PROJECT NAME:-

FAKE AND REAL NEWS DETECTION

SUBMITTED BY:

Shivam Sharma

INTRODUCTION

Problem Statement :

Fake news has become one of the biggest problems of our age. It has serious impact on our online as well as offline discourse. One can even go as far as saying that, to date, fake news poses a clear and present danger to western democracy and stability of the society.

Fake news's simple meaning is to incorporate information that leads people to the wrong path. Nowadays fake news spreading like water and people share this information without verifying it. This is often done to further or impose certain ideas and is often achieved with political agendas.

For media outlets, the ability to attract viewers to their websites is necessary to generate online advertising revenue. So it is necessary to detect fake news.

So we have to build a model which can predict the News whether it is Fake or Real?

Review of Literature

The topic is about to build a Model which can predict the the News whether it is Fake or Real by analyzing the two datasets that is given to us in which one dataset contain News are Fake and Other is Real so we have to merge these two datsets and make a label column in which fake are be given as 0 and to real is 1.

Motivation for the Problem Undertaken

Fake news has become one of the biggest problems of our age. It has serious impact on our online as well as offline discourse. One can even go as far as saying that, to date, fake news poses a clear and present danger to western democracy and stability of the society. That's y we have to make a model by which we can our model predict which news is real or which is fake.

Data Sources and their format

- Kaggle
- Fake.csv
- True.csv

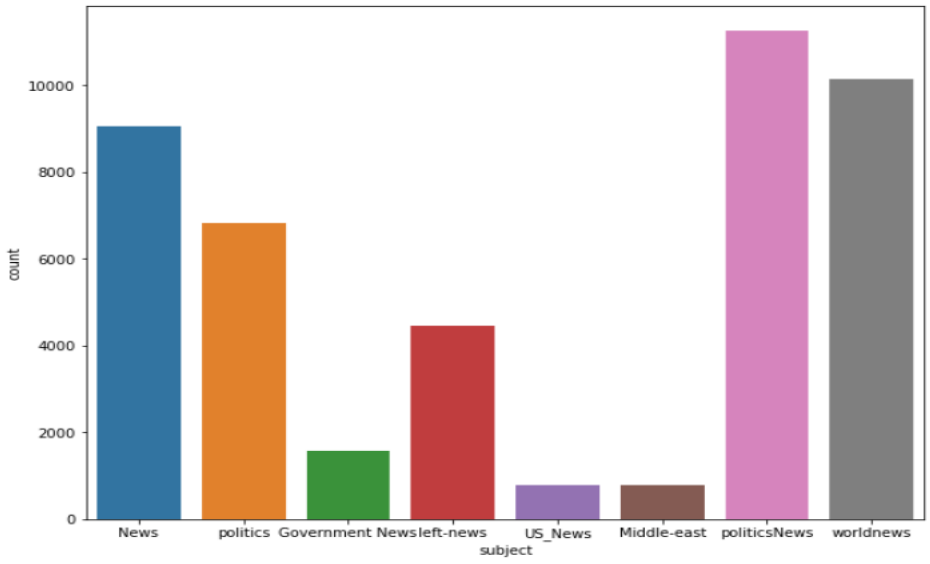
Data Preprocessing and EDA

The steps followed for the cleaning and EDA of the data:-

1) First we analyse the data by simply data.info() and data.describe() methods so that we can check about the datatypes and checked themean , median and deviation.
2) Then we checked for null value if any present there by Isnull().sum() method.
3) Then we add a column for both datsets named as 'label' and gave fake as 0 and real as 1.
4) Then we merge both columns(row wise) by concat methods.

