

SELLAPPANMURALI /
Ai_phase3

<> Code

Issues

Pull requests

Projects

Wiki

Security

Insights

Settings

Ai_phase3 / main.py



SELLAPPANMURALI Update main.py

1 minute ago



102 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
16 for dirname, _, filenames in os.walk('/kaggle/input'):
17     for filename in filenames:
18         print(os.path.join(dirname, filename))
19
20
21
22 true=pd.read_csv("/kaggle/input/fake-and-real-news-dataset/True.csv")
23 fake=pd.read_csv("/kaggle/input/fake-and-real-news-dataset/Fake.csv")
24 true.head(50)
25 true["subject"].value_counts()
26
27 fake.head()
28 fake["subject"].value_counts()
29
30 true.isnull().sum()
31
32 fake.isnull().sum()
```

34 true.shape

```
35
36     fake.shape
37
38     true.head()
39
40     fake.head()
41
42     true["label"]=1
43     fake["label"]=0
44
45     true.head()
46
47     fake.head()
48
49     data=pd.concat([fake,true],ignore_index=True)
50     data.head()
51
52     X=data["text"]
53     y=data["label"]
54     X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
55
56     vectorizer=CountVectorizer()
57     X_train_vectors=vectorizer.fit_transform(X_train)
58     X_test_vectors=vectorizer.transform(X_test)
59
60     vectorizer = CountVectorizer()
61     X_vectors = vectorizer.fit_transform(data['text'])
62     X_train, X_test, y_train, y_test = train_test_split(X_vectors, data['label'], test_size=0.2, r
63     classifier = MultinomialNB()
64     classifier.fit(X_train, y_train)
65     y_pred = classifier.predict(X_test)
66     accuracy = accuracy_score(y_test, y_pred)
67     print("Accuracy:", accuracy)
68
69     new_texts = ["This news article is definitely fake.",
70                 "The research study confirms the truth of the news."]
71     new_texts_vectors = vectorizer.transform(new_texts)
72     predictions = classifier.predict(new_texts_vectors)
73     for text, label in zip(new_texts, predictions):
74         print(f"Text: {text}\nPrediction: {'Fake' if label == 0 else 'True'}\n")
75
76
77     true_df = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/True.csv')
78     fake_df = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/Fake.csv')
79     fake_df['label'] = 0
80     true_df['label'] = 1
81     combined_df = pd.concat([fake_df, true_df], ignore_index=True)
82     combined_df = combined_df.sample(frac=1, random_state=42).reset_index(drop=True)
83     X = combined_df['title'] + " " + combined_df['text']
84     y = combined_df['label']
85     vectorizer = TfidfVectorizer()
```

```
--      - - - - - \\'
86     X_vectors = vectorizer.fit_transform(X)
87     classifier = MultinomialNB(alpha=1.0)
88     classifier.fit(X_vectors, y)
89     ✓ def predict_label(input_title):
90         input_text = ""
91         input_data = input_title + " " + input_text
92         input_vector = vectorizer.transform([input_data])
93         label = classifier.predict(input_vector)[0]
94         return label
95     input_title ="WASHINGTON (Reuters) - The special counsel"
96     predicted_label = predict_label(input_title)
97     if predicted_label == 0:
98         print("Predicted Label: Fake")
99     else:
100         print("Predicted Label: True")
```