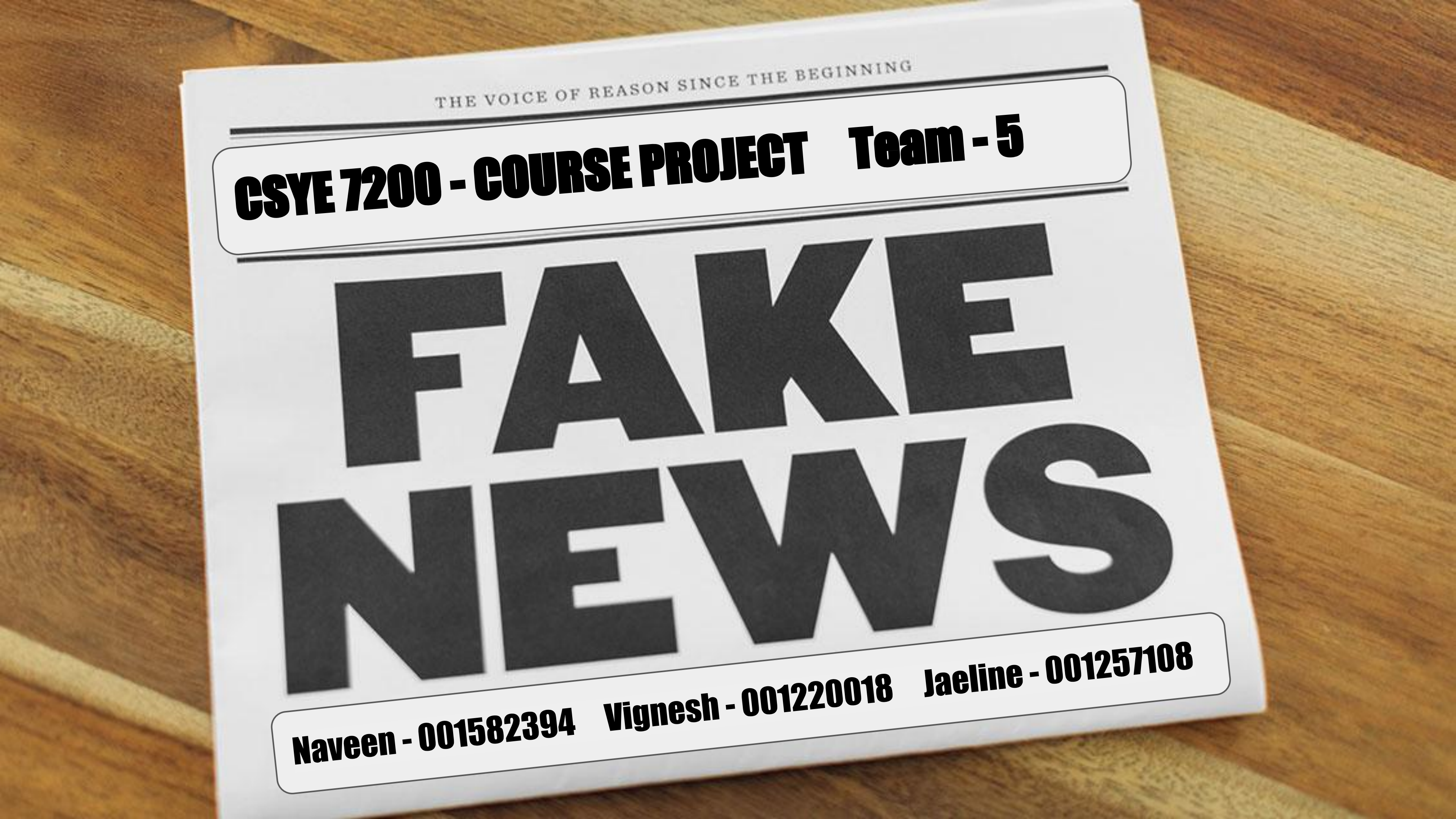


CSYE 7200 - COURSE PROJECT Team - 5

Naveen - 001582394 Vignesh - 001220018 Jaeline - 001257108

Table of Contents

01 Use Case

02 UI Design

03 Goal

04 Methodology

05 TFIDF

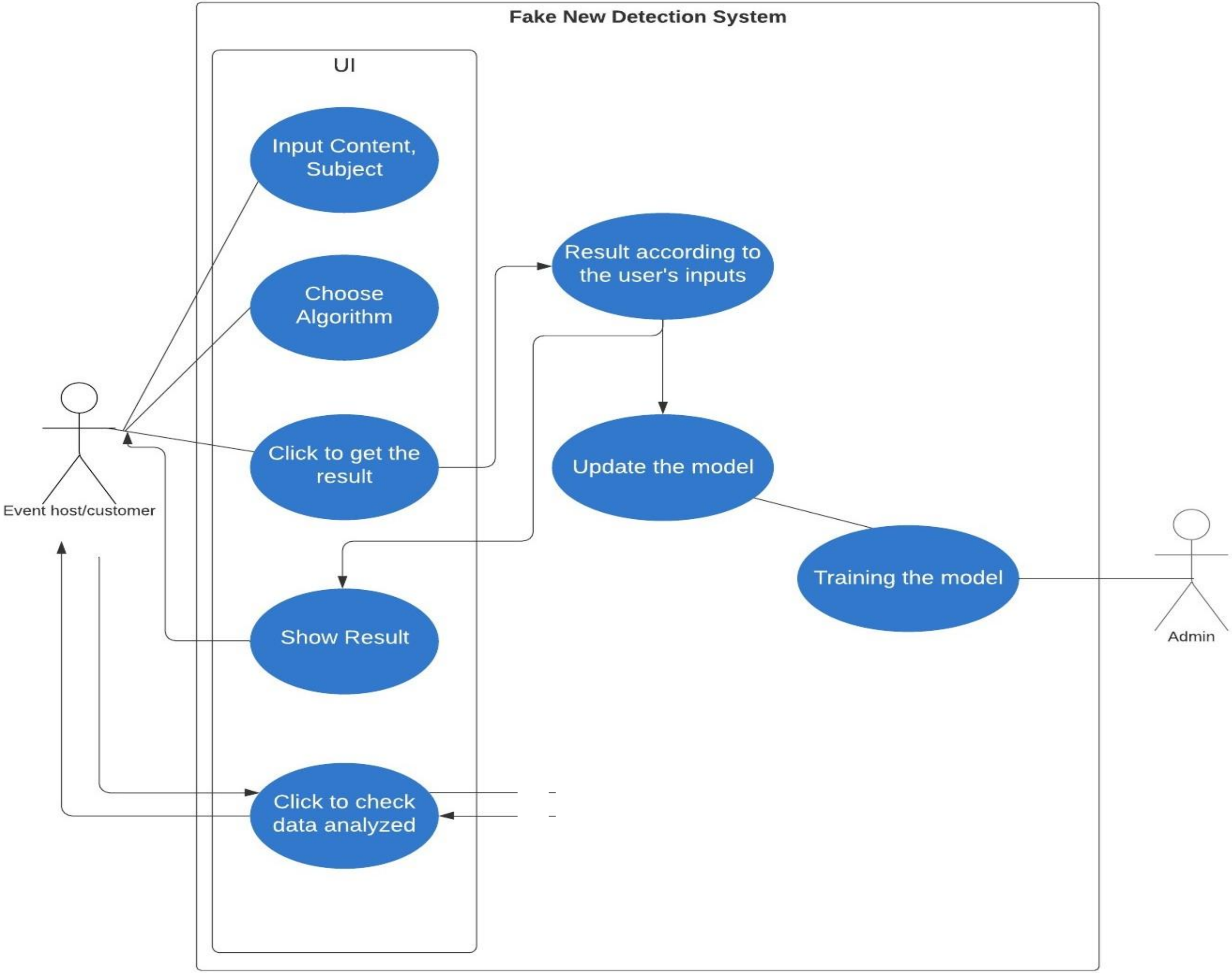
06 Random Forest

07 Naive Bayes

08 Results

09 Acceptance Criteria

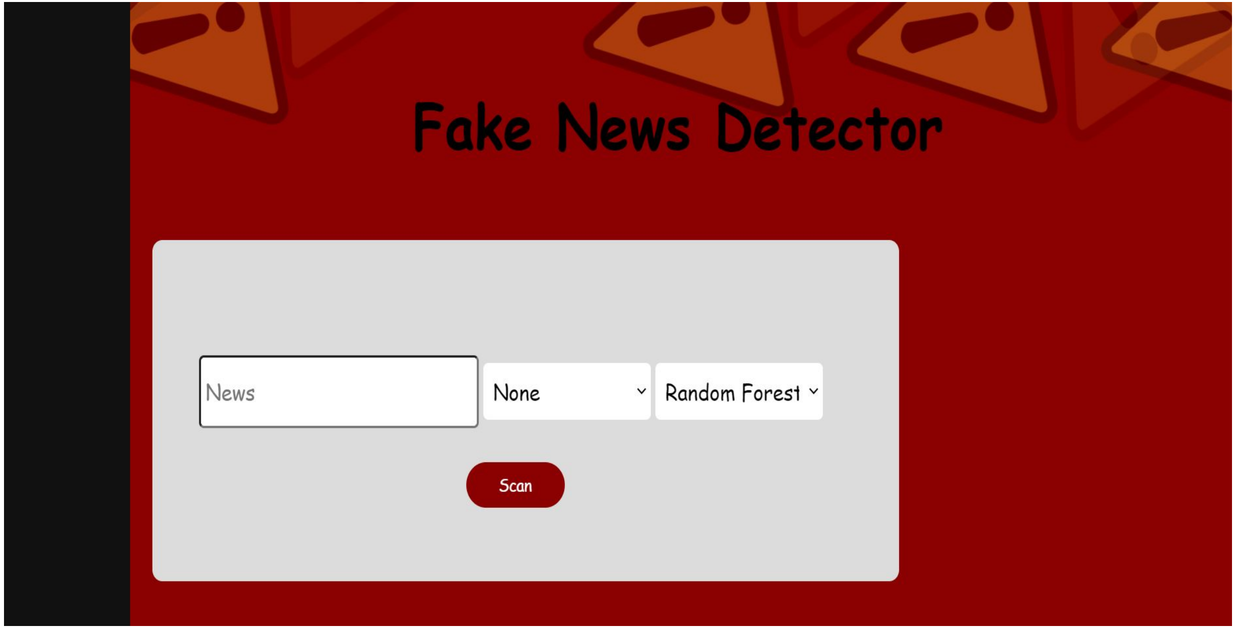
Use Case


Fake New Detection System
UI
Input Content, Subject
Choose Algorithm
Click to get the result
Show Result
Click to check data analyzed
Result according to the user's inputs
Update the model
Training the model
Event host/customer
Admin

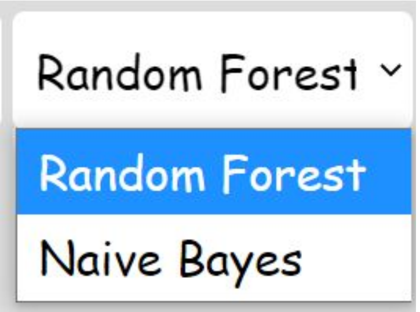
Project Goals

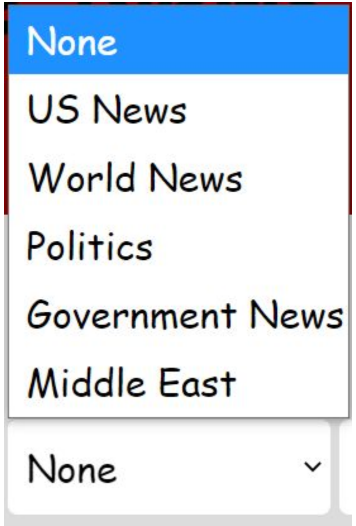
- *A Interactive page to accept the user input*
- *A well trained model to analyze the characteristics*
- *A higher probability output of the data provided*

UI Design

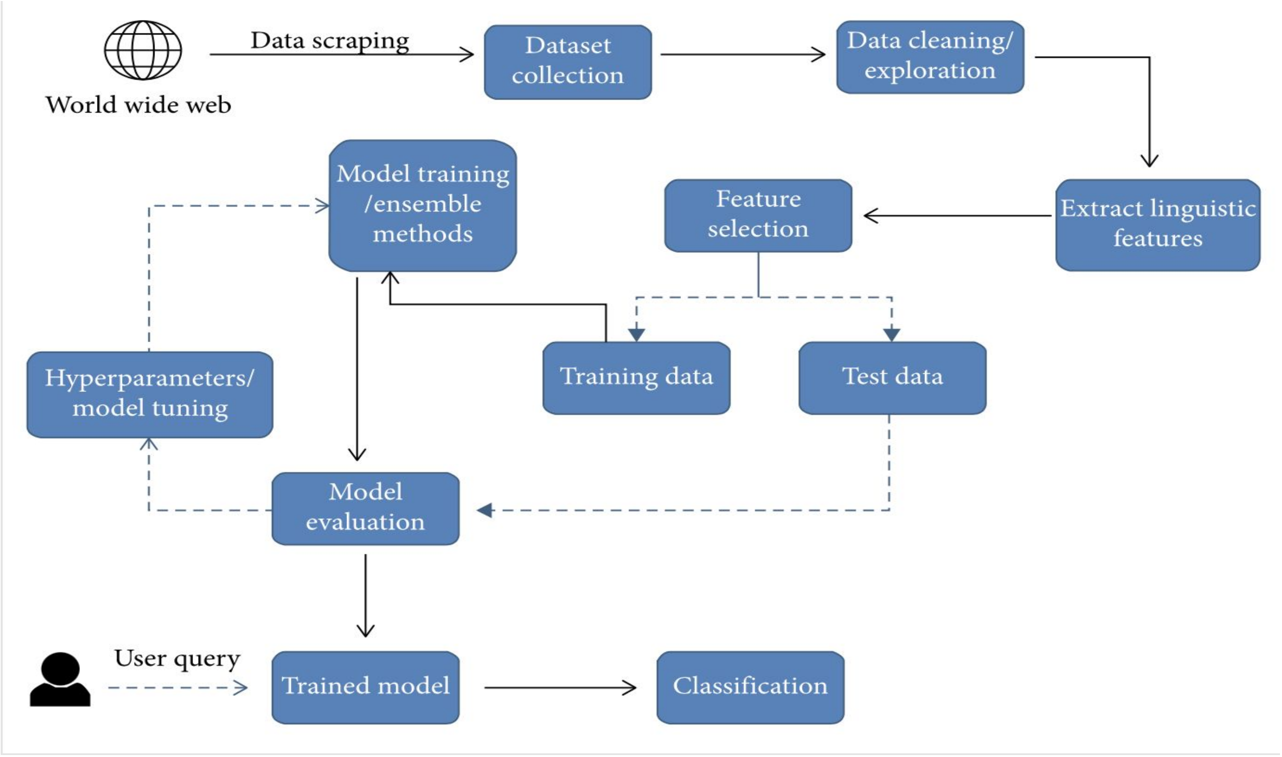


Fake News Detector



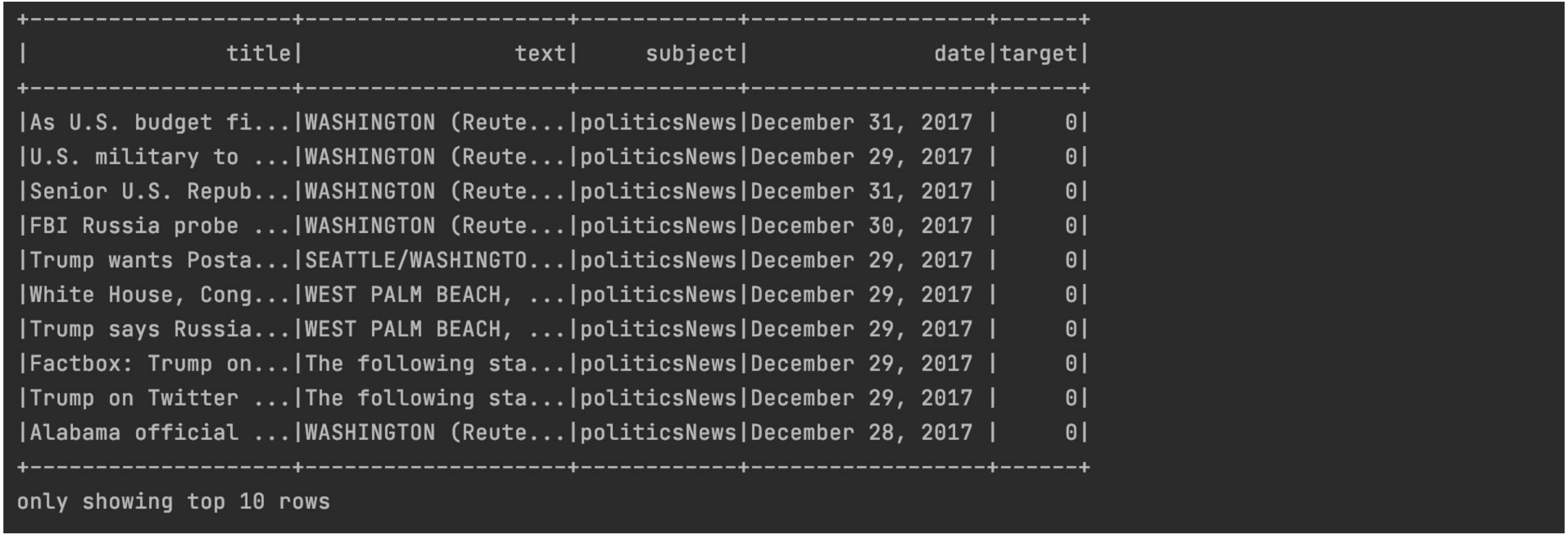


Methodology



Data Source:

- *Data ~ 44898 rows*



```
+--------------------+--------------------+------------+------------------+------+
|               title|                text|     subject|              date|target|
+--------------------+--------------------+------------+------------------+------+
|As U.S. budget fi...|WASHINGTON (Reute...|politicsNews|December 31, 2017 |     0|
|U.S. military to ...|WASHINGTON (Reute...|politicsNews|December 29, 2017 |     0|
|Senior U.S. Repub...|WASHINGTON (Reute...|politicsNews|December 31, 2017 |     0|
|FBI Russia probe ...|WASHINGTON (Reute...|politicsNews|December 30, 2017 |     0|
|Trump wants Posta...|SEATTLE/WASHINGTO...|politicsNews|December 29, 2017 |     0|
|White House, Cong...|WEST PALM BEACH, ...|politicsNews|December 29, 2017 |     0|
|Trump says Russia...|WEST PALM BEACH, ...|politicsNews|December 29, 2017 |     0|
|Factbox: Trump on...|The following sta...|politicsNews|December 29, 2017 |     0|
|Trump on Twitter ...|The following sta...|politicsNews|December 29, 2017 |     0|
|Alabama official ...|WASHINGTON (Reute...|politicsNews|December 28, 2017 |     0|
+--------------------+--------------------+------------+------------------+------+
only showing top 10 rows
```

TF IDF

-extract features and find keywords

- TF(t,d) - number of times a term t appears in document d
- DF(t,D) - number of documents d that contains term t
- |D| is the total number of documents in the corpus

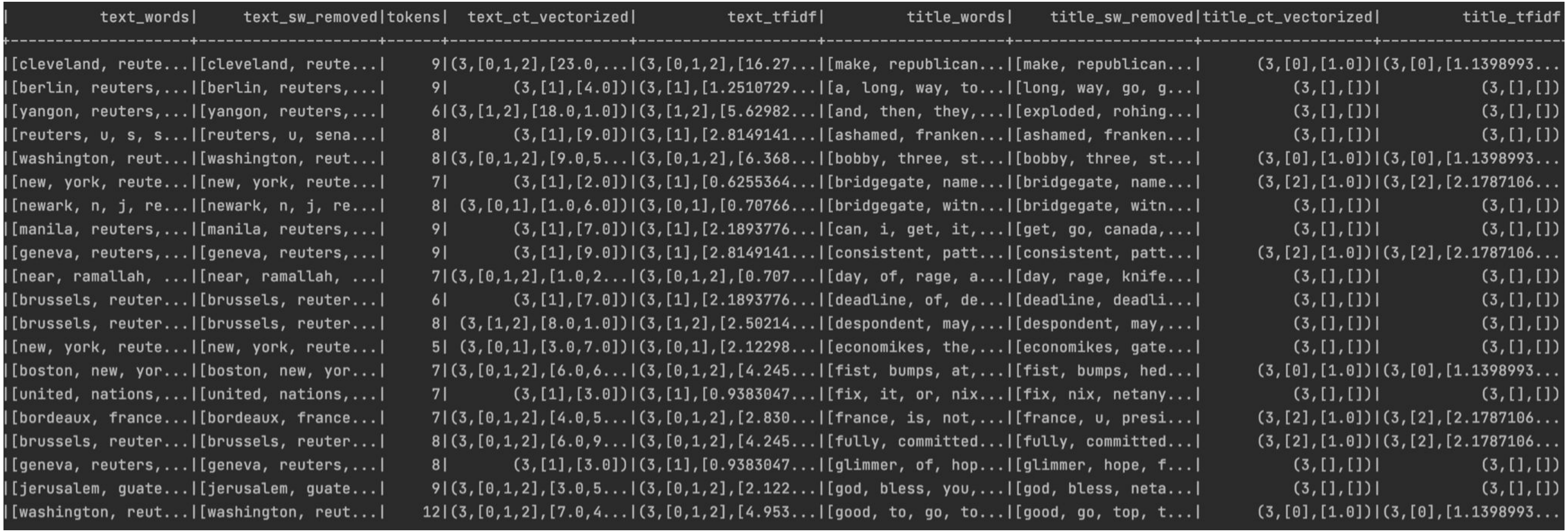
$$idf(t) = \log(N/(df + 1))$$

$$tf\text{-}idf(t, d) = tf(t, d) * \log(N/(df + 1))$$

Natural Language Processing

- *Extract tokens from title*
- *Remove stop words from title*
- *Compute Term frequency from title*
- *Compute Term frequency-inverse document frequency from title*
- *Extract tokens from text*
- *Remove stop words from text*
- *Compute Term frequency from text*
- *Compute Term frequency-inverse document frequency text*
- *StringIndexer subject*

