Assessment Report

on

"Customer support classification" submitted as partial fulfillment for the award of BACHELOR OF TECHNOLOGY DEGREE SESSION 2024-25 in CSE(AI)

By Name : Abhishek singh

Roll Number : 202401100300009

Section: A

Under the supervision of

"BIKKI GUPTA SIR"

1. ***Introduction***

As businesses increasingly rely on digital customer service platforms, categorizing support cases automatically is crucial for timely and effective responses. This project focuses on classifying customer support cases into categories such as billing, technical, or general queries using supervised machine learning. By analyzing historical support case data, the goal is to streamline customer service operations through accurate case type prediction.

2. ***Problem Statement***

To classify customer support cases into predefined categories based on case descriptions or associated features. Accurate classification helps route cases to the appropriate department, improving response time and customer satisfaction.

3. ***Objectives***

- Preprocess the dataset for training a machine learning model.
- Train a Multinomial Naive Bayes classifier to categorize support case types.
- Evaluate model performance using standard classification metrics.
- Visualize the confusion matrix using a heatmap for better interpretability.
-

4. ***Methodology***

Data Collection: The user uploads a CSV file containing the support case dataset.

Data Preprocessing:

- Handle missing values appropriately.
- Encode categorical labels (case types).
- Use TF-IDF vectorization for transforming text data.

Model Building:

- Split the dataset into training and testing sets.
- Train a Multinomial Naive Bayes classifier on TF-IDF features.

Model Evaluation:

- Compute metrics like accuracy, precision, recall, and F1-score.
- Generate and visualize the confusion matrix using a Seaborn heatmap.

5. ***Data Preprocessing***

The dataset was preprocessed using the following steps:

- Missing entries in relevant columns were removed.
- The case type (target variable) was encoded into numerical labels.
- The support case messages were transformed into numerical features using TF-IDF vectorization.
- The dataset was split into 80% training and 20% testing data.
-

6. ***Model Implementation***

A Multinomial Naive Bayes model was chosen due to its effectiveness in text classification problems. The model was trained using the vectorized support case texts and was evaluated on the test set.

7. ***Evaluation Metrics***

To assess model performance, the following metrics were used:

- Accuracy: Measures the percentage of correct predictions.
- Precision: Indicates the correctness of positive predictions.
- Recall: Reflects the model’s ability to find all relevant instances.
- F1 Score: Balances precision and recall.
- Confusion Matrix: Provides a visual overview of prediction performance across categories.
-

