FLIP ROBO

FAKE NEWS DETECTION

Submitted by:

PRATHAMESH VILAS MISTRY`

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to “FlipRobo Technologies” that made this project possible. I would like thank my SME Miss. Khusboo Garg for her guidance in building this project. I would also like to thank Data Trained Institution to making me capable of making proper decisions in the field of damascene and Machine learning. Lastly, I would like to thank my parents to make all of this happen.

INTRODUCTION

- Business Problem Framing

  The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed.

- Conceptual Background of the Domain Problem

  Fake news has existed for a very long time, nearly the same amount of time as news began to circulate widely after the printing press was invented in 1439. Fake news on social media has been occurring for several years; however, there is no agreed upon definition
  of the term "fake news". To better guide the future directions of fake news detection research, appropriate clarifications are necessary.
  Social media has proved to be a powerful source for fake news dissemination. There are some emerging patterns that can be utilized for fake news detection in social media. A review on existing fake news detection methods under various social media scenarios can provide a basic understanding on the state-of-the-art fake news detection methods.
  Fake news detection on social media is still in the early

age of development, and there are still many challenging issues that need further investigations. It is necessary to discuss potential research directions that can
improve fake news detection and mitigation capabilities.

- Review of Literature

The extensive spread of fake news can have a serious negative impact on individuals and society. First, fake news can break the authenticity balance of the news ecosystem. For example, it is evident that the most popular fake news was even more widely spread on Facebook than the most popular authentic mainstream news during the U.S. 2016 president election4

  - . Second, fake news intentionally persuades consumers to accept biased or false beliefs. Fake news is

usually manipulated by propagandists to convey political messages or influence. For example, some report shows that Russia has created fake accounts and social bots to spread false stories5

  - . Third, fake news changes the way people interpret and respond to real news. For example, some fake

news was just created to trigger people's distrust and make them confused, impeding their abilities to differentiate what is true from what is not.

  - . To help mitigate the negative effects caused by fake news–both to benefit the public and the news ecosystem–It’s critical that we develop methods to automatically detect fake news on social media.

- Motivation for the Problem Undertaken

  The problem was undertaken in order to interpret and analyse the news and detect and report what are the main reasons that are trying to be promoted through the fake news.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  The problem is of Binary Classification. We have to classifiy is 0. The news in genuine or 1. The news is fake.

  We have 20.8K news recorss in the dataset.

  There are some missing values in the dataet , but are very less compared to the density of the dataset.

  There are in total 5 features in the dataset.

create word dictionary and word cloud for further and future analytics.

- Data Sources and their formats

The data was provided by the client to “FlipRobo Technologies”. The data is in the form of a comma separated file (CSV). The data i.e. the features and the target are in the single file.

- Data Preprocessing Done
- The Data pre-processing done is as follows:
1. Removing Stop words from the data.
2. Removing punctuations and other special characters from the records
3. Some more granular cleaning for treating hyphen and underscore joined words.
4. Removing the words which are less than 3 letters in length
5. Perform Stemming using PorterStemmer class from sklearn library
6. Further, we remove all the words which do not convey any meaning in the context of the English Language
7. Vectorize the data using tf-idf Vectoriser
8. Note that we have not been provided with the testing data to evaluate our model. Thus first we will split the given records into training and testing data in the ratio of 0.3. Performing best practices we will again further split the training data into training and validation. Find the best model on the validation data, find the hyperparameters and lastly, test the model on the test data we split earlier.

Data Inputs- Logic- Output Relationships

Data is fed in the form of a Pandas data frame to the model. The data is the vectorised meaningful words of the records. For the output we get the predicted label value of the record, that is whether the document is likely to be a same email of not. The output results in a binary value either 1 or 0 respectively.

- ## State the set of assumptions (if any) related to the problem under consideration

There are no such formal assumptions expect the splitting of the data into training, validation and testing.

- ## Hardware and Software Requirements and Tools Used

Hardware Required:

• A computer with a processor i3 or above.

• More than 4 GiB of Ram.

• GPU preferred.

• Around 100 Mib of Storage Space.

Software Required:

• Python 3.6 or above

• Jupyter Notebook.

• Google Collab.

• Excel

Tools/Libraries Used:

1. Computing Tools:

• Numpy

• Pandas

- Scipy
- Sk-learn
- NLTK

2. Visualizing Tools:

- Matplotlib
- Seaborn

3. Saving Tools:

- Joblib

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  We could merge the author and the title feature to create a new feature that will help us to detect the possible frequent words which occur in the fake news more.

  Splitting the dataset into 3 parts , that is training , validation and testing is always helps and it ensures that the model was not biased on the testing data which could be cause by randomized splitting of data

- Testing of Identified Approaches (Algorithms)
  The Algorithms used for testing, training and Validating the models are as follows:
  - Logistic Regression
  - SVM

- ➢ Decision Tree
- ➢ Random Forest
- ➢ AdaBoost
- ➢ Naïve Bayes

- **Run and Evaluate selected models**

  Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

Logistic Regression →

```
In [143]: train_n_pred_model(LogisticRegression(max_iter=1000))

Log loss --> 1.7337316675216656
Recall --> 0.9509419152276295
Precision --> 0.9487079091620987
F1 Score -->  0.9498235985887886
Classification Report -->
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      2552
           1       0.95      0.95      0.95      2548

    accuracy                           0.95      5100
   macro avg       0.95      0.95      0.95      5100
weighted avg       0.95      0.95      0.95      5100
```

Decision Tree →

```
In [147]: train_n_pred_model(DecisionTreeClassifier())

Log loss --> 3.7654443081779085
Recall --> 0.8830455259026687
Precision --> 0.8971291866028708
F1 Score -->  0.8900316455696202
Classification Report -->
              precision    recall  f1-score   support

           0       0.89      0.90      0.89      2552
           1       0.90      0.88      0.89      2548

    accuracy                           0.89      5100
   macro avg       0.89      0.89      0.89      5100
weighted avg       0.89      0.89      0.89      5100
```

Random Forest →

```
9]: train_n_pred_model(RandomForestClassifier())
```

```
Log loss --> 2.045259453823664
Recall --> 0.9368131868131868
Precision --> 0.9442246835443038
F1 Score -->  0.9405043341213554
Classification Report -->
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      2552
           1       0.94      0.94      0.94      2548

    accuracy                           0.94      5100
   macro avg       0.94      0.94      0.94      5100
weighted avg       0.94      0.94      0.94      5100
```

AdaBoost →

```
153]: train_n_pred_model(AdaBoostClassifier())
```

```
Log loss --> 2.153625649621106
Recall --> 0.9536891679748822
Precision --> 0.9239543726235742
F1 Score -->  0.9385863267670915
Classification Report -->
              precision    recall  f1-score   support

           0       0.95      0.92      0.94      2552
           1       0.92      0.95      0.94      2548

    accuracy                           0.94      5100
   macro avg       0.94      0.94      0.94      5100
weighted avg       0.94      0.94      0.94      5100
```

Naiive Bayes →

```
In [57]: train_n_pred_model(GaussianNB())
```

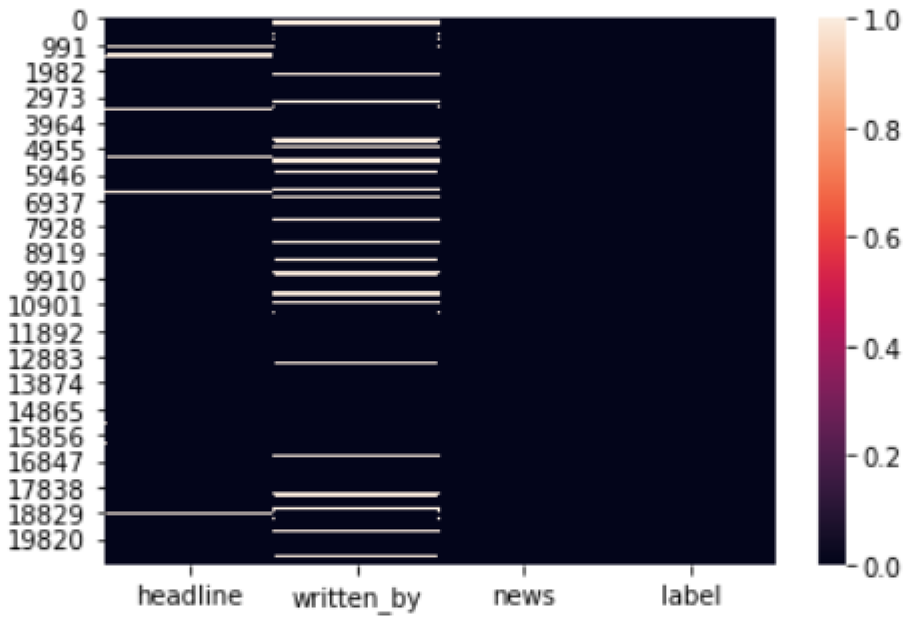
```
Log loss --> 5.356989938910343
Recall --> 0.923469387755102
Precision --> 0.7978975924042048
F1 Score -->  0.8561033290885938
Classification Report -->
              precision    recall  f1-score   support

           0       0.91      0.77      0.83      2552
           1       0.80      0.92      0.86      2548

    accuracy                           0.84      5100
   macro avg       0.85      0.84      0.84      5100
weighted avg       0.85      0.84      0.84      5100
```

- Key Metrics for success in solving problem under consideration

  The key metrics into consideration were precision, f1 score, auc_roc score and log_loss

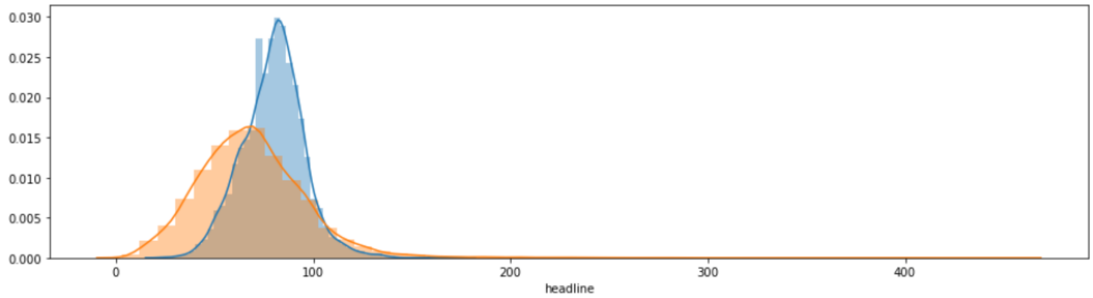
- Visualizations

  Presence of missing data:



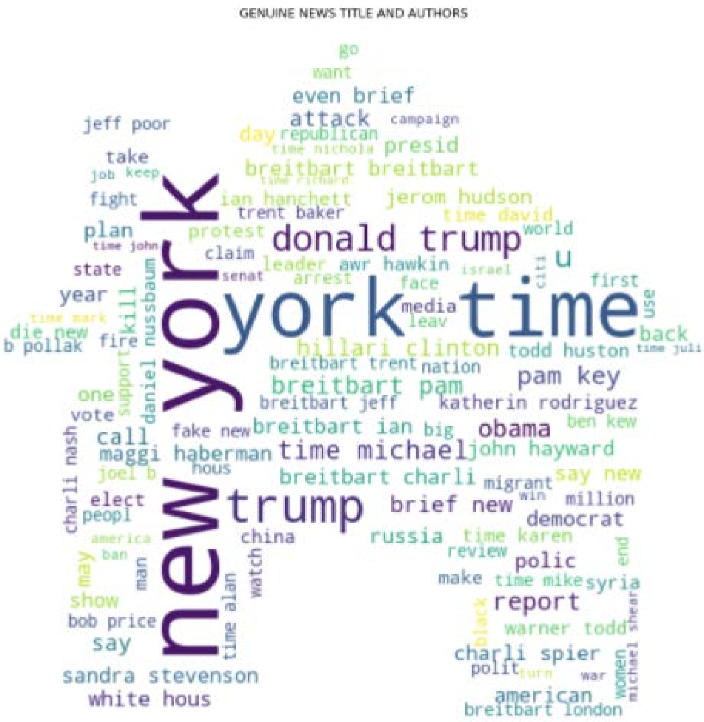
- There are missing values present in the dataset.

Word count in title of the news:

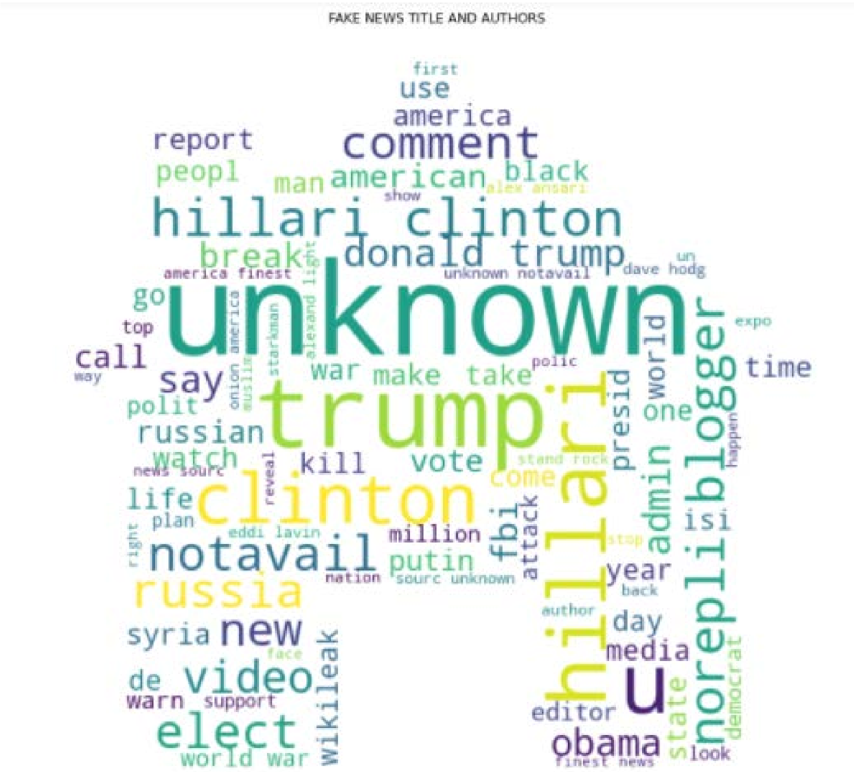


- Most of the headlines have between 50 to 120 words.
- Fake news are less word count in the title relative to genuine news records.

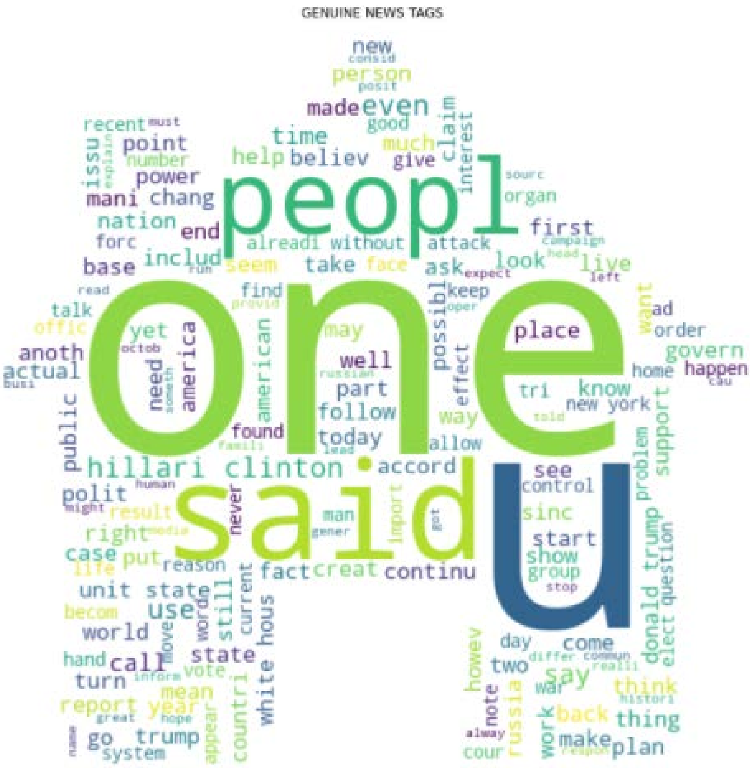
Genuine Title and Authors:



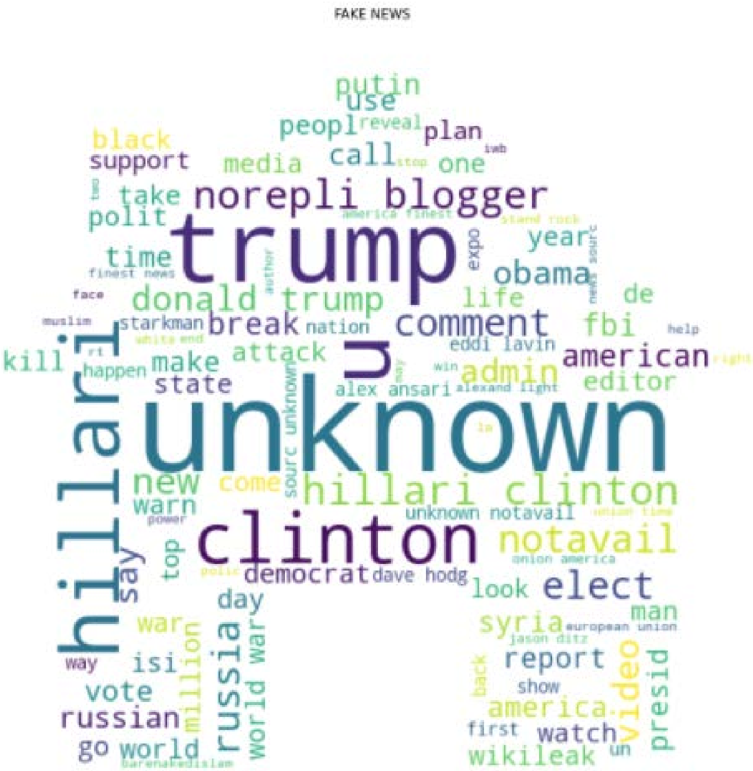
Fake titles and authors:



Genuine news most weighted words tag:



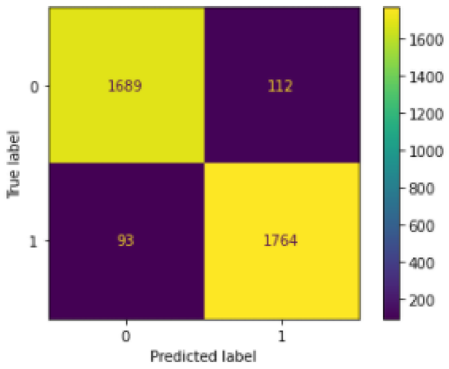
Fake news most weighted words tag:



Confusion Matrix:

```
In [59]: plot_confusion_matrix(model,test_vect,testing_data_ground_truth)
```

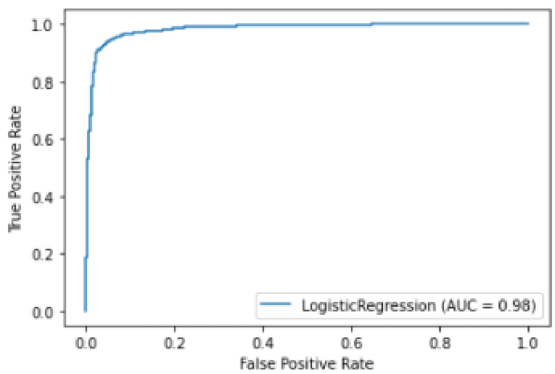
```
Out[59]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x13eb9fd0>
```



ROC Curve:

```
]: plot_roc_curve(model,test_vect,testing_data_ground_truth)
```

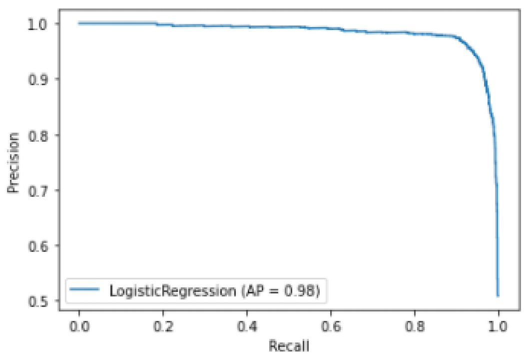
```
]: <sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x510b7340>
```



Sensitivity and Specificity Curve:

```
plot_precision_recall_curve(model,test_vect,testing_data_ground_truth)
```

```
<sklearn.metrics._plot.precision_recall_curve.PrecisionRecallDisplay at 0x1
```



- Interpretation of the Results

  Logistic Regression has given us the best results. We have used count vectorise for vectorising the dataset. We have received over 95% score on sensitivity, specificity as well as roc and log loss and f1 on the unseen test data.

CONCLUSION

- Key Findings and Conclusions of the Study

  NLP gets hard are humans are not used to typing as proper grammar as years progress.
  Sweet spot should be found between whether to pick stemming or lemmatization or both.
  Naïve Bayes algorithms are quicker than rest of the algorithms.
  Decision Tree gives best results when we have huge data.
  Sometimes the simplest model provide us with the best results

- Learning Outcomes of the Study in respect of Data Science
-

  Outcomes of the Study:
  Almost 90 percent of the time is spent of data cleaning and data modelling.
  you do not get a Gaussian distribution in real-word problem.
  NLP becomes difficult due to sloppy use of language by humans
  This also created issue while teaching to machines
  Algorithms like Support Vector Machines and K nearest neighbours may take a long time to converge on a Hugh dataset like this.

Limitations of this work and Scope for Future Work

The following model could be integrated with news filtering middleware which would run once a month on a new/blog website that will find the possible fake news and report them. Other that this we could create a rest api to get the classification results on any other application.