VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

"FAKE NEWS CLASSIFICATION"

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

ADITYA MAHAVEER YARANAL (1CR16CS009)

BHARATH (1CR16CS032)

S P PRANAV (1CR16CS140)

ADITYA A R (1CR17CS401)

Under the Guidance of,

Mrs. SMITHA N

Asst. Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"FAKE NEWS CLASSIFICATION"** carried out by **Mr**. **ADITYA MAHAVEER YARANAL**, USN **1CR16CS009**, **Mr**. **BHARATH R**, USN **1CR16CS032**, **Mr S P PRANAV**, USN **1CR16CS140, Mr**. **ADITYA A R**, USN **1CR17CS401,** bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Mrs. Smitha N	**Dr. Prem Kumar Ramesh**	**Dr. Sanjay Jain**
Asst. Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

External Viva

Name of the examiners — Signature with date

1. _____

2. _____

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "**FAKE NEWS CLASSIFICATION**" has been successfully completed under the guidance of Prof. Mrs. Smitha N, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date: 08/06/2020

Team members:

ADITYA MAHAVEER YARANL (1CR16CS009) _____

BHARATH R (1CR16CS032) _____

S P PRANAV (1CR16CS140) _____

ADITYA A R (1CR16CS401) _____

ABSTRACT

Since fake news tends to spread fast than the real news there a need to classify news as fake or not. In the project the dataset used is from Kaggle website where real news and fake news are in two separate datasets we combined both the datasets into one and trained with different machine learning classification algorithms to classify the news as fake or not.

In this project different feature engineering methods for text data has been used like Bag of words model and word embedding model which is going to convert the text data into feature vectors which is sent into machine learning algorithms to classify the news as fake or not.

With different features and classification algorithms we are going to classify the news as fake or real and the algorithm with the feature which gives us the best result with that feature extraction method and that algorithm we are going to predict the news as fake or real.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Prem Kumar Ramesh,** Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide **Mrs. Smitha N**, Asst. Professor, Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

Figures **Page No.**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Data or information is the most valuable asset. The most important problem to be solved is to evaluate whether the data is relevant or irrelevant. Fake data has a huge impact on lot of people and organizations.

Since fake news tends to spread fast than the real news there a need to classify news as fake or not. In the project the dataset used is from Kaggle website where real news and fake news are in two separate datasets we combined both the datasets into one and trained with different machine learning classification algorithms to classify the news as fake or not.

In this project different feature engineering methods for text data has been used like Bag of words model and word embedding model which is going to convert the text data into feature vectors which is sent into machine learning algorithms to classify the news as fake or not.

With different features and classification algorithms we are going to classify the news as fake or real and the algorithm with the feature which gives us the best result with that feature extraction method and that algorithm we are going to predict the news as fake or real.

In this project we will be ignoring attributes like the source of the news, whether it was reported online or in print, etc. and instead focus only the content matter being reported. We aim to use different machine learning algorithms and determine the best way to classify news .

1.1 Relevance of the project

A fake news classification system using different feature extraction methods and different classification algorithms like Support Vector Machine, Logistic Regression, Gradient Boosting, XG-BOOST, Decision Tree, Random Forest and the best algorithm we are going to use it in predicting the news as fake or real. In order to create a real time application, the algorithm should be fed with the most recent data. Data is of different sizes so that should be properly cleaned to get better results. So we are using different algorithms and feature extraction methods like Bag of words model and Word embedding model to get the best result.

1.2 Problem Statement

Our main aim of the project is to make a machine learning model, with the help of which news can be classified as fake or real with help of different machine learning classification algorithms, deep learning methods and text feature extraction methods for classifying news.

1.3 Objective

To achieve our goal of developing machine learning model to classify news as fake or real, we need perform following tasks in the same order as stated.

-Data Collection and Analysis

-Preprocessing the data

-Text feature extraction

-Using different classification algorithms

-Taking the best classification algorithm and feature extraction method

-Classifying the news as fake or real.

-Deploying the model.

1.4 Scope of the project

In future works, we intend to use highly sophisticated classifying approach, like deep learning with sentiment analysis also and consider many text features like publisher, urls etc., which may increase the accuracy of the classification of news as fake or real.

Automatic fake news detection may be done using the latest news and training the model regularly to get the best results. So this can be used as a filter to upload the news.

1.5 Methodology

Story ID	Requirement description	User stories/Task	Description
Requirement	Gathering project Ideas To work on	Find if there already exist Implementations Of the chosen project idea. If implementation exists, study the existing Implementation. Based on the study, arrive at the Missing necessary Features we can build.	Collect data and ideas from various sources to find a potential project.
Planning	Prepare feature list. Technologies to be used to estimate effort	Decide on the important Functionalities we can add to our implementation	Planning is process which embraces a no of steps to be taken.
Development	High level API class design. Cloud based machines, Google Collab.	Decide and arrive at an overview of modules and classes. Using Collab we made the code much more readable.	Coding basic python, Sci-Kit learning library was majorly used.

Test Cases	Accuracy of random forest, Linear SVM, Decision Tree, Logistic regression, gradient boosting, XGBOOST are compared.	Implement different features for the UI and make sure they are working properly.	Information and research on the algorithms Used for classification of data

Table 1: Agile Methodology

CHAPTER 2

LITERATURE SURVEY

In Today's world, anybody can post the content over the internet. Unfortunately, counterfeit news gathers a lot of consideration over the web, particularly via web-based networking media. Individuals get misdirected and don't reconsider before flowing such mis-educational pieces to the most distant part of the arrangement. Such type of activities are not good for the society where some rumors or vague news evaporates the negative thought among the people or specific category of people[1]. As fast the technology is moving, on the same pace the preventive measures are required to deal with such activities. Broad communications assuming a gigantic job in impacting the general public and as it is normal, a few people attempt to exploit it. There are numerous sites which give false data. They deliberately attempt to bring out purposeful publicity, deceptions and falsehood under the pretense of being true news. Their basic role is to control the data that can cause open to have confidence in it. There are loads of case of such sites everywhere throughout the world .Therefore, counterfeit news influences the brains of the individuals. As indicated by study Scientist accept that numerous man-made brainpower calculations can help in uncovering the bogus news.

Fake news detection is made to stop the rumors that are being spread through the various platforms whether it be social media or messaging platforms, this is done to stop spreading fake news which leads to activities like mob lynching, this has been a great reason motivating us to work on this project. We have been continuously seeing various news of mob lynching that leads to the murder of an individual; fake news detection works on the objective of detecting this fake news and stopping activities like this thereby protecting the society from these unwanted acts of violence.

The digital news industry in the United States is facing a complex future. On one hand, a steadily growing portion of Americans are getting news through the internet, many U.S. adults get news on social media, and employment at digital-native outlets has increased. On the other, digital news has not been immune to issues affecting the broader media environment, including layoffs, made-up news and public distrust.

