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: Sachin Kumar

Roll Number: 202401100300208

Section: C

KIET Group of Institutions, Ghaziabad

1. Introduction

Understanding customer behaviour is essential for businesses to tailor their marketing strategies, improve customer engagement, and optimize product offerings. With a growing wealth of customer data, it becomes increasingly important to classify customers based on their purchasing habits and preferences. This project focuses on classifying customers into two groups: bargain hunters and premium buyers.

Bargain hunters are price-sensitive customers who prioritize discounts and deals, while premium buyers are willing to pay a premium for higher-quality products or exclusive offerings. By accurately predicting these customer types, businesses can enhance their targeting strategies, improve customer satisfaction, and increase sales.

In this project, we used a Random Forest classifier, a machine learning model known for its effectiveness in handling complex datasets and providing reliable predictions. We aim to build a model that can predict the customer type based on various features and evaluate its performance through key metrics such as accuracy, precision, and recall.

This report covers the data preprocessing steps, model training, evaluation, and insights gained from the analysis, demonstrating how machine learning can be applied to understand and predict customer behavior.

2. Methodology

The methodology used in this classification problem consists of the following steps:

1. **Data Loading and Cleaning**:
    - Load the dataset and clean column names.
    - Check for and handle any inconsistencies or missing values.
2. **Feature Preparation**:
    - Map target labels (`buyer_type`) to numeric values (0 for "Bargain Hunter," 1 for "Premium Buyer").
    - Validate mapping to ensure there are no unrecognized labels.
3. **Feature-Target Splitting**:
    - Separate features and target variable for model training.
4. **Data Partitioning**:
    - Split the dataset into training and testing subsets using a 70:30 ratio.
5. **Model Training**:
    - Train a Random Forest classifier using the training subset.
6. **Evaluation**:
    - Evaluate the model using metrics such as accuracy, precision, recall, and a confusion matrix.
7. **Visualization**:
    - Plot the matrix heatmap, feature importance, and distribution of customer types.

3. CODE

```
# Import required libraries

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

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

import seaborn as sns

import matplotlib.pyplot as plt


# --------------------------------

# Step 1: Load and clean the dataset

# --------------------------------


# Load the CSV file (ensure it's in the working directory or provide full path)

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


# Remove extra spaces or newline characters from column names

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


# Print the column names to verify

print("Columns in dataset:", df.columns.tolist())


# Display first few rows of data

print("\nSample Data:")

print(df.head())
```

```
# --------------------------------

# Step 2: Prepare target variable

# --------------------------------


# Map target labels to numeric values: 0 for bargain_hunter, 1 for premium_buyer

df['buyer_type'] = df['buyer_type'].map({'bargain_hunter': 0, 'premium_buyer': 1})


# Check if mapping introduced any NaN values (due to typos or unrecognized classes)

if df['buyer_type'].isnull().any():

    raise ValueError("Error: Unknown values found in 'buyer_type'. Please check your data.")


# --------------------------------

# Step 3: Feature-target split

# --------------------------------


# Input features (exclude the target column)

X = df.drop(columns=['buyer_type'])


# Target column

y = df['buyer_type']


# --------------------------------

# Step 4: Split data into training and testing sets

# --------------------------------
```

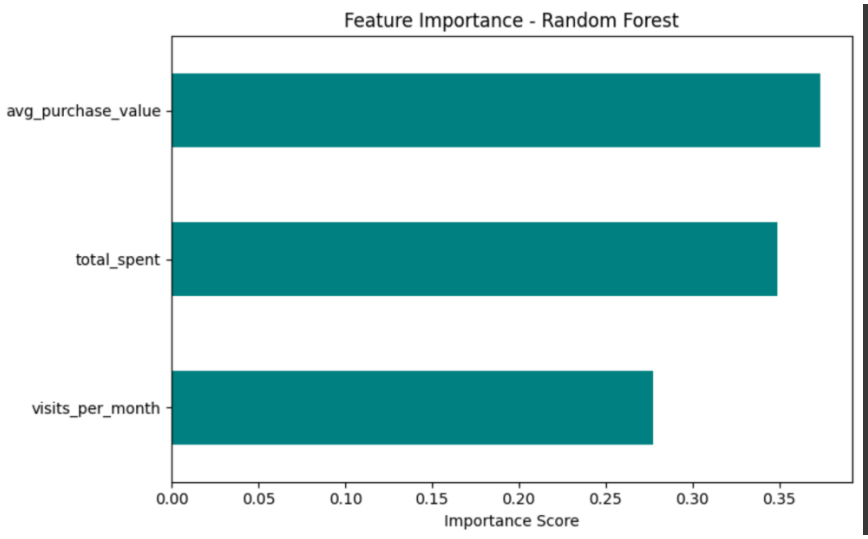
```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

# --------------------------------
# Step 5: Train the Random Forest classifier
# --------------------------------

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

# --------------------------------
# Step 6: Make predictions and evaluate
# --------------------------------

y_pred = clf.predict(X_test)

# Calculate metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print("\nModel Evaluation Metrics:")
print(f"Accuracy : {accuracy:.2f}")
```

