KIET Group Institute of Technology

Aryan Kumar Srivastava CSE AI

- 2024 / 2025 -

CUSTOMER SUPPORT CASE TYPE CLASSIFICATION

🟦 2. INTRODUCTION

Customer service centers often receive thousands of support queries daily. Manually sorting these cases slows down response time and increases operational costs. This project aims to automate the classification of support cases into billing, technical, or general categories using machine learning techniques. The classification will improve ticket management and customer satisfaction.

🧪 3. Methodology

⚙️ Methodology

The methodology adopted includes the following steps:

Data Loading and Cleaning:

Loading the dataset from a CSV file.

Text preprocessing: removing special characters, converting text to lowercase, and normalizing whitespace.

Feature Extraction:

Converting text data into numerical features using TF-IDF (Term Frequency-Inverse Document Frequency).

Model Building:

Training two machine learning models: Multinomial Naive Bayes and Logistic Regression.

Model Evaluation:

Evaluating performance using metrics like accuracy, precision, recall, F1-score, and confusion matrix.

Classification Function:

A user-defined function classify_support_case() to predict the category of new support cases.

Visualization:

Visualizing model performance through confusion matrices and metric charts.

4. CODE

```python
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# Preprocessing Function
def clean_text(text):
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    text = text.lower()
    text = re.sub(r'\s+', ' ', text).strip()
    return text
```

```python
# Load Dataset
df = pd.read_csv('/content/support_cases.csv')

# Clean Text Data
df['case_type'] = df['case_type'].apply(clean_text)

# Feature Extraction
vectorizer = TfidfVectorizer(max_features=5000)
X = vectorizer.fit_transform(df['case_type'])
y = df['category']

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model 1: Multinomial Naive Bayes
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)
nb_pred = nb_model.predict(X_test)

# Model 2: Logistic Regression
lr_model = LogisticRegression()
lr_model.fit(X_train, y_train)
lr_pred = lr_model.predict(X_test)
```

```python
# Model Evaluation
print("Naive Bayes Classification Report:\n", classification_report(y_test, nb_pred))
print("Logistic Regression Classification Report:\n", classification_report(y_test, lr_pred))

# Confusion Matrix
conf_mat = confusion_matrix(y_test, lr_pred)
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Logistic Regression')
plt.show()

# Classification Function
def classify_support_case(message):
    clean_msg = clean_text(message)
    vectorized_msg = vectorizer.transform([clean_msg])
    prediction = lr_model.predict(vectorized_msg)[0]
    confidence = max(lr_model.predict_proba(vectorized_msg)[0])
    return {"category": prediction, "confidence": round(confidence, 2)}
```

🙌 5. Credits

- Project By: Aryan Kumar Srivastava
- To : Mr. Bikki Kumar

Credits :

- Scikit-learn Documentation
- Text Analytics - Scikit-learn Tutorial
- TF-IDF Concept - Wikipedia
- Google Colab for providing a free environment to run ML models
- Pandas Library