KIET Group of Institutions, Ghaziabad

Assessment Report

on

“Predict Product Return”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Prakhar Singh

Roll Number : 202401100300172

Section: C

Under the supervision of

“MAYANK LAKHOTIA”

Introduction

In the rapidly growing landscape of e-commerce, product returns pose a significant challenge to both customer satisfaction and business profitability. Returns not only increase operational costs but also reflect gaps in customer expectations, logistics, or product quality. Accurately predicting whether a product will be returned enables businesses to take proactive measures such as improving product descriptions, flagging high-risk transactions, or tailoring post-purchase support.

This project aims to build a machine learning model that predicts the likelihood of a product being returned based on key features from the transaction, such as purchase amount, customer review score, and delivery time. By analyzing historical data, the model learns patterns and relationships that help forecast future return behavior. The goal is to assist companies in making data-driven decisions to reduce return rates, optimize logistics, and enhance the overall customer experience.

Methodology

1. Data Collection & Understanding

The dataset used contains historical records of customer purchases, including features such as:

- **Purchase Amount** (`purchase_amount`)

- **Review Score** (`review_score`)
- **Delivery Time** (`days_to_delivery`)
- **Return Status** (`returned`, the target variable)

An initial exploration was conducted to understand feature distributions, identify missing values, and verify data consistency.

---

2. Data Preprocessing

- **Column Cleanup:** Column names were standardized by removing whitespace and renaming for clarity.
- **Missing Value Treatment:** Missing numerical values in features such as `review_score`, `shipping_time`, and `price` were filled using the median of each column.
- **Feature Scaling:** All numeric features were normalized using **StandardScaler** to ensure uniformity in scale, which is beneficial for many machine learning algorithms.

---

3. Feature Engineering

The final set of features used for modeling included:

- Review Score
- Purchase Amount (renamed as Price)
- Shipping Time (days to delivery)

These features were selected for their potential predictive value in influencing customer satisfaction and return behavior.

---

4. Model Training

A **Random Forest Classifier** was chosen due to its robustness, ability to handle non-linear relationships, and feature importance interpretation. The data was split into training and testing sets using an 80/20 split. The model was trained on the training data and evaluated on the test data.

---

5. Model Evaluation

Model performance was assessed using:

- **Classification Report** (Precision, Recall, F1-score)
- **Confusion Matrix** (to visualize true vs. predicted labels)
- **Feature Importance** (to identify which features contributed most to predictions)

---

6. Visualization

- A confusion matrix was plotted to evaluate classification performance visually.

- A bar chart of feature importances was generated to interpret the influence of each feature.

CODE OF PROBLEM

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay



# Load dataset

df = pd.read_csv("/content/product_return.csv")



# Clean column names

df.columns = df.columns.str.strip()



# Rename for consistency

df.rename(columns={
```

```python
    'purchase_amount': 'price',
    'days_to_delivery': 'shipping_time',
    'returned': 'return'
}, inplace=True)

# Fill missing values
df['review_score'] = df['review_score'].fillna(df['review_score'].median())
df['shipping_time'] = df['shipping_time'].fillna(df['shipping_time'].median())
df['price'] = df['price'].fillna(df['price'].median())

