Fake News Detection

NLP and ML Project

Contents

- What is Fake News Detection?
- Problem Statement
- Data source
- Importing libraries
- Importing data
- Understanding data
- Visualization and EDA
- Data PreProcessing
- Vectorization
- SMOTE
- Model Building
- Accuracy of different model
- Joblib
- Conclusion

What is Fake News Detection?

- The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media.
- Essentially, Linguistic Cue approaches detect fake news by catching the information manipulators in the writing style of the news content.
- 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.

Problem Statement

- Further dangerlies in other electronic media using this as a source for their news thereby carrying forward further spread of such news.
- The problem is to identify the authenticity of the news and online content. Equally important problem is to identify the bots involved in spreading false news.
- 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.

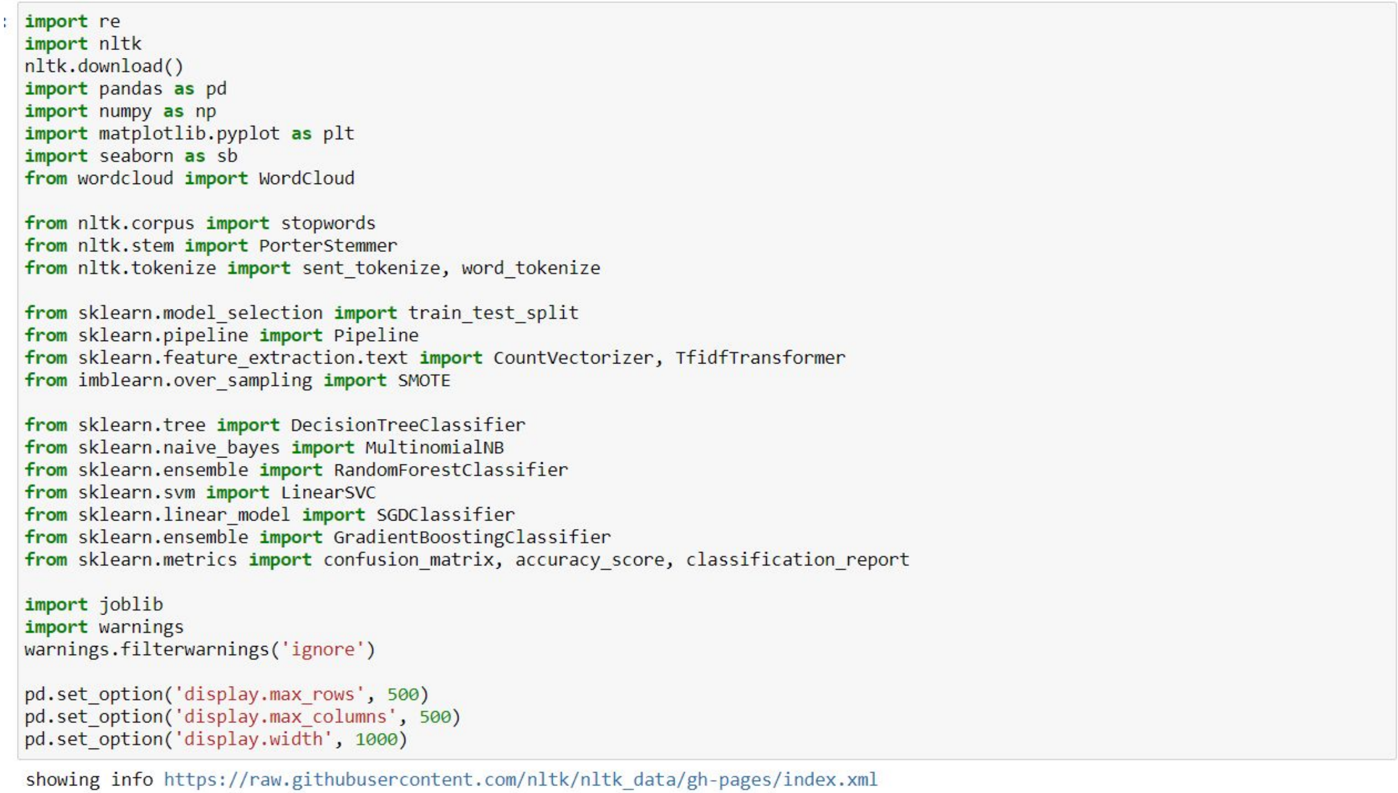
Data source

- In this project, we got a dataset in the fake-news_data.zip folder from organization.
- The folder contains a CSV files train_news.csv and you have to use the train_news.csv data to build a model to predict whether a news is fake or not fake.
- We have to try out different models on the dataset, evaluate their performance, and finally report the best model you got on the data and its performance.

Data Description

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

Importing libraries

