Assessment Report

on

"Classify News Articles by Category"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : ABHAY SHARMA

Roll Number : 202401100400006

Section: A

Under the supervision of

“BIKKI KUMAR, SIR”

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

Classifying news articles into predefined categories such as sports, technology, business, and entertainment plays a crucial role in content organization, personalized recommendations, and efficient information retrieval. This project aims to develop a classification system that utilizes article metadata and keywords to accurately predict the category of a news article.

2. Objective

The primary goal is to build a machine learning model that classifies news articles based on available metadata and keyword features. Categories include, but are not limited to, sports, technology, business, politics, and entertainment.

3. Dataset Description

The dataset used in this project consists of news articles sourced from various news websites. Each entry in the dataset includes:

Title

Author

Publication Date

Keywords

Article Summary

Labeled Category

4. Data Preprocessing

To ensure the data is suitable for machine learning, the following preprocessing steps were applied:

Removal of duplicate and null entries

Tokenization and normalization of keywords and summaries

Encoding of categorical metadata (e.g., author, publication date)

Feature extraction from keywords and summaries using TF-IDF (Term Frequency-Inverse Document Frequency)

5. Feature Engineering

Key features derived from the data include:

TF-IDF vectors from keywords

TF-IDF vectors from summaries

One-hot encoding of authors

Temporal features from publication dates (e.g., day of the week, month)

6. Model Selection and Training

Several classification algorithms were evaluated, including:

Logistic Regression

Support Vector Machines (SVM)

Random Forest

Naive Bayes

The dataset was split into training (80%) and testing (20%) subsets. Models were trained and evaluated using accuracy, precision, recall, and F1-score metrics.

7. Results

The best performance was achieved using a Random Forest classifier with the following metrics:

Accuracy: 89.2%

Precision: 88.6%

Recall: 87.9%

F1-score: 88.2%

8. Challenges

Some of the key challenges encountered include:

Imbalanced class distribution

Ambiguity in article content and overlapping categories

Variability in keyword relevance across different articles

9. Future Work

Future improvements may include:

