



FAKE NEWS DETECTION

Submitted by:

PRITI CHAUHAN

ACKNOWLEDGMENT

I would like to acknowledge Ms. Khshboo Garg for giving this assignment.

I would want to convey my sincere thanks Datatrained Academy and their guidance without them, the task would not have been accomplished.

The website that I referred are:

<https://learning.datatrained.com>

<https://github.com>

<https://www.geeksforgeeks.org>

<https://www.kaggle.com>

INTRODUCTION

- Business Problem Framing

Describe the business problem and how this problem can be related to the real world.

Context 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. Content what's inside is more than just rows and columns. Make it easy for others to get started by describing how you acquired the data and what time period it represents, too. What is a Fake News? 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.

- Workflow

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

-Natural Language Processing Machine learning data only works with numerical features so we have to convert text data into numerical columns. So we have to pre-process the text and that is called natural language processing. In-text pre-process we are cleaning our text by steaming, lemmatization, remove stopwords, remove special symbols and numbers, etc. After cleaning the data we have to feed this text data into a vectorizer which will convert this text data into numerical features.

-Dataset You can find many datasets for fake news detection on Kaggle or many other sites. I download these datasets from Kaggle. 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. You have to insert one

label column zero for fake news and one for true news. We are combined both datasets using pandas built-in function.

- **Conceptual Background of the Domain Problem**

Describe the domain related concepts that you think will be useful for better understanding of the project.

Answer: Product, product type, machine learning models, Web scraping, Natural language programming.

- **Motivation for the Problem Undertaken**

Describe your objective behind to make this project, this domain and what the motivation is behind.

Answer: It can be seen that fake news and true news. The fake news can be eliminated so that it doesn't affect decision making.

- **Analytical Problem Framing**

Mathematical/ Analytical Modeling of the Problem

Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

Answer: NLP toolkit is then used for converting the reviews to vectors where these vectors are employed for the machine learning models. Machine learning models like MLP, Naïve bias etc.

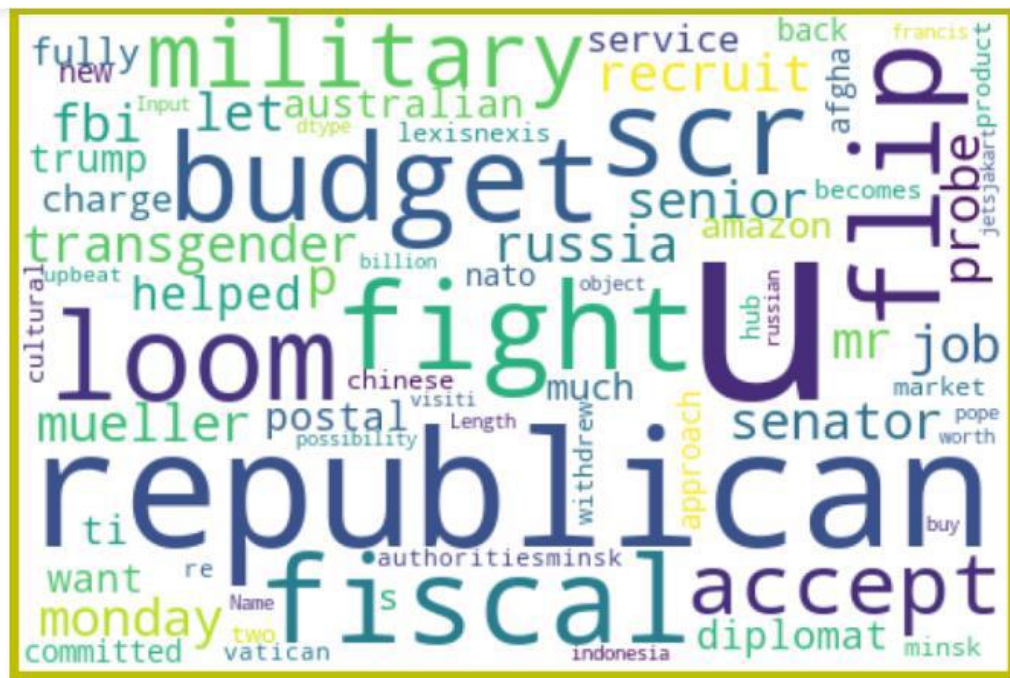
- **Data Sources and their formats**

What are the data sources, their origins, their formats and other details that you find necessary? They can be described here. Provide a proper data description. You can also add a snapshot of the data.

Answer: The data set can be web scraped in our case it was given by customer.

- **Data Pre-processing Done**

What were the steps followed for the cleaning of the data? What were the assumptions done and what were the next actions steps over that?



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

Here, you can describe any presumptions taken by you.

Answer: We are testing with customers data.

- **Hardware and Software Requirements and Tools Used**

Listing down the hardware and software requirements along with the tools, libraries and packages used. Describe all the software tools used along with a detailed description of tasks done with those tools.

Answer: Selenium and beautiful soup for web scraping, pandas, numpy, matplotlib, seaborn for data handling. Nltk tool kit like stopwords, lematization, vectorization etc. for cleaning and conversion of data into input a trainable model.

