Objective

The objective of this assignment is to develop a Semantic Classification model. We will be using Word2Vec method to extract the semantic relations from the text and develop a basic understanding of how to train supervised models to categorise text based on its meaning, rather than just syntax. We will explore how this technique is used in situations where understanding textual meaning plays a critical role in making accurate and efficient decisions.

In this assignment, we will develop a Semantic Classification model that uses the Word2Vec method to detect recurring patterns and themes in news articles. Using supervised learning models, the goal is to build a system that classifies news articles as either fake or true.

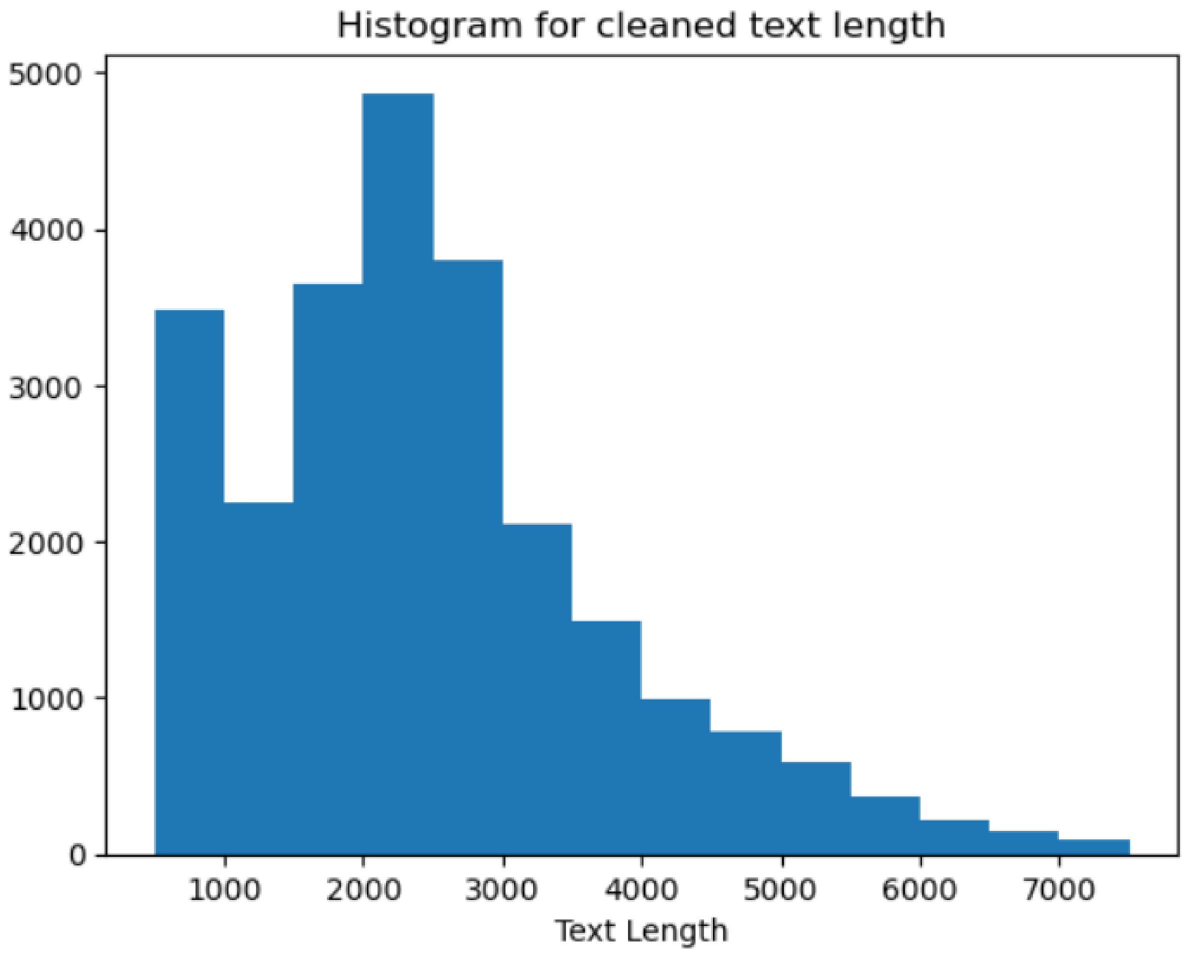
Below is the visual representation/report of the python file and our findings

