Fake News Classifier

DONE BY: GROUP CPFZ

AUSTIN PUA (U2222547C)

CALVIN (U2222207J)

PENG LIANG (U2222144J)

FAYE (U2221317A)

Our Dataset

- Fake and Real News Dataset
- Consist of 2 csv. files that has a list of news articles labelled ‘real’ and ‘fake’ respectively
- Use the text parameter of each article as our predictor

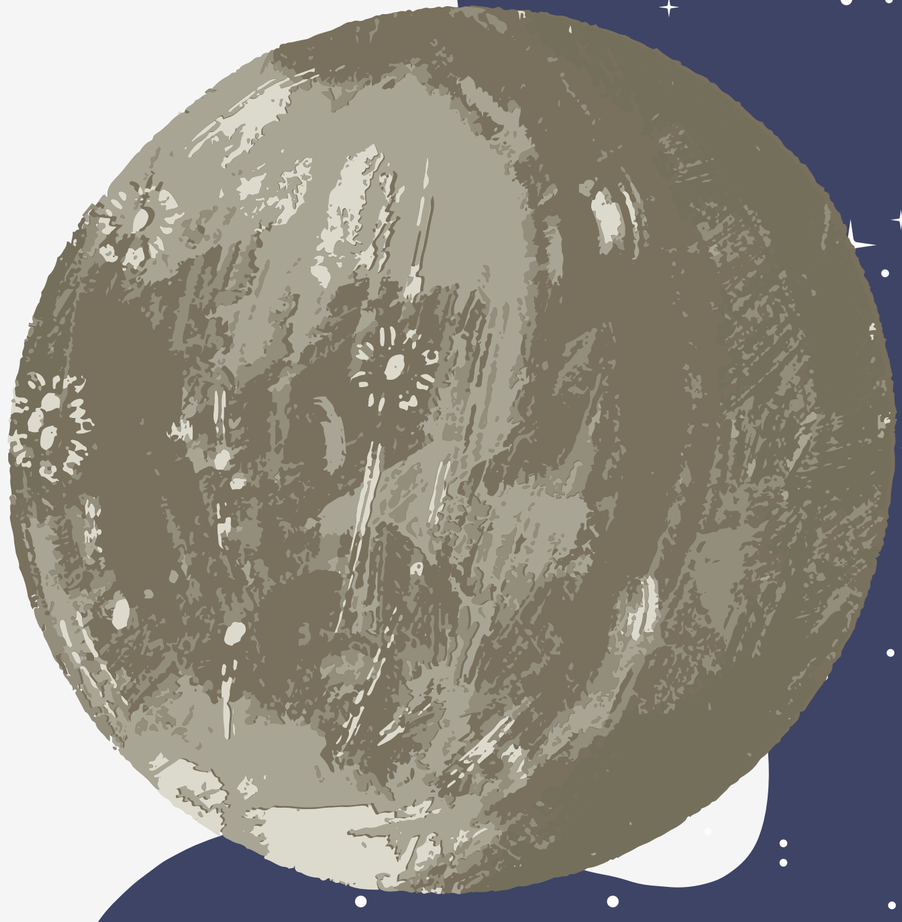
Problem Formulation

1. **Find Dataset:** Looked for an extensive data set
2. **Analyse Dataset:** **Look at the features it offers**
3. **Find Context:** **What is the significance of this data**
4. **Formulate:** **Articulate this into a problem to explore**

Problem Statement:
“Is the news article REAL or FAKE?”

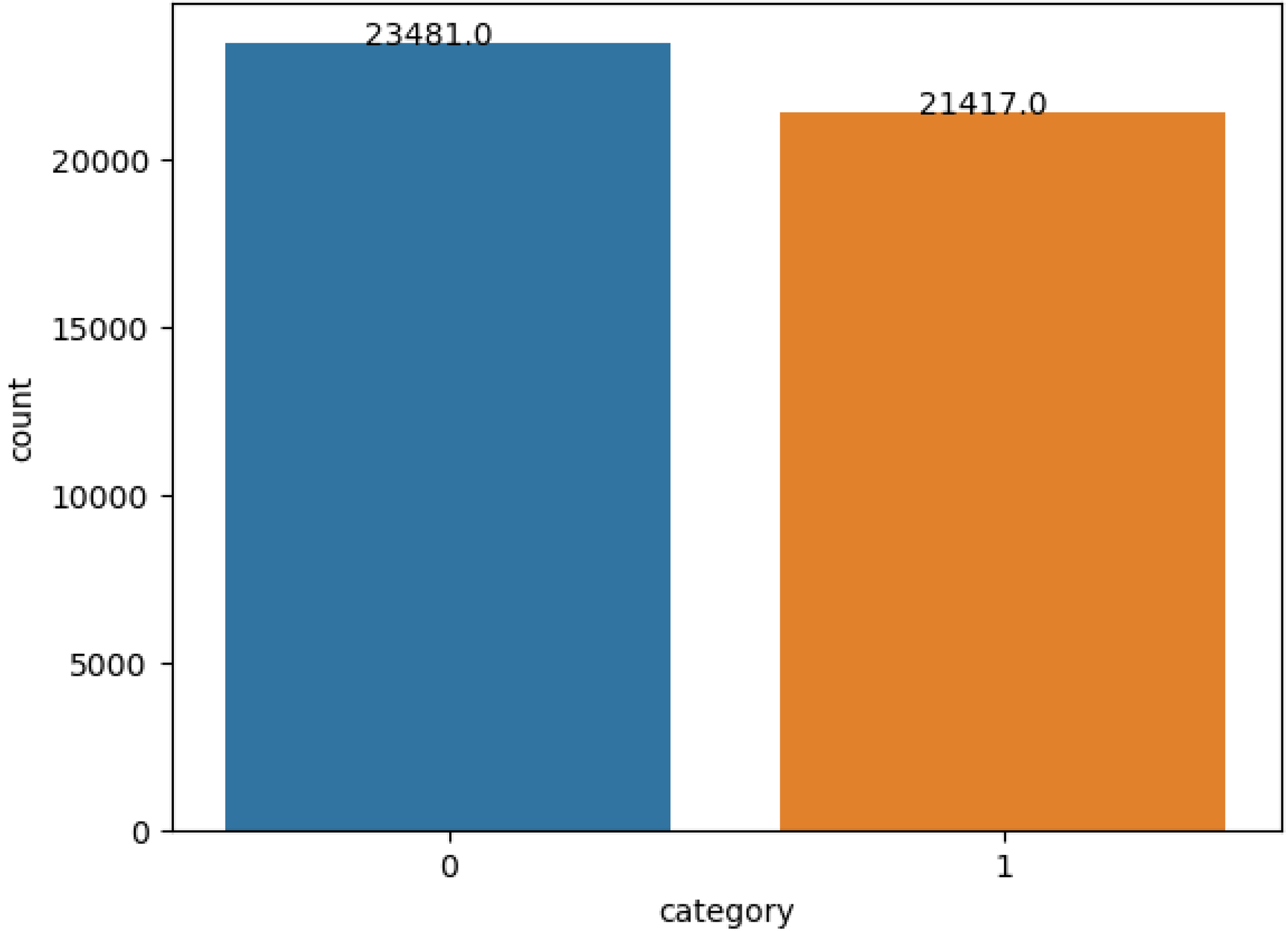
Exploratory Analysis

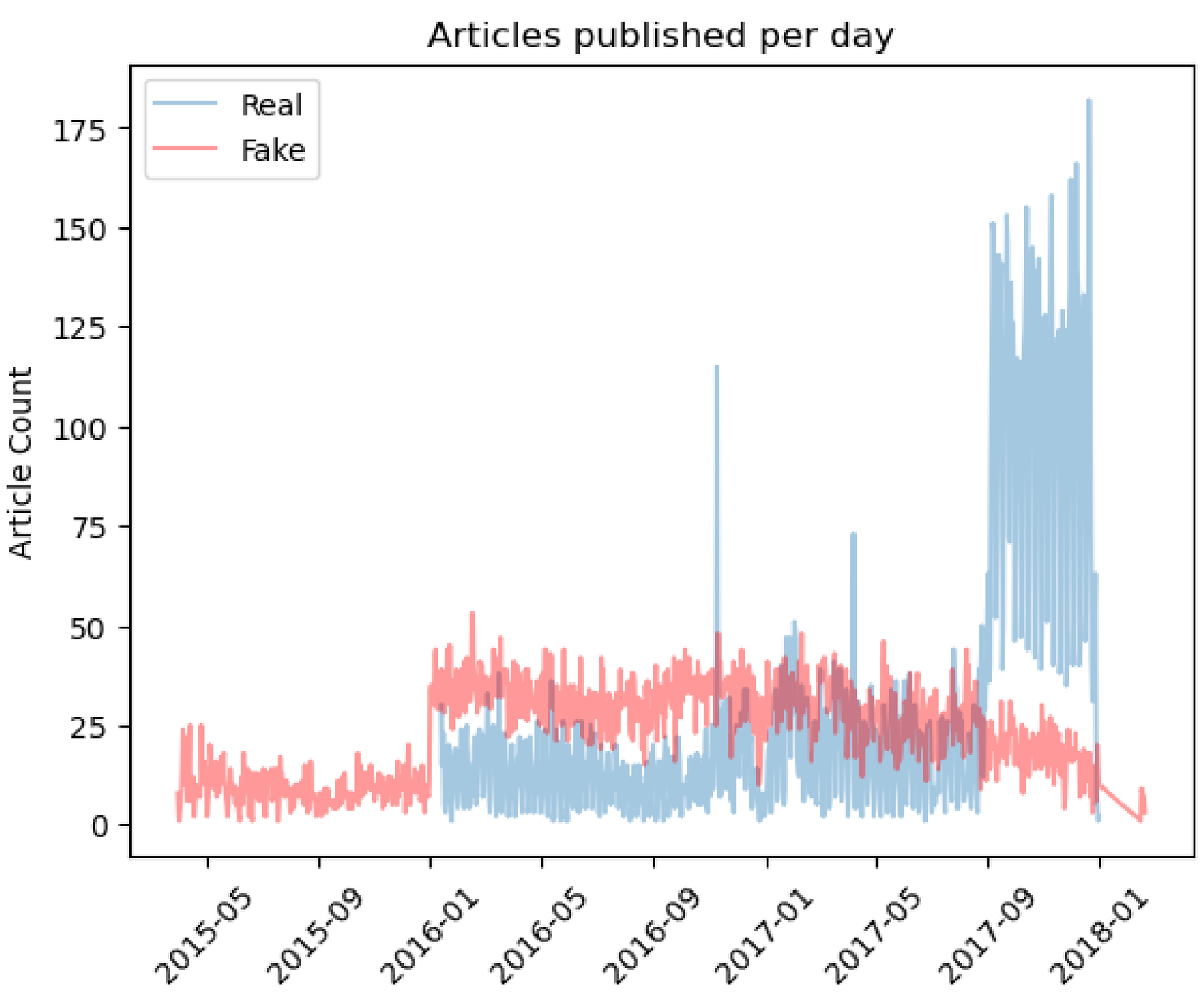
- Frequency analysis
- Subject analysis
- Length analysis