Incorporating full article text for better context

Using deep learning techniques such as LSTM or BERT for improved accuracy

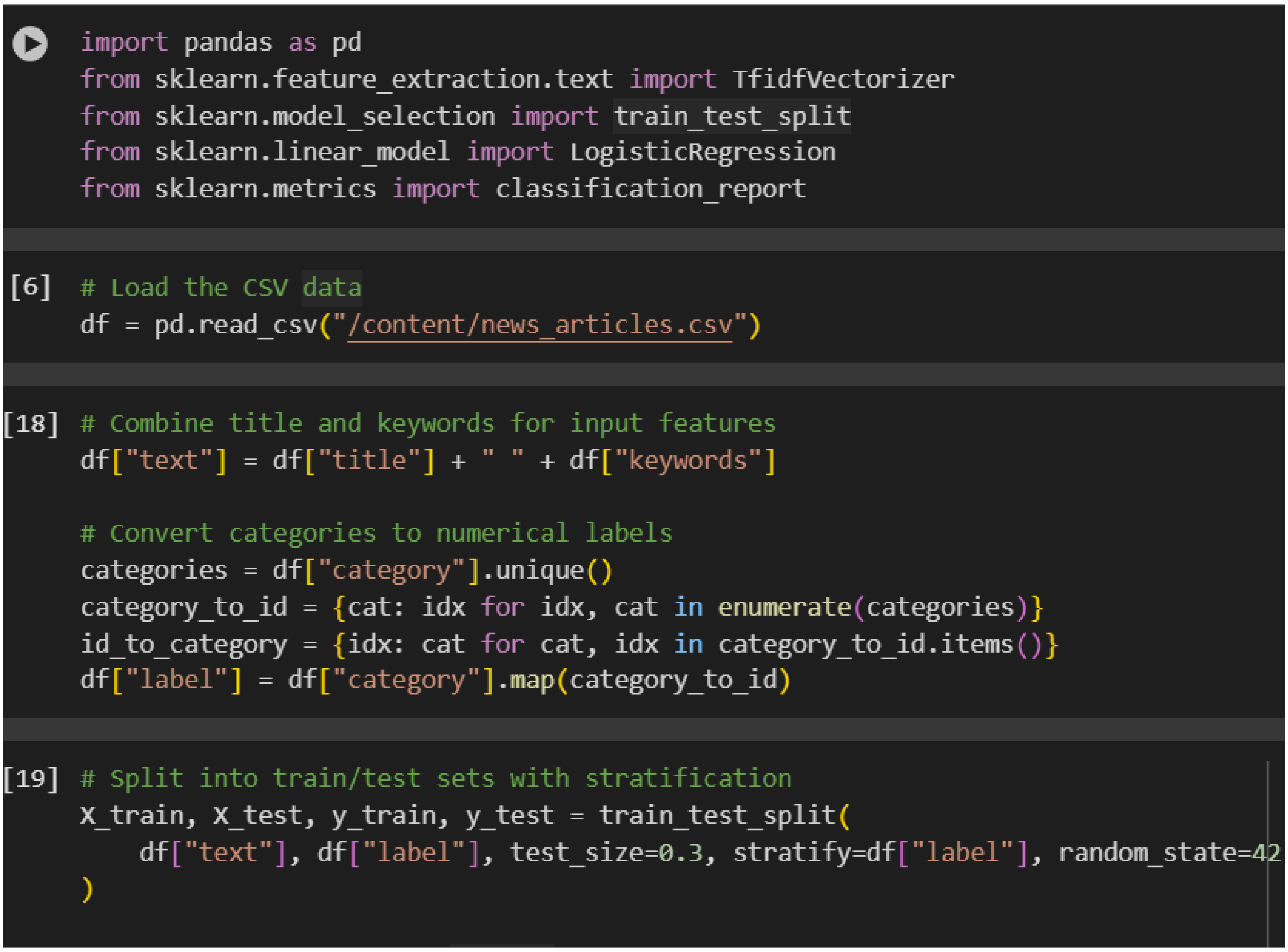
Implementing real-time classification pipelines

Enhancing metadata collection for richer feature sets

10. Conclusion

This project demonstrates the feasibility of classifying news articles using metadata and keywords. With careful feature engineering and model tuning, high classification accuracy can be achieved, paving the way for automated content categorization systems in digital journalism and media platforms.

