HOME WORK AND LAB

TITANIC DATASET EDA AND FEATURE ENGINEERING

Homework & LAB: Titanic Dataset – EDA and Feature Engineering

Task Overview

- The Titanic dataset is a classic dataset used to predict if a passenger survived the shipwreck. Before building a predictive model, you will perform data exploration and feature preparation steps.

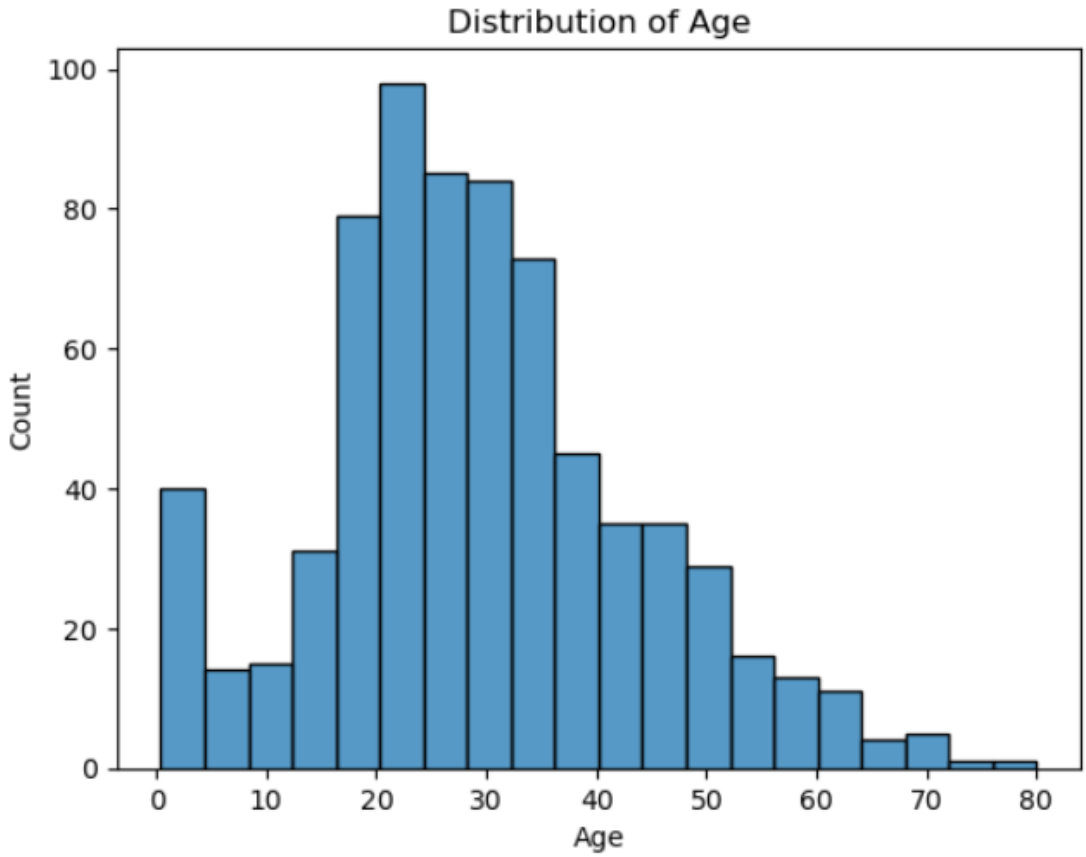
Tasks

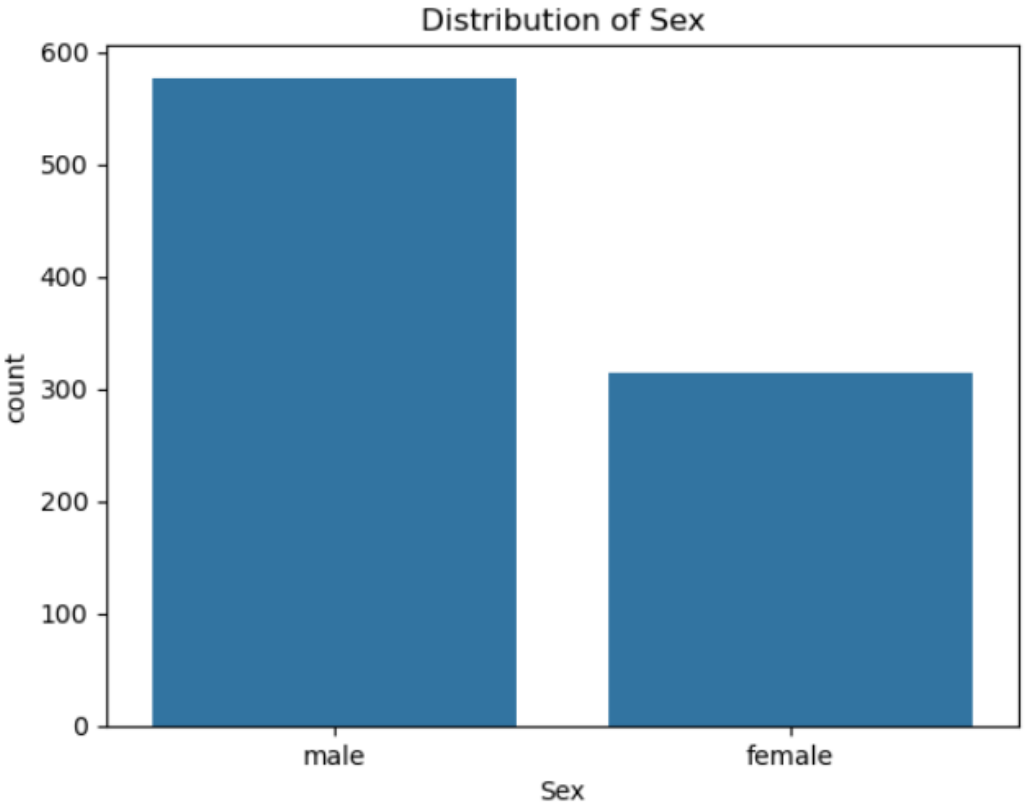
- Exploratory Data Analysis (EDA)
  - Load the Titanic dataset.
  - Summarize missing values and data types.
  - Visualize distributions of key features such as Age, Sex, Pclass, Fare, and Embarked.
  - Analyze relationships between features and survival rates (e.g., survival by Sex, Pclass).
- Data Cleaning and Imputation
  - Handle missing values for Age, Embarked, and Fare.
  - Drop irrelevant columns like PassengerId, Name, Ticket, and Cabin if needed.
- Feature Engineering
  - Create a new feature `FamilySize` by adding `SibSp` and `Parch`.
  - Extract Titles from the Name feature (e.g., Mr, Mrs) as a new categorical feature.
  - Convert categorical features (Sex, Embarked, Title) into numeric using one-hot encoding or label encoding.
- Prepare Data for Modeling
  - Finalize features and split data into training and test sets.
  - Check data readiness for model training.

