FLIP ROBO

FLIP ROBO

FAKE NEWS CLASSIFIER

SUBMITTED BY

SREYASHEE JENA

ACKNOWLEDGEMENT

First of all, I extol the Almighty for pouring his blessings on me and giving me potentiality and opportunity to carry the work to its end with a success.

"It's impossible to prepare a project report without the help and fillip of some people and certainly this project report is of no exception."

At the commencement of this project report I would like to evince my deepest sense of gratitude to Ms. Astha Mishra, my honored mentor. Without her guidance, insightful decision, valuable comments and correction it would not have possible to reach up to this mark.

I would like to draw my gratitude to Flip Robo and Data Trained for providing me a suitable environment and guidance to complete my work. Last but not least thanks to the brilliant authors from where I have got the idea to carry out the project.

References were taken from various articles from Medium, KDnuggets, Towards Data Science, Machine Learning Mastery, Analytics Vidya, American Statistical Association, Research Gate and documentations of Python and Sklearn.

SREYASHEE JENA

CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 BUSINESS PROBLEM

In current world scenario it's easy to find news but it's becoming harder and harder to guarantee the authenticity of that piece of news. These kinds of news are swarming all over corrupting everything. 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.

1.2 BACKGROUND

The rate of production of fake news have increased exponentially. In the past news obtained from newspaper, radio or TV were considered as the best and authentic source of information about the real world and ongoing situations but now everything has changed. In the run of popularity and ill mind set the media houses and social media are spreading fake news. It's becoming harder and harder to say whether a piece of news is real or fabricated. Its becoming a new weapon of destruction.

The effect of fake news can be seen everywhere. The fake news leads to communal disturbance, character assassination, mental trauma, sometimes it is used as a weapon to achieve some illicit plans etc. these are like wild fire which spread too quickly and difficult to control. Its really hard for a human being to differentiate between a fake news and an authentic news.

However, this news can be classified into fake or authentic news group with the help of python and algorithms of Machine Learning and Natural Language Processing.

1.3 MOTIVATION

The project was the first provided to me by FlipRobo as a part of the internship programme. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary objective. However, the motivation for taking this project was that it is relatively a new field of research. Here we have many options but less concrete solutions. The main motivation was to classify the news in order to bring awareness and reduce unwanted chaos.

CHAPTER-2

ANALYTICAL PROBLEM FRAMING

2.1 ANALYTICAL MODELING OF PROBLEM

People can get infected with fake news very quickly with misleading words and images and post them without any fact-checking. The social media life has been used to distribute counterfeit data, which has a significant negative influence on individual consumers and on a wider community. The fake news problem is tackled using a machine learning algorithm

The dataset provided here has a shape of (20800, 6). Which means it has 20800 rows and 6 columns. Here the target or the dependent variable named “Label” have two distinct values 0 and 1. Where 0 represents the news which are not fake or authentic while 1 represents category of fake news. As the target column ‘Label’ is giving binary outputs and all the independent variables has text so it is clear that it is supervised machine learning problem where we can use, we can use the techniques of NLP and classification-based algorithms of Machine learning.

Here we will use NLP techniques like word tokenization, lemmatization and tfidf vectorizer then those processed data will be used to create best model using various classification based supervised machine learning algorithms like Logistic Regression, Passive Aggressive Classifier, Multinomial NB, Complement NB, Random Forest Classifier.

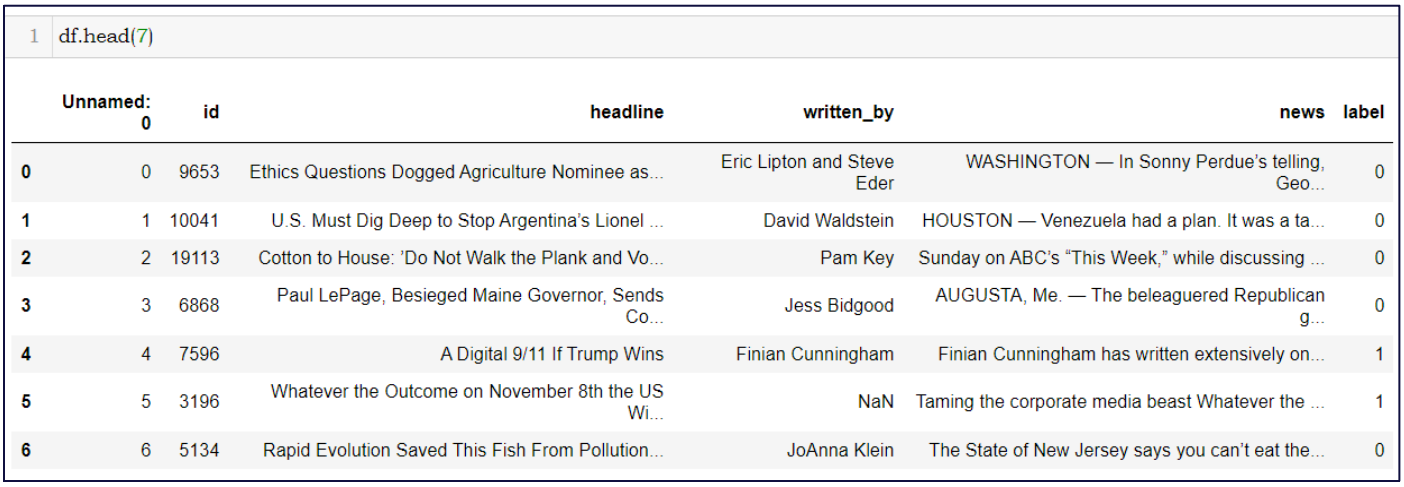
The passive Aggressive Classifier belongs to the family of online machine learning algorithms and it is very much helpful in processing large scale data. It remains passive for a correct classification and turns aggressive in case of a misclassification. Its aim is to make updates that corrects the loss causing very little change in the weight vector.

2.2 DATA SOURECE AND FORMATS

The data was provided by FlipRobo in CSV format. After loading the dataset into Jupyter Notebook using Pandas after loading with help of df.head() **[Fig. 1]** it can be seen that there are six columns named as "*Unnamed: 0, id, headline, written_by, news, label*". The metadata table is provided below [**Table 1**].

Unnamed: 0	**SERIAL NUMBER/ INDEX NUMBER**
Id	**UNIQUE ID OF EACH NEWS ARTICLE**
Headline	**THE TITLE OF THE NEWS**
Written_By	**THE AUTHOR OF THE NEWS ARTICLE**
News	**FULL TEXT OF THE NEWS ARTICLE**
Label	**IT TELLS WHETHER THE NEWS IS FAKE (1) OR NOT FAKE (0)**

(Table 1: METADATA)



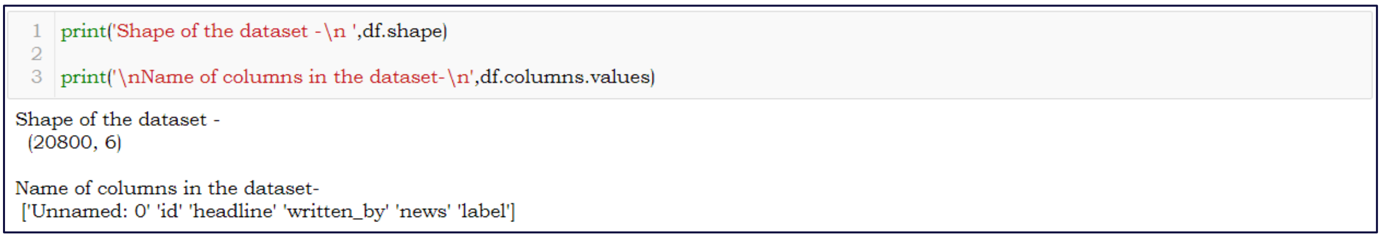
```
df.head(7)
```

	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
5	5	3196	Whatever the Outcome on November 8th the US Wi...	NaN	Taming the corporate media beast Whatever the ...	1
6	6	5134	Rapid Evolution Saved This Fish From Pollution...	JoAnna Klein	The State of New Jersey says you can't eat the...	0

(Fig 1. DATASET)

As mentioned earlier the shape of the dataset is (20800, 6). The shape of the dataset in form of a tuple can be accessed using df.shape() [**Fig 2**]. The dataset has no duplicated values but have few null values. The number of duplicated values can be seen using df.duplicated().sum() and the null values can be seen using df.isnull().sum() **[Fig 3]**.

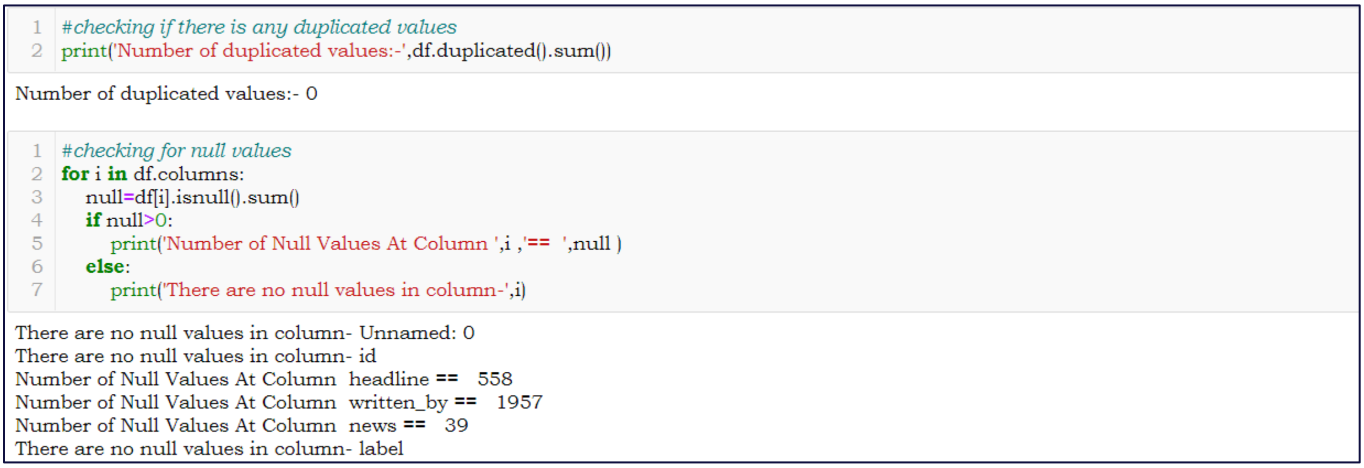
The null values can also be visualized with help of seaborn and matplotlib library **[Fig 4].**



```
print('Shape of the dataset -\n ',df.shape)

print('\nName of columns in the dataset-\n',df.columns.values)
```

```
Shape of the dataset -
  (20800, 6)

Name of columns in the dataset-
 ['Unnamed: 0' 'id' 'headline' 'written_by' 'news' 'label']
```

