Text Classification Using Logistic Regression

Natural Language Processing Project using BBC Text Dataset

h by hager ayman

Project Overview

- **Goal:**

  Build a machine learning model to automatically classify news articles into one of five categories.

- **Problem Statement:**

  With the increasing volume of news content, automated classification helps in organizing and analyzing text efficiently.

Dataset Description:

Dataset Name: BBC Text Dataset
Source: News articles from the BBC
Total Records: ~2225 articles
Target Classes:

- Tech
- Business
- Sport
- Entertainment
- Politics

Each record contains:

- category (label)
- text (news content

Data Preprocessing

To prepare the text for model training, we performed the following steps:

- Text Cleaning (removing punctuation, lowercasing)
- Tokenization
- Stopwords Removal
- Vectorization using **TF-IDF**
- Splitting into train/test se

Why Logistic Regression?

1

Reasons for choosing Logistic Regression:

- Simple yet effective for linear classification tasks
- Interpretable and fast to train
- Performs well with sparse data like TF-IDF vectors
- Suitable baseline for text classification problem

Model Training

Algorithm Used: Logistic Regression

Library: Scikit-learn(`sklearn.linear_model.LogisticRegression`)

Training/Test Split: 80% training / 20% testing

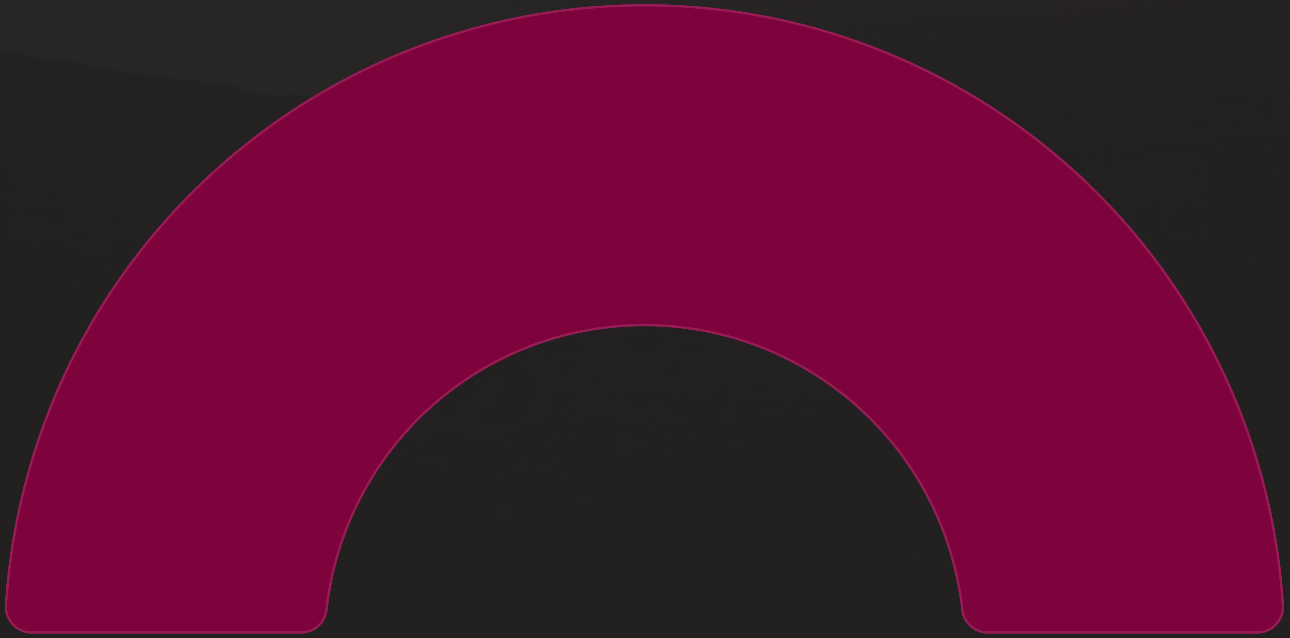