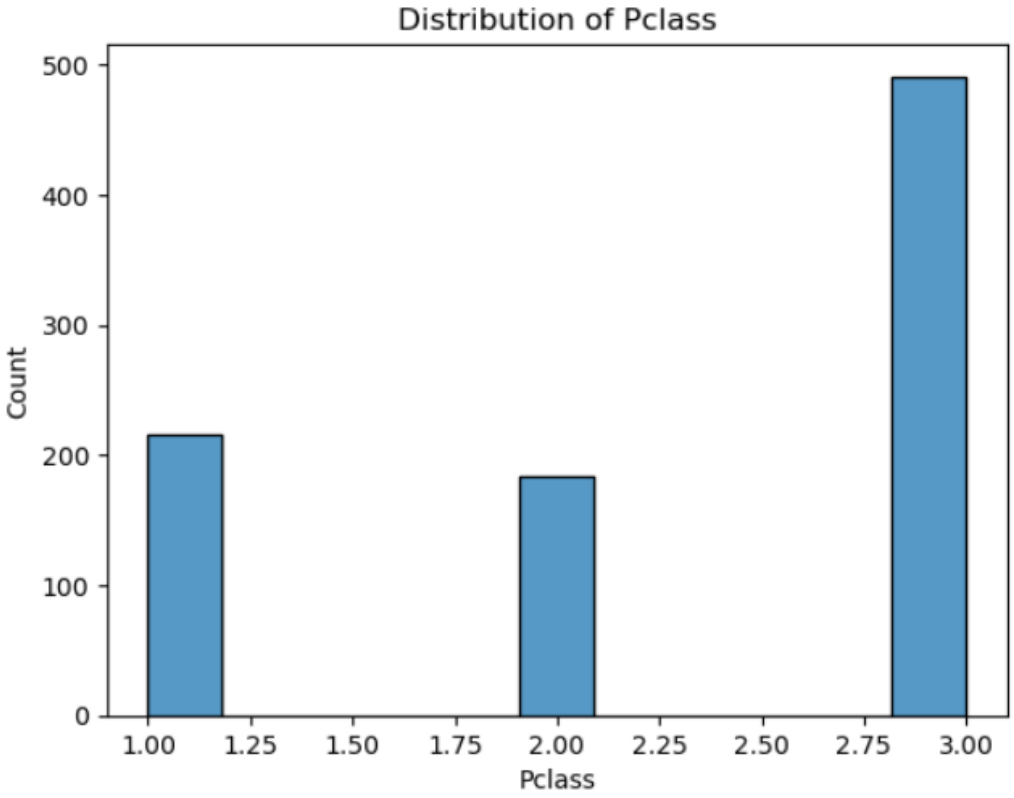
[1]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

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

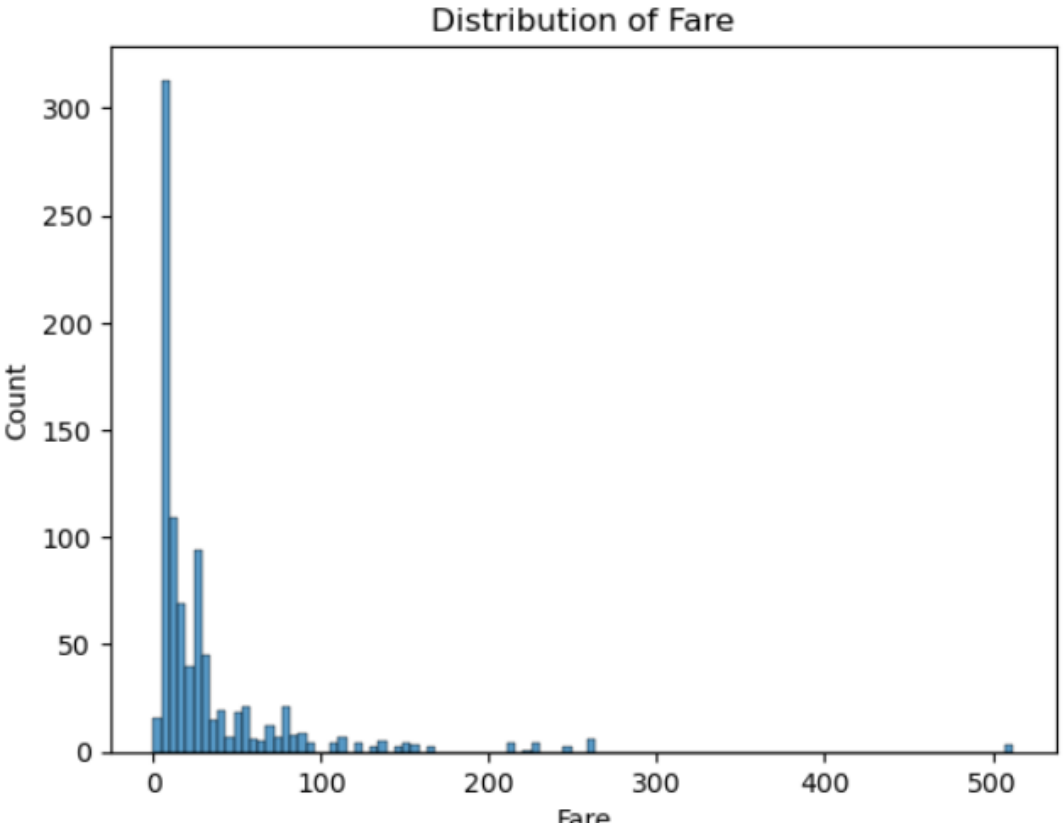
[2]:
```python
# Display data info and missing value counts
print(df.info())
print(df.isnull().sum())

# Optional: Visualize feature distributions
for feature in ['Age', 'Sex', 'Pclass', 'Fare', 'Embarked']:
    plt.figure()
    if df[feature].dtype == 'O':
        sns.countplot(x=feature, data=df)
    else:
        sns.histplot(df[feature].dropna(), kde=False)
    plt.title(f'Distribution of {feature}')
    plt.show()

# Optional: Analyze relationships with survival
for feature in ['Sex', 'Pclass', 'Embarked']:
    pd.crosstab(df[feature], df['Survived'], normalize='index').plot(kind='bar')
    plt.title(f'Survival Rate by {feature}')
    plt.show()
```

