Distinguishing Real and Fake News

One of the most popular buzzwords in recent political history is “Fake News”. Coined by Donald Trump, this word is used to describe a piece of media that is considered biased or fraud. A big issue facing the today’s world is that it can be difficult to distinguish what news is real and what is fake. None of us want to be misinformed, and that is why our group has come up with a solution to this popular problem, through data analysis.

Our **four main goals** for this project were:

1. *To compare sentiment analysis results for fake vs real news articles from a CSV file*
2. *To compare the most and least commonly used words in fake and real news articles*
3. *To visualize the data using charts and WordClouds*
4. *To create a machine learning model that can determine which articles are fake/real*

First, by using sentiment analysis, our group hopes to see if there’s any sort of bias in fake news articles compared to real news articles. For instance, sentiment analysis allows us to look at the positive, negative, and neutral sentiment. Thus, we can see if there’s a higher amount of neutral sentiment in real news articles versus fake news articles which can then potentially indicate that real news is less biased.

Secondly, our group hopes to compare the most and least commonly used words and titles in real news and fake news so that we can see what were the most popular topics. By obtaining this data, our group can then visualize the data, using bar plots and word clouds, to get a better understanding of the differences and similarities between the topics in fake news versus real news.

Lastly, machine learning is becoming a popular tool for classifying data. As a result, we hope that by the end of this project our group can create a machine learning model that can accurately classify real news and fake news.

The Data

We obtained an online CSV dataset, which includes around 8000 pieces of media with four columns:

1. *The number of words in the article*
2. *The title of the article*
3. *The text content in the article*
4. *Label “fake” or “real”*

This CSV dataset is an online public dataset created by the DataFlair team on their website, Data-Flair.Training. Overall, our group thought that this dataset was exactly what we needed to achieve our goals for this project. The labeling of fake and real articles along with the article contents gave us an interesting opportunity to analyze data.

The Libraries

In our code, the notable python libraries we used were:

- *Natural Language Toolkit*
- *Matplotlib*
- *Pandas*
- *Numpy*
- *Contractions*
- *Inflect*
- *Beautiful Soup*
- *Sklearn*
- *Tensorflow*
- *Keras*

Pre-Data Analysis Steps

Before doing any sort of data analysis on the CSV file, it's important to preprocess all the text contained in the dataset. Data preprocessing is an important step in the data mining process where we can essentially 'clean' up the text within a dataset, thus removing irrelevant information. This is important because if there's a lot of redundant information then the data can be unreliable and thus it's difficult to perform any sort of accurate data analysis.

Using the Python libraries mentioned in the previous section, such as Natural Language Toolkit, we could clean up the dataset. First, we removed any non-ASCII characters from the dataset and lowercase all characters so that when doing data analysis, the program doesn't confuse words that are the same but have different capitalizations. Afterwards, we removed any punctuation and stop words (a set of commonly used words such as "the", "is", "and"). Lastly, we stem words (reducing inflection in words to the root word) and lemmatize verbs (similar to stemming, but it brings context to the words) throughout the dataset. After completing these steps, our dataset is now finished with preprocessing!

Next, it's important to check the distribution between fake news and real news in our dataset to make sure there isn't an uneven distribution. This is important because if the distribution is uneven then it would be difficult to analyze the dataset properly. Furthermore, it would be difficult to create a machine learning model to predict fake news vs real news because the

model could become biased if given an uneven dataset. As a result, when looking at the dataset distribution between fake and real news, our group found that there's 3171 real news articles and 3164 fake news articles, meaning there's a good distribution between the two and we can now begin our data analysis.

Sentiment Analysis

First, our group decided to separate the real news and fake news text into different lists to make analyzing easier. Afterwards, we decided to calculate the sentiment score of all the fake news text and real news text using VADER. The results are as follow:

```
sentiment_scores(realNewsText)
sentiment_scores(fakeNewsText)
```

```
Overall sentiment dictionary is :  {'neg': 0.093, 'neu': 0.805, 'pos': 0.102, 'compound': 0.4811}
sentence was rated as  9.3 % Negative
sentence was rated as  80.5 % Neutral
sentence was rated as  10.2 % Positive
Overall sentiment dictionary is :  {'neg': 0.167, 'neu': 0.719, 'pos': 0.114, 'compound': -0.997}
sentence was rated as  16.7 % Negative
sentence was rated as  71.89999999999999 % Neutral
sentence was rated as  11.4 % Positive
```

The results were slightly different, as the fake news articles had more of both positive and negative sentiment reviews. Based on the results below we can see that the text from the real news articles and fake news articles are primarily neutral. However, it's interesting to see that the text from the fake news articles have a greater negative sentiment score and positive sentiment score compared to real news text. Thus, based off this sentiment analysis we can determine that real news articles are less biased than fake news articles such that real news articles have a 10% greater neutrality sentiment in their text.

