

FAKE

NEWS

DETECTION

**Submitted By:**

Ruchi Tiwari

Contents

[**ACKNOWLEDGMENT 3**](#_Toc65157879)

[**INTRODUCTION 4**](#_Toc65157880)

[Business Problem Framing 4](#_Toc65157881)

[Conceptual Background of the Domain Problem 4](#_Toc65157882)

[Review of Literature 4](#_Toc65157883)

[Motivation for the Problem Undertaken 4](#_Toc65157884)

[**Analytical Problem Framing 4**](#_Toc65157885)

[Data Sources and their formats 4](#_Toc65157886)

[Data Preprocessing Done 5](#_Toc65157887)

[Hardware and Software Requirements and Tools Used 6](#_Toc65157888)

[**Model/s Development and Evaluation 7**](#_Toc65157889)

[Testing of Identified Approaches (Algorithms) 7](#_Toc65157890)

[Run and Evaluate selected models 7](#_Toc65157891)

[Key Metrics for success in solving problem under consideration 8](#_Toc65157892)

[Visualizations 9](#_Toc65157893)

[Interpretation of the Results 11](#_Toc65157894)

[**CONCLUSION 12**](#_Toc65157895)

[Key Findings and Conclusions of the Study 12](#_Toc65157896)

[Learning Outcomes of the Study in respect of Data Science 12](#_Toc65157897)

# ACKNOWLEDGMENT

Following are the external references which I used:

[www.w3school.com](http://www.w3school.com)

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.google.com](http://www.google.com)

[www.geeksforgeeks.org](http://www.geeksforgeeks.org)

# INTRODUCTION

## Business Problem Framing

In the news industry, in particular, but also in society at large, fake news detection has become a central discussion topic, as the need to permanently assess the veracity of digital content has been raised by the constant spread of false news / information. Information veracity is a long-term issue affecting society both for printed and digital media. The sensationalism of not-so-accurate eye catching and intriguing headlines aimed at retaining the attention of audiences to sell information has persisted all throughout the history of all kinds of information broadcast. On social networking websites, the reach and effects of information spread are however significantly amplified and occur at such a fast pace, that distorted, inaccurate or false information acquires a tremendous potential to cause real impacts, within minutes, for millions of users

## Conceptual Background of the Domain Problem

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.

## Review of Literature

From the dataset I get to know that it is a classification problem and there are two categories which are fake news and not fake news.

## Motivation for the Problem Undertaken

As this project is based on NLP and there are so many new libraries which are new for me so it is really motivates me to learn and understand the concept and do the coding on some new things which are totally new for me.

# Analytical Problem Framing

## Data Sources and their formats

* There are 6 columns in the dataset provided to you. The description of each of the column is given below:
* “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).

## Data Preprocessing Done

Here is a checklist to use the clean data Remove all irrelevant characters such as any non alphanumeric characters

1. Tokenize your text by separating it into individual words
2. Remove words that are not relevant,
3. Convert all characters to lowercase, in order to treat words such as “hello”, “Hello”, and “HELLO” the same
4. Consider combining misspelled or alternately spelled words to a single representation (e.g. “cool”/”kewl”/”cooool”)
5. Consider lemmatization (reduce words such as “am”, “are”, and “is” to a common form such as “be”)

Firstly I read the data and see the first five rows and the last five rows after that I check the null values and see that in three columns which are headline, written\_by, news are having the null values. Then I replace the null value with the spaces and again recheck it. Now in the dataset we don’t have null values for the preprocessing /cleaning the data I perform the stop words removal, Tokenization, Lemmatization .The Stop words are those common words that appear in a text many times and do not contribute to machine’s understanding of the text. We don’t want these words to appear in our data. So, we remove these words. Tokenization: Word tokenization is the process of splitting a large sample of text into words. Lemmatization: Lemmatization is the process of grouping together the different inflected forms of same root word so they can be analyzed as a single item. Then I convert all the text words in the lower case since python is a case sensitive language .Then I used regex for cleaning the sentence. Then I print the loud words of fake news and a not fake news Count Vectorizer In order to use textual data for predictive modelling, the text must be parsed to remove certain words — this process is called tokenization. These words need to then be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms. This process is called feature extraction (or vectorization).

TF-IDF Vectorizer TF-IDF stands for Term Frequency — Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. These are the things which I performed in the Preprocessing.

