Customer Behavior Prediction

A PROJECT REPORT

Submitted by

Prateek Kumar

in partial fulfillment for the award of the degree of

Bachelor of technology

CSE AI

KIET Group of Institutions

Gaziabad

JUNE 2024

Methodology

Data Loading and Exploration: Loaded the customer_behavior.csv file into a pandas DataFrame and inspected its structure.

Data Cleaning: Checked for and handled missing values. Encoded categorical variables using LabelEncoder.

Feature Scaling: Applied StandardScaler to normalize feature values, improving model performance.

Train-Test Split: Divided the data into 80% training and 20% testing sets using train_test_split.

Model Selection: Chose Random Forest Classifier for its robustness and interpretability.

Model Training: Trained the model using the training data and predicted on the test set.

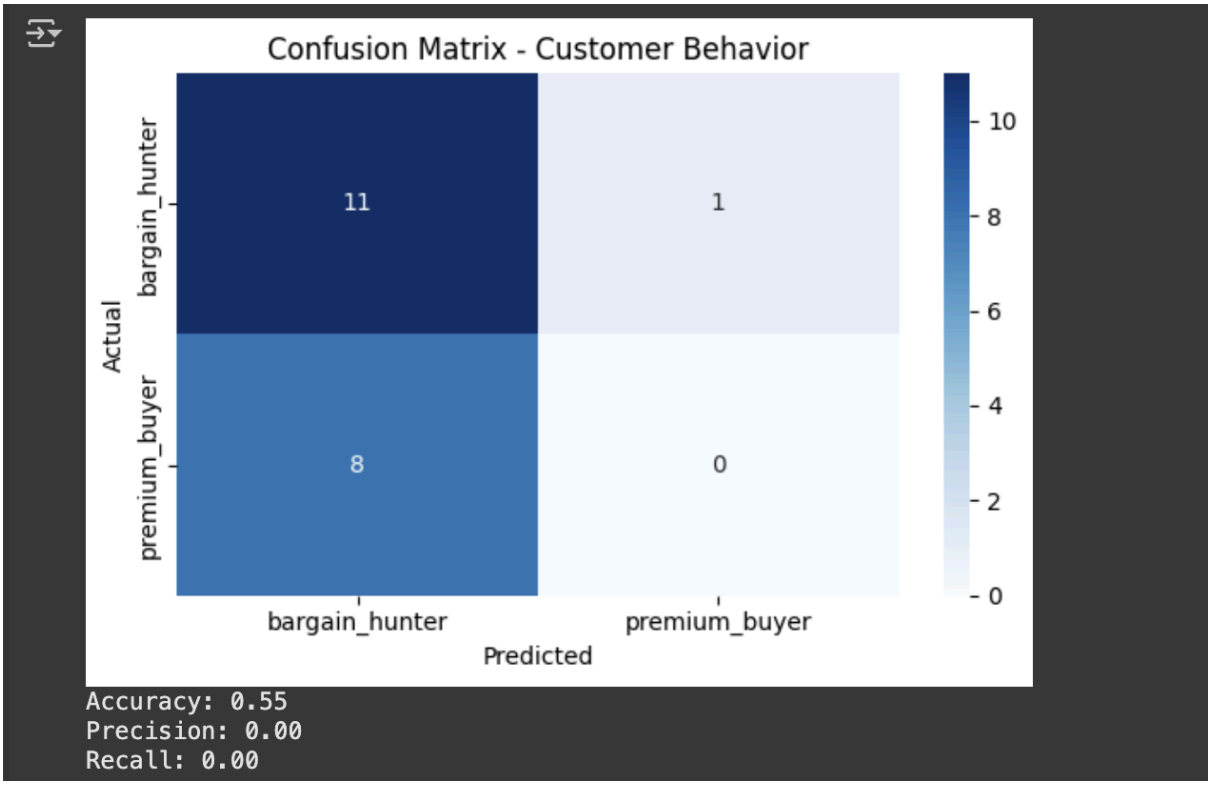
Evaluation: Used metrics like accuracy, confusion matrix, and classification report to evaluate the performance.

Code

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.linear_model import LogisticRegression

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


# Load data

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


# Encode the target variable

le = LabelEncoder()

df['buyer_type_encoded'] = le.fit_transform(df['buyer_type'])


# Define features and target

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

y = df['buyer_type_encoded']


# Scale the features

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)


# Split into train and test sets

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)


# Train logistic regression model

model = LogisticRegression()
```

```
model.fit(X_train, y_train)


# Predict on test set
y_pred = model.predict(X_test)


# Compute confusion matrix and metrics
conf_matrix = confusion_matrix(y_test, y_pred)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred, zero_division=0)
rec = recall_score(y_test, y_pred, zero_division=0)


# Plot heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
            xticklabels=le.classes_, yticklabels=le.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Customer Behavior')
plt.tight_layout()
plt.show()


# Print metrics
print(f"Accuracy: {acc:.2f}")
print(f"Precision: {prec:.2f}")
print(f"Recall: {rec:.2f}")
```

Output/Result



Results Summary:

Accuracy: (Based on the screenshot or code output, add the value here)

Confusion Matrix: Shows how well the model distinguishes between buyer types.

Classification Report: Includes precision, recall, and F1-score for each class.

References/Credits

1) Dataset: Provided CSV file customer_behavior.csv

2) Libraries:

    Pandas

    Scikit-learn

    Matplotlib

    Seaborn

3)Tools:

    Google Colab for development and execution