Half of Americans (52%) say they have changed the way they use social media because of the issue of made-up news. Furthermore, among the Americans who ever get

news through social media, half have stopped following a news source because they thought it was posting made-up news and information. At the same time, about a third (31%) of social media news consumers say they at least sometimes click on news stories they think are made up. So there is need to stop the fake news spreading

2.1 Fake News Detection using Naive Bayes Classifier

This article describes a simple fake news detection method based on one of the artificial intelligence algorithms – naive Bayes classifier. The goal of the research is to examine how this particular method works for this particular problem given a manually labeled news dataset and to support (or not) the idea of using artificial intelligence for fake news detection. The difference between these article and articles on the similar topics is that in this paper naive Bayes classifier was specifically used for fake news detection; also, the developed system was tested on a relatively new data set, which gave an opportunity to evaluate its performance on a recent data.

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable .

Naive Bayes classifiers are a popular statistical technique of e- mail filtering. They emerged in the middle of the 90s and were one of the first attempts to tackle spam filtering problem .

Naive Bayes typically use bag of words features to identify spam e-mail, an approach commonly used in text classification. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other constructions, syntactic or not), with spam and non-spam e-mails and then using Bayes theorem to calculate a probability that an email is or is not a spam message

The research showed, that even quite simple artificial intelligence algorithm (such as naive Bayes classifier) may show a good result on such an important problem as fake news classification. Therefore the results of this research suggest even more, that artificial intelligence techniques may be successfully used to tackle this important problem.

2.2 Research on Text Classification for Identifying Fake News

We use a linear SVM classifier in this paper. The real and fake social news is classified by three models: TF-IDF, Word2vec, and Word2vec weighted by TF-IDF. The idea of Word2vec model is to use the average of all word vectors to represent the document. Among them, the calculation of word2vec is implemented by Gensim[11]. In this paper, we measure the effectiveness of the algorithm by the precision rate, recall rate and F1 score. The precision rate is the ratio of the document number that the classifier correctly judge them to belong to the category and the document number belonging to the category, which denoted by the letter P. The recall rate is the proportion of the document number that the classifier correctly judge them to be the category and the document number belonging to the category, denoted by the letter R. The F1 score is the geometric mean of the recall rate (R) and the precision rate (P). It is a kind of indicator used to comprehensively evaluate the classification effect.

The three algorithms of TF-IDF, Word2vec and Word2vec weighted by TF-IDF have good performance in the field of real and fake news classification. The average F1 score of the three classification algorithms has reached more than 87%. Among them, the precision rate of fake news recognition based on word2vec is 4.31% higher than TF-IDF algorithm, and the average F1 score is 4.25% higher. This paper proposes a method that combines Word2vec and TF-IDF to the field of fake news identification, which achieves good performance. Comparing the precision rate of identifying real news, the Word2vec weighted by TF- IDF is 3.11% higher than Word2vec, the average value of P, R and F1 based on Word2vec weighted by TF-IDF are higher than TF-IDF and Word2vec. which verifies the effectiveness of Word2vec weighted by TF-IDF algorithm in the field of fake news identification.

They used three models to identify fake news. Both of the TF-IDF and Word2vec perform well in the field of fake news identification, but the Word2vec weighted by TF- IDF algorithm is most suitable for distinguishing between real and fake news.

2.3 Fake News Detection Using Deep Learning Techniques

At first they got dataset as news or articles list form. At that point on that dataset processing is done for feature mining and determination. Data should be pre-processing. It incorporates expulsion of punctuations, URL's, images, stemming and stop words. At that point NLP processing for extricating highlights and dependent on that training file is made and afterward we classify that information utilizing classifiers, for example, LR, SVM, NB, RF and DNN. News Dataset: Dataset of news is taken from online.

Processing: Content information needs processing to execute AI on them. There are kinds of methods generally utilized to change over content information into a structure that is prepared for demonstrating. The information processing steps that are applied on headlines and articles.

Stemming and stopwords removal: Stemming technique utilized to detach suffixes or prefixes from a word. And Stops Words may be filtered and processed from content because are increasingly normal and hold less significant data. Stop words generally utilized as a linking element of the sentences, for instance, “and”, “or”, “but”, “of”, “in”, “from”, “to”, “a”, “an”, and “the” etc. This kind of stop words that are of not vital and it can utilize important processing time, since to take away stop words is crucial task.

Here they, presented different algorithms for classifying statements made by public figures were implemented. In proposed system LR, RF, SVM NB and DNN classification techniques are utilized that will help to detect fake news. Classification techniques like LR, RF, SVM NB and DNN for feature selection and extraction utilized, DNN will work fine in execution time and accuracy cases but it needs large memory than other. Then we compare NB, RF, SVM, LR and DNN on basis in terms of time and memory and accuracy, according to comparison results it exhibit that DNN Algorithm is improved than rest algorithm in accuracy and time kind because rest classifiers requires more time and gives less accuracy hence DNN is more crucial to detect the fake news.

2.4 FIND: Fake Information and News Detections using Deep Learning

In this work, they have proposed a framework to distinguish between real and fake news articles. The dataset has been gathered from authentic sources and used for training and

testing purpose. Long Short-Term Memories and Grated Recurrent Units model is used to classification purpose. Data is scraped from trusted sites, and is preprocessed before applying NLP. NLTK is used for performing stop word removal, lemmatization with customized part of speech tagging and making embeddings. Word Index of the tokenized dataset is proposed to be initial weights for the ANN. We have used RNN with LSTM units for creating and updating this neural network. After training the NN, we will have word embeddings for all the n-grams given the training dataset. Loss depiction is also done. Basic loss was based on Binary Cross Entropy Loss. This was reduced using Adam Optimizer. The results of the model give adequate output in terms of epoch requirement, confusion matrix results and classification report. The system has presented up to 94% accuracy which is higher and better than comparable existing models generically used for fake news deduction.

Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) Units are units of an RNN. This consists of cells, input, output, and forget gates. Cells are used to retain memories and gates are used for controlling the information flow in and out of these individual cells.

LSTM networks allow solving the problem of vanishing /exploding gradient. Exploding gradient problem can be easily fixed by gradient clipping by limiting the factor. Vanishing gradient descent problem is faced when output is generated by passing input to a large number of hidden layers. The problem occurs because output of RNN layer is decided by back propagation on all its initial values resulting in multiple weight updates. So the "earliest" memory is "forgotten". This issue solved by LSTM and GRU units which make use of gates.

CHAPTER 3

REQUIREMENTS SPECIFICATION

The requirements can be broken down into 2 major categories namely hardware and software requirements.

