FLIP ROBO

FAKE NEWS PROJECT

Submitted by:

B.SheebaraniB.E,M.B.A

Batch no.:1844

ACKNOWLEDGMENT

I pay my deep sense of gratitude to Flip Robo Technologies to provide me the opportunity to prepare the project. I would like to express my sincere gratitude to my SME for providing the guidance, comments and suggestion throughout the project.

Successful completion of any project requires help from various web sites. I have taken references from scikit-learn documentation w3schools, towards datascience, medium, stackoverflow, upgrad, Geeks for Geeks,kaggle and youtube.

A heartful thanks also owed to Data trained LMS which gives the basic references for the project.

INTRODUCTION

Now a days social media is extremely powerful and useful for the ability to allow users to discuss and share ideas and debate over issues such as politics, education, health. So, such platforms are also used with negative perspective by certain people for monetary gain and in other cases for creating biased opinions, manipulating mindsets and spreading satire. This phenomenon is known as fake news.

On social media, the reach and effects of information spread occur rapidly and so amplified, that distorted, inaccurate or false information acquires a tremendous potential to cause real-world impacts within minutes for millions of users. The articles shared through online such as online news platforms, blogs, social media feeds and other digital formats do not conform to facts has led to many problems not just limited to politics but covering various other domains such as sports, health and also science. One of such area is financial markets, where a rumour can have disastrous consequences and may bring the market to a halt.

The world wide web contains data in diverse formats such as documents, videos, and audios. News published online in an unstructured format is relatively difficult to detect and classify as it strictly requires human expertise. So, computational techniques such as natural language processing (NLP) can be used to detect anomalies that separate a text article that is deceptive in nature from articles that are based on facts.

The name of the project is ‘Fake News Project’. This project will help millions of people to classify the news or news title is either fake or true. This project contains detailed analysis, findings and conclusion from the given dataset. This project is completely done in Jupyter Notebook. In this project I have used textual features for training the various machine learning models. In this project I have used term frequency-inverse document frequency for converting the text to vectors. And train the data with various model and find the best one out of it.

Analytical Problem Framing

The dataset is in csv file format, which is given by the Flip Robo Technologies. First imported libraries such as matplotlib, pandas and numpy into the Jupyter Notebook and then load the csv file named 'train_news' into it. The loaded data is put into the data frame and give the name as 'df'.
The dataset has 20800 instances and 6 attributes. Then check the column names of the dataset. It consists of column names such as

'Unnamed 0': It is the serial number
'id': Unique id of each news article
'headline': It is the title of the news
'written_by': It represents the author of the news article
'news': It contains the full text of the news article
'label': It tells whether the news is fake (1) or not fake(0)

```
df_merge = pd.concat([df_fake, df_true], axis =0 )
df_merge.head(10)
```

	title	text	subject	date	class
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	0
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0
5	Racist Alabama Cops Brutalize Black Boy While...	The number of cases of cops brutalizing and ki...	News	December 25, 2017	0
6	Fresh Off The Golf Course, Trump Lashes Out A...	Donald Trump spent a good portion of his day a...	News	December 23, 2017	0
7	Trump Said Some INSANELY Racist Stuff Inside ...	In the wake of yet another court decision that...	News	December 23, 2017	0
8	Former CIA Director Slams Trump Over UN Bully...	Many people have raised the alarm regarding th...	News	December 22, 2017	0
9	WATCH: Brand-New Pro-Trump Ad Features So Muc...	Just when you might have thought we d get a br...	News	December 21, 2017	0

In which 'headline', 'written_by' and 'news' are of object datatype and rest of them are of int datatype. After that I have make a copy of the dataset and named it as 'df1_news'. Then combined the 'headline' column and 'news' column to a single column named 'text'. After that drop the 'Unnamed 0', 'headline' and 'news' columns from the dataset because we don't need it further. Then I checked for null values in the dataset. From that it is found that two columns have null values in it. So I have

dropped those nan value columns from the dataset using .dropna() method. Then reset the index so that it could have an ordered index number.

Then again checked the shape of the dataset. At that time it shows 18285 instances and 5 attributes. Then split the dataset to x and y. In which x consist of dataset other than the 'label' column and y consists of the 'label' column. Later I imported nltk and re(regular expressions) libraries. Then I have imported PorterStemmer from nltk.stem.porter library. We use porter stemmer because if we use lemmatization instead of porter stemmer it will take a lot of time to compute. After that imported stopwords from the nltk.corpus library. Then imported TfidfVectorizer from sklearn.feature_extraction.text to convert text to vector form.