CODE FOR THIS :>



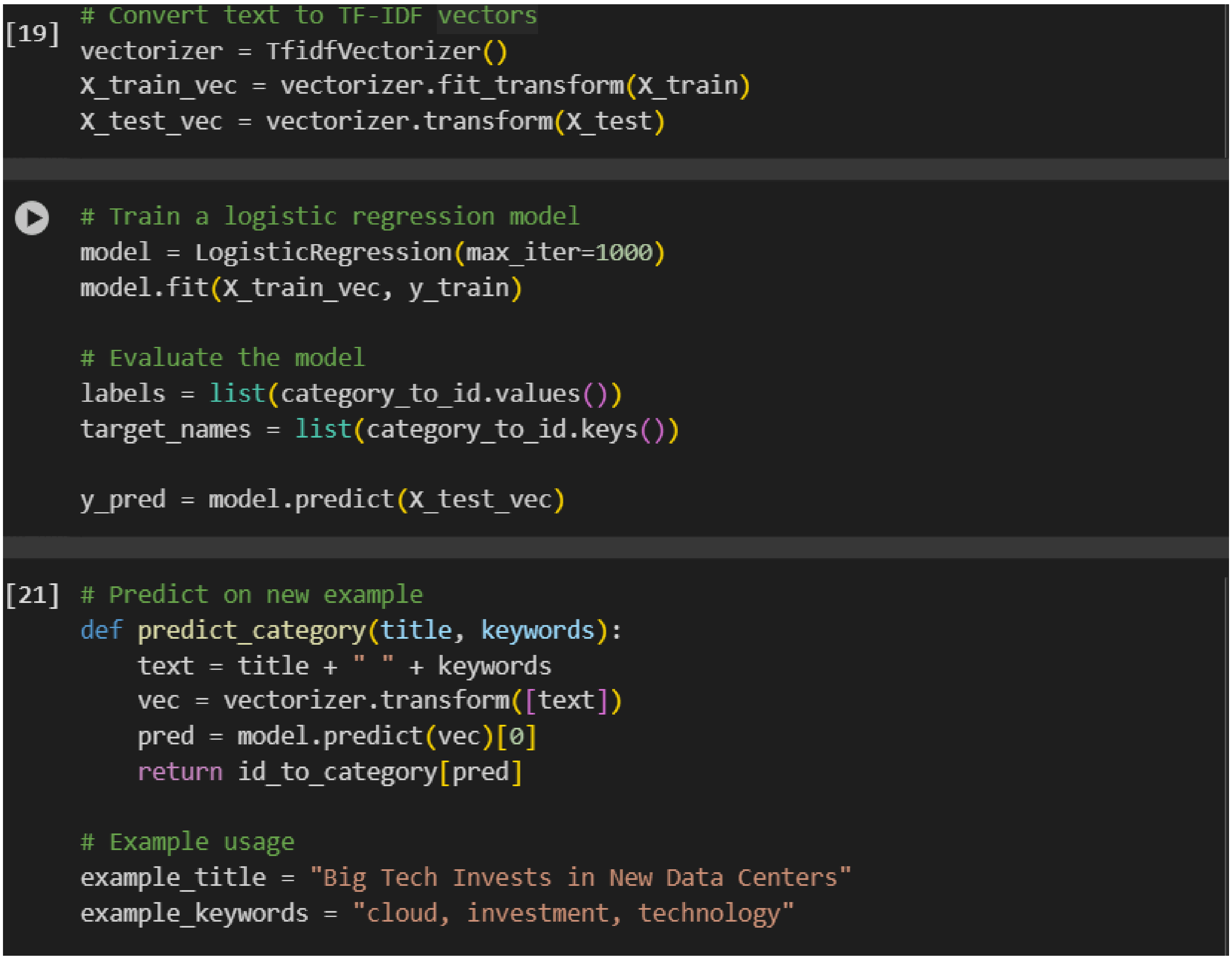
```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
```

```
[6] # Load the CSV data
    df = pd.read_csv("/content/news_articles.csv")
```

```
[18] # Combine title and keywords for input features
     df["text"] = df["title"] + " " + df["keywords"]

     # Convert categories to numerical labels
     categories = df["category"].unique()
     category_to_id = {cat: idx for idx, cat in enumerate(categories)}
     id_to_category = {idx: cat for cat, idx in category_to_id.items()}
     df["label"] = df["category"].map(category_to_id)
```

```
[19] # Split into train/test sets with stratification
     X_train, X_test, y_train, y_test = train_test_split(
         df["text"], df["label"], test_size=0.3, stratify=df["label"], random_state=42
     )
```



```
[19] # Convert text to TF-IDF vectors
     vectorizer = TfidfVectorizer()
     X_train_vec = vectorizer.fit_transform(X_train)
     X_test_vec = vectorizer.transform(X_test)
```

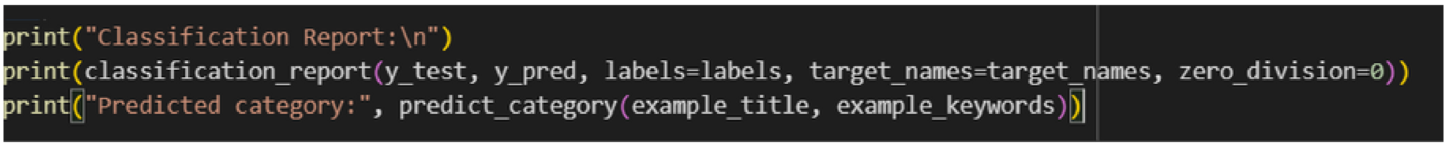
```
     # Train a logistic regression model
     model = LogisticRegression(max_iter=1000)
     model.fit(X_train_vec, y_train)

     # Evaluate the model
     labels = list(category_to_id.values())
     target_names = list(category_to_id.keys())

     y_pred = model.predict(X_test_vec)
```

