Titanic Survival Prediction

This notebook explores the famous Titanic dataset from Kaggle.

Goals

- Understand the dataset and its structure.
- Perform data cleaning and preprocessing.
- Conduct Exploratory Data Analysis (EDA) with visualizations.
- Build a basic Machine Learning model to predict survival.

Dataset Description

The dataset includes various features about passengers such as age, sex, ticket class, and whether they survived.

Columns Explanation:

- `Survived`: Target variable (1 = Survived, 0 = Died)
- `Pclass`: Passenger class (1st, 2nd, 3rd)
- `Sex`: Gender of passenger
- `Age`: Age in years
- `SibSp`: Number of siblings/spouses aboard
- `Parch`: Number of parents/children aboard
- `Ticket`: Ticket number
- `Fare`: Ticket price
- `Cabin`: Cabin number (if available)
- `Embarked`: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

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

# Load dataset
train_data = pd.read_csv('/kaggle/input/titanic/train.csv')
test_data = pd.read_csv('/kaggle/input/titanic/test.csv')
train_data.head()
```

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

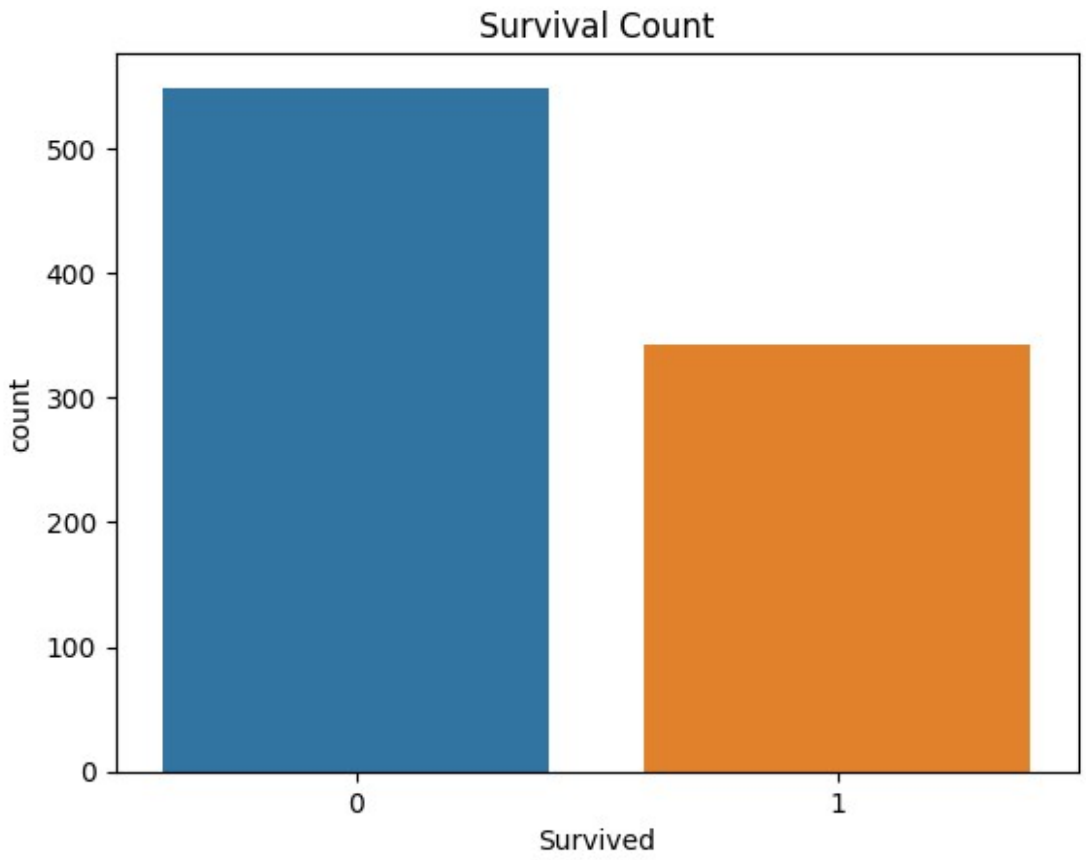
```
                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, 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
```

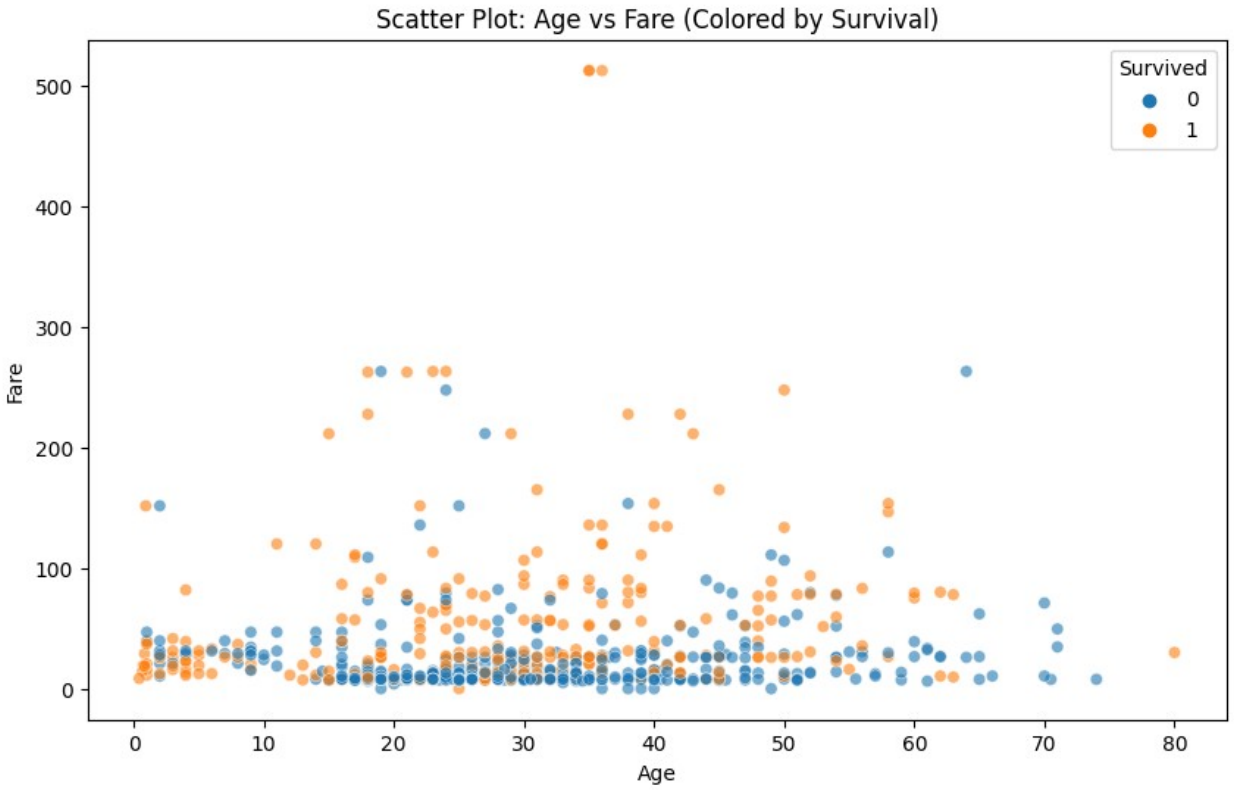
Exploratory Data Analysis (EDA)

Let's visualize survival rates based on key features.

