

Manikkandan1404 /
FakeNewsDetectionusingNLP-phase-3[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

FakeNewsDetectionusingNLP-phase-3 / main.py



Manikkandan1404 Add files via upload

now



99 lines (78 loc) · 2.96 KB

Code Blame

Raw



```
1  import numpy as np
2  import pandas as pd
3  from sklearn.model_selection import train_test_split, GridSearchCV
4  from sklearn.naive_bayes import MultinomialNB
5  from sklearn.feature_extraction.text import CountVectorizer
6  from sklearn.svm import LinearSVC
7  from sklearn.feature_extraction.text import TfidfVectorizer
8  from sklearn.metrics import accuracy_score
9  from sklearn.utils import shuffle
10 from scipy.sparse import hstack
11 from sklearn.model_selection import cross_val_score, learning_curve
12 import matplotlib.pyplot as plt
13
14 import os
15 for dirname, _, filenames in os.walk('/kaggle/input'):
16     for filename in filenames:
17         print(os.path.join(dirname, filename))
18
19 true=pd.read_csv("/kaggle/input/fake-and-real-news-dataset/True.csv")
20 fake=pd.read_csv("/kaggle/input/fake-and-real-news-dataset/Fake.csv")
21 true.head(50)
22 true["subject"].value_counts()
23
24 fake.head()
25 fake["subject"].value_counts()
26
27 true.isnull().sum()
28
29 fake.isnull().sum()
30
31 true.shape
32
33 fake.shape
34
35 true.head()
36
37 fake.head()
38
39 true["label"]=1
40 fake["label"]=0
41
42 true.head()
43
44 fake.head()
45
46 data=pd.concat([fake, true], ignore_index=True)
```

```

46 data=pd.concat([fake,true],ignore_index=True)
47 data.head()
48
49 X=data["text"]
50 y=data["label"]
51 X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
52
53 vectorizer=CountVectorizer()
54 X_train_vectors=vectorizer.fit_transform(X_train)
55 X_test_vectors=vectorizer.transform(X_test)
56
57 vectorizer = CountVectorizer()
58 X_vectors = vectorizer.fit_transform(data['text'])
59 X_train, X_test, y_train, y_test = train_test_split(X_vectors, data['label'], test_size=0.2, random_state=42)
60 classifier = MultinomialNB()
61 classifier.fit(X_train, y_train)
62 y_pred = classifier.predict(X_test)
63 accuracy = accuracy_score(y_test, y_pred)
64 print("Accuracy:", accuracy)
65
66 new_texts = ["This news article is definitely fake.",
67              "The research study confirms the truth of the news."]
68 new_texts_vectors = vectorizer.transform(new_texts)
69 predictions = classifier.predict(new_texts_vectors)
70 for text, label in zip(new_texts, predictions):
71     print(f"Text: {text}\nPrediction: {'Fake' if label == 0 else 'True'}\n")
72
73
74 true_df = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/True.csv')
75 fake_df = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/Fake.csv')
76 fake_df['label'] = 0
77 true_df['label'] = 1
78 combined_df = pd.concat([fake_df, true_df], ignore_index=True)
79 combined_df = combined_df.sample(frac=1, random_state=42).reset_index(drop=True)
80 X = combined_df['title'] + " " + combined_df['text']
81 y = combined_df['label']
82 vectorizer = TfidfVectorizer()
83 X_vectors = vectorizer.fit_transform(X)
84 classifier = MultinomialNB(alpha=1.0)
85 classifier.fit(X_vectors, y)
86 ✓ def predict_label(input_title):
87     input_text = ""
88     input_data = input_title + " " + input_text
89     input_vector = vectorizer.transform([input_data])
90     label = classifier.predict(input_vector)[0]
91     return label
92 input_title ="WASHINGTON (Reuters) - The special counsel"
93 predicted_label = predict_label(input_title)
94 if predicted_label == 0:
95     print("Predicted Label: Fake")
96 else:
97     print("Predicted Label: True")

```