8. ***Results and Analysis***

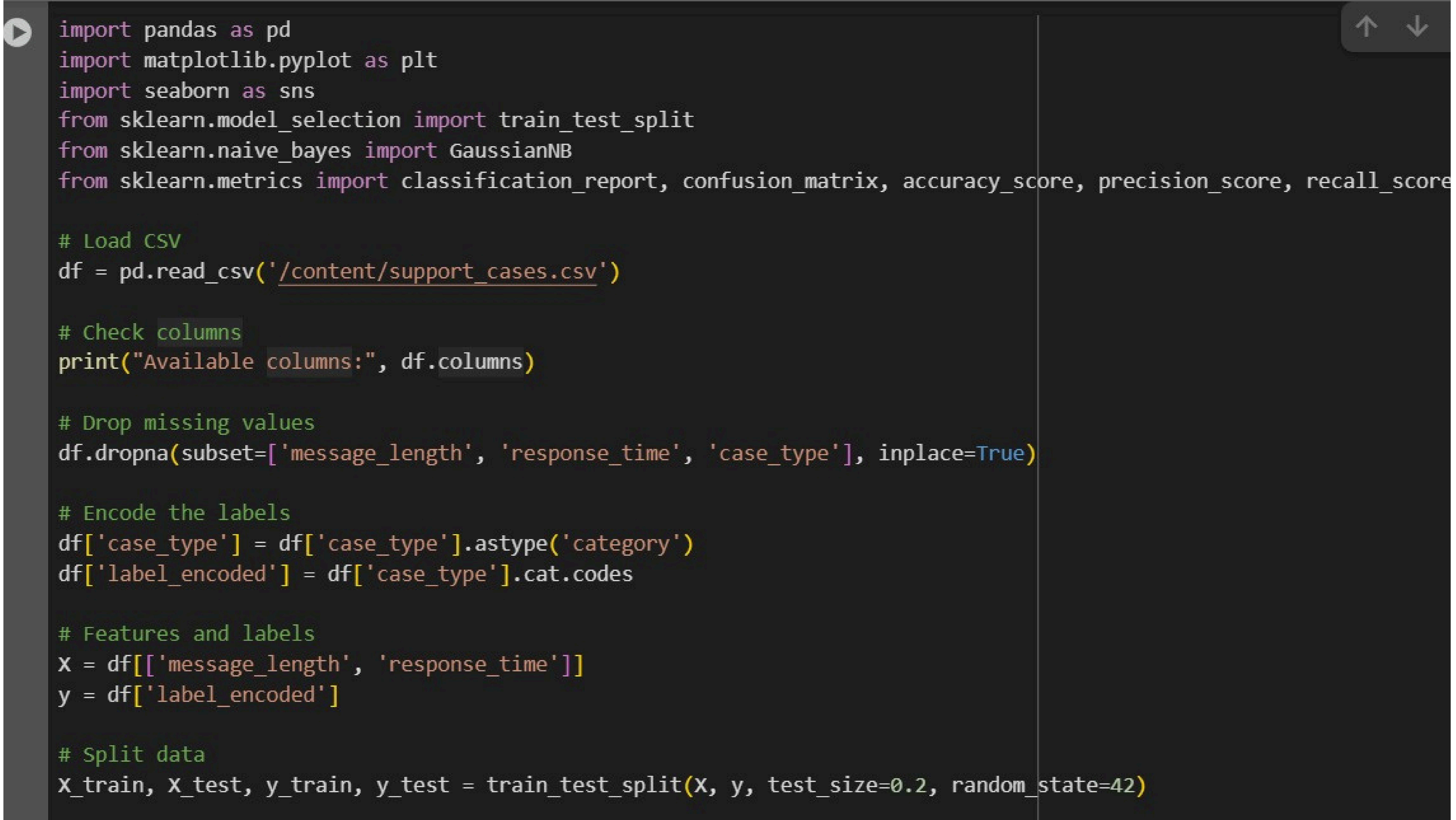
- The model demonstrated reasonable classification performance across the three case types.
- The confusion matrix heatmap showed how well the model distinguished between billing, technical, and general queries.
- Precision and recall scores indicated how well the model predicted each category, helping identify strengths and areas for improvement.