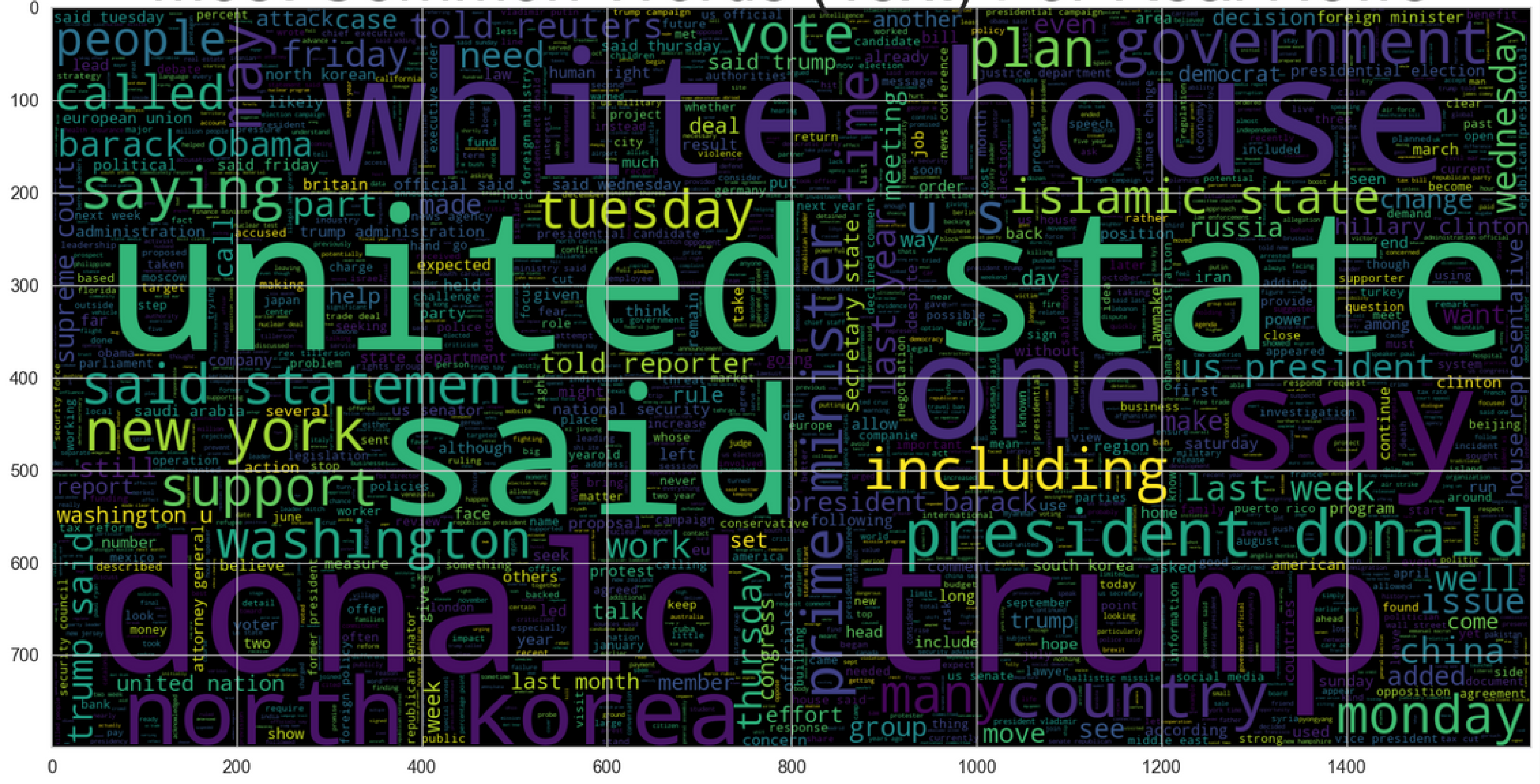
Frequency Analysis

- Number of real/fake news in the dataset
- Real/fake news articles published based on the date
- Most common words
- Unigram, Bigram, and Trigram analysis of words in articles

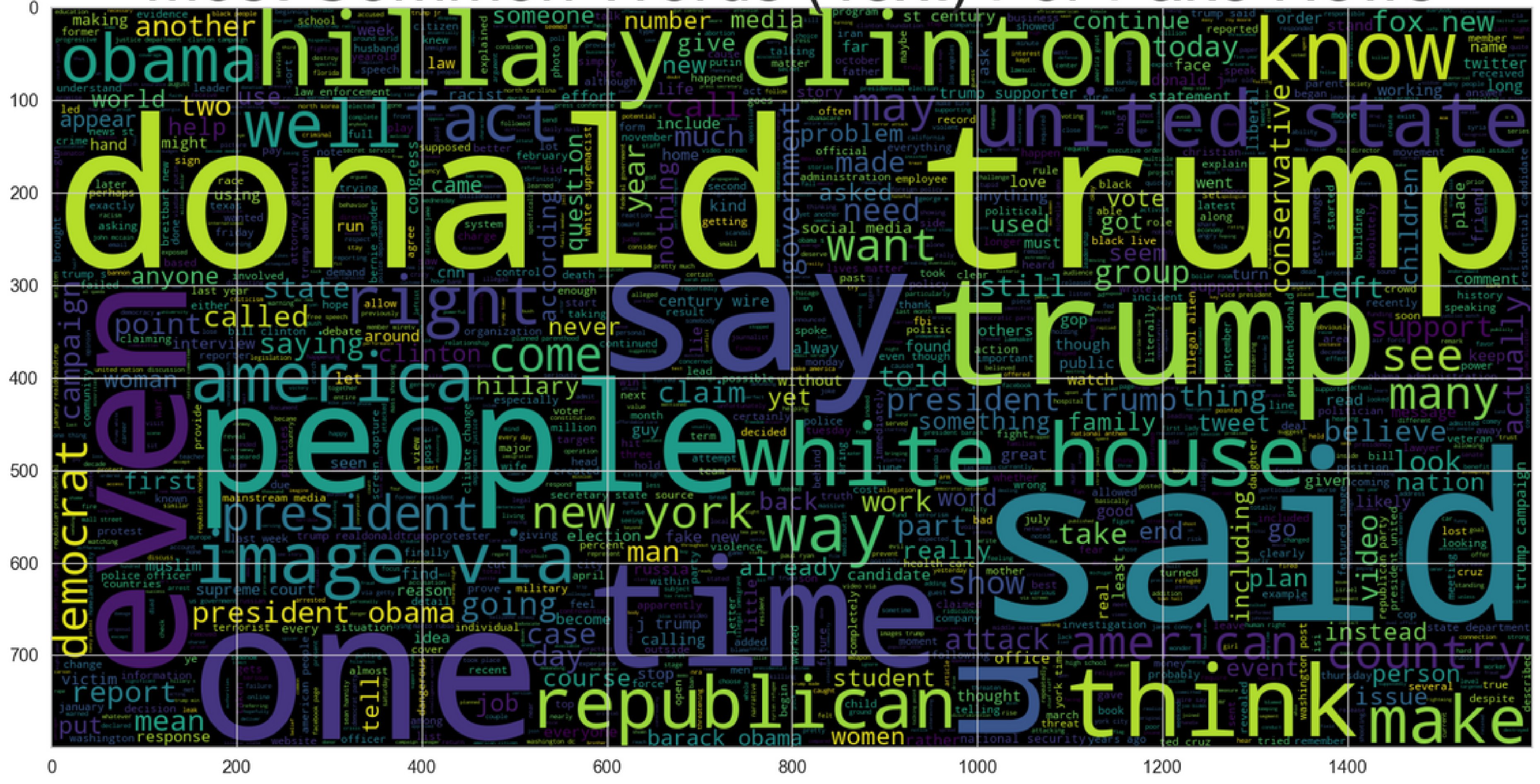

23481.0
21417.0
20000
15000
10000
5000
0
count
0
1
category

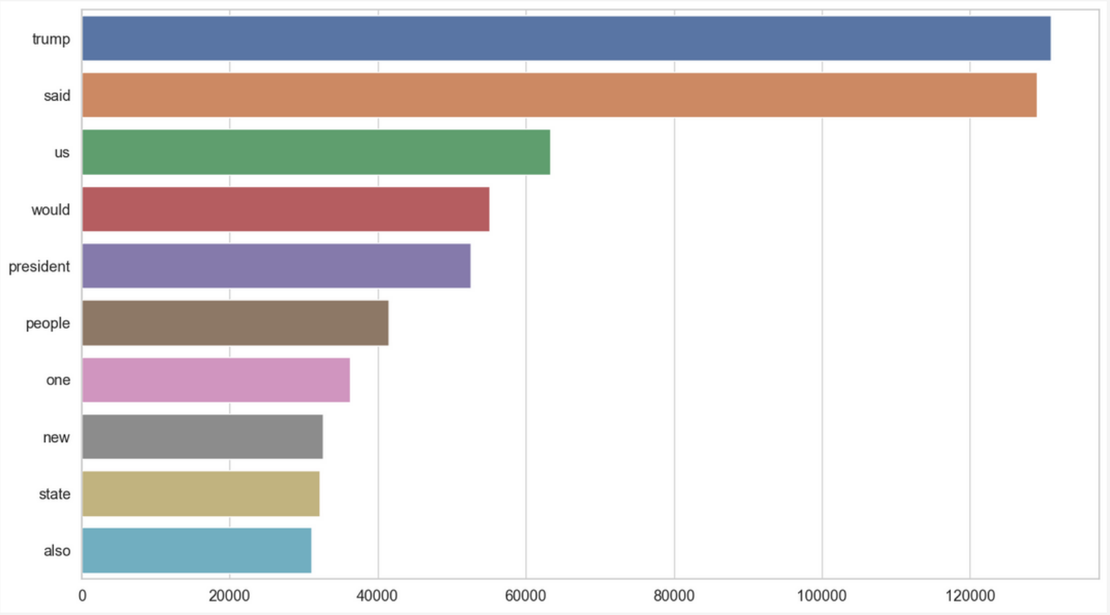

Articles published per day
Real
Fake
Article Count
175
150
125
100
75
50
25
0
2015-05
2015-09
2016-01
2016-05
2016-09
2017-01
2017-05
2017-09
2018-01

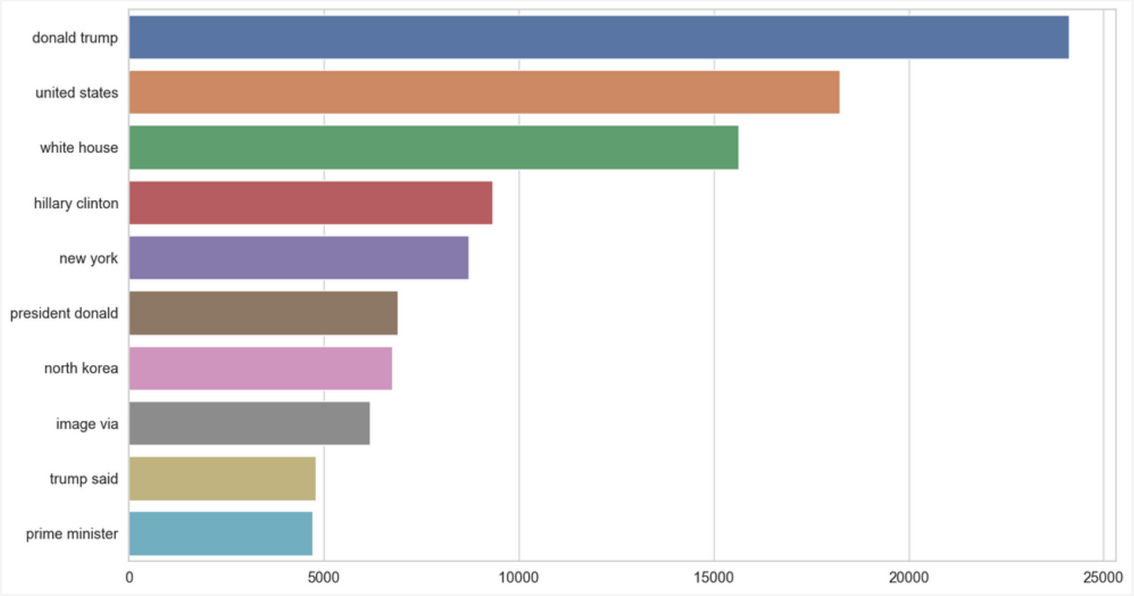
Most Common Words (Text) For Real News

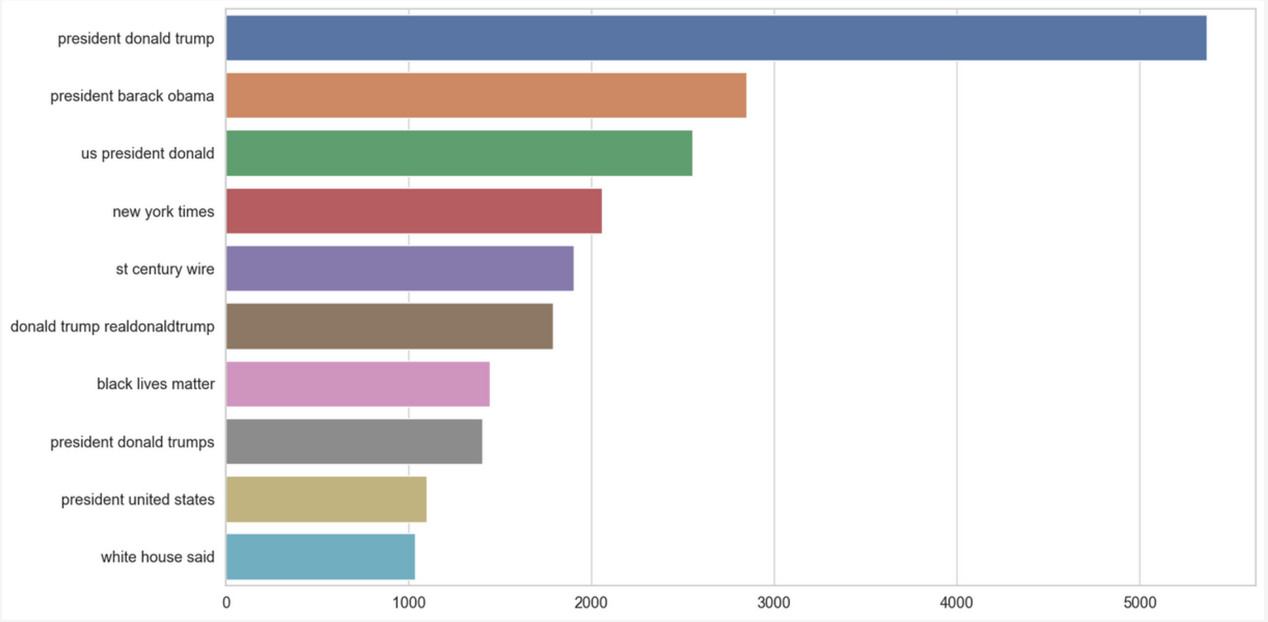


Most Common Words (Text) For Fake News



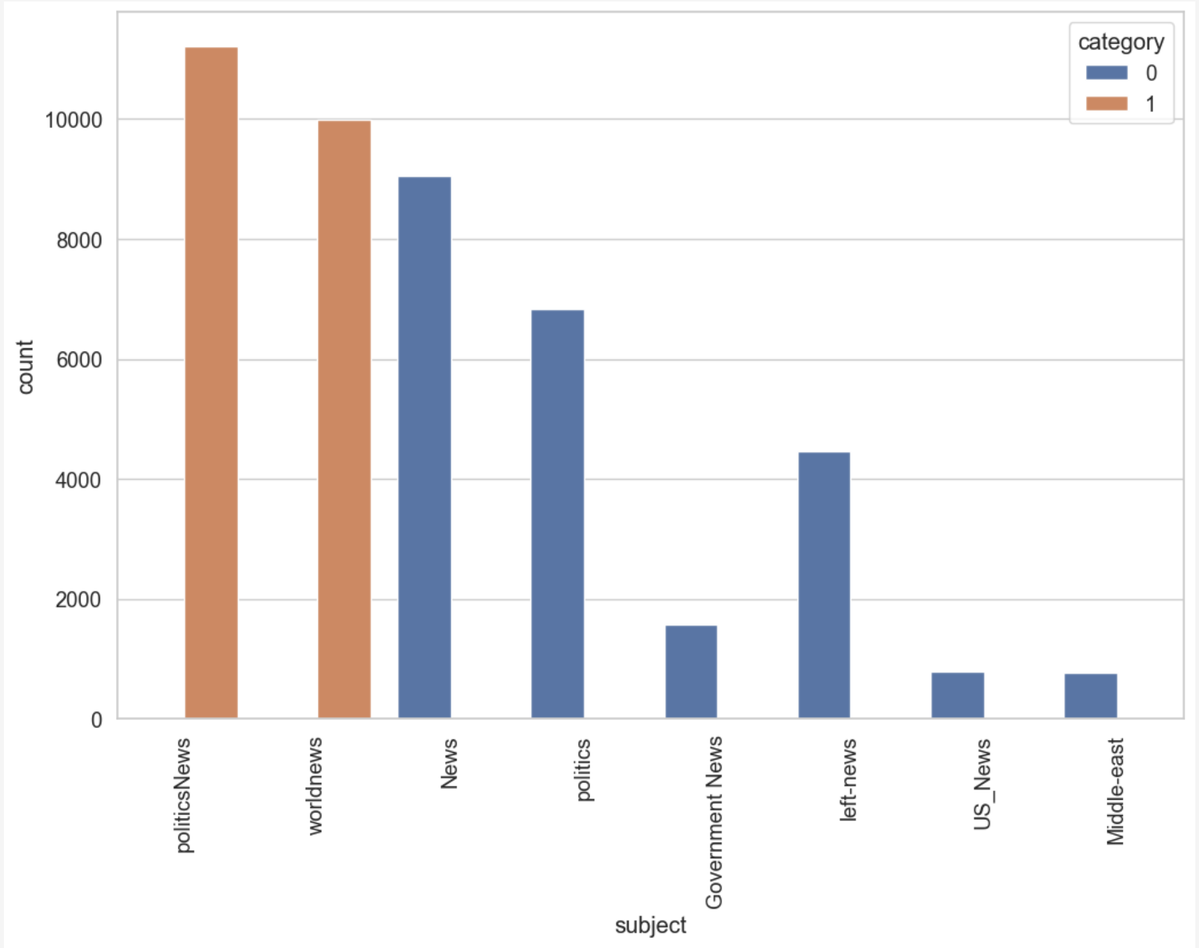

trump
said
us
would
president
people
one
new
state
also
0
20000
40000
60000
80000
100000
120000

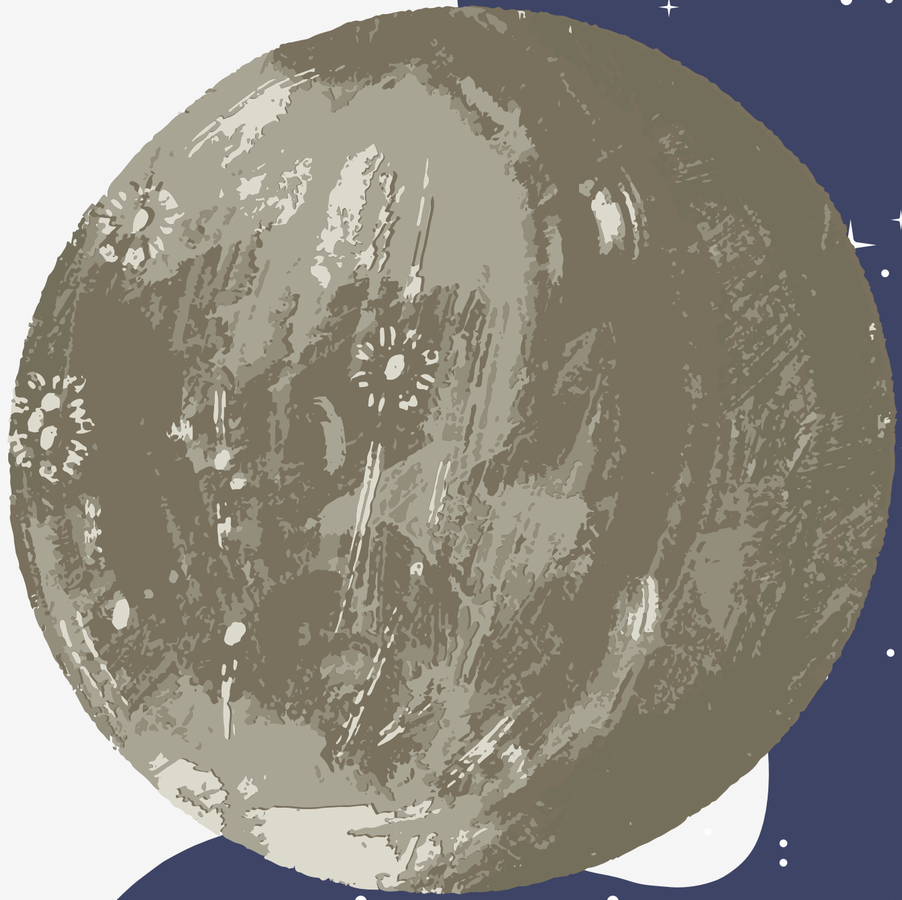

donald trump
united states
white house
hillary clinton
new york
president donald
north korea
image via
trump said
prime minister
0
5000
10000
15000
20000
25000


president donald trump
president barack obama
us president donald
new york times
st century wire
donald trump realdonaldtrump
black lives matter
president donald trumps
president united states
white house said
0
1000
2000
3000
4000
5000