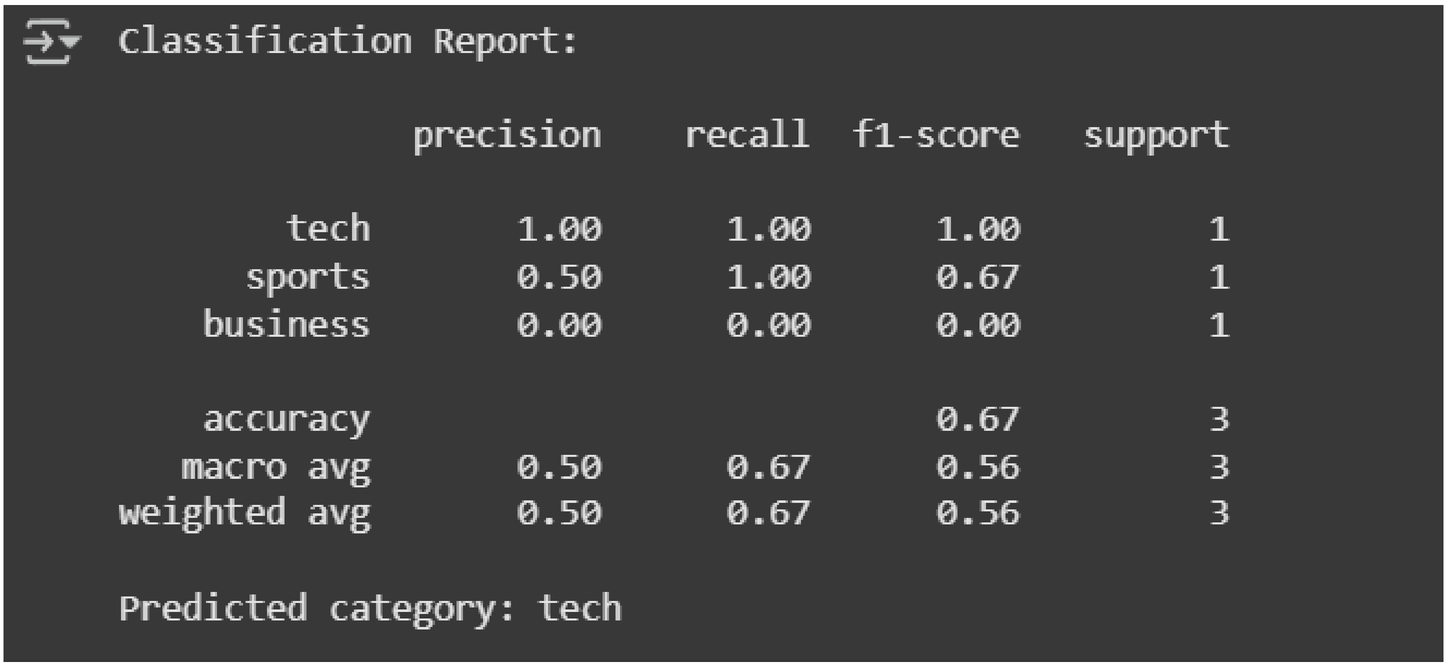
```
[21] # Predict on new example
     def predict_category(title, keywords):
         text = title + " " + keywords
         vec = vectorizer.transform([text])
         pred = model.predict(vec)[0]
         return id_to_category[pred]

     # Example usage
     example_title = "Big Tech Invests in New Data Centers"
     example_keywords = "cloud, investment, technology"
```



```
print("Classification Report:\n")
print(classification_report(y_test, y_pred, labels=labels, target_names=target_names, zero_division=0))
print("Predicted category:", predict_category(example_title, example_keywords))
```

OUTPUT OF THIS :>



```
Classification Report:

              precision    recall  f1-score   support

        tech       1.00      1.00      1.00         1
      sports       0.50      1.00      0.67         1
    business       0.00      0.00      0.00         1

    accuracy                           0.67         3
   macro avg       0.50      0.67      0.56         3
weighted avg       0.50      0.67      0.56         3

Predicted category: tech
```