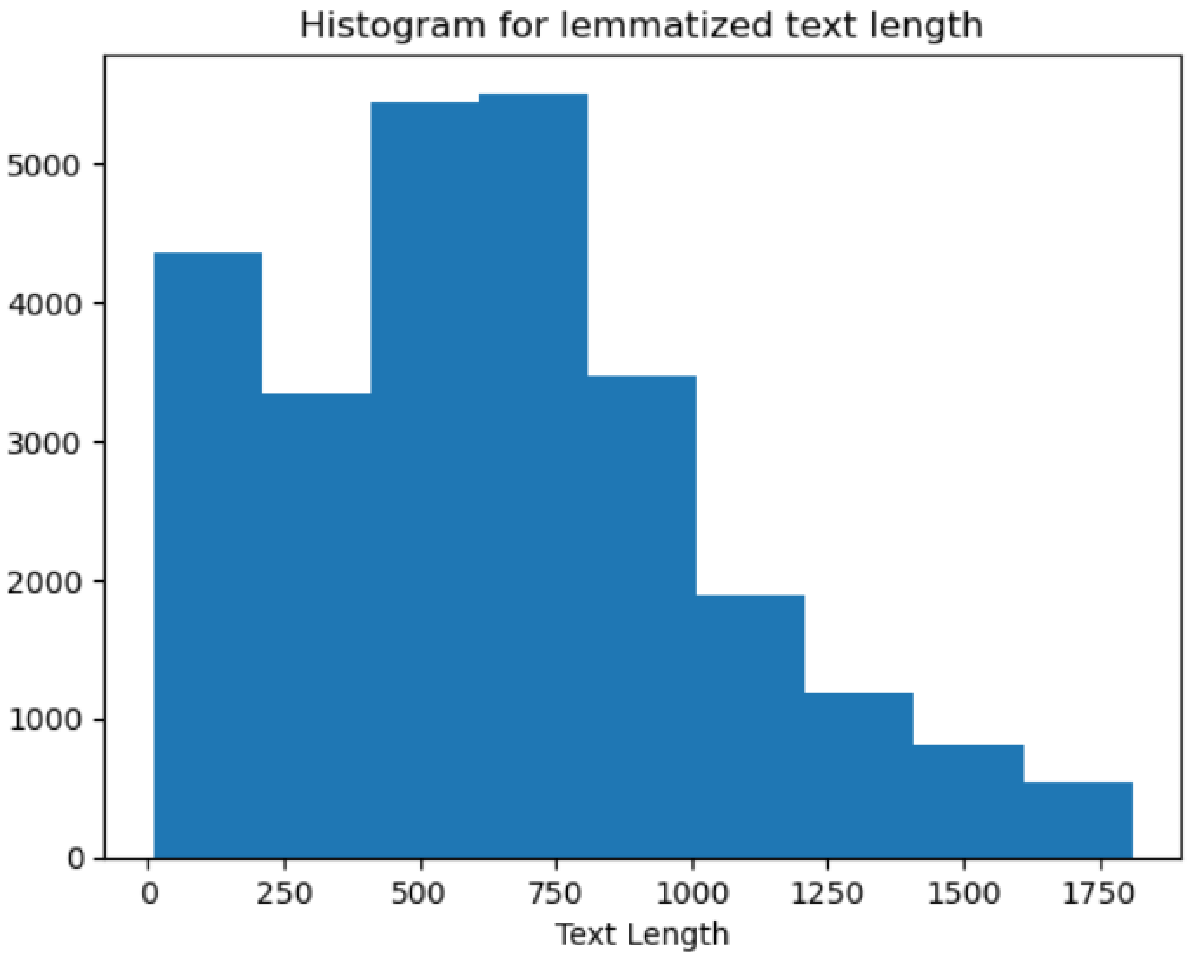
1. Initially we imported the necessary libraries, loaded and prepared the data.

While preparing the data, we added labels as 1 for True news and 0 for Fake news. We also checked for null values and dropped NA values and subsequently cleaned the text to remove unnecessary elements. POS tagging and lemmatization was next followed by Test-Train split.

