IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

An Effective Content and Network Analysis on Internet to Identify and rate Digital Misinformation

[1]Jyothsna Katkuri, [2]G.Venkat Rami Reddy

[1]Student, [2]Professor

[1] Computer Science

[1]School of Information Technology, JNTUH, Hyderabad, India

Abstract: Digital Misinformation [pages diffusing Conspiracy-disseminating controversial Information, typically deficit of authentic evidence and sometimes contradictory of the official news] has become so pervasive on Internet that it has been listed by the World Economic Forum (WEF) as one of the major threats to our human society. The objective of the proposed project work is to review and analyze the content and Interpersonal network on Internet platforms which includes Forums, Discussion boards and related areas to identify a lot of misinformation which have been propagated on the Internet platform. In this work, extensive network analysis using Machine Learning mechanism and creation of effective data analysis system etc. which can be used for real-time detection of fake/ misinformation.

Index Terms **– Tokenization, Stemming, N-Grams, Bag of Words, Tf-idf, and Support Vector Machine.**

I. INTRODUCTION

- Misinformation is false or inaccurate information
- Digital Misinformation [pages diffusing Conspiracy-disseminating controversial Information, typically deficit of reliable and authentic evidence also sometimes contradictory of the official news] has become so pervasive on Internet that it has been listed by the World Economic Forum as one of the major threats to our human society.
- The objective of the proposed project work is to review and analyze the content and Interpersonal network to identify a lot of misinformation which have been propagated on the Internet platform.
- In this work, extensive network analysis using Machine Learning mechanism and creation of effective data analysis system etc. which can be used for real-time detection of fake/ misinformation.

1.1. Fake news examples and how they are seen

1) **Click bait**: These are stories that are deliberately fabricated to gain more website visitors and increase advertising revenue for websites.
2) **Propaganda**: The abstract, biased and unobjective information created deliberately to influence and mislead an audience to promote a political cause or an agenda.
3) **Satire/parody**: Lots of websites and social media accounts publish fake news stories for entertainment and parody. For example; The Onion (satirical digital media company), Waterford Whispers, The Daily Mash, etc.
4) **Sloppy journalism**: Without checking all of the facts sometimes reporters or journalists may publish a story with unreliable information which can mislead audiences
5) **Misleading headings**: Unverified yet significant content or the information that is not typically false can be distorted using misleading or sensationalist headlines.
6) **Biased or slanted news**: Many people are drawn to news or stories that coincide with their personal interests or beliefs thus these fake news can influence and prey on these prejudices and biased views. (Tend to display news that they think we will like based on our personalized searches e.g. Clinton could still be president over Trump)

1.2. Why Misinformation is created? (Who benefits from it?)

- It is created to be widely shared online for the purpose of financial gain (generating ad revenue) via web traffic or
- Discrediting a public figure, political movement, company, etc.

Impact of it

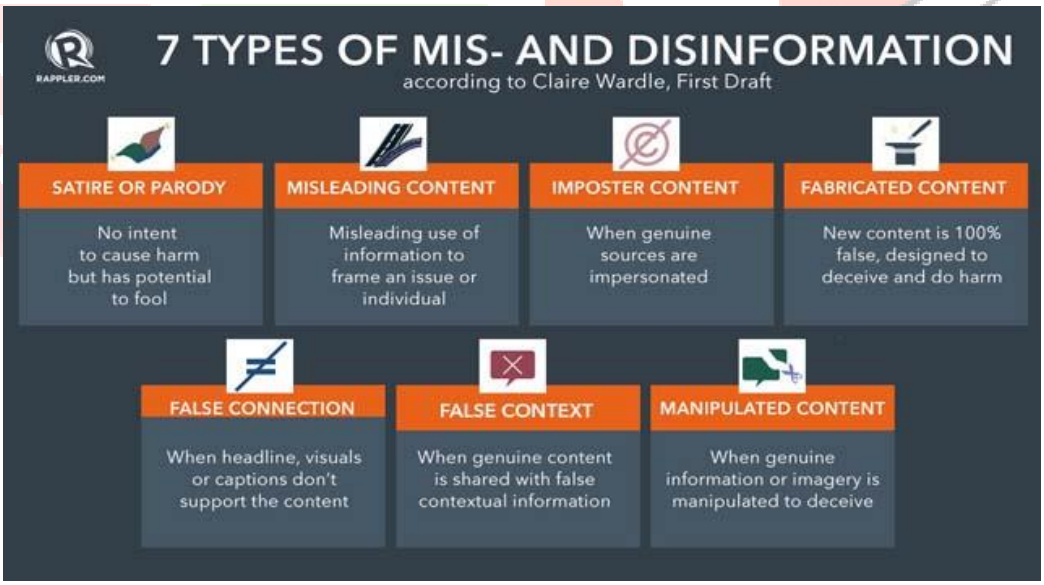
- In the real world, false information has been shown to have significant impact on the stock market
- hampering response during natural disasters
- deliberately misleading 'news' could pose a threat to national security(terroristic activity)
- Fake news destroys your credibility. If your arguments are built on bad information, it will be much more difficult for people to believe you in the future.
- People start to mistrust information all together. They stop listening to industry news or reports, and disengage entirely, slowing their professional growth and development. Ultimately, this can damage learning culture.
- unsure of who to trust

