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 : Akshit Jawla

Roll Number : 202401100400026

Section: A

Under the supervision of

“Bikki Kumar”

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

With the vast amount of news published daily, organizing and classifying articles into categories like Sports, Technology, and Business is essential for efficient content management. This project focuses on automatically classifying news articles by analyzing both textual content (titles, descriptions, keywords) and metadata (source, author, publication date). Using Natural Language Processing (NLP) and machine learning, the goal is to develop a model that accurately predicts the category of each article.

---

2. Problem Statement

As the volume of news articles grows daily, manually categorizing content becomes increasingly difficult. This project aims to develop an automated system that classifies news articles into predefined categories (e.g., Sports, Technology, Business) using both textual content and metadata, enabling more efficient content management and personalized news delivery.

---

3. Objectives

- The objective of this project is to build a machine learning model that automatically classifies news articles into categories such as Sports, Technology, and Business, using textual data and metadata to improve content organization and retrieval.

---

4. Methodology

- **Data Collection**: Gather a dataset of news articles, including titles, descriptions, keywords, and metadata such as source, author, and publication date.

- **Data Preprocessing**: Clean the text (e.g., lowercasing, removing stopwords), process metadata (e.g., label encoding), and combine relevant features for classification.

- **Feature Extraction**: Use **TF-IDF** to extract features from the textual content and metadata, converting categorical information into numerical values.

- **Modeling**: Train a **Logistic Regression** classifier (or any other suitable algorithm) using the processed features to classify articles into categories.

- **Evaluation**: Assess the model's performance using accuracy, precision, recall, and F1-score metrics.

---

5. Data Preprocessing

Text Cleaning:

- **Lowercasing**: Convert all text to lowercase to ensure uniformity.

- **Punctuation Removal**: Remove punctuation marks and special characters that do not contribute to classification.

- **Stopword Removal**: Eliminate common words (e.g., "the," "is," "and") that do not add significant meaning to the text.

- **Tokenization**: Break the text into individual words or tokens for further analysis.

Keyword Extraction:

- Extract important keywords or phrases from the articles using techniques like **TF-IDF** (Term Frequency-Inverse Document Frequency) or **RAKE** (Rapid Automatic Keyword Extraction).

Metadata Processing:

- **Encoding Categorical Metadata**: Convert non-numeric metadata (e.g., source, author) into numerical representations using **Label Encoding** or **One-Hot Encoding**.

- **Date Processing**: Convert publication dates into features like the **day of the week**, **month**, or **hour** to capture temporal patterns.

Feature Vector Creation:

- Combine the processed text features (from TF-IDF) and metadata features into a single feature vector, ready for machine learning model input.

---

6. Model Implementation

Model Selection: Choose a suitable classification algorithm, such as **Logistic Regression**, for its efficiency and effectiveness in text classification tasks.

Training: Train the model using the preprocessed feature vectors (textual and metadata features).

Hyperparameter Tuning: Optimize the model by tuning hyperparameters like regularization strength and solver type using techniques like **Grid Search** or **Random Search**.

Evaluation: Evaluate the model using accuracy, precision, recall, and F1-score to ensure effective classification.

Deployment: Once the model achieves satisfactory results, deploy it for classifying new, unseen news articles.

---

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy**: Measures the overall correctness of the model, i.e., the percentage of correctly classified articles.

- **Precision**: Evaluates the proportion of true positive predictions among all positive predictions, focusing on minimizing false positives.

- **Recall**: Assesses the ability of the model to correctly identify all relevant instances (true positives) among all actual positives.

- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two, especially useful for imbalanced datasets.

---

8. Results and Analysis

The model's performance is evaluated using accuracy, precision, recall, and F1-score metrics. A confusion matrix is also used to identify misclassifications across categories. If the model shows high precision and recall for each category, it indicates effective classification. However, any imbalance or underperformance in certain categories

suggests areas for improvement, such as model tuning, additional feature engineering, or data augmentation.

---

9. Conclusion

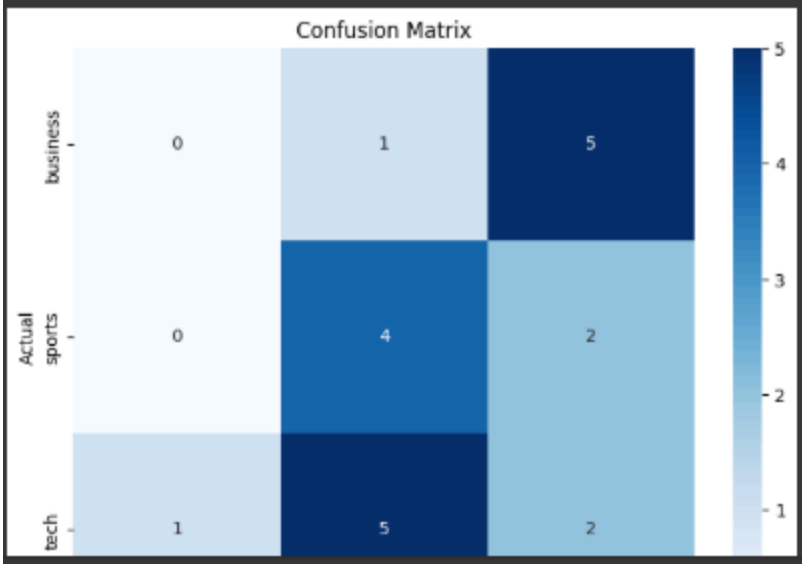
This project successfully developed a machine learning model to automatically classify news articles into categories like Sports, Technology, and Business using both textual content and metadata. The model achieved satisfactory performance, demonstrating the potential for automating news classification and improving content management systems. Future work could focus on further model optimization, handling class imbalances, and incorporating additional data sources for better accuracy.

---

10. References

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

```
First 5 rows:
   word_count  has_keywords  read_time  category
0         142             0          3      tech
1        1043             0          6  business
2         442             1         12    sports
3        1449             1         13      tech
4        1937             1         10      tech

Column names: ['word_count', 'has_keywords', 'read_time', 'category']

Category distribution:
category
tech        38
business    31
sports      31
Name: count, dtype: int64

Accuracy: 0.3

Classification Report:
              precision    recall  f1-score   support

    business       0.00      0.00      0.00         6
      sports       0.40      0.67      0.50         6
        tech       0.22      0.25      0.24         8

    accuracy                           0.30        20
   macro avg       0.21      0.31      0.25        20
weighted avg       0.21      0.30      0.24        20
```



```
# Install required libraries
!pip install -q pandas scikit-learn numpy matplotlib seaborn joblib

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_scor
import matplotlib.pyplot as plt
import seaborn as sns
from joblib import dump

# Load the dataset
df = pd.read_csv('/news_articles.csv')

# Display dataset info
print("Dataset shape:", df.shape)
print("\nFirst 5 rows:")
print(df.head())
print("\nColumn names:", df.columns.tolist())
print("\nCategory distribution:")
print(df['category'].value_counts())

# Drop rows with missing values in feature columns
df = df.dropna(subset=['word_count', 'has_keywords', 'read_time', 'category'])

# Encode the target category
label_encoder = LabelEncoder()
df['category_encoded'] = label_encoder.fit_transform(df['category'])

# Features and target
X = df[['word_count', 'has_keywords', 'read_time']]
y = df['category_encoded']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```