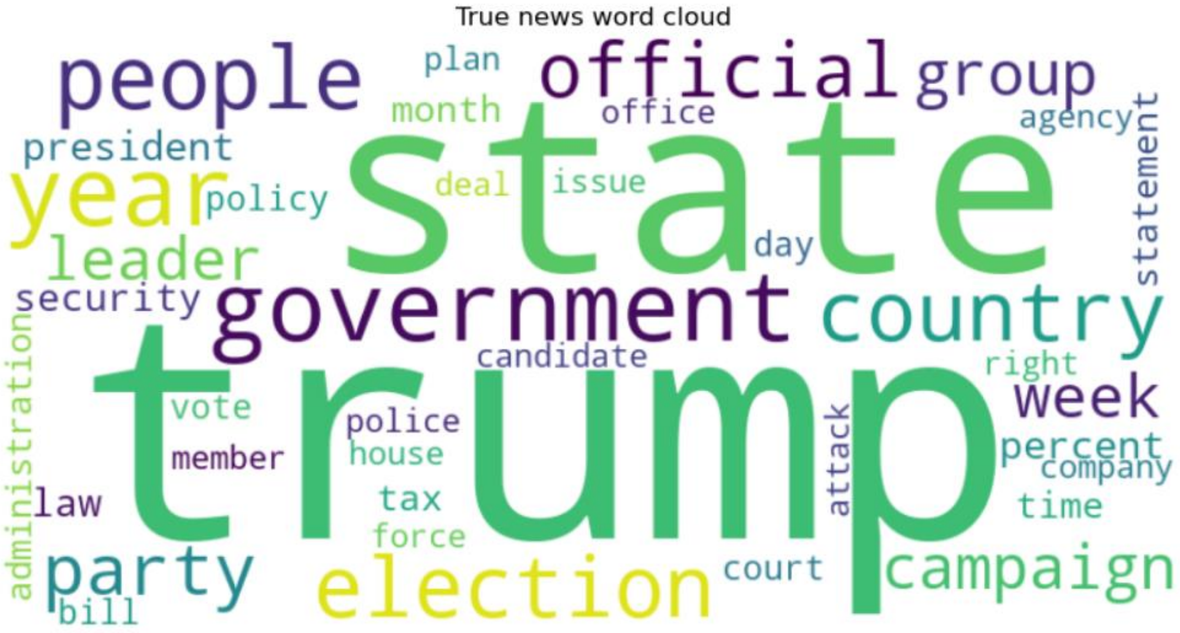
2. EDA

We created Histograms to check the character lengths

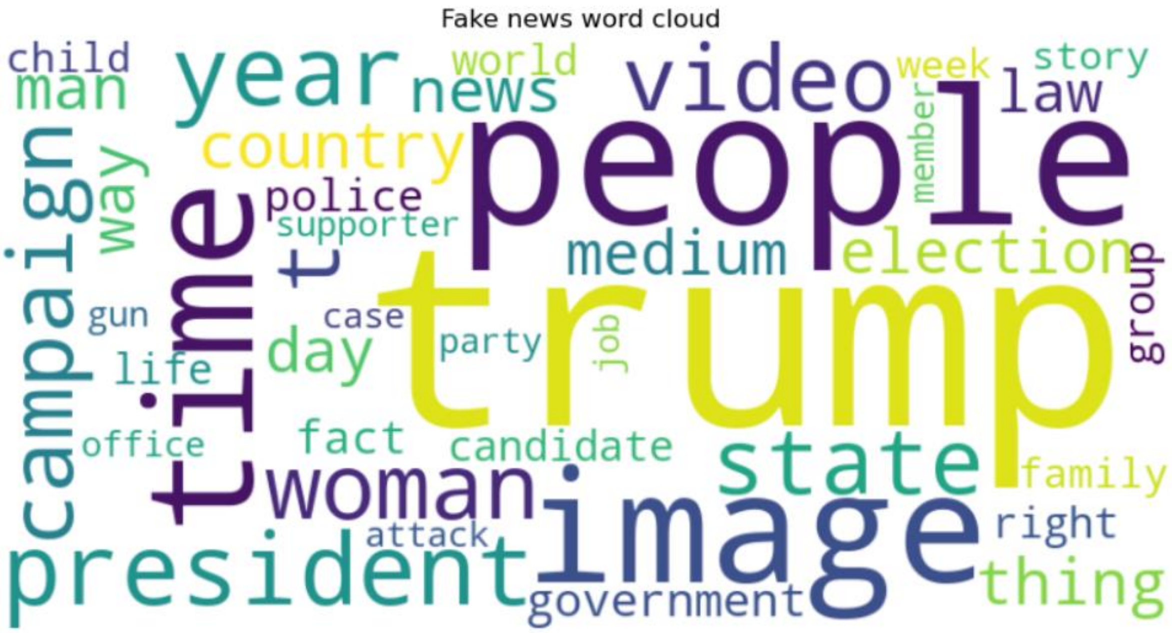




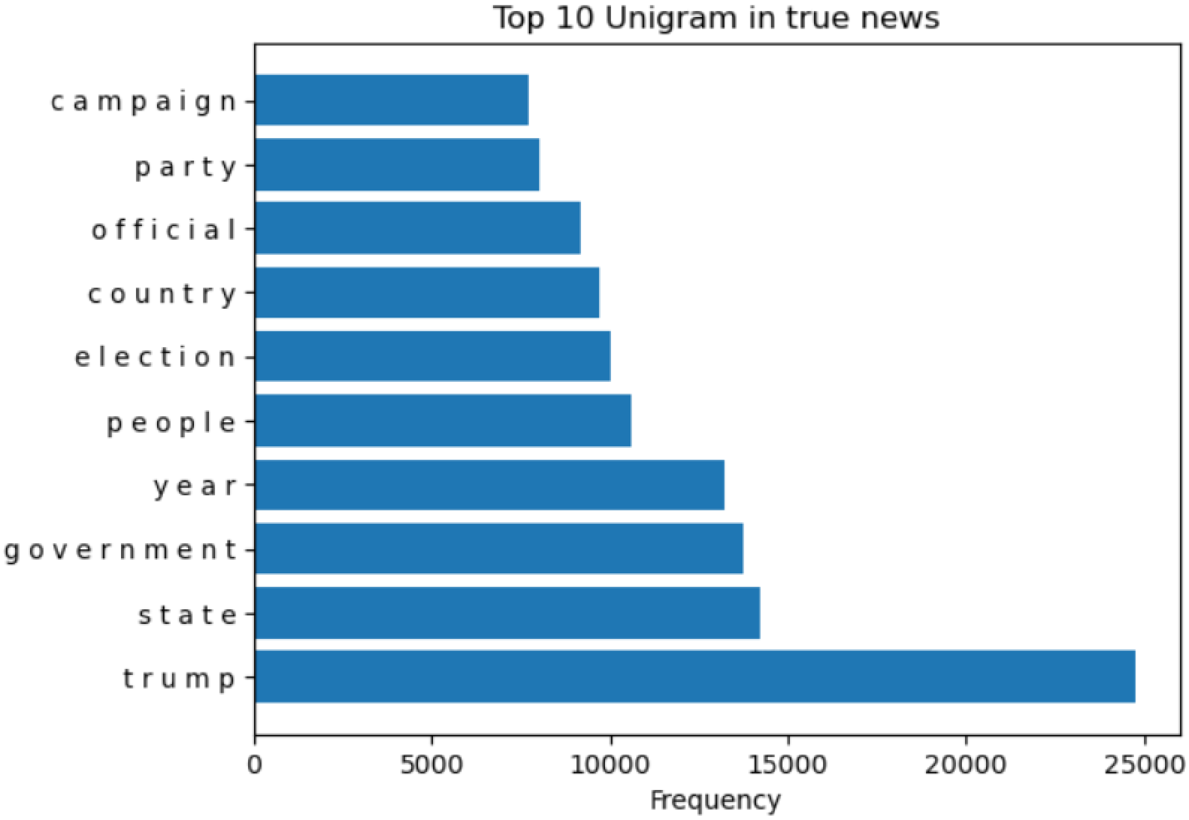
Word cloud for True News:



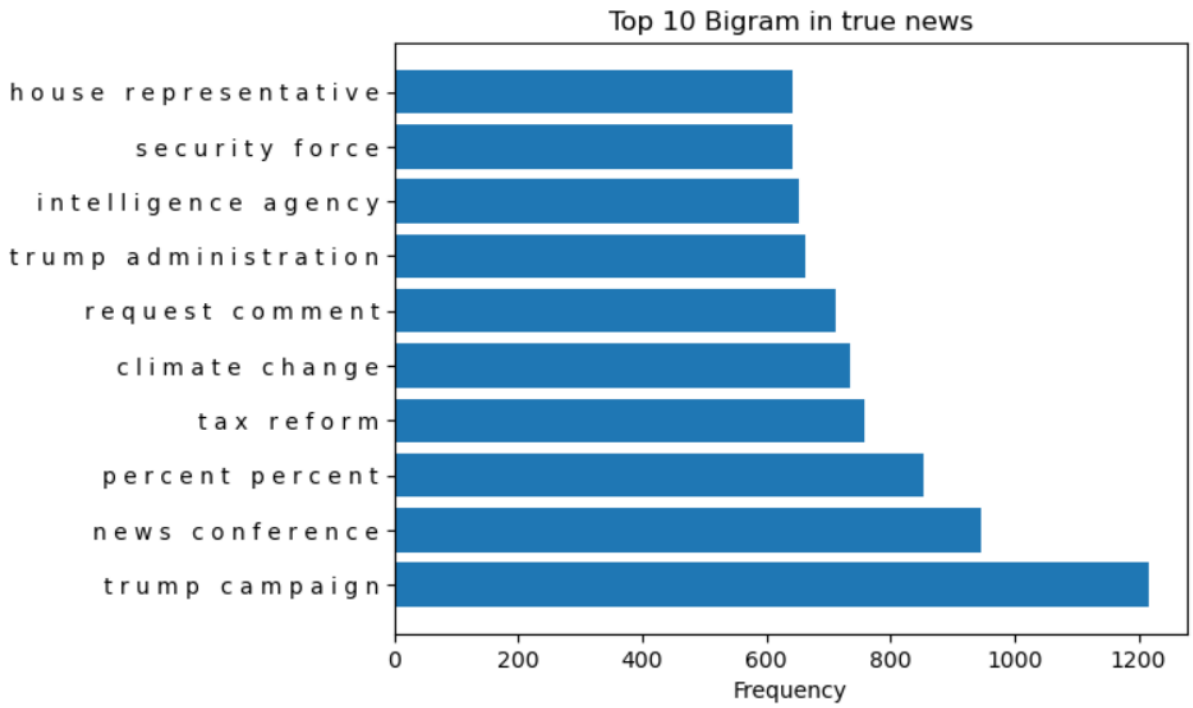
Word cloud for Fake news :