(Fig 2. SHAPE AND COLUMNS)



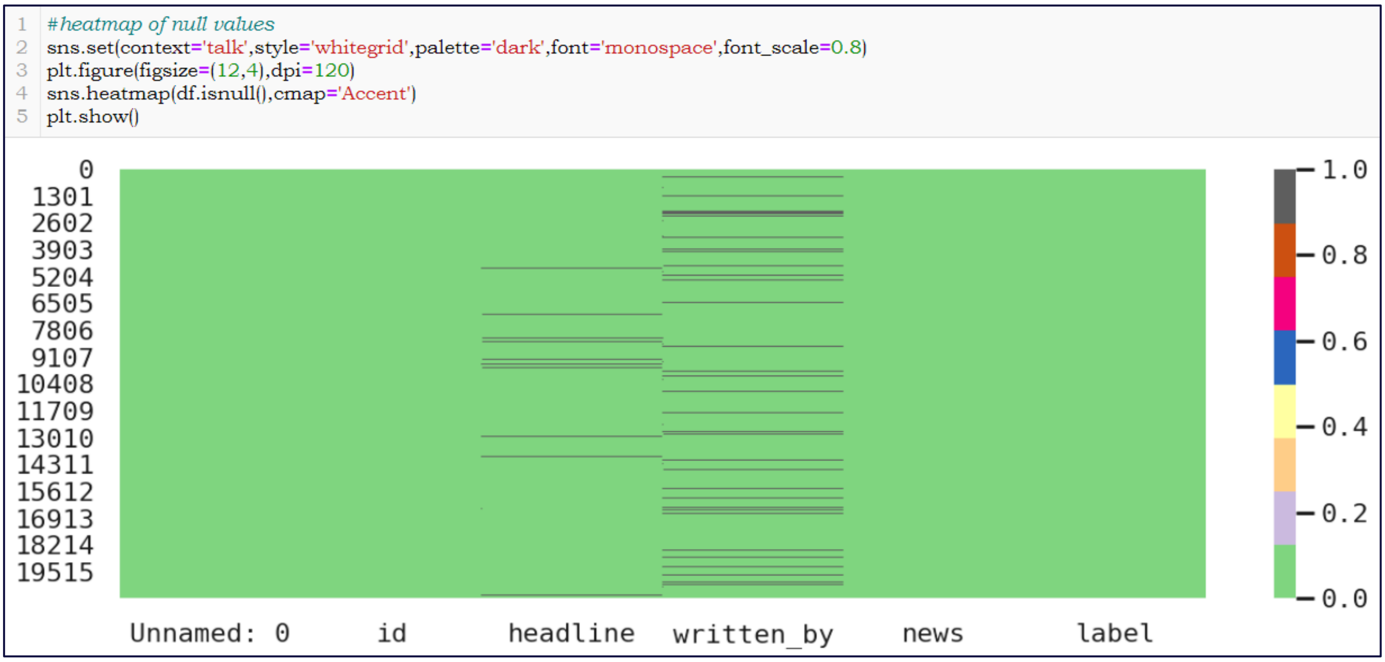
```
#checking if there is any duplicated values
print('Number of duplicated values:-',df.duplicated().sum())
```

```
Number of duplicated values:- 0
```

```
#checking for null values
for i in df.columns:
    null=df[i].isnull().sum()
    if null>0:
        print('Number of Null Values At Column ',i ,'==  ',null )
    else:
        print('There are no null values in column-',i)
```

```
There are no null values in column- Unnamed: 0
There are no null values in column- id
Number of Null Values At Column  headline ==   558
Number of Null Values At Column  written_by ==   1957
Number of Null Values At Column  news ==   39
There are no null values in column- label
```

(Fig 3. DUPLICATES AND NULL VALUES)

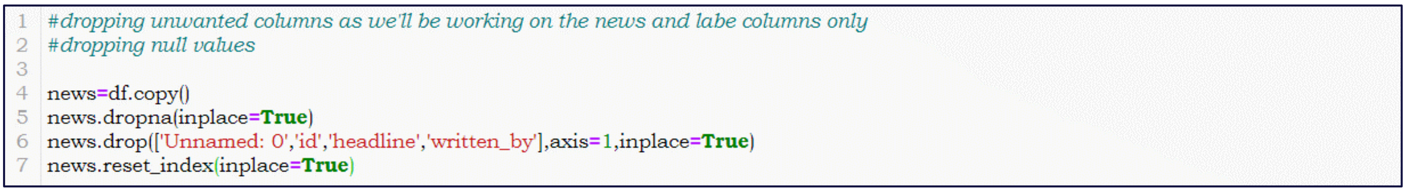


```
#heatmap of null values
sns.set(context='talk',style='whitegrid',palette='dark',font='monospace',font_scale=0.8)
plt.figure(figsize=(12,4),dpi=120)
sns.heatmap(df.isnull(),cmap='Accent')
plt.show()
```

(Fig 4. HEATMAP OF NULL VALUES)

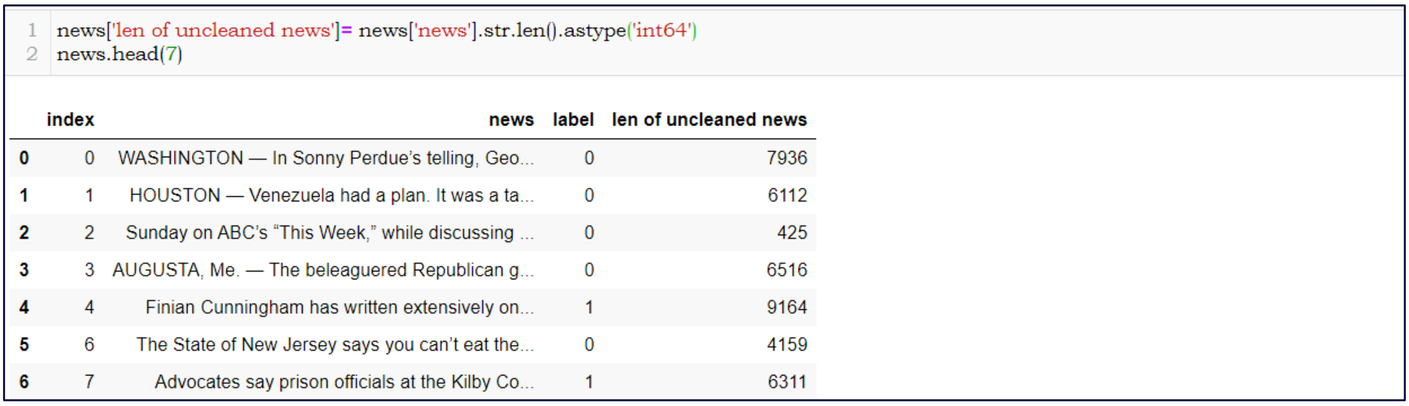
2.3 DATA PREPROCESSING

After the dataset is loaded and the shape, null values and duplicated values were checked then the data- set is further treated where the unwanted columns like **“***Unnamed: 0, id, headline, written_by***”** removed as we will work on the columns “*news, label”*. So a copy of the dataset was made using df.copy() and these columns were dropped from the new dataset created using df.drop(). As we have found there are numbers of null values and null values are not good for modelling. So these values were dropped using df.dropna().After dropping the null values the index was reset . **[Fig 5]**. After removing these null values and unwanted columns a new column representing the string length of the ‘news’ column is added to the dataset [**Fig 6**]. It’ll help to know the earlier length of the strings in ‘news’ columns and new length of those strings after processing. A calculation showed that total of 2515 null values were dropped and the shape of the dataset after removing unwanted columns and null values became (18285,4) [**Fig 7**].



```python
#dropping unwanted columns as we'll be working on the news and labe columns only
#dropping null values

news=df.copy()
news.dropna(inplace=True)
news.drop(['Unnamed: 0','id','headline','written_by'],axis=1,inplace=True)
news.reset_index(inplace=True)
```

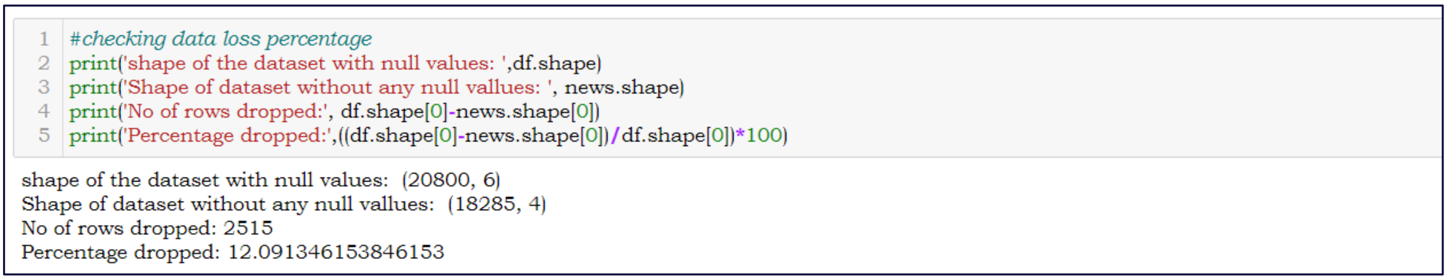
(Fig 5. DROPPING COLUMNS AND NULL VALUES)



```python
news['len of uncleaned news']= news['news'].str.len().astype('int64')
news.head(7)
```

	index	news	label	len of uncleaned news
0	0	WASHINGTON — In Sonny Perdue’s telling, Geo...	0	7936
1	1	HOUSTON — Venezuela had a plan. It was a ta...	0	6112
2	2	Sunday on ABC’s “This Week,” while discussing ...	0	425
3	3	AUGUSTA, Me. — The beleaguered Republican g...	0	6516
4	4	Finian Cunningham has written extensively on...	1	9164
5	6	The State of New Jersey says you can’t eat the...	0	4159
6	7	Advocates say prison officials at the Kilby Co...	1	6311

