SMARTBRIDGE EXTERNSHIP (Applied Data Science)-Assignment 2

Name: Bellam Sindhura

Reg no: 20BCB7101

Campus: VIT-AP

1. **Download the dataset: Titanic.csv**
2. **Load the dataset.**

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

# Load the dataset
df = pd.read_csv('titanic.csv')
```

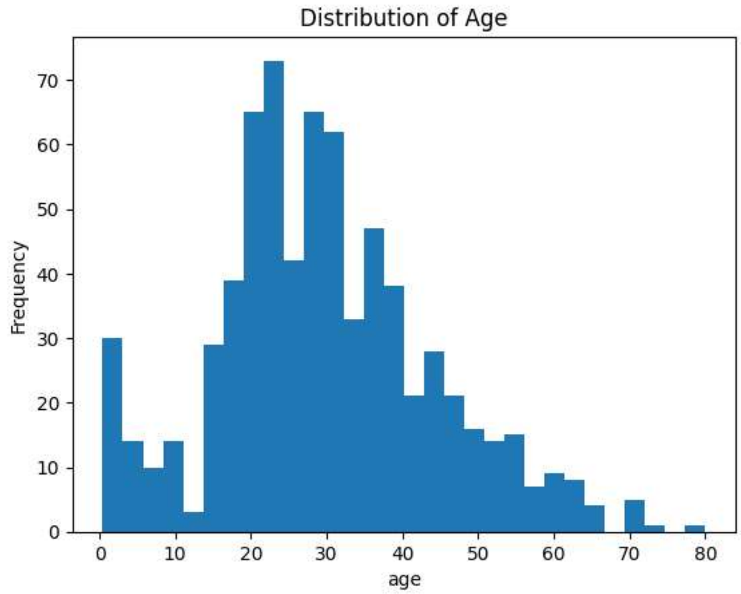
3. **Perform Below visualizations**

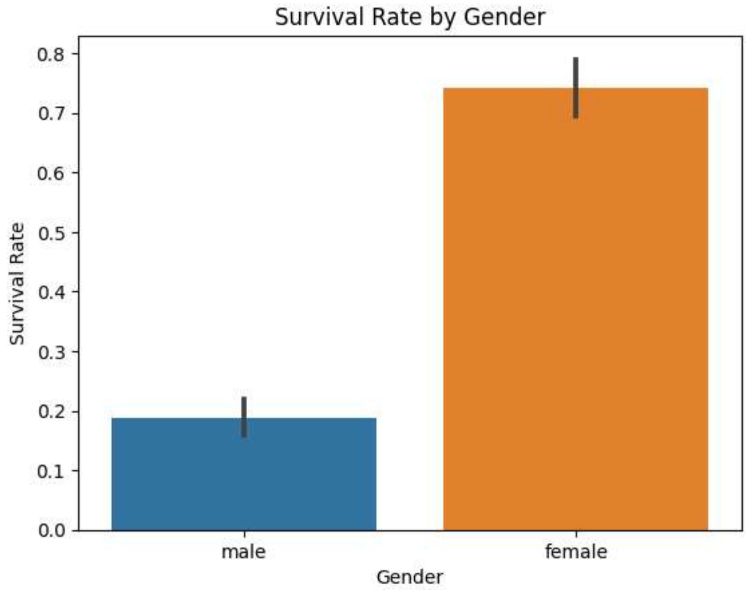
- **Univariate analysis**
- **Bi - variate analysis**
- **Multi-Variate analysis**

```
# Univariate Analysis
# Example: Histogram of Age
plt.hist(df['Age'].dropna(), bins=30)
plt.xlabel('age')
plt.ylabel('Frequency')
plt.title('Distribution of Age')
plt.show()

# Bi-Variate Analysis
# Example: Bar plot of Survival Rate by Gender
sns.barplot(x='Sex', y='Survived', data=df)
plt.xlabel('Gender')
plt.ylabel('Survival Rate')
plt.title('Survival Rate by Gender')
plt.show()

# Multi-Variate Analysis
# Example: Heatmap of Correlations between Variables
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```





```
<ipython-input-4-25f00a6ea1ad>:19: FutureWarning: The default value of numeric_only i
  corr_matrix = df.corr()
```

Correlation Matrix

4) Perform descriptive statistics on the dataset

```
# Calculate descriptive statistics
descriptive_stats = df.describe()

# Display the descriptive statistics
print(descriptive_stats)
```

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
```

```
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

             Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

5) Handle the Mising Values

```
# Impute missing values with the mean of the column
df['Age'].fillna(df['Age'].mean(), inplace=True)

# Impute missing values with the mode of the column
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

6) Find the outliers and replace the outliers

```
import numpy as np
from scipy.stats import zscore

# Calculate z-scores for numerical columns
numeric_columns = ['Age', 'Fare']
z_scores = np.abs(zscore(df[numeric_columns]))

# Set a threshold for identifying outliers
threshold = 3

# Find indices of outliers based on z-scores
outlier_indices = np.where(z_scores > threshold)

# Replace outliers with the median of the column
df[numeric_columns] = np.where(z_scores > threshold, df[numeric_columns].median(), df[numeric_columns])
```

7) Check for Categorical columns and perform encoding

```
# Identify categorical columns
categorical_columns = df.select_dtypes(include='object').columns

# Perform one-hot encoding
encoded_df = pd.get_dummies(df, columns=categorical_columns)

# Display the encoded DataFrame
print(encoded_df)
```

```
     PassengerId  Survived  Pclass        Age  SibSp  Parch     Fare  \
0              1         0       3  22.000000      1      0   7.2500
1              2         1       1  38.000000      1      0  71.2833
2              3         1       3  26.000000      0      0   7.9250
3              4         1       1  35.000000      1      0  53.1000
4              5         0       3  35.000000      0      0   8.0500
..           ...       ...     ...        ...    ...    ...      ...
886          887         0       2  27.000000      0      0  13.0000
887          888         1       1  19.000000      0      0  30.0000
888          889         0       3  29.699118      1      2  23.4500
889          890         1       1  26.000000      0      0  30.0000
890          891         0       3  32.000000      0      0   7.7500

