Department of Computer Science Engineering (AI)

ARTIFICAL INTELLIGENCE

Project Report

On

Predict Product Return

By

ANSHU KUMARI

202401100300055

Dataset Used: product_return.csv

Date: 22/04/2025

TABLE OF CONTENTS

Page No.

1.1 INTRODUCTION

In today's e-commerce environment, product returns significantly affect profitability and customer satisfaction. Predicting which products are likely to be returned can help reduce operational costs and improve decision-making.

This project involves creating a machine learning model to predict the return status of products using features such as purchase amount, customer review score, and days taken for delivery. The dataset contains information on these parameters along with the return status.

1.2 Methodology

We followed these steps to develop the predictive model:

1. **Data Preprocessing:**
   - Encoded the target variable 'returned' from categorical ("yes", "no") to binary (1, 0).
   - Selected features: `purchase_amount`, `review_score`, and `days_to_delivery`.
2. **Model Selection:**
   - Used **Logistic Regression**, a simple and efficient classification model suitable for binary outputs.
3. **Data Splitting:**
   - Split the data into 80% training and 20% testing using `train_test_split()`.

4. **Training:**
    - Trained the model using the training set.
5. **Evaluation:**
    - Evaluated model performance using Accuracy Score, Confusion Matrix, and Classification Report.

CODE

```python
# Import required libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns

# Load and preprocess data
df = pd.read_csv("/content/product_return.csv")
df['returned'] = LabelEncoder().fit_transform(df['returned'])  # yes=1, no=0

# Define features and label
X = df[['purchase_amount', 'review_score', 'days_to_delivery']]
y = df['returned']

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
model = LogisticRegression()
model.fit(X_train, y_train)
```

✓ 3s completed at 10:28 AM

```python
# Define features and label
X = df[['purchase_amount', 'review_score', 'days_to_delivery']]
y = df['returned']

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, cmap="Blues", fmt='d', xticklabels=['Not Returned', 'Returned'], yticklabels=['Not Returned', 'Returned'])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

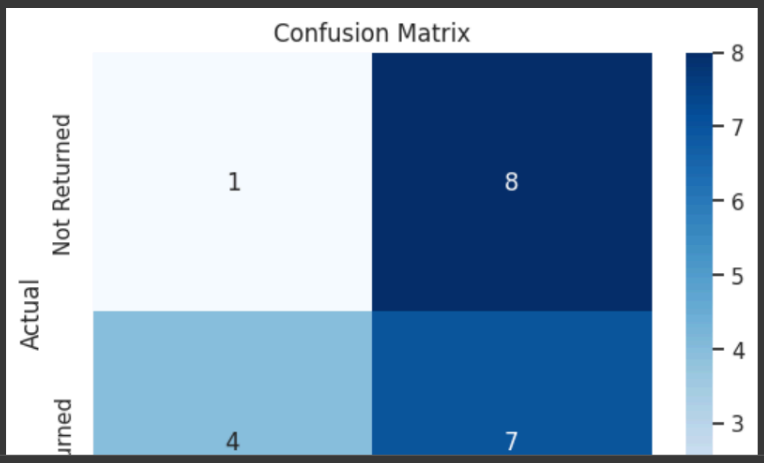
✓ 3s completed at 10:28 AM

****OUTPUT******

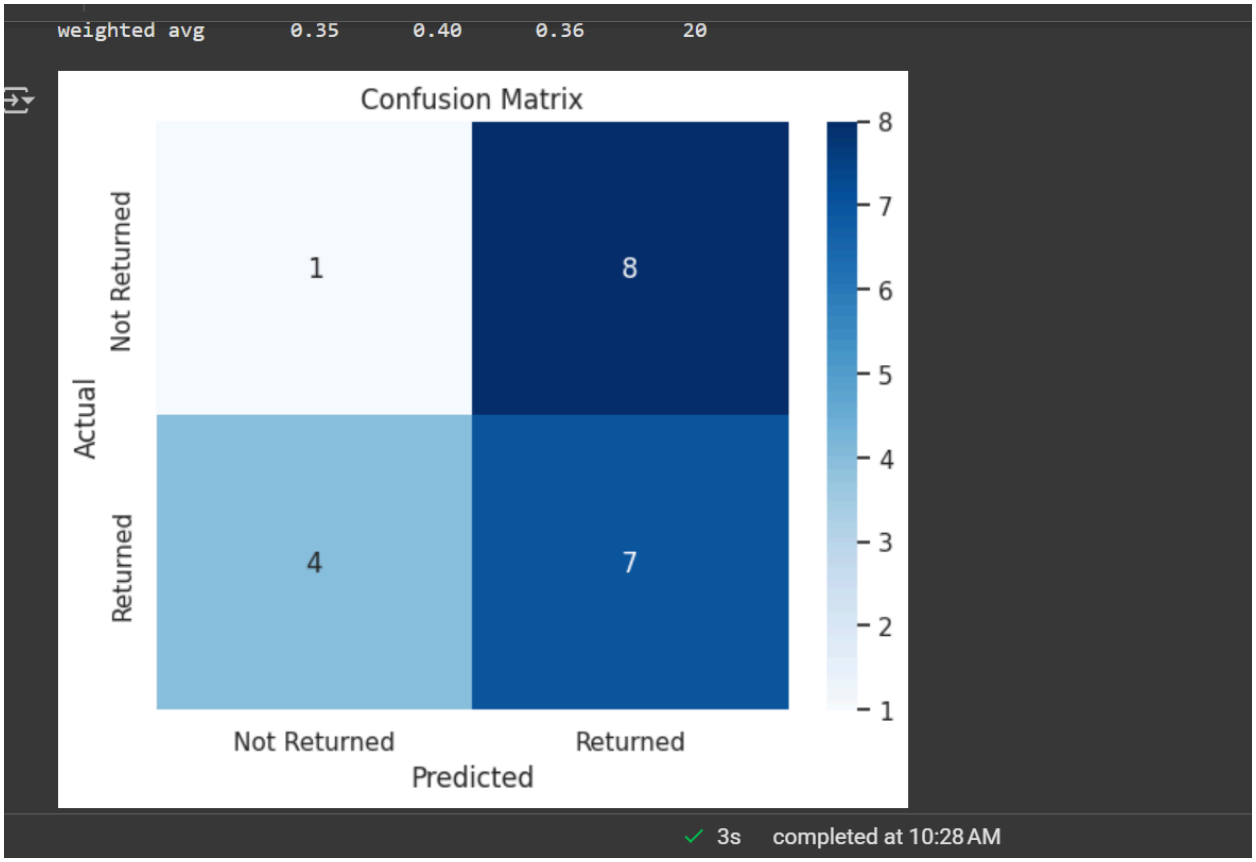
```
Accuracy: 0.4
              precision    recall  f1-score   support

           0       0.20      0.11      0.14         9
           1       0.47      0.64      0.54        11

    accuracy                           0.40        20
   macro avg       0.33      0.37      0.34        20
weighted avg       0.35      0.40      0.36        20
```



✓ 3s completed at 10:28 AM



*******References/Credits

- Dataset: Provided by instructor (`product_return.csv`)

- Libraries: Scikit-learn, Pandas, Matplotlib, Seaborn
- Tools: Google Colab, GitHub