(Fig 6. NEW COLUMNS FOR STRING LENGTH)

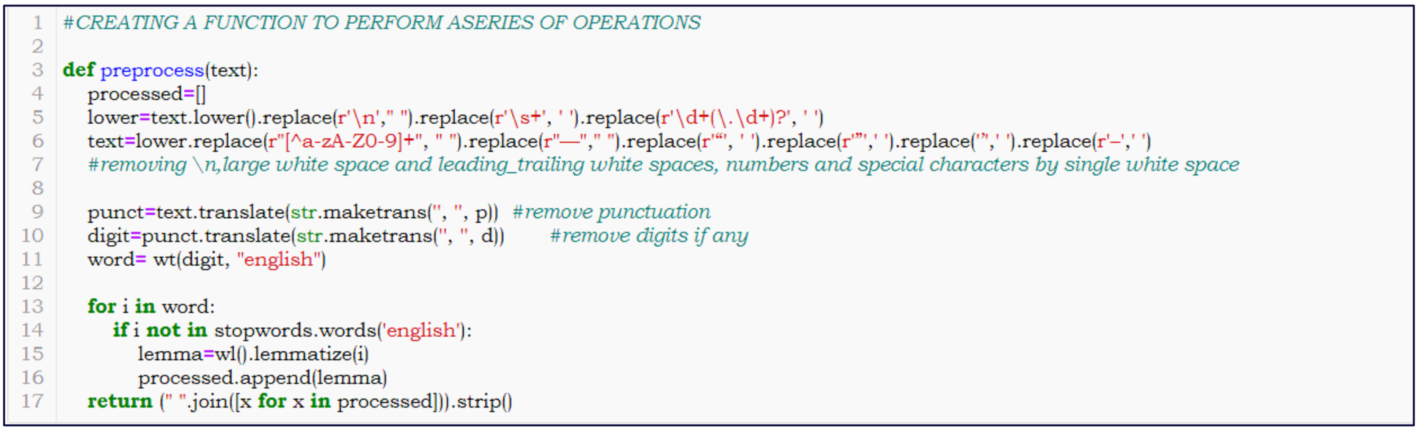


```
#checking data loss percentage
print('shape of the dataset with null values: ',df.shape)
print('Shape of dataset without any null vallues: ', news.shape)
print('No of rows dropped:', df.shape[0]-news.shape[0])
print('Percentage dropped:',((df.shape[0]-news.shape[0])/df.shape[0])*100)
```

```
shape of the dataset with null values:  (20800, 6)
Shape of dataset without any null vallues:  (18285, 4)
No of rows dropped: 2515
Percentage dropped: 12.091346153846153
```

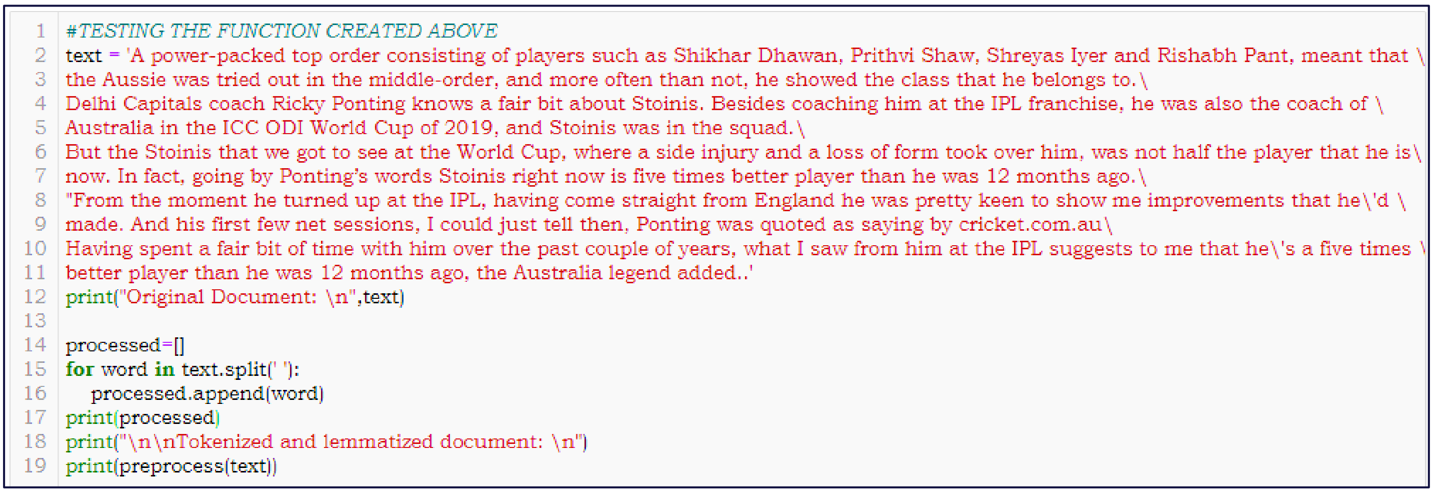
(Fig 7. LOSS & NEW SHAPE)

After the dataset is treated and freed from unwanted columns and null values the NLP techniques were implemented for processing the texts in the 'news' columns. In the preprocessing the string converted to lower case as it is easier to understand for the machine then from the string stopwords, special characters, digits were dropped using proper format. After those unnecessary characters were removed the string is tokenized using word_tokenization() of NLTK library those tokenized words were lemmatized using wordnetLemmatizer() which brings back all words to their root form with proper meaning. Then again, those tokenized words were joined to form a string. All these operations were compiled inside a function [**Fig 8**]. After the function was created a test run was done on a sample text to check the effectiveness of the function so created [**Fig 9-10**]. After successful testing the entire 'news' column was processed using the function created to get a clear and pure form of data for further operations [**Fig 11**].



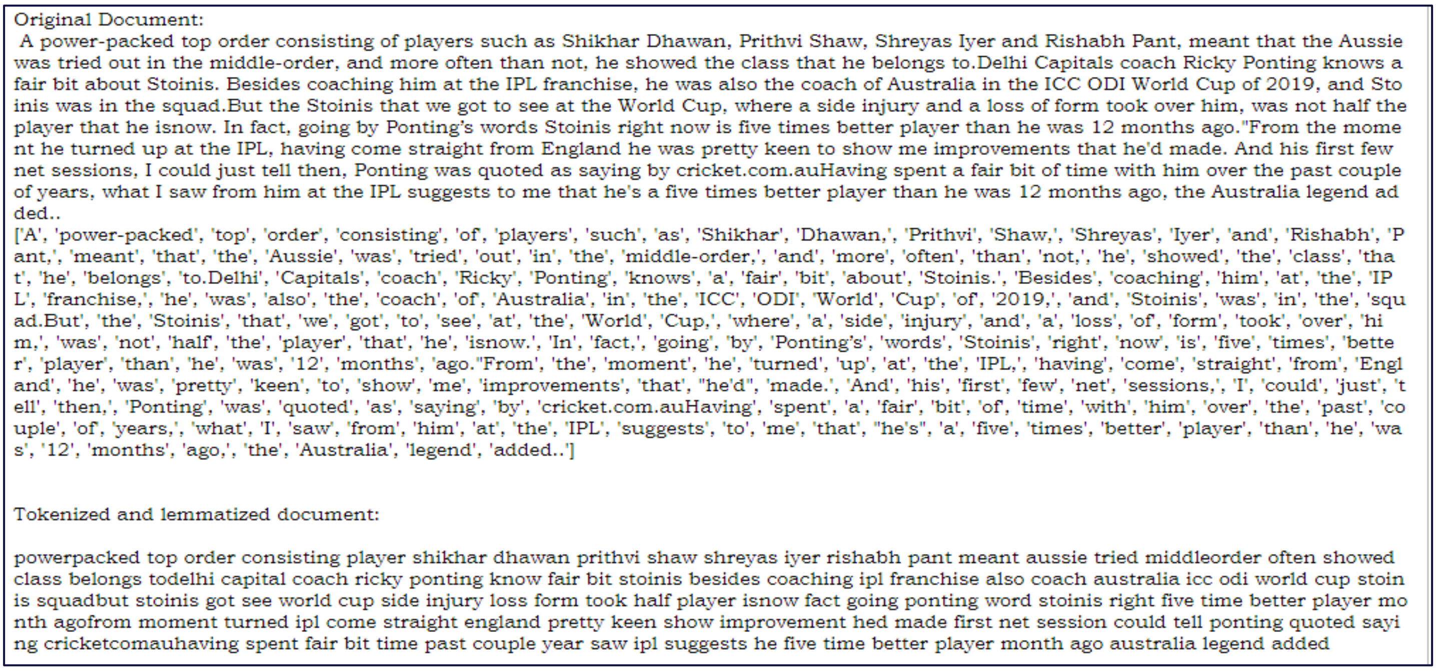
```
#CREATING A FUNCTION TO PERFORM ASERIES OF OPERATIONS

def preprocess(text):
    processed=[]
    lower=text.lower().replace(r'\n'," ").replace(r'\s+', ' ').replace(r'\d+(\.\d+)?', ' ')
    text=lower.replace(r"[^a-zA-Z0-9]+", " ").replace(r"—"," ").replace(r'“', ' ').replace(r'”',' ').replace('’',' ').replace(r'–',' ')
    #removing \n,large white space and leading_trailing white spaces, numbers and special characters by single white space

    punct=text.translate(str.maketrans('', '', p))  #remove punctuation
    digit=punct.translate(str.maketrans('', '', d))       #remove digits if any
    word= wt(digit, "english")

    for i in word:
        if i not in stopwords.words('english'):
            lemma=wl().lemmatize(i)
            processed.append(lemma)
    return (" ".join([x for x in processed])).strip()
```

(Fig 8. FUNCTION FOR PROCESSING DATA)



