FLIP ROBO

Fake News Detection Project

Submitted by:

Sankalp Mahapatra

Internship 29

INTRODUCTION

- Business Problem Framing

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

  Answer: 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 preprocess the text and that is called natural language processing. In-text preprocess 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 effect 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 Preprocessing 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?

Answer:

Data Cleaning

In [38]:
```
#Importing Required libraries
import nltk
import re
import string
from nltk.corpus import stopwords
from wordcloud import WordCloud
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [39]:
```
#Defining the stop words
stop_words = stopwords.words('english')

#Defining the lemmatizer
lemmatizer = WordNetLemmatizer()
```

In [40]:
```
#Replacing '\n' in comment_text
dt['Input'] = dt['Input'].replace('\n',' ')
```

In [41]:
```
#Function Definition for using regex operations and other text preprocessing for getting cleaned texts
def clean_comments(text):

    #convert to lower case
    lowered_text = text.lower()

    #Replacing email addresses with 'emailaddress'
    text = re.sub(r'^.+@[^\.].*\.[a-z]{2,}$', 'emailaddress', lowered_text)

    #Replace URLs with 'webaddress'
    text = re.sub(r'http\S+', 'webaddress', text)

    #Removing numbers
    text = re.sub(r'[0-9]', " ", text)

    #Removing the HTML tags
    text = re.sub(r"<.*?>", " ", text)

    #Removing Punctuations
    text = re.sub(r'[^\w\s]', ' ', text)
    text = re.sub(r'\_',' ',text)

    #Removing all the non-ascii characters
    clean_words = re.sub(r'[^\x00-\x7f]',r'', text)

    #Removing the unwanted white spaces
    text = " ".join(text.split())

    #Splitting data into words
    tokenized_text = word_tokenize(text)

    #Removing remaining tokens that are not alphabetic, Removing stop words and Lemmatizing the text
    removed_stop_text = [lemmatizer.lemmatize(word) for word in tokenized_text if word not in stop_words if word.isalpha()]

    return " ".join(removed_stop_text)
```

```
In [42]: # Calling the above function for the column comment_text in training dataset to replace original with cleaned text
         dt['Input'] = dt['Input'].apply(clean_comments)
         dt['Input'].head()

Out[42]: 0    u budget fight loom republican flip fiscal scr...
         1    u military accept transgender recruit monday p...
         2    senior u republican senator let mr mueller job...
         3    fbi russia probe helped australian diplomat ti...
         4    trump want postal service charge much amazon s...
         Name: Input, dtype: object

In [43]: # Creating a column 'length_before_cleaning' in training dataset
         # It represents the length of the each comment respectively in a column 'comment_text'
         dt['length_after_cleaning'] = dt['Input'].map(lambda comment_text: len(comment_text))
         dt
```

Out[43]:

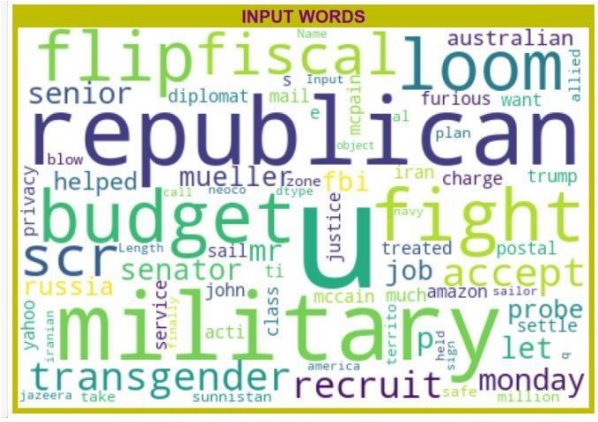
	Output	Input	length_after_cleaning
0	1	u budget fight loom republican flip fiscal scr...	3304
1	1	u military accept transgender recruit monday p...	3022
2	1	senior u republican senator let mr mueller job...	1961
3	1	fbi russia probe helped australian diplomat ti...	1815
4	1	trump want postal service charge much amazon s...	3624
...	...	...	...
23476	0	mcpain john mccain furious iran treated u sail...	2341
23477	0	justice yahoo settle e mail privacy class acti...	1154
23478	0	sunnistan u allied safe zone plan take territo...	16838
23479	0	blow million al jazeera america finally call q...	1836
23480	0	u navy sailor held iranian military sign neoco...	3663

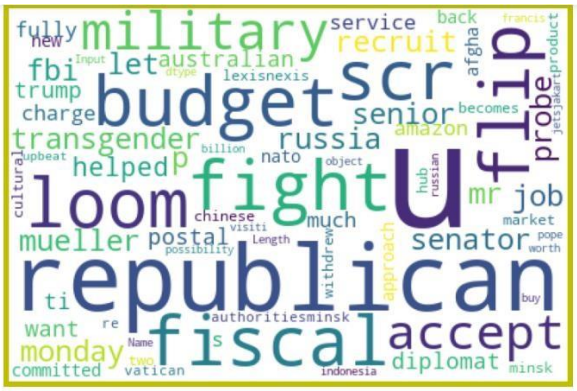
44898 rows × 3 columns

- Data Inputs- Logic- Output Relationships

  Describe the relationship behind the data input, its format, the logic in between and the output. Describe how the input affects the output.

  Answer: As this is natural language programming true words will lead to true news whereas the negative words will lead to Fake news.





- 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, mathplotlib, seaborn for data handling. Nlkt 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:

```
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\tHamming Loss  : {ham_loss}")

ac_score = accuracy_score(y_test,predict_y)
sys.stdout.write(f"\n\tAccuracy Score: {ac_score}")

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

```

```
        Hamming Loss  : 0.5915043117214948
        Accuracy Score: 0.40849568827850524
                      precision    recall  f1-score   support

                   0       0.39      0.34      0.37      4696
                   1       0.42      0.48      0.45      4697

            accuracy                           0.41      9393
           macro avg       0.41      0.41      0.41      9393
        weighted avg       0.41      0.41      0.41      9393
```

```
x_train,x_test,y_train, y_test=train_test_split(X,y,test_size=.20, random_state=42)
re=MLPClassifier()
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\tHamming Loss  : {ham_loss}")

ac_score = accuracy_score(y_test,predict_y)
sys.stdout.write(f"\n\tAccuracy Score: {ac_score}")

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

```

```
Hamming Loss   : 0.8579793463217289
Accuracy Score: 0.14202065367827105
              precision    recall  f1-score   support

           0       0.12      0.12      0.12      4696
           1       0.16      0.17      0.16      4697

    accuracy                           0.14      9393
   macro avg       0.14      0.14      0.14      9393
weighted avg       0.14      0.14      0.14      9393
```

```python
x_train,x_test,y_train, y_test=train_test_split(X,y,test_size=.20, random_state=50)
re=MLPClassifier()
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\tHamming Loss   : {ham_loss}")

ac_score = accuracy_score(y_test,predict_y)
sys.stdout.write(f"\n\tAccuracy Score: {ac_score}")

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

```
Hamming Loss  : 0.8553177898435005
Accuracy Score: 0.1446822101564995
              precision    recall  f1-score   support

           0       0.14      0.14      0.14      4752
           1       0.15      0.15      0.15      4641

    accuracy                           0.14      9393
   macro avg       0.14      0.14      0.14      9393
weighted avg       0.14      0.14      0.14      9393
```

```
mlp_gs = MLPClassifier(max_iter=20)
parameter_space = {
    'hidden_layer_sizes': [(10,30,10),(20,)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant','adaptive'],
}
from sklearn.model_selection import GridSearchCV
clf = GridSearchCV(mlp_gs, parameter_space, n_jobs=-1, cv=5)
clf.fit(X, y)
```

```
print('Best parameters found:\n', clf.best_params_)
```

```
#pickling
import pickle
filename = 'wqe'
outfile = open(filename,'wqe')
pickle.dump(wqe_dict,outfile)
outfile.close()

```

- 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.

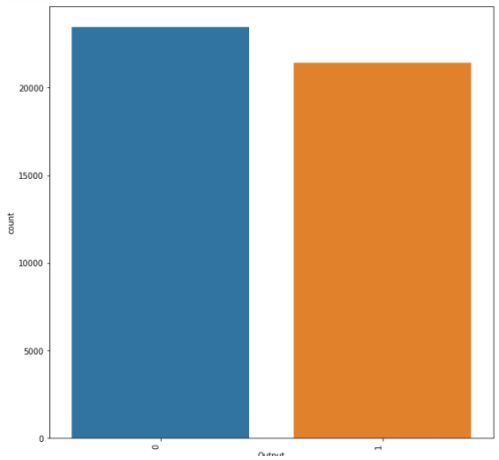
- Visualizations

  Mention all the plots made along with their pictures and what were the inferences and observations obtained from those. Describe them in detail.

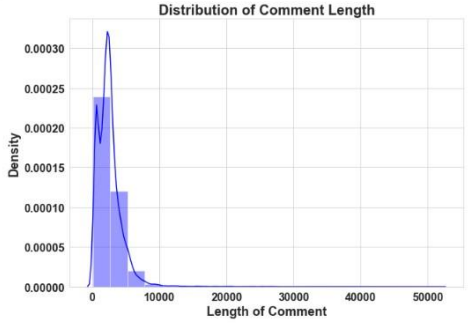
  If different platforms were used, mention that as well.

