hari0377 /
fakenewsdetectionusingNLP-phase-3

Code Issues Pull requests Actions Projects Wiki Security Insights Set

fakenewsdetectionusingNLP-phase-3 / main.py

hari0377 Add files via upload now

99 lines (78 loc) · 2.96 KB

Code Blame Raw

```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import LinearSVC
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
from sklearn.utils import shuffle
from scipy.sparse import hstack
from sklearn.model_selection import cross_val_score,learning_curve
import matplotlib.pyplot as plt

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

true=pd.read_csv("/kaggle/input/fake-and-real-news-dataset/True.csv")
fake=pd.read_csv("/kaggle/input/fake-and-real-news-dataset/Fake.csv")
true.head(50)
true["subject"].value_counts()

fake.head()
fake["subject"].value_counts()

true.isnull().sum()

fake.isnull().sum()

true.shape

fake.shape

true.head()

fake.head()

true["label"]=1
fake["label"]=0

true.head()

fake.head()

data=pd.concat([fake,true],ignore_index=True)
```

```python
data=pd.concat([fake,true],ignore_index=True)
data.head()

X=data["text"]
y=data["label"]
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)

vectorizer=CountVectorizer()
X_train_vectors=vectorizer.fit_transform(X_train)
X_test_vectors=vectorizer.transform(X_test)

vectorizer = CountVectorizer()
X_vectors = vectorizer.fit_transform(data['text'])
X_train, X_test, y_train, y_test = train_test_split(X_vectors, data['label'], test_size=0.2, random_state=42)
classifier = MultinomialNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

new_texts = ["This news article is definitely fake.",
             "The research study confirms the truth of the news."]
new_texts_vectors = vectorizer.transform(new_texts)
predictions = classifier.predict(new_texts_vectors)
for text, label in zip(new_texts, predictions):
    print(f"Text: {text}\nPrediction: {'Fake' if label == 0 else 'True'}\n")


true_df = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/True.csv')
fake_df = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/Fake.csv')
fake_df['label'] = 0
true_df['label'] = 1
combined_df = pd.concat([fake_df, true_df], ignore_index=True)
combined_df = combined_df.sample(frac=1, random_state=42).reset_index(drop=True)
X = combined_df['title'] + " " + combined_df['text']
y = combined_df['label']
vectorizer = TfidfVectorizer()
X_vectors = vectorizer.fit_transform(X)
classifier = MultinomialNB(alpha=1.0)
classifier.fit(X_vectors, y)
def predict_label(input_title):
    input_text = ""
    input_data = input_title + " " + input_text
    input_vector = vectorizer.transform([input_data])
    label = classifier.predict(input_vector)[0]
    return label
input_title ="WASHINGTON (Reuters) - The special counsel"
predicted_label = predict_label(input_title)
if predicted_label == 0:
    print("Predicted Label: Fake")
else:
    print("Predicted Label: True")
```