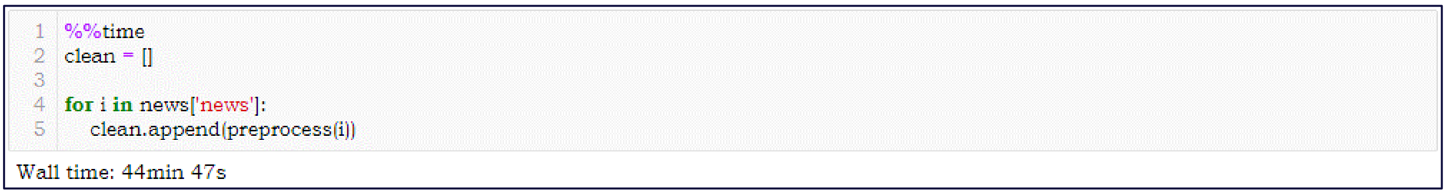
```
#TESTING THE FUNCTION CREATED ABOVE
text = 'A power-packed top order consisting of players such as Shikhar Dhawan, Prithvi Shaw, Shreyas Iyer and Rishabh Pant, meant that \
the Aussie was tried out in the middle-order, and more often than not, he showed the class that he belongs to.\
Delhi Capitals coach Ricky Ponting knows a fair bit about Stoinis. Besides coaching him at the IPL franchise, he was also the coach of \
Australia in the ICC ODI World Cup of 2019, and Stoinis was in the squad.\
But the Stoinis that we got to see at the World Cup, where a side injury and a loss of form took over him, was not half the player that he is\
now. In fact, going by Ponting's words Stoinis right now is five times better player than he was 12 months ago.\
"From the moment he turned up at the IPL, having come straight from England he was pretty keen to show me improvements that he\'d \
made. And his first few net sessions, I could just tell then, Ponting was quoted as saying by cricket.com.au\
Having spent a fair bit of time with him over the past couple of years, what I saw from him at the IPL suggests to me that he\'s a five times \
better player than he was 12 months ago, the Australia legend added..'
print("Original Document: \n",text)

processed=[]
for word in text.split(' '):
    processed.append(word)
print(processed)
print("\n\nTokenized and lemmatized document: \n")
print(preprocess(text))
```

(Fig 9. SAMPLE TESTING)



```
Original Document:
 A power-packed top order consisting of players such as Shikhar Dhawan, Prithvi Shaw, Shreyas Iyer and Rishabh Pant, meant that the Aussie
was tried out in the middle-order, and more often than not, he showed the class that he belongs to.Delhi Capitals coach Ricky Ponting knows a
fair bit about Stoinis. Besides coaching him at the IPL franchise, he was also the coach of Australia in the ICC ODI World Cup of 2019, and Sto
inis was in the squad.But the Stoinis that we got to see at the World Cup, where a side injury and a loss of form took over him, was not half the
player that he isnow. In fact, going by Ponting's words Stoinis right now is five times better player than he was 12 months ago."From the mome
nt he turned up at the IPL, having come straight from England he was pretty keen to show me improvements that he'd made. And his first few
net sessions, I could just tell then, Ponting was quoted as saying by cricket.com.auHaving spent a fair bit of time with him over the past couple
of years, what I saw from him at the IPL suggests to me that he's a five times better player than he was 12 months ago, the Australia legend ad
ded..
['A', 'power-packed', 'top', 'order', 'consisting', 'of', 'players', 'such', 'as', 'Shikhar', 'Dhawan,', 'Prithvi', 'Shaw,', 'Shreyas', 'Iyer', 'and', 'Rishabh', 'P
ant,', 'meant', 'that', 'the', 'Aussie', 'was', 'tried', 'out', 'in', 'the', 'middle-order,', 'and', 'more', 'often', 'than', 'not,', 'he', 'showed', 'the', 'class', 'tha
t', 'he', 'belongs', 'to.Delhi', 'Capitals', 'coach', 'Ricky', 'Ponting', 'knows', 'a', 'fair', 'bit', 'about', 'Stoinis.', 'Besides', 'coaching', 'him', 'at', 'the', 'IP
L', 'franchise,', 'he', 'was', 'also', 'the', 'coach', 'of', 'Australia', 'in', 'the', 'ICC', 'ODI', 'World', 'Cup', 'of', '2019,', 'and', 'Stoinis', 'was', 'in', 'the', 'squ
ad.But', 'the', 'Stoinis', 'that', 'we', 'got', 'to', 'see', 'at', 'the', 'World', 'Cup,', 'where', 'a', 'side', 'injury', 'and', 'a', 'loss', 'of', 'form', 'took', 'over', 'hi
m,', 'was', 'not', 'half', 'the', 'player', 'that', 'he', 'isnow.', 'In', 'fact,', 'going', 'by', 'Ponting's', 'words', 'Stoinis', 'right', 'now', 'is', 'five', 'times', 'bette
r', 'player', 'than', 'he', 'was', '12', 'months', 'ago."From', 'the', 'moment', 'he', 'turned', 'up', 'at', 'the', 'IPL,', 'having', 'come', 'straight', 'from', 'Engl
and', 'he', 'was', 'pretty', 'keen', 'to', 'show', 'me', 'improvements', 'that', "he'd", 'made.', 'And', 'his', 'first', 'few', 'net', 'sessions,', 'I', 'could', 'just', 't
ell', 'then,', 'Ponting', 'was', 'quoted', 'as', 'saying', 'by', 'cricket.com.auHaving', 'spent', 'a', 'fair', 'bit', 'of', 'time', 'with', 'him', 'over', 'the', 'past', 'co
uple', 'of', 'years,', 'what', 'I', 'saw', 'from', 'him', 'at', 'the', 'IPL', 'suggests', 'to', 'me', 'that', "he's", 'a', 'five', 'times', 'better', 'player', 'than', 'he', 'wa
s', '12', 'months', 'ago,', 'the', 'Australia', 'legend', 'added..']


Tokenized and lemmatized document:

powerpacked top order consisting player shikhar dhawan prithvi shaw shreyas iyer rishabh pant meant aussie tried middleorder often showed
class belongs todelhi capital coach ricky ponting know fair bit stoinis besides coaching ipl franchise also coach australia icc odi world cup stoin
is squadbut stoinis got see world cup side injury loss form took half player isnow fact going ponting word stoinis right five time better player mo
nth agofrom moment turned ipl come straight england pretty keen show improvement hed made first net session could tell ponting quoted sayi
ng cricketcomauhaving spent fair bit time past couple year saw ipl suggests he five time better player month ago australia legend added
```

(Fig 10. TEST RESULTS)

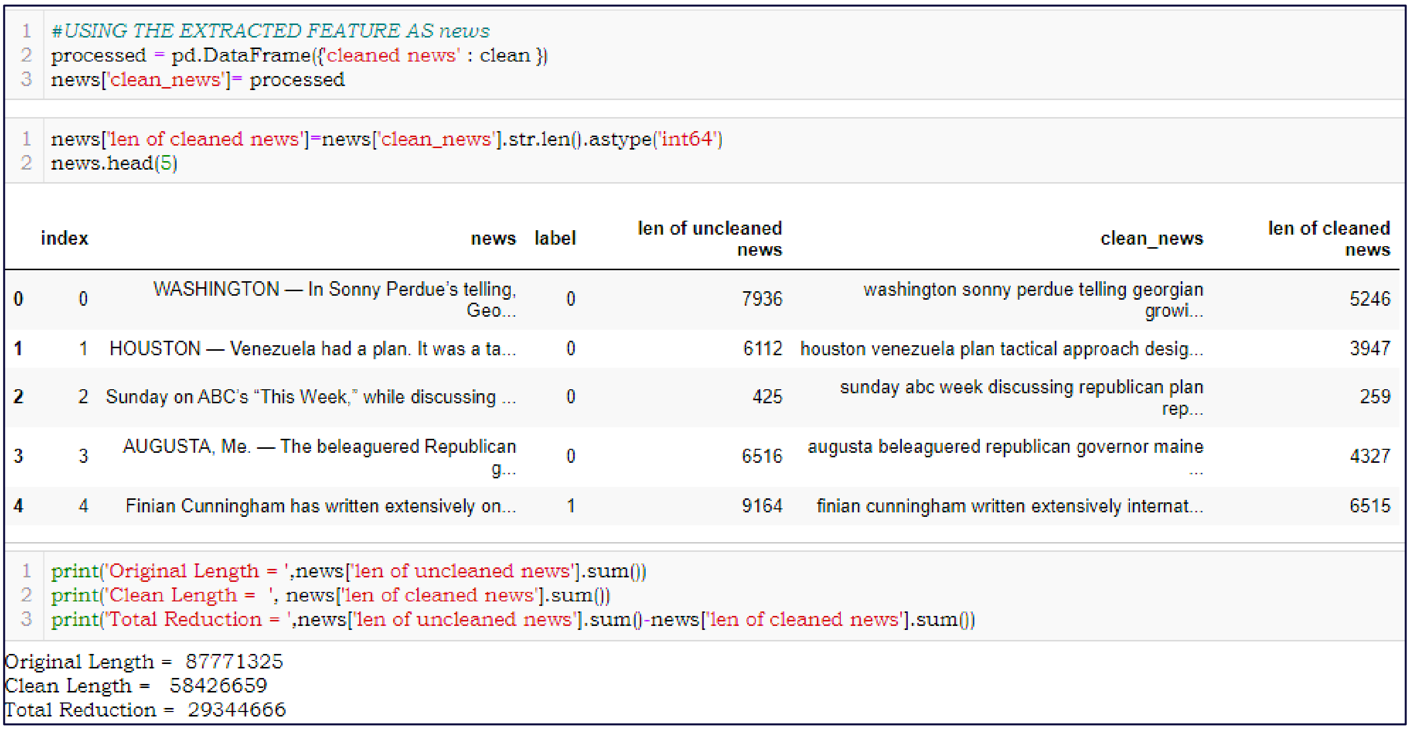


```
%%time
clean = []

for i in news['news']:
    clean.append(preprocess(i))
```

```
Wall time: 44min 47s
```

(Fig 11. FUNCTION IMPLEMENTATION)

After the procedure was completed a list of cleaned data were obtained which was added to the dataset by column name 'clean news' and another column name 'len of clean news' was added showing the length of words in the 'len of clean news' column. Further calculation revelled that there were total of 87771325 words were present in the news column after processing it became 58426659 [**Fig 12**].



```
#USING THE EXTRACTED FEATURE AS news
processed = pd.DataFrame({'cleaned news' : clean })
news['clean_news']= processed
```

```
news['len of cleaned news']=news['clean_news'].str.len().astype('int64')
news.head(5)
```

	index	news	label	len of uncleaned news	clean_news	len of cleaned news
0	0	WASHINGTON — In Sonny Perdue's telling, Geo...	0	7936	washington sonny perdue telling georgian growi...	5246
1	1	HOUSTON — Venezuela had a plan. It was a ta...	0	6112	houston venezuela plan tactical approach desig...	3947
2	2	Sunday on ABC's "This Week," while discussing ...	0	425	sunday abc week discussing republican plan rep...	259
3	3	AUGUSTA, Me. — The beleaguered Republican g...	0	6516	augusta beleaguered republican governor maine ...	4327
4	4	Finian Cunningham has written extensively on...	1	9164	finian cunningham written extensively internat...	6515

```
print('Original Length = ',news['len of uncleaned news'].sum())
print('Clean Length =  ', news['len of cleaned news'].sum())
print('Total Reduction = ',news['len of uncleaned news'].sum()-news['len of cleaned news'].sum())
```

```
Original Length =  87771325
Clean Length =   58426659
Total Reduction =  29344666
```

