Titanic Survival Prediction Using Machine Learning

Based on Kaggle Titanic Dataset
Data Science Project

Tools :

- **Python**
- **Pandas**
- **Seaborn**
- **Scikit-learn**

By : Komal Digwal

KOMAL

Introduction (Project Overview)

 The Titanic disaster of 1912 is one of the most well-known shipwrecks in history. Over 1500 passengers lost their lives after the ship hit an iceberg during its maiden voyage.

 This dataset from Kaggle contains real data about the passengers onboard the Titanic — including their **age, sex, ticket class, fare, and more**.

  The project focuses on using **machine learning algorithms** to analyze this data and **predict which passengers were more likely to survive**.

 Tools & Technologies Used:

- Python
- Pandas, NumPy
- Seaborn & Matplotlib
- Scikit-learn

KOMAL

Dataset Description

💼 The dataset contains information about passengers aboard the Titanic. It includes demographic details, travel class, fare, and survival status.

❗ Below are some key columns:

Column Name	Description
PassengerId	Unique ID for each passenger
Survived	Survival status (0 = No, 1 = Yes)
Pclass	Ticket class (1 = 1st, 2 = 2nd, 3 = 3rd)
Name	Name of the passenger
Sex	Gender of the passenger
Age	Age of the passenger
SibSp	# of siblings/spouses aboard
Parch	# of parents/children aboard
Fare	Ticket fare
Embarked	Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

🧾 Total Rows: 891

📊 Missing Values Found In: Age, Cabin, Embarked

KOMAL

💻 Codes

🔧 Importing Libraries

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


from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix


import kagglehub
```

Download Dataset from KaggleHub

```
path = kagglehub.dataset_download("yasserh/titanic-dataset")
print("Dataset downloaded at:", path)
```

Output

```
Dataset downloaded at: C:\Users\Admin\.cache\kagglehub\datasets\yasserh\titanic-dataset\versions\1
```

Load the CSV File

```
file_path = os.path.join(path, "Titanic-Dataset.csv")
df = pd.read_csv(file_path)
df.head()
```

Output

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Basic Dataset Info

```
df.info()
df.describe()
```

KOMAL

Output

```
<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
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Visualize Missing Values

Data Cleaning

```
df['Age'] = df['Age'].fillna(df['Age'].median())
df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])
df.drop(['Cabin', 'Ticket', 'Name', 'PassengerId'], axis=1, inplace=True, errors='ignore')
```

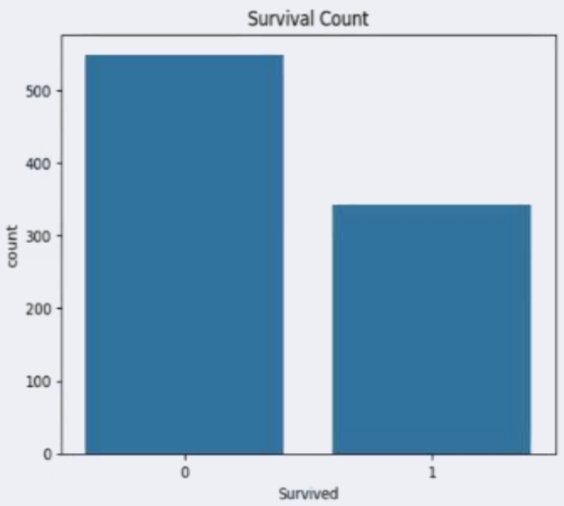
Visualize Important Features

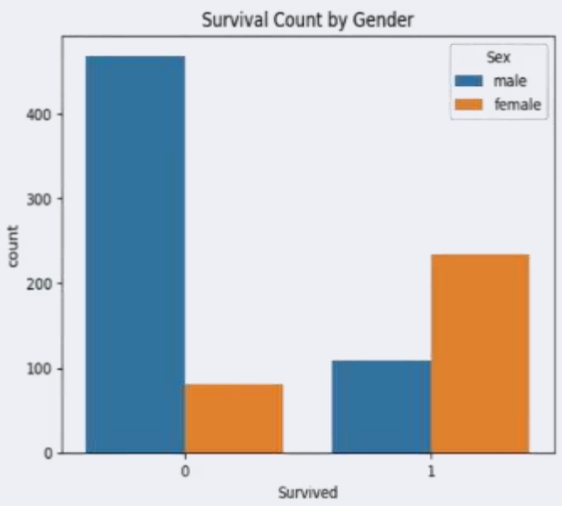
```
sns.countplot(data=df, x='Survived')
plt.title('Survival Count')
plt.show()
```

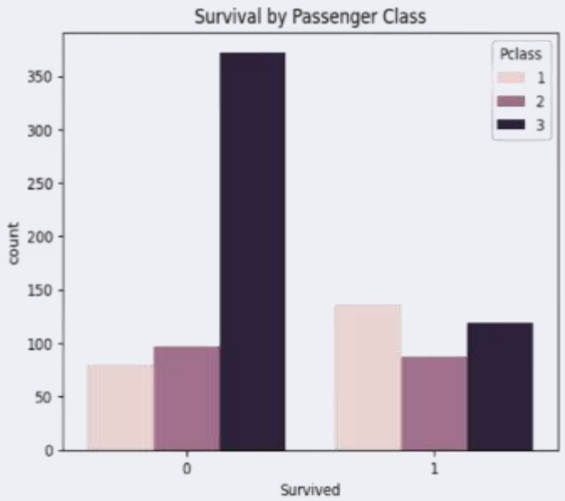
KOMAL

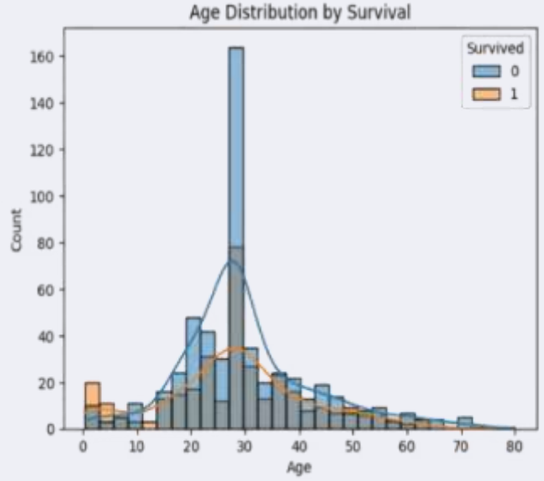
```
sns.countplot(data=df, x='Survived', hue='Sex')
plt.title('Survival Count by Gender')
plt.show()

sns.countplot(data=df, x='Survived', hue='Pclass')
plt.title('Survival by Passenger Class')
plt.show()

sns.histplot(data=df, x='Age', hue='Survived', kde=True, bins=30)
plt.title('Age Distribution by Survival')
plt.show()
```

Output









KOMAL

Encoding Categorical Variables

```
df = pd.get_dummies(df, columns=['Sex', 'Embarked'],
drop_first=True)
df.head()
```

Model Training – Logistic Regression

```
y = df['Survived']
X = df.drop('Survived', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

Output

	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	0	3	22.0	1	0	7.2500	True	False	True
1	1	1	38.0	1	0	71.2833	False	False	False
2	1	3	26.0	0	0	7.9250	False	False	True
3	1	1	35.0	1	0	53.1000	False	False	True
4	0	3	35.0	0	0	8.0500	True	False	True

Output

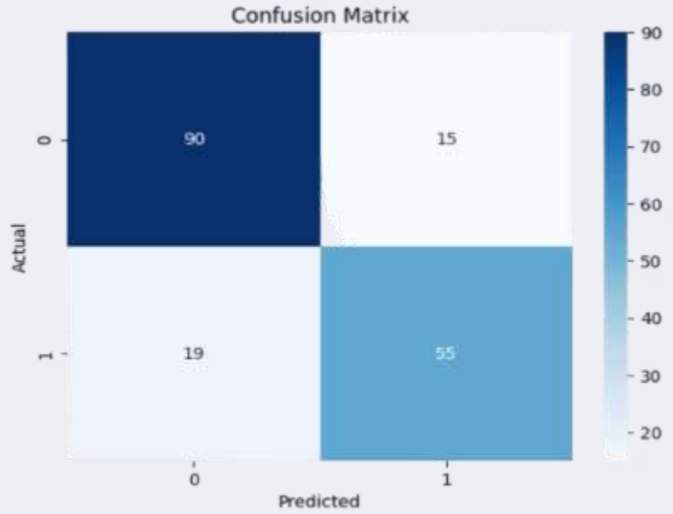
KOMAL

Model Evaluation & Performance Metrics

```
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Output

```
Accuracy: 0.8100558659217877

Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.84       105
           1       0.79      0.74      0.76        74

    accuracy                           0.81       179
   macro avg       0.81      0.80      0.80       179
weighted avg       0.81      0.81      0.81       179
```



KOMAL