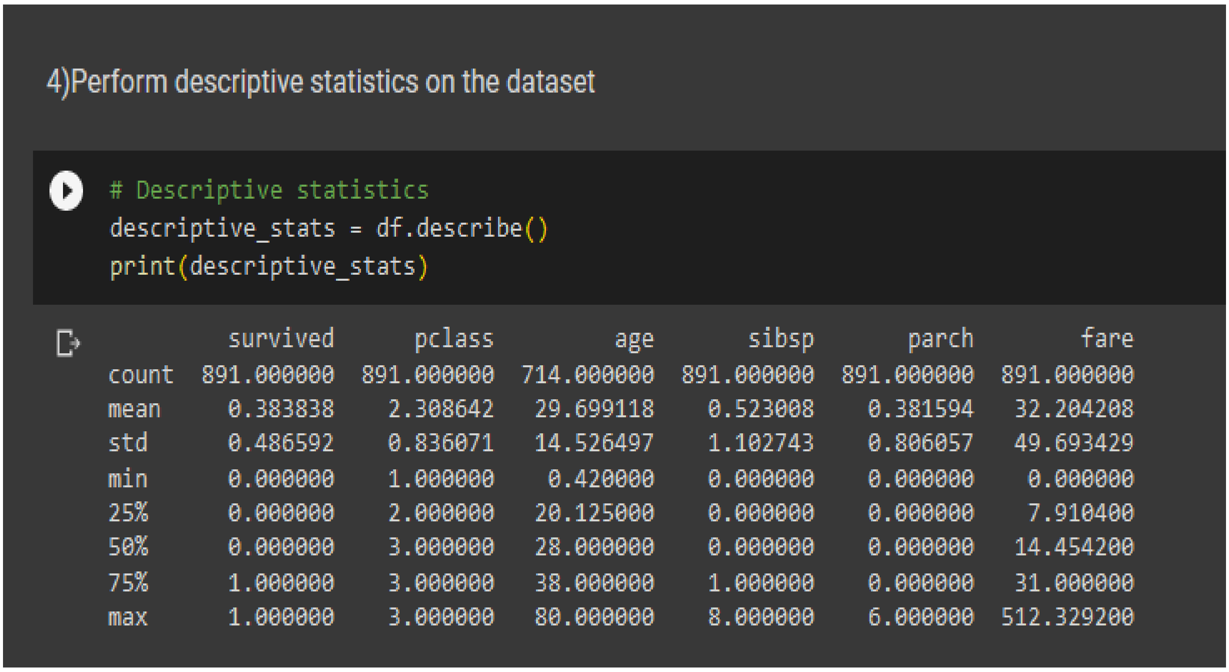


3-b)Bi-Variate Analysis

```
import seaborn as sns
# Scatter plot of age vs. fare
sns.scatterplot(x='age', y='fare', hue='survived', data=df)
plt.xlabel('age')
plt.ylabel('fare')
plt.title('age vs. fare')
plt.show()
```



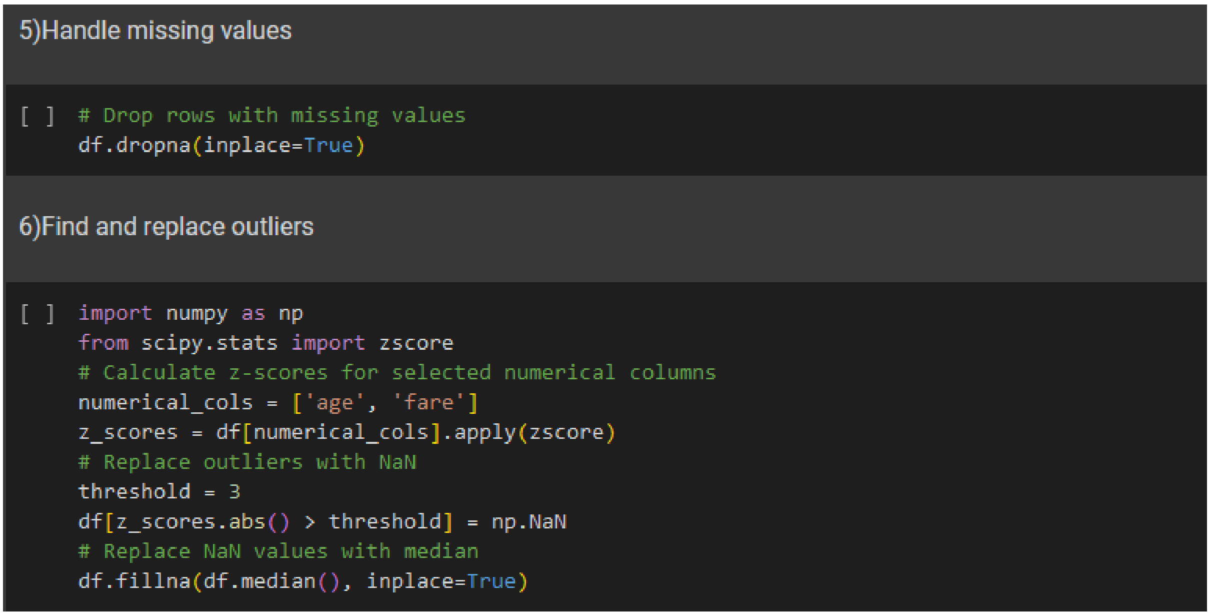
80
60
40
20
0
age
500
400
300
200
100
0
fare
3.0
2.5
2.0
1.5
1.0
pclass
0
50
age
0
200
400
600
fare
1
2
3
pclass
survived
● 0
● 1



