Assessment Report

ON

“ Customer Behavior Prediction ”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Ridhima Goyal

Roll Number : 202401100300198

Section: C

Introduction-

Understanding customer purchasing behavior is essential for businesses aiming to personalize marketing strategies, optimize sales, and build long-term customer relationships. Customers often fall into different behavioral categories based on how frequently they shop, how much they spend, and what kinds of products they prioritize.

In this project, we aim to build a machine learning model that can classify customers into two key categories:

- **Bargain Hunters: These customers are typically price-sensitive, make smaller purchases, and may visit frequently looking for deals.**
- **Premium Buyers: These customers tend to make larger purchases, possibly less frequently, and are less sensitive to price.**

By analyzing patterns in historical customer data—including total spending, average purchase amount, and visit frequency—we can predict a customer's category. This kind of classification can help businesses:

- **Tailor marketing campaigns**
- **Offer customized deals**
- **Improve customer retention through personalized services**

We use a logistic regression model to perform the classification, supported by visual analysis and performance metrics such as accuracy, precision, and recall.

Methodology

The process of predicting customer behavior involves multiple steps—ranging from data collection and preprocessing to model training and evaluation. Below is a detailed explanation of each step followed in this project:

◈ 1. Data Collection and Loading

The dataset used in this project contains customer purchase information with the following columns:

- total_spent: Total amount a customer has spent
- avg_purch: Average purchase amount
- visits_per: Number of visits per period
- buyer_type: Label indicating customer type (bargain_hunter or premium_buyer)

The dataset was loaded into a Pandas DataFrame using:

python

CopyEdit

data = pd.read_csv('customer_data.csv')

◈ 2. Data Preprocessing

To ensure data quality and model performance, the following preprocessing steps were taken:

- **Handling Missing Values:**
  Checked and removed any rows containing null values using data.dropna().
- **Label Encoding:**
  Converted the categorical target buyer_type into numeric values:
  - bargain_hunter → 0
  - premium_buyer → 1
- **Feature Scaling:**
  Applied standardization using StandardScaler to normalize the feature values. This ensures that all features contribute equally to the model.

◈ 3. Feature Selection

Three features were selected as inputs for the model:

- total_spent
- avg_purchase_value
- visits_per_month

These features were chosen because they directly reflect spending habits and purchasing behavior.

◈ 4. Model Training

A **logistic regression** model was used due to its simplicity and effectiveness for binary classification problems.

The dataset was split into training and testing sets in an 80:20 ratio using train_test_split.

model = LogisticRegression()

model.fit(X_train, y_train)

5. Model Evaluation

The trained model was evaluated using the test dataset. The following metrics were used:

- **Confusion Matrix**
- **Accuracy**
- **Precision**
- **Recall**
- **F1-score**

These metrics were calculated using confusion_matrix() and classification_report() from sklearn.metrics.

◈ 6. Data Visualization

A scatter plot was created using matplotlib to visualize customer distribution based on:

- total_spent vs. avg_purch
- Colored by buyer type

This helps in understanding how the two customer segments differ visually and supports the model's decision-making.

CODE

```
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.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report

# ---------------------------------------------
# Step 1: Load the dataset
# ---------------------------------------------

data = pd.read_csv('/content/customer_behavior.csv')

print("First 5 rows of the dataset:")
print(data.head())

# ---------------------------------------------
# Step 2: Preprocessing
# ---------------------------------------------

# Check for missing values
print("\nMissing values:")
print(data.isnull().sum())

# Drop rows with missing values (if any)
data.dropna(inplace=True)
```

