#Logistic Regression(AIR-IV)-UNIT-2

```
# Import necessary libraries
import numpy as np
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

# Step 1: Create a synthetic dataset
data = {
    "Age": [22, 25, 47, 52, 46, 56, 55, 26, 29, 49],
    "EstimatedSalary": [19000, 20000, 43000, 57000, 46000, 83000, 73000, 21000, 25000, 67000],
    "Purchased": [0, 0, 1, 1, 1, 1, 1, 0, 0, 1]
}
df = pd.DataFrame(data)

# Step 2: Define features (X) and target (y)
X = df[['Age', 'EstimatedSalary']]  # Features
y = df['Purchased']  # Target

# Step 3: Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Step 4: Feature scaling (important for Logistic Regression)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)

# Step 5: Train the Logistic Regression model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# Step 6: Make predictions
y_pred = model.predict(X_test_scaled)

# Step 7: Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)

# Step 8: Test with a new customer (real-time input)
new_customer = [[30, 50000]]  # Age=30, EstimatedSalary=50,000
new_customer_scaled = scaler.transform(new_customer)  # Scale the input
prediction = model.predict(new_customer_scaled)

if prediction[0] == 1:
    print("The customer is likely to purchase the product.")
else:
    print("The customer is unlikely to purchase the product.")
```