KIET GROUP OF INSTITUTIONS

INTRODUCTION TO AI MSE-2

Harshita Sharma

CSE(AI)-B

202401100300120

Problem Statement:
Classify news articles into different categories such as sports, tech, business, etc., based on available metadata.

Introduction:

The objective of this project is to classify news articles into various categories like sports, technology, and business using available metadata features such as word_count, has_keywords, and read_time.
Since the dataset does not contain article text, the classification is based solely on these numerical features. This presents challenges as limited metadata makes it difficult for the model to accurately predict the categories.

A confusion matrix and evaluation metrics such as accuracy, precision, and recall are used to evaluate the model's performance.

In this AI model,

Methodology:

1. **Dataset Used**:
    - Dataset: news_articles.csv
    - Features: word_count, has_keywords, read_time
    - Target: category
2. **Approach**:
    - Read the CSV file and load the dataset.
    - Define the features (X) and labels (y).
    - Split the data into training (80%) and testing (20%) sets.
    - Train a Random Forest Classifier.
    - Predict the categories on the test set.
    - Evaluate the model using accuracy, precision, recall, and confusion matrix.

- Visualize the confusion matrix using a heatmap.

3. **Libraries Used**:
    - pandas, scikit-learn, seaborn, matplotlib

Code:

```
# Install required libraries

# !pip install pandas scikit-learn matplotlib seaborn


import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, precision_score, recall_score

import matplotlib.pyplot as plt

import seaborn as sns


# 1. Load the dataset

file_path = 'news_articles.csv'

data = pd.read_csv(file_path)


# 2. Define features (X) and labels (y)

X = data[['word_count', 'has_keywords', 'read_time']]

y = data['category']
```

```
# 3. Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 4. Initialize and train the classifier
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)

# 5. Make predictions
y_pred = clf.predict(X_test)

# 6. Calculate evaluation metrics
```

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted', zero_division=0)
recall = recall_score(y_test, y_pred, average='weighted', zero_division=0)
print("\nClassification Report:\n", classification_report(y_test, y_pred, zero_division=0))
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")

# 7. Create the confusion matrix
cm = confusion_matrix(y_test, y_pred)

# 8. Plot the heatmap
```

MSE2.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text

RAM Disk

Files

Analyze your files with code written by Gemini Upload

..

sample_data

README.md

anscombe.json

california_housing_test.csv

california_housing_train.csv

mnist_test.csv

mnist_train_small.csv

news_articles.csv

```
precision = precision_score(y_test, y_pred, average='weighted', zero_division=0)
recall = recall_score(y_test, y_pred, average='weighted', zero_division=0)
print("\nClassification Report:\n", classification_report(y_test, y_pred, zero_division=0))
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")

# 7. Create the confusion matrix
cm = confusion_matrix(y_test, y_pred)

# 8. Plot the heatmap
```

```
Classification Report:
              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

Accuracy: 0.4000
Precision: 0.4000
Recall: 0.4000
```

References/Credits:

- Dataset provided by instructor / assignment.
- Libraries used:
  - Pandas Documentation
  - Scikit-learn Documentation
  - Seaborn Documentation
  - Matplotlib Documentation