```
# Survival count
sns.countplot(x='Survived', data=train_data)
plt.title('Survival Count')
plt.show()
```

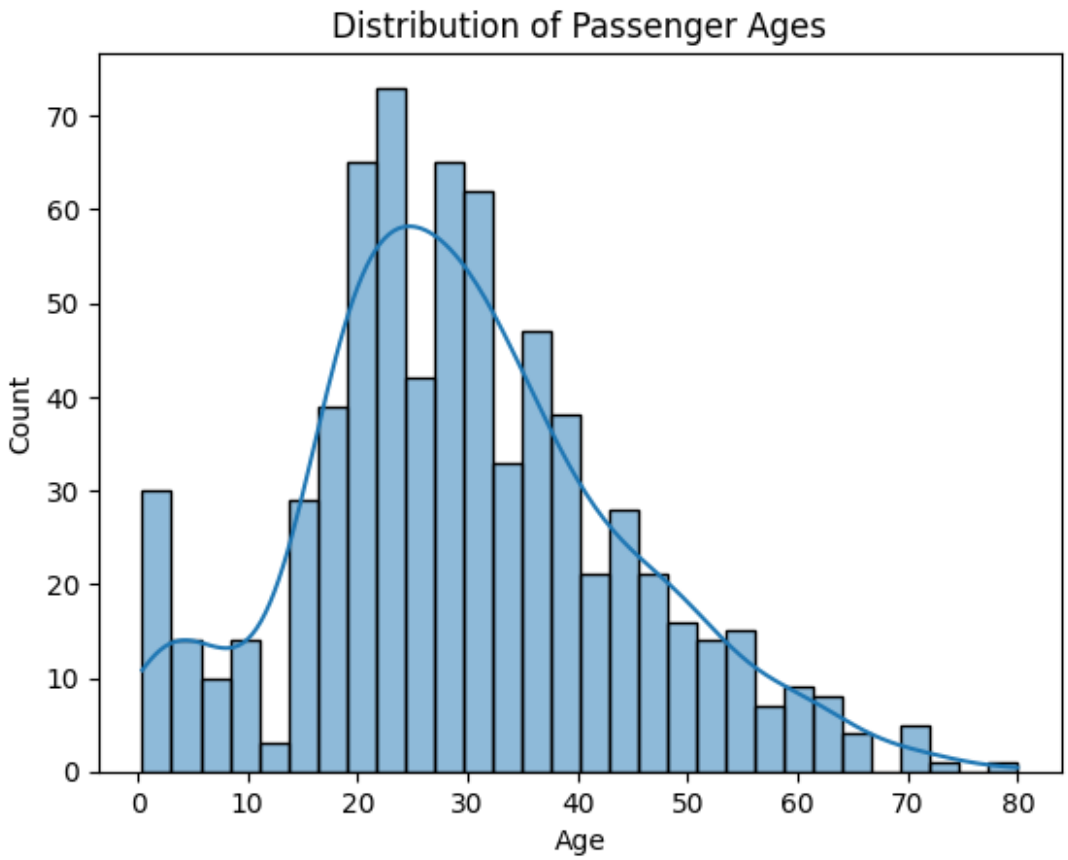