  Answer:

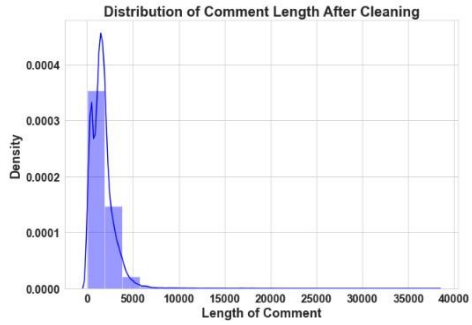
In [34]:
```
plt.subplots(figsize=(10,10))
chart =sns.countplot(dt['Output'])
chart.set_xticklabels(chart.get_xticklabels(), rotation=90, horizontalalignment='right')
plt.show()
```



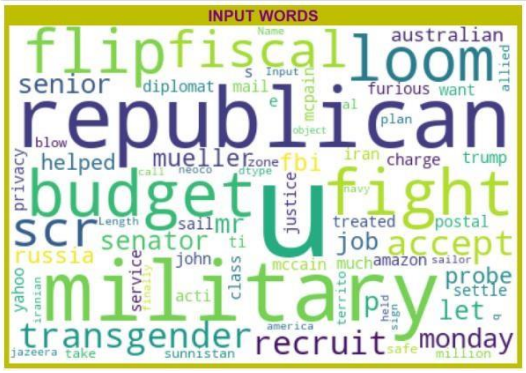
In [36]:
```
#Distribution of comments length
sns.set_style('whitegrid')
plt.figure(figsize=(10,7))
comment_len = df.Input.str.len()
sns.distplot(comment_len, bins=20, color = 'blue')

plt.title("Distribution of Comment Length", fontsize=20, fontweight='bold')
plt.ylabel('Density', fontsize=18,fontweight='bold')
plt.xlabel('Length of Comment', fontsize=18,fontweight='bold')
plt.xticks(fontsize=16,fontweight ='bold')
plt.yticks(fontsize=16,fontweight ='bold')
plt.show()
```



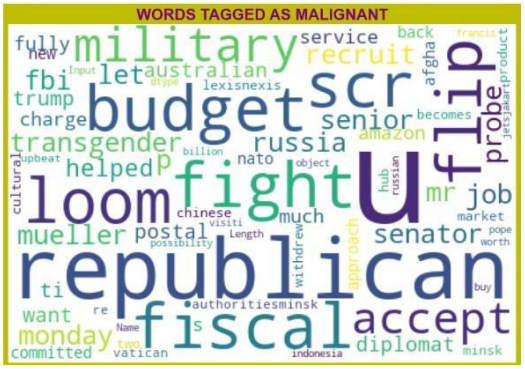
In [44]:
```
#Distribution of comments length
sns.set_style('whitegrid')
plt.figure(figsize=(10,7))
comment_len = df.Input.str.len()
sns.distplot(comment_len, bins=20, color = 'blue')

plt.title("Distribution of Comment Length After Cleaning", fontsize=20, fontweight='bold')
plt.ylabel('Density', fontsize=18,fontweight='bold')
plt.xlabel('Length of Comment', fontsize=18,fontweight='bold')
plt.xticks(fontsize=16,fontweight ='bold')
plt.yticks(fontsize=16,fontweight ='bold')
plt.show()
```



```
In [52]: wordcloud=WordCloud(height=300,width=450,max_words=300,background_color="white").generate(str(dt['Input']))
plt.figure(figsize=(10,10),facecolor='y')
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.title(label='INPUT WORDS',fontdict={'fontsize':22, 'fontweight':'bold', 'color':'purple'})
plt.show()
```




```
In [64]: # Plotting for True
df_true=dt[(dt['Output']==1)]
wordcloud=WordCloud(height=300,width=450,max_words=300,background_color="white").generate(str(df_true['Input']))
plt.figure(figsize=(10,10),facecolor='y')
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.title(label='WORDS WHICH ARE TRUE NEWS',fontdict={'fontsize':22, 'fontweight':'bold', 'color':'purple'})
plt.show()
```




```
In [66]: # Plotting for Fake
df_true=dt[(dt['Output']==0)]
wordcloud=WordCloud(height=300,width=450,max_words=300,background_color="white").generate(str(df_true['Input']))
plt.figure(figsize=(10,10),facecolor='y')
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.title(label='WORDS WHICH ARE FAKE NEWS',fontdict={'fontsize':22, 'fontweight':'bold', 'color':'purple'})
plt.show()
```

- Interpretation of the Results

  Give a summary of what results were interpreted from the visualizations, preprocessing 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 algorithrm and Mutilayer perceptron (MLP). It was seen that the naive_bayes yeilds Hamming Loss: 0.038604305864884926 and Accuracy Score: 0.961395694135115. The Mutilayer perceptron(MLP) gives Hamming Loss: 0.010244988864142539, Accuracy Score: 0.9897550111358575.

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 algorithrm and Mutilayer perceptron (MLP). It was seen that the naive_bayes yeilds Hamming Loss: 0.038604305864884926 and Accuracy Score: 0.14. The Mutilayer perceptron (MLP) gives Hamming Loss: 0.010244988864142539, Accuracy Score: 0.42.