FLIP ROBO

FAKE NEWS PREDICTION

Submitted by:

SAYAN KUMAR BHUNIA

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped me and guided me in completion of the project.

- https://towardsdatascience.com/
- https://anshikaaxena.medium.com/
- https://medium.com/https://medium.com/

INTRODUCTION

➢ Business Problem Framing

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.

➢ Conceptual Background of the Domain Problem

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.

➢ Motivation for the Problem Undertaken

In this project, we are using some machine learning and Natural language processing libraries like NLTK, re (Regular Expression), Scikit Learn.

Analytical Problem Framing

➢ Mathematical/ Analytical Modelling of the Problem

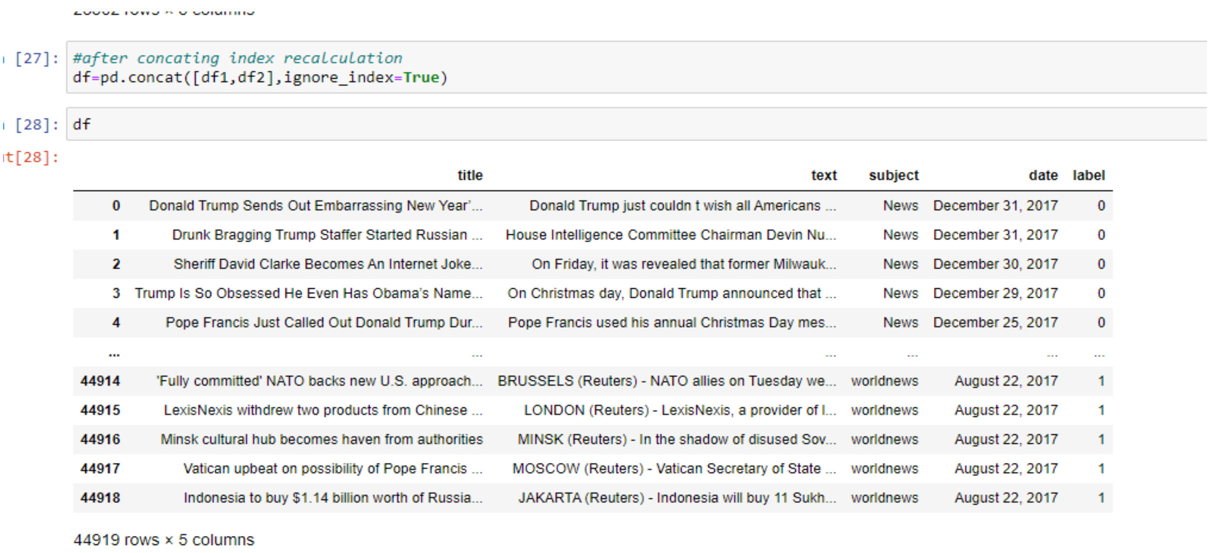
We shall build a supervised Classification model to predict the price of a car based on given features.

Now, when we talk about building a supervised Classification catering to certain use-case, following three things come into our minds:

- **Data** appropriate to the business requirement or use-case we are trying to solve
- A **Classification model** which we think (or, rather assess) to be the best for our solution.
- **Optimize** the chosen model to ensure best performance.

➢ Data Sources and their formats

There are two datasets one for fake news and one for true news. In true news, there is 21417 news, and in fake news, there is 23481 news. I have to insert one label column zero for fake news and one for true news. I have combined both datasets using pandas built-in function concat.



```
[27]: #after concating index recalculation
      df=pd.concat([df1,df2],ignore_index=True)
```

```
[28]: df
```

t[28]:

	title	text	subject	date	label
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
...	...	...	...	...	...
44914	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	1
44915	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	1
44916	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	1
44917	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	1
44918	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	1

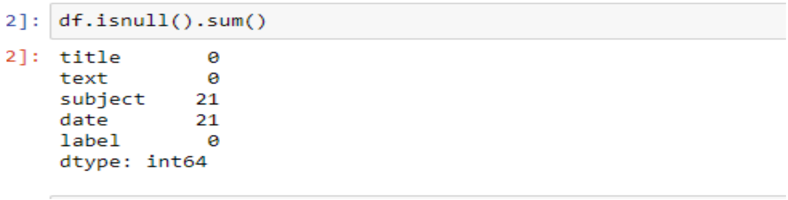
44919 rows × 5 columns

➢ Data Pre-processing

Following steps have been performed on the data.