To get more specific then decided to compare the titles of the articles. We did this by looking at the sentiment scores for each title separately and put all the real news titles and fake news titles, along with their positive scores, negative scores, and neutral scores, into three separate lists (giving us 6 different lists to analyze in total). Afterward, we split the data into three different charts, comparing the titles of fake news and real news articles with titles that had the highest neutral score, positive score, and negative score.

Top 10 most negative fake news titles vs Top 10 most negative real news titles:

NEGATIVE REAL NEWS TITLES

	Titles	NegativeScore
1117	poll: Alarm, anxiety	80.4
1141	2 killed, 8 terror suspects held in France ter...	77.4
976	U.S. drone strike accidentally killed 2 hostages	73.7
3021	Pain, anger and fear: US voters deprived of a ...	70.9
792	Charlie Hebdo attack: Three terrorists killed ...	70.1
474	Terror attack over, 147 dead at Kenya university	68.4
143	Clinton condemns Trump over Chicago violence	64.9
2762	14 dead as Islamic rebels attack in Philippines	64.8
1717	Suspects In Paris Magazine Attack Killed; Mark...	64.1
367	Rivals Slam Trump over Violent Rallies: 'He In...	63.9

NEGATIVE FAKE NEWS TITLES

	Titles	NegativeScore
2460	AMERICAN EVIL	81.5
146	Doomsday Election	79.2
2350	TERROR THREAT WARNING MONDAY | RedFlag News	79.2
655	Ooh Fuck	77.8
560	Cars That Broke Bad	75.9
2478	Democratic decay	73.0
1506	Victim Blaming the Planet	72.6
1770	The World War 3 Conspiracy — Episode 1	70.9
1229	Punishment Is Violent And Counterproductive	70.3
2487	The Trump — Epstein Rape Lawsuit	68.8

Top 10 most positive fake news titles vs Top 10 most positive real news titles:

POSITIVE REAL NEWS TITLES

	Titles	PosScore
524	Sanders' challenge: Winning over Obama supporters	71.7
371	Supreme Court Readies Blockbuster Rulings	71.4
139	Yes, Ted Cruz could win	68.4
538	Sure, We Want An Honest And Trustworthy Presid...	60.0
2661	Donald Trump throws a grand old party	58.8
1612	Presumptive Nominee? Trump Indiana Win Creates...	58.6
389	Admit it: You love Tax Day! (Opinion)	55.7
751	Who won the debate?	55.2
2297	Democratic debate: National security dominates	54.5
1153	Can Libertarian Rand Paul Win A Republican Pri...	53.3

POSITIVE FAKE NEWS TITLES

	Titles	PosScore
889	In Consideration of the Supreme Importance of ...	67.3
46	Kevin MacDonald celebrates Trump's Amazing Vic...	65.2
1908	Best of Luck With the Wall	64.3
2952	UK Interested in Strong Energy Sector, Stable ...	63.2
3009	Top 10 Pet Care Tips	62.5
2089	Re: WOW! What Josh Earnest admitted about Obam...	57.4
336	'Tolerant' Liberals Show Nothing But Contempt,...	57.3
1916	The Dangers Of Romantic Love	57.0
3038	A Hillary Win Will Be Google's Win of Everything	55.9
1504	This Is How Putin Celebrated Trump's Election Win	55.6

Top 10 most neutral fake news titles vs Top 10 most neutral real news titles:

NEUTRAL REAL NEWS TITLES

	Titles	NeuScore
2	‘Britain’s Schindler’ Dies at 106	100.0
3	Fact check: Trump and Clinton at the 'commande...	100.0
4	Iran reportedly makes new push for uranium con...	100.0
7	Trump takes on Cruz, but lightly	100.0
8	How women lead differently	100.0
10	The 1 chart that explains everything you need ...	100.0
12	Hillary Clinton Makes A Bipartisan Appeal on S...	100.0
14	Anti-Trump forces seek last-ditch delegate revolt	100.0
15	Sanders Trounces Clinton in W. Va. -- But Will...	100.0
20	First Take: Wall Street bids goodbye to June hike	100.0

