Assessment Report

on

"Customer Behavior Prediction"

SESSION 2025-26

By

Angelina (202401100300041)

KIET Group of Institutions, Ghaziabad

TITLE

INTRODUCTION

NAME=ANGELINA

UNIVERSITY ROLL NO=202401100300041

PROBLEM STATEMENT

Classify customers as 'bargain hunters' or 'premium buyers' using purchase history.

INTRODUCTION

Problem Overview:

The goal is to **predict customer behaviour** by analyzing their **past purchasing patterns**. Specifically, you want to **classify** each customer into one of two categories:

- **Bargain Hunter**: A customer who mostly buys discounted or low-priced products, often seeking the best deals.

- **Premium Buyer**: A customer who prefers high-end or full-priced products, showing less sensitivity to price.

METHODLOGY

Approach Summary:

1. **Data Collection**: Gather customer purchase history (e.g., product type, price, frequency).

2. **Data Preprocessing**: Clean data, handle missing values, normalize features, encode categorical variables.

3. **Feature Engineering**: Extract features like average spend, purchase frequency, preferred price range, product categories.

4. **Labelling**: Define rules or use existing tags to label customers as 'bargain hunters' or 'premium buyers'.

5. **Model Selection**: Choose classification algorithms (e.g., Logistic Regression, Decision Trees, Random Forest, or XG Boost).

6. **Training & Evaluation**: Train model on labelled data; evaluate using metrics like accuracy, precision, recall, F1-score.

7. **Prediction**: Apply the trained model to classify new customers.

8. **Interpretation**: Analyze feature importance to understand behaviour drivers.