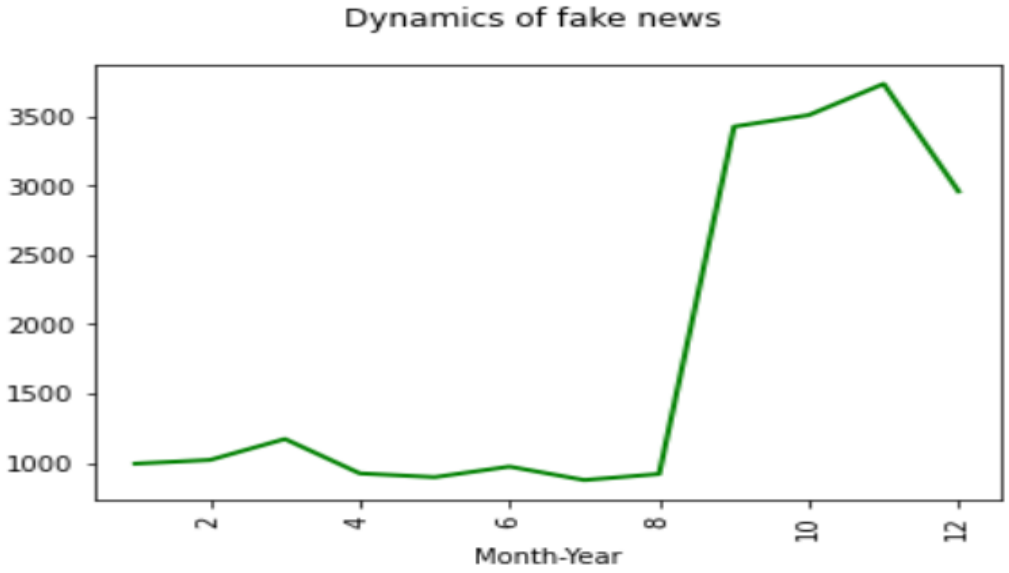
```
# Lets merge both dataframes

data = pd.concat([fake,true],axis=0)
data.sample(frac=1)                  # shuffle
```

5) After make a new dataset we plot a graph by count plot on subject so that we can find which news are of which kind of subject.

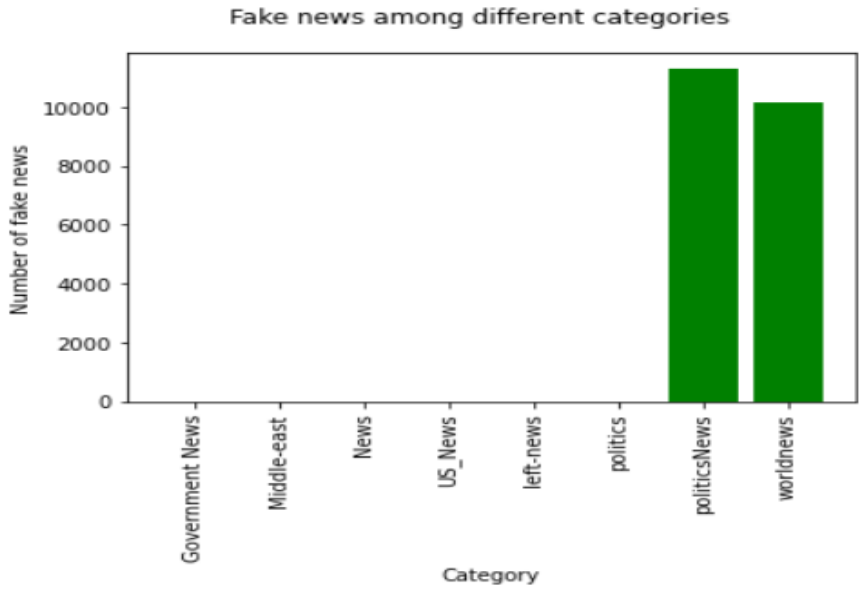


Max news have subjects about Political or from world news

6) We can see that the date format is not the one we need. I will apply the appropriate date format for future purposes.
7) Then make differently column for day month and year so that we can visualize some features by them.
8) After that we drop date column.
9) After this we create a new series to check the dynamics of fake news so that we can understand the No of fake news spiked after july month.



10) We did same with subject to see the dynamics



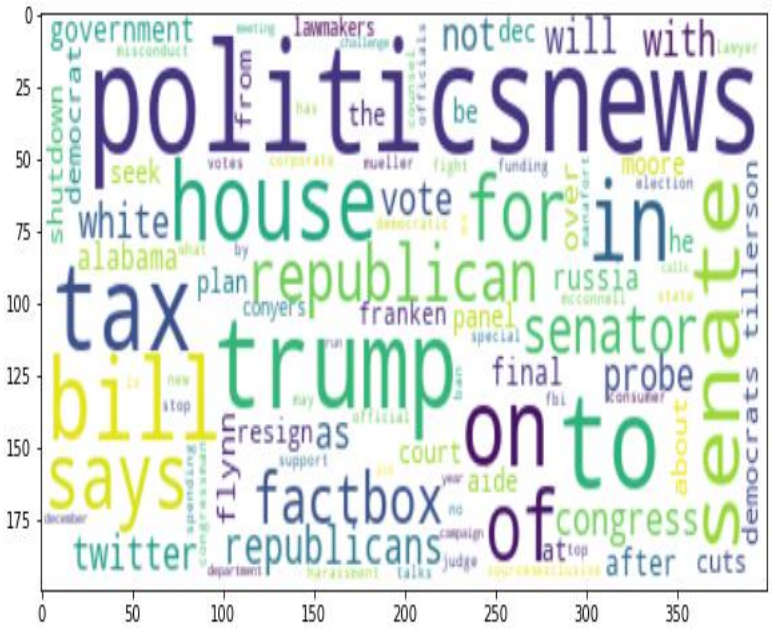
11) Then we copy data into nlp and in which we add subject in the title column as we can see subject affect a lot or we can perform some more visuals to do so.