Subject Analysis

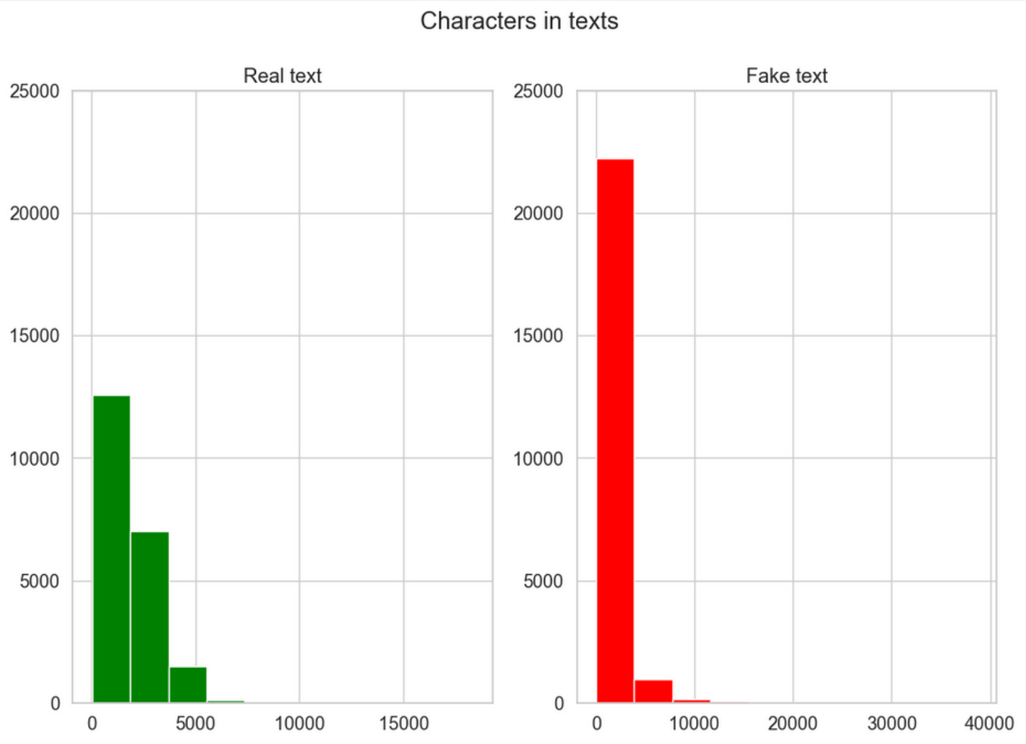
Types of news in Real/Fake articles

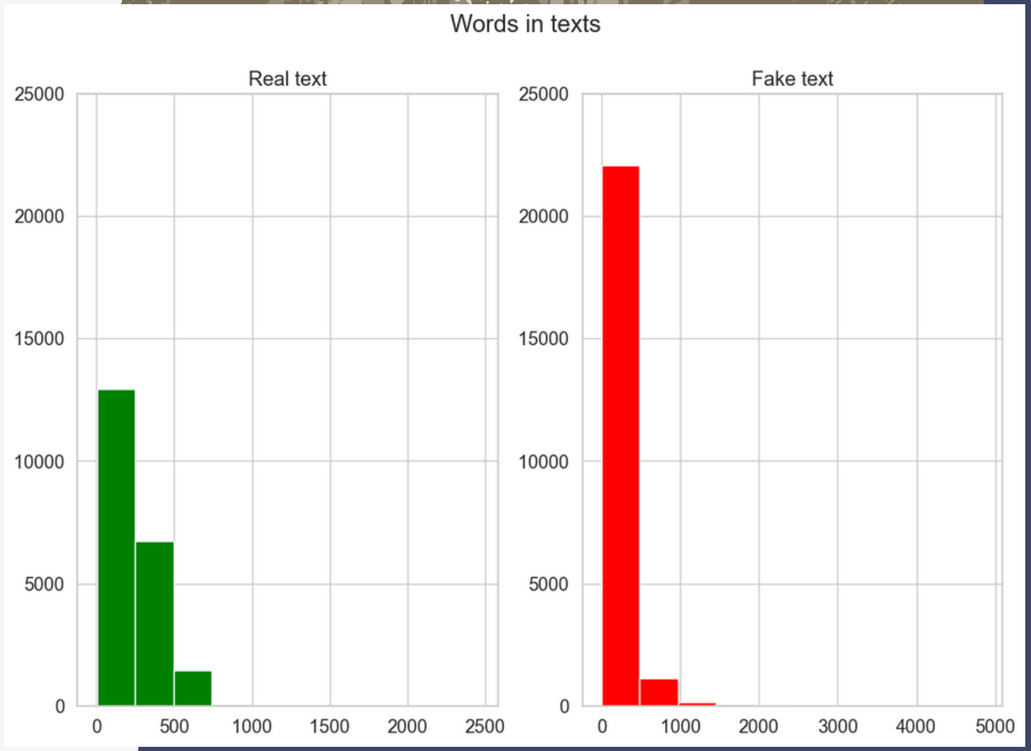


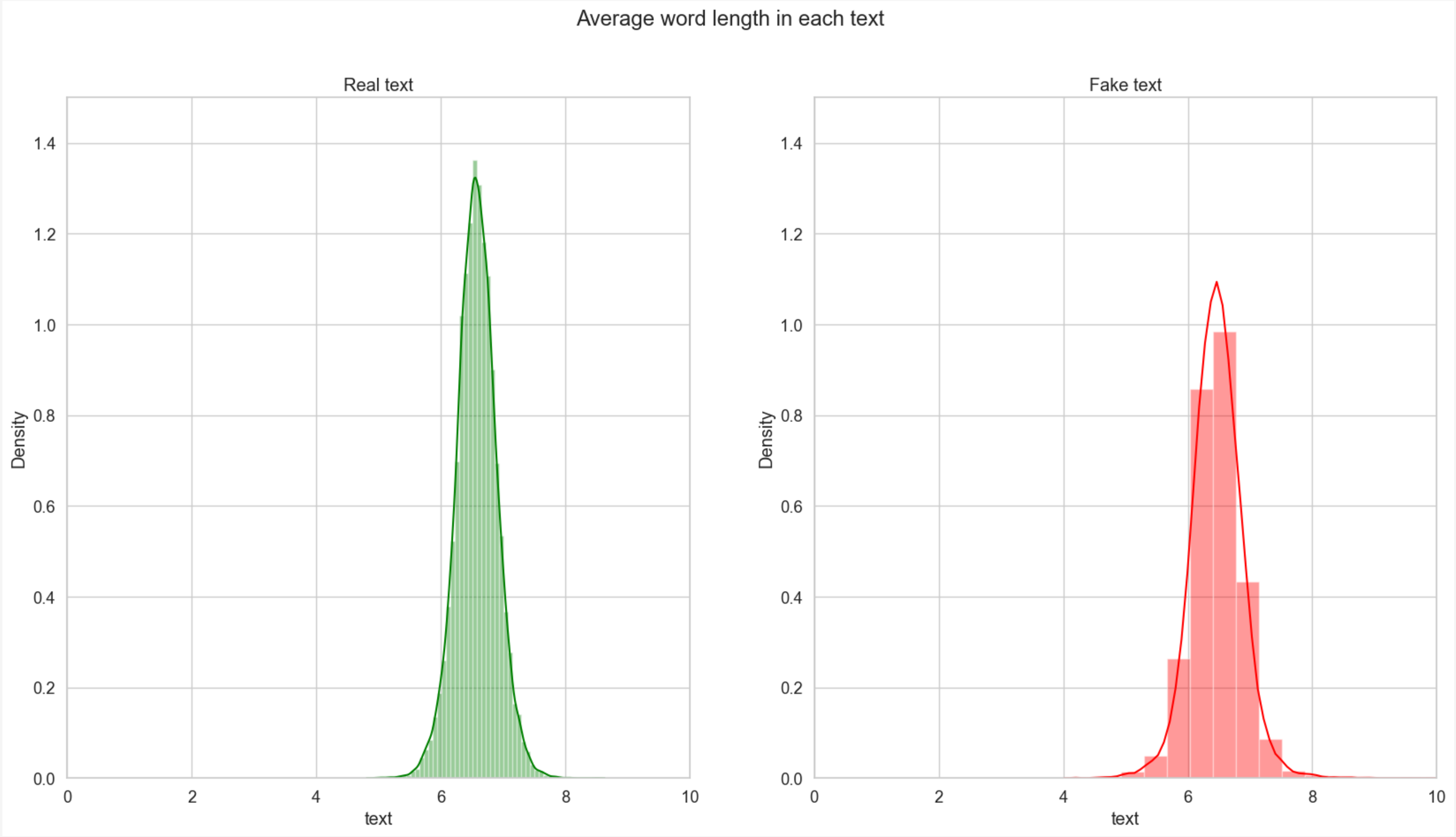


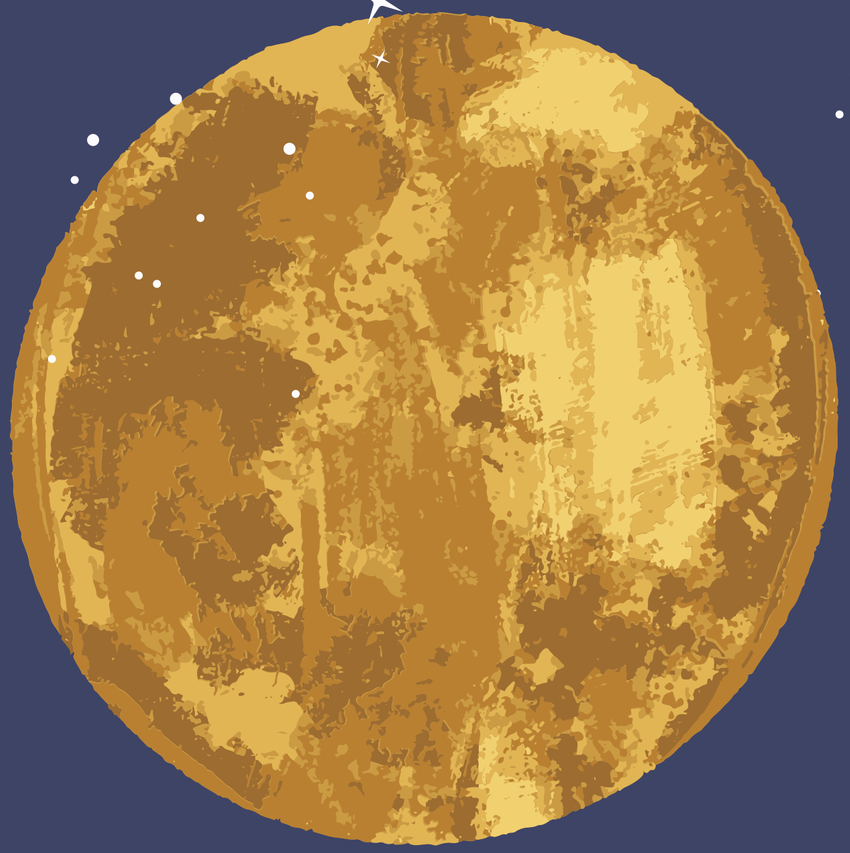
Length Analysis

Distribution of Real/Fake news article lengths






Average word length in each text
Real text
Fake text
Density
text
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
2
4
6
8
10