Natural Language Processing

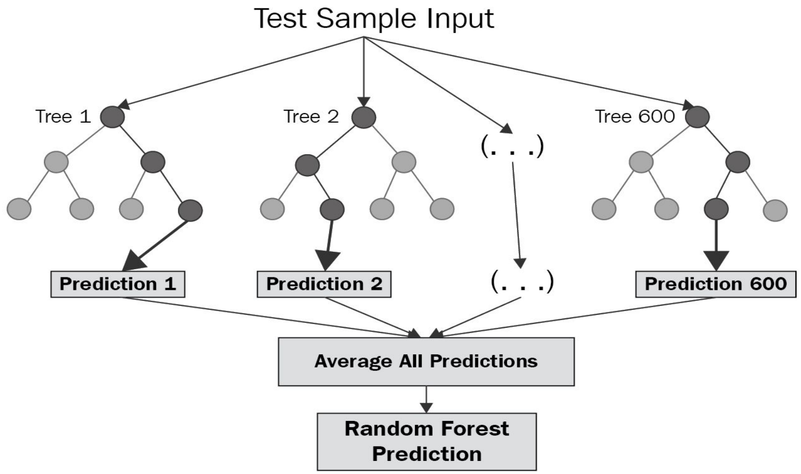


```
|          text_words|     text_sw_removed|tokens|  text_ct_vectorized|          text_tfidf|         title_words|    title_sw_removed|title_ct_vectorized|         title_tfidf
+--------------------+--------------------+------+--------------------+--------------------+--------------------+--------------------+-------------------+--------------------
|[cleveland, reute...|[cleveland, reute...|     9|(3,[0,1,2],[23.0,...|(3,[0,1,2],[16.27...|[make, republican...|[make, republican...|      (3,[0],[1.0])|(3,[0],[1.1398993...
|[berlin, reuters,...|[berlin, reuters,...|     9|       (3,[1],[4.0])|(3,[1],[1.2510729...|[a, long, way, to...|[long, way, go, g...|          (3,[],[])|           (3,[],[])
|[yangon, reuters,...|[yangon, reuters,...|     6|(3,[1,2],[18.0,1.0])|(3,[1,2],[5.62982...|[and, then, they,...|[exploded, rohing...|          (3,[],[])|           (3,[],[])
|[reuters, u, s, s...|[reuters, u, sena...|     8|       (3,[1],[9.0])|(3,[1],[2.8149141...|[ashamed, franken...|[ashamed, franken...|          (3,[],[])|           (3,[],[])
|[washington, reut...|[washington, reut...|     8|(3,[0,1,2],[9.0,5...|(3,[0,1,2],[6.368...|[bobby, three, st...|[bobby, three, st...|      (3,[0],[1.0])|(3,[0],[1.1398993...
|[new, york, reute...|[new, york, reute...|     7|       (3,[1],[2.0])|(3,[1],[0.6255364...|[bridgegate, name...|[bridgegate, name...|      (3,[2],[1.0])|(3,[2],[2.1787106...
|[newark, n, j, re...|[newark, n, j, re...|     8| (3,[0,1],[1.0,6.0])|(3,[0,1],[0.70766...|[bridgegate, witn...|[bridgegate, witn...|          (3,[],[])|           (3,[],[])
|[manila, reuters,...|[manila, reuters,...|     9|       (3,[1],[7.0])|(3,[1],[2.1893776...|[can, i, get, it,...|[get, go, canada,...|          (3,[],[])|           (3,[],[])
|[geneva, reuters,...|[geneva, reuters,...|     9|       (3,[1],[9.0])|(3,[1],[2.8149141...|[consistent, patt...|[consistent, patt...|      (3,[2],[1.0])|(3,[2],[2.1787106...
|[near, ramallah, ...|[near, ramallah, ...|     7|(3,[0,1,2],[1.0,2...|(3,[0,1,2],[0.707...|[day, of, rage, a...|[day, rage, knife...|          (3,[],[])|           (3,[],[])
|[brussels, reuter...|[brussels, reuter...|     6|       (3,[1],[7.0])|(3,[1],[2.1893776...|[deadline, of, de...|[deadline, deadli...|          (3,[],[])|           (3,[],[])
|[brussels, reuter...|[brussels, reuter...|     8| (3,[1,2],[8.0,1.0])|(3,[1,2],[2.50214...|[despondent, may,...|[despondent, may,...|          (3,[],[])|           (3,[],[])
|[new, york, reute...|[new, york, reute...|     5| (3,[0,1],[3.0,7.0])|(3,[0,1],[2.12298...|[economikes, the,...|[economikes, gate...|          (3,[],[])|           (3,[],[])
|[boston, new, yor...|[boston, new, yor...|     7|(3,[0,1,2],[6.0,6...|(3,[0,1,2],[4.245...|[fist, bumps, at,...|[fist, bumps, hed...|      (3,[0],[1.0])|(3,[0],[1.1398993...
|[united, nations,...|[united, nations,...|     7|       (3,[1],[3.0])|(3,[1],[0.9383047...|[fix, it, or, nix...|[fix, nix, netany...|          (3,[],[])|           (3,[],[])
|[bordeaux, france...|[bordeaux, france...|     7|(3,[0,1,2],[4.0,5...|(3,[0,1,2],[2.830...|[france, is, not,...|[france, u, presi...|      (3,[2],[1.0])|(3,[2],[2.1787106...
|[brussels, reuter...|[brussels, reuter...|     8|(3,[0,1,2],[6.0,9...|(3,[0,1,2],[4.245...|[fully, committed...|[fully, committed...|      (3,[2],[1.0])|(3,[2],[2.1787106...
|[geneva, reuters,...|[geneva, reuters,...|     8|       (3,[1],[3.0])|(3,[1],[0.9383047...|[glimmer, of, hop...|[glimmer, hope, f...|          (3,[],[])|           (3,[],[])
|[jerusalem, guate...|[jerusalem, guate...|     9|(3,[0,1,2],[3.0,5...|(3,[0,1,2],[2.122...|[god, bless, you,...|[god, bless, neta...|          (3,[],[])|           (3,[],[])
|[washington, reut...|[washington, reut...|    12|(3,[0,1,2],[7.0,4...|(3,[0,1,2],[4.953...|[good, to, go, to...|[good, go, top, t...|      (3,[0],[1.0])|(3,[0],[1.1398993...
```

Random Forest Classifier

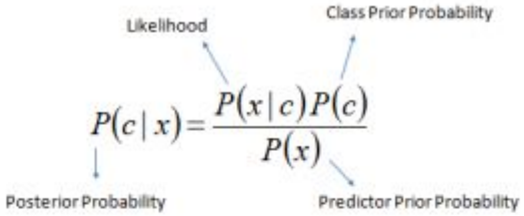
- builds multiple decision trees and merges them together to get a more accurate and stable prediction

- Given a input training dataset X = x1... xn with response Y = y1...yn, we select a random sample with replacement from the training dataset and fit the tree



Naive Bayes Classifier:

- We convert the input data into frequency table as from the TFIDF class
- Generate the Likelihood table by finding the probabilities.
- use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.



Likelihood

Class Prior Probability

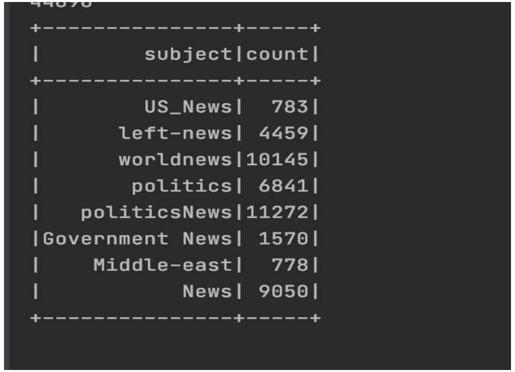
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Posterior Probability

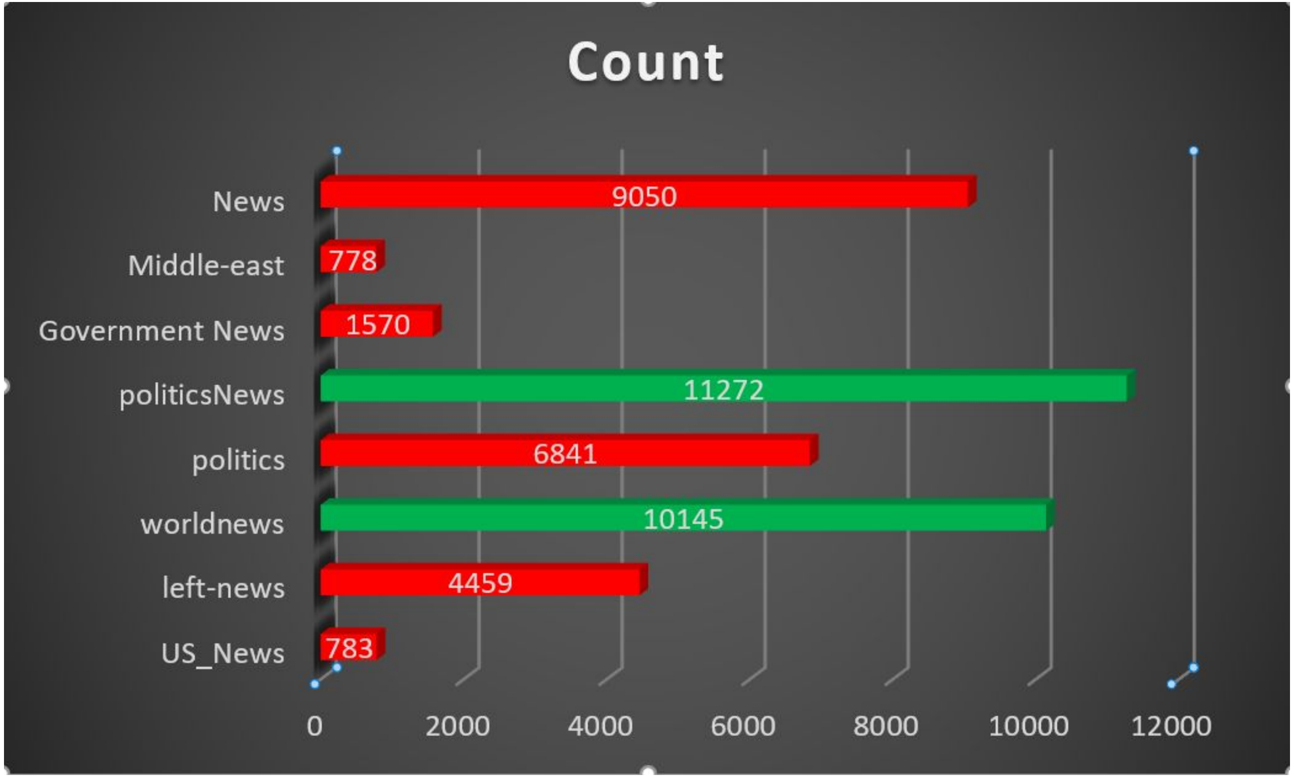
Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