- **Model/s Development and Evaluation**

- **Testing of Identified Approaches (Algorithms)**

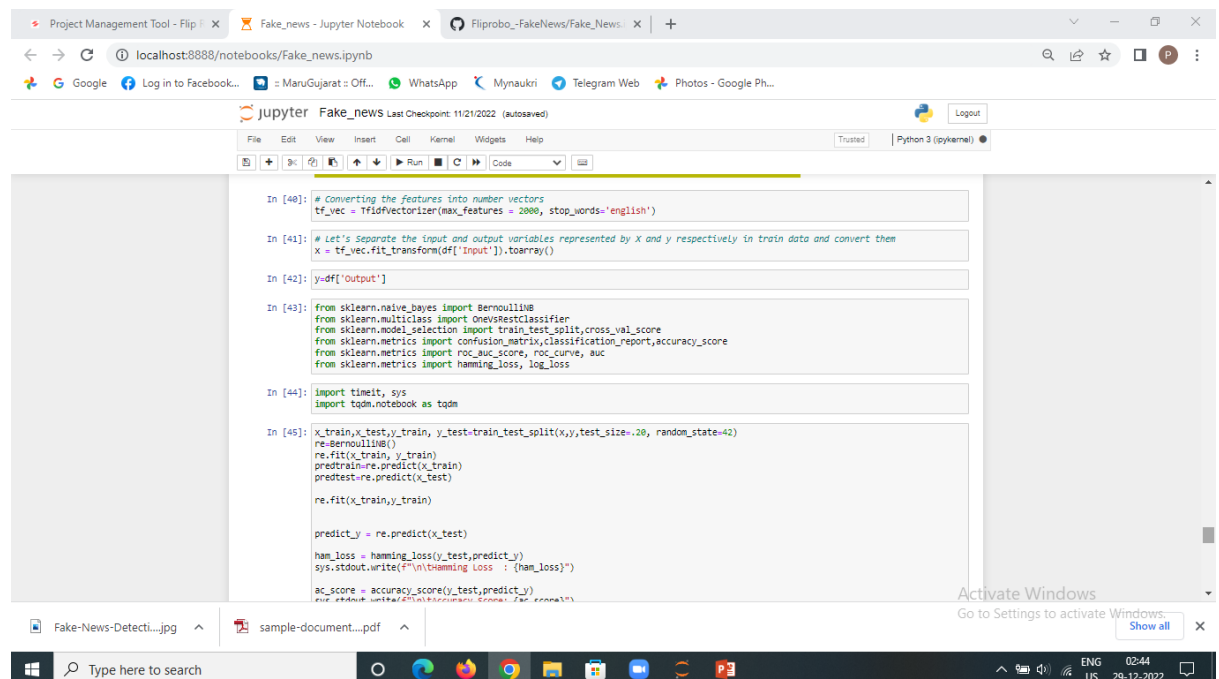
Listing down all the algorithms used for the training and testing.

Answer: Machine learning models like Multi-layer perceptron, MultinomialNB, Naïve bias algorithm etc.

- **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.**

Answer:



```
In [40]: # Converting the features into number vectors
tf_vec = TfidfVectorizer(max_features = 2000, stop_words='english')

In [41]: # Let's Separate the input and output variables represented by x and y respectively in train data and convert them
x = tf_vec.fit_transform(df['Input']).toarray()

In [42]: y=df['Output']

In [43]: from sklearn.naive_bayes import BernoulliNB
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.metrics import roc_auc_score, roc_curve, auc
from sklearn.metrics import hamming_loss, log_loss

In [44]: import timeit, sys
import tqdm.notebook as tqdm

In [45]: x_train,x_test,y_train, y_test=train_test_split(x,y,test_size=.20, random_state=42)
re=BernoulliNB()
re.fit(x_train, y_train)
predtrain=re.predict(x_train)
predtest=re.predict(x_test)
re.fit(x_train,y_train)

predict_y = re.predict(x_test)

ham_loss = hamming_loss(y_test,predict_y)
sys.stdout.write(f'\n\nHamming Loss : {ham_loss}')

ac_score = accuracy_score(y_test,predict_y)
```



```
In [42]: y=df['Output']

In [43]: from sklearn.naive_bayes import BernoulliNB
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.metrics import roc_auc_score, roc_curve, auc
from sklearn.metrics import hamming_loss, log_loss

In [44]: import timeit, sys
import tqdm.notebook as tqdm

In [45]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.20, random_state=42)
re=BernoulliNB()
re.fit(x_train,y_train)
predtrain=re.predict(x_train)
predtest=re.predict(x_test)

re.fit(x_train,y_train)

predict_y = re.predict(x_test)

ham_loss = hamming_loss(y_test,predict_y)
sys.stdout.write(f'\nHamming Loss : {ham_loss}')

ac_score = accuracy_score(y_test,predict_y)
sys.stdout.write(f'\nAccuracy Score : {ac_score}')

cl_report = classification_report(y_test,predict_y)
sys.stdout.write(f'\n{cl_report}')
```

Hamming Loss : 0.031999910913140314
Accuracy Score: 0.968040089808597
precision recall f1-score support

- **Key Metrics for success in solving problem under consideration**
What were the key metrics used along with justification for using it?
You may also include statistical metrics used if any.
Answer: The key metrics used along the model prediction are accuracy score, precision, recall, f1-score and hamming score.
- **Visualizations**
Mention all the plots made along with their pictures and what were the inferences and observations obtained from those. Describe them in detail.
If different platforms were used mention that as well.
- **Interpretation of the Results**
Give a summary of what results were interpreted from the visualizations, pre-processing and modelling.
Answer: From the visualization we can see that the data can be used for prediction as it a balanced dataset. The word cloud shows different words for true news and false news.

- CONCLUSION

- Key Findings and Conclusions of the Study

The fake and true news was predicted using naive_bayes algorithm and Multilayer perceptron (MLP). It was seen that the naive_bayes yields Hamming Loss: 0.031959910913140314 Accuracy Score: 0.968040089086859

The Multilayer perceptron (MLP) gives Hamming Loss: 0.012806236080178173 Accuracy Score: 0.9871937639198218.

Learning Outcomes of the Study in respect of Data Science

By visualization we can learn data distribution, words distribution before and after cleaning, Positive words and negative words etc. The fake and true news was predicted using naive_bayes algorithm and Multilayer perceptron (MLP).