Data Cleaning

- Removing ‘Reuters’

```
print(true.text.str.contains("\(Reuters\)").mean()*100,"%")
```

```
99.20623803520567 %
```

Data Cleaning

Removing duplicate articles

```
df.duplicated().sum()
print("Before removing:")
df.describe()
```

Before removing:

	category
count	44898.000000
mean	0.477015
std	0.499477
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

```
df.drop_duplicates(inplace=True)
print("After removing:")
df.describe()
```

After removing:

	category
count	44689.000000
mean	0.474636
std	0.499362
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

Data Cleaning

- Stopwords - reducing noise and improving efficiency of text processing

```
# printing the stopwords in English
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Preprocessing: Vectorization

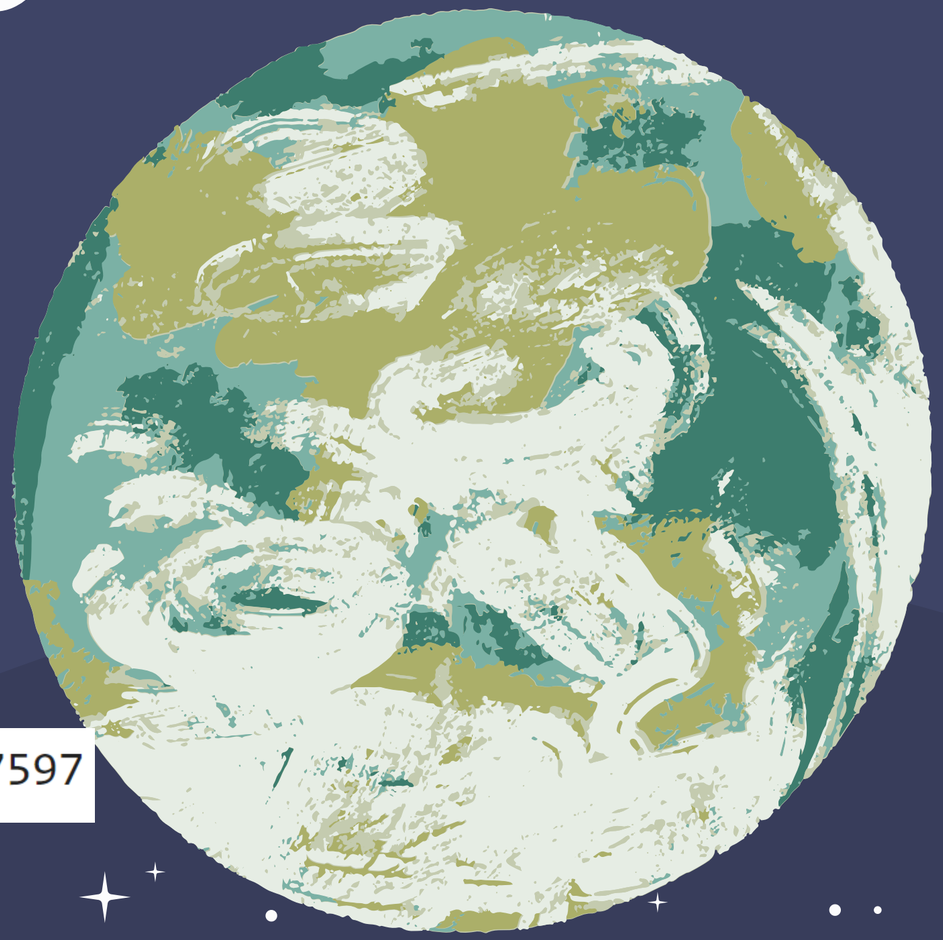
- Term Frequency (TF)
- Inverse Document Frequency (IDF)
- TF-IDF Calculation
- Vectorization

Preprocessing: Vectorization

- CountVectorizer
- n-grams

Preprocessing: Comparison

- E.g.: For Logistic Regression



TfidfVectorizer:

```
Accuracy: 0.983890132020586
```

CountVectorizer:

```
Accuracy: 0.989706869457597
```

CountVectorizer with n-grams:

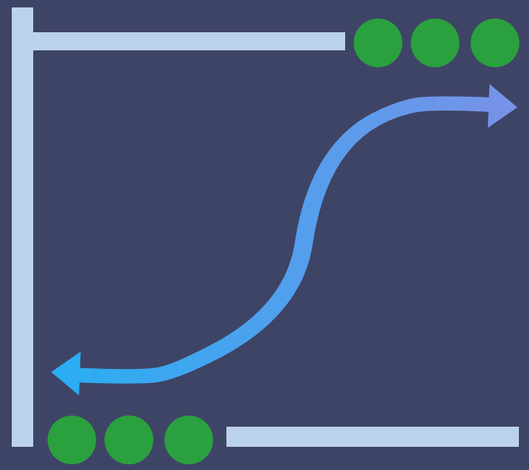
```
Accuracy: 0.991161336316849
```

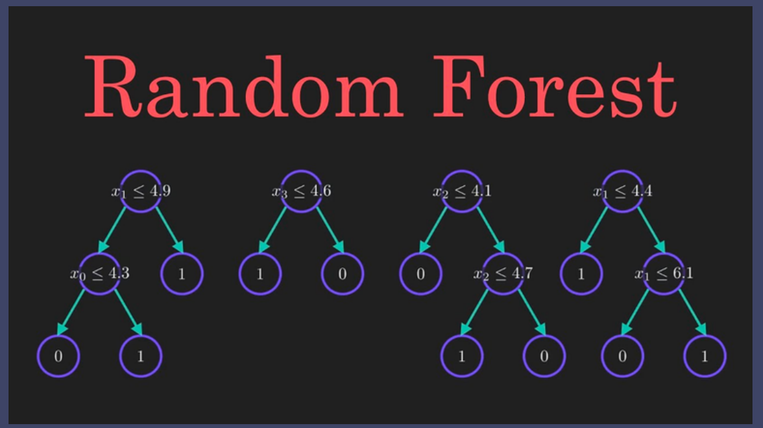
Machine Learning Algorithms

- Logistic Regression
- Random Forest
- Naive Bayes
- Passive Aggressive Classifier
- XGboost

Machine Learning Algorithms

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$





Naive Bayes

Based on Bayes Theorem
Assumes conditional independence for each observation

Works well in sentiment analysis

Logistic Regression

Maps input and output using a sigmoid function

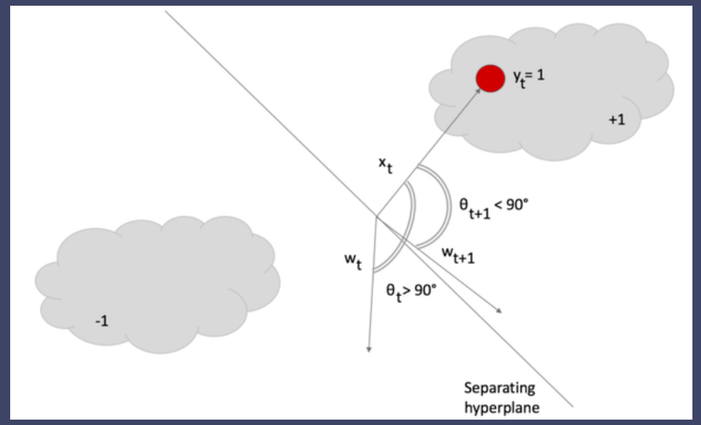
Commonly used in binary classification problems

Random Forest

Builds a multitude of decision trees and merges their predictions by averaging or voting on the prediction of each tree

Used in classification and regression tasks

Machine Learning Algorithms



Passive Aggressive

Passive - updates with small learning rate for correct predictions
Aggressive - larger updates for incorrect predictions

Commonly used for online learning tasks, such as text classification and fraud detection

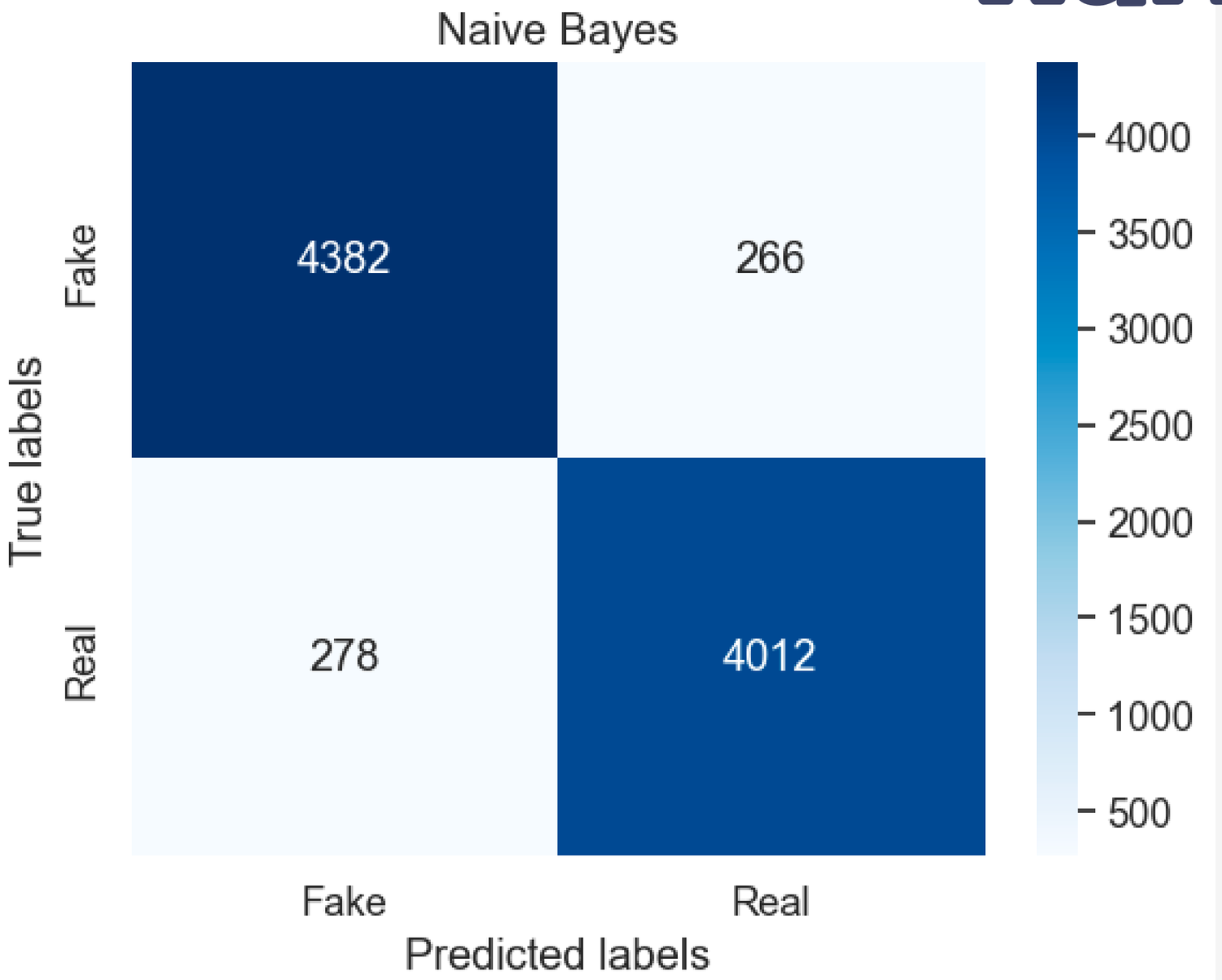
XGBoost

Extreme Gradient Boosting

Builds a series of decision trees, merging their predictions. Utilizes regularization terms in objective function, preventing overfitting.

Widely used in various ML competitions and real-world applications. Good with structured/tabular data, handles both classification and regression tasks well.