Features: TF-IDF vectors of text data

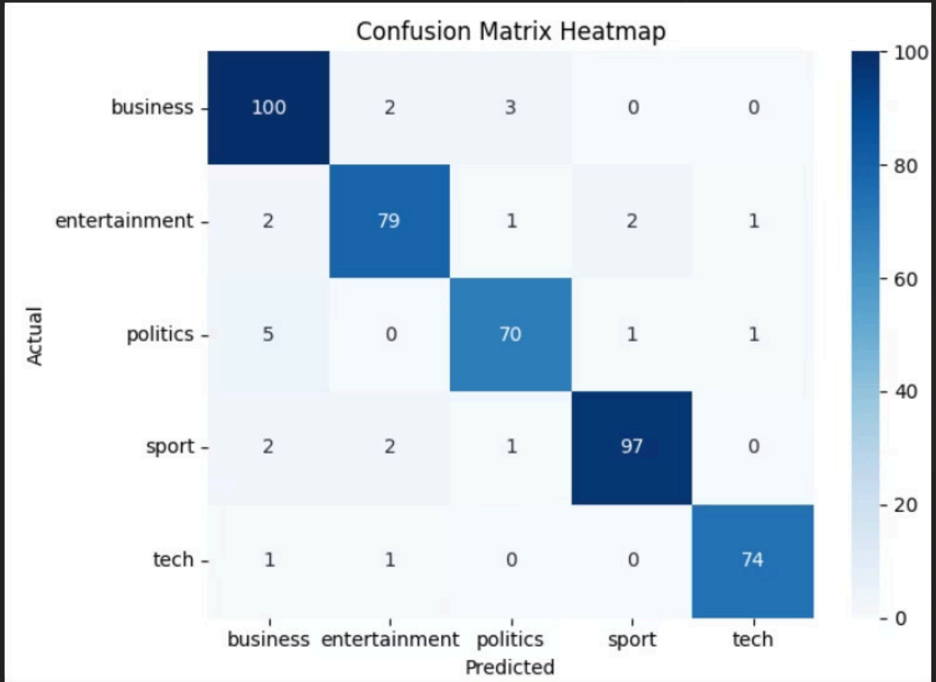
Evaluation Metrics

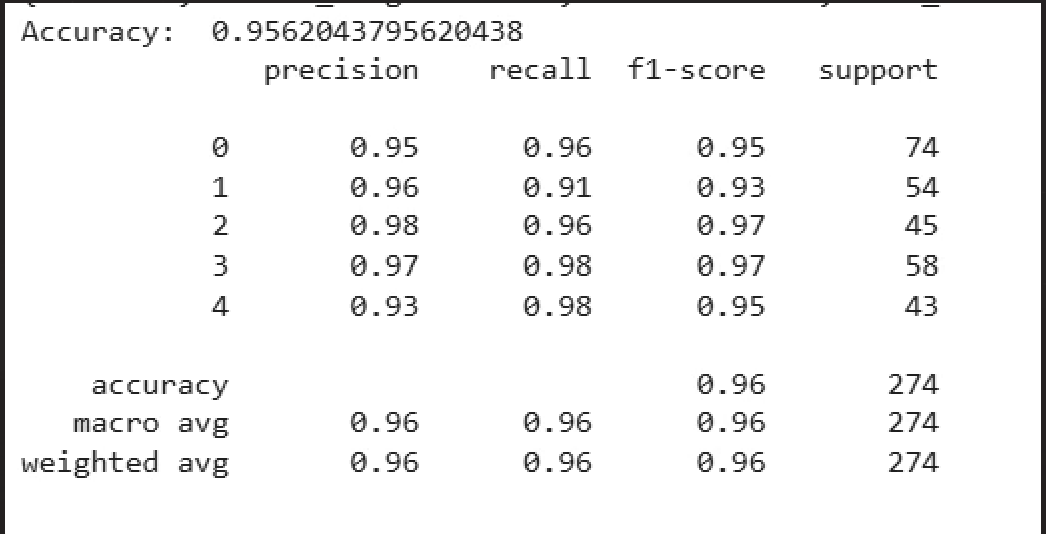
We used the following metrics to evaluate performance:

- ✅ Accuracy
- 📊 Precision
- 🔁 Recall
- 🎯 F1-Score

Each metric was calculated per class using `classification_report` from scikit-learn.







```
Accuracy:  0.9562043795620438
              precision    recall  f1-score   support

           0       0.95      0.96      0.95        74
           1       0.96      0.91      0.93        54
           2       0.98      0.96      0.97        45
           3       0.97      0.98      0.97        58
           4       0.93      0.98      0.95        43

    accuracy                           0.96       274
   macro avg       0.96      0.96      0.96       274
weighted avg       0.96      0.96      0.96       274
```

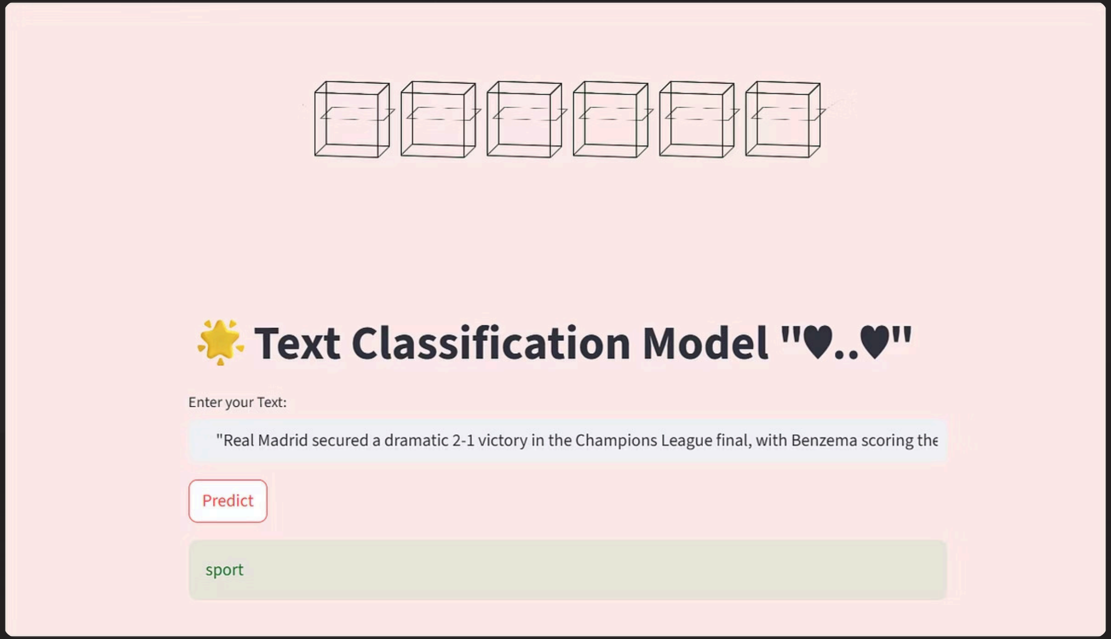
Deployment using Streamlit

Make the NLP model accessible through a simple web app.

Tool Used: 🟣 Streamlit

Features of the Web App:

- User can input or paste news text
- The model predicts the category (e.g., *Politics*, *Tech*)
- Shows prediction result instantly



🌟 Text Classification Model "♥..♥"

Enter your Text:

"Real Madrid secured a dramatic 2-1 victory in the Champions League final, with Benzema scoring the

Predict

sport