www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assessment Report

on

“Classifying News Articles by Category Using Metadata”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Ganesh Gupta

Roll Number : 202401100300110

Section: B

Under the supervision of

“Shivansh Prasad”

KIET Group of Institutions, Ghaziabad

Classifying News Articles by Category Using Metadata

Introduction

This project aims to classify news articles into categories such as sports, tech, and business using only metadata. The metadata includes features like word count, whether keywords are present, and the estimated read time. This is a practical machine learning task where textual content is not used, making it an interesting challenge in feature-based classification.

Methodology

We used the following steps to build our classifier:

1. Load and clean the dataset.
2. Encode the target labels.
3. Use structured metadata (word_count, has_keywords, read_time) as features.
4. Split the data into training and test sets.
5. Train a Random Forest classifier.
6. Evaluate the model using accuracy and classification metrics.

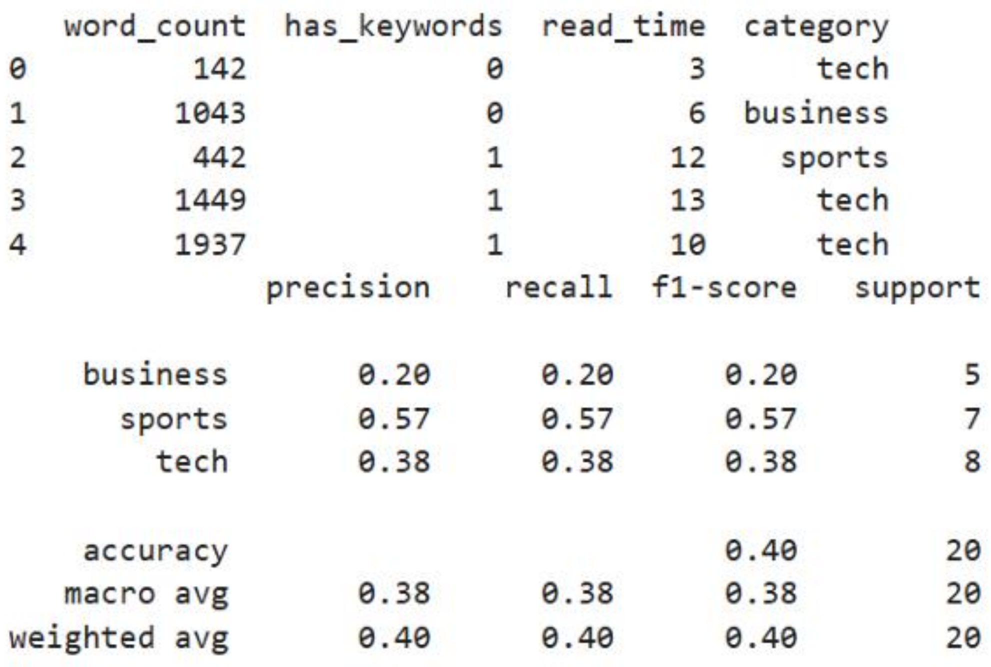
Code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report

# Load data
df = pd.read_csv('/mnt/data/news_articles.csv')
df = df.dropna(subset=['word_count', 'has_keywords', 'read_time', 'category'])

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

# Feature selection
```

```
features = ['word_count', 'has_keywords', 'read_time']
X = df[features]
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)

# Train model
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)

# Evaluate
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred, target_names=label_encoder.classes_))
```

Output/Result

Below is the screenshot of the model's output (classification report):



```
   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
              precision    recall  f1-score   support

    business       0.20      0.20      0.20         5
      sports       0.57      0.57      0.57         7
        tech       0.38      0.38      0.38         8

    accuracy                           0.40        20
   macro avg       0.38      0.38      0.38        20
weighted avg       0.40      0.40      0.40        20
```

References/Credits

Dataset provided by instructor.
Libraries used: pandas, scikit-learn, matplotlib (if applicable).
Code written and executed using Python.