1.3. Categorization of False Info

False Information is categorized as follows:
1) Based on Intent --- (Misinformation, Dis information)
2) Based on Knowledge --- (opinion, fact based)

- common causes of Misinformation include misrepresentation or distortion of important and true information by an actor, due to misunderstanding, inadvertence or even cognitive biases;
- Disinformation is spread with the intent to deceive(e.g.: driving online traffic to target websites to earn money by advertisements, political disinformation)
- Opinion-based fanciful and inaccurate information expresses individual belief or opinion (whether honestly expressed or not) with no absolute ground truth. Creator of the opinion piece knowingly or unknowingly creates false (e.g.: fake reviews of products on e-commerce websites) opinions, potentially to influence the readers' opinion or decision.
- Fact-based false information involves information which contradicts, a single-valued ground truth information. The motive of this type of info is to make it harder for the reader to distinguish true from false info, and make them believe in the false version of the information (fake news, rumors, and hoaxes).

1.4. Types of Fake News



II. PREPROCESSING TECHNIQUES

A. TOKENIZATION

Words which are there in a particular sentence are considered as "Tokens" and similarly the sentences which are there in a particular paragraph are considered as "Tokens". From this it is clear that tokenization is nothing but a process in which the words get split from a sentence and sentence gets split from a paragraph which are considered as tokens. To perform this tokenization nltk library is required which contains the tokenize () module. The 2 methods that are used are word_tokenize () and sentence_tokenize ().

B. STEMMING

Stemming is like a method of Normalizing. Abundant word variations deliver the same meaning, irrespective of the tense. Say, the word "Imagine", the stemming algorithm reduces words
"Imagining",
"Imagined",
"Imagines" etc. to its root word.
Stemming is extensively used in Search Engines i.e. in an information retrieval system. Even in Stemming there arises two kinds of errors. They are "Over stemming" and "Under stemming". Assume 2 words that are not of different stems, are stemmed to the single same root then it is referred as under stemming. If 2 words are being stemmed to same single root that are of different stems that it is referred as over stemming.
Implementation of stemming using Nltk.

```
# import these modules
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

# choose some words to be stemmed
words = ["program", "programs", "programer", "programing", "programers"]

for w in words:
    print(w, " : ", ps.stem(w))
```

2.1. FEATURE SELECTION METHODS

A. N-GRAMS

The N-grams are nothing but a set of co-occurring words which typically moves one word forward. N-grams are used in Natural Language Processing tasks as well as in Text mining. The words which are present in a particular sentence are Unigrams, here N=1. In case of "Bigrams", N=2. If it is a "Trigram", N=3. If N=4 or 5 they are treated as 4 grams or 5grams.
Here the following are the N-grams generated using nltk library

```
import nltk
from nltk.util import ngrams

# Function to generate n-grams from sentences.
def extract_ngrams(data, num):
    n_grams = ngrams(nltk.word_tokenize(data), num)
    return [ ' '.join(grams) for grams in n_grams]

data = 'A class is a blueprint for the object.'

print("1-gram: ", extract_ngrams(data, 1))
print("2-gram: ", extract_ngrams(data, 2))
print("3-gram: ", extract_ngrams(data, 3))
print("4-gram: ", extract_ngrams(data, 4))
```

```
OUTPUT:
 1-gram:  ['A', 'class', 'is', 'a', 'blueprint', 'for',
                    'the', 'object']
 2-gram:  ['A class', 'class is', 'is a', 'a blueprint',
       'blueprint for', 'for the', 'the object']
 3-gram:  ['A class is', 'class is a', 'is a blueprint',
    'a blueprint for', 'blueprint for the', 'for the
                        object']
 4-gram:  ['A class is a', 'class is a blueprint', 'is a
  blueprint for', 'a blueprint for the', 'blueprint for
                     the object']
```

B. Bag of Words

It is known that when an algorithm applied in Natural Language Processing, it works on numbers. Since the text cannot be fed into the algorithm directly, this model pre-processes the text in order to convert the text into bag of words/ large corpus of the words, which will keep a count of the total occurrences of the words that are used very often. Bag of word is one of the ways to extract most important features from the given text (Machine Learning's Modelling we use).
Initially a dictionary should be declared in order to hold these bag of words, for each word we need to verify if it is present in it or not. If that particular word is present then we increment the count else we add it as a new one and set the as count=1

```
# Creating the Bag of Words model
word2count = {}
for data in dataset:
    words = nltk.word_tokenize(data)
    for word in words:
        if word not in word2count.keys():
            word2count[word] = 1
        else:
            word2count[word] += 1
```

When our dataset is very large the words may reach hundreds of thousands there we select a particular number of the words which are very often. A vector is constructed to let us know the word frequency. If it is a word which is frequent then set it as 1 else 0.

```
X = []
for data in dataset:
    vector = []
    for word in freq_words:
        if word in nltk.word_tokenize(data):
            vector.append(1)
        else:
            vector.append(0)
    X.append(vector)
X = np.asarray(X)
```