Next step is to clean the 'text' data column. For that first created an instance for PorterStemmer as 'pst'. Then create an empty list which named as 'corpus'. Then started a for loop that ranges from 0 to length of the df1_news under that, first substitute the all-other characters which are other than alphabets with space, then converted all the text to lower case. Then split the words and after that stem the words which are not in the stopwords by using the porter stemmer. Then join the text and at last append the text to corpus. Next I have created instance for the TfidfVectorizer with max_features as 5000 and n_gram as 1,3 as parmeters and saved in a variable named 'tfidfv'. The max_features 5000 means 5000 most occurring words in the text

Model/s Development and Evaluation

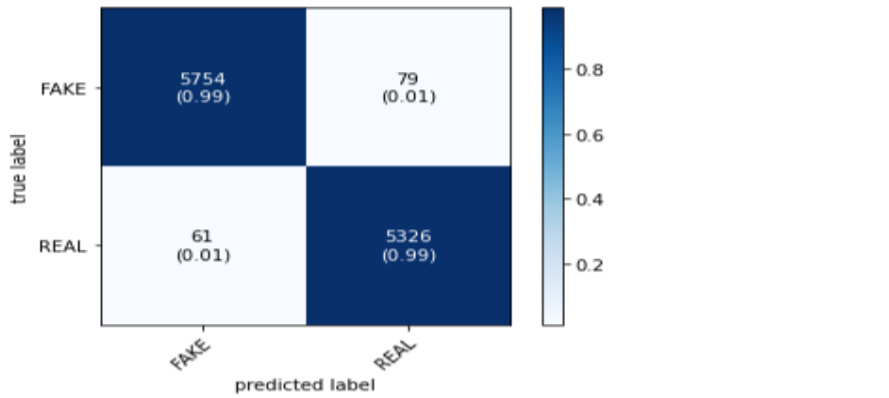
For modelling first, the data is split into ‘x’ and ‘y’. for that the variable ‘corpus’ is then fitted and transformed with TfidfVectorizer and saved as ‘x’. The ‘label’ column of thedataset is saved as ‘y’. Then I have imported train_test_split from sklearn.model_selection library. After that it is again split into x_train, x_test, y_train, y_test.

Thereafter, get the first 20 feature_names of ‘tfidfv’. Then get the feature_names of ‘tfidfv’ of x_train data and put it in a dataframe and save it in a variable named df_vector. Then I have imported classification_report, confusion_matrix, accracy_score from sklearn.metrics library. Also imported plot_confusion_matrix from the mlxtend.plotting library. After that imported the first ML algorithm LogisticRegression.

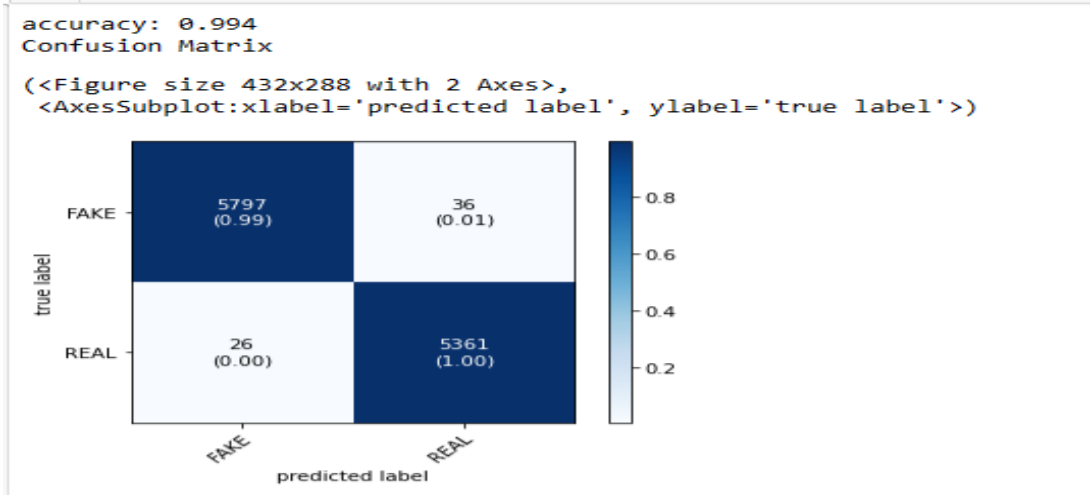
First make an instance for the LogisticRegression as ‘lr’. Then fit that model with xv_train and y_train. After that predict the model with xv_test with .predict() method and saved in a variable named ‘y_predlr’. Then find the accuracy score and store it in ‘scorelr’. Next, find the confusion matrix using y_test and y_predlr and store it in ‘cmlr’. And at last plot the confusion matrix in pictorial manner by using plot_confusion_matrix() method. For this model I got the accuracy of 98.8%.

```
accuracy: 0.988
Confusion Matrix
```

```
: (<Figure size 432x288 with 2 Axes>,
   <AxesSubplot:xlabel='predicted label', ylabel='true label'>)
```