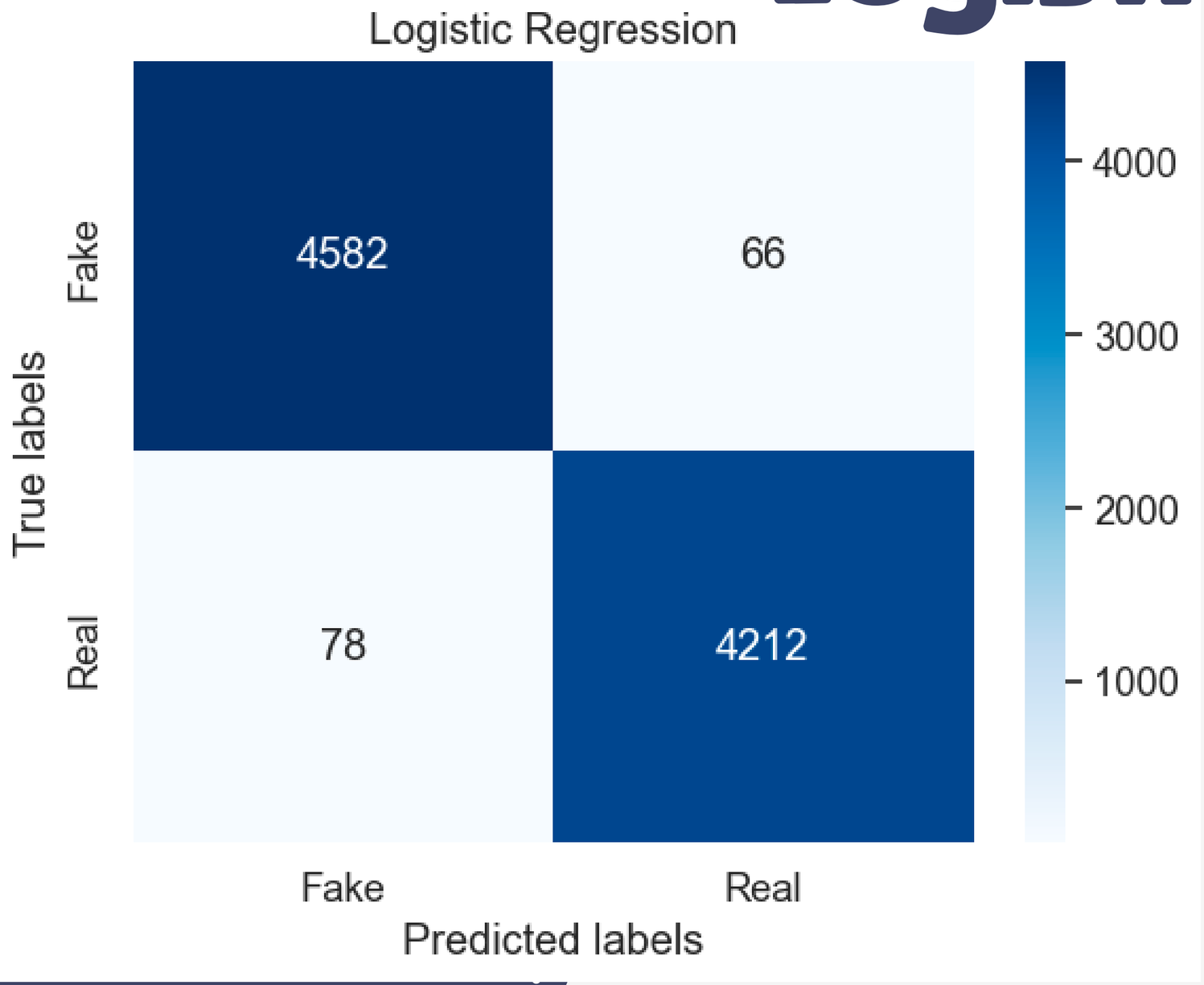
Naive Bayes



Accuracy - 0.939136

Precision - 0.937821

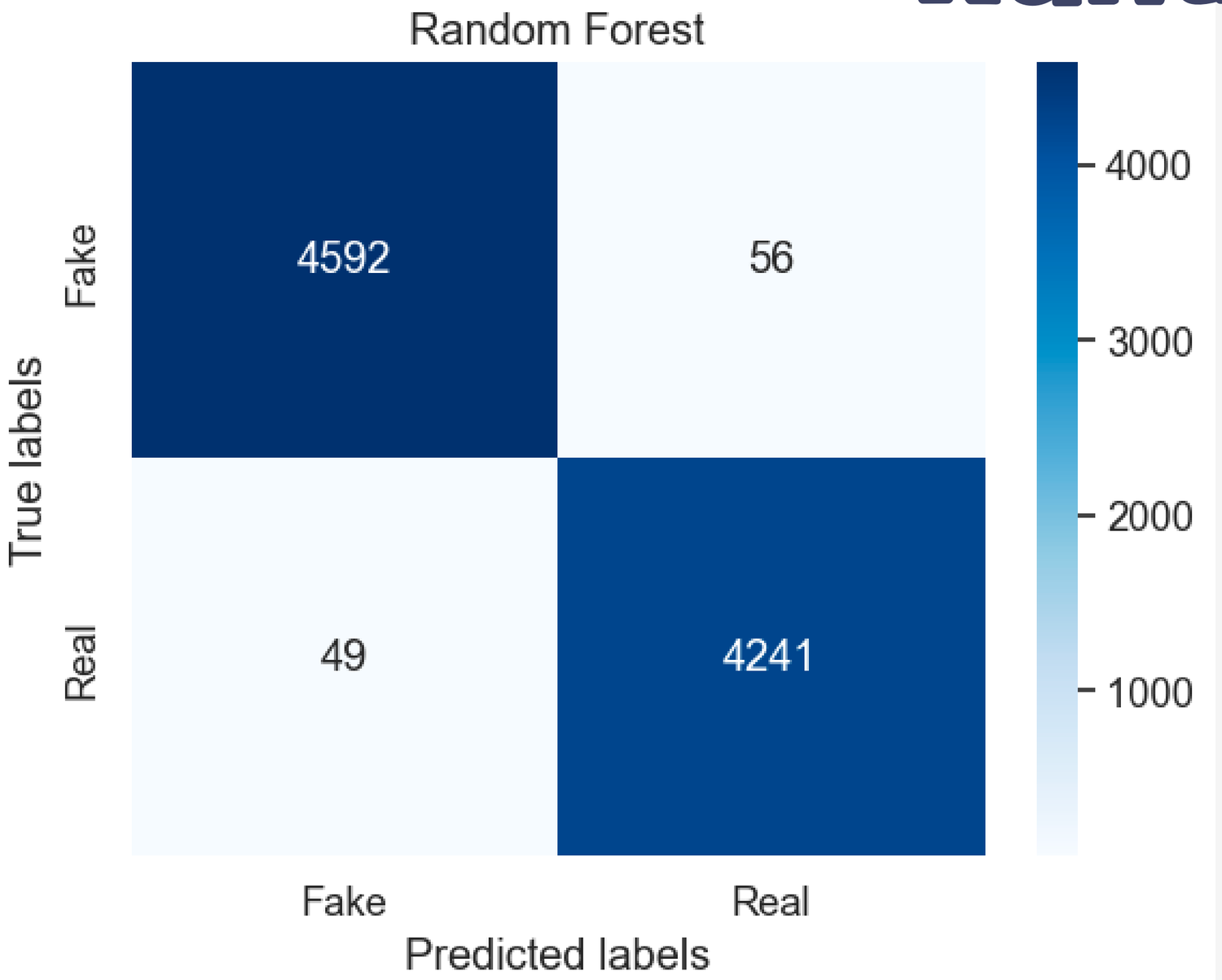
Logistic Regression



Accuracy - 0.983889

Precision - 0.984572

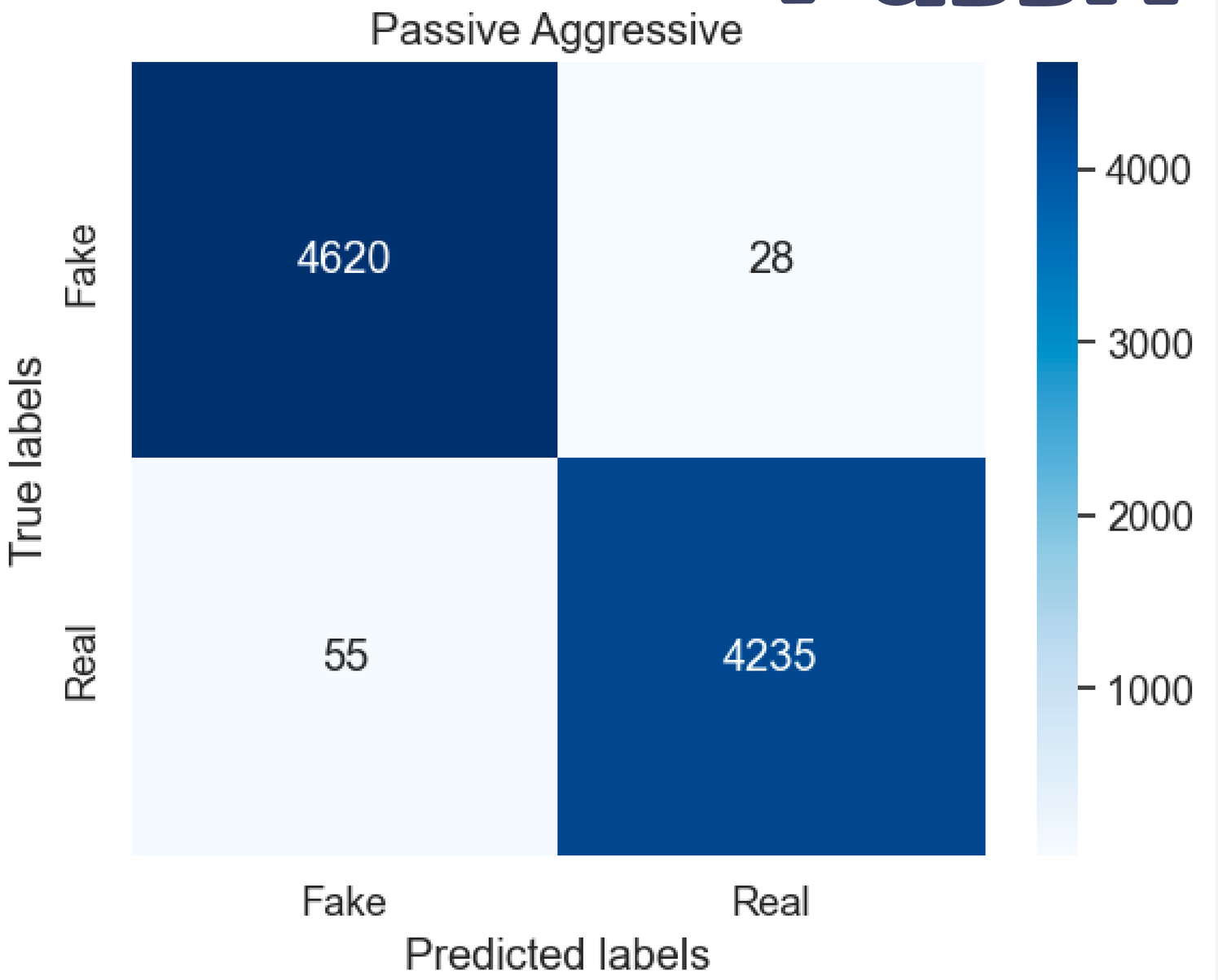
Random Forest



Accuracy - 0.988252

Precision - 0.986967

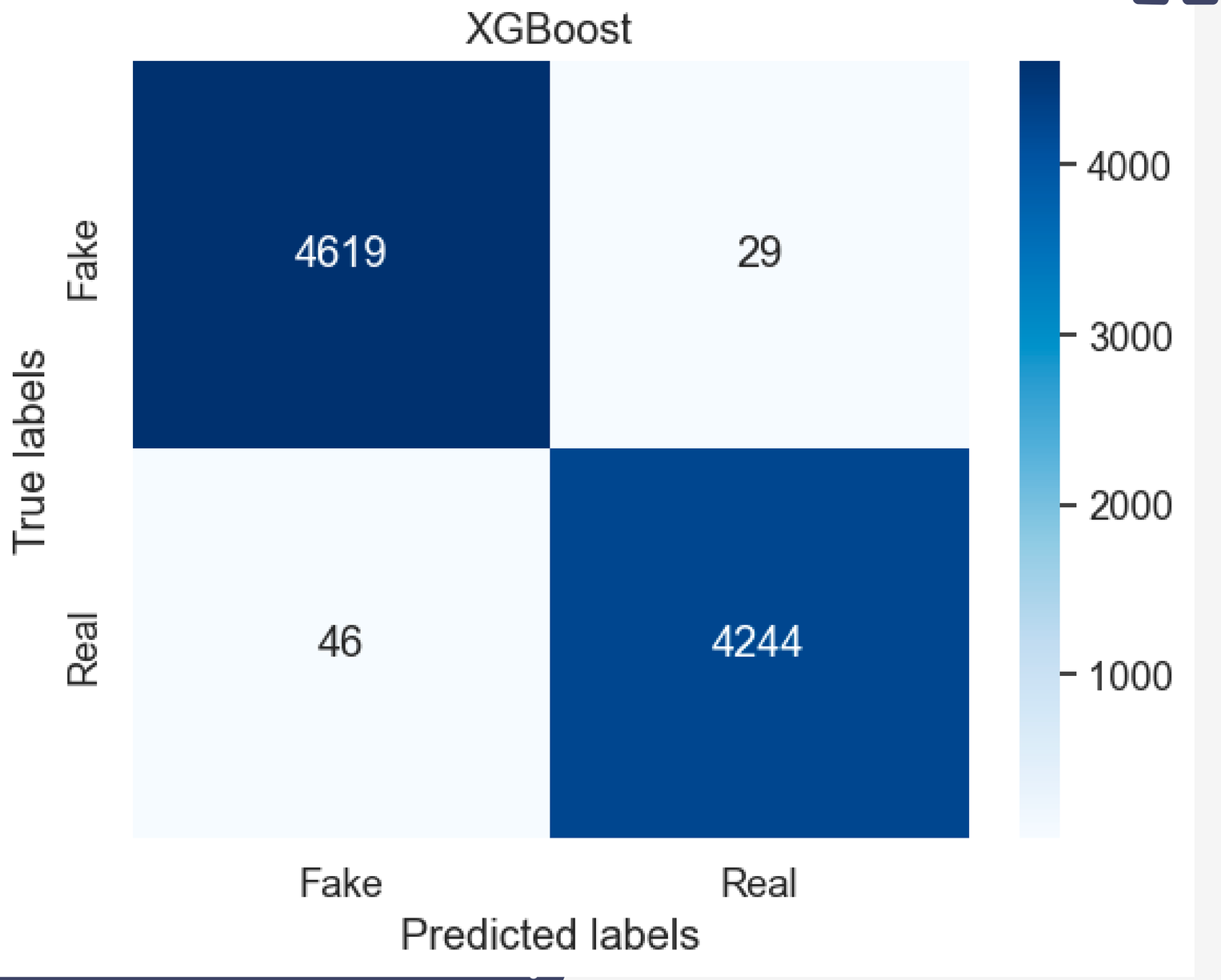
Passive Aggressive



Accuracy - 0.990713

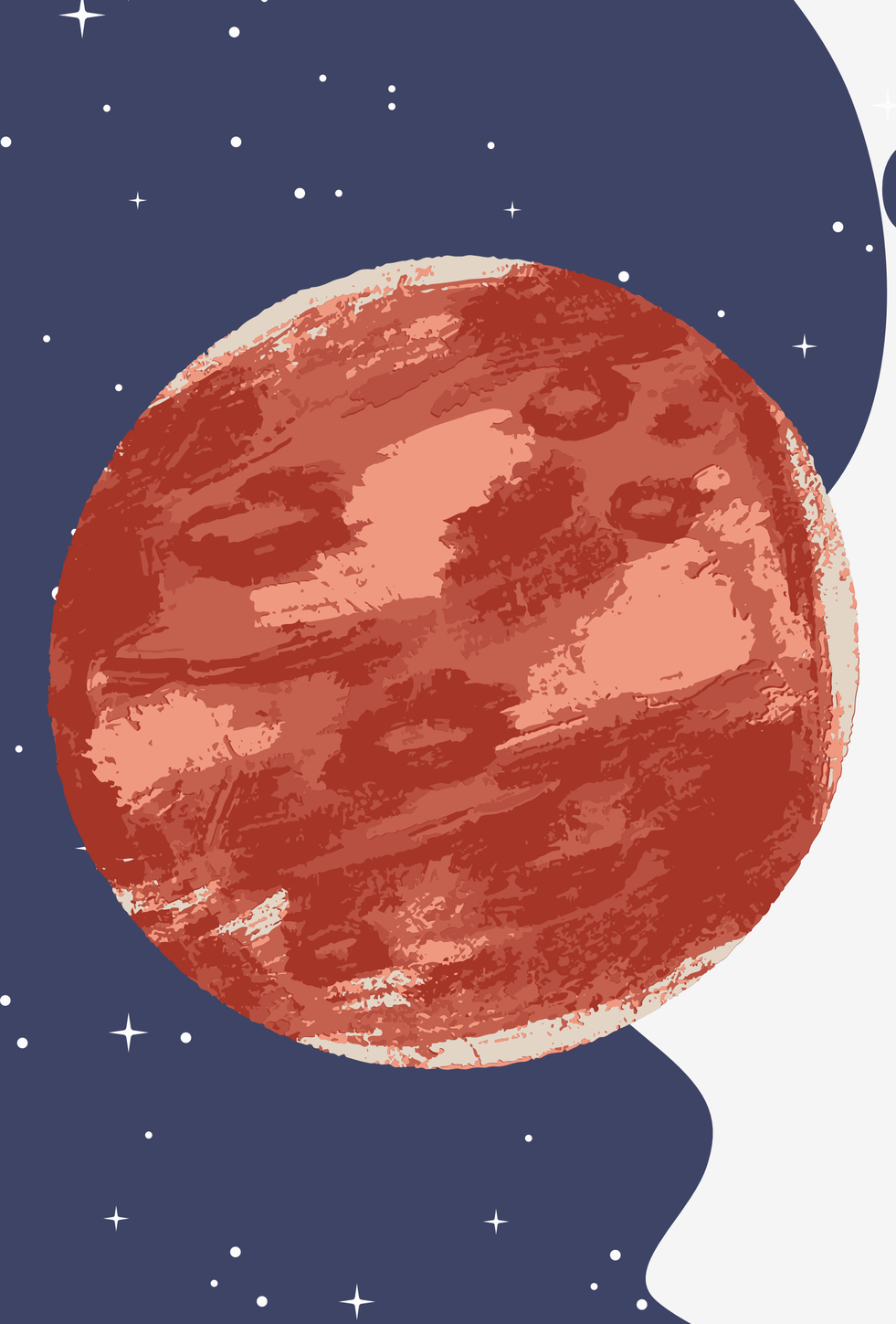
Precision - 0.993432

XGBoost



Accuracy - 0.991608

Precision - 0.993213



Comparing the Models

- Good performance all around
- Passive Aggressive and XGBoost best for accuracy
  - 99% accuracy
- Naive Bayes best for speed
  - 76 times faster than Random Forest
  - 93% accuracy

Naive Bayes

Advantages:

- Simple to implement
- Easy to interpret model

Disadvantages:

- Assumes features are independent of one another, which is not the case in language

Logistic Regression

Advantages:

- Straightforward and easy to understand. Output can be seen as probabilities.
- Low computational requirements. can be applied efficiently in real-time.

Disadvantages:

- Not capable of capturing intricate relationships

Random Forest

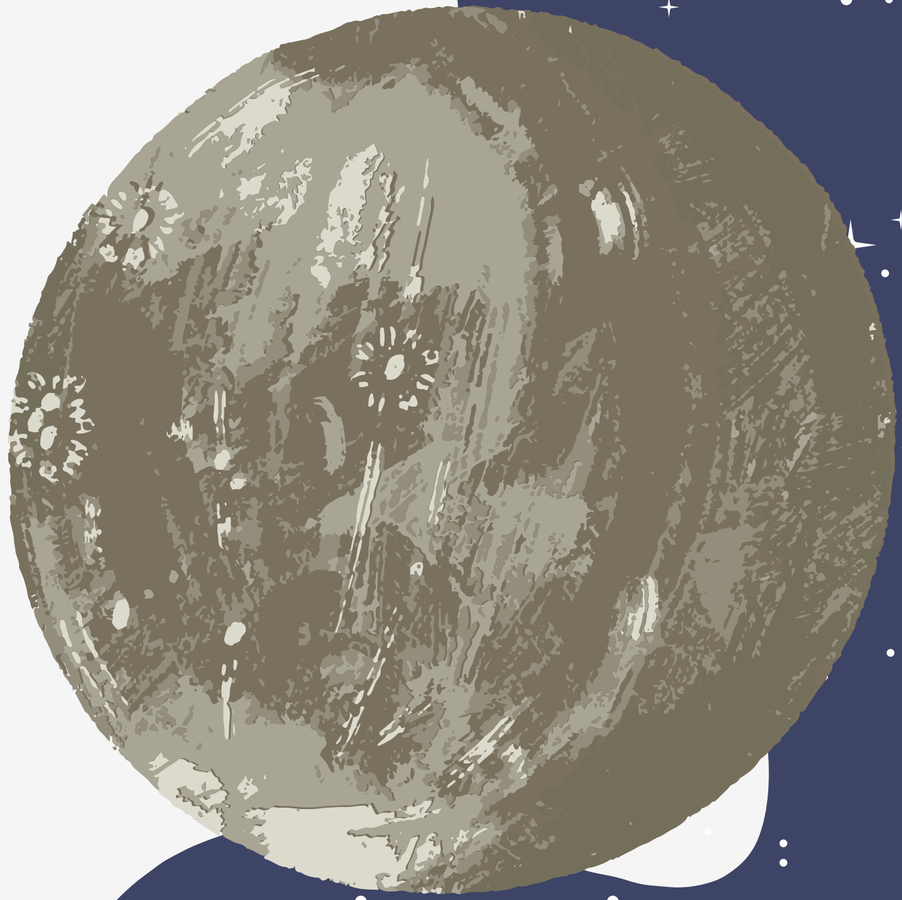
Advantages:

- Reduces overfitting and helps to improve accuracy
- Easily interpretable

Disadvantages:

- Requires a lot of computational power as well as time for training(59.2737 seconds)

Passive Aggressive

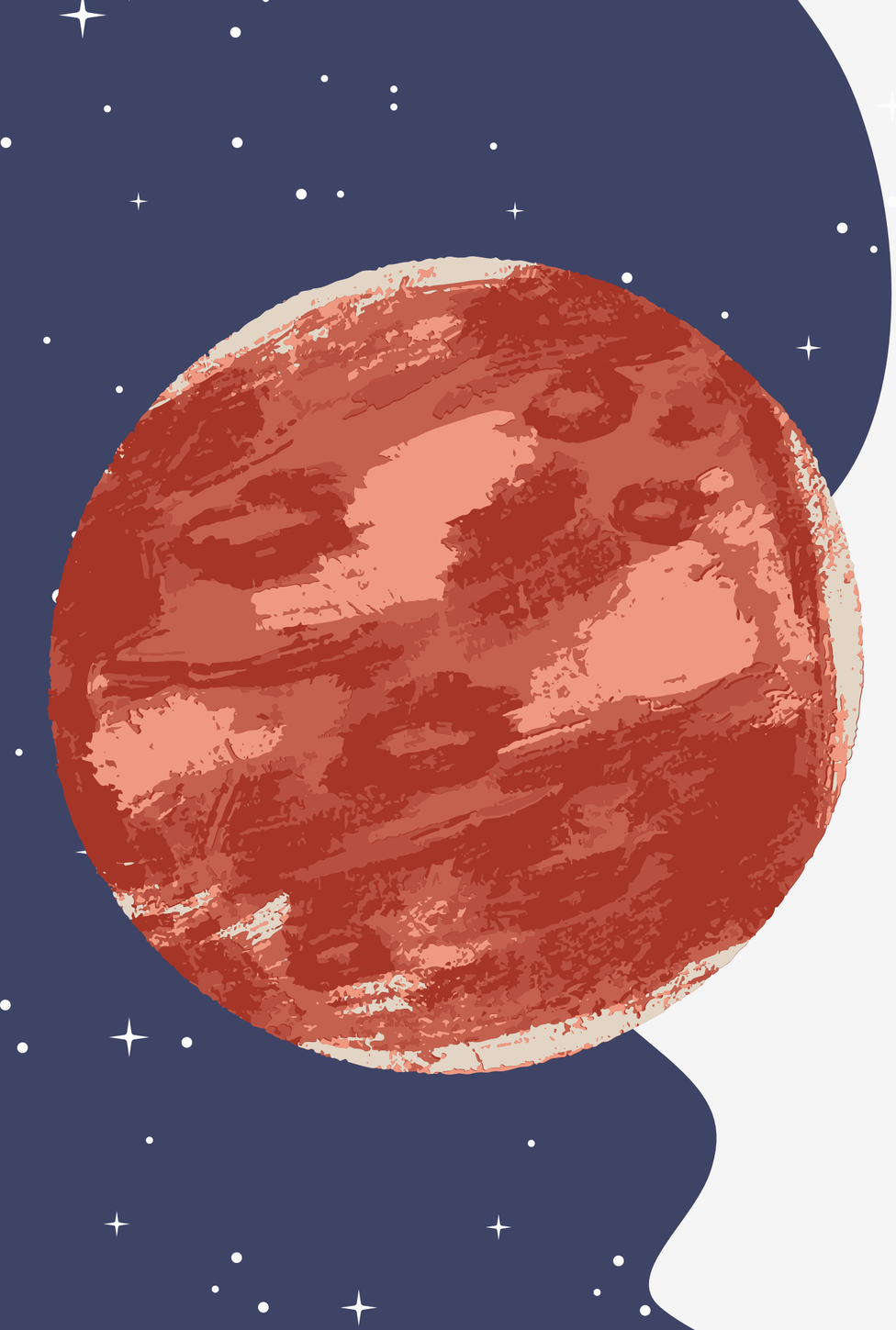