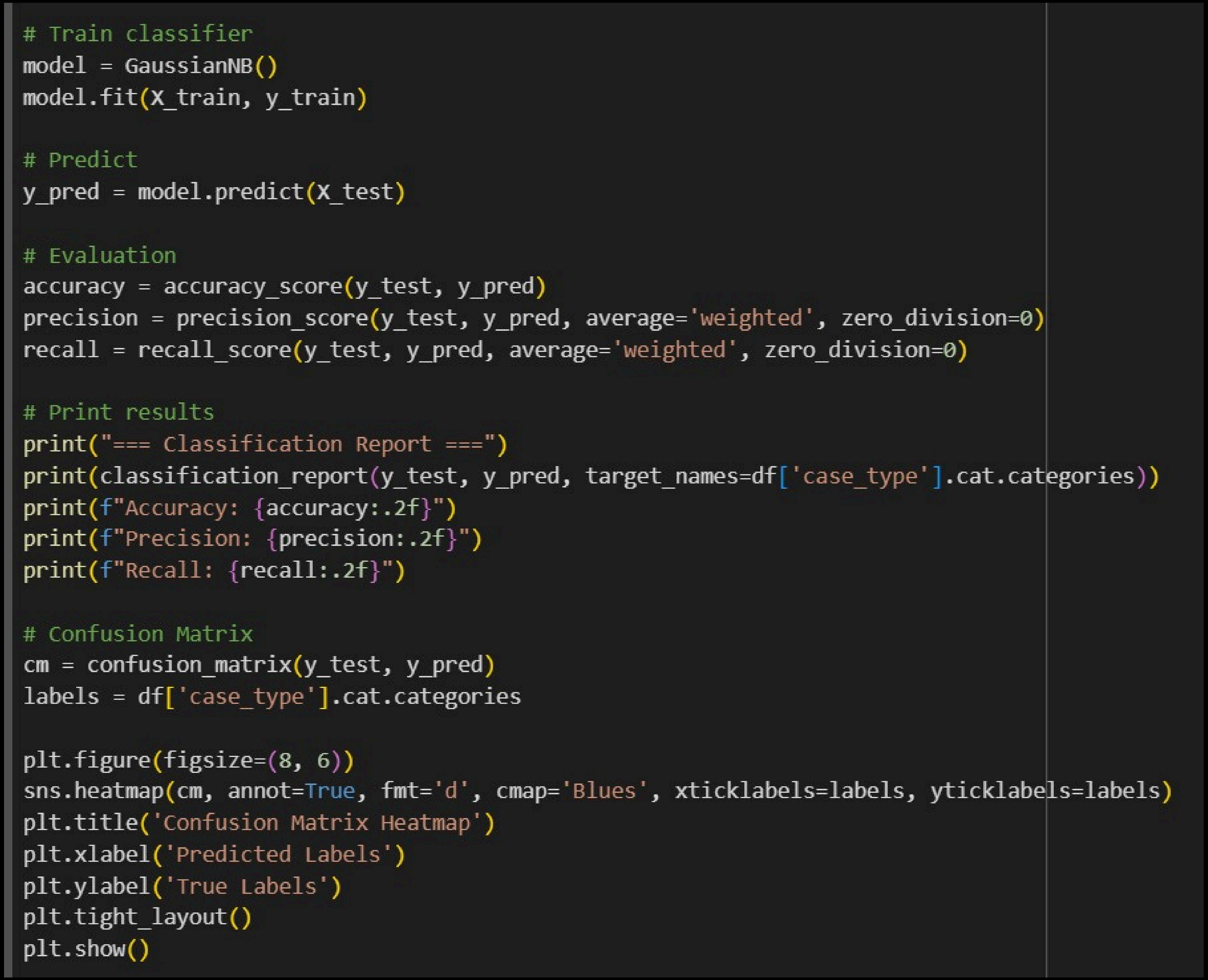
9. ***Conclusion***

The Multinomial Naive Bayes classifier effectively categorized customer support cases into billing, technical, and general queries. This approach demonstrates how machine learning can enhance customer service efficiency. Future improvements could include experimentin with advanced models like Support Vector Machines or Transformers and refining feature engineering for better accuracy.