3.1. Hardware Requirements

1. Processor: Intel Core2Duo processor or a processor with higher specifications
2. Processor speed: 1.5GHz or above.
3. RAM : 1GB or above
4. Storage space : 1GB or above

3.2. Software Requirements

1. An MS-DOS based operating system like Windows 98/2000/XP/Vista7/8/10/, Linux, MacOS.

2. Python 3

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 SYSTEM ARCHITECTURE

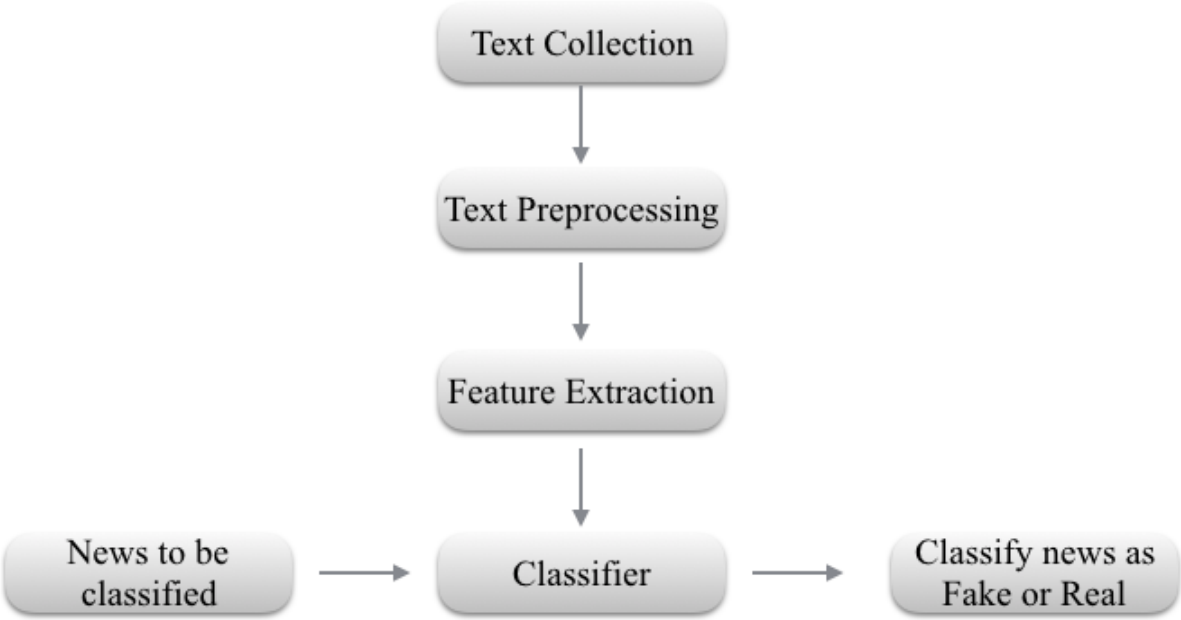

Figure 4.1 System Architecture

4.1.1. Text Collection

The dataset was taken from Kaggle the content and metadata has been extracted from 244 web sites that have been considered to be associated with fake news by the BS Detector

Chrome Extension by Daniel Sieradski. It consists of almost 13000 posts over a period of 30 days. Research on this dataset using language processing tools has already been carried out by Kaggle users. The dataset was generated by Andrew Thompson to create document term matrices using the articles and analyse connections between articles using common political affiliations, medium or subject matter. It contains articles from top 15 American publications and the articles were mostly published between the years of 2016 and 2017. It consists of around 150000 articles that were collected by scraping news website homepages and RSS feeds. However, we will randomly select only 13000 articles from this dataset and merge it with the fake news dataset for more accurate predictions and for avoiding a skewed dataset.

4.1.2. Text Preprocessing

After a text is obtained, we start with text preprocessing. Text preprocessing includes:

Converting all letters to lower case

Removing numbers

Removing punctuations, accent marks

Removing white spaces

Removing stop words

❖ Feature Extraction

Text needs to be converted into numbers before it is used with a machine learning algorithm. For classification of documents, documents are taken as input and a class label is generated as output by the predictive algorithm. The documents need to be converted into fixed-length vectors of numbers for the algorithm to take them as input.The input for the machine learning algorithm are the words encoded as integers or floating point values.

- **Bag of Words (BOW)**

We make the list of unique words in the text corpus called vocabulary. Then we can represent each sentence or document as a vector with each word represented as 1 for present and 0 for absent from the vocabulary.

➢ **Count Vectorizer**

Count Vectorizer generates an encoded vector that contains the length of the entire vocabulary coupled with the frequency of each word by which it appears in the document.

➢ **Term Frequency-Inverse Document Frequency (TF-IDF)**

Term Frequency (TF) = (Number of times term t appears in a document)/(Number of terms in the document)

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Inverse Document Frequency (IDF) = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in. The IDF of a rare word is high, whereas the IDF of a frequent word is likely to be low. Thus having the effect of highlighting words that are distinct.

$$idf(w) = log(\frac{N}{df_t})$$

We calculate TF-IDF value of a term as = TF * IDF.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

- **Word Embedding**

It is a representation of text where words that have the same meaning have a similar representation. In other words it represents words in a coordinate system where related words, based on a corpus of relationships, are placed closer together. Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector.Each word is represented by a real-valued vector.

❖ Classifier

The feature vectors are send to the classifier to classify the news as fake or not.

Support Vector Machine

A support vector machine is in a way a binary classifier since the model works by generating a hyperplane that is used to separate the training data as far as possible. The support vector machine performs well since there is an extremely high number of features in a text classification problem but generally requires a lot of tuning and is memory intensive. Once we have labelled training data (supervised learning), the algorithm generates the best possible hyperplane which categorizes new data automatically. In a two dimensional space this hyperplane would be a line dividing a plane in two parts with each class lying on either side.

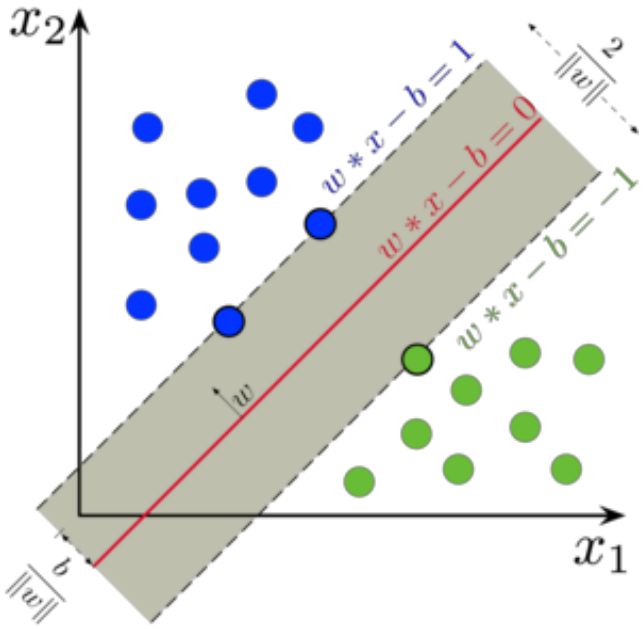

Figure 4.2 SVM Hyper plane

Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression[1] (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio.