For the given input (as follows)

Beans. I was trying to explain to somebody as we were flying in, that's corn. That's beans. And they were very impressed at my agricultural knowledge. Please give it up for Amaury once again for that outstanding introduction. I have a bunch of good friends here today, including somebody who I served with, who is one of the finest senators in the country, and we're lucky to have him, your Senator, Dick Durbin is here. I also noticed, by the way, former Governor Edgar here, who I haven't seen in a long time, and somehow he has not aged and I have. And it's great to see you, Governor. I want to thank President Killeen and everybody at the U of I System for making it possible for me to be here today. And I am deeply honored at the Paul Douglas Award that is being given to me. He is somebody who set the path for so much outstanding public service here in Illinois. Now, I want to start by addressing the elephant in the room. I know people are still wondering why I didn't speak at the commencement.

Sample Output: (in a vector format)

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	0	0	1	0	0	0	1
2	0	0	0	0	0	0	0	0	1	0	0	0	0
3	0	0	0	1	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	1	0	1	0	0	0	0
5	1	1	1	1	1	1	0	0	0	1	1	1	1
6	1	1	0	1	1	1	0	0	0	1	0	1	0
7	0	0	1	1	0	0	0	0	0	0	0	0	0
8	1	1	1	1	0	1	1	1	0	0	0	0	0
9	1	1	1	1	0	0	0	1	1	0	1	0	0
10	0	1	0	0	1	1	1	0	0	1	1	0	1
11	1	1	1	0	1	0	0	0	0	0	0	0	0
12	1	1	0	0	0	0	0	1	0	0	0	0	0

C. TERM FREQUENCY (TF) - INVERSE DOCUMENT FREQUENCY (IDF)

Term Frequency gives the word frequency. Though the word is appearing multiple times or if it is of higher frequency, it doesn't mean that it has higher weightage. For example words like – I, is, are, was etc.
So, in order to give more importance to the word which is of higher priority (words that hold much importance) though less frequent, normalization is required so Inverse Document Frequency does this. It is calculated in the following way.

TFIDF

For a term i in document j:

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

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

The following is an example of Td-Idf.

sentence 1	earth is the third planet from the sun
sentence 2	Jupiter is the largest planet

Word	TF (Sentence 1)	TF (Sentence 2)	IDF	TF*IDF (sentence 1)	TF*IDF (Sentence 2)
earth	1/8	0	log(2/1)=0	0.0375	0
is	1/8	1/5	log(2/2)=0	0	0
the	2/8	1/5	log(2/2)=0	0	0
third	1/8	0	log(2/1)=0.3	0.0375	0
planet	1/8	1/5	log(2/2)=0	0	0
from	0	0	log(2/1)=0.3	0	0
sun	1/8	0	log(2/1)=0.3	0.0375	0
largest	0	1/5	log(2/1)=0.3	0	0.06
Jupiter	0	1/5	log(2/1)=0.3	0	0.06

III. ANALYSIS

3.1 Existing Model And Its Disadvantages

Over on going days Machine Learning especially Supervised Machine Learning algorithms are effective in solving the classification problems. But the existing models are not effectively using all the feature selection, preprocessing techniques which are very useful in predicting the test data accurately.

3.2. Proposed Model And Its Advantages

The proposed model is able to utilize the various preprocessing techniques as well as Feature Selection methods and can predict the test data accurately
It is effective and feasible to use.

3.3. External Interface Requirements

System requirements:

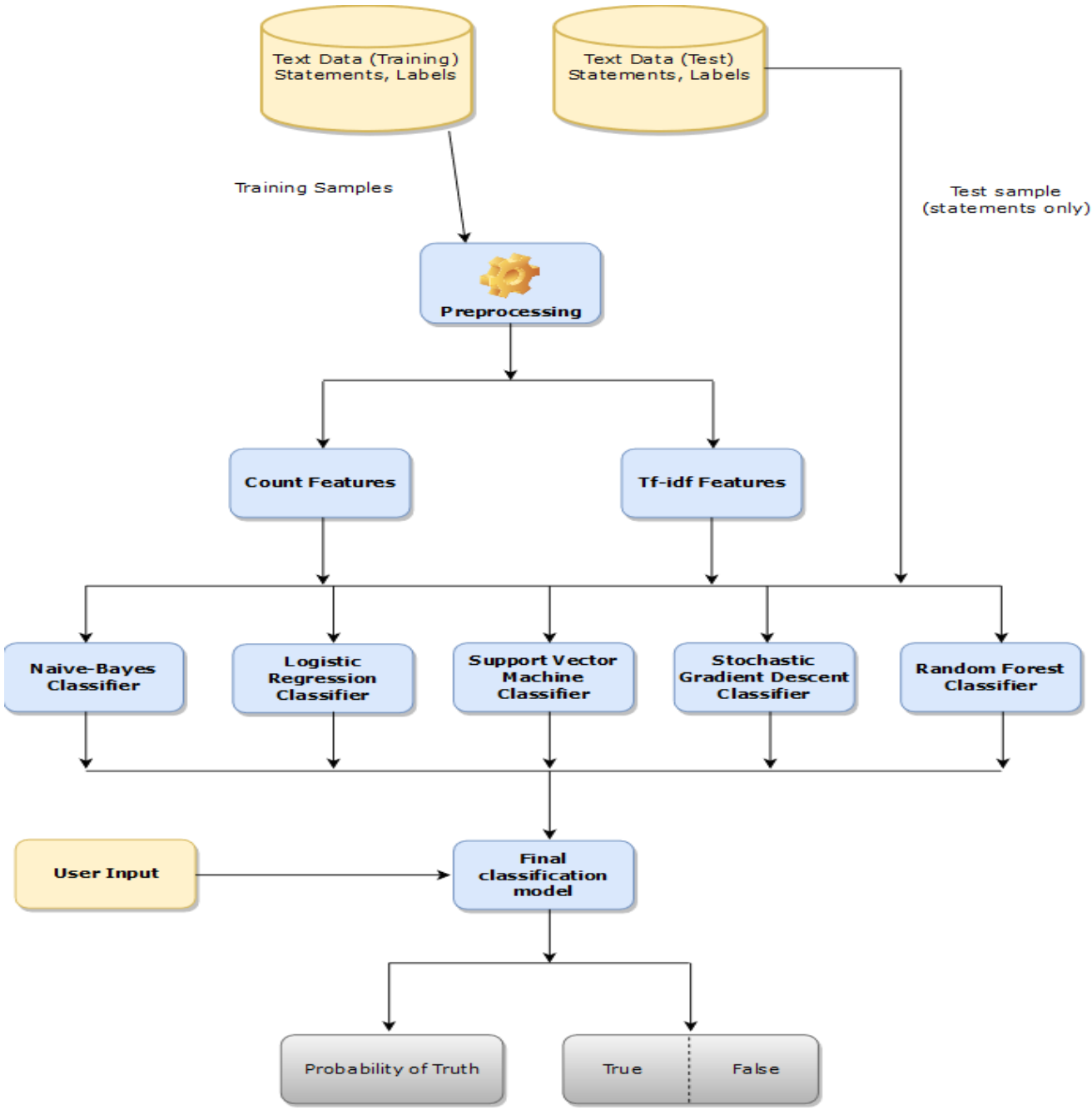
- Windows XP
- RAM:16 GB
- Programming language : Python

Libraries Used:

- ✓ numpy
- ✓ pandas
- ✓ sklearn
- ✓ pipeline
- ✓ pickle
- ✓ nltk
- ✓ seaborn

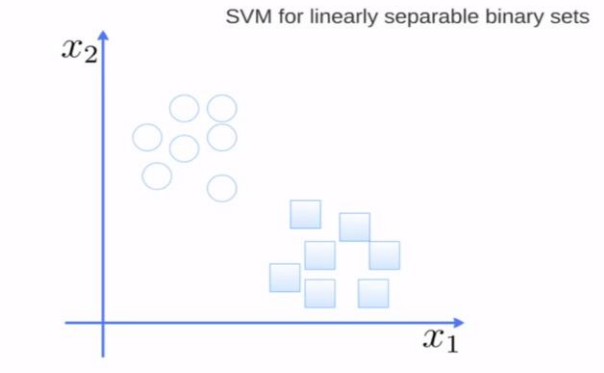
IV. DESIGN

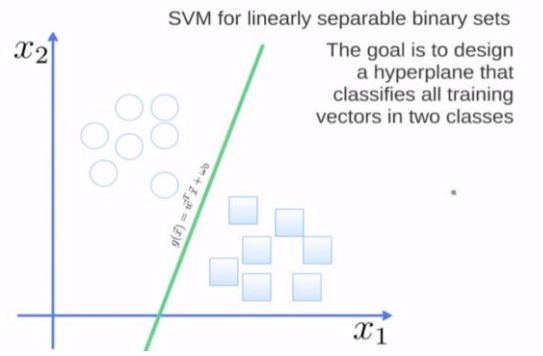
4.1. MODEL ARCHITECTURE

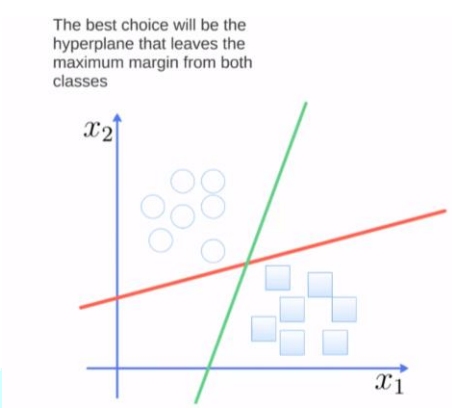


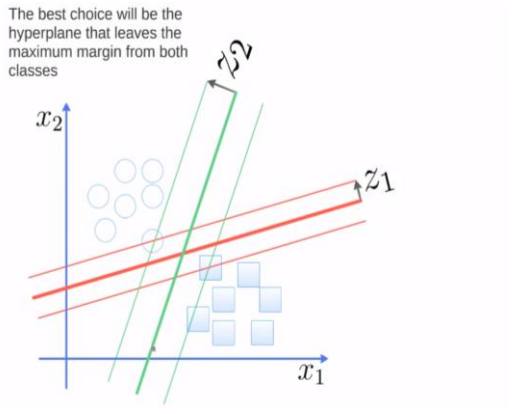
4.2. SUPPORT VECTOR MACHINE

Inorder to solve the problems such as classification etc. Machine Learning algorithms especially Supervised ones are largely used. Support Vector Machine is one such algsorithm. Not only the liner problems, but it can also solve the non-liner problems too.This algorithm creates a hyper plane nothing but a line that seperates the given data into classes. In this SVM algorithm, we plot each data item as a point in the n-dimensional space (where ‘n’ represents number of features). Suppose x1 and x2 are the features here.





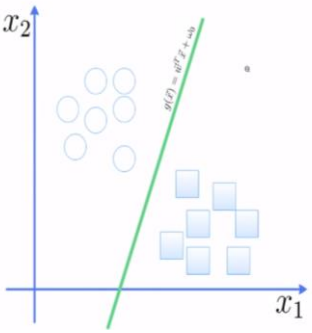




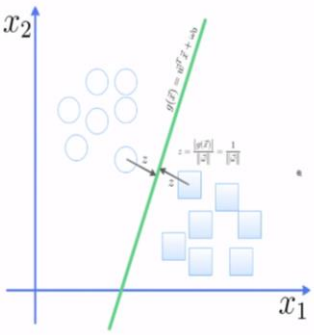
$$z_2 > z_1$$

From the above diagram it is clear that Z2 > Z1 i.e...Z2 is leaving the maximum margin (the distance from hyper plane to the closest element), so this hyper plane is best in this case.

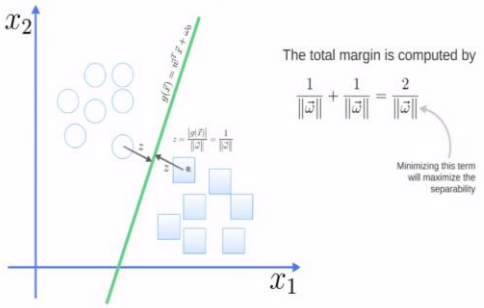
$$g(\vec{x}) \geqslant 1, \quad \forall \vec{x} \in \text{class 1}$$
$$g(\vec{x}) \leqslant -1, \quad \forall \vec{x} \in \text{class 2}$$



$$g(\vec{x}) \geqslant 1, \quad \forall \vec{x} \in \text{class 1}$$
$$g(\vec{x}) \leqslant -1, \quad \forall \vec{x} \in \text{class 2}$$



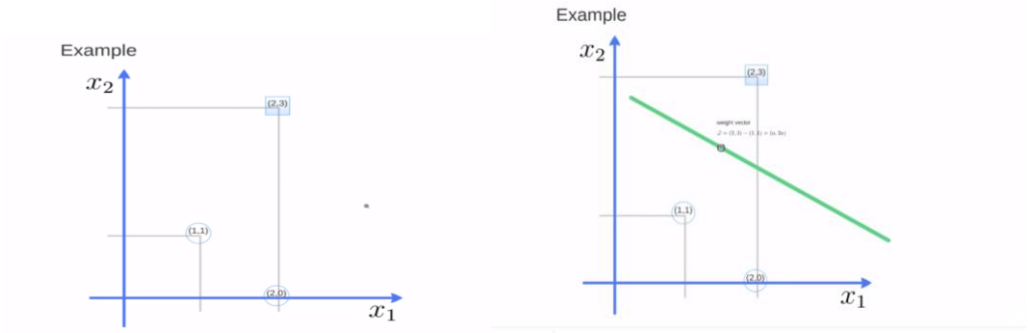
$$g(\vec{x}) \geqslant 1, \quad \forall \vec{x} \in \text{class 1}$$
$$g(\vec{x}) \leqslant -1, \quad \forall \vec{x} \in \text{class 2}$$

