Assignment-1

Name : Medha Aggarwal

Roll No. : 2201CS90

Course : APR (CS502)

Introduction:

I applied Logistic Regression on the Titanic dataset to predict passenger survival. The dataset contains demographic and travel information such as age, class, gender, and fare, etc. along with survival labels.

Dataset:

The dataset used in this assignment is the **Titanic: Machine Learning from Disaster** dataset, available on Kaggle and searchable via https://datasetsearch.research.google.com/. It contains passenger information from the Titanic shipwreck, with the goal of predicting whether a passenger survived or not.

- Number of Instances (Rows): 891 passengers
- Number of Features (Columns): 12

Methodology:

- Data Preprocessing: Missing values in 'Age' were filled with the median, and missing values in 'Embarked' were filled with the most frequent value.

Categorical columns ('Sex' and 'Embarked') were encoded using Label Encoding. Features were standardized using StandardScaler.

- Model Training: Logistic Regression was used as the classification algorithm. The dataset was split into 80% training and 20% testing. Performance evaluated using Accuracy, Confusion Matrix, and Classification Report. A correlation heatmap was also generated.

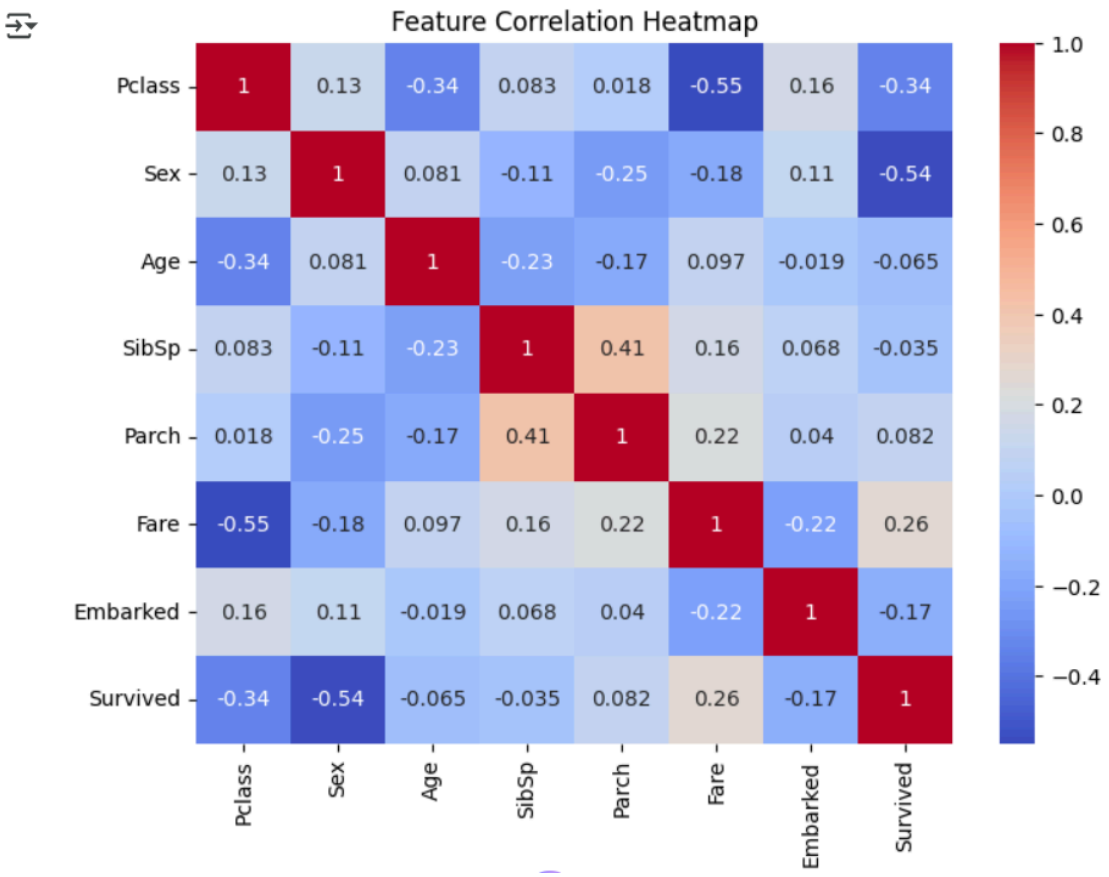
Code:

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

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

# Load dataset
data = pd.read_csv("train.csv")

# Preprocessing
data["Age"] = data["Age"].fillna(data["Age"].median())
data["Embarked"] = data["Embarked"].fillna(data["Embarked"].mode()[0])

# Encode categorical variables
le = LabelEncoder()
data["Sex"] = le.fit_transform(data["Sex"])
data["Embarked"] = le.fit_transform(data["Embarked"])

# Select features
features = ["Pclass", "Sex", "Age", "SibSp", "Parch", "Fare", "Embarked"]
X = data[features]
```

```
y = data["Survived"]

# Standardize features
scaler=StandardScaler()
X_scaled=scaler.fit_transform(X)

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

# Logistic Regression
log_reg = LogisticRegression(max_iter=500)
log_reg.fit(X_train, y_train)
y_pred_log = log_reg.predict(X_test)

# Results
print(" 🔹 Logistic Regression Results")
print("Accuracy:", accuracy_score(y_test, y_pred_log))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_log))
print("Classification Report:\n", classification_report(y_test,
y_pred_log))

# Correlation Heatmap
plt.figure(figsize=(8,6))
sns.heatmap(data[features + ["Survived"]].corr(), annot=True,
cmap="coolwarm")
plt.title("Feature Correlation Heatmap")
plt.show()
```

Outputs:

```
Logistic Regression Results
Accuracy: 0.8044692737430168
Confusion Matrix:
 [[90 15]
 [20 54]]
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.86      0.84       105
           1       0.78      0.73      0.76        74

    accuracy                           0.80       179
   macro avg       0.80      0.79      0.80       179
weighted avg       0.80      0.80      0.80       179
```



Results and Discussion:

The Logistic Regression model achieved an accuracy of approximately 80% on the test dataset. From the outputs, it is observed that passenger class (Pclass) and gender (Sex) are strong predictors of survival. Women and higher-class passengers had higher chances of survival.