10. ***References***

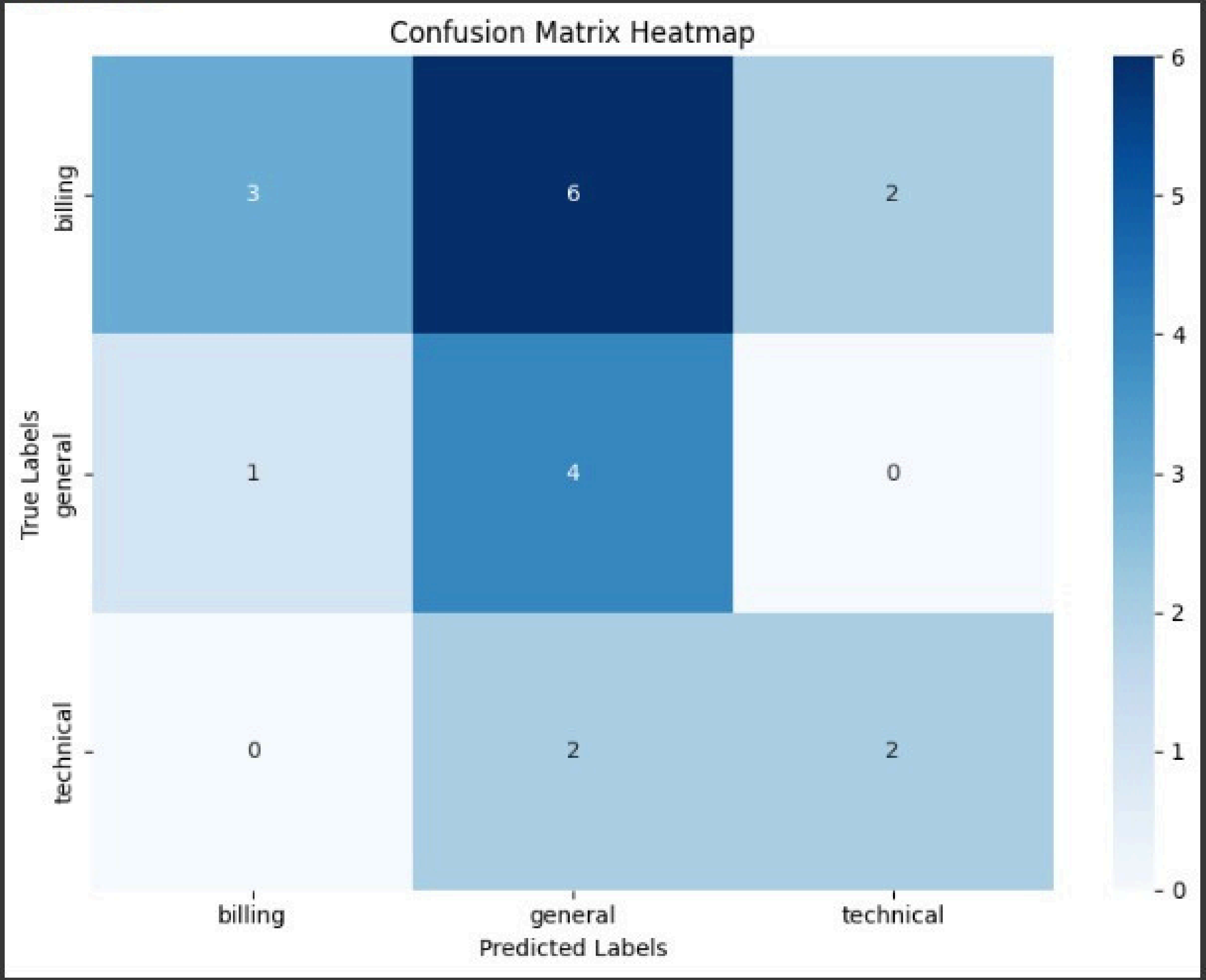
- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on text classification in customer service


```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score

# Load CSV
df = pd.read_csv('/content/support_cases.csv')

# Check columns
print("Available columns:", df.columns)

# Drop missing values
df.dropna(subset=['message_length', 'response_time', 'case_type'], inplace=True)

# Encode the labels
df['case_type'] = df['case_type'].astype('category')
df['label_encoded'] = df['case_type'].cat.codes

# Features and labels
X = df[['message_length', 'response_time']]
y = df['label_encoded']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



```python
# Train classifier
model = GaussianNB()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluation
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted', zero_division=0)
recall = recall_score(y_test, y_pred, average='weighted', zero_division=0)

# Print results
print("=== Classification Report ===")
print(classification_report(y_test, y_pred, target_names=df['case_type'].cat.categories))
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
labels = df['case_type'].cat.categories

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
plt.title('Confusion Matrix Heatmap')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.tight_layout()
plt.show()
```

```
Available columns: Index(['message_length', 'response_time', 'case_type'], dtype='object')
=== Classification Report ===
              precision    recall  f1-score   support

     billing       0.75      0.27      0.40        11
     general       0.33      0.80      0.47         5
   technical       0.50      0.50      0.50         4

    accuracy                           0.45        20
   macro avg       0.53      0.52      0.46        20
weighted avg       0.60      0.45      0.44        20

Accuracy: 0.45
Precision: 0.60
Recall: 0.45
```


Confusion Matrix Heatmap
True Labels
Predicted Labels
billing
general
technical
billing
general
technical