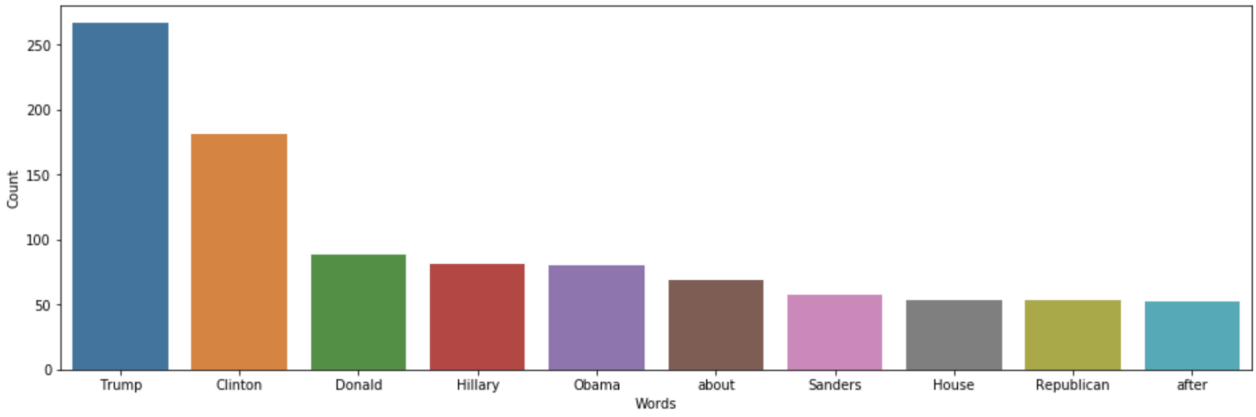
NEUTRAL FAKE NEWS TITLES

	Titles	NeuScore
3	Tehran, USA	100.0
10	‘Inferno’ and the Overpopulation Myth	100.0
17	First Ever Hindu Woman Elected into Congress	100.0
18	Donald Groped Hillary in 2005! Trump and Weine...	100.0
21	Real Disclosure! Secret Alien Base Found In Mo...	100.0
27	Tree Shaped Vertical Farms That Grow 24 Acres ...	100.0
28	New Comment Features have been Added	100.0
29	Ying and Yang (the Gold and Silver Set-Up)	100.0
31	Detroit women’s Halloween decorations depict ‘...	100.0
33	Is google and YouTube in the Hillary's purse?	100.0

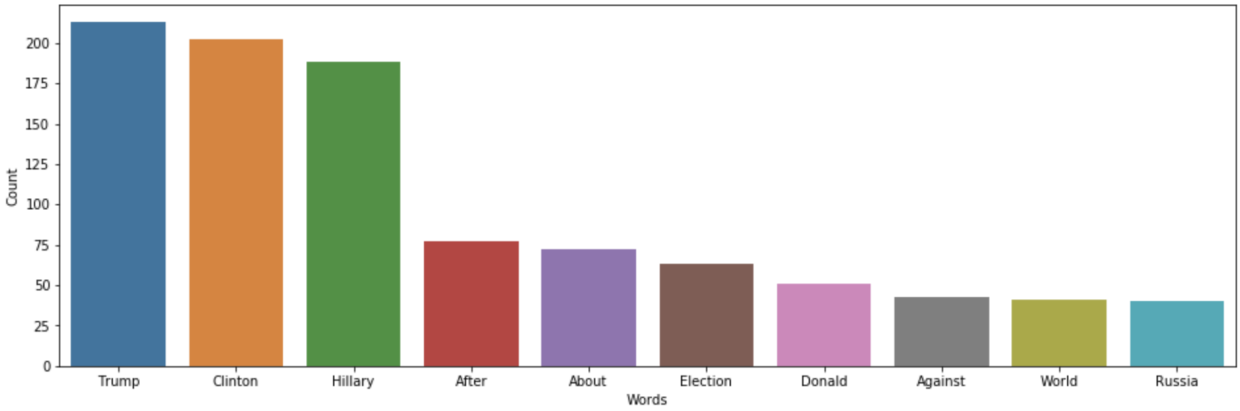
Word Frequency and Data Visualizations

From here, we decided to take a look at the frequency of words within the real and fake news articles.

Here is chart displaying the Top 10 most frequently used words in REAL news articles:



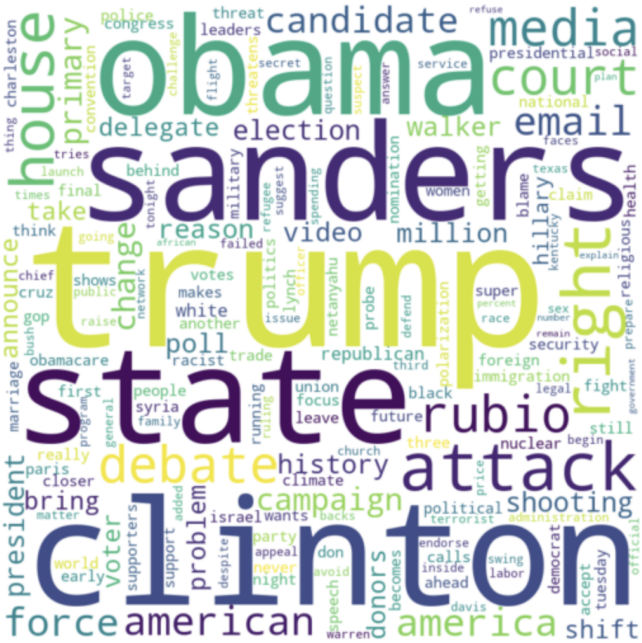
Here is chart displaying the Top 10 most frequently used words in FAKE news articles:



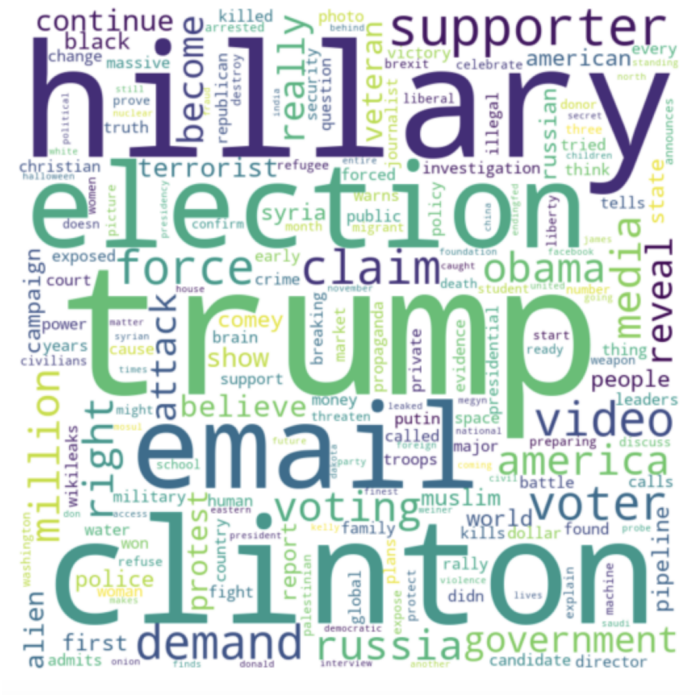
After the first two most commonly used words, the differences become very interesting. It seems that the word “Hillary” is used much more commonly in fake articles than real ones. Also, Barack Obama and Bernie Sanders are mentioned much more frequently in real news articles than fake articles.

To visualize these numbers, we made two WordClouds to represent the frequency of word usage.

WordCloud for real news articles:



WordCloud for fake news articles:



TFIDFVectorizer and Passive Aggressive Classifier

For this section of the project we are going to try running different machine learning algorithms and models to classify fake news and real news articles from the dataset. First, we look at using a TFIDFVectorizer. ‘TF’ stands for Term Frequency and it looks at the number of times a word appears in a text. ‘IDF’ stands for Inverse Document Frequency which measures how significant a word is within a text. Next, we look at using some machine learning algorithm and in this case, we decide to use a ‘Passive Aggressive Classifier’ because this algorithm has a purpose to make updates that correct any miscalculations while making small changes to any weight vector.

Overall, it’s important to understand TFIDFVectorizer and the Passive Aggressive Classifier because we can then use the Sklearn Python module to classify our dataset into real news and fake new articles.

First, using the ‘train_test_split’ function from the Sklearn module, we split up our dataset into training and testing sets, with the training set containing 80% of the overall dataset and the testing set containing the other 20%. Next, we can initialize a TfidfVectorizer to fit and transform

our training and testing sets. Then, we can initialize a Passive Aggressive Classifier to fit the training sets and finally, predict on the test set from TfidfVectorizer.

```
dataText = data_news['text']
dataLabel = data_news['label']
xTrain,xTest,yTrain,yTest = train_test_split(dataText, dataLabel, test_size=0.2, random_state=7)
```

```
# Initalize a Tfidf vector
tfidf_vector = TfidfVectorizer(max_df=0.7)
```

```
# Fit and transform the train and test data
tfidf_train = tfidf_vector.transform(xTrain)
tfidf_test = tfidf_vector.transform(xTest)

# Initalize PassiveAgressiveClassifier
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,yTrain)
yPred = pac.predict(tfidf_test)
```

Overall, we get an accuracy of around 92.7% and using a confusion_matrix we can see that we got a total of 592 true positives, 583 true negatives, 42 false positives, and 49 false negatives.

```
score = accuracy_score(yTest, yPred)
print(score)

# 592 true positive, 583 true negatives, 42 false positives, 49 false negatives
confusion_matrix(yTest, yPred, labels=['FAKE', 'REAL'])
```

```
0.9273875295974744

array([[592,  46],
       [ 46, 583]])
```