12) We took some fake news and generate wordcloud so that we can see some loud words .

```
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

df_words.T.sum(axis=1)

Cloud = WordCloud(background_color="white", max_words=100).generate_from_frequencies(df_words.T.sum(axis=1))
```

```
plt.figure(figsize=(12,5))
plt.imshow(Cloud, interpolation='bilinear')
```

```
<matplotlib.image.AxesImage at 0x2592c5e1a90>
```



13) Then after we Convert all messages in lower case, we replaced email addresses with 'email', URLs with 'webaddress', money symbols with 'moneysymb' (£ can by typed with ALT key + 156), 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber', and numbers with 'numbr'…

```
# Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'

nlp['title'] = nlp['title'].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$','phonenumber')
```

```
# Replace email addresses with 'email'

nlp['title'] = nlp['title'].str.replace(r'^.+@[^\.].*\.[a-z]{2,}$','emailaddress')
```

```
# Replace URLs with 'webaddress'

nlp['title'] = nlp['title'].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$','webaddress')
```

```
# Replace money symbols with 'moneysymb'

nlp['title'] = nlp['title'].str.replace(r'£|\$', 'dollers')
```

```
# Replace numbers with 'numbr'
nlp['title'] = nlp['title'].str.replace(r'\d+(\.\d+)?', 'numbr')
```

14) Then after we apply lambda function to remove the Stopwords andpunctuations from the comment and then we used WordNetLemmatizer so that we can make Lemmas and words can bereduced into their meaningful roots..

```
import nltk
from nltk import word_tokenize

nlp['title'] = nlp['title'].apply(lambda x: word_tokenize(str(x)))
```

```
# An important step in every NLP-task is to get the roots of words in order not to distract the model by 'different' words.

from nltk.stem import SnowballStemmer

snowball = SnowballStemmer(language='english')
nlp['title'] = nlp['title'].apply(lambda x: [snowball.stem(y) for y in x])
```

```
nlp['title'] = nlp['title'].apply(lambda x: ' '.join(x))
```

```
# Take the standard english bag of stopwords from nltk.

from nltk.corpus import stopwords
stopwords = stopwords.words('english')
```

15) Then after this we are ready to convert the messages into the Vector form (as machine can not understand object/strings) with thehelp of TF-IDF Vectorizer…

16) Now we are ready for the **model building..**

Algorithm used in this project:-

For building machine learning models there are several models present inside the Sklearn module.

Sklearn provides two types of models i.e. regression and classification. Our dataset's target variable is to predict whether fraud is reported or not. So for this kind of problem we use classification models.

But before the model fitting we have to seprate the predictor and target variable, then we pass this variable to the train_test_split method to create the training set and testing setfor the model training and prediction.

We can build as many models as we want to compare the accuracy given by these models and to select the best model among them.

I have selected 4 models:

1. LINEAR SVC
2. MultinomialNB
3. Logistic Regression
4. Random forest Classifier

BEST Algo. From these And why?

LINEAR SVC is the best algo. From all of these algorithms which is used in this data to predict because it almost predict the 100% accuracy which is best and also I tried with different algorithms which also gives alomost 99.9%

- **Save the model for later predictions**

Confusion matrix for the linear svc

```
: print(classification_report(y_test,y_pred_SVM))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	7750
1	1.00	1.00	1.00	7067
accuracy			1.00	14817
macro avg	1.00	1.00	1.00	14817
weighted avg	1.00	1.00	1.00	14817

 # Now my model is ready to predict

CONCLUSIONS

We have trained & tested 4 models for NLP task (implementing the traditional NLP preprocessing strategies). They all perform very good, however this is most likely due to the high correlation of the target other categorical features (such as 'subject').

If we did not add it to analysis, the result could have been totally different.

Bar plot for all algo. That was used and there predictions.