Second, I have imported the SVC from skelearn.svm library. Next, I have created an instance for SVC as ‘svc’. Then fit that model with xv_train and y_train. After that predict the model with xv_test with .predict() method and saved in a variable named ‘y_predsvc’. Then find the accuracy score and store it in ‘scoresvc’. Next, find the confusion matrix using y_test and y_predsvc and store it in ‘cmsvc’. And at last plot the confusion matrix in pictorial manner by using plot_confusion_matrix() method. And for this model I got the accuracy of 99.4%.

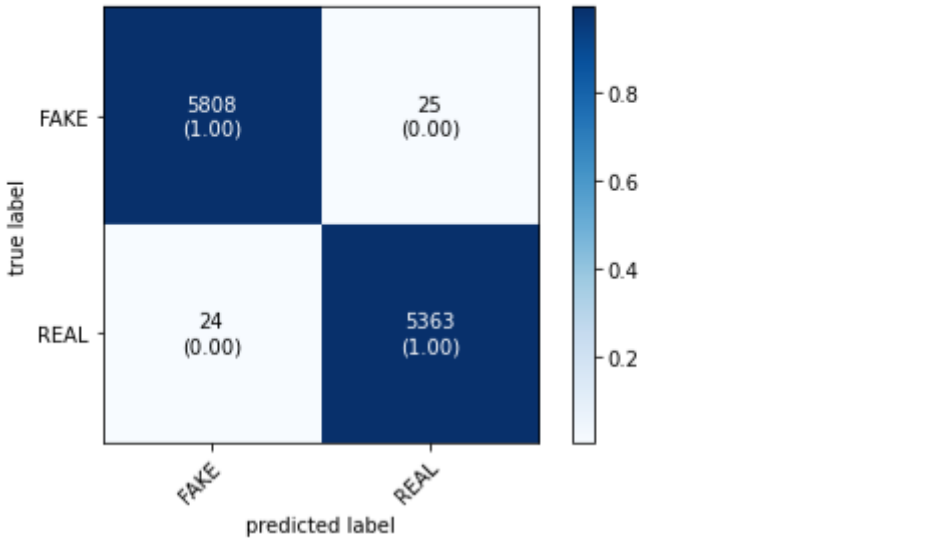


The next model I have tested is MultinonmialNB. This wasimported from sklearn.naive_bayes. After that created an instance for MultinomialNB as ‘mnb’ and fit the ‘mnb’ with x_train and y_train. Then predict this model with x_test and save it in ‘y_predmnb’. Then find the accuracy score of this model with y_test and y_predmnb as the parameters and save it in scoremnb.After that find the confusion matrix and save it in ‘cmmnb’.

Thisconfusion matrix is then plotted into a picture format by using the plot_confusion_matrix() method. The accuracy of the MultinomialNB is 89.5%. This is the model with least accuracy.

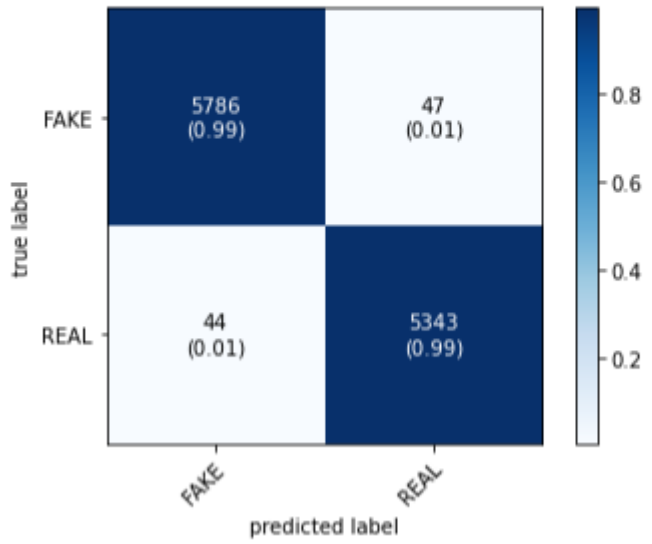
Next algorithm is PassiveAggressiveClassifier which is very good for the text classification. It is imported from the sklearn.linear_model classifier. Then the model is fitted with x_train and y_train. After that the model predicts by giving the x_test data and stored it into 'y_predpac' variable. Then find the accuracy score by giving y_test and y_predpac as parameter and then stored it in 'scorepac'. After that find the confusion matrix using y_test and y_predpac and store in a variable named 'cmpac'. This confusion matrix is then displayed in a picture format using plot_confusion_matrix() method. And this model got 95.1% accuracy.

```
accuracy : 0.996
Confusion Matrix

(<Figure size 432x288 with 2 Axes>,
 <AxesSubplot:xlabel='predicted label', ylabel='true label'>)
```