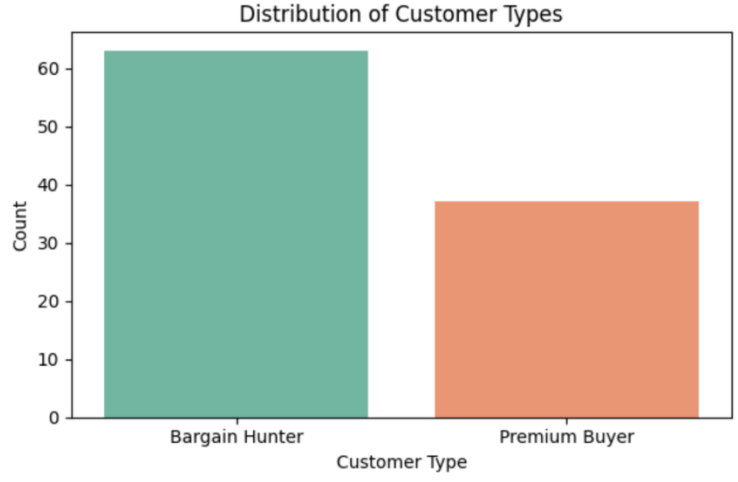
```
print(f"Precision: {precision:.2f}")
print(f"Recall   : {recall:.2f}")

# --------------------------------
# Step 7: Plot confusion matrix heatmap
# --------------------------------

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Bargain Hunter', 'Premium Buyer'],
            yticklabels=['Bargain Hunter', 'Premium Buyer'])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.tight_layout()
plt.show()

# --------------------------------
# Step 8: Plot feature importance
# --------------------------------

feature_importances = pd.Series(clf.feature_importances_, index=X.columns)
feature_importances = feature_importances.sort_values(ascending=True)
```

```
plt.figure(figsize=(8, 5))
feature_importances.plot(kind='barh', color='teal')
plt.title('Feature Importance - Random Forest')
plt.xlabel('Importance Score')
plt.tight_layout()
plt.show()

# ------------------------------
# Step 9: Count plot of buyer types
# ------------------------------

plt.figure(figsize=(6, 4))
sns.countplot(x='buyer_type', data=df, palette='Set2', hue='buyer_type', legend=False)  # Corrected
plt.xticks(ticks=[0, 1], labels=['Bargain Hunter', 'Premium Buyer'])
plt.title('Distribution of Customer Types')
plt.xlabel('Customer Type')
plt.ylabel('Count')
plt.tight_layout()
plt.show()
```

5.Output

```
Sample Data:
   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

Model Evaluation Metrics:
Accuracy : 0.53
Precision: 0.14
Recall   : 0.11
```





6. References

1. **Scikit-learn Documentation**
   Machine learning library used for modeling and evaluation.
   🔗 https://scikit-learn.org
2. **Seaborn Documentation**
   Statistical data visualization library built on top of Matplotlib.
   🔗 https://seaborn.pydata.org
3. **Matplotlib Documentation**
   Core plotting library for Python visualizations.
   🔗 https://matplotlib.org
4. **Random Forests – Breiman, L. (2001)**
   Original paper on the Random Forest algorithm.
   🔗 https://link.springer.com/article/10.1023/A:1010933404324
5. **Kaggle: Customer Segmentation and Classification Datasets**
   Browse datasets related to customer segmentation and classification problems.
   https://www.kaggle.com/datasets?search=customer+classification