- **checking missing values**-
  - If there is any missing value present in your data set then for a better and correct accuracy you have to impute it.
  - If missing data present in object type column, then you have to take most frequent value for your missing data.
  - If missing data present in int or float type column then use mean/median for missing value.

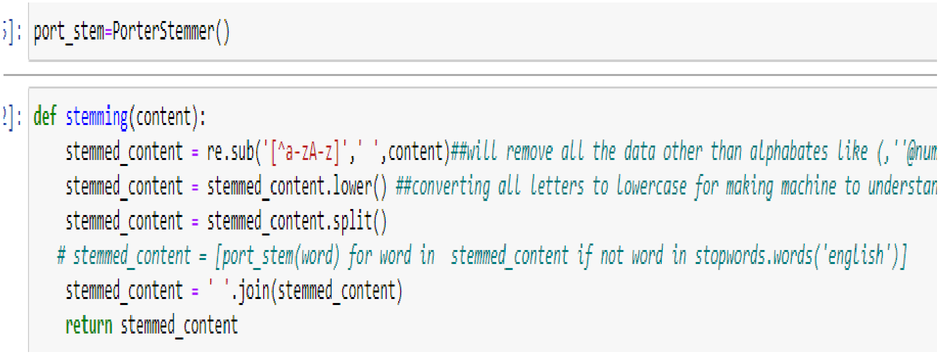
  In the following case no missing value present:



```
2]: df.isnull().sum()

2]: title        0
    text         0
    subject     21
    date        21
    label        0
    dtype: int64
```

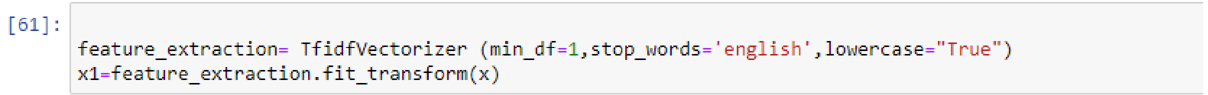
- **Stemming**- Stemming is a natural language processing technique that lowers inflection in words to their root forms, hence aiding in the preprocessing of text, words, and documents for text normalization.
  PorterStemmer() is a module in NLTK that implements the Porter Stemming technique.



```
]: port_stem=PorterStemmer()

]: def stemming(content):
       stemmed_content = re.sub('[^a-zA-z]',' ',content)##will remove all the data other than alphabates like (,''@num
       stemmed_content = stemmed_content.lower() ##converting all letters to lowercase for making machine to understan
       stemmed_content = stemmed_content.split()
     # stemmed_content = [port_stem(word) for word in  stemmed_content if not word in stopwords.words('english')]
       stemmed_content = ' '.join(stemmed_content)
       return stemmed_content
```

- **TfidfVectorizer-**. Tf-idf stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (data-set).



```
[61]:
feature_extraction= TfidfVectorizer (min_df=1,stop_words='english',lowercase="True")
x1=feature_extraction.fit_transform(x)
```

