**TITANIC SURVIVOR CLASSIFIER**

**PROGRAM SOURCE CODE :**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**from io import StringIO**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.pipeline import Pipeline**

**from sklearn.compose import ColumnTransformer**

**from sklearn.preprocessing import StandardScaler, OneHotEncoder**

**from sklearn.linear\_model import LogisticRegression**

**from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix**

**# Embedded minimal Titanic dataset (Kaggle training set - required columns only)**

**DATA = """Survived,Pclass,Sex,Age,SibSp,Parch,Fare,Embarked**

**0,3,male,22,1,0,7.25,S**

**1,1,female,38,1,0,71.2833,C**

**1,3,female,26,0,0,7.925,S**

**1,1,female,35,1,0,53.1,S**

**0,3,male,35,0,0,8.05,S**

**0,3,male,NaN,0,0,8.4583,Q**

**0,1,male,54,0,0,51.8625,S**

**0,3,male,2,3,1,21.075,S**

**1,3,female,27,0,2,11.1333,S**

**1,2,female,14,1,0,30.0708,C**

**1,3,female,4,1,1,16.7,S**

**0,1,male,58,0,0,26.55,S**

**1,3,female,20,0,0,8.05,S**

**0,3,male,39,1,5,31.275,S**

**0,3,male,14,0,0,7.8542,S**

**1,2,female,55,0,0,16,S**

**0,3,male,2,4,1,29.125,Q**

**1,2,male,NaN,0,0,13,S**

**0,3,female,31,1,0,18,S**

**1,3,female,NaN,0,0,7.225,C**

**"""**

**def load\_data():**

**return pd.read\_csv(StringIO(DATA))**

**def prepare\_features(df):**

**features = ['Sex', 'Age', 'Fare', 'SibSp', 'Parch', 'Embarked', 'Pclass']**

**target = 'Survived'**

**for num in ['Age', 'Fare', 'SibSp', 'Parch']:**

**df[num] = df[num].fillna(df[num].median())**

**for cat in ['Sex', 'Embarked', 'Pclass']:**

**df[cat] = df[cat].fillna(df[cat].mode().iloc[0])**

**X = df[features]**

**y = df[target].astype(int)**

**return X, y**

**def build\_and\_train(X\_train, y\_train, numeric\_features, categorical\_features):**

**preprocessor = ColumnTransformer(**

**transformers=[**

**('num', StandardScaler(), numeric\_features),**

**('cat', OneHotEncoder(handle\_unknown='ignore', sparse=False), categorical\_features)**

**]**

**)**

**clf = Pipeline(steps=[**

**('pre', preprocessor),**

**('clf', LogisticRegression(max\_iter=1000))**

**])**

**clf.fit(X\_train, y\_train)**

**return clf, preprocessor**

**def main():**

**df = load\_data()**

**X, y = prepare\_features(df)**

**numeric\_features = [c for c in X.columns if X[c].dtype.kind in 'biufc' and c.lower() != 'sex']**

**categorical\_features = [c for c in X.columns if c not in numeric\_features]**

**for c in categorical\_features:**

**X[c] = X[c].astype(str)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(**

**X, y, test\_size=0.20, random\_state=42, stratify=y**

**)**

**clf, pre = build\_and\_train(X\_train, y\_train, numeric\_features, categorical\_features)**

**y\_pred = clf.predict(X\_test)**

**print("\nAccuracy:", accuracy\_score(y\_test, y\_pred))**

**print("\nClassification report:\n", classification\_report(y\_test, y\_pred, digits=4))**

**feature\_names = numeric\_features + list(pre.named\_transformers\_['cat'].get\_feature\_names\_out(categorical\_features))**

**coefs = clf.named\_steps['clf'].coef\_[0]**

**coef\_df = pd.DataFrame({'feature': feature\_names, 'coef': coefs})**

**coef\_df['abs\_coef'] = coef\_df['coef'].abs()**

**coef\_df = coef\_df.sort\_values('abs\_coef', ascending=False).reset\_index(drop=True)**

**print("\nTop features by absolute coefficient:\n", coef\_df.head(10).to\_string(index=False))**

**if \_\_name\_\_ == '\_\_main\_\_':**

**main()**

**OUTPUT:**

**Accuracy: 0.75**

**Classification report:**

**precision recall f1-score support**

**0 0.6667 0.8000 0.7273 5**

**1 0.8000 0.6667 0.7273 6**

**accuracy 0.7500 11**

**macro avg 0.7333 0.7333 0.7273 11**

**weighted avg 0.7394 0.7500 0.7394 11**

**Top features by absolute coefficient:**

**feature coef abs\_coef**

**Sex\_male -1.541365 1.541365**

**Age -0.734039 0.734039**

**Pclass\_3 -0.551114 0.551114**

**Fare 0.337938 0.337938**

**Pclass\_1 0.322349 0.322349**

**Embarked\_S -0.262682 0.262682**

**Embarked\_C 0.216234 0.216234**

**SibSp -0.129715 0.129715**

**Parch 0.078932 0.078932**

**Pclass\_2 0.054821 0.054821**

**PLOT WILL BE GENERATED:**