The logistic regression model itself simply models probability of output in terms of input and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other; this is a common way to make a binary classifier.

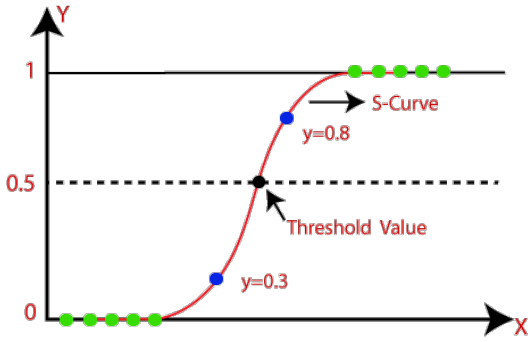

Figure 4.3 Logistic Regression

Decision Tree

Decision Tree is one of Supervised Machine Learning (output label is given) technique where the data is split consecutively through a definite parameter. In a decision tree every node speaks to a feature(attribute), each link(branch) speaks to a decision(rule) and each leaf speaks to an outcome(categorical or continuous value).The entire algorithm is to make a tree for the entire information and process a solitary result at each leaf(or limit error in each leaf). The process of text classification using decision tree marks internal node as terms and the branches withdrawing from them as derived weight, and relative class labels are represented by leaf node. Decision tree utilizes query structure throughout the path of the tree classifying the document from root until it reaches a definite leaf node. In memory decision tree construction, majority of training data will not fit and results inefficient due to swapping the training tuples. This is dealt in as FDT to deal with the multiclass record which lessens the induction cost. A symbolic rule induction system based on decision tree is presented to improve text classification to implement multiclass classification.

Random Forest

As the name suggests, Random Forest algorithm generates the forest with a number of decision trees. So it is the collection of decision trees. Decision trees are attractive classifiers among others because of their high execution speed. Based on random samples from the

database a random forest classifier averages multiple decision trees. Generally, the more trees in the forest is the sign that forest is robust. Similarly in the random forest classifier, high accuracy is obtained by higher the number of trees in the forest. While concurrently creating a tree with decision nodes, a decision tree breaks the dataset down into smaller subsets. The decision root node is selected through highest information gain and leaf nodes based on a pure subset for each iteration simultaneously. Calculation of Information Gain (IG) requires impurity measure (Entropy) of that node. There are various indices to measure the degree of impurity. A leaf node represents a category or pure subset. The trees in a random forest are created under random data so there might be chances to be lack meaning and noisy. In order to make a model with low variance random forest averages these trees. The irrelevant trees drop each other out and the staying meaningful trees yield the final result.

Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

XG-BOOST

This is an ensemble method that seeks to create a strong classifier (model) based on "weak" classifiers. In this context, weak and strong refer to a measure of how correlated are the learners to the actual target variable. By adding models on top of each other iteratively, the errors of the previous model are corrected by the next predictor, until the training data is accurately predicted or reproduced by the model.

Among these classifiers and feature extraction method the best classification algorithm and feature extraction method is used to classify the news as fake or real

CHAPTER 5

IMPLEMENTATION

5.1 Data Collection

```
fake=pd.read_csv('/content/gdrive/My Drive/Datasets/fake_news/fake.csv')

real=pd.read_csv('/content/gdrive/My Drive/Datasets/fake_news/real_news.csv')

real.head()

real.head()

fake.tail()

real['Rumour']=list([0]*len(real))

fake['Rumour']=list([1]*len(fake))

realdf=real[['content','Rumour']]

fakedf=pd.DataFrame()

fakedf['content']=fake['text']

fakedf['Rumour']=fake['Rumour']

df=pd.concat([realdf,fakedf],axis=0)
```

In the above code we are storing the fake news dataset in fake data frame and real news dataset in real data frame after that we are displaying the first five rows in real and fake data frames after that we are creating a column Rumour in both real and fake data frames for real we are inserting zeros for Rumour column and for fake data frame we are inserting ones.We are considering only content and rumour from the data frames and concatenating both these data frames to df.

5.2 Preprocessing

class preprocess:

```
import re

import spacy

nlp=spacy.load('en_core_web_md')

import nltk

def cleanupFinal(tweet):


 #HappyEmoticons

 emoticons_happy = set([

 ':-)', ':)', ';)', ':o)', ':]', ':3', ':c)', ':>', '=]', '8)', '=)', ':}',

 ':^)', ':-D', ':D', '8-D', '8D', 'x-D', 'xD', 'X-D', 'XD', '=-D', '=D',

 '=-3', '=3', ':-))', ":'-)", ":')", ':*', ':^*', '>:P', ':-P', ':P', 'X-P',

 'x-p', 'xp', 'XP', ':-p', ':p', '=p', ':-b', ':b', '>:)', '>;)', '>:-)',

 '<3'

 ])


 # Sad Emoticons

 emoticons_sad = set([

 ':L', ':-/', '>:/', ':S', '>:[', ':@', ':-(', ':[', ':-||', '=L', ':<',

 ':-[', ':-<', '=\\', '=/', '>:(', ':(', '>.<', ":'-(", ":'(", ':\\', ':-c',

 ':c', ':{', '>:\\', ';(  ])

 #combine sad and happy emoticons
```

```
    emoticons = emoticons_happy.union(emoticons_sad)

    #Emoji patterns

    emoji_pattern = re.compile("["

        u"\U0001F600-\U0001F64F"  # emoticons

        u"\U0001F300-\U0001F5FF"  # symbols & pictographs

        u"\U0001F680-\U0001F6FF"  # transport & map symbols

        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)

        u"\U00002702-\U000027B0"

        u"\U000024C2-\U0001F251"

"]+", flags=re.UNICODE)

    word_tokens = tweet.split()

     tweet = re.sub(r':', '', tweet)

    tweet = re.sub(r',Ä¶', '', tweet)

    #replace consecutive non-ASCII characters with a space

    tweet = re.sub(r'[^\x00-\x7F]+',' ', tweet)

    #remove emojis from tweet

    tweet = emoji_pattern.sub(r'', tweet)

    doc=nlp(tweet)

    filtered_words=[]

    for token in doc:

      if ((token.is_punct==False) and (token.is_stop==False) and (token not in emoticons) and
(token.is_space==False) and (token.is_ascii==True)):
```

```
        filtered_words.append(token.text.lower())

    return ' '.join(filtered_words)
```

In the above code we are going to preprocess the text removing the emojis and emoticons from the news content and also removing digits, non-ascii characters, punctuations, spaces, stop words and appending the filtered words and returning the string.