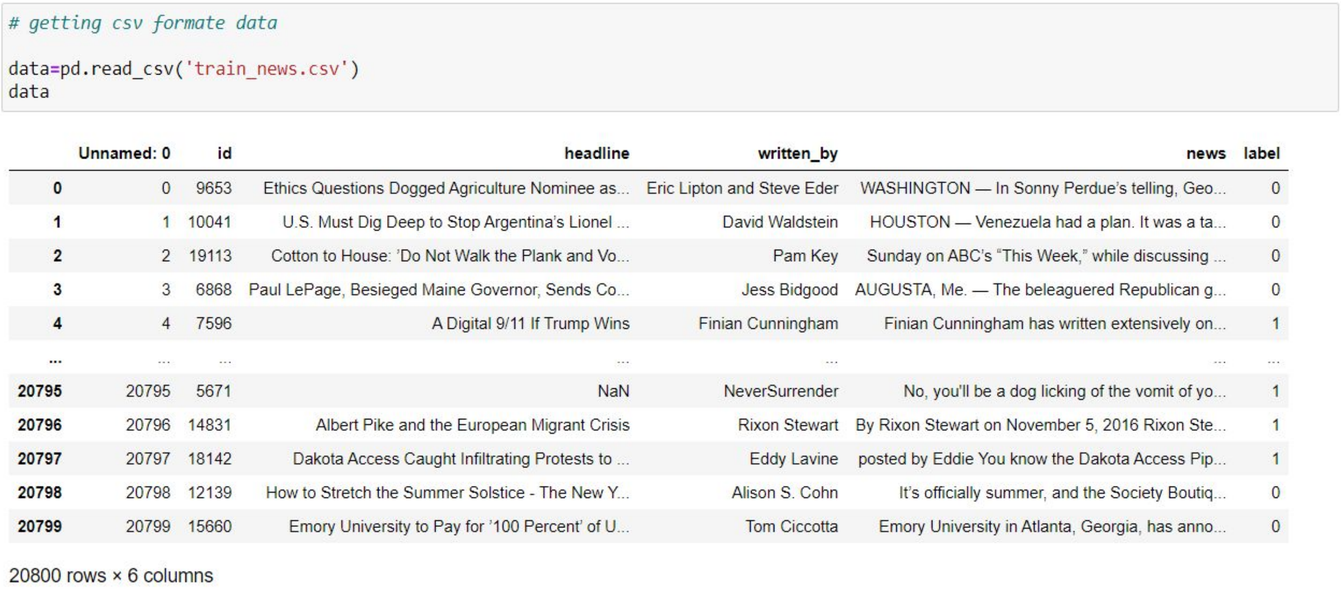


```
import re
import nltk
nltk.download()
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
from wordcloud import WordCloud

from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize

from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from imblearn.over_sampling import SMOTE

from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.linear_model import SGDClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

import joblib
import warnings
warnings.filterwarnings('ignore')

pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

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

Importing data

Getting csv format data and importing data using pandas



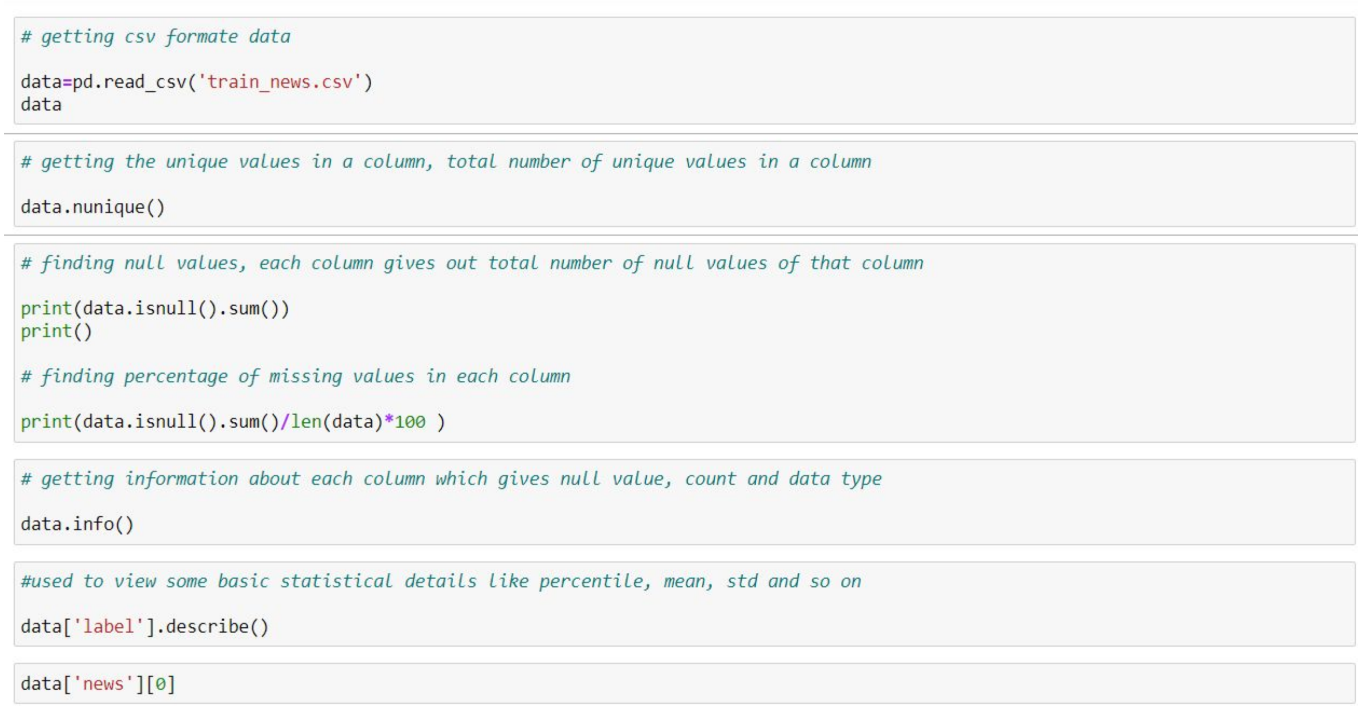
```
# getting csv formate data

