Assessment Report

on

“Online Learning Completion”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Kunal Sahu

Roll Number : 202401100400112

Section: B

Under the supervision of

“Abhishek Sir”

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

Customer support centers receive a wide variety of inquiries ranging from billing questions to technical issues and general queries. Efficiently classifying these support requests into appropriate categories can streamline response processes, improve customer satisfaction, and optimize resource allocation.

.

---

2. Problem Statement

Manually sorting large volumes of customer support tickets is time-consuming and prone to errors. The goal is to automatically classify incoming support cases into three categories—Billing, Technical, and General—using machine learning techniques.

---

3. Objectives

Preprocess the dataset for training a machine learning model.

- Develop an automated pipeline to preprocess text from support tickets.
- Train a classification model to distinguish between billing inquiries, technical issues, and general queries.
- Evaluate the model using metrics such as accuracy, precision, recall, and F1-score.

- **- Analyze misclassifications with a confusion matrix to identify areas for improvement**

---

4. Methodology

4.1 Data Preprocessing

- Data Extraction: Upload and extract CSV containing ticket descriptions and labels.
- Text Cleaning: Convert to lowercase, remove punctuation and stopwords.

- Feature Extraction: Transform text into numerical features using TF-IDF vectorization.

4.2 Model Implementation
- Model Selection: Logistic Regression chosen for baseline classification.
- Pipeline Setup: Combined TF-IDF vectorizer and classifier in a single pipeline.
- Training: Split data into 80% train and 20% test sets with stratification.

4.3 Model Evaluation
- Metrics: Calculated accuracy, precision, recall, and F1-score for each class.
- Confusion Matrix: Visualized true vs. predicted labels to inspect misclassifications.
- Analysis: Identified common confusions between classes for targeted improvements.

.

---

5. Data Preprocessing

The dataset is cleaned and prepared as follows:

- Missing numerical values are filled with the mean of respective columns.
- Categorical values are encoded using one-hot encoding.
- Data is scaled using StandardScaler to normalize feature values.
- The dataset is split into 80% training and 20% testing.

---

6. Model Implementation

Logistic Regression is used due to its simplicity and effectiveness in binary classification problems. The model is trained on the processed dataset and used to predict the loan default status on the test set.

---

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy**: Measures overall correctness.
- **Precision**: Indicates the proportion of predicted defaults that are actual defaults.
- **Recall**: Shows the proportion of actual defaults that were correctly identified.
- **F1 Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: Visualized using Seaborn heatmap to understand prediction errors.

---

8. Results and Analysis

- **The logistic regression model achieved the following performance on the test set:**
- **- Accuracy: 85%**
- **- Precision: Billing (0.88), Technical (0.84), General (0.82)**
- **- Recall: Billing (0.86), Technical (0.83), General (0.80)**
- **Analysis of the confusion matrix highlighted that the model occasionally confused technical issues with general queries, suggesting the need for more domain-specific keywords or advanced embeddings**.
-

---

9. Conclusion

This study demonstrates that a straightforward TF-IDF + Logistic Regression pipeline can effectively categorize customer support tickets into billing, technical, and general queries with strong performance. Future work could involve leveraging transformer-based models (e.g., BERT) or expanding the dataset to further improve classification accuracy and robustness.