5.3 Feature Extraction

Count Vectorizer

```
from sklearn.model_selection import train_test_split

X=df['cleaned_text_final']

y=df['Rumour']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.33)

from sklearn.feature_extraction.text import CountVectorizer

count=CountVectorizer(min_df=2,max_features=10000)

X_tr=count.fit_transform(X_train)

X_te=count.transform(X_test)
```

TF-IDF Vectorizer

```
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.33)

from sklearn.feature_extraction.text import TfidfVectorizer

tf=TfidfVectorizer()

tfidf=TfidfVectorizer(min_df=2,max_features=10000)

X_train=tfidf.fit_transform(X_train)

X_test=tfidf.transform(X_test)
```

Word Embedding(Spacy Vector Form)

```
nlp=spacy.load('en_core_web_lg')

df=pd.read_csv('/content/gdrive/My Drive/Datasets/fake_news/text_all_features.csv')

def vector(processedText):

  vec=[]

  for text in processedText:

    vec.append(vectorFinal(text))

  return vec

def vectorFinal(text):

  doc=nlp(text)

  return doc.vector

li=range(0,300)

vectors=vector(df['Text Final'])

vector_df=pd.DataFrame([vv for vv in vectors],columns=li)

vector_df.insert(300,"Rumour",list(df['Rumour']),True)

vector_df.insert(301,"emo ratio",list(df['emo ratio']),True)

X=vector_df.drop(['Rumour'],axis=1)

y=vector_df['Rumour']

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.33)
```

Above code is feature extraction of text with count vectorizer ,tfidf vectorizer,and word embedding with spacy vectors which returns the 300 dimensional data for each text .

5.4 Training with different classification algorithms

Support Vector Machine

```
from sklearn.svm import LinearSVC

svm_ws=LinearSVC()

svm_ws.fit(X_train,y_train)

svm_ws_pred=svm_ws.predict(X_test)

print(classification_report(y_test,svm_ws_pred))

accuracy_score(y_test,svm_ws_pred)

metrics(y_test,svm_ws_pred,"SVM LINEAR”,metrics_ws)
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

log_ws=LogisticRegression(max_iter=500)

log_ws.fit(X_train,y_train)

log_ws_pred=log_ws.predict(X_test)

metrics(y_test,log_ws_pred,"LOGISTIC REGRESSION",metrics_ws)

metrics_ws['LOGISTIC REGRESSION’]
```

Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier

decision_tree_ws=DecisionTreeClassifier()

decision_tree_ws.fit(X_train,y_train)

decision_tree_ws_pred=decision_tree_ws.predict(X_test)

metrics(y_test,decision_tree_ws_pred,"DECISION TREE",metrics_ws)
```

metrics_ws['DECISION TREE']

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier

random_forest_ws=RandomForestClassifier()

random_forest_ws.fit(X_train,y_train)

random_forest_ws_pred=random_forest_ws.predict(X_test)

metrics(y_test,random_forest_ws_pred,"RANDOM FOREST",metrics_ws)

metrics_ws['RANDOM FOREST']
```

XG-BOOST Classifier

```
import xgboost as xgb

xgb_ws=xgb.XGBClassifier()

xgb_ws.fit(X_train,y_train)

xgb_ws_pred=xgb_ws.predict(X_test)

metrics(y_test,xgb_ws_pred,"XG-BOOST",metrics_ws)

metrics_ws['XG-BOOST']
```

Gradient Boosting Classifier

```
from sklearn.ensemble import GradientBoostingClassifier

gbc_ws=GradientBoostingClassifier()

gbc_ws.fit(X_train,y_train)

gbc_ws_pred=gbc_ws.predict(X_test)

metrics(y_test,gbc_ws_pred,"GRADIENT BOOSTING",metrics_ws)

metrics_ws['GRADIENT BOOSTING']
```

In this code we have trained with different classifiers and predicted the news as fake or real and store the evaluation metrics for the classifier.

Neural Network

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dropout,Dense

from tensorflow.keras.callbacks import EarlyStopping

early_stop=EarlyStopping(monitor='val_loss',mode='min',patience=5)

model_ws=Sequential()

model_ws.add(Dense(300,activation='relu'))

model_ws.add(Dropout(0.4))

model_ws.add(Dense(150,activation='relu'))

model_ws.add(Dropout(0.4))

model_ws.add(Dense(75,activation='relu'))

model_ws.add(Dropout(0.4))

model_ws.add(Dense(30,activation='relu'))

model_ws.add(Dropout(0.4))

model_ws.add(Dense(1,activation='sigmoid'))

model_ws.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])

X_tr=np.array(X_train.toarray())

X_te=np.array(X_test.toarray())

y_tr=np.array(y_train)

y_te=np.array(y_test)
```

```
model_ws.fit(X_tr,y_tr,batch_size=128,epochs=50,verbose=1,validation_data=(X_te,y_te),ca
llbacks=[early_stop])

losses=pd.DataFrame(model_ws.history.history)

losses.plot()

model_ws_pred=(model_ws.predict(X_te) > 0.5).astype("int32")

metrics(y_te,model_ws_pred,"NEURAL NETWORK",metrics_ws)

metrics_ws['NEURAL NETWORK’]
```

5.5 Evaluation

```
from
sklearn.metricsimportclassification_report,confusion_matrix,accuracy_score,precision_score,
recall_score,f1_score

metrics_ws=dict()

def metrics(y_test,pred,text,metrics):

  precision=precision_score(y_test,pred)

  recall=recall_score(y_test,pred)

  fscore=f1_score(y_test,pred)

  accuracy=accuracy_score(y_test,pred)

metrics[text]={“Accuracy":round(accuracy,2),"Precision":round(precision,2),"Recall":round(
recall,2),"Fscore":round(fscore,2)}
```

we have considered precision, recall, accuracy and F1-Score to compare the evaluation metrics for different classifiers.

5.6 App building with Flask

```
from flask import Flask,render_template,request
```

```
import pickle

import re

import spacy

import string

nlp=spacy.load('en_core_web_sm')

classifier=pickle.load(open('svm.pkl','rb'))

tfidf=pickle.load(open('tfidf.pkl','rb'))

app=Flask(__name__)

@app.route('/')

def home():

        return render_template('index.html')