CODE

```python
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from xgboost import XGBClassifier

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score, classification_report,
roc_auc_score


# Step 1: Load the data

df = pd.read_csv("/content/customer_behavior.csv")  # Make
sure this is in your working directory


# Step 2: Encode the target variable

le = LabelEncoder()
```

```python
df['buyer_type_encoded'] =
le.fit_transform(df['buyer_type'])  # bargain_hunter -> 0,
premium_buyer -> 1


# Step 3: Select features and target

X = df[['total_spent', 'avg_purchase_value', 'visits_per_month']]

y = df['buyer_type_encoded']


# Step 4: Train-test split

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)


# Step 5: Train XGBoost model

model = XGBClassifier(use_label_encoder=False,
eval_metric='logloss', random_state=42)

model.fit(X_train, y_train)


# Step 6: Make predictions

y_pred = model.predict(X_test)
```

```python
y_prob = model.predict_proba(X_test)[:, 1]


# Step 7: Evaluate the model
cm = confusion_matrix(y_test, y_pred)

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

roc_auc = roc_auc_score(y_test, y_prob)

report = classification_report(y_test, y_pred,
target_names=le.classes_)


# Step 8: Plot confusion matrix
plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt='d', cmap='Greens',
        xticklabels=le.classes_, yticklabels=le.classes_)

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('XGBoost Confusion Matrix')

plt.tight_layout()
```

```python
plt.show()


# Step 9: Print evaluation results

print("Accuracy:", round(accuracy, 2))

print("Precision:", round(precision, 2))

print("Recall:", round(recall, 2))

print("ROC AUC:", round(roc_auc, 2))

print("\nClassification Report:\n", report)
```
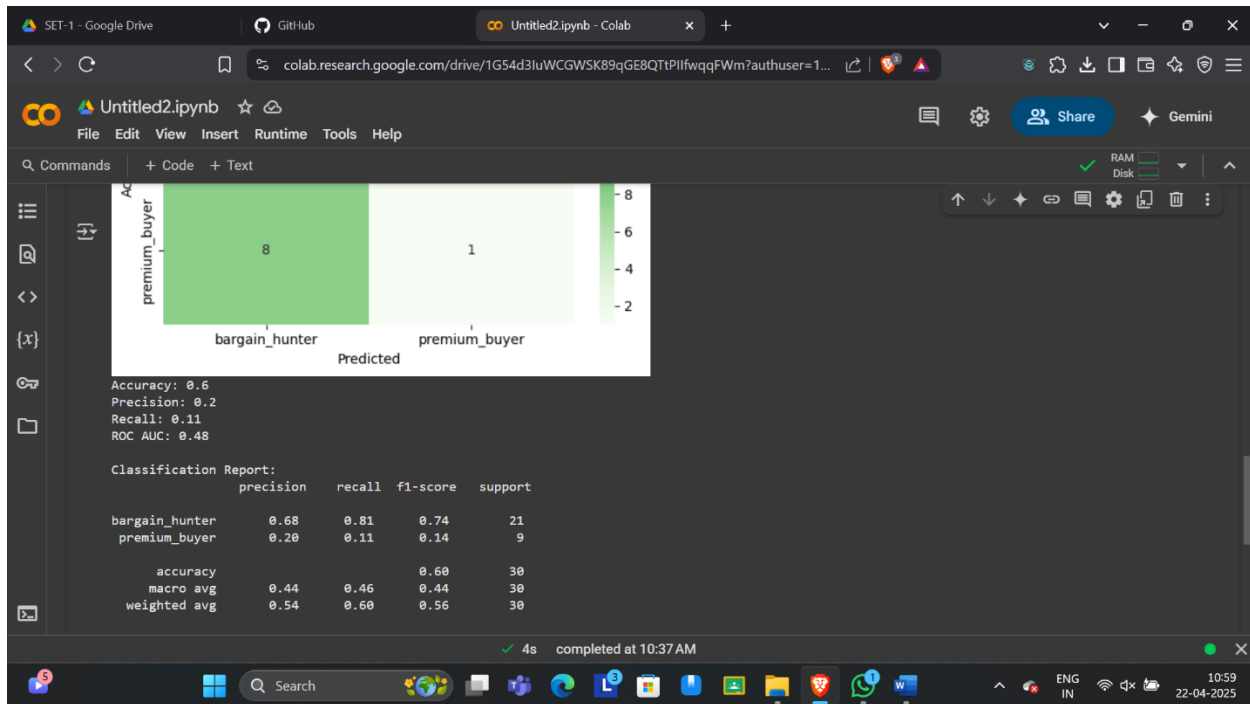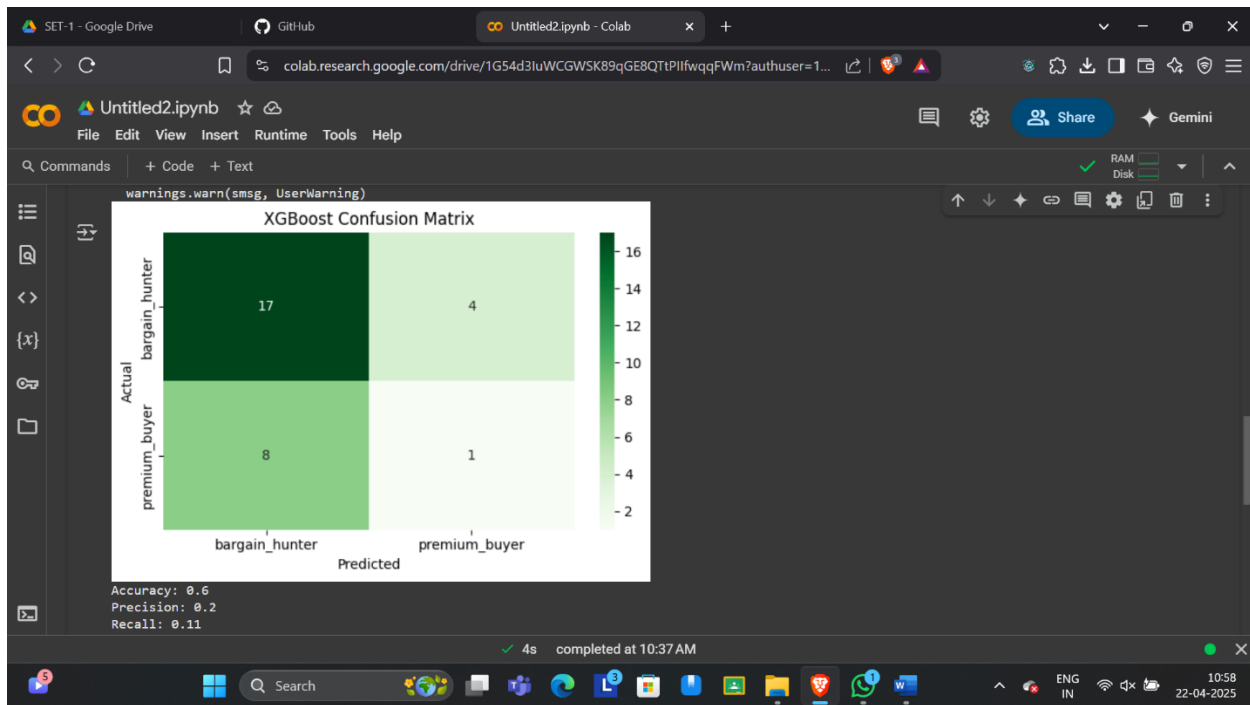
```
warnings.warn(smsg, UserWarning)
```


