Titanic Dataset Analysis :

RAW FILE :

titanic.csv

DATA CLEANING CODE :

SOFTWARES USED FOR DATA CLEANING :

- EXEL
- DATA WRANGLER
- JUPITER NOTEBOOK
- PYTHON LIBRARY PANDAS

CODE :

```
import pandas as pd

# checking for null values
data.isnull().sum()

# data manupulation
data = pd.read_csv("titanic.csv")
data['FamilyMembers'] = data['SibSp'] + data['Parch'] + 1
data['IS_Cabin'] = data['Cabin'].notna().astype(int)
data = data.drop(columns = ['SibSp', 'Parch', 'Name', 'Cabin'])
data['Age'] = data['Age'].fillna(data['Age'].median())
data['Embarked'] = data['Embarked'].fillna(data['Embarked'].mode()[0])

# data information
data.head(3)
```

```
data.describe()
data.info()
```

MACHINE LEARNING MODEL :

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Load data
data = pd.read_csv("titanic.csv")



# Feature engineering
data["FamilySize"] = data["SibSp"] + data["Parch"] + 1
data["HasCabin"] = data["Cabin"].notnull().astype(int)

# data droping
data.drop(["PassengerId", "Cabin", "Name", "Ticket"], axis=1, errors="ignore", inplace=True)

# Handle missing values
data["Age"] = data["Age"].fillna(data["Age"].median())
data["Embarked"] = data["Embarked"].fillna(data["Embarked"].mode()[0])

# Separate features and target
X = data.drop("Survived", axis=1)
y = data["Survived"]

# One-hot encoding
X = pd.get_dummies(X, columns=["Sex", "Embarked"], drop_first=True)
```

```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Scale numerical features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train Logistic Regression
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Predictions
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Evaluation
print("Training Accuracy:", accuracy_score(y_train, y_train_pred))
print("Test Accuracy:", accuracy_score(y_test, y_test_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_test_pred))
print("\nClassification Report:\n", classification_report(y_test, y_test_pred))
```

EDA CODE :

```
import numpy as np
import pandas as pd

pd.set_option('display.max_columns', None)

data = pd.read_csv("titanic.csv")
```

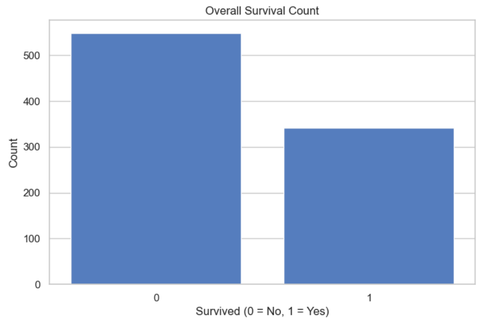
```
# Drop PassengerId (not useful for prediction)
data.drop(columns=['PassengerId'], inplace=True)

# Fill missing Age with median
data['Age'] = data['Age'].fillna(data['Age'].median())

# Fill missing Embarked with mode
data['Embarked'] = data['Embarked'].fillna(data['Embarked'].mode()[0])

# Create FamilySize feature
data['FamilySize'] = data['SibSp'] + data['Parch'] + 1

total = len(data)
survived = data['Survived'].sum()
died = total - survived

print(f"Survived: {survived} / {total} ({survived/total:.2%})")
print(f"Died: {died} / {total} ({died/total:.2%})")

sex_survival = data.groupby('Sex')['Survived'].agg(['sum', 'count'])
sex_survival['SurvivalRate'] = sex_survival['sum'] / sex_survival['count']
print(sex_survival)

bins = [0, 21, 41, 81]
labels = ['<21', '21-40', '41-80']
data['AgeGroup'] = pd.cut(data['Age'], bins=bins, labels=labels)

age_group_survival = data.groupby('AgeGroup')['Survived'].agg(['sum', 'count'])
age_group_survival['SurvivalRate'] = age_group_survival['sum'] / age_group_survival['count']
print(age_group_survival)

age_sex = data.groupby(['AgeGroup', 'Sex'])['Survived'].agg(['sum', 'count'])
age_sex['SurvivalRate'] = age_sex['sum'] / age_sex['count']
print(age_sex)
```