## Hardware and Software Requirements and Tools Used

**Hardware** – Laptop

**Software** - anaconda jupyter notebook

**Libraries**- numpy, pandas, seaborn, matplotlib.pyplot, warning

#Pre-processing/ Cleaning the Data

#For preprocessing the data, we will need some libraries.

import nltk

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

import re

nltk.corpus.stopwords

nltk.corpus.util.LazyCorpusLoader

from wordcloud import WordCloud

from sklearn.feature\_extraction.text import TfidfTransformer

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

**from sklearn.model\_selection import train\_test\_split,cross\_val\_score**

Train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

The algorithm is trained and tested K times, each time a new set is used as testing set while remaining sets are used for training. Finally, the result of the K-Fold Cross-Validation is the average of the results obtained on each set.

**from sklearn.neighbors import KNeighborsClassifier**

K Nearest Neighbor(KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition

**from sklearn.linear\_model import LogisticRegression**

The library sklearn can be used to perform logistic regression in a few lines as shown using the LogisticRegression class. It also supports multiple features. It requires the input values to be in a specific format hence they have been reshaped before training using the fit method.

**from sklearn.tree import DecisionTreeClassifier**

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy

**from sklearn.naive\_bayes import GaussianNB**

Naive Bayes are a group of supervised machine learning classification algorithms based on the Bayes theorem. It is a simple classification technique, but has high functionality.

# Model/s Development and Evaluation

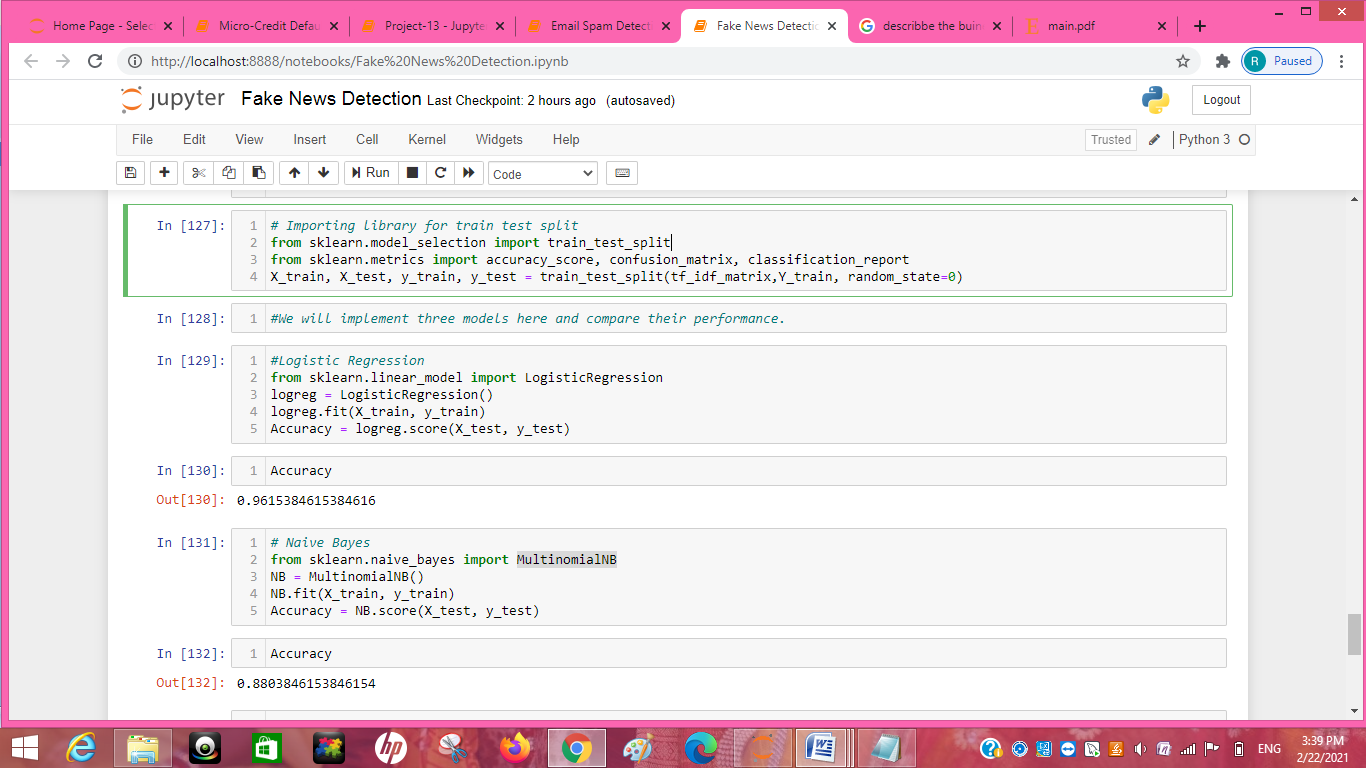
## Testing of Identified Approaches (Algorithms)