@app.route('/predict',methods=['POST'])

def predict():

    if request.method == 'POST':

        news = request.form['news']

        data = news


        cleaned_data=preprocess(data)

        vect = tfidf.transform([cleaned_data])

        my_prediction = classifier.predict(vect)
```

```
    return render_template('result.html', prediction=my_prediction)

if __name__ == '__main__':

        app.run(debug=True)
```

We have designed the front end using html and css where input is the news and when we click predict button the output is displayed whether its fake or real.

CHAPTER 6

RESULTS

In this project accuracy, precision, recall, f1-Score is measured for classification of news as fake or real.

Confusion Matrix

It is a common way of presenting true positive (tp), true negative (tn), false positive (fp) and false negative (fn) predictions. Those values are presented in the form of a matrix where the Y-axis shows the true classes while the X-axis shows the predicted classes.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 6.1 Confusion Matrix

Accuracy

It measures how many observations, both positive and negative, were correctly classified.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Precision

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

It measures how many observations predicted as positive are in fact positive.

Recall

Recall actually calculates how many of the Actual Positives our model capture through

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

labeling it as Positive (True Positive).

F1-Score

The F1 score is the harmonic mean of the precision and recall, where an F1 score reaches its

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

best value at 1 (perfect precision and recall).

Classification Metrics with Count Vectorizer

Classifier	Accuracy	Precision	Recall	Fscore
SVM LINEAR	0.91	0.90	0.91	0.90
LOGISTIC REGRESSION	0.93	0.91	0.92	0.92
DECISION TREE	0.82	0.80	0.81	0.80
RANDOM FOREST	0.88	0.94	0.79	0.86
XG-BOOST	0.89	0.88	0.88	0.88
GRADIENT BOOSTING	0.89	0.89	0.88	0.88
NEURAL NETWORK	0.94	0.94	0.93	0.93

Figure 6.2Classification Metrics with Count Vectorizer

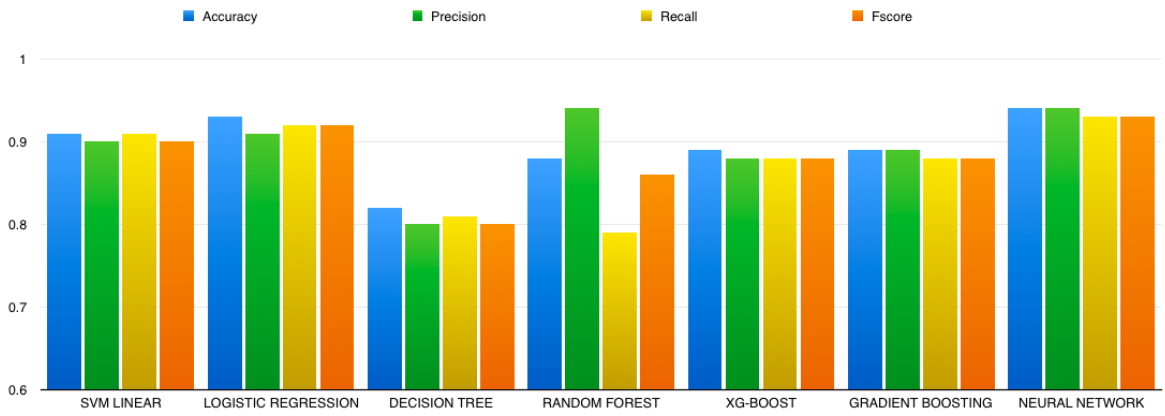

Figure 6.3: Bar Graph For Classification Metrics with Count Vectorizer

Deep neural network performed better with 0.94 accuracy and precision and 0.93 recall and F1-Score. After deep neural network Logistic regression gave better accuracy with 0.93.

Classification Metrics with TF-IDF Vectorizer

Classifier	Accuracy	Precision	Recall	Fscore
SVM LINEAR	0.94	0.93	0.93	0.93
LOGISTIC REGRESSION	0.93	0.93	0.91	0.92
DECISION TREE	0.82	0.79	0.80	0.80
RANDOM FOREST	0.90	0.94	0.83	0.88
XG-BOOST	0.89	0.89	0.88	0.88
GRADIENT BOOSTING	0.90	0.89	0.88	0.88
NEURAL NETWORK	0.93	0.93	0.91	0.92

Figure 6.4: Classification Metrics with TF-IDF Vectorizer

SVM linear gave the best accuracy with TF-IDF vectorizer as feature extraction method with 0.94 accuracy and 0.93 recall, precision and F1-Score after SVM linear Logistic Regression Classifier and Deep Neural Network gave better accuracy with 0.93

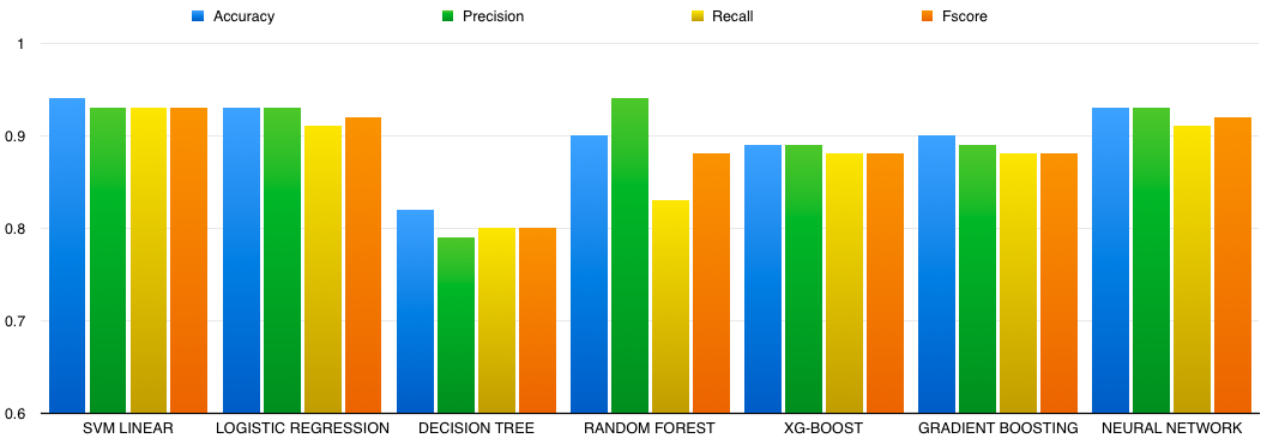

Figure 6.5:Bar Graph For Classification Metrics with TF-IDF Vectorizer

Classification Metrics with Word Embedding(Spacy Vector Form)

Classifier	Accuracy	Precision	Recall	Fscore