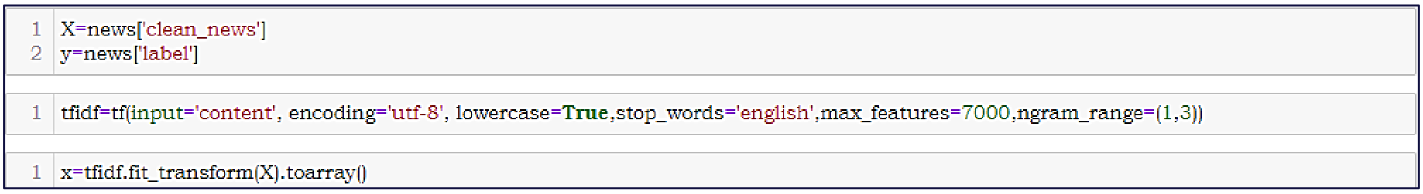
(Fig 12. FUNCTION RESULT)

After getting a cleaned data TF-IDF vectorizer will be used. It'll help to transform the text data to feature vector which can be used as input in our modelling. The TFIDF stands for Term Frequency Inverse Document Frequency. It is a common algorithm to transform text into vectors or numbers. It measures the originality of a word by comparing the frequency of appearance of a word in a document with the number of documents the words appear in.

Mathematically,

TF-IDF =TF(t*d) * IDF (t,d)

So here the dataset is divided into two parts X and Y. X represents the column 'clean_news' and Y represents the column 'label'. After the splitting the tfidf vectorizer was initialized and X is fitted into it and converted into array form. [**Fig 13**].



```
X=news['clean_news']
y=news['label']
```

```
tfidf=tf(input='content', encoding='utf-8', lowercase=True,stop_words='english',max_features=7000,ngram_range=(1,3))
```

```
x=tfidf.fit_transform(X).toarray()
```

(Fig 13. TF-IDF)

2.4 HARDWARE & TOOL USED

In this project the below mentioned Hardware, IDE, Language, Packages were used.

HARDWARE	LAPTOP: ASUS TUF A17 OS: WIN 10 HOME BASIC PROCESSOR: AMD RYZEN 7 4800H RAM: 16GB VRAM: 6GB NVIDIA GTX 1660Ti
LANGUAGE	Python 3.8
IDE	JUPYTER NOTEBOOK 6.0.3
PACKAGES	PANDAS, NLTK, SKLEARN, MATPLOTLIB, SEABORN

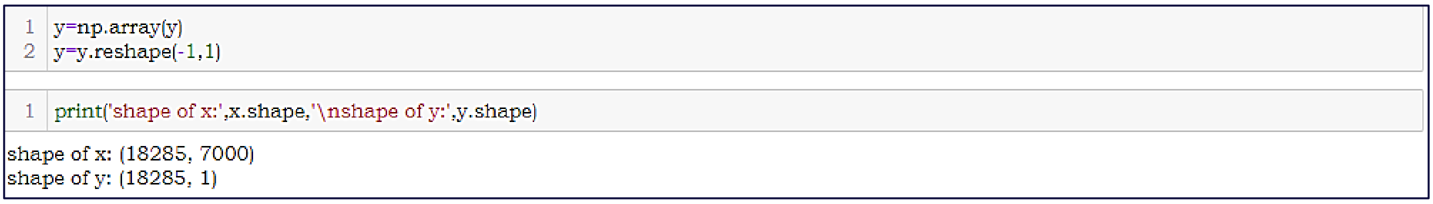
(Table 2: HARDWARE & TOOLS)

CHAPTER-3

DEVELOPMENT AND EVALUTION

3.1 IDENTIFACTION OF POSSIBLE PROBLEM-SOLVING APPROACHES

After TF-IDF implementation array conversion we have x and y for modelling. The shape of x found to be (18285,7000) and y was converted into an array and reshaped and the shape found to be (18285,) [**Fig 14].**



```
y=np.array(y)
y=y.reshape(-1,1)
```

```
print('shape of x:',x.shape,'\nshape of y:',y.shape)
```

```
shape of x: (18285, 7000)
shape of y: (18285, 1)
```

(Fig 14. X & Y PREPARATION)

3.2 TESTING OF IDENTIFIED ALGORITHMS

Here the output column 'label' is generating binary output so it’s a classification-based problem and we can use the following algorithms for modelling and the highest performing algorithms to get our final model;

- Logistic Regression()
- Multinomial NB()
- Complement NB()
- Random Forest Classifier()

With the help of GridSearchCV hyper parameter tuning will be done and the best parameters for each model will be found. During modelling various metrices like f1 score, confusion matrix, accuracy score, classification report, roc curve, roc auc score, mean squared error, precision score, recall score, log loss will be used to determine the performance of the model.

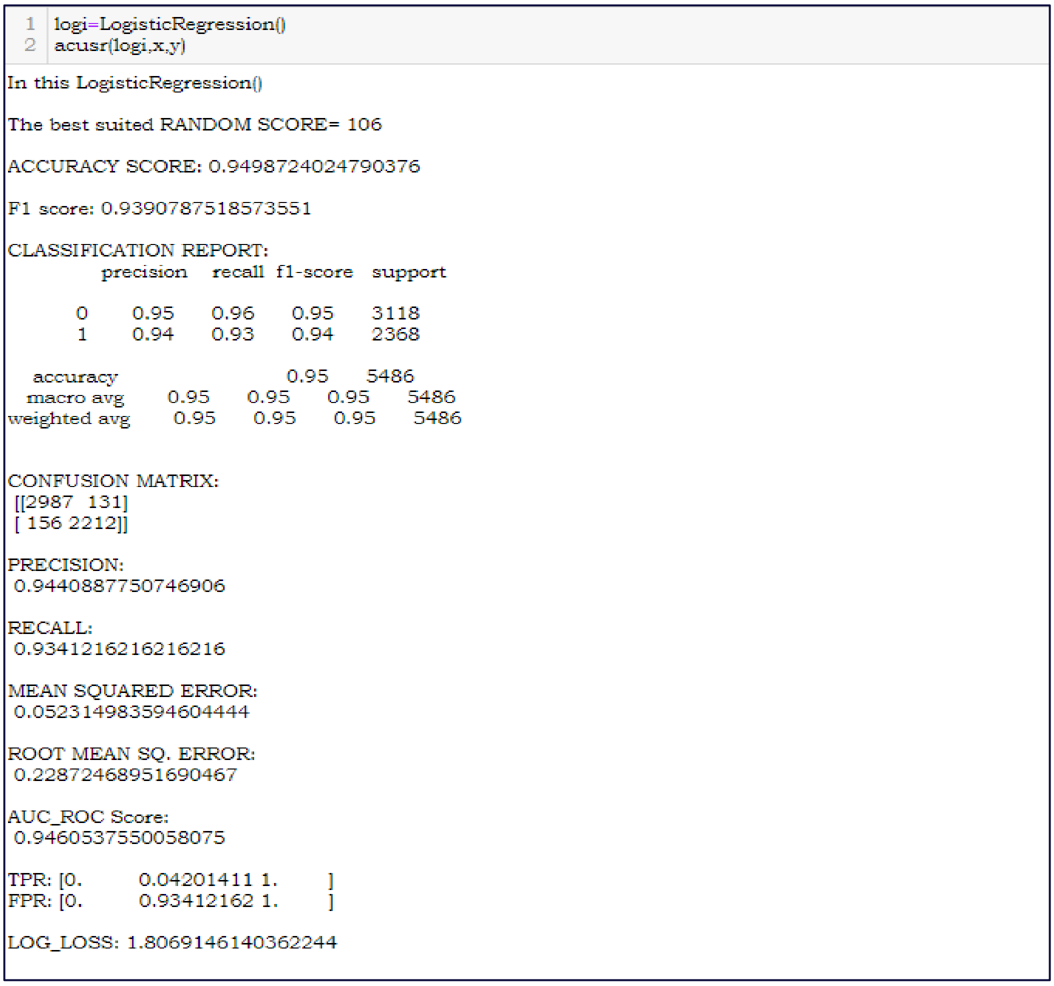
To check whether the model suffering from over fitting or underfitting cross val score will be used where ‘f1’ will be used as scorer. To view the best performing model AUC Curve will be used and heatmap will be used to visualize the confusion matrix. All the result will save to a DataFrame and at the end the best model will be saved using Joblib library.

3.3 TESTING OF IDENTIFIED ALGORITHMS

Here I have created a number of list named as F1, ACCURACY, PRECISION, RECALL, RMSE, MSE, AUC, TPR, FPR, CV_ACC, LOG_LOSS to hold the values of matrices like f1 scores, accuracy scores, precision values, recall values, root mean squared error values, mePage | 18an squared error values, auc scores, tpr values, fpr values, cross validation with f1 values, log loss values respectively.

Here I have generated a function which will find the best random score for the model in a range of 25 to 180. It’ll also show the confusion matrix, accuracy score, classification report, roc curve, auc, roc auc score, mean squared error, precision score, recall score, tpr, fpr, f1 score, log loss value for an algorithm at the best random score. The values obtained will be added to their respective lists.