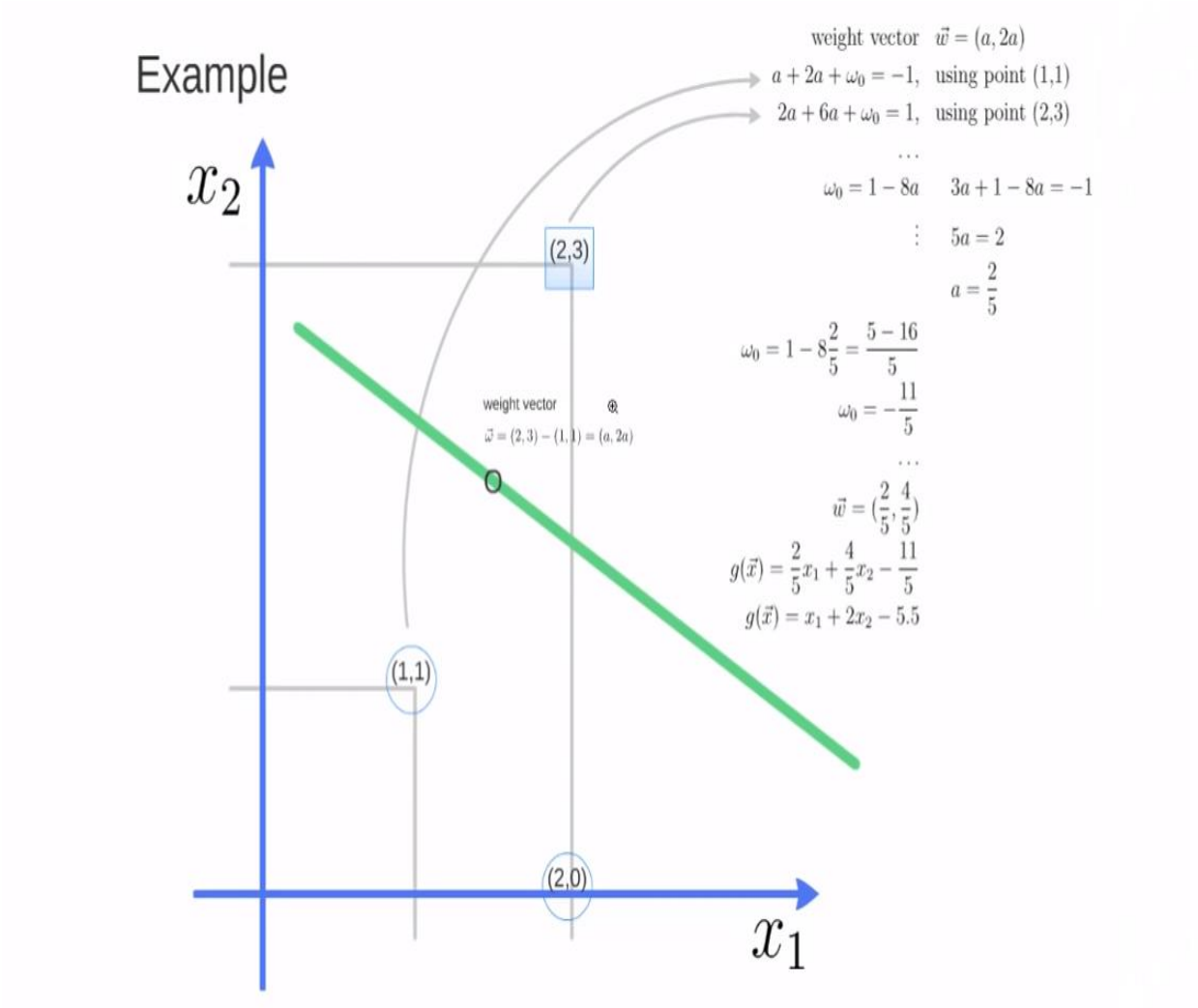


Minimizing $\vec{\omega}$ is a nonlinear optimization task, solved by the Karush-Kuhn-Tucker (KKT) conditions, using Langrange multipliers λ_i

$$\vec{\omega} = \sum_{i=0}^{N} \lambda_i y_i \vec{x}_i$$

$$\sum_{i=0}^{N} \lambda_i y_i = 0$$





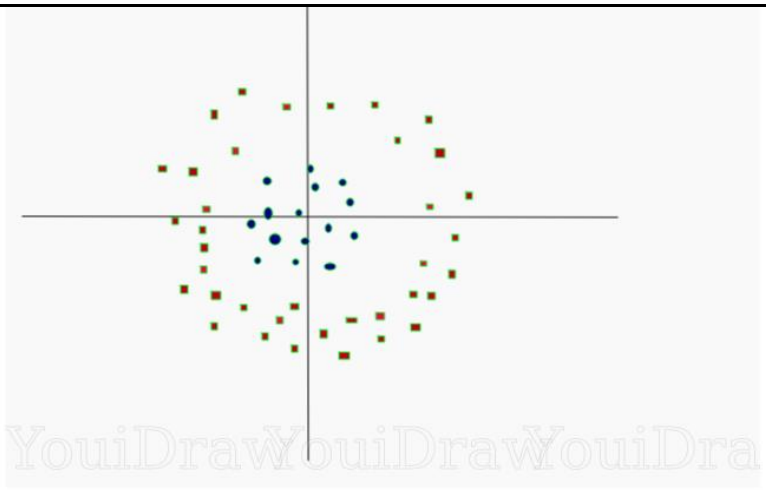
These are called the Support Vectors . . .

$$\vec{w} = (\frac{2}{5}, \frac{4}{5})$$

$$g(\vec{x}) = \frac{2}{5}x_1 + \frac{4}{5}x_2 - \frac{11}{5}$$

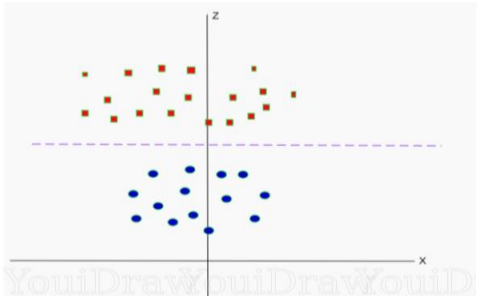
$$g(\vec{x}) = x_1 + 2x_2 - 5.5$$

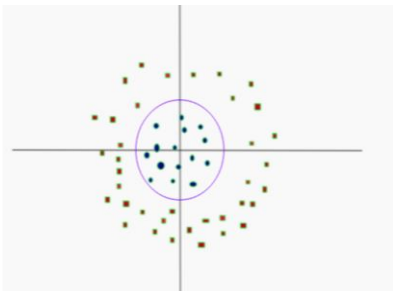
On substitution of these values (2/5, 4/5) results in the above equation.
This equation will define the final hyper plane. And this hyper plane classifies elements using support vector machines.



Non-linearly separable data

The above one is a bit complex dataset because that is not a linearly separable one. We cannot draw a straight line which can classify the above data. But, this data can be converted to linearly separable data in higher dimension. By adding one more dimension assume it as z-axis. Let the co-ordinates on z-axis be governed by the constraint, $z = x^2+y^2$ .So, basically z co-ordinate is nothing but the square of the distance of the point from the origin. Now, let's plot the data on z-axis.





The data is linearly separable now. Let the (purple) line separating the data in higher dimension be z=k, where k is a constant. Since, $z=x^2+y^2$ we get $x^2 + y^2 = k$; which is nothing but an equation of a circle. So, by using this transformation, we can project this linear separator in higher dimension back in original dimensions.

Hence we are able to classify data by adding an extra dimension to it (i.e., 'z') so that it becomes linearly separable and then projecting the decision boundary back to its original dimensions using mathematical transformation. But finding the correct transformation for any given dataset is bit difficult. Using kernels in sklearn's SVM implementation helps in performing this job successfully.

4.3. Methodology:

The machine learning algorithm used in the project is Support vector machine. This is very effective in predicting (accurately) and memory efficient even. It uses Kernel functions to segregate the data by adding more dimensions to a low dimension space. It even converts the in separable problem to a separable one. Kernel trick helps to make more accurate classifiers. The different types of Kernel are:

a) Linear Kernel: can be used as a dot product between any two given observations.
b) Polynomial Kernel: It's a generalised form of Linear Kernel. It can distinguish curved and non- linear input space.
c) Radial Basis Function (RBF): this kernel is used in classification to map the space in finite dimension.

Sequence of actions/ Step by step procedure:

1) Installing the Libraries such as sklearn, pandas, numpy, csv, nltk and importing the modules such as svm, datasets and test-train-split.
2) Exploring the data
3) Splitting the dataset into test and train data (e.g.: 30% test data and 70% train data or 40% test data and 60% train data) e
4) Generating the model i.e... Classifier
5) Training the model (using "fit")
6) Predicting the response (using "predict")
7) Printing the accuracy
8) Plotting Confusion matrix etc.

v. IMPLEMENTATION

5.1. DataPrep.py

The Dataprep.py file contains all the pre-processing functions in order to process all the texts and input documents i.e. {Tokenization, Stemming} etc.

```python
# -*- coding: utf-8 -*-

#import os
import pandas as pd
import csv
import numpy as np
import nltk
from nltk.stem import SnowballStemmer
from nltk.stem.porter import PorterStemmer
from nltk.tokenize import word_tokenize
import seaborn as sb

#reading data files

test_filename = 'test.csv'
train_filename = 'train.csv'
valid_filename = 'valid.csv'

train_news = pd.read_csv(train_filename)
test_news = pd.read_csv(test_filename)
valid_news = pd.read_csv(valid_filename)


#data observation
def data_obs():
    print("training dataset size:")
    print(train_news.shape)
    print(train_news.head(10))

    #below dataset were used for testing and validation purposes
    print(test_news.shape)
    print(test_news.head(10))
```

5.2. FeatureSelection.py

The FeatureSelection.py file contains feature extraction and selection methods like Bag of Words, n-grams, term frequency-inverse document frequency (td-idf) etc.

```python
# -*- coding: utf-8 -*-
"""
Note: before we can train an algorithm to classify fake news labels, we need to extract features from it.
It means reducing the mass of unstructured data into some uniform set of attributes that an algorithm
can understand. For fake news detection, it could be word counts (bag of words).
"""
import DataPrep
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
import nltk
import nltk.corpus
from nltk.tokenize import word_tokenize
from gensim.models.word2vec import Word2Vec


#we will start with simple bag of words technique
#creating feature vector - document term matrix
countV = CountVectorizer()
train_count = countV.fit_transform(DataPrep.train_news['Statement'].values)

print(countV)
print(train_count)

#print training doc term matrix
#we have matrix of size of (10240, 12196) by calling below
def get_countVectorizer_stats():
```

```
    #vocab size
    train_count.shape

    #check vocabulary using below command
    print(countV.vocabulary_)

    #get feature names
    print(countV.get_feature_names()[:25])


#create tf-df frequency features
#tf-idf
tfidfV = TfidfTransformer()
train_tfidf = tfidfV.fit_transform(train_count)

def get_tfidf_stats():
    train_tfidf.shape
    #get train data feature names
    print(train_tfidf.A[:10])


#bag of words - with n-grams
#countV_ngram = CountVectorizer(ngram_range=(1,3),stop_words='english')
#tfidf_ngram  = TfidfTransformer(use_idf=True,smooth_idf=True)

tfidf_ngram = TfidfVectorizer(stop_words='english',ngram_range=(1,4),use_idf=True,smooth_idf=True)
```