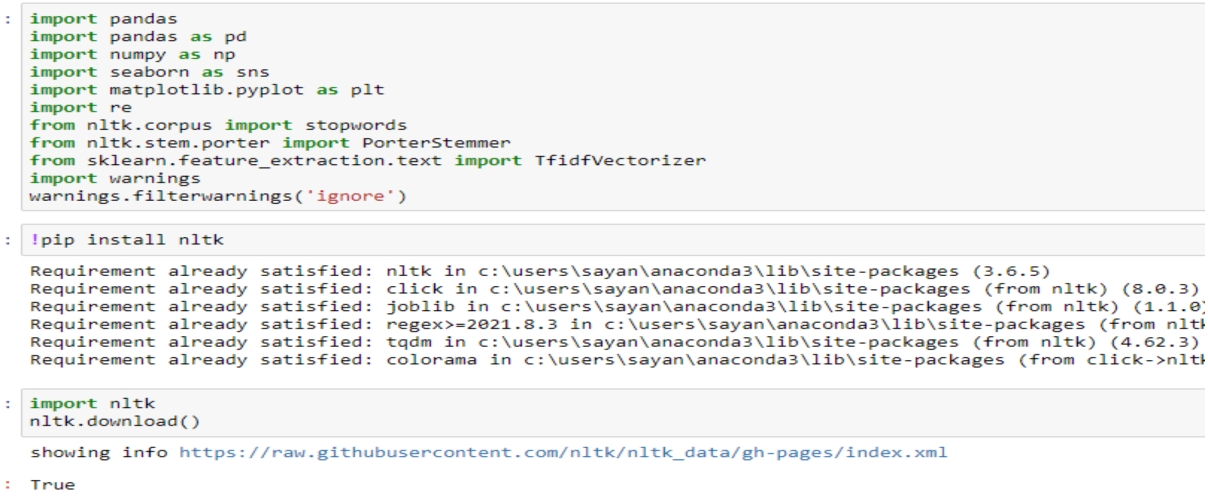
- **Feature scaling-** Feature Scaling ensures that all features will get equal importance in supervised classifier models. Standard scaler was used to scale all features in the data.

- **Reducing dimension of the data-** Sklearn's pca can be used to apply principal component analysis on the data. This helped in finding the vectors of maximal variance in the data.
  In this case iam not using it.

- **Outliers detection-** In simple words, an outlier is an observation that diverges from an overall pattern on a sample.

  In the following case no need to detect outliers.

➢ Software Requirements and library Used



```
: import pandas
  import pandas as pd
  import numpy as np
  import seaborn as sns
  import matplotlib.pyplot as plt
  import re
  from nltk.corpus import stopwords
  from nltk.stem.porter import PorterStemmer
  from sklearn.feature_extraction.text import TfidfVectorizer
  import warnings
  warnings.filterwarnings('ignore')

: !pip install nltk

  Requirement already satisfied: nltk in c:\users\sayan\anaconda3\lib\site-packages (3.6.5)
  Requirement already satisfied: click in c:\users\sayan\anaconda3\lib\site-packages (from nltk) (8.0.3)
  Requirement already satisfied: joblib in c:\users\sayan\anaconda3\lib\site-packages (from nltk) (1.1.0
  Requirement already satisfied: regex>=2021.8.3 in c:\users\sayan\anaconda3\lib\site-packages (from nlt
  Requirement already satisfied: tqdm in c:\users\sayan\anaconda3\lib\site-packages (from nltk) (4.62.3)
  Requirement already satisfied: colorama in c:\users\sayan\anaconda3\lib\site-packages (from click->nlt

: import nltk
  nltk.download()

  showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml

: True
```

- **NumPy**

  NumPy is a popular Python library for multi-dimensional array and matrix processing because it can be used to perform a great variety of mathematical operations. Its capability to handle linear algebra, Fourier transform, and more, makes NumPy ideal for machine learning and artificial intelligence (AI) projects, allowing users to manipulate the matrix to easily improve machine learning performance. NumPy is faster and easier to use than most other Python libraries.

- **Scikit-learn**

  Scikit-learn is a very popular machine learning library that is built on NumPy and SciPy. It supports most of the classic supervised and unsupervised learning algorithms, and it can also be used for data mining, modelling, and analysis.

- **Seaborn**

  Seaborn is another open-source Python library, one that is based on Matplotlib (which focuses on plotting and data visualization) but features Pandas' data structures. Seaborn is often used in ML projects because it can generate plots of learning data. Of all the Python libraries, it produces the most aesthetically pleasing graphs and plots, making it an effective choice if you'll also use it for marketing and data analysis.

- **Pandas**

  Pandas is another Python library that is built on top of NumPy, responsible for preparing high-level data sets for machine learning and training. It relies on two types of data structures, one-dimensional (series) and two-dimensional (Data Frame). This allows Pandas to be applicable in a variety of industries including finance, engineering, and statistics. Unlike the slow-moving animals themselves, the Pandas library is quick, compliant, and flexible.