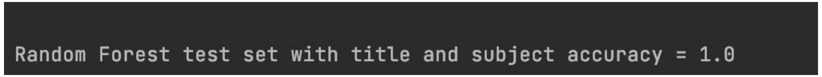
Visualization:



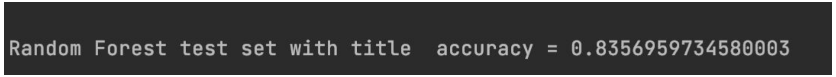
```
44898
+---------------+-----+
|        subject|count|
+---------------+-----+
|        US_News|  783|
|      left-news| 4459|
|      worldnews|10145|
|       politics| 6841|
|   politicsNews|11272|
|Government News| 1570|
|    Middle-east|  778|
|           News| 9050|
+---------------+-----+
```



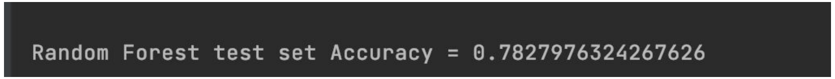
Model Performance:



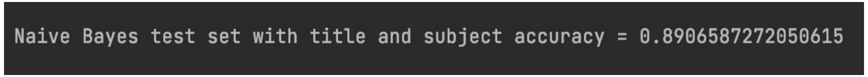
```
Random Forest test set with title and subject accuracy = 1.0
```



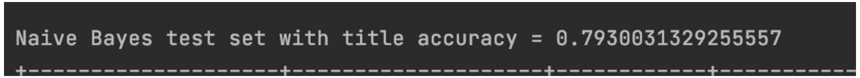
```
Random Forest test set with title  accuracy = 0.8356959734580003
```



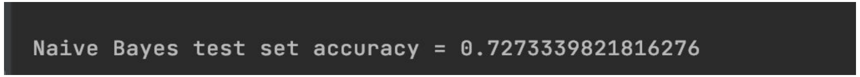
```
 Random Forest test set Accuracy = 0.7827976324267626
```



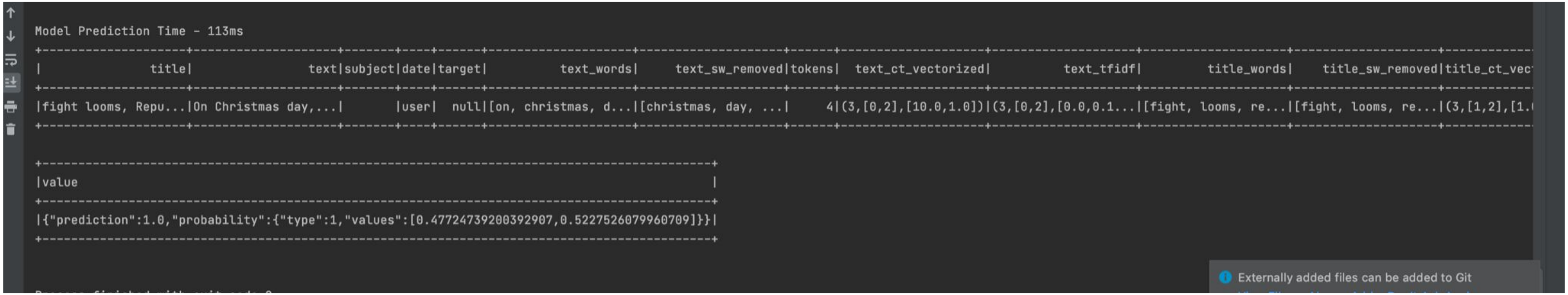
```
Naive Bayes test set with title and subject accuracy = 0.8906587272050615
```



```
Naive Bayes test set with title accuracy = 0.7930031329255557
+--------------------+--------------------+-----------+----------
```



```
 Naive Bayes test set accuracy = 0.7273339821816276
```

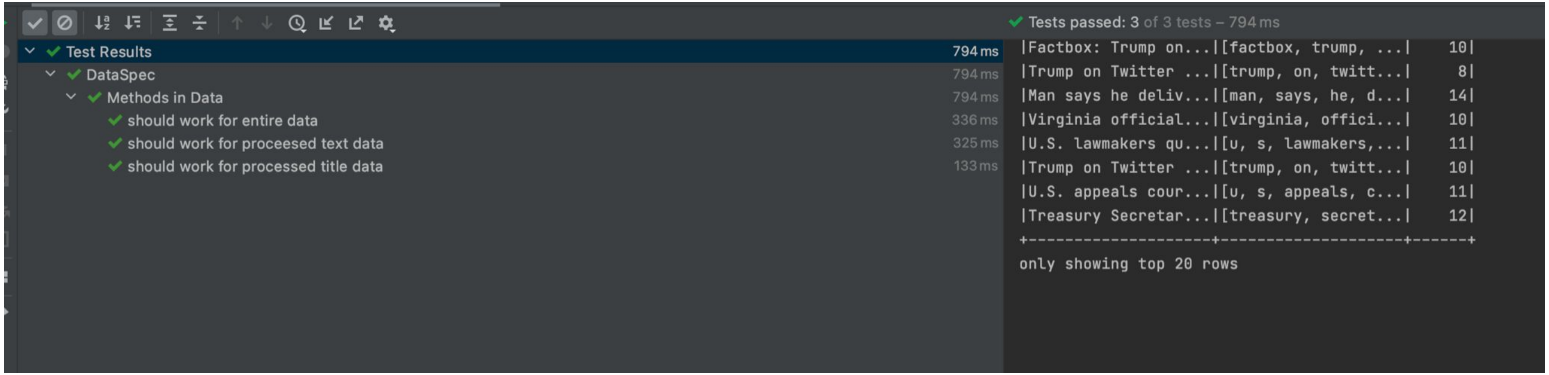


```
Model Prediction Time - 113ms
+--------------------+--------------------+-------+----+------+--------------------+--------------------+------+--------------------+--------------------+--------------------+--------------------+-----------
|               title|                text|subject|date|target|          text_words|     text_sw_removed|tokens|  text_ct_vectorized|          text_tfidf|         title_words|    title_sw_removed|title_ct_vec
+--------------------+--------------------+-------+----+------+--------------------+--------------------+------+--------------------+--------------------+--------------------+--------------------+-----------
|fight looms, Repu...|On Christmas day,...|       |user|  null|[on, christmas, d...|[christmas, day, ...|     4|(3,[0,2],[10.0,1.0])|(3,[0,2],[0.0,0.1...|[fight, looms, re...|[fight, looms, re...|(3,[1,2],[1.(
+--------------------+--------------------+-------+----+------+--------------------+--------------------+------+--------------------+--------------------+--------------------+--------------------+-----------


+---------------------------------------------------------------------------------------------+
|value                                                                                        |
+---------------------------------------------------------------------------------------------+
|{"prediction":1.0,"probability":{"type":1,"values":[0.47724739200392907,0.5227526079960709]}}|
+---------------------------------------------------------------------------------------------+


Process finished with exit code 0
```