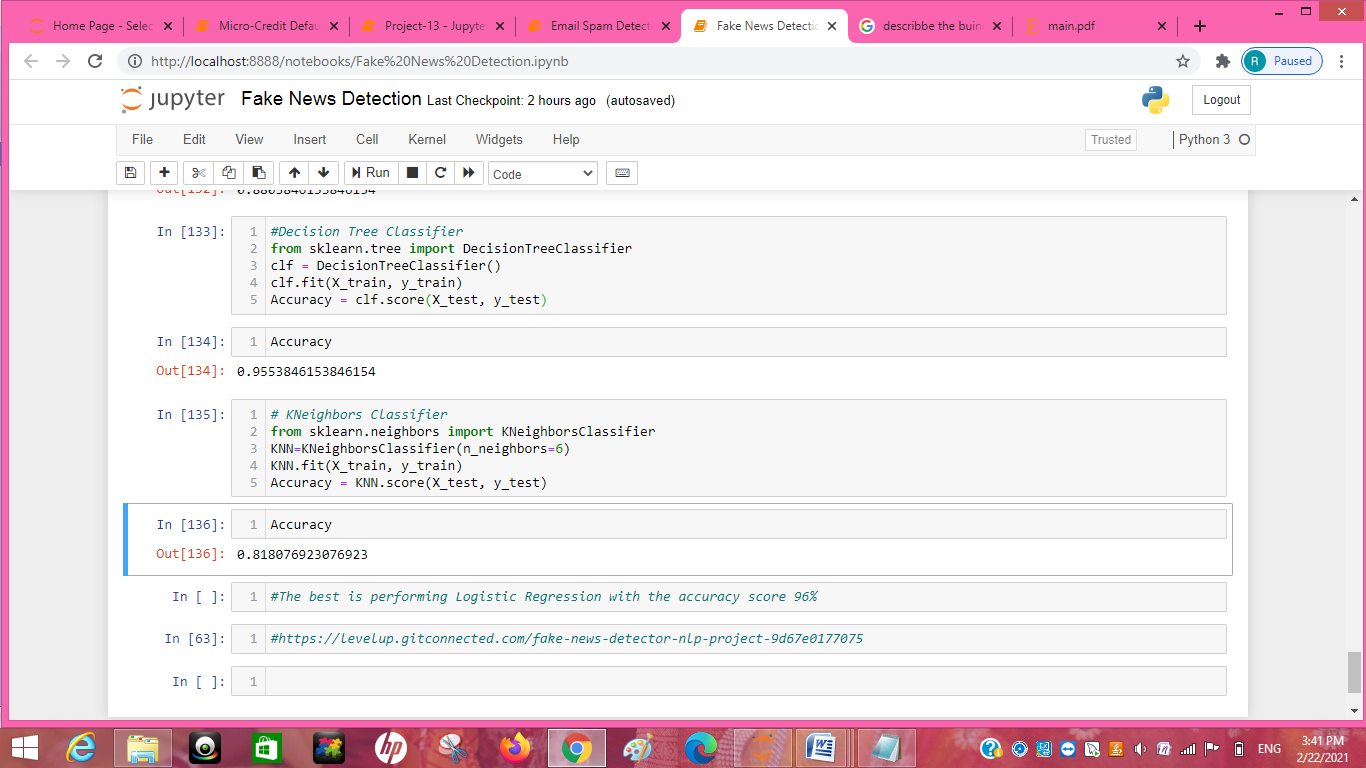
Listing down all the algorithms used for the training and testing.

* KNN=KNeighborsClassifier(n\_neighbors=6)
* LR=LogisticRegression()
* DT=DecisionTreeClassifier(random\_state=6)
* NB= MultinomialNB

I applied all these algorithms in the dataset.

## Run and Evaluate selected models





## Key Metrics for success in solving problem under consideration

Precision: can be seen as a measure of quality, **higher** **precision** means that an algorithm returns more relevant results than irrelevant ones

**Recall** is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.

**Accuracy score** is used when the True Positives and True negatives are more important. **Accuracy** can be used when the class distribution is similar

**F1**-**score** is used when the False Negatives and False Positives are crucial. While F1-score is a better metric when there are imbalanced classes.