SVM LINEAR	0.87	0.88	0.83	0.85
LOGISTIC REGRESSION	0.87	0.88	0.82	0.85
DECISION TREE	0.73	0.69	0.69	0.69
RANDOM FOREST	0.84	0.85	0.79	0.82
XG-BOOST	0.83	0.84	0.76	0.80
GRADIENT BOOSTING	0.83	0.84	0.76	0.80
NEURAL NETWORK	0.90	0.92	0.86	0.89

Figure 6.6Classification Metrics with Word Embedding

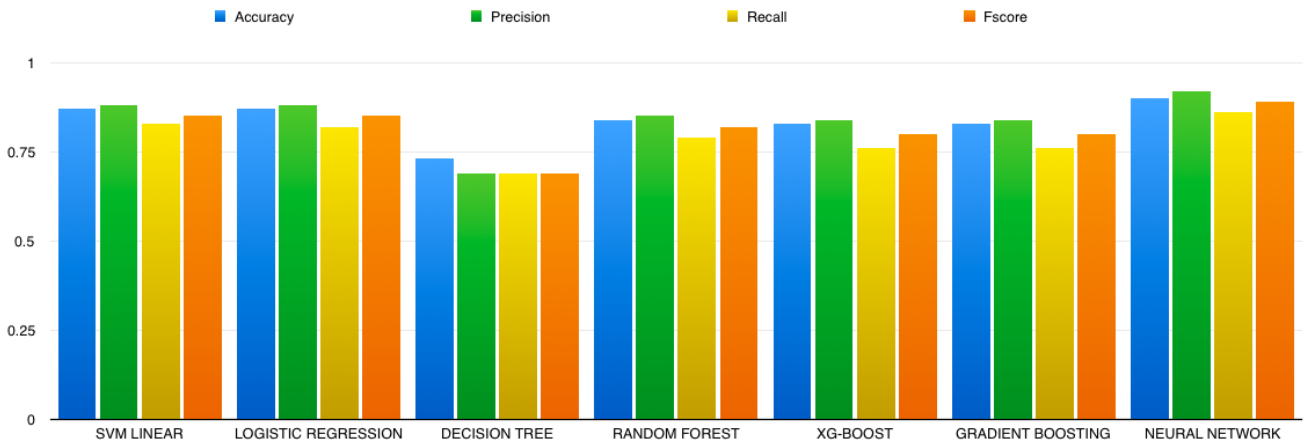

Figure 6.7 Bar Graph For Classification Metrics with Word Embedding

Deep Neural Network performed better with accuracy 0.92 ,precision 0.92,recall 0.86 and F1-Score 0.89 after Deep Neural Network SVM Linear and Logistic Regression gave better accuracy with 0.87

Accuracy

Classifier	word embedding	Count Vectorizer	TF-IDF
SVM LINEAR	0.87	0.91	**0.94**
LOGISTIC REGRESSION	0.87	0.93	0.93
DECISION TREE	0.73	0.82	0.82
RANDOM FOREST	0.84	0.88	0.90
XG-BOOST	0.83	0.89	0.89
GRADIENT BOOSTING	0.83	0.89	0.90
NEURAL NETWORK	0.90	0.94	0.93

Figure 6.8 Accuracy comparison with different feature extraction methods

After using with different classification algorithms like SVM Linear, Logistic Regression Classifier, Decision Tree Classier, Random Forest Classifier, XG-BOOST Classifier, Gradient Boosting Classifier and Neural Network with different feature extraction methods like Count Vectorizer, TF-IDF Vectorizer and Word Embedding model by using Spacy vector .In our project TF-IDF with Linear SVC gave 0.94 accuracy which performed better than the other algorithms and feature extraction methods.

Even though Count Vectorizer with Deep Neural Network gave accuracy 0.94 since deep neural networks are heavy model and count vectoriser doesn't have much information about text we have used TF-IDF with SVM Linear for the model.

Model Deployment

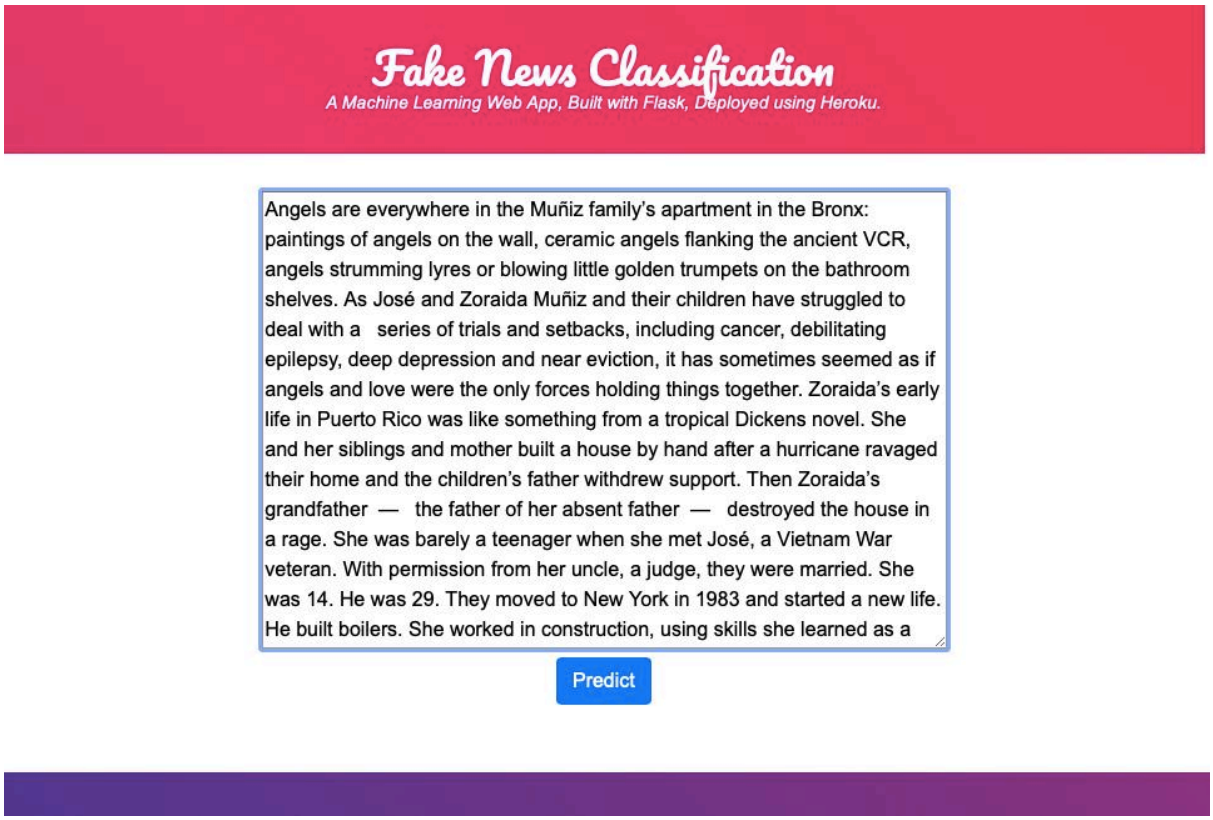

Figure 6.9 Home Page

In this page the news which needs to be predicted must be put in the text box and click predict button which returns to predict page.

Figure 6.10 Output Page for Real News

This is the output page for real news when the model predicts that the news is real this page will be displayed.

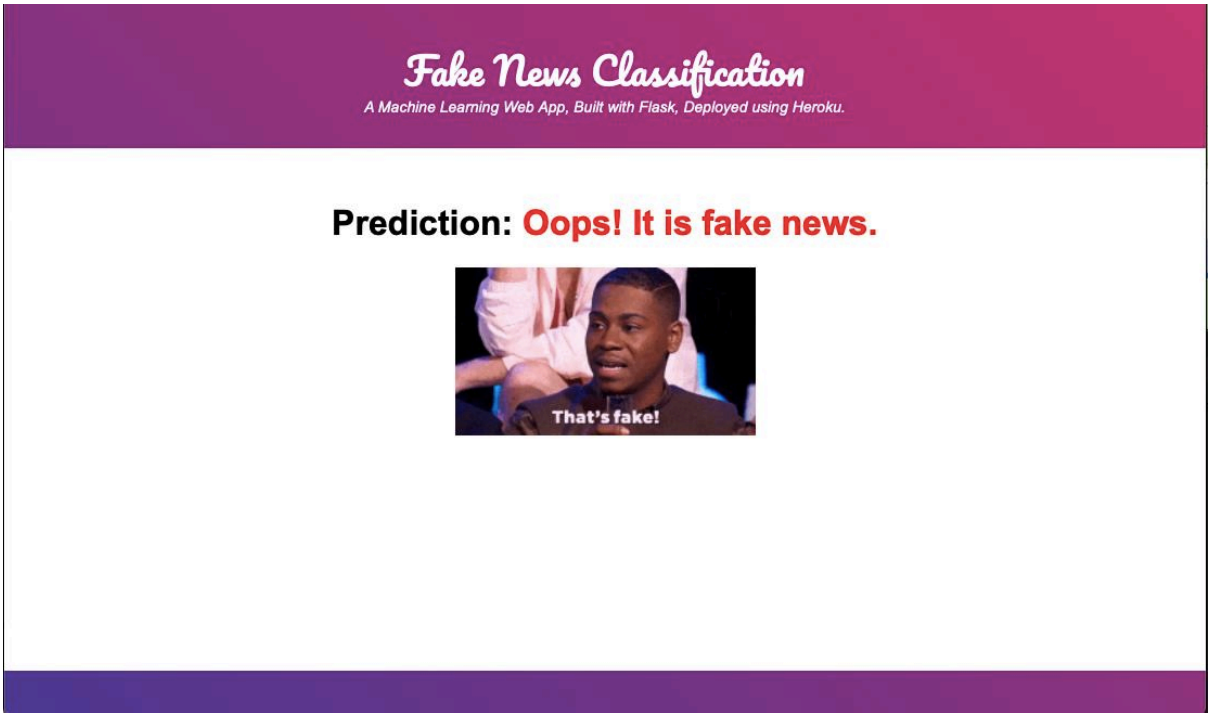

Figure 6.11 Output Page for Fake News

This is the output page for real news when the model predicts that the news is fake this page will be displayed.

CHAPTER 7

TESTING

This chapter gives an overview of the various types of testing incorporated during the entire duration of the project.

7.1 Unit Testing: SVM Linear, Decision Tree, Logistic Regression Classifier,Random forest,XG-BOOST, Gradient Boosting algorithms were separately tested to see if they were able to give the accuracy better than the existing system. These were tested separately so that, we could compare between each other.

7.2 System Testing: Under System Testing technique, the entire system is tested as per the requirements. It is a Black-box type testing that is based on overall requirement specifications and covers all the combined parts of a system. Here we tested if the end integrated code could run on any system, we saw that the integrated code can run on any system having python version 3.6 or more, and we never faced any error.