# Feature selection
features = ['review_score', 'price', 'shipping_time']
X = df[features]
y = df['return']

# 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)

# Train classifier
```

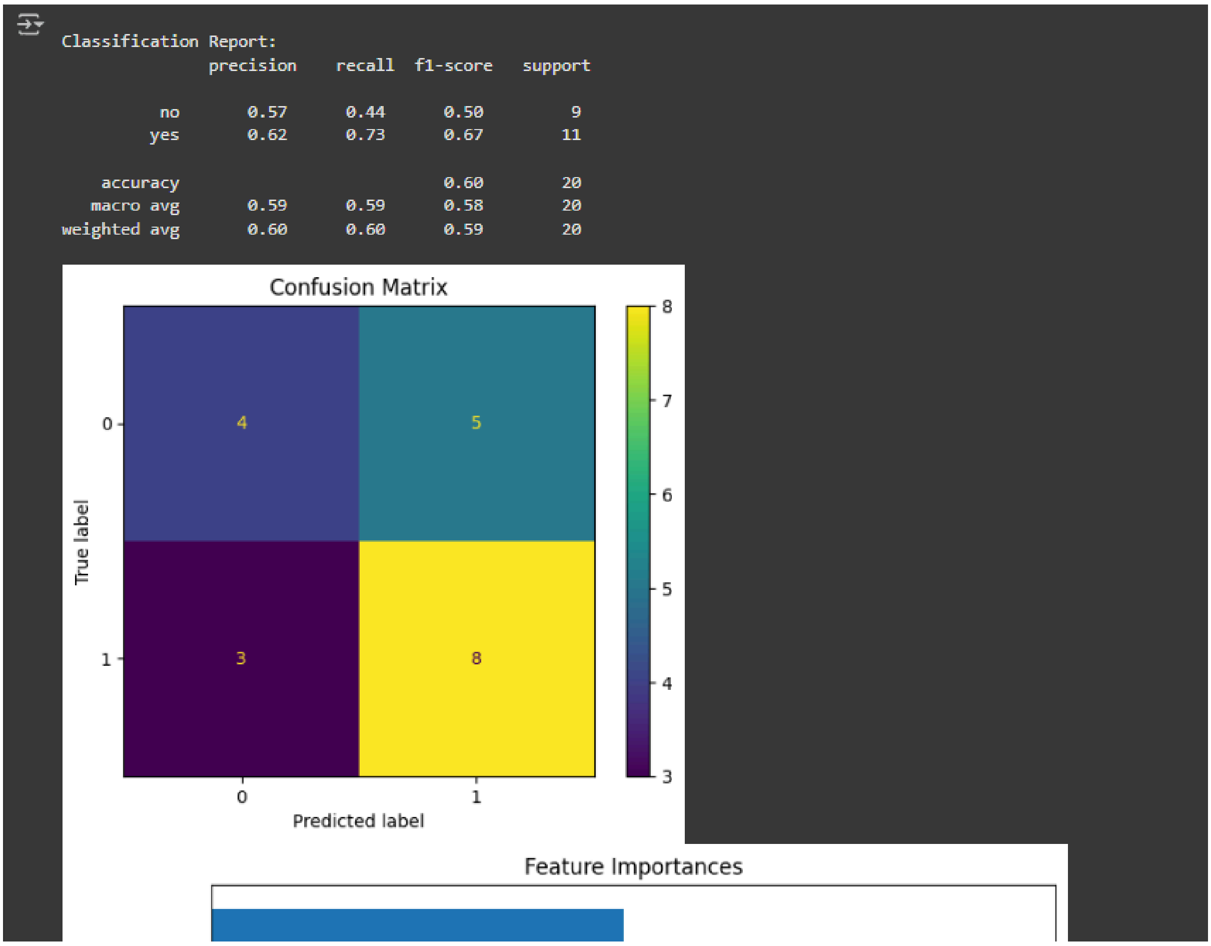
```
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)

# Predict & evaluate
y_pred = clf.predict(X_test)
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.title("Confusion Matrix")
plt.show()

# Feature importance plot
importances = clf.feature_importances_
plt.figure(figsize=(8, 5))
plt.title("Feature Importances")
plt.barh(features, importances)
plt.xlabel("Importance")
plt.tight_layout()
plt.show()
```

OUTPUT OF PROBLEM



```
Classification Report:
              precision    recall  f1-score   support

          no       0.57      0.44      0.50         9
         yes       0.62      0.73      0.67        11

    accuracy                           0.60        20
   macro avg       0.59      0.59      0.58        20
weighted avg       0.60      0.60      0.59        20
```

📚 References

1. **Breiman, L. (2001)**. *Random Forests*. Machine Learning, 45(1), 5–32.
 https://doi.org/10.1023/A:1010933404324

2. **Choi, T.-M., Li, D., & Xu, L. (2019)**. *Return Policy Design in Online Retailing: A Review of Literature*. International Journal of Production Research, 57(15-16), 5155–5175.
 https://doi.org/10.1080/00207543.2018.1530474

3. **Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011)**. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
 https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

4. **Pang, B., & Lee, L. (2008)**. *Opinion Mining and Sentiment Analysis*. Foundations and Trends® in Information Retrieval, 2(1–2), 1–135.

https://doi.org/10.1561/1500000011

5. **Zhang, M., Zhao, K., & Voss, G. B. (2016)**. *Improving Consumer Satisfaction Through Online Product Return Policies: Evidence from Consumer Reviews*. Journal of Retailing, 92(2), 194–204.
https://doi.org/10.1016/j.jretai.2015.11.004

6. **Pandas Documentation** – *Data analysis and manipulation tool.*
https://pandas.pydata.org/docs/

7. **Scikit-learn Documentation** – *Tools for machine learning in Python.*
https://scikit-learn.org/stable/

8. **Matplotlib Documentation** – *Comprehensive 2D plotting library.*
https://matplotlib.org/stable/index.html