data=pd.read_csv('train_news.csv')
data
```

	Unnamed: 0	id	headline	written_by	news	label
0	0	9653	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0
1	1	10041	U.S. Must Dig Deep to Stop Argentina's Lionel ...	David Waldstein	HOUSTON — Venezuela had a plan. It was a ta...	0
2	2	19113	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0
3	3	6868	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0
4	4	7596	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1
...	...	...	...	...	...	...
20795	20795	5671	NaN	NeverSurrender	No, you'll be a dog licking of the vomit of yo...	1
20796	20796	14831	Albert Pike and the European Migrant Crisis	Rixon Stewart	By Rixon Stewart on November 5, 2016 Rixon Ste...	1
20797	20797	18142	Dakota Access Caught Infiltrating Protests to ...	Eddy Lavine	posted by Eddie You know the Dakota Access Pip...	1
20798	20798	12139	How to Stretch the Summer Solstice - The New Y...	Alison S. Cohn	It's officially summer, and the Society Boutiq...	0
20799	20799	15660	Emory University to Pay for '100 Percent' of U...	Tom Ciccotta	Emory University in Atlanta, Georgia, has anno...	0

20800 rows × 6 columns

Understanding data

Understanding data using the following functions



```
# getting csv formate data

data=pd.read_csv('train_news.csv')
data
```

```
# getting the unique values in a column, total number of unique values in a column

data.nunique()
```

```
# finding null values, each column gives out total number of null values of that column

print(data.isnull().sum())
print()

# finding percentage of missing values in each column

print(data.isnull().sum()/len(data)*100 )
```

```
# getting information about each column which gives null value, count and data type

data.info()
```

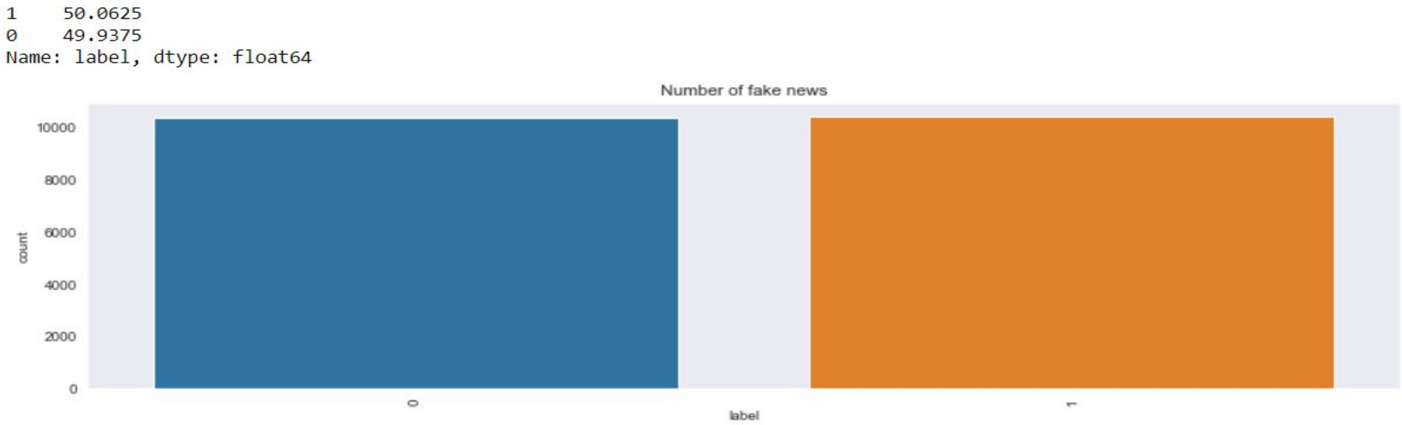
```
#used to view some basic statistical details like percentile, mean, std and so on

data['label'].describe()
```

```
data['news'][0]
```

Visualization and EDA

Visual presentation of Class percentage

50.06% news is fake and 29.93 % news is genuine



```
1    50.0625
0    49.9375
Name: label, dtype: float64
```

Data PreProcessing

- **Filling nan using fillna**

  dropping null values from the dataset

  ```
  data = data.fillna(' ')
  ```

- **Cleaning text process**

  The function to clean text