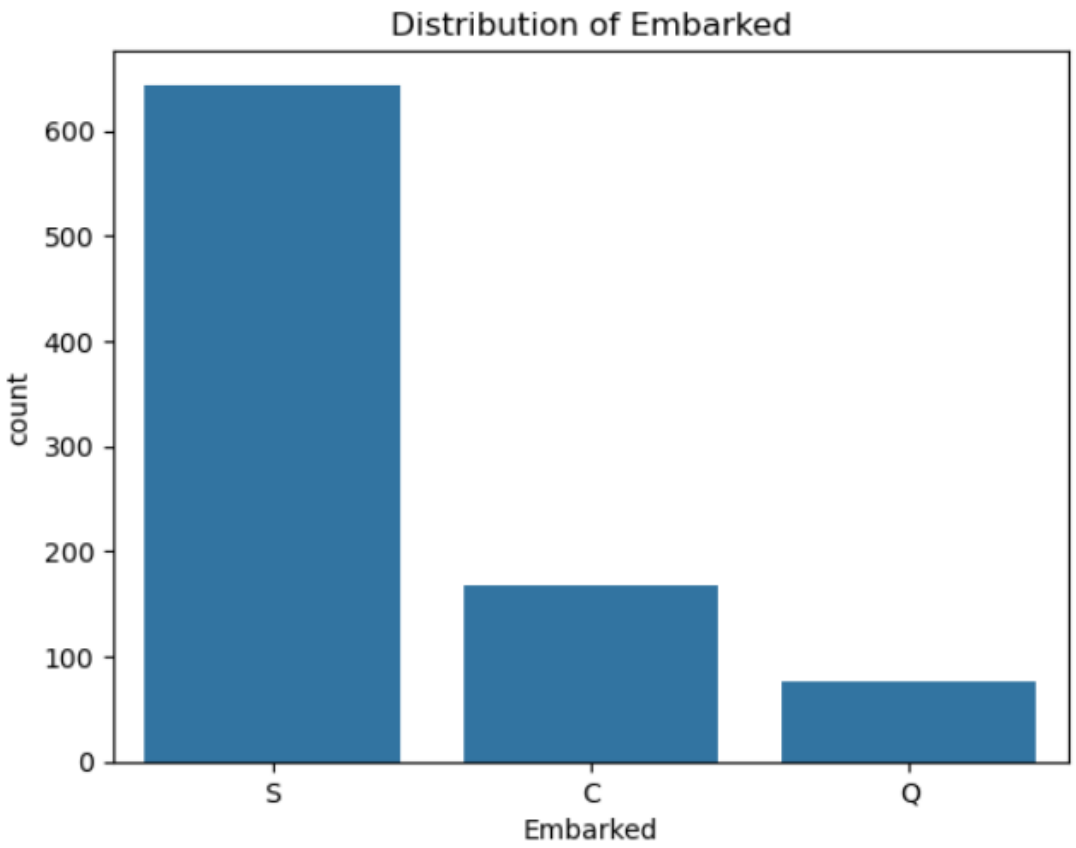
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    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
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
```

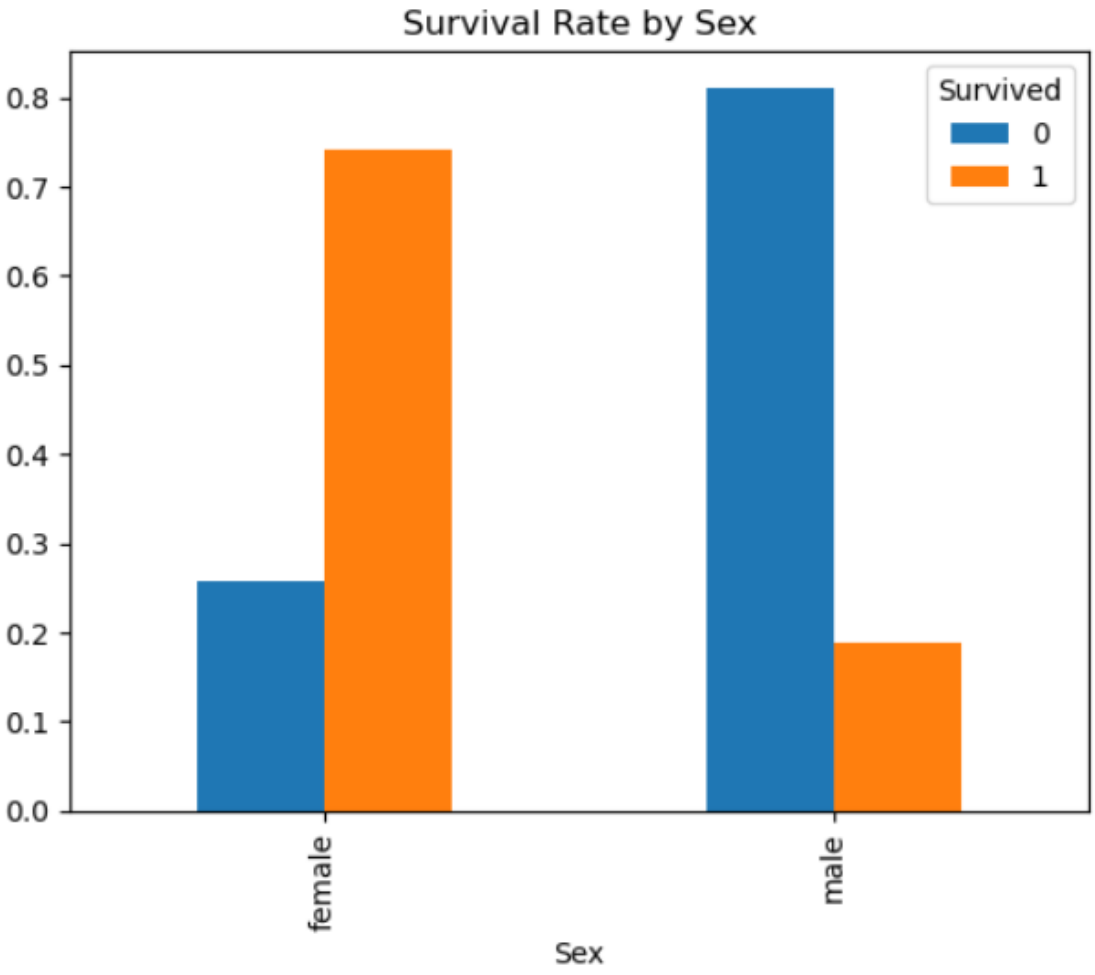

Distribution of Age
Count
Age
0
20
40
60
80
100
0
10
20
30
40
50
60
70
80

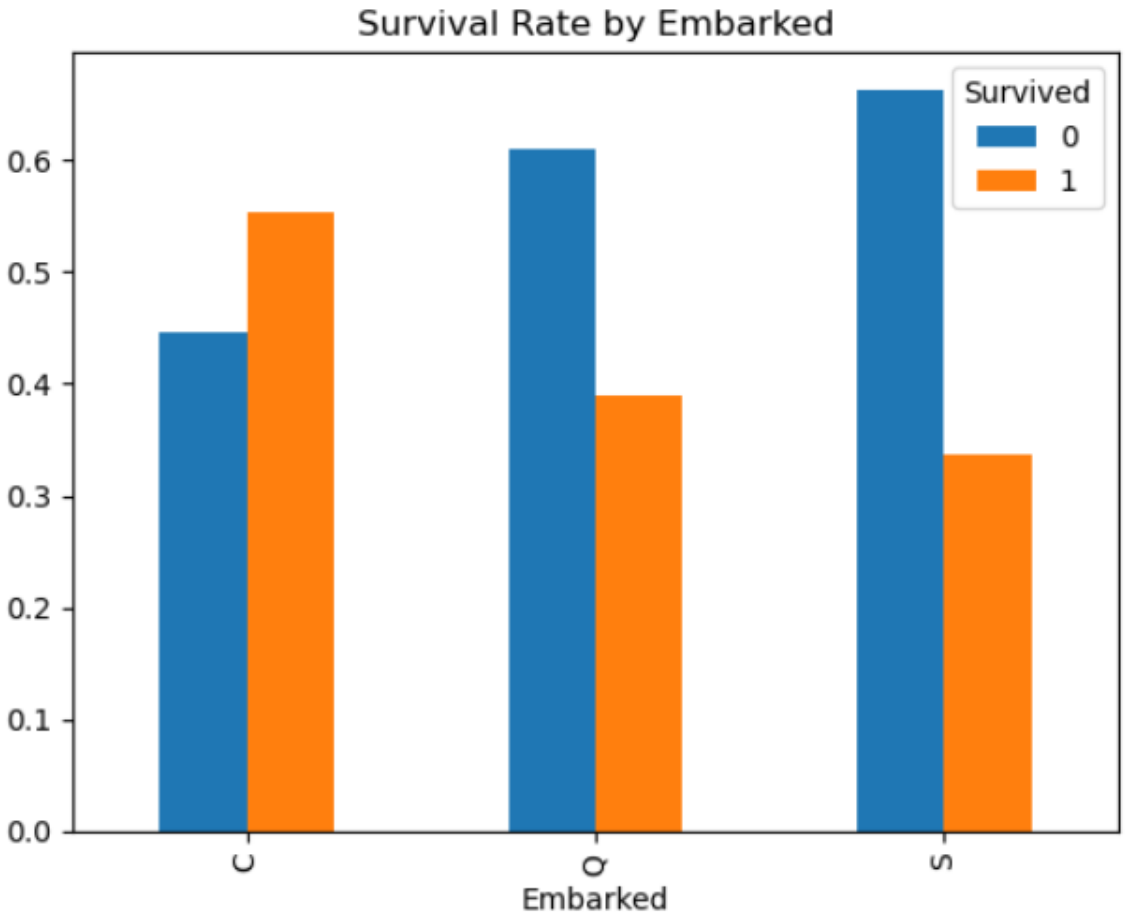

Distribution of Sex