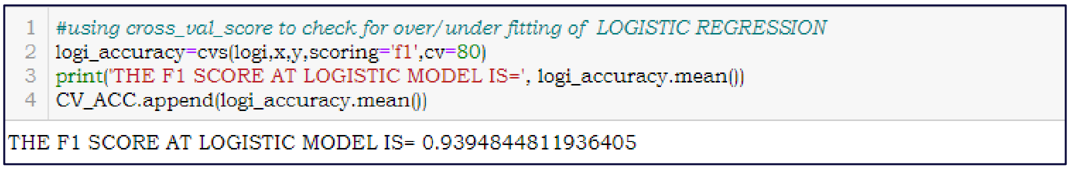
Below are few images of function performed on algorithms. Showing the metrics, heatmap of confusion matrix, AUC ROC curve, Cross validated values.



```
logi=LogisticRegression()
acusr(logi,x,y)
```

```
In this LogisticRegression()

The best suited RANDOM SCORE= 106

ACCURACY SCORE: 0.9498724024790376

F1 score: 0.9390787518573551

CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.95      0.96      0.95      3118
           1       0.94      0.93      0.94      2368

    accuracy                           0.95      5486
   macro avg       0.95      0.95      0.95      5486
weighted avg       0.95      0.95      0.95      5486


CONFUSION MATRIX:
 [[2987  131]
 [ 156 2212]]

PRECISION:
 0.9440887750746906

RECALL:
 0.9341216216216216

MEAN SQUARED ERROR:
 0.05231498359460444

ROOT MEAN SQ. ERROR:
 0.22872468951690467

AUC_ROC Score:
 0.9460537550058075

TPR: [0.         0.04201411 1.        ]
FPR: [0.         0.93412162 1.        ]

LOG_LOSS: 1.8069146140362244
```

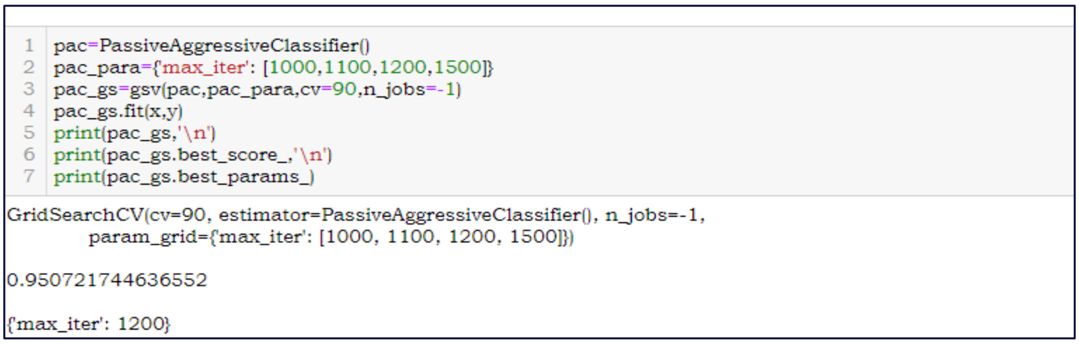
(Fig 15. LOGISTIC REGRESSION)



```
#using cross_val_score to check for over/under fitting of  LOGISTIC REGRESSION
logi_accuracy=cvs(logi,x,y,scoring='f1',cv=80)
print('THE F1 SCORE AT LOGISTIC MODEL IS=', logi_accuracy.mean())
CV_ACC.append(logi_accuracy.mean())
```

```
THE F1 SCORE AT LOGISTIC MODEL IS= 0.9394844811936405
```

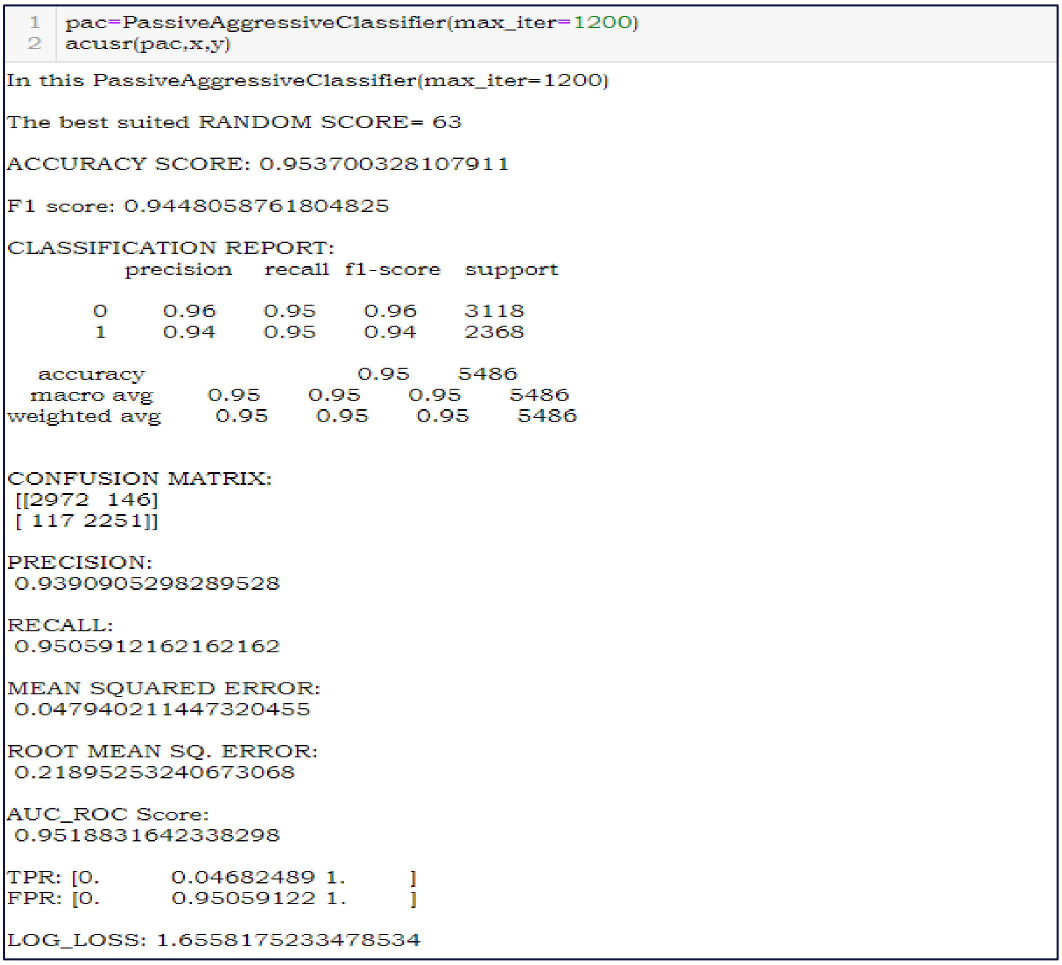
(Fig 16. CROSS VALIDATION OF LOGISTIC REGRESSION)



```
pac=PassiveAggressiveClassifier()
pac_para={'max_iter': [1000,1100,1200,1500]}
pac_gs=gsv(pac,pac_para,cv=90,n_jobs=-1)
pac_gs.fit(x,y)
print(pac_gs,'\n')
print(pac_gs.best_score_,'\n')
print(pac_gs.best_params_)
```

```
GridSearchCV(cv=90, estimator=PassiveAggressiveClassifier(), n_jobs=-1,
             param_grid={'max_iter': [1000, 1100, 1200, 1500]})

0.950721744636552

{'max_iter': 1200}
```

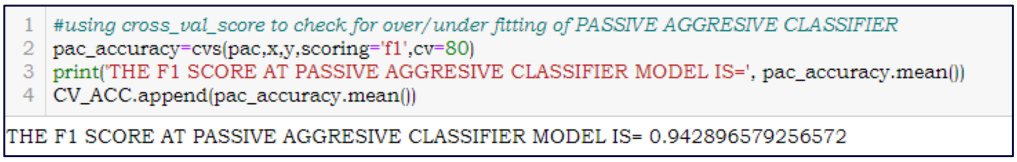
(Fig 17. HYPERPARAMETER TUNING OF PASSIVE AGGRESSIVE CLASSIFIER)



```
pac=PassiveAggressiveClassifier(max_iter=1200)
acusr(pac,x,y)
```

```
In this PassiveAggressiveClassifier(max_iter=1200)

The best suited RANDOM SCORE= 63

ACCURACY SCORE: 0.953700328107911

F1 score: 0.9448058761804825

CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.96      0.95      0.96      3118
           1       0.94      0.95      0.94      2368

    accuracy                           0.95      5486
   macro avg       0.95      0.95      0.95      5486
weighted avg       0.95      0.95      0.95      5486


CONFUSION MATRIX:
 [[2972  146]
 [ 117 2251]]

PRECISION:
 0.9390905298289528

RECALL:
 0.9505912162162162

MEAN SQUARED ERROR:
 0.04794021144732O455

ROOT MEAN SQ. ERROR:
 0.21895253240673068

AUC_ROC Score:
 0.951883164233829 8

TPR: [0.         0.04682489 1.        ]
FPR: [0.         0.95059122 1.        ]

LOG_LOSS: 1.6558175233478534
```

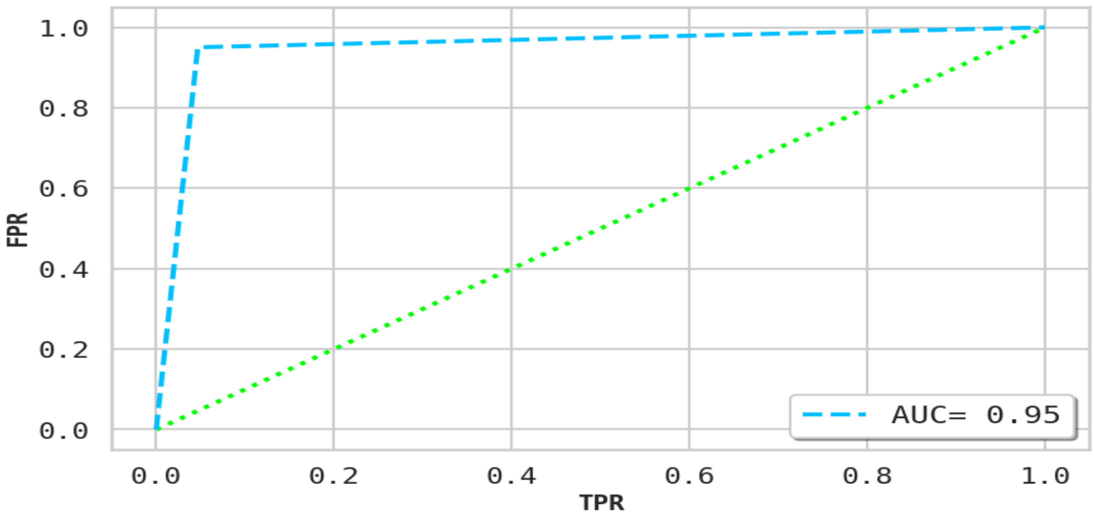
(Fig 18. PASSIVE AGGRESSIVE CLASSIFIER)