```
# Scatter plot of Age vs Fare, colored by Survival
plt.figure(figsize=(10,6))
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=train_data,
alpha=0.6)
plt.title('Scatter Plot: Age vs Fare (Colored by Survival)')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```

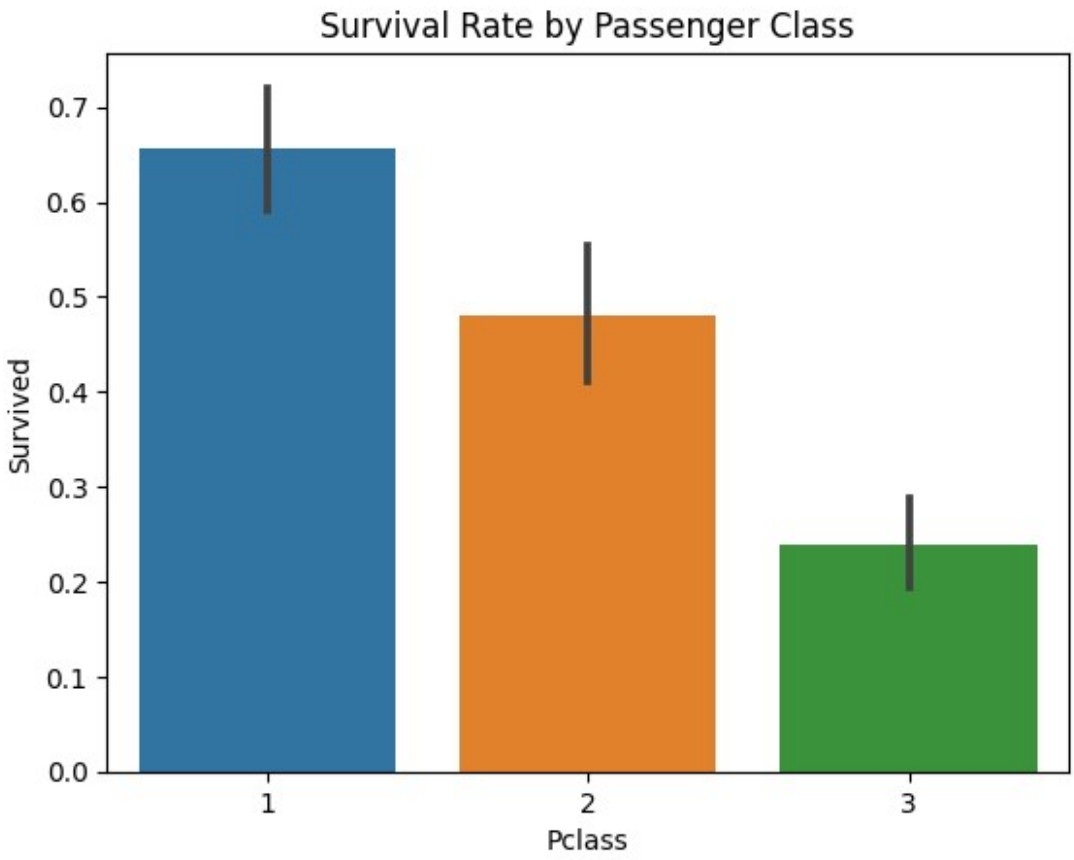



```
# Fixing Seaborn FutureWarning
train_data.replace([np.inf, -np.inf], np.nan, inplace=True)
sns.histplot(train_data['Age'], bins=30, kde=True)
plt.title('Distribution of Passenger Ages')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```

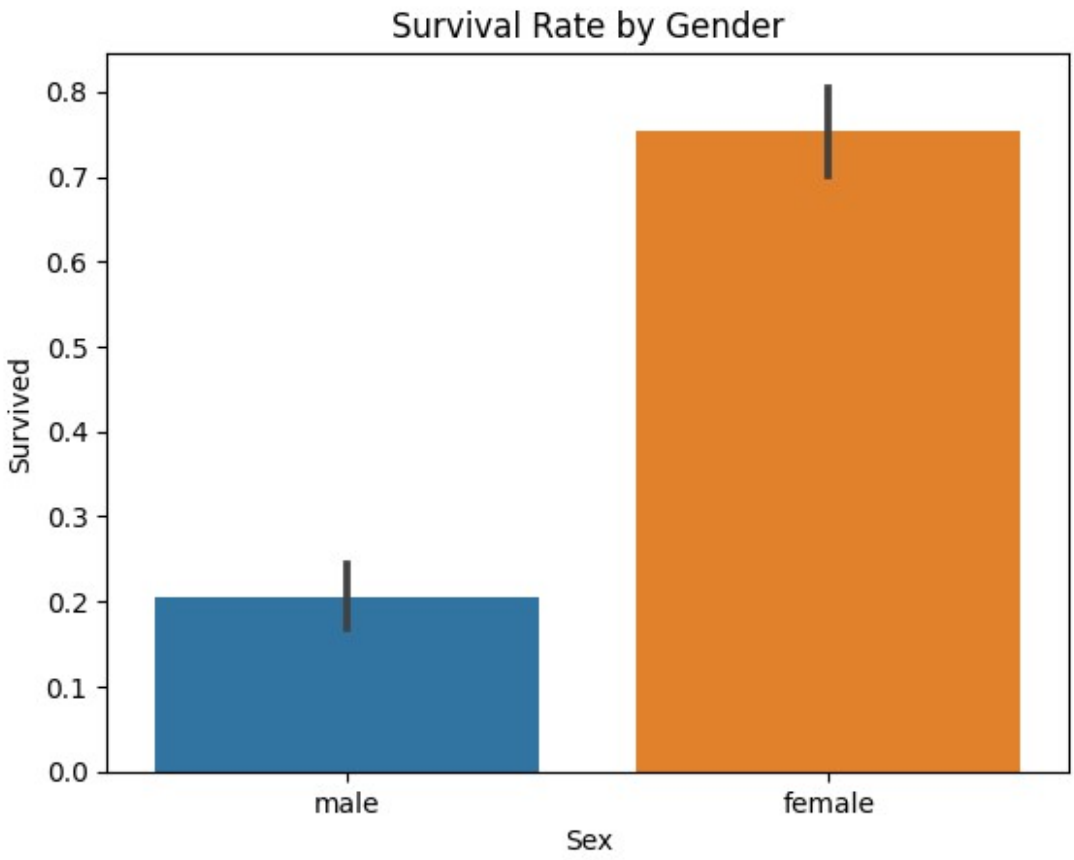
```
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
# Survival rate by class
sns.barplot(x='Pclass', y='Survived', data=train_data)
plt.title('Survival Rate by Passenger Class')
plt.show()
```