➢ Class imbalance problem

The first challenge we hit upon exploring the data, is class imbalanced problem. Imbalance data will lead to a bad accuracy of a model. To achieve better accuracy, we'll balance the data by using Smote Over Sampling or under sampling Method .But in this project data is balanced so I am not using it.

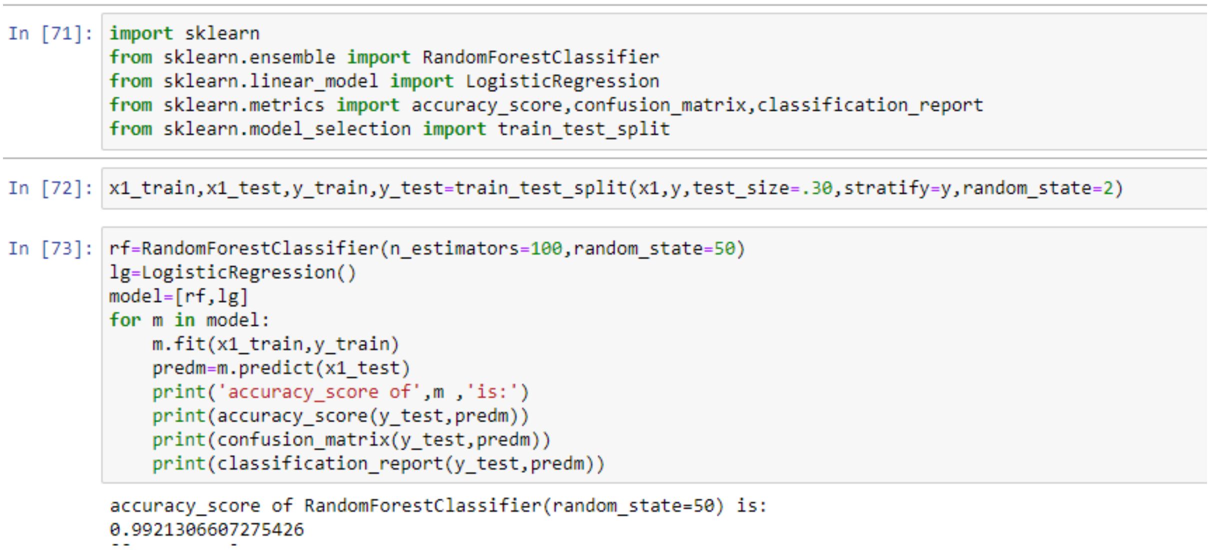
Model/s Development and Evaluation

➢ Run and evaluate selected models

Let's select our classification model for this project:

- Logistic regression
- RandomForestClassifier

➢ Testing of Identified Approaches (Algorithms)



```
In [71]: import sklearn
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.linear_model import LogisticRegression
         from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
         from sklearn.model_selection import train_test_split
```

```
In [72]: x1_train,x1_test,y_train,y_test=train_test_split(x1,y,test_size=.30,stratify=y,random_state=2)
```

```
In [73]: rf=RandomForestClassifier(n_estimators=100,random_state=50)
         lg=LogisticRegression()
         model=[rf,lg]
         for m in model:
             m.fit(x1_train,y_train)
             predm=m.predict(x1_test)
             print('accuracy_score of',m ,'is:')
             print(accuracy_score(y_test,predm))
             print(confusion_matrix(y_test,predm))
             print(classification_report(y_test,predm))
```

```
accuracy_score of RandomForestClassifier(random_state=50) is:
0.9921306607275426
```

➢ Key Metrics for success in solving problem under consideration

Selection of a model requires evaluation and evaluation requires a good metric. This is indeed important. If we optimize a model based on incorrect metric, then, our model might not be suitable for the business goals.

We have a number of metrics, for example, accuracy, recall, precision, F1 score, area under receiver operating characteristic curve, to choose from.

CONCLUSION

Key aspects of building successful classifier are:

- Selecting correct data according to the purpose or problem statement.
- Proper processing and understanding of the data
- Selecting the model and optimizing the model.

I have used tfidfvectorizer for convert object data type into int datatype as machine doesn't understand object type data.

I have used three Classification model and found Random Forest Classifier to be the best fit. It is giving 99% accuracy.