4)Perform descriptive statistics on the dataset

```
# Descriptive statistics
descriptive_stats = df.describe()
print(descriptive_stats)
```

```
         survived      pclass         age       sibsp       parch        fare
count  891.000000  891.000000  714.000000  891.000000  891.000000  891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057   49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200
```

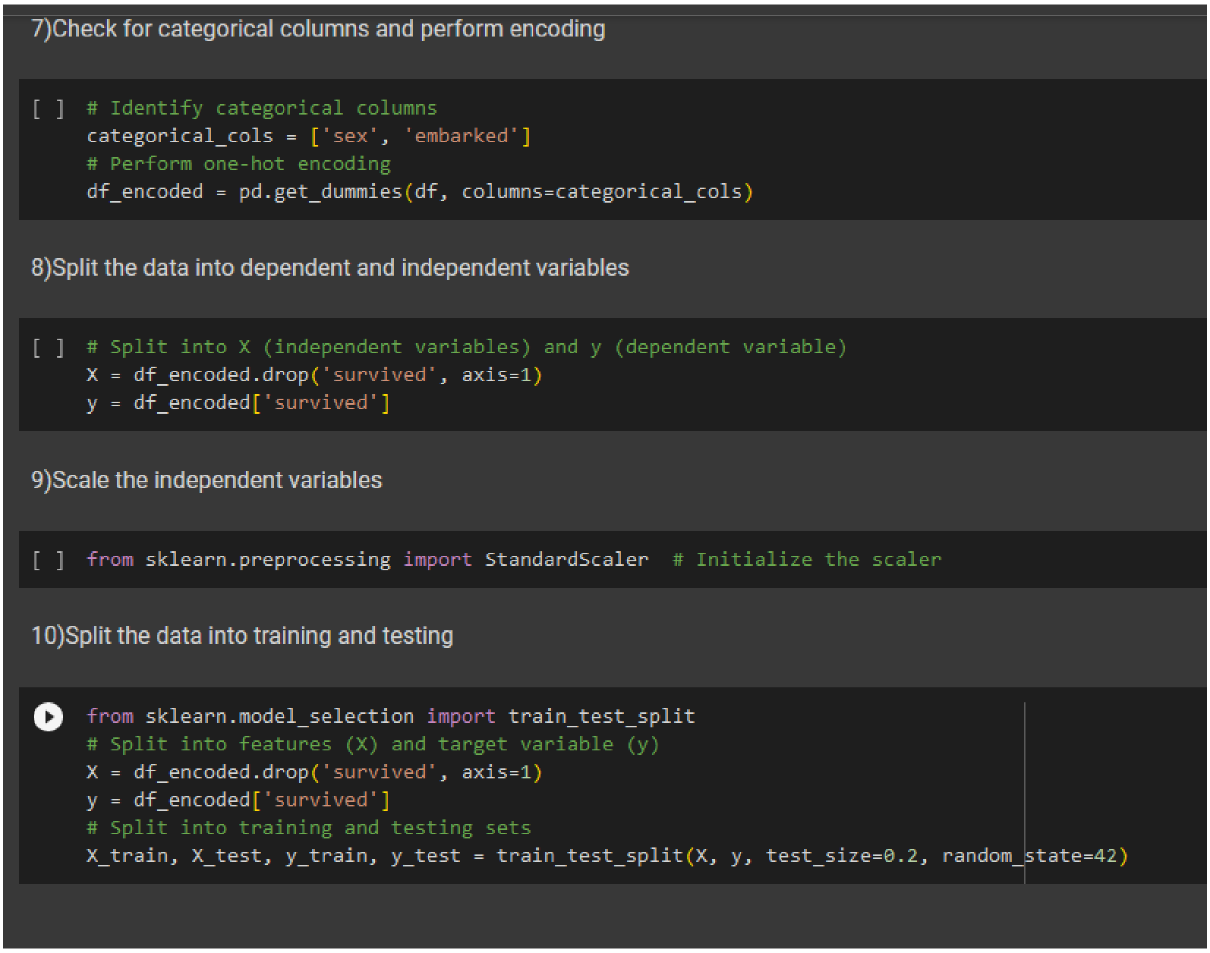


5)Handle missing values

```
[ ]  # Drop rows with missing values
     df.dropna(inplace=True)
```

6)Find and replace outliers

```
[ ]  import numpy as np
     from scipy.stats import zscore
     # Calculate z-scores for selected numerical columns
     numerical_cols = ['age', 'fare']
     z_scores = df[numerical_cols].apply(zscore)
     # Replace outliers with NaN
     threshold = 3
     df[z_scores.abs() > threshold] = np.NaN
     # Replace NaN values with median
     df.fillna(df.median(), inplace=True)
```



7)Check for categorical columns and perform encoding

```
# Identify categorical columns
categorical_cols = ['sex', 'embarked']
# Perform one-hot encoding
df_encoded = pd.get_dummies(df, columns=categorical_cols)
```

8)Split the data into dependent and independent variables

```
# Split into X (independent variables) and y (dependent variable)
X = df_encoded.drop('survived', axis=1)
y = df_encoded['survived']
```

9)Scale the independent variables

```
from sklearn.preprocessing import StandardScaler  # Initialize the scaler
```

10)Split the data into training and testing

```
from sklearn.model_selection import train_test_split
# Split into features (X) and target variable (y)
X = df_encoded.drop('survived', axis=1)
y = df_encoded['survived']
# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```