```
# Survival rate by gender
sns.barplot(x='Sex', y='Survived', data=train_data)
plt.title('Survival Rate by Gender')
plt.show()
```



Feature Engineering & Model Building

We'll build a basic model using Logistic Regression.

```
from sklearn.model_selection import train_test_splitfrom
sklearn.preprocessing import LabelEncoderfrom sklearn.linear_model
import LogisticRegressionfrom sklearn.metrics import accuracy_score#
Handling missing
valuestrain_data['Age'].fillna(train_data['Age'].median(),
inplace=True)train_data['Fare'].fillna(train_data['Fare'].median(),
inplace=True)train_data['Embarked'].fillna(train_data['Embarked'].mode
()[0], inplace=True)# Encoding categorical variablestrain_data['Sex']
=
LabelEncoder().fit_transform(train_data['Sex'])train_data['Embarked']
= LabelEncoder().fit_transform(train_data['Embarked'])# Selecting
featuresfeatures = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare',
'Embarked']X = train_data[features]y = train_data['Survived']#
Splitting dataX_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=42)# Training the modelmodel =
LogisticRegression(max_iter=200)model.fit(X_train, y_train)# Making
predictionsy_pred = model.predict(X_test)accuracy =
accuracy_score(y_test, y_pred)print(f'Logistic Regression Model
Accuracy: {accuracy:.2f}')
```

```
Logistic Regression Model Accuracy: 0.74
```

Conclusion

- We cleaned and explored the Titanic dataset.
- We visualized survival rates based on different features.
- We built a basic Logistic Regression model, achieving reasonable accuracy.

Further improvements can be made using advanced models like Random Forest or Neural Networks!