At last I have imported RandomForestClassifier from sklearn.ensemble library. Then after created an instance for RandomForestClassifier and named as ‘rfc’. After that predict this model with x_test and save it in ‘y_predrfc’. Then find the accuracy score of this model with y_test and y_predrfc as the parameters and save it in scorerfc. After that find the confusion matrix and save it in ‘cmrfc’. This confusion matrix is then plotted into a picture format by using the plot_confusion_matrix()method. The accuracy of the MultinomialNB is 96.0%. This is best model with highest accuracy.

```
accuracy: 0.992
Confusion Matrix

(<Figure size 432x288 with 2 Axes>,
 <AxesSubplot:xlabel='predicted label', ylabel='true label'>)
```



Since the RandomForestClassifier gives the highest accuracy I have checked the parmeters used in that model using .get_params() method. So, in that n_estimators is set as 100. So, to improve the accuracy I have done the hyperparameter tuning by varying the n_estimatrors values such as 100, 300, 500, 800, 1200. From that I got best accuracy of 96.27 from the n_estimator of 800. So, I have again made the instance as RandomForestClassifier with the n_estimator as 800 as the parmeter.

```
last_score = 0
n_estimators = [100, 300, 500, 750, 800, 1200]
for n_est in n_estimators:
    sub_rfc = RandomForestClassifier(n_estimators = n_est)
    sub_rfc.fit(xv_train,y_train)
    y_predr = sub_rfc.predict(xv_test)
    scorer = accuracy_score(y_test,y_predr)
    if scorer>last_score:
        rfc = sub_rfc
    print('n_estimator : {}, Score : {}'.format(n_est,scorer))
```

```
n_estimator : 100, Score : 0.9881461675579323
n_estimator : 300, Score : 0.9909982174688057
n_estimator : 500, Score : 0.9915329768270945
```

Then fit the model again with x_train and y_train. Then predict this model with x_test and save it in 'y_predrfc'. Then find the accuracy score of this model with y_test and y_predrfc as the parameters and save it in scorerfc. After that find the confusion matrix and save it in 'cmrfc'. This confusion matrix is then plotted into a picture format by using the plot_confusion_matrix() method. The accuracy of Hyperparameter tuned RandomForestClassifier is 99.2%. Finally I print the classification report by using the classification_report() method and get the f1-score for 0 (not-fake) has 0.97 and for 1(fake) as 0.96

```
In [44]: 1 print(classification_report(y_test,y_predrfc))
```

	precision	recall	f1-score	support
0	0.96	0.97	0.97	3390
1	0.96	0.95	0.96	2645
accuracy			0.96	6035
macro avg	0.96	0.96	0.96	6035
weighted avg	0.96	0.96	0.96	6035

CONCLUSION

In this research, to classify the news article in online media uses the machine learning models and NLP. The data I have used in this work is collected from FlipRoboTechnologies and contains news headlines and news articles. This will help many online platforms such as news platforms, blogs, social media feeds and other digital formats to avoid the spread of fake news that can led many problems to politics, health, sports, science and financial market.

From the pictorial representation of confusion matrix, we can easily find the True positive, True negative, False positive and False negative counts of each model very easily. Here NLP techniques such as porter stemming, removing stopwords to pre-process the data. Then the words are vectorized using Vectorizer. For this project I haveuse 5 models for training and predict the data. Models and theiraccuracies are listed below.

LogisticRegression: 94.8%

SVC: 99.4%

MultinomialNB: 93.6%

Passive Aggressive Classifier: 99%

RandomForestClassifier: 99.2%

From the above result it is clearly visible that the Random Forest Classifier gives the highest accuracy score. So, the hyperparameter tuning has done in the same model to improve the accuracy. For that, the parameter named n_estimator is given with various values and tested again. So, from that it is concluded that the n_estimator with the value 800 gives the highest accuracy of 99.2%. Then again test the Random Forest Classifier with n_estimator as 800 and got the best accuracy score of 99.2%.