  ```
  def clean_text(words):
      words = re.sub("[^a-zA-Z]"," ", words)
      text = words.lower().split()
      return " ".join(text)
  data['News_cleaned_data'] = data['News'].apply(clean_text)
  ```

- **Removing stop words**

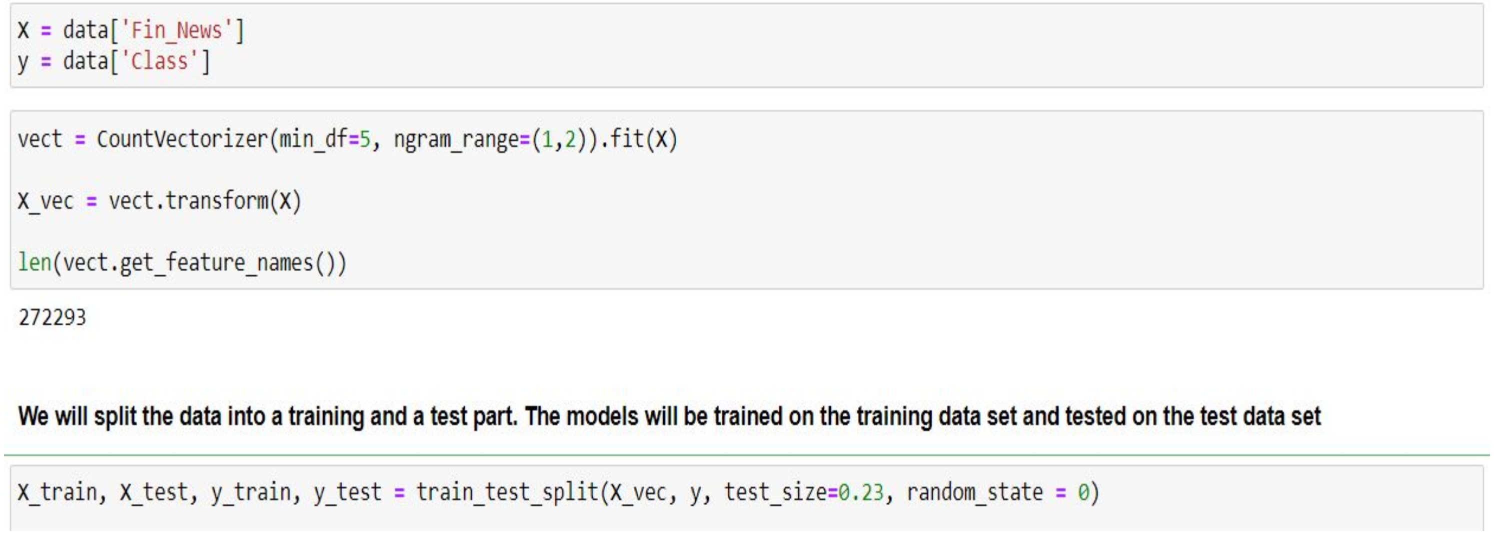
  ```
  def remove_stopwords(text):
      text = [word.lower() for word in text.split() if word.lower() not in stop_words]
      return " ".join(text)
  data['News_removed_stopwords'] = data['News_cleaned_data'].apply(remove_stopwords)
  ```

● Applying Stemming

```
from nltk.stem import PorterStemmer

 porter = PorterStemmer()

 The function to apply stemming

 def stemmer(stem_text):
    stem_text = [porter.stem(word) for word in stem_text.split()]
    return " ".join(stem_text)
 data['Fin_News'] = data['News_removed_stopwords'].apply(stemmer)
```

Vectorization

- To run machine learning algorithms we need to convert text files into numerical feature vectors.
- We will use bag of words model for our analysis.
- We splitting the complete data into X as message and y as class values.



```
X = data['Fin_News']
y = data['Class']
```

```
vect = CountVectorizer(min_df=5, ngram_range=(1,2)).fit(X)

X_vec = vect.transform(X)

len(vect.get_feature_names())
```

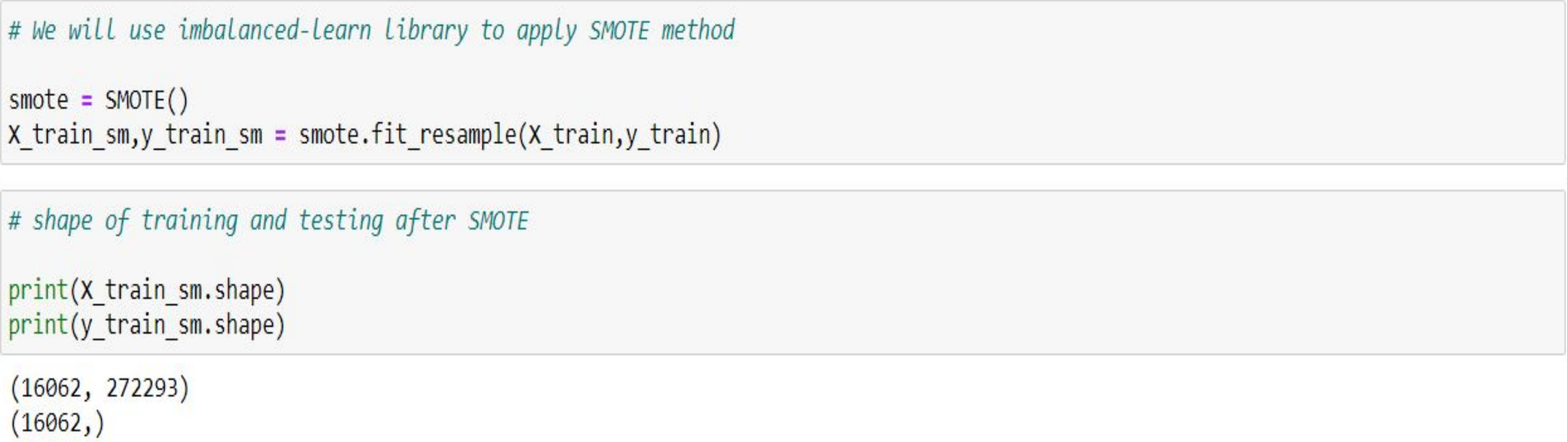
```
272293
```

We will split the data into a training and a test part. The models will be trained on the training data set and tested on the test data set