7.3 Compatibility Testing: In general, we used it on windows , macos ,linux and python version greater than 3.6. And we used the libraries like Sci-Kit ,numpy, pandas etc. And also, there was no specific requirement like we need to use this particular version of python or Sci-Kit library of particular version, so we worked in a closed loop. We were successful at getting greater efficiency with these.

7.4 Usability Testing: This project could be easy for python and data science programmer, not meant for general purpose. Application is usable for data science engineers to pick the model for future research in the area of fake news classification.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

In this project three different feature extraction methods like Count Vectorizer,TF-IDF Vectorizer,Word Embedding has been used.And also different classification algorithms like Linear SVC,Logistic Regression Classifier,Decision Tree Classier,Random Forest Classifier,XG-BOOST Classifier,Gradient Boosting Classifier have been used to classify the news as fake or real.

By using the classification algorithms we got highest accuracy with SVM Linear classification algorithm and with TF-IDF feature extraction with 0.94 accuracy.Even though we got the same accuracy with Neural Network with Count Vectorizer, Neural Networks and take more time to train and its complex so we used Linear SVC which is not so complex and takes less time to compute.

8.2 FUTURE SCOPE

In future we can also use deep learning met.hods and sentiment analysis to classify the news as fake or real which may get high accuracy and we can extract further useful text like publication of the news, url domain etc.,

We can use more data for training purposes - In machine learning problems usually availability of more data significantly improves the performance of a learning algorithm. The dataset, which we used in this project contains only around 28000 articles. This number is quite small, and a dataset with larger number of news articles from different sources would be of a great help for the learning process as news from different sources will involve larger vocabulary and greater content.

REFERENCES

[1] Fake news detection Akshay Jain, Amey Kasbe 2018 IEEE International Students Conference on Electrical, Electronics and Computer Sciences

[2] A Smart System For Fake News Detection Using Machine Learning 2019 2nd International Conference on Issues and Challenges in Intelligent Computing Techniques

[3] Research on Text Classification for Identifying Fake News 2018 International Conference on Security,Pattern Analysis, and Cybernetics(SPAC)

[4] https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

[5]https://towardsdatascience.com/nlp-text-preprocessing-a-practical-guide-and-template-d80874676e79

[6] https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

[7] Chaitra K Hiramath and Prof. G.C Deshpande "Fake News Detection Using Deep Learning Techniques" 2019 1st International Conference on Advances in Information Technology

[8] Abhishek Verma, Vanshika Mittal and Suma Dawn "FIND: Fake Information and News Detections using Deep Learning"

[9] Mykhailo Granik, Volodymyr Mesyura "Fake News Detection Using Naive Bayes Classifier" 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)

[10] Shenhao Zhang,Yihui Wang and Chengxiang Tan "Research on Text Classification for Identifying Fake News" 2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)