count
Sex
0
100
200
300
400
500
600
male
female

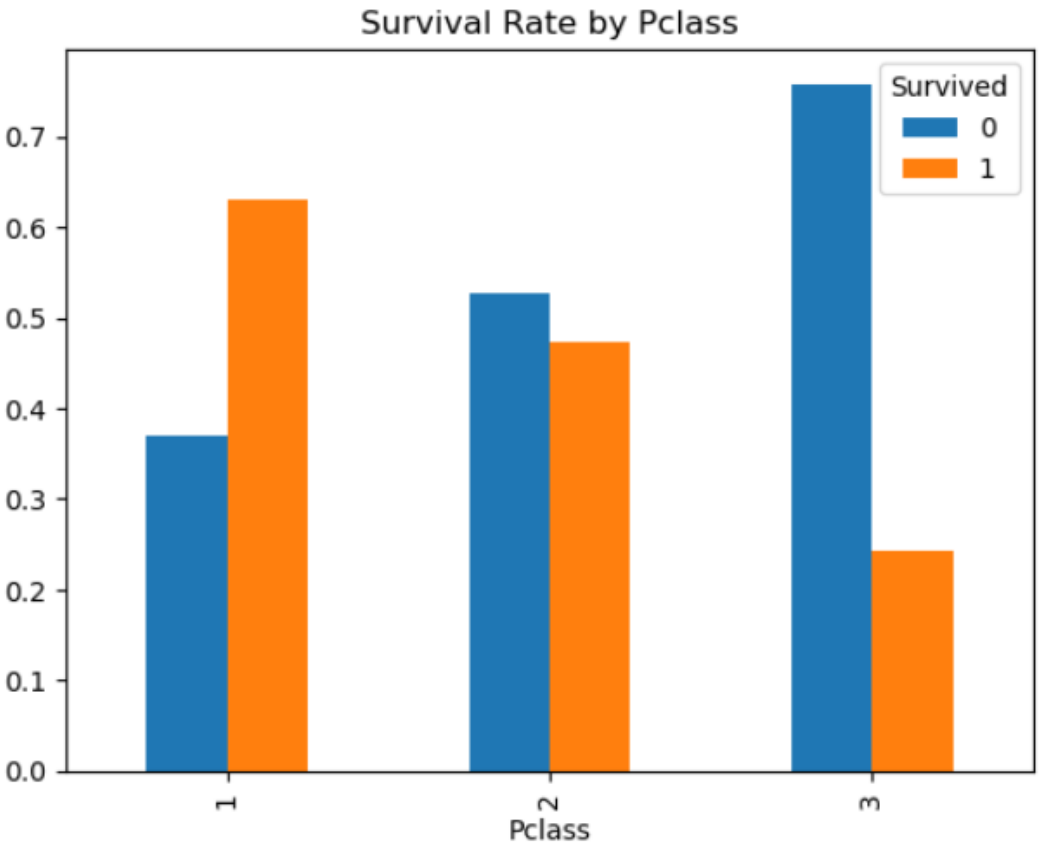

Distribution of Pclass
Count
Pclass
0
100
200
300
400
500
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00


Distribution of Fare
Count
Fare
0
50
100
150
200
250
300
0
100
200
300
400
500


Distribution of Embarked
count
600
500
400
300
200
100
0
S
C
Q
Embarked


Survival Rate by Sex
Survived
0
1
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
female
male
Sex


Survival Rate by Embarked
Survived
0
1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
C
Q
S
Embarked


Survival Rate by Pclass
Survived
0
1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
1
2
3
Pclass

[3]:
```
# Fill missing Age with median
df['Age'] = df['Age'].fillna(df['Age'].median())