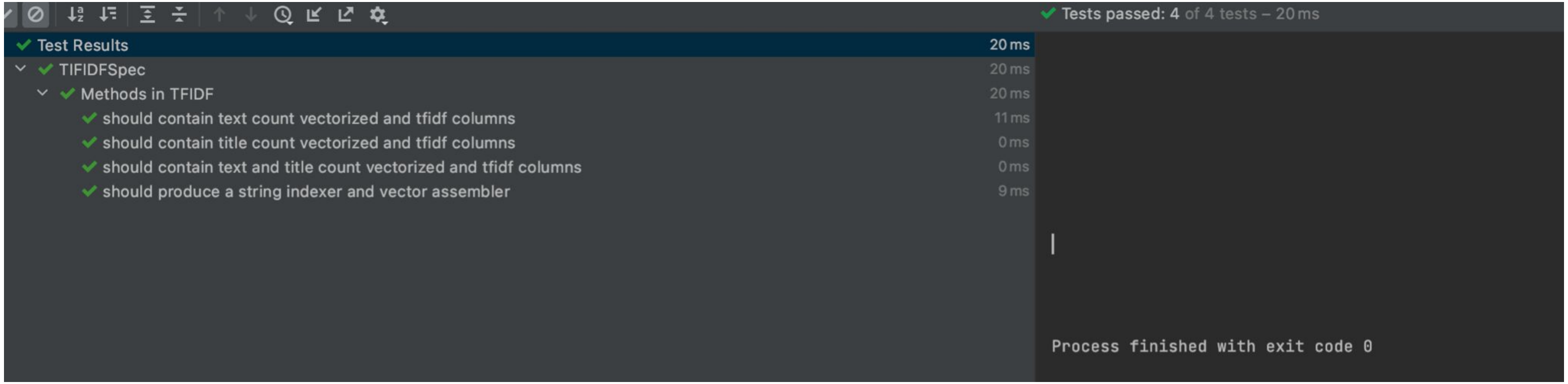
Unit Testing:



✓ Tests passed: 3 of 3 tests – 794 ms

```
✓ Test Results                                   794 ms
  ✓ DataSpec                                     794 ms
    ✓ Methods in Data                            794 ms
      ✓ should work for entire data              336 ms
      ✓ should work for proceesed text data      325 ms
      ✓ should work for processed title data     133 ms
```

```
|Factbox: Trump on...|[factbox, trump, ...|    10|
|Trump on Twitter ...|[trump, on, twitt...|     8|
|Man says he deliv...|[man, says, he, d...|    14|
|Virginia official...|[virginia, offici...|    10|
|U.S. lawmakers qu...|[u, s, lawmakers,...|    11|
|Trump on Twitter ...|[trump, on, twitt...|    10|
|U.S. appeals cour...|[u, s, appeals, c...|    11|
|Treasury Secretar...|[treasury, secret...|    12|
+--------------------+--------------------+------+
only showing top 20 rows
```



✓ Tests passed: 4 of 4 tests – 20 ms

```
✓ Test Results                                                          20 ms
  ✓ TIFIDFSpec                                                          20 ms
    ✓ Methods in TFIDF                                                  20 ms
      ✓ should contain text count vectorized and tfidf columns          11 ms
      ✓ should contain title count vectorized and tfidf columns          0 ms
      ✓ should contain text and title count vectorized and tfidf columns 0 ms
      ✓ should produce a string indexer and vector assembler             9 ms
```

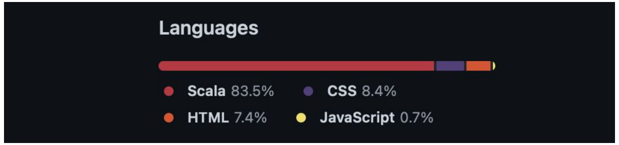
```
Process finished with exit code 0
```

Acceptance Criteria:

- UI to accept Users input and visualize the data
- Model Response < 5s
  - ○ Response time achieved < 1s
- Model Accuracy

65%

  - ○ Accuracy achieved through models



Languages

● Scala 83.5% ● CSS 8.4%

● HTML 7.4% ● JavaScript 0.7%

References:

- *https://spark.apache.org/docs/latest/ml-features.html*
- *https://www.tutorialspoint.com/apache_kafka/apache_kafka_real_time_application.htm*

Thank You