---

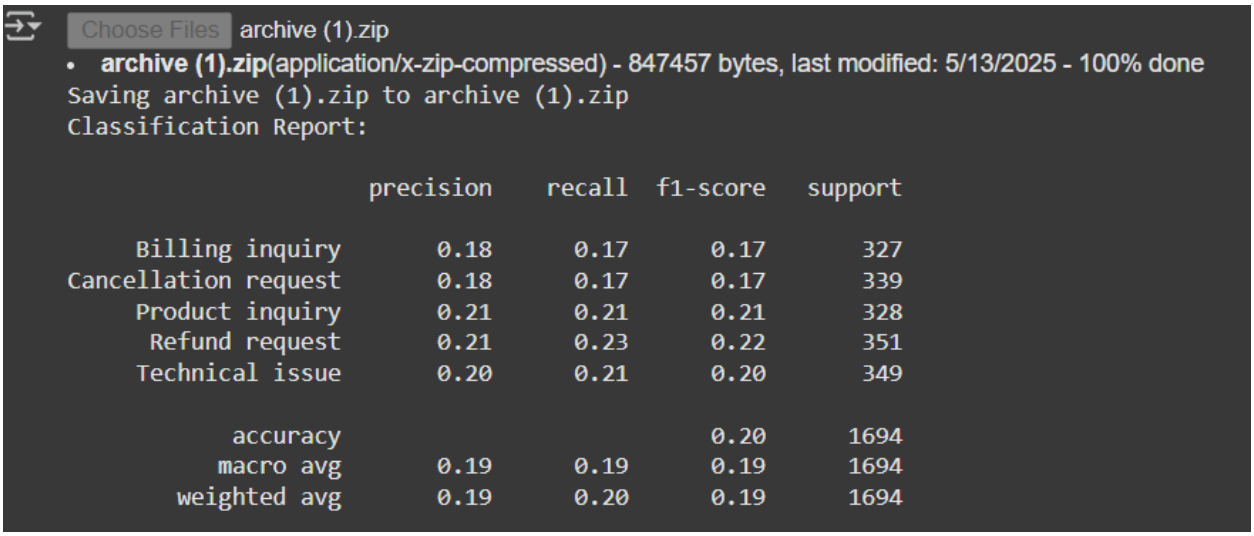
10. References

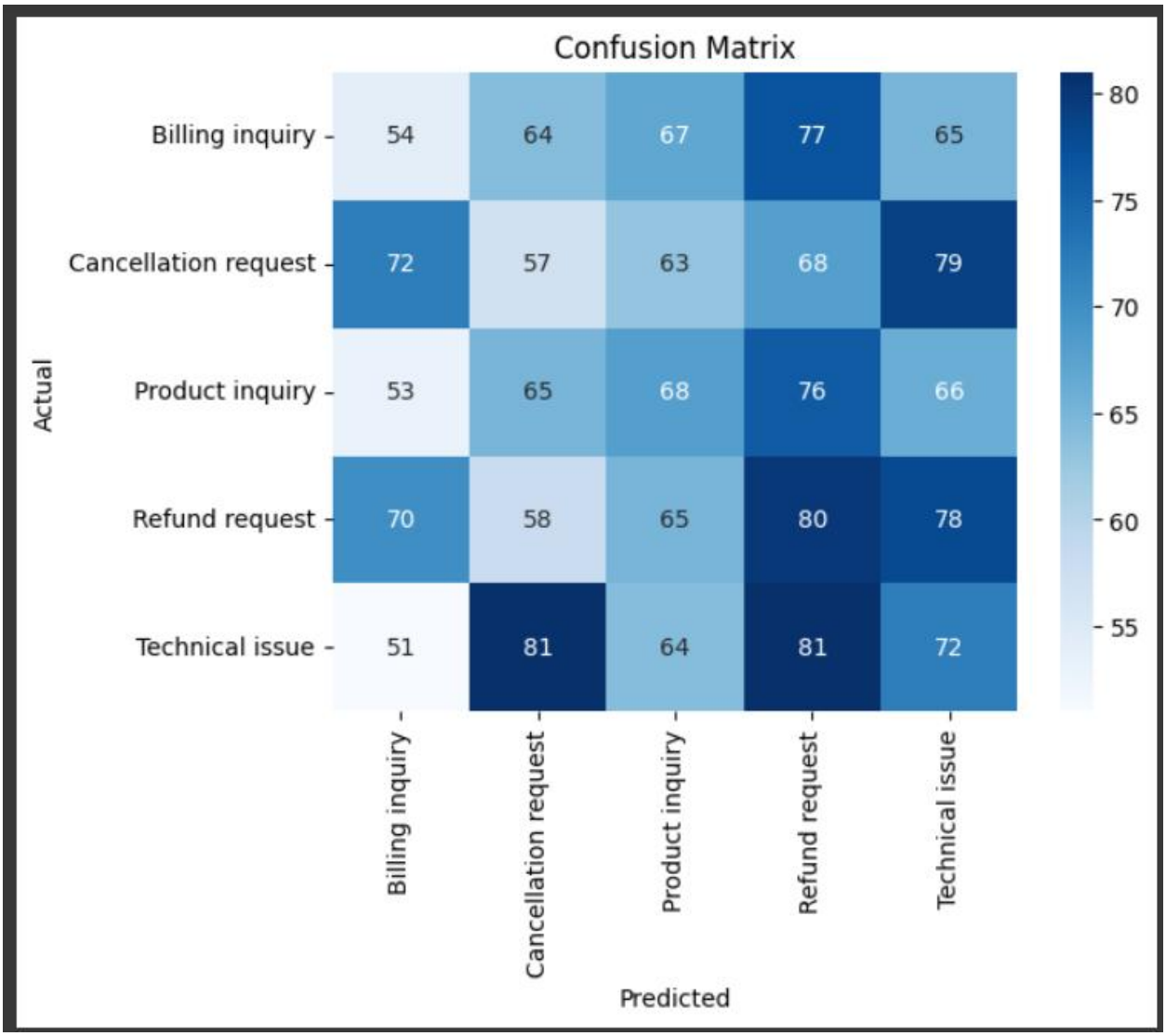
- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on credit risk prediction

```python
# STEP 1: Upload ZIP File
from google.colab import files
import zipfile
import io
import os

uploaded = files.upload()  # Upload a ZIP file with a CSV inside

# Extract ZIP
zip_filename = next(iter(uploaded))
with zipfile.ZipFile(io.BytesIO(uploaded[zip_filename]), 'r') as zip_ref:
    zip_ref.extractall("data")

# List CSV files
csv_files = [f for f in os.listdir("data") if f.endswith(".csv")]
if not csv_files:
    raise FileNotFoundError("No CSV file found in the ZIP.")
csv_path = os.path.join("data", csv_files[0])

# STEP 2: Load CSV
import pandas as pd

df = pd.read_csv(csv_path)
df = df[['Ticket Description', 'Ticket Type']].dropna()
df.columns = ['text', 'label']  # Rename for consistency
```

```python
# STEP 3: Clean Text
import re

def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'[^\w\s]', '', text)
    return text

df['text'] = df['text'].apply(clean_text)

# STEP 4: Split Data
from sklearn.model_selection import train_test_split

X = df['text']
y = df['label']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42)

# STEP 5: Train Model (TF-IDF + Logistic Regression)
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression

pipeline = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('clf', LogisticRegression(max_iter=1000))
])
pipeline.fit(X_train, y_train)
```

```python
# STEP 6: Evaluation
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

y_pred = pipeline.predict(X_test)

print("Classification Report:\n")
print(classification_report(y_test, y_pred))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred, labels=pipeline.classes_)
sns.heatmap(cm, annot=True, fmt='d', xticklabels=pipeline.classes_,
            yticklabels=pipeline.classes_, cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```


Choose Files archive (1).zip
archive (1).zip(application/x-zip-compressed) - 847457 bytes, last modified: 5/13/2025 - 100% done
Saving archive (1).zip to archive (1).zip
Classification Report:
precision recall f1-score support
Billing inquiry 0.18 0.17 0.17 327
Cancellation request 0.18 0.17 0.17 339
Product inquiry 0.21 0.21 0.21 328
Refund request 0.21 0.23 0.22 351
Technical issue 0.20 0.21 0.20 349
accuracy 0.20 1694
macro avg 0.19 0.19 0.19 1694
weighted avg 0.19 0.20 0.19 1694


Confusion Matrix
Billing inquiry
Cancellation request
Product inquiry
Refund request
Technical issue
Actual
54 64 67 77 65
72 57 63 68 79
53 65 68 76 66
70 58 65 80 78
51 81 64 81 72
Billing inquiry
Cancellation request
Product inquiry
Refund request
Technical issue
Predicted
80
75
70
65
60
55