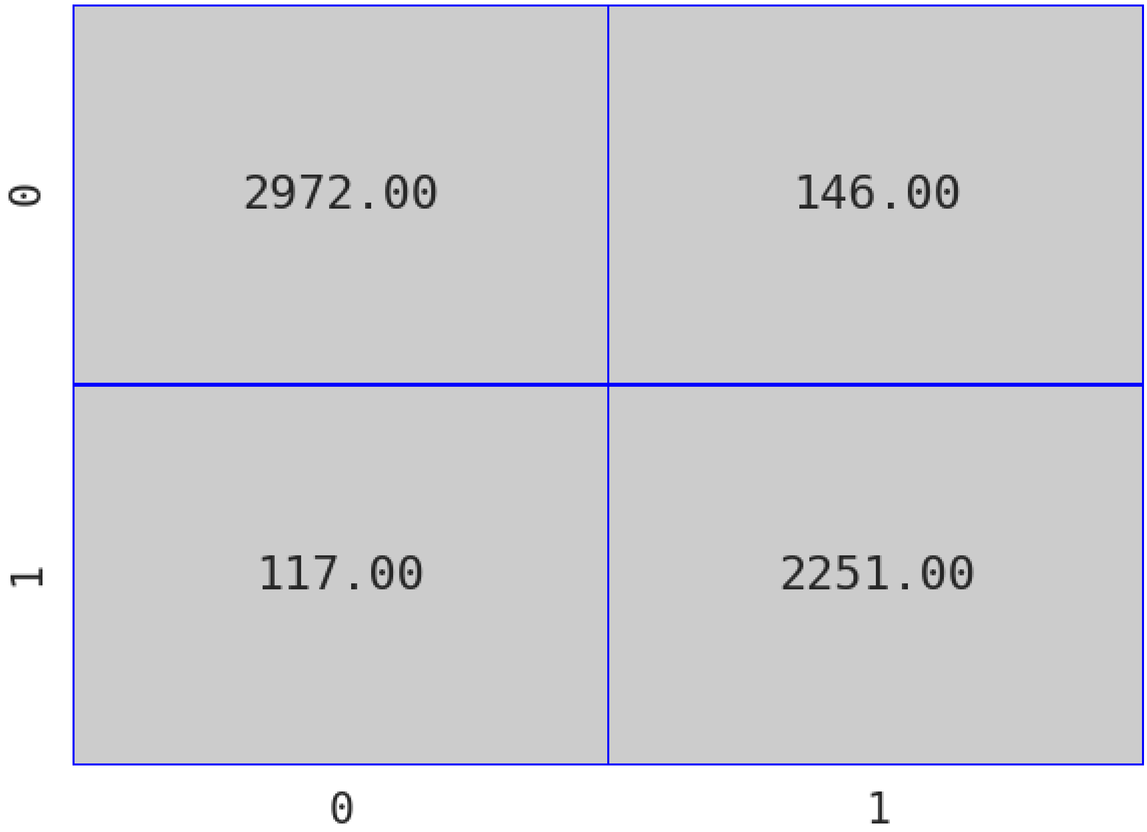
```
#using cross_val_score to check for over/under fitting of PASSIVE AGGRESIVE CLASSIFIER
pac_accuracy=cvs(pac,x,y,scoring='f1',cv=80)
print('THE F1 SCORE AT PASSIVE AGGRESIVE CLASSIFIER MODEL IS=', pac_accuracy.mean())
CV_ACC.append(pac_accuracy.mean())
```

```
THE F1 SCORE AT PASSIVE AGGRESIVE CLASSIFIER MODEL IS= 0.942896579256572
```

(Fig 19. CROSS VALIDATION OF PASSIVE AGGRESSIVE CLASSIFIER)



(Fig 20. AUC CURVE OF PASSIVE AGGRESSIVE CLASSIFIER)



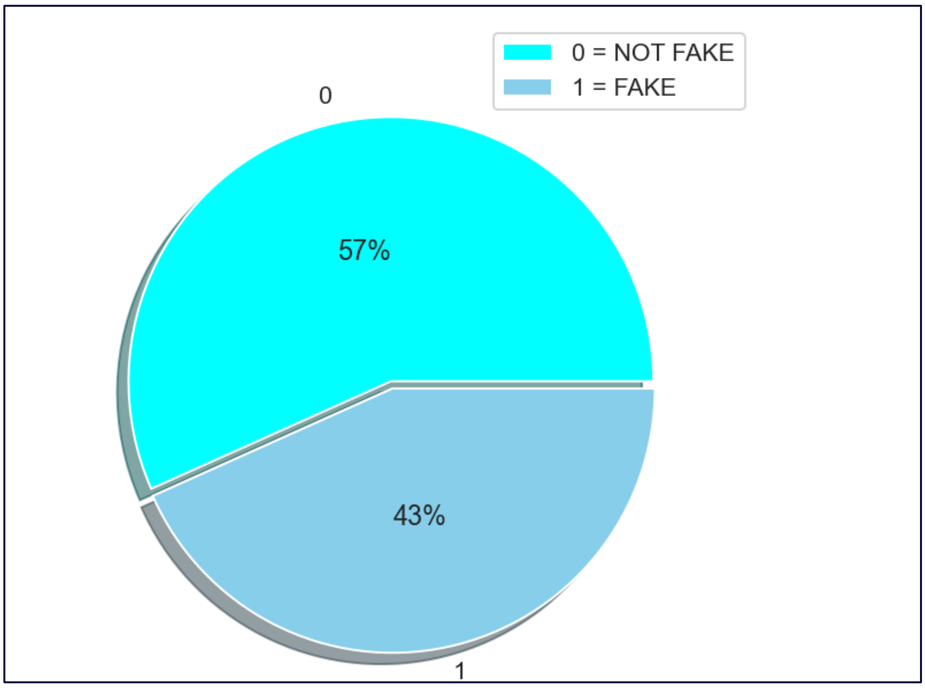
(Fig 21. HEATMAP OF CONFUSION MATRIX OF PASSIVE AGGRESSIVE CLASSIFIER)

3.4 METRICE OF EVALUATION

In the modeling I have chosen metrices like F1 score, Accuracy score, Precision, Recall, Mean Squared Error, Root Mean Square Error, Classification Report, Confusion Matrix, AUC, tpr, fpr and Cross val Score with f1 as scorer as my evaluation criteria. All the values were stored in a list and later they were saved in form of a DataFrame for proper evaluation and visualization of the values. Basing on the values the best model has been selected [**Fig 21**].

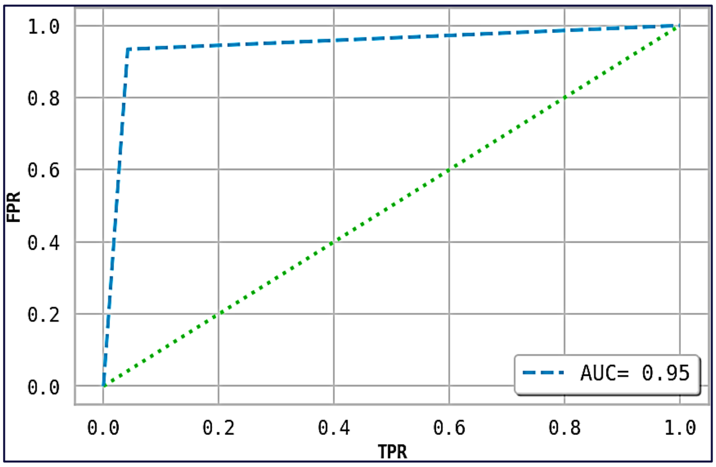
3.5 VISUALIZATION

Visualization plays a crucial role in EDA as well as during modelling. It gives a better idea about the things going on beautifully. Below are the few visualizations used during this project to understand the dataset and performance of the algorithms.

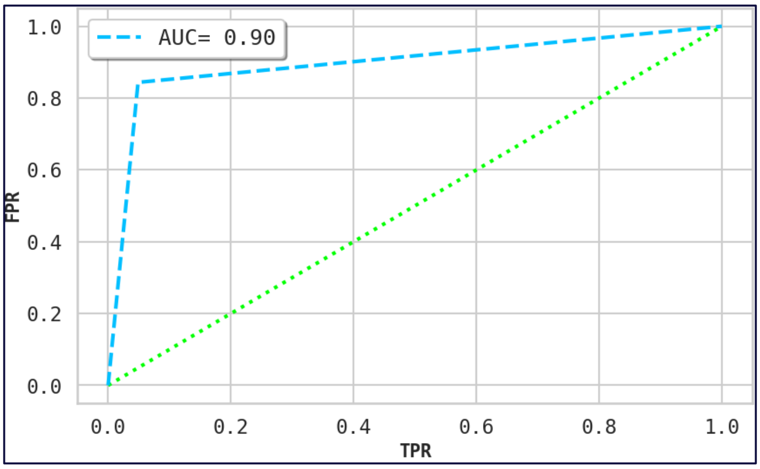


(Fig 22. PIE PLOT OF LABEL)