XGBoost Confusion Matrix

```
Accuracy: 0.6
Precision: 0.2
Recall: 0.11
```

```
Accuracy: 0.6
Precision: 0.2
Recall: 0.11
ROC AUC: 0.48

Classification Report:
                precision    recall  f1-score   support

bargain_hunter      0.68      0.81      0.74        21
premium_buyer       0.20      0.11      0.14         9

      accuracy                          0.60        30
     macro avg       0.44      0.46      0.44        30
  weighted avg       0.54      0.60      0.56        30
```

REFRENCES

total_spent,avg_purchase_value,visits_per_month,buyer_type

4007.982066632277,235.56067760525445,3,bargain_hunter

3117.968386815934,313.88391236568543,13,bargain_hunter

4232.062645588372,122.28080416870793,15,bargain_hunter

577.8201956403975,470.74740606424984,20,premium_buyer

2839.005107088893,23.207422027466134,19,bargain_hunter

2555.3306132847656,390.70448727936457,3,bargain_hunter

3560.3465965204573,320.09828937461225,4,premium_buyer

2718.0237042790113,199.30630450288731,5,bargain_hunter

2749.5993789522104,44.2346933179558,1,premium_buyer

2205.020123522717,321.1935544564541,4,premium_buyer

1613.28867763597,205.09448617551107,12,premium_buyer

2463.0953701357416,469.4082269183452,28,premium_buyer

680.6125579651629,280.15523594708765,27,bargain_hunter

4811.415361017115,471.03272542088393,24,bargain_hunter

3600.6127417703315,125.83217286718596,10,bargain_hunter

4767.088871770449,368.3484369747696,23,bargain_hunter

1210.96417341682,316.2795300073883,25,bargain_hunter

4408.770998864988,374.91595031902966,21,premium_buyer

4063.0709243613383,123.04057470681767,29,bargain_hunter

3418.046821042601,339.19736779413137,26,bargain_hunter

3162.643011817508,259.05938866729707,20,bargain_hunter

1651.093359812563,376.5559879550318,24,premium_buyer

4100.412846328651,176.98142919287253,25,premium_buyer

2557.425761250696,447.22737222332125,13,bargain_hunter

447.3687218672983,60.464631013448184,23,premium_buyer

1633.029488190137,407.45087301977026,14,bargain_hunter

4682.955782872105,433.4334203912436,23,bargain_hunter

4268.333276963746,412.28110945607415,5,premium_buyer

4911.337920575042,266.1086692456525,4,premium_buyer

3177.1675747600807,230.42479041476543,14,bargain_hunter

1312.7993583880186,498.2058975319409,10,bargain_hunter

897.7937434166325,130.0025365827978,17,premium_buyer

1015.880758981462,282.5004636205014,10,bargain_hunter

196.9987259015465,341.9242521835163,26,premium_buyer

2014.342343831102,356.41646521348736,7,bargain_hunter

1037.9956344477582,117.93183955451778,16,premium_buyer

2117.111756237842,333.9171643231046,1,bargain_hunter

4681.47271600995,412.8173919110419,17,bargain_hunter

2209.647026585231,44.00891687413386,21,premium_buyer

4402.124035225691,457.5302207961519,29,bargain_hunter

3066.973356417795,34.16194646969885,9,bargain_hunter

1310.9446178408205,190.86609559442928,6,premium_buyer

1843.688320370136,27.590334191821793,19,bargain_hunter

511.20572552500414,57.029680898999295,13,bargain_hunter

1733.9667086204513,457.360776886438,3,bargain_hunter

3419.950632967317,190.33319875815096,7,premium_buyer

453.3443553542324,223.89033460402538,22,premium_buyer

1106.4669015423437,123.36530306362526,9,premium_buyer

1969.0611029252261,306.6059696896255,22,bargain_hunter

4912.042731378217,305.50554304533904,5,bargain_hunter

3945.8229324985127,198.65618807610525,10,bargain_hunter

3694.085057811309,179.85673846659122,23,bargain_hunter

669.3356869746852,444.7828679266821,17,premium_buyer

3926.879878713458,203.97643502825667,12,bargain_hunter

3947.8171764284743,256.8804762341159,23,bargain_hunter

3045.4032085592085,155.9862859639667,16,bargain_hunter

4952.772731792023,397.6493616114211,3,premium_buyer

1693.7441390303045,210.77511531156398,25,bargain_hunter

4712.234996385073,266.53330234127577,22,bargain_hunter

4582.18246328668675,141.61144162863746,9,bargain_hunter

156.25523277195143,494.07660432380084,29,bargain_hunter

2305.570979726574,159.917955643387,15,bargain_hunter

4887.81074195271,389.9564148346648,5,premium_buyer

605.2946757995715,225.1979121712444,23,bargain_hunter

2784.1450024782653,118.98435444038714,13,bargain_hunter

4896.89178450141,403.86500670313336,14,bargain_hunter

2821.927578118715,180.1312587339407,20,bargain_hunter

1030.0949845635644,435.29280675509493,15,bargain_hunter

2952.69782745369,459.32344876055504,16,bargain_hunter

1101.7588123527794,432.7154409878124,23,bargain_hunter

2646.550888182973,36.80379910726671,13,bargain_hunter

462.80686777050516,397.93920560245067,17,bargain_hunter

3066.865079150583,217.44294391958906,23,bargain_hunter

2632.270701727975,439.38700168746806,8,premium_buyer

3726.790706931169,326.2180460886746,21,premium_buyer

3190.4323086780464,420.14870438910003,2,premium_buyer

2793.0707158007076,259.727249536081,5,premium_buyer

3479.916361315794,276.81224220748777,6,bargain_hunter

805.3153765039311,89.30606342416027,12,premium_buyer

3017.26234016716,396.3139321708906,18,premium_buyer

943.3419538316107,221.30701360014638,22,bargain_hunter

4635.058818474216,260.03845677035963,22,premium_buyer

434.8371137434606,396.84690276593966,1,premium_buyer

2208.836273484887,307.4457920381307,17,bargain_hunter

3509.40764051168,116.52738917248037,11,bargain_hunter

2344.308837174468,303.3116763924285,10,bargain_hunter

551.2667408412215,215.66377183023477,19,premium_buyer

2811.557179020866,245.37226018105406,14,premium_buyer

3243.9963532540373,33.58941171328054,26,bargain_hunter

3548.5823238534877,120.81408776781517,10,bargain_hunter

1928.9730908675187,154.89827906289355,15,premium_buyer

2861.933858040935,362.22080377222676,28,bargain_hunter

2308.3637272006745,25.51638718309535,19,premium_buyer

3393.074589886042,216.21308538998338,4,bargain_hunter

2959.4291398916107,463.53289780435176,2,bargain_hunter

2676.2054758397608,73.70269594357944,9,premium_buyer

1322.66939750741 84,267.88370634493333,10,bargain_hunter

109.8907369663462,397.53074354760656,28,bargain_hunter

953.0002090260034,472.415027609656,11,bargain_hunter

673.6236029784264,378.45278697618465,14,premium_buyer