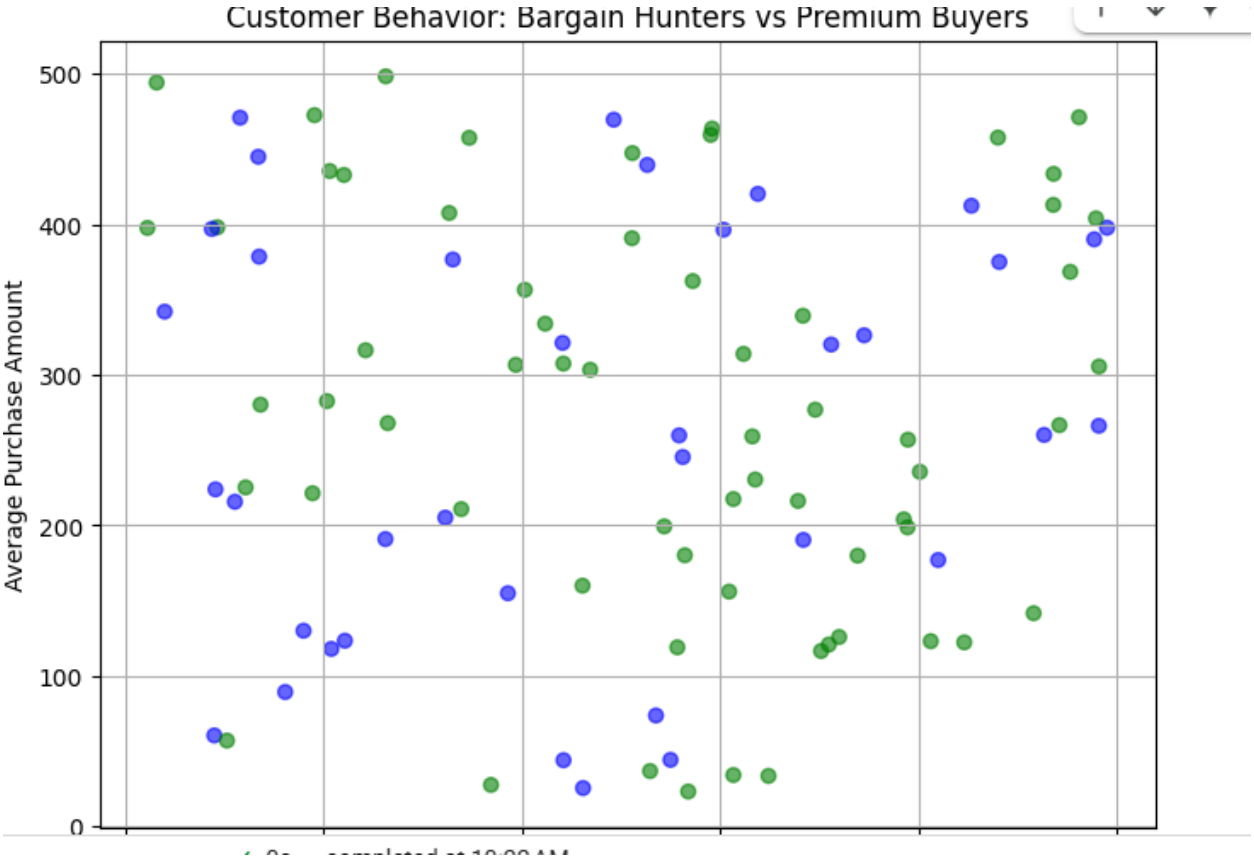
```
# Define features and label
X = data[['total_spent', 'avg_purchase_value', 'visits_per_month']]  # Features
y = data['buyer_type']                          # Target

# Scale the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split into train/test sets
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)

# ---------------------------------------------
# Step 3: Train a logistic regression model
# ---------------------------------------------

model = LogisticRegression()
model.fit(X_train, y_train)

# ---------------------------------------------
# Step 4: Evaluate the model
# ---------------------------------------------

y_pred = model.predict(X_test)

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
print("\nClassification Report:")
print(classification_report(
    y_test, y_pred, target_names=['Bargain Hunter', 'Premium Buyer']
))

# ---------------------------------------------
# Step 5: Visualize customer distribution (optional)
# ---------------------------------------------

# Visualize total_spent vs avg_purch, colored by buyer type
colors = data['buyer_type'].map({'bargain_hunter': 'green', 'premium_buyer': 'blue'})

plt.figure(figsize=(8,6))
plt.scatter(data['total_spent'], data['avg_purchase_value'], c=colors, alpha=0.6)
plt.xlabel('Total Spent')
plt.ylabel('Average Purchase Amount')
plt.title('Customer Behavior: Bargain Hunters vs Premium Buyers')
plt.grid(True)
plt.show()
```

OUTPUT/RESULT

```
First 5 rows of the dataset:
   total_spent  avg_purchase_value  visits_per_month      buyer_type
0  4007.982067          235.560678                 3  bargain_hunter
1  3117.968387          313.883912                13  bargain_hunter
2  4232.062646          122.280804                15  bargain_hunter
3   577.820196          470.747406                20   premium_buyer
4  2839.005107           23.207422                19  bargain_hunter

Missing values:
total_spent           0
avg_purchase_value    0
visits_per_month      0
buyer_type            0
dtype: int64

Confusion Matrix:
[[11  1]
 [ 8  0]]

Classification Report:
                precision    recall  f1-score   support

Bargain Hunter       0.58      0.92      0.71        12
 Premium Buyer       0.00      0.00      0.00         8

      accuracy                           0.55        20
     macro avg       0.29      0.46      0.35        20
```

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

Bargain Hunter       0.58      0.92      0.71        12
 Premium Buyer       0.00      0.00      0.00         8

      accuracy                           0.55        20
     macro avg       0.29      0.46      0.35        20
  weighted avg       0.35      0.55      0.43        20
```


Customer Behavior: Bargain Hunters vs Premium Buyers
Average Purchase Amount
500
400
300
200
100
0

REFERENCES/CREDITS

Python Libraries

- **Pandas** – for data loading, cleaning, and analysis
  https://pandas.pydata.org/
- **NumPy** – for numerical operations
  https://numpy.org/
- **Matplotlib** – for data visualization
  https://matplotlib.org/
- **Scikit-learn (sklearn)** – for machine learning models and evaluation
  https://scikit-learn.org/

📊 **Dataset**

- **Customer Behavior Dataset**
  (Assumed to be custom or internally generated. If sourced from a public domain like Kaggle or UCI, please add the exact source link here.)

💡 **Additional Learning Resources**

- Scikit-learn Documentation: https://scikit-learn.org/stable/user_guide.html
- Towards Data Science Blog (for conceptual understanding): https://towardsdatascience.com/
- Analytics Vidhya Tutorials: https://www.analyticsvidhya.com/