KGugan /
FakenewsdewtectionusingNLP-phase-4

<> Code   ⊙ Issues   ⑂ Pull requests   ▷ Actions   ⊞ Projects   📖 Wiki   ⊘ Security   📈 Insights   ⚙ Set

FakenewsdewtectionusingNLP-phase-4 / **main.py** ⧉

👤 **KGugan** Update main.py   now ••• 🕘

95 lines (73 loc) · 7.3 KB

| Code | Blame | Raw ⧉ ⬇ ✎ ▾ ‹›

```python
1    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:09.090352Z","iopub.execute_input":"2023-10-27
2    # This Python 3 environment comes with many helpful analytics libraries installed
3    # It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
4    # For example, here's several helpful packages to load
5
6    import numpy as np # linear algebra
7    import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
8
9    # Input data files are available in the read-only "../input/" directory
10   # For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input dir
11
12   import os
13   for dirname, _, filenames in os.walk('/kaggle/input'):
14       for filename in filenames:
15           print(os.path.join(dirname, filename))
16
17   # You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you
18   # You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
19
20   # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:09.114058Z","iopub.execute_input":"2023-10-27
21   import pandas as pd
22   import nltk
23   from nltk.sentiment import SentimentIntensityAnalyzer
24   import warnings
25   warnings.filterwarnings("ignore")
26   from sklearn.model_selection import train_test_split
27   from sklearn.feature_extraction.text import TfidfVectorizer
28   from sklearn.svm import SVC
29   from sklearn.metrics import accuracy_score, classification_report
30
31   # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:10.243502Z","iopub.execute_input":"2023-10-27
32   fake = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/Fake.csv')
33   true = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/True.csv')
34
35   # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.787408Z","iopub.execute_input":"2023-10-27
36   fake['Category'] = 'fake'
37   fake
38
39   # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.815088Z","iopub.execute_input":"2023-10-27
40   true['Category'] = 'true'
41   true
42
43   # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.832993Z","iopub.execute_input":"2023-10-27
44   data = pd.concat([fake, true], ignore_index=True)
45   data
46
```

```python
47    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.859531Z","iopub.execute_input":"2023-10-27
48    data['Category'].value_counts()
49
50    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.879444Z","iopub.execute_input":"2023-10-27
51    from sklearn.preprocessing import LabelEncoder
52    le = LabelEncoder()
53    data['Category'] = le.fit_transform(data['Category'])
54    data['date'] = le.fit_transform(data['date'])
55    data['subject'] = le.fit_transform(data['subject'])
56
57    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.940237Z","iopub.execute_input":"2023-10-27
58    data['Category']
59
60    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.949272Z","iopub.execute_input":"2023-10-27
61    data['date']
62
63    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.961812Z","iopub.execute_input":"2023-10-27
64    data['subject'].value_counts()
65
66    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.973990Z","iopub.execute_input":"2023-10-27
67    data['title'].shape
68
69    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:12.984114Z","iopub.execute_input":"2023-10-27
70    vectorizer = TfidfVectorizer()
71    title = vectorizer.fit_transform(data['title'])
72    text = vectorizer.transform(data['text'])
73
74
75    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:32.117764Z","iopub.execute_input":"2023-10-27
76     title
77
78    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:32.124993Z","iopub.execute_input":"2023-10-27
79    from sklearn.model_selection import train_test_split
80    X = title
81    y = data['Category']
82    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
83
84    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:25:32.147343Z","iopub.execute_input":"2023-10-27
85    model = SVC()
86    model.fit(X_train, y_train)
87
88    # %% [code] {"execution":{"iopub.status.busy":"2023-10-27T04:29:37.364346Z","iopub.execute_input":"2023-10-27
89    y_pred = model.predict(X_test)
90    accuracy = accuracy_score(y_test, y_pred)
91    print("Accuracy:", accuracy)
92    print("Classification Report:")
93    print(classification_report(y_test, y_pred))
94
95    # %% [code]
```