# Fill missing Embarked with most frequent value
df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])

# Fill missing Fare with median
df['Fare'] = df['Fare'].fillna(df['Fare'].median())

# Drop columns not needed for modeling
df = df.drop(['PassengerId', 'Ticket', 'Cabin'], axis=1)
```

[4]:
```
# Create FamilySize feature
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1

# Extract Title from Name
df['Title'] = df['Name'].str.extract(r' ([A-Za-z]+)\.', expand=False)

# Drop Name column after extracting Title
df = df.drop(['Name'], axis=1)
```

[5]:
```
# Encode Sex using LabelEncoder
df['Sex'] = LabelEncoder().fit_transform(df['Sex'])

# One-hot encode Embarked and Title
df = pd.get_dummies(df, columns=['Embarked', 'Title'], drop_first=True)
```

[6]:
```
# Set features and target
X = df.drop('Survived', axis=1)
y = df['Survived']

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Check readiness
print(X_train.head())
print(y_train.head())
```

```
     Pclass  Sex   Age  SibSp  Parch     Fare  FamilySize  Embarked_Q  \
331       1    1  45.5      0      0  28.5000           1       False
733       2    1  23.0      0      0  13.0000           1       False
382       3    1  32.0      0      0   7.9250           1       False
704       3    1  26.0      1      0   7.8542           2       False
813       3    0   6.0      4      2  31.2750           7       False

     Embarked_S  Title_Col  ...  Title_Major  Title_Master  Title_Miss  \
331        True      False  ...        False         False       False
733        True      False  ...        False         False       False
382        True      False  ...        False         False       False
704        True      False  ...        False         False       False
813        True      False  ...        False         False        True

     Title_Mlle  Title_Mme  Title_Mr  Title_Mrs  Title_Ms  Title_Rev  \
331       False      False      True      False     False      False
733       False      False      True      False     False      False
382       False      False      True      False     False      False
704       False      False      True      False     False      False
813       False      False     False      False     False      False

     Title_Sir
331      False
733      False
382      False
704      False
813      False

[5 rows x 25 columns]
331    0
733    0
382    0
704    0
813    0
Name: Survived, dtype: int64
```

[7]:
```
print(df.info())
print(df.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 26 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Survived         891 non-null    int64
 1   Pclass           891 non-null    int64
 2   Sex              891 non-null    int32
 3   Age              891 non-null    float64
 4   SibSp            891 non-null    int64
 5   Parch            891 non-null    int64
 6   Fare             891 non-null    float64
 7   FamilySize       891 non-null    int64
 8   Embarked_Q       891 non-null    bool
 9   Embarked_S       891 non-null    bool
 10  Title_Col        891 non-null    bool
 11  Title_Countess   891 non-null    bool
 12  Title_Don        891 non-null    bool
 13  Title_Dr         891 non-null    bool
 14  Title_Jonkheer   891 non-null    bool
 15  Title_Lady       891 non-null    bool
 16  Title_Major      891 non-null    bool
 17  Title_Master     891 non-null    bool
 18  Title_Miss       891 non-null    bool
 19  Title_Mlle       891 non-null    bool
 20  Title_Mme        891 non-null    bool
 21  Title_Mr         891 non-null    bool
 22  Title_Mrs        891 non-null    bool
 23  Title_Ms         891 non-null    bool
 24  Title_Rev        891 non-null    bool
 25  Title_Sir        891 non-null    bool
dtypes: bool(18), float64(2), int32(1), int64(5)
memory usage: 68.0 KB
None
Survived          0
Pclass            0
Sex               0
Age               0
SibSp             0
Parch             0
Fare              0
FamilySize        0
Embarked_Q        0
Embarked_S        0
Title_Col         0
Title_Countess    0
Title_Don         0
Title_Dr          0
Title_Jonkheer    0
Title_Lady        0
Title_Major       0
Title_Master      0
Title_Miss        0
Title_Mlle        0
Title_Mme         0
Title_Mr          0
Title_Mrs         0
Title_Ms          0
Title_Rev         0
Title_Sir         0
dtype: int64
```