```
X_train, X_test, y_train, y_test = train_test_split(X_vec, y, test_size=0.23, random_state = 0)
```

SMOTE

- The target class variable is imbalanced, The simplest way to improve imbalanced dataset is balancing them by oversampling instances of the minority class or undersampling instances of the majority class.
- We will try to balancing classes by using one of the advanced techniques like the SMOTE method.
- SMOTE technique is one of the most commonly used oversampling methods to solve the imbalance problem. It goal is to balance class distribution by randomly increasing minority class examples by replicating them.



```
# We will use imbalanced-learn library to apply SMOTE method

smote = SMOTE()
X_train_sm,y_train_sm = smote.fit_resample(X_train,y_train)
```

```
# shape of training and testing after SMOTE

print(X_train_sm.shape)
print(y_train_sm.shape)
```

```
(16062, 272293)
(16062,)
```

Model Building

We use the following classification models:

1. Decision tree Classifier
2. Naive Bayes Classifier
3. Random Forest Classifier
4. Gradient Boosting
5. SVM (Support Vector Machine)
6. Stochastic Gradient Descent

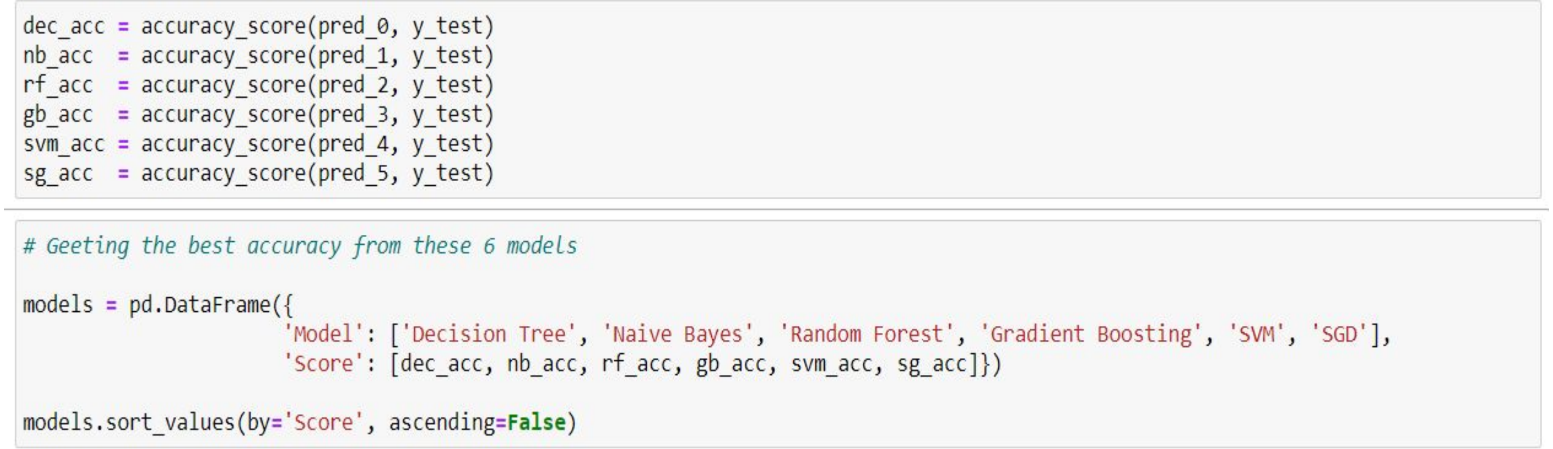
To make the vectorizer > transformer > classifier easier to work with, we will use Pipeline class in SKLearn