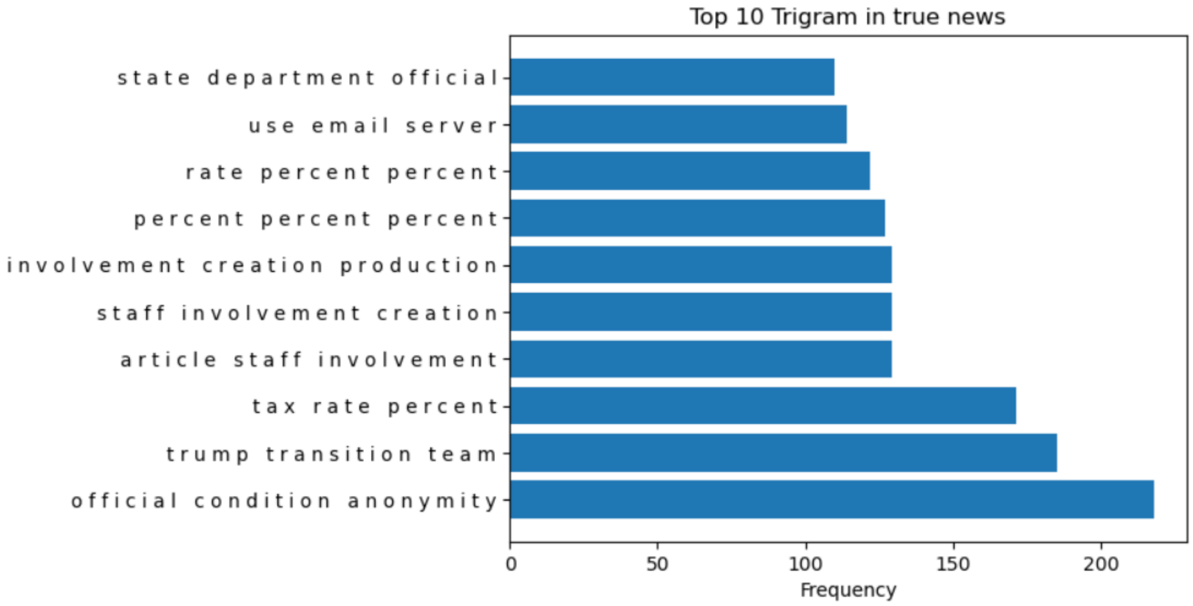
Top ten Unigrams in True news:



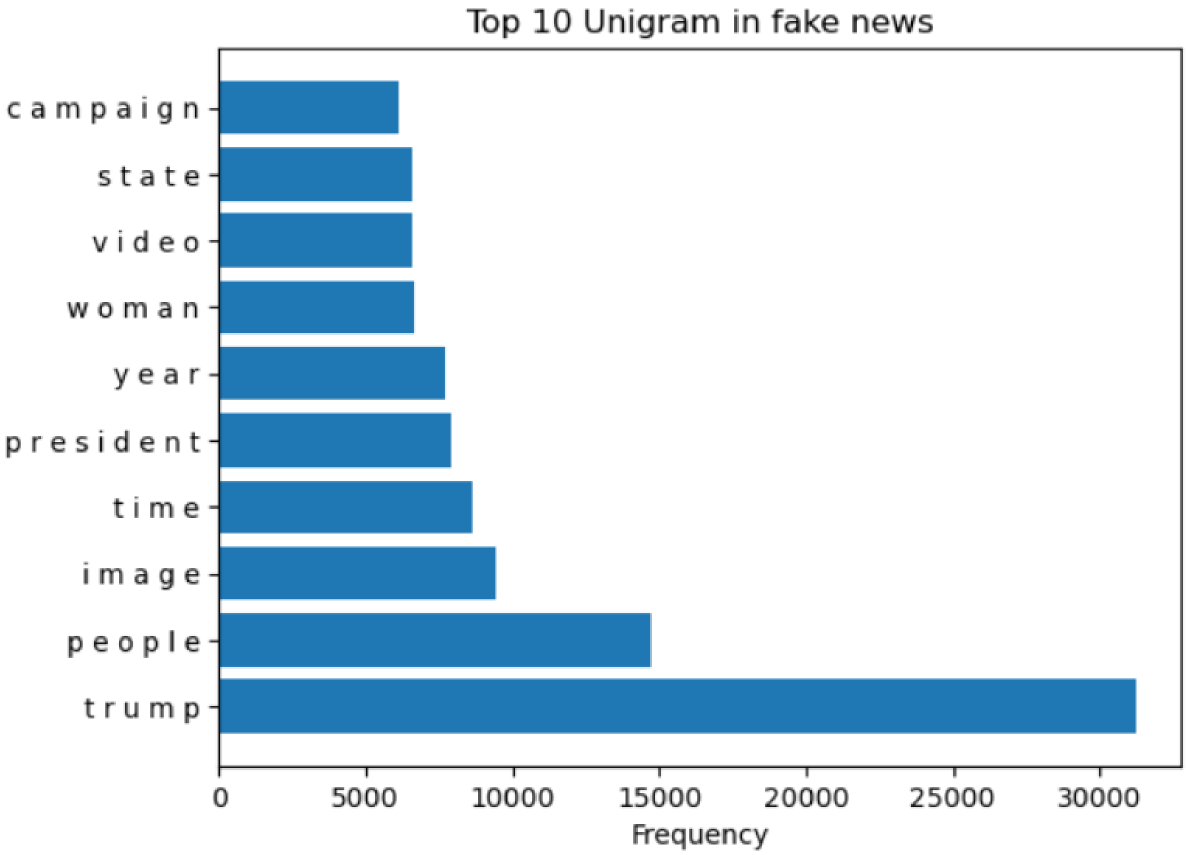
Top ten Bigrams in True news:



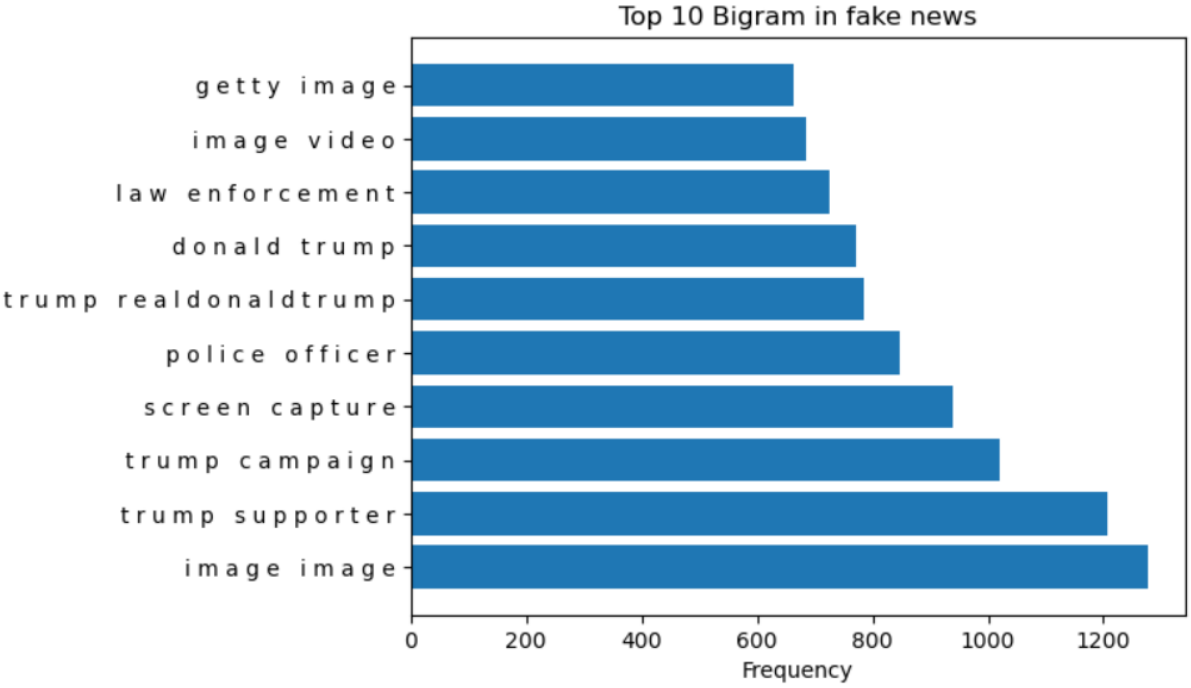
Top ten Trigrams in True news:



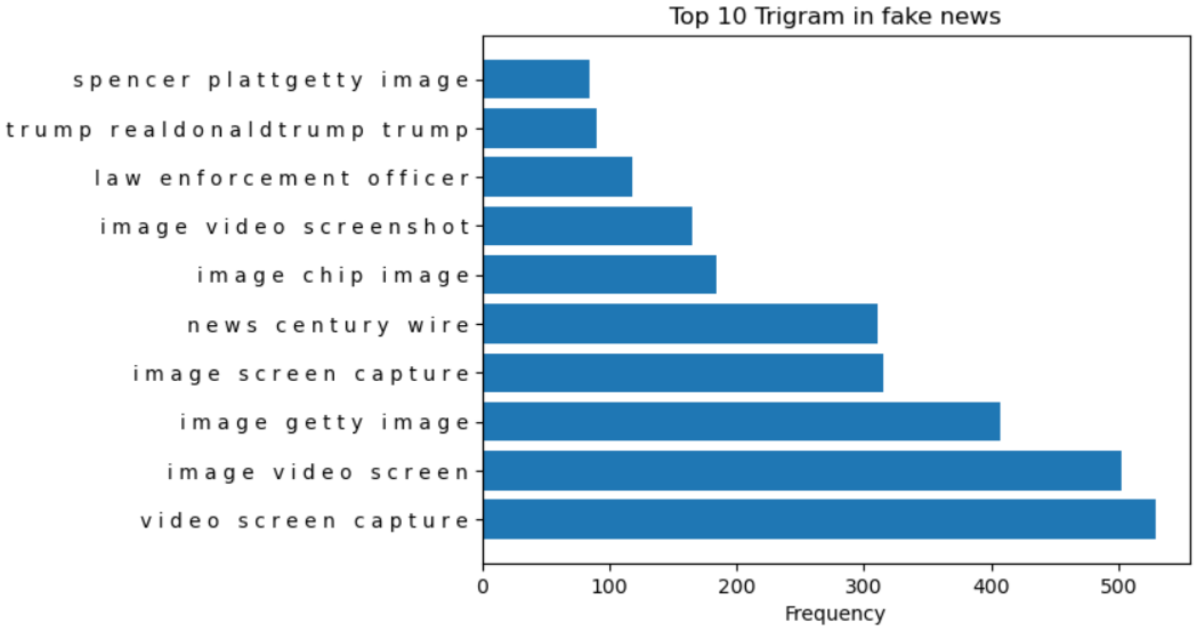
Top ten Unigrams in Fake news:



Top ten Bigrams in Fake news:



Top ten Trigrams in fake news:



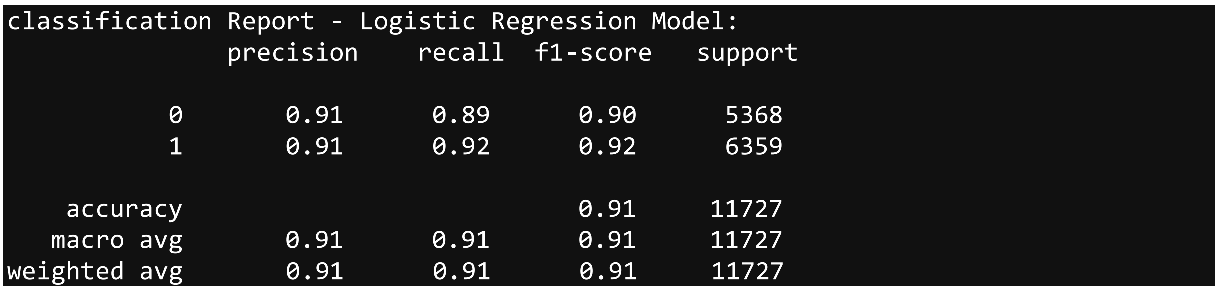
Next we proceeded for Feature extraction

As part of this we use the Word2Vec Vectorizer to create vectors from textual data. Word2Vec model captures the semantic relationship between words. We extract vectors for cleaned news data.

Post this we use various models and evaluated them one by one. Train data:

Logistic Regression

Accuracy_train : 0.9161306826487355
Precision_train : 0.9165670253753155
Recall_train : 0.929972366381344
f1-score_train : 0.923221036432371

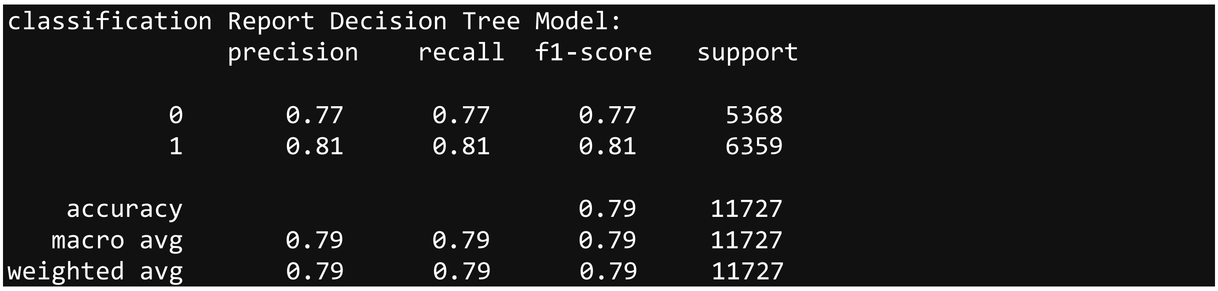


```
classification Report - Logistic Regression Model:
               precision    recall  f1-score   support

           0       0.91      0.89      0.90      5368
           1       0.91      0.92      0.92      6359

    accuracy                           0.91     11727
   macro avg       0.91      0.91      0.91     11727
weighted avg       0.91      0.91      0.91     11727
```

Decision Tree model

'max_depth': 9, 'max_features': 9, 'min_samples_leaf': 1

Accuracy_dtm_train : 0.8572577108609852
Precision_dtm_train : 0.8676757484022872
Recall_dtm_train : 0.869313203477792
f1-score_dtm_train : 0.8684937041276681



```
classification Report Decision Tree Model:
              precision    recall  f1-score   support

           0       0.77      0.77      0.77      5368
           1       0.81      0.81      0.81      6359

    accuracy                           0.79     11727
   macro avg       0.79      0.79      0.79     11727
weighted avg       0.79      0.79      0.79     11727
```

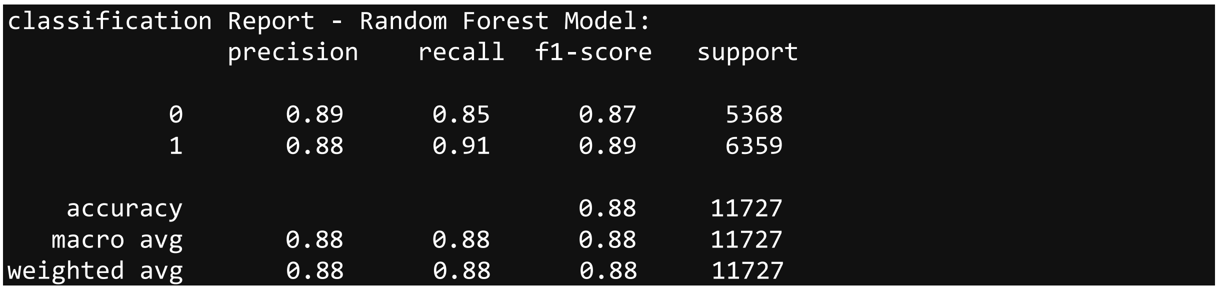
Random Forest Model

'max_depth': 9, 'max_features': 9, 'min_samples_leaf': 1, 'n_estimators': 50

Accuracy_train_rfm : 0.9198216634994884
Precision_train_rfm : 0.9151507191173573
Recall_train_rfm : 0.9392060389566624
f1-score_train_rfm : 0.9270223523150612

Validation data:

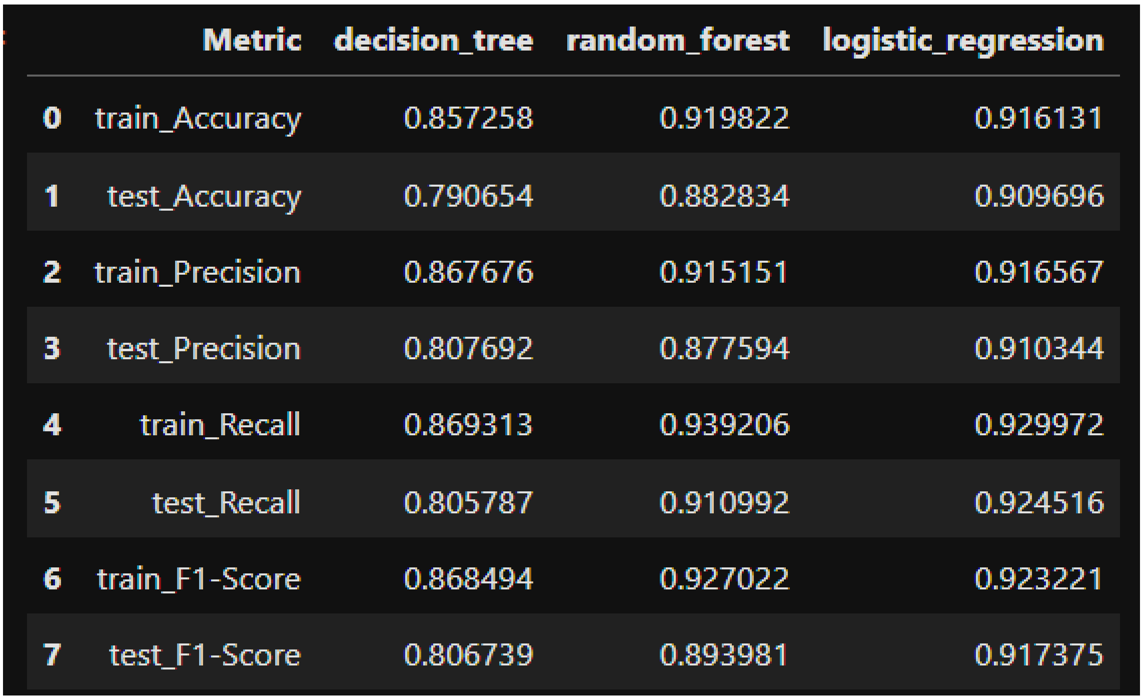
Accuracy test : 0.8828344845228959
Precision test : 0.8775943038933495
Recall test : 0.9109922943859098
f1-score test : 0.8939814814814815



```
classification Report - Random Forest Model:
              precision    recall  f1-score   support

           0       0.89      0.85      0.87      5368
           1       0.88      0.91      0.89      6359

    accuracy                           0.88     11727
   macro avg       0.88      0.88      0.88     11727
weighted avg       0.88      0.88      0.88     11727
```

Conclusion

- As observed from the word cloud of true and fake news does not seem to be much different in top 40 frequent words.
- we have performed Unigram, Bigram and Traigram analysis on true and fake data to see if we can find any pattern
- Unigrams are shows nothing much difference but essentially the word cloud words just represented in a bar chart format.
- in Bigram analysis we can see some difference we see that continuous set of words of "Police officer", "getty image", "image image" etc are the word which appear frequently in the fake news articles.
- Trigram gives us much more clear picture of fake news articles. we can see that words which mostly involves images and video along with bigram words tends to be from fake news articles.
- Of all the 3 models we trained above can clearly see that basic logistic regression model provides us relatively better result in terms of precision, recall and f1-score matrix comparison. See below table for comparison



	Metric	decision_tree	random_forest	logistic_regression
0	train_Accuracy	0.857258	0.919822	0.916131
1	test_Accuracy	0.790654	0.882834	0.909696
2	train_Precision	0.867676	0.915151	0.916567
3	test_Precision	0.807692	0.877594	0.910344
4	train_Recall	0.869313	0.939206	0.929972
5	test_Recall	0.805787	0.910992	0.924516
6	train_F1-Score	0.868494	0.927022	0.923221
7	test_F1-Score	0.806739	0.893981	0.917375

Here even though all the metrics are really important we can see both precision and recall is having almost similar scores. We have analysed F1 score metric as well which take both precision and recall in consideration and create a balanced output of both. This balanced metric prevents a scenario where a model might be highly accurate but fail to identify enough true positives or generate too many false positives. This is particularly important when dealing with imbalanced datasets.