```
pclass_survival = data.groupby('Pclass')['Survived'].agg(['sum', 'count'])
pclass_survival['SurvivalRate'] = pclass_survival['sum'] / pclass_survival['c
ount']
print(pclass_survival)

pclass_sex = data.groupby(['Pclass', 'Sex'])['Survived'].agg(['sum', 'coun
t'])
pclass_sex['SurvivalRate'] = pclass_sex['sum'] / pclass_sex['count']
print(pclass_sex)

embarked_survival = data.groupby('Embarked')['Survived'].agg(['sum', 'co
unt'])
embarked_survival['SurvivalRate'] = embarked_survival['sum'] / embarked_
survival['count']
print(embarked_survival)

embarked_sex = data.groupby(['Embarked', 'Sex'])['Survived'].agg(['sum',
'count'])
embarked_sex['SurvivalRate'] = embarked_sex['sum'] / embarked_sex['cou
nt']
print(embarked_sex)

data['HasCabin'] = data['Cabin'].notnull().astype(int)

cabin_survival = data.groupby('HasCabin')['Survived'].agg(['sum', 'count'])
cabin_survival['SurvivalRate'] = cabin_survival['sum'] / cabin_survival['cou
nt']
print(cabin_survival)
```

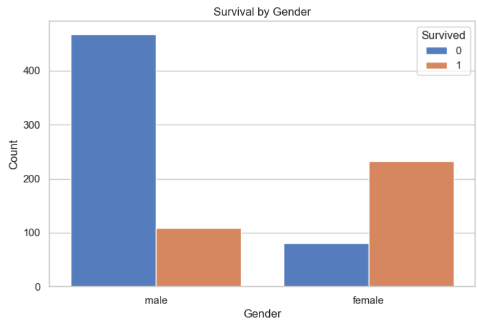
VISUALIZATION CODE :

```
sns.countplot(x='Survived', data=data)
plt.title("Overall Survival Count")
plt.xlabel("Survived (0 = No, 1 = Yes)")
```

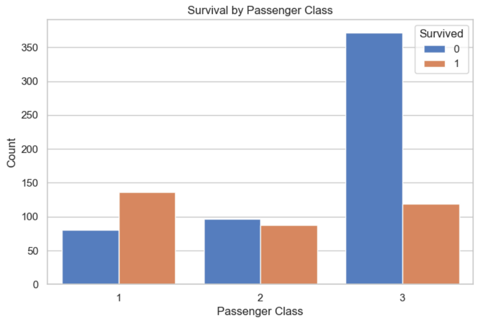
```
plt.ylabel("Count")
plt.show()
```



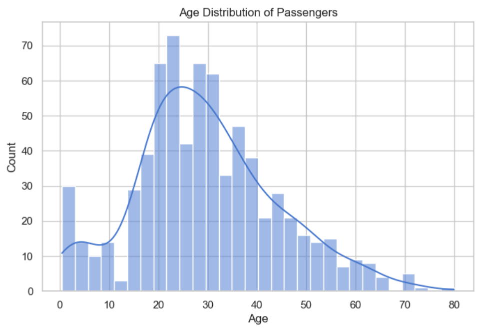

```
sns.countplot(x='Sex', hue='Survived', data=data)
plt.title("Survival by Gender")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.legend(title="Survived")
plt.show()
```



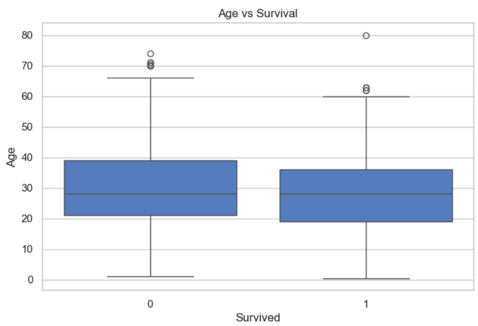

```
sns.countplot(x='Pclass', hue='Survived', data=data)
plt.title("Survival by Passenger Class")
plt.xlabel("Passenger Class")
plt.ylabel("Count")
plt.show()
```



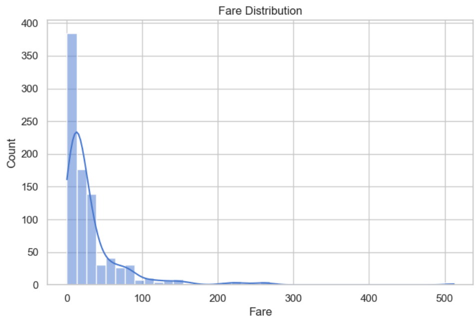

```
sns.histplot(data['Age'], bins=30, kde=True)
plt.title("Age Distribution of Passengers")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
```



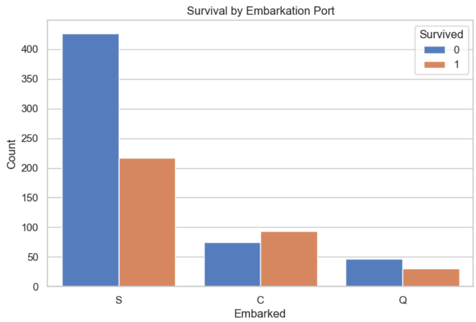

```
sns.boxplot(x='Survived', y='Age', data=data)
plt.title("Age vs Survival")
plt.xlabel("Survived")
plt.ylabel("Age")
plt.show()
```



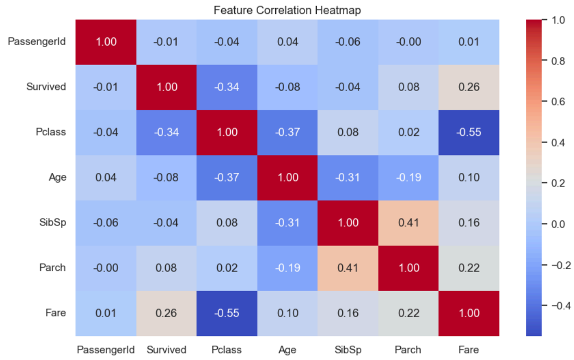
```
sns.histplot(data['Fare'], bins=40, kde=True)
plt.title("Fare Distribution")
plt.xlabel("Fare")
plt.ylabel("Count")
plt.show()
```



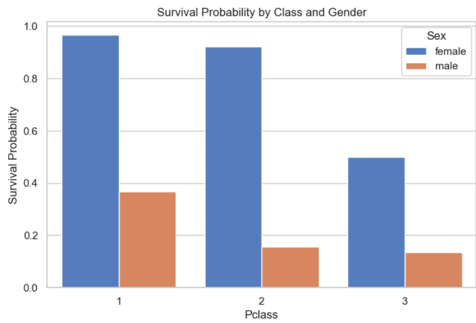

```
sns.countplot(x='Embarked', hue='Survived', data=data)
plt.title("Survival by Embarkation Port")
plt.xlabel("Embarked")
plt.ylabel("Count")
plt.show()
```




```
plt.figure(figsize=(10,6))
sns.heatmap(data.corr(numeric_only=True), annot=True, cmap='coolwar
m', fmt=".2f")
plt.title("Feature Correlation Heatmap")
plt.show()
```




```
survival_rate = data.groupby(['Sex','Pclass'])['Survived'].mean().reset_index()

sns.barplot(x='Pclass', y='Survived', hue='Sex', data=survival_rate)
plt.title("Survival Probability by Class and Gender")
plt.ylabel("Survival Probability")
plt.show()
```



🚢 Titanic Survival Prediction using Logistic Regression

📌 Project Overview

This project builds a machine learning model to predict passenger survival , complete exploratory data analysis and visualization of the Titanic dataset.

The workflow covers **data cleaning, feature engineering, EDA, visualization with graphs , model training, and evaluation**, following best practices in applied data science.

🎯 Objective

To predict whether a passenger survived the Titanic disaster based on demographic and travel-related features using a **Logistic Regression** classifier.

📊 Dataset

- **Source:** Titanic Dataset (`titanic.csv`)
- **Total records:** 891
- **Target variable:** `Survived`
  - `1` → Survived
  - `0` → Did not survive

Class Distribution

Class	Count	Percentage
Survived	342	38%
Did Not Survive	549	62%

> The dataset is moderately imbalanced, so metrics beyond accuracy are required.

🛠️ Data Preprocessing

Data Cleaning

- `Age` → imputed using **median**
- `Embarked` → imputed using **mode**
- Dropped non-informative text columns:
  - `PassengerId`
  - `Name`
  - `Ticket`

Feature Engineering

- **FamilySize** = `SibSp + Parch + 1`
- **IS_Cabin**
  - `1` → Cabin information available
  - `0` → Cabin missing

Encoding & Scaling

- One-hot encoding applied to:
  - `Sex`
  - `Embarked`
- Numerical features scaled using **StandardScaler**
- Data split:
  - **80% training**
  - **20% testing**
  - `random_state = 42`

🤖 Model

- **Algorithm:** Logistic Regression
- **Reason for choice:**
  - Strong baseline for binary classification
  - Interpretable
  - Fast and efficient

📈 Model Performance

Accuracy

- **Training Accuracy:** 79.78%
- **Test Accuracy:** 82.12%

> Close train–test performance indicates good generalization and no overfitting.

Confusion Matrix

```
[[91 14]
 [18 56]]
```

Metric	Count
True Negatives	91
False Positives	14
False Negatives	18
True Positives	56

Classification Report

Class	Precision	Recall	F1-score
Did Not Survive (0)	0.83	0.87	0.85
Survived (1)	0.80	0.76	0.78
Overall Accuracy			**0.82**

🔍 Key Insights

- **Sex** is the strongest predictor of survival.
- **Passenger class** and **cabin availability** have significant impact.
- The model captures historical survival patterns accurately (e.g., higher survival for females and higher classes).

✅ Conclusion

The Logistic Regression model provides a **strong and interpretable baseline**, achieving 82% test accuracy with balanced precision and recall.

This makes it a reliable foundation for further improvements and experimentation.

🚀 Future Improvements

- ROC–AUC analysis
- Threshold tuning to reduce false negatives
- Cross-validation
- Feature importance analysis
- Ensemble models (Random Forest, Gradient Boosting)

🧠 Skills Demonstrated

- Data Cleaning & Imputation
- Feature Engineering
- Categorical Encoding
- Model Training & Evaluation
- Real-world ML debugging
- Interpretation of results

🔎 Exploratory Data Analysis (EDA)

Comprehensive exploratory analysis was conducted to understand survival patterns across demographic and socio-economic features.

📌 Overall Survival Statistics

- **Total passengers:** 891
- **Survived:** 342 (**38%**)
- **Did not survive:** 549 (**62%**)

This confirms a **moderately imbalanced dataset**, reinforcing the need for evaluation metrics beyond accuracy.

👩 Survival by Sex

Sex	Survived	Total	Survival Rate
Female	233	314	**74.2%**

Sex	Survived	Total	Survival Rate
Male	109	577	**19.6%**

🔹 **Insight:**

Sex is the strongest individual predictor of survival. Females were significantly more likely to survive than males.

👶🧓 Survival by Age Groups

Age < 21

- **Survived:** 82 / 180 → **45.3%**
- Males: 103 total → 48 survived (**46.6%**)
- Females: 77 total → 34 survived (**44.2%**)

Age 21–40

- **Survived:** 205 / 563 → **36.5%**
- Males: 374 total → 62 survived (**16.6%**)
- Females: 189 total → 143 survived (**75.7%**)

Age 41–80

- **Survived:** 55 / 148 → **37.2%**
- Males: 100 total → 18 survived (**18.0%**)
- Females: 48 total → 37 survived (**77.1%**)

🔹 **Insights:**

- Female survival rates remain high across all age groups.
- Male survival rates are consistently low regardless of age.
- Age alone is a weak predictor, but **Age + Sex interaction** provides valuable signal.

🎟️ Survival by Passenger Class (Pclass)

Pclass 1

- Survived: 136 / 216 → **63.0%**

- Female survival: **96.8%**
- Male survival: **36.9%**

Pclass 2

- Survived: 119 / 184 → **64.7%**
- Female survival: **92.1%**
- Male survival: **15.7%**

Pclass 3

- Survived: 87 / 491 → **17.7%**
- Female survival: **50.0%**
- Male survival: **13.5%**

🔹 **Insights:**

- Passenger class is a strong ordinal predictor.
- Third-class males had the lowest survival probability.
- First- and second-class females had extremely high survival rates.

🌍 Survival by Embarkation Port

Port	Survived	Total	Survival Rate
Southampton (S)	219	646	33.9%
Cherbourg (C)	93	168	**55.4%**
Queenstown (Q)	30	77	39.0%

Gender Breakdown

- Females consistently outperformed males across all embarkation points.
- Cherbourg passengers had the highest overall survival.

🔹 **Insight:**

Embarkation port adds secondary predictive value and is correlated with passenger class.

🛏️ Cabin Availability

Cabin Status	Survived	Total	Survival Rate
Cabin Present	136	204	**66.7%**
Cabin Missing	206	687	30.0%

🔹 **Insight:**

Cabin availability strongly correlates with survival and socio-economic status.

This justified the creation of the `IS_Cabin` feature.

🧠 Key EDA Takeaways

- **Sex** is the single most influential feature.
- **Passenger class** and **cabin availability** provide strong socio-economic signals.
- **Age alone** is noisy but becomes informative when combined with sex.
- Feature engineering is essential to capture these interactions.