```
model_dtc = Pipeline([('tfidf', TfidfTransformer()),
                     ('model',DecisionTreeClassifier()),
                    ])

model_dtc.fit(X_train_sm,y_train_sm)

ytest = np.array(y_test)
pred_0 = model_dtc.predict(X_test)

print('accuracy %s' % accuracy_score(pred_0, y_test))
print(classification_report(ytest, pred_0))
```

The same code is applied for the following models.

Accuracy of different model

We tested six different models and now we will be finding the best model with accuracy.



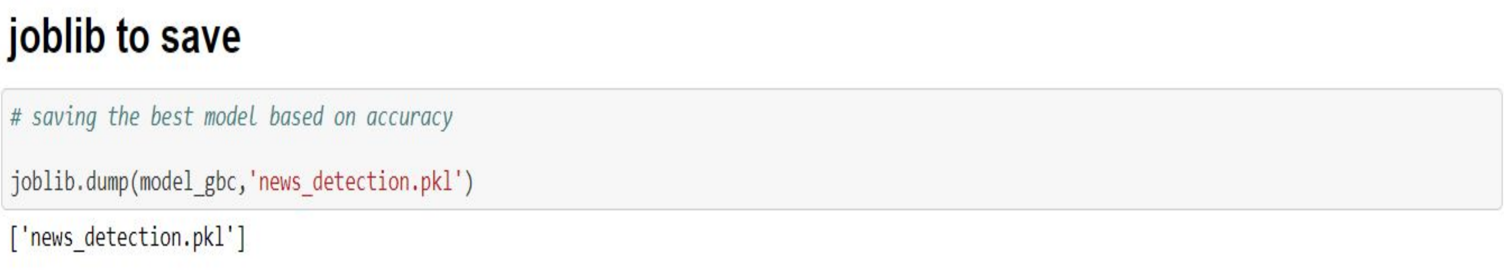
```
dec_acc = accuracy_score(pred_0, y_test)
nb_acc  = accuracy_score(pred_1, y_test)
rf_acc  = accuracy_score(pred_2, y_test)
gb_acc  = accuracy_score(pred_3, y_test)
svm_acc = accuracy_score(pred_4, y_test)
sg_acc  = accuracy_score(pred_5, y_test)
```

```
# Geeting the best accuracy from these 6 models

models = pd.DataFrame({
                      'Model': ['Decision Tree', 'Naive Bayes', 'Random Forest', 'Gradient Boosting', 'SVM', 'SGD'],
                      'Score': [dec_acc, nb_acc, rf_acc, gb_acc, svm_acc, sg_acc]})

models.sort_values(by='Score', ascending=False)
```

Gradient Boosting	0.983
Decision Tree	0.977
SVM	0.977
SGD	0.974
Random Forest	0.930
Naive Bayes	0.750

joblib to save

Saving the best model with best accuracy



joblib to save

```
# saving the best model based on accuracy

joblib.dump(model_gbc,'news_detection.pkl')
```

```
['news_detection.pkl']
```

Conclusion

- This project was aimed to text classification to determined whether the News is Fake or genuine. We started with the data cleaning and text mining, which cover change text into tokens, remove punctuation, stop words and normalization them by stemming.
- Following we used bag of words model to convert the text into numerical feature vectors.
- Finally we started training six different classification models and we got the best accuracy of 0.982 for Gradient Boosting and 0.978 for Decision Tree.