Advantages:

- Useful for handling large number of input data due to it being an online-learning algorithm which allows its learning model to be updated step by step
- Allows it to adapt to changes in language patterns and adapt to new information efficiently

Disadvantages:

- Limited interpretability of model



XGBoost

Advantages:

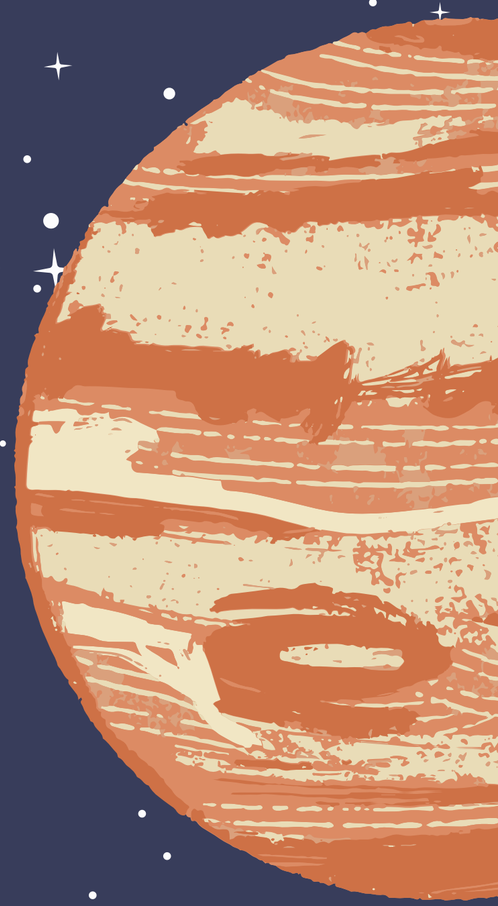
- XGBoost is known for its high accuracy
- It is designed to be fast and efficient
- Able to handle missing words

Disadvantages:

- Complex algorithm which requires high degree of technical expertise to implement
- Lack of transparency due to XGBoost being a “black box” algorithm, which makes it hard to understand how it arrives at its prediction

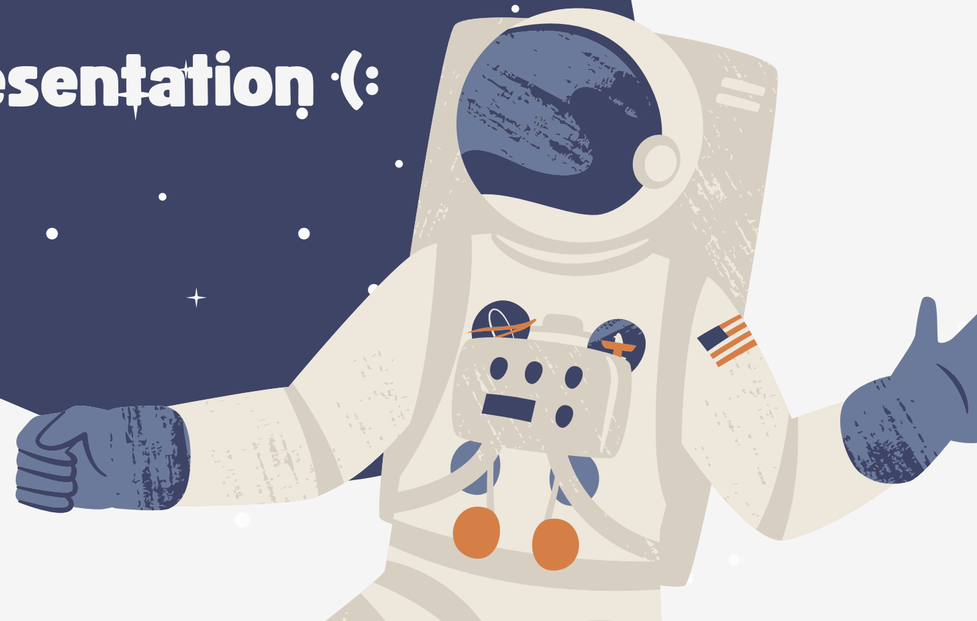
Conclusion

- Fake news articles/scams are rampant on the internet nowadays. Urgent need for fact-checking and fake news detection.
- Trends and patterns in fake and real news are distinguishable, allowing machine learning to detect fake news for large databases of articles
- Limitations: Dataset consists mostly of USA news, model does not work as well for local news.
- Hence the need for more, higher quality datasets of news globally to train different models



Thank you Prof Wesley!

We hope you enjoyed our presentation (:



Individual Contributions

Name	Contributions
Austin	Background Info, Exploratory Analysis
Faye	Data Cleaning and Text Preprocessing
Calvin	ML Models
Peng Liang	Comparing of Models