AUC ROC CURVE – LOGISTIC REGRESSION

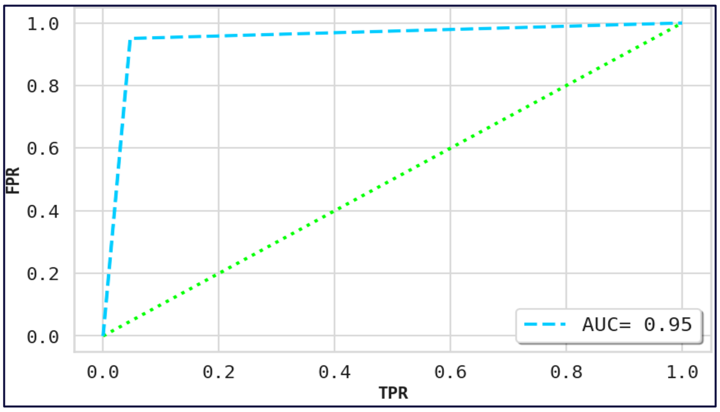


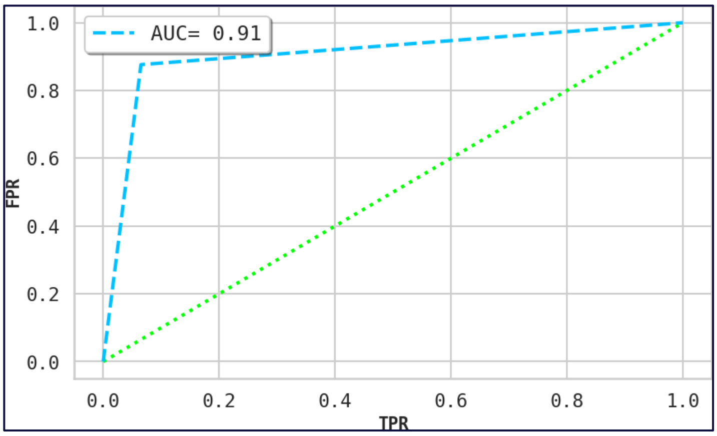
AUC ROC CURVE – MULTINOMIAL NB



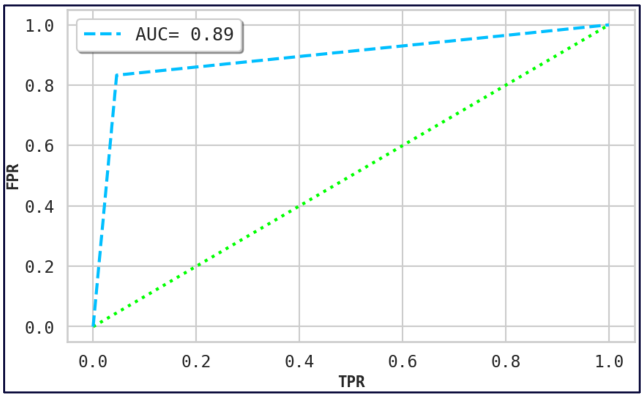
AUC ROC CURVE – PAC

AUC ROC CURVE – COMPLEMENT NB





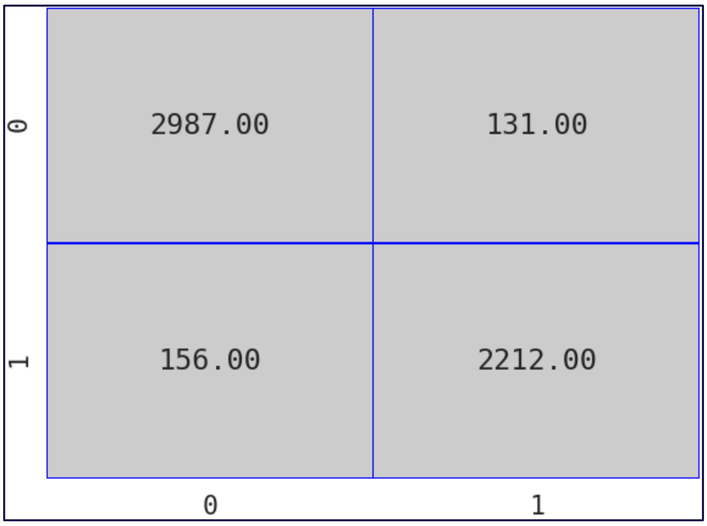
AUC ROC CURVE – RFC



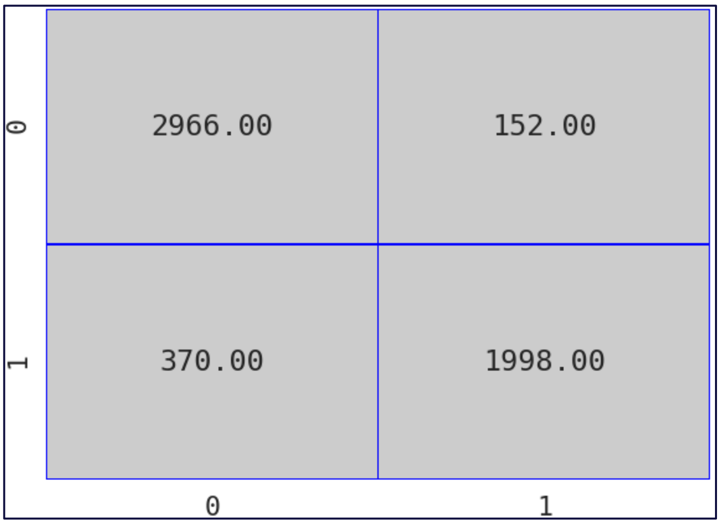
(Fig 23. AUC_ROC CURVES)

CONFUSION MATRIX – LOGISTIC REG

CONFUSION MATRIX – MULTINOMIAL NB



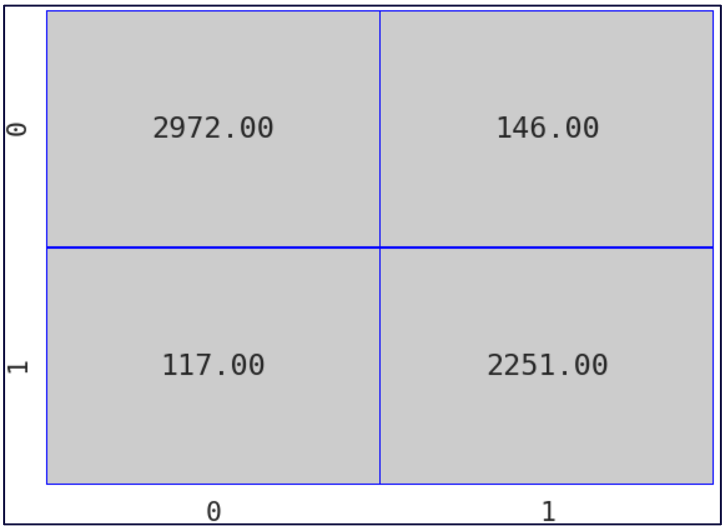
	0	1
0	2987.00	131.00
1	156.00	2212.00



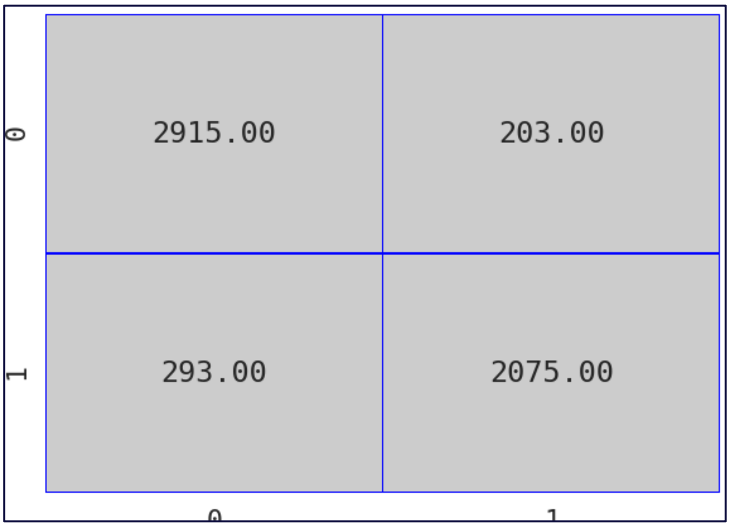
	0	1
0	2966.00	152.00
1	370.00	1998.00

CONFUSION MATRIX – PAC

CONFUSION MATRIX – COMPLEMENT NB

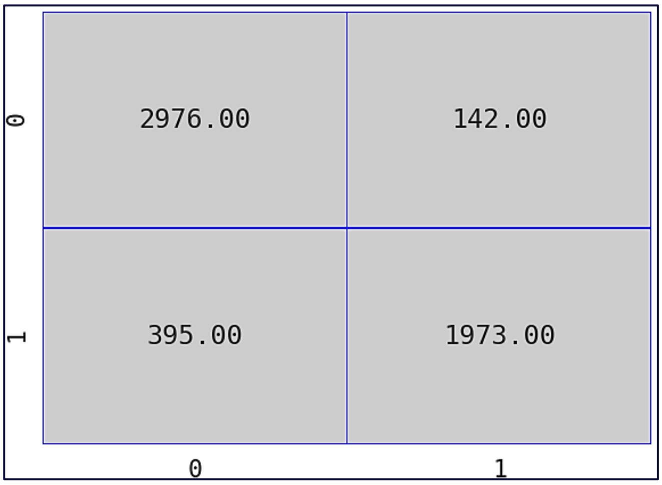


	0	1
0	2972.00	146.00
1	117.00	2251.00



	0	1
0	2915.00	203.00
1	293.00	2075.00

CONFUSION MATRIX – RFC

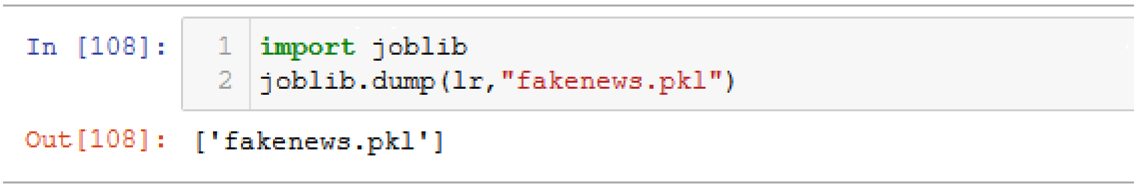


	0	1
0	2976.00	142.00
1	395.00	1973.00

(Fig 24. HEATMAP OF CONFUSION MATRIXS)

3.6 INTERPRETATION

Basing on the result obtained ‘LOGISTIC REGRESSION’ have performed well and has given better result as compared to other models. So LOGISTIC REGRESSION has been selected as final model and it will be saved using joblib library.



```
In [108]:  import joblib
           joblib.dump(lr,"fakenews.pkl")

Out[108]: ['fakenews.pkl']
```

(**Fig 25. SAVING MODEL**)

CHAPTER-4

CONCLUSION

4.1 KEY FINDINGS

From the above analysis the below mentioned results were achieved which depicts the chances and conditions of a news being a fake;

- With the increasing popularity of social media, more and more people consume news from social media instead of traditional news media. However, social media has also been used to spread fake news, which has strong negative impacts on individual users and broader society.

4.2 LEARNING OUTCOMES

- It is possible to classify news content into the required categories of authentic and fake news however there will be always a bias to this kind of classification which depends on the behavioural pattern of the listener. However, using this kind of project an awareness can be created to know what is fake and authentic.

4.3 LIMITATION AND SCOPE OF WORK

Nothing is perfect and this project is of no exception. There are certain areas which can be enhanced. Fake news detection is an emerging research area with few public datasets. So a lot of works need to be done on this field.