     Name_Abbing, Mr. Anthony  Name_Abbott, Mr. Rossmore Edward  \
0                           0                                 0
1                           0                                 0
2                           0                                 0
3                           0                                 0
4                           0                                 0
..                        ...                               ...
886                         0                                 0
887                         0                                 0
888                         0                                 0
```

```
889                          0                                 0
890                          0                                 0

     Name_Abbott, Mrs. Stanton (Rosa Hunt)  ...  Cabin_F G73  Cabin_F2  \
0                                        0  ...            0         0
1                                        0  ...            0         0
2                                        0  ...            0         0
3                                        0  ...            0         0
4                                        0  ...            0         0
..                                     ...  ...          ...       ...
886                                      0  ...            0         0
887                                      0  ...            0         0
888                                      0  ...            0         0
889                                      0  ...            0         0
890                                      0  ...            0         0

     Cabin_F33  Cabin_F38  Cabin_F4  Cabin_G6  Cabin_T  Embarked_C  \
0            0          0         0         0        0           0
1            0          0         0         0        0           1
2            0          0         0         0        0           0
3            0          0         0         0        0           0
4            0          0         0         0        0           0
..         ...        ...       ...       ...      ...         ...
886          0          0         0         0        0           0
887          0          0         0         0        0           0
888          0          0         0         0        0           0
889          0          0         0         0        0           1
890          0          0         0         0        0           0

     Embarked_Q  Embarked_S
0             0           1
1             0           0
2             0           1
3             0           1
4             0           1
```

8) Split the data into dependent and independent variables

```
# Split into dependent (target) variable and independent variables
X = df.drop('Survived', axis=1)  # Independent variables
y = df['Survived']  # Dependent (target) variable

# Display the independent variables
print(X.head())

# Display the dependent variable
print(y.head())
```

```
   PassengerId  Pclass                                               Name  \
0            1       3                            Braund, Mr. Owen Harris
1            2       1  Cumings, Mrs. John Bradley (Florence Briggs Th...
2            3       3                             Heikkinen, Miss. Laina
3            4       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
4            5       3                           Allen, Mr. William Henry

      Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
0    male  22.0      1      0         A/5 21171   7.2500   NaN        S
1  female  38.0      1      0          PC 17599  71.2833   C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
3  female  35.0      1      0            113803  53.1000  C123        S
4    male  35.0      0      0            373450   8.0500   NaN        S
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

9) Scale the independent variables

```
from sklearn.preprocessing import StandardScaler

# Perform one-hot encoding on categorical variables
X_encoded = pd.get_dummies(X)

# Perform scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_encoded)
```

```
# Display the scaled independent variables
scaled_df = pd.DataFrame(X_scaled, columns=X_encoded.columns)
print(scaled_df.head())
```

```
   PassengerId    Pclass       Age     SibSp     Parch      Fare  \
0    -1.730108  0.827377 -0.592704  0.432793 -0.473674 -0.654170
1    -1.726220 -1.566107  0.695087  0.432793 -0.473674  1.549441
2    -1.722332  0.827377 -0.270757 -0.474545 -0.473674 -0.630941
3    -1.718444 -1.566107  0.453626  0.432793 -0.473674  0.923690
4    -1.714556  0.827377  0.453626 -0.474545 -0.473674 -0.626639

   Name_Abbing, Mr. Anthony  Name_Abbott, Mr. Rossmore Edward  \
0                  -0.03352                          -0.03352
1                  -0.03352                          -0.03352
2                  -0.03352                          -0.03352
3                  -0.03352                          -0.03352
4                  -0.03352                          -0.03352

   Name_Abbott, Mrs. Stanton (Rosa Hunt)  Name_Abelson, Mr. Samuel  ...  \
0                               -0.03352                  -0.03352  ...
1                               -0.03352                  -0.03352  ...
2                               -0.03352                  -0.03352  ...
3                               -0.03352                  -0.03352  ...
4                               -0.03352                  -0.03352  ...

   Cabin_F G73  Cabin_F2  Cabin_F33  Cabin_F38  Cabin_F4  Cabin_G6  Cabin_T  \
0    -0.047431 -0.058124  -0.058124   -0.03352 -0.047431 -0.067153 -0.03352
1    -0.047431 -0.058124  -0.058124   -0.03352 -0.047431 -0.067153 -0.03352
2    -0.047431 -0.058124  -0.058124   -0.03352 -0.047431 -0.067153 -0.03352
3    -0.047431 -0.058124  -0.058124   -0.03352 -0.047431 -0.067153 -0.03352
4    -0.047431 -0.058124  -0.058124   -0.03352 -0.047431 -0.067153 -0.03352

   Embarked_C  Embarked_Q  Embarked_S
0   -0.482043   -0.307562    0.615838
1    2.074505   -0.307562   -1.623803
2   -0.482043   -0.307562    0.615838
3   -0.482043   -0.307562    0.615838
4   -0.482043   -0.307562    0.615838

[5 rows x 1730 columns]
```

10)Split the data into training and testing

```
from sklearn.model_selection import train_test_split

# Split the data 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)

# Display the shapes of the subsets
print("Training set shape:", X_train.shape, y_train.shape)
print("Testing set shape:", X_test.shape, y_test.shape)
```

```
Training set shape: (712, 11) (712,)
Testing set shape: (179, 11) (179,)
```