**Cross\_val\_score** :- To run **cross**-**validation** on multiple metrics and also to return train **scores**, fit times and **score** times. Get predictions from each split of **cross**-**validation** for diagnostic purposes. Make a scorer from a performance metric or loss function.

roc \_auc \_score :-  **ROC curve**. It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0

## Visualizations

***sns.countplot(x='label',data=train)***

***plt.xlabel('label (0=not fake, 1=fake)')***

***plt.show()***



#Getting sense of loud words in Fake

from wordcloud import WordCloud

fake = train['total'][train['label']==1]

fake\_cloud = WordCloud(width=700,height=500,background\_color='white',max\_words=20).generate(' '.join(fake))

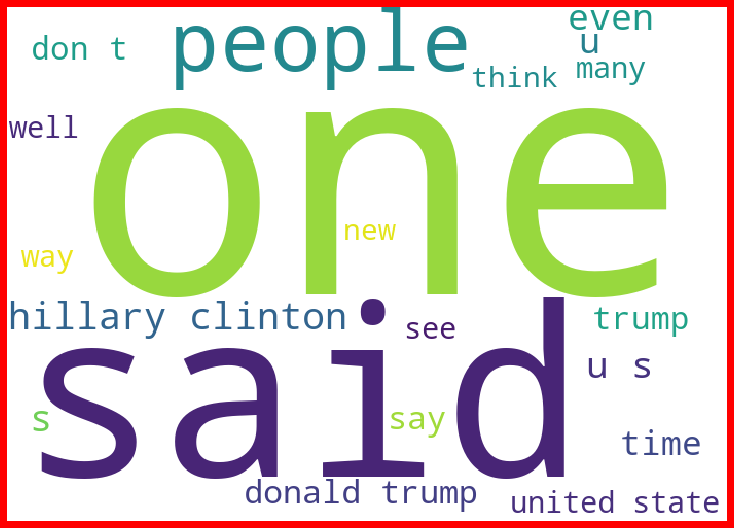
plt.figure(figsize=(10,8),facecolor='r')

plt.imshow(fake\_cloud)

plt.axis('off')

plt.tight\_layout(pad=0)

plt.show()



#Getting sense of loud words in Real

real= train['total'][train['label']==0]

real\_cloud = WordCloud(width=600,height=400,background\_color='white',max\_words=50).generate(' '.join(real))

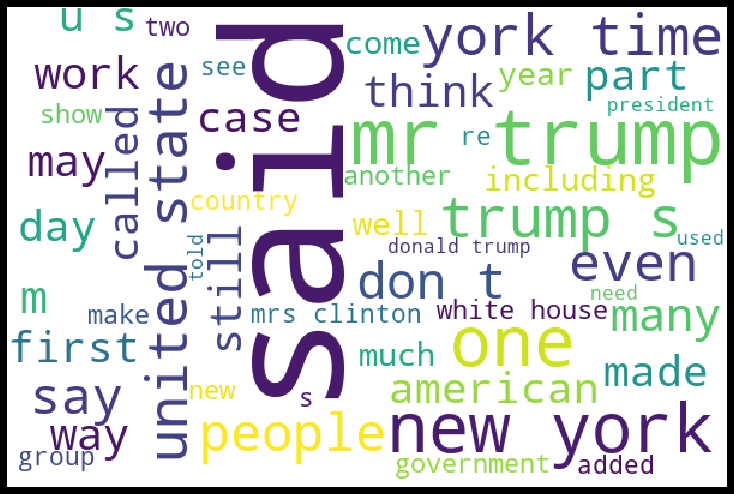
plt.figure(figsize=(10,8),facecolor='k')

plt.imshow(real\_cloud)

plt.axis('off')

plt.tight\_layout(pad=0)

plt.show()



# CONCLUSION

## Key Findings and Conclusions of the Study

From this dataset I get to know that each feature play a very import role to understand the data. Data format plays a very important role in the visualization and Appling the models and algorithms. Reprocessing and data cleaning is the very import step than after only you will get the best accuracy score of the model

## Learning Outcomes of the Study in respect of Data Science

My learnings :-the power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important step to remove missing value or null value fill it with appropriate method Various algorithms I used in this dataset and to get out best result and save that model . algorithm works best in according to this given data the best algorithm is Logistic Regression.