5.3. classifier.py

In this classifier.py file the classifier is build and the features that are extracted are fed into this classifier.

```
import DataPrep
import FeatureSelection
import numpy as np
import pandas as pd
import pickle
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn import svm
from sklearn.metrics import confusion_matrix, f1_score, classification_report
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import learning_curve
import matplotlib.pyplot as plt
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import average_precision_score

#string to test
doc_new = ['Lockdown has been imposed in 2020']

#the feature selection has been done in FeatureSelection.py module.
# here we will create models using those features for prediction
```

```
#building Linear SVM classfier
svm_pipeline = Pipeline([
        ('svmCV',FeatureSelection.countV),
        ('svm_clf',svm.LinearSVC())
        ])

svm_pipeline.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_svm = svm_pipeline.predict(DataPrep.test_news['Statement'])
np.mean(predicted_svm == DataPrep.test_news['Label'])

#User defined functon for K-Fold cross validatoin
def build_confusion_matrix(classifier):

    k_fold = KFold(n_splits=5)
    scores = []
    confusion = np.array([[0,0],[0,0]])

    for train_ind, test_ind in k_fold.split(DataPrep.train_news):
        train_text = DataPrep.train_news.iloc[train_ind]['Statement']
        train_y = DataPrep.train_news.iloc[train_ind]['Label']

        test_text = DataPrep.train_news.iloc[test_ind]['Statement']
        test_y = DataPrep.train_news.iloc[test_ind]['Label']

        classifier.fit(train_text,train_y)
        predictions = classifier.predict(test_text)

        confusion += confusion_matrix(test_y,predictions)
        score = f1_score(test_y,predictions)
        scores.append(score)
```

5.4. prediction.py

```
import pickle

var = input("Please enter the news text you want to verify: ")
print("You entered: " + str(var))


#function to run for prediction
def detecting_fake_news(var):
#retrieving the  model for prediction call
    load_model = pickle.load(open('final_model.sav', 'rb'))
    prediction = load_model.predict([var])
    prob = load_model.predict_proba([var])

    return (print("The given statement is ",prediction[0]),
        print("The truth probability score is ",prob[0][1]))


if __name__ == '__main__':
    detecting_fake_news(var)
```

VI. CONFUSION MATRIX

In order to **describe the performance of a classification model** or the "classifier" on a set of test data for which the true values are to be known, we use a confusion matrix (It's like a table). The confusion matrix though relatively simple to understand, but the terminology related to it can be confusing.

	Predicted Fake (0)	Predicted Real(1)
Actual Fake (0)	TRUE NEGATIVE	FALSE POSITIVE
Actual Real(1)	FALSE NEGATIVE	TRUE POSITIVE

True Positive: Interpretation: We predicted real and it's real. We predicted that the news is the real news and it actually is.
True Negative: Interpretation: We predicted fake and it's fake. We predicted that the news is not real and actually it is fake
False Positive: (Type 1 Error): Interpretation: We predicted real and it's fake. We predicted that the news is real but actually is not.
False Negative: (Type 2 Error): Interpretation: We predicted fake and it's real. We predicted that the news is fake but is real.

The following are the rates that are frequently calculated from a confusion matrix for a classifier which is binary:

N=165	Predicted NO	Predicted Yes	
Actual NO	TN=50	FP=10	(50+10)=60
Actual YES	FN=5	TP=100	(5+100)=105
	(50+5)=55	(10+100)=110	

- **Accuracy:** On comprehensive analysis, the rate at which the classifier is correct:
  - (TP+TN)/total = (100+50)/165 = 0.91

- **Misclassification Rate:** On comprehensive analysis, the rate at which the classifier is wrong:
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"

- **True Positive Rate:** When the assertion is true, the probability at which the prediction is true?
  - TP/actual yes = 100/105 = 0.95
  - also known as "Sensitivity" or "Recall"

- **False Positive Rate:** When the assertion is false, the probability at which the prediction is true?
  - FP/actual no = 10/60 = 0.17

- **True Negative Rate:** When the assertion is false, the probability at which the prediction is false?
  - TN/actual no = 50/60 = 0.83
  - equivalent to 1 minus False Positive Rate
  - also known as "Specificity"

- **Precision:** When the prediction is true, the rate at which the prediction is correct is
  - TP/predicted yes = 100/110 = 0.91

- **Prevalence:** How often does the yes condition actually occur