Big Data Analysis and Project

Assignment 1 – Part B

To analyze the impact of the figure of speech, sentiment, and social media on the detection and propagation of fake news over social media.

With the help of all the data gathered during the first phase of our project, we were able to understand the major challenges we need to accomplish during this project. For now, we would be focusing on the two major data sets- The tweets data set and the fake and real news data set.

The tweets data set comprises 2 files: train.csv and test.csv. Both the CSV files contain 2 columns in total:

Tweet	Class
The text of the tweet	The respective class to which the tweet belongs. There are 4 classes -: • Irony • Sarcasm • Regular • Figurative (both irony and sarcasm)

The fake and true news data set also comprises two files, true.csv, and fake.csv. Both files have the same features as follows:

Title	Text	Subject	Date
Title of the news article	The main body of the news article.	The main subject of the article can be news, politics, sports, etc.	The date on which the article was published.

As mentioned in the first question of the previous report (*How can we effectively detect and deal with the proliferation of fake news on social media platforms?*), the above data will be converted into word vector formats for the fake and true news data set and the tweets data set will be processed and cleaned (removal of emojis, hashtags, usernames, URLs) so that it is ready for processing. When the data is ready for processing, we can apply a machine learning technique like Naïve-Bayes classifier to detect if the article is fake or not. An initial analysis has also been executed which shows the classification of the data.

Note- For the initial data analysis and classification, I have extracted the "Title" from the news data sets and both the fields from the tweets data set for the pre-processing of the data.

First, loading the dataset was completed using the pandas library and from the CSVs, the data was stored in two separate data frames. Upon heading both the True and the False data sets, the general overview of the data is as follows:

```
#loading true news dataset
df_true = pd.read_csv("D:\\UoA\\Trimester 2\\Big Data Analysis and Project\\True.csv")
print(df_true.shape)
df_true.head()
```

[2] ✓ 0.7s

```
(21417, 4)
```

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

```
#loading fake news dataset
df_fake = pd.read_csv("D:\\UoA\\Trimester 2\\Big Data Analysis and Project\\Fake.csv")
print(df_fake.shape)
df_fake.head()
```

[4] ✓ 0.6s

```
(23481, 4)
```

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

Figure 1 - General overview of the True and False news data sets.

Now upon loading the data sets for the tweets, from two CVSs containing the train and the test data, the overview looks like the below:

```
train_data = pd.read_csv("D:\\UoA\\Trimester 2\\Big Data Analysis and Project\\train.csv")
test_data = pd.read_csv("D:\\UoA\\Trimester 2\\Big Data Analysis and Project\\test.csv")

train_data.head()
```

[18] ✓ 0.1s

	tweets	class
0	Be aware dirty step to get money #staylight ...	figurative
1	#sarcasm for #people who don't understand #diy...	figurative
2	@lminworkJeremy @medsingle #DailyMail readers ...	figurative
3	@wilw Why do I get the feeling you like games?...	figurative
4	-@TeacherArthurG @rweingarten You probably jus...	figurative

```
test_data.head()
```

[17] ✓ 0.0s

	tweets	class
0	no one ever predicted this was going to happen...	figurative
1	@Stooshie its as closely related as Andrews or...	figurative
2	I find it ironic when Vegans say they love foo...	figurative
3	Quick rt that throwing money vine I've not see...	figurative
4	yep, keep adding me to your #devops lists.... ...	figurative

Figure 2- General overview of the Tweets dataset, both test and train.

For getting a brief idea about the summary of the data, we will be using the value_counts() function to determine the total values present in the data and get an idea about what all is to be dealt with:

```
train_data['class'].value_counts()
[29] ✓ 0.0s
...
class
sarcasm       15404
regular       15285
irony         12784
figurative     7873
Name: count, dtype: int64

test_data['class'].value_counts()
[30] ✓ 0.0s
...
class
sarcasm       2054
irony         2029
figurative    1911
regular       1859
Name: count, dtype: int64
```

Figure 3- Brief Summary of what types and counts of values our data holds.

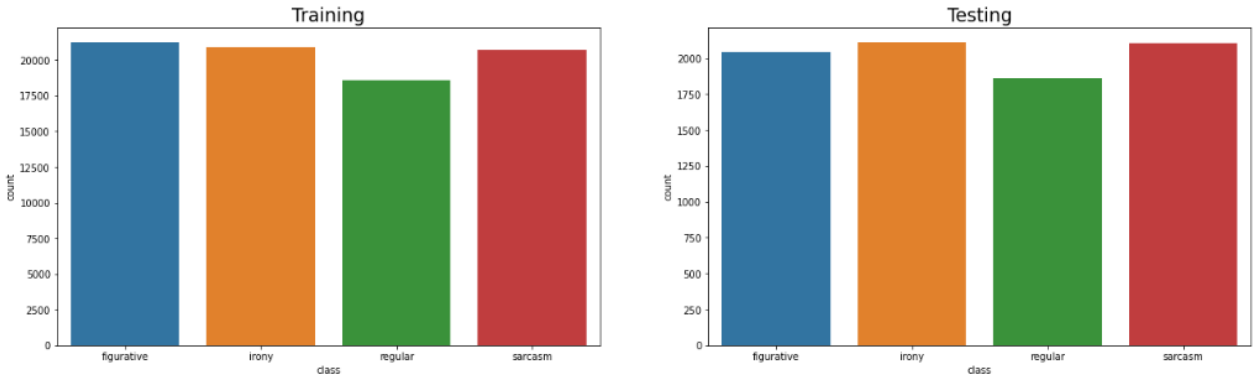


Figure 4- A graph showing both the training and the test data, with the counts of the different figures of speech.

Initial Data Pre-Processing

Fake and True news data set

With the help of the spacy library, which is used for natural language processing in Python, we will convert the original data into word vector form, which will make it easier for us to use ahead and apply models. For ease of understanding and computing, the news articles will be marked as '0' and '1' for fake and true respectively. Later, classification will be performed on this data to predict the news' truth. First, the news titles are converted into vector form, as depicted in the short code snippet below:

```
#transform title column from true news dataset to vectors
df_true['vector'] = df_true['title'].apply(lambda text:nlp(text).vector)
df_true.head()
[18] ✓ 1m 57.9s
```

	title	text	subject	date	label	vector
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	1	[-2.0529835, -0.569027, -3.1433282, 0.55970275...
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	1	[-1.4419098, 2.333012, -2.0302525, 2.1737072, ...
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	1	[0.557869, 2.2574153, 0.1913623, -0.041898414,...
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	1	[-1.4035959, 0.07810003, -0.17810063, 2.512658...
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	1	[0.010546171, 1.6784946, -0.5903046, 0.4184245...

Figure 5- Conversion of the article titles into word vector forms, making it compatible for further processing.

The above code is for the 'true' articles, the same procedure is to be followed for the fake news as well. After both the data sets have been converted into a word vector form, we will concatenate both to make one consolidated data set for both the fake and the true news. The same has been illustrated in the code snippet below:

```
#concatenate fake and true word vector dataset
concatenated_df = pd.concat([df_true[['vector','label']], df_fake[['vector','label']]], ignore_index=True)
print(concatenated_df.shape)
concatenated_df.head()
```

[20] ✓ 0.0s

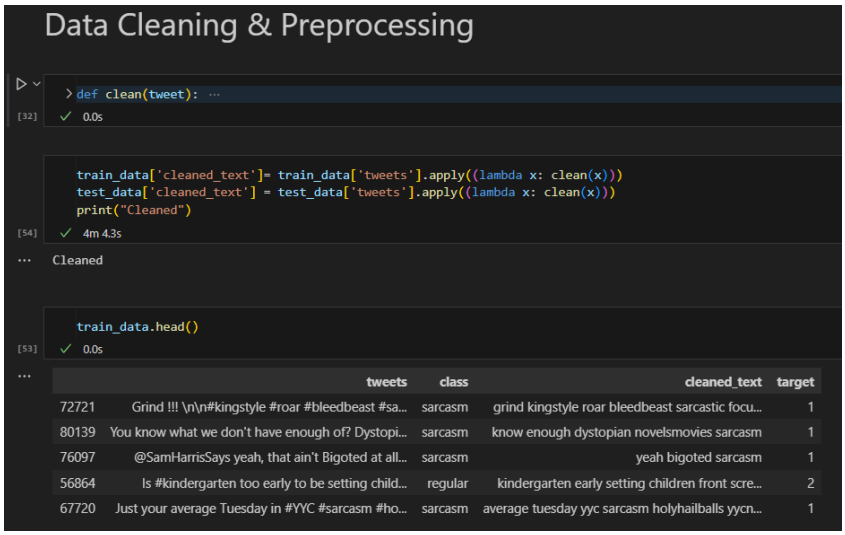
```
(44898, 2)
```

	vector	label
0	[-2.0529835, -0.569027, -3.1433282, 0.55970275...	1
1	[-1.4419098, 2.333012, -2.0302525, 2.1737072, ...	1
2	[0.557869, 2.2574153, 0.1913623, -0.041898414,...	1
3	[-1.4035959, 0.07810003, -0.17810063, 2.512658...	1
4	[0.010546171, 1.6784946, -0.5903046, 0.4184245...	1

Figure 6- Final overview of the fake and true news dataset after completion of initial data cleaning and normalizing.

Tweets data set

For all the data sets used, we will be performing cleaning of the data to bring it into a homogeneous form to make it easier for processing and modeling. Since the tweets can contain multiple types of unwanted characters like emojis, URLs, symbols, username mentions, etc., we need to remove them to make our data more convenient to handle. With the help of the substitute function, we have cleaned our data into a more refined form. The below code shows the tweets before and after they were cleaned.



Data Cleaning & Preprocessing

```
def clean(tweet): ···
```

[32] ✓ 0.0s

```
train_data['cleaned_text']= train_data['tweets'].apply((lambda x: clean(x)))
test_data['cleaned_text'] = test_data['tweets'].apply((lambda x: clean(x)))
print("Cleaned")
```

[54] ✓ 4m 4.3s

```
Cleaned
```

```
train_data.head()
```

[53] ✓ 0.0s

	tweets	class	cleaned_text	target
72721	Grind !!! \n\n#kingstyle #roar #bleedbeast #sa...	sarcasm	grind kingstyle roar bleedbeast sarcastic focu...	1
80139	You know what we don't have enough of? Dystopi...	sarcasm	know enough dystopian novelsmovies sarcasm	1
76097	@SamHarrisSays yeah, that ain't Bigoted at all...	sarcasm	yeah bigoted sarcasm	1
56864	Is #kindergarten too early to be setting child...	regular	kindergarten early setting children front scre...	2
67720	Just your average Tuesday in #YYC #sarcasm #ho...	sarcasm	average tuesday yyc sarcasm holyhailballs yycn...	1

Figure 7- The Clean function, which cleans the data, i.e., removal of emojis, URLs, usernames, and other unwanted data items in the data entries.

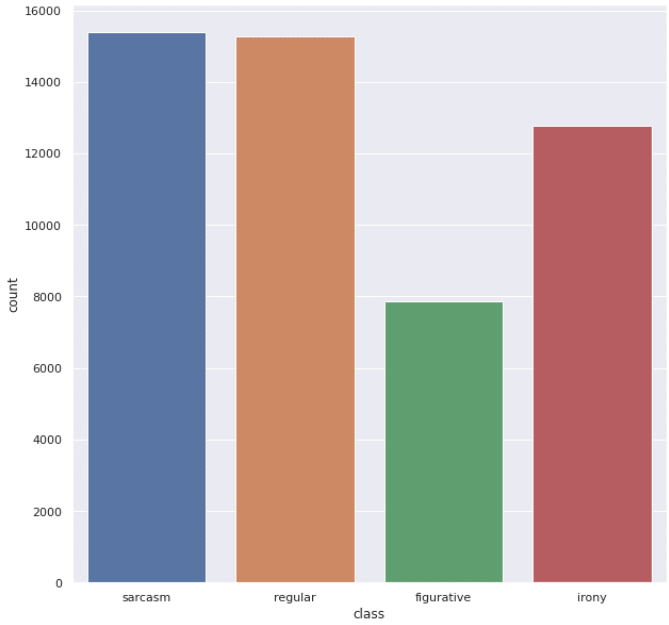


*Figure 8- A graph showing all the types of tweets present in our **training data** set based on the tone, and the count of those tweets. (After the cleaning has been completed)*

Data Classification

For the data classification, we will make use of the concatenated dataset. The concatenated dataset is split into training and testing sets. Two models are trained using pipelines. The first model applies MinMaxScaler to scale the features and utilizes MultinomialNB, a Naive Bayes classifier. The second model also employs MinMaxScaler but uses KNeighborsClassifier, a classifier based on the k-nearest neighbors algorithm. After training, the performance of both models is evaluated on the testing set using classification_report, which provides metrics such as precision, recall, and F1-score for each label. This allows for an assessment of how well the models classify the data into true and fake categories. Below are the classification results for the analysis:

```
print(classification_report(y_test, clf1.predict(X_test)))
[24]  ✓ 0.0s
...                  precision    recall  f1-score   support

           0       0.98      0.98      0.98      7136
           1       0.98      0.97      0.98      6334

    accuracy                           0.98     13470
   macro avg       0.98      0.98      0.98     13470
weighted avg       0.98      0.98      0.98     13470
```

Figure 9- The Classification report for the fake and true news data sets. (Note- 0 stands for the fake news while 1 stand for the true news dataset.)

From all the initial data analysis conducted on the data set, we can address our question from the previous report - *Can machine learning algorithms effectively differentiate between fake news and genuine news articles before they get published to the public?* We can further go ahead and combine the two data sets' results and relatively link the tweets' influence on fake news detection. Therefore, we can say that with the help of such machine learning algorithms,

we can effectively differentiate between fake and genuine news, but a further deep study is needed for assurance, which will be conducted in the next phase of the project.

References:

1- www.kaggle.com. (n.d.). *Fake and real news dataset*. [online] Available at: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

2- Bostonia. (n.d.). *Fake News Influences Real News*. [online] Available at: https://www.bu.edu/bostonia/2017/fake-news-influences-real-news/ [Accessed 12 Jun. 2023].

3- Quantitative Finance & Algo Trading Blog by QuantInsti. (2022). *Natural Language Processing in Python Using spaCy*. [online] Available at: https://blog.quantinsti.com/spacy-python/ [Accessed 4 Jul. 2023].

4- www.kaggle.com. (n.d.). *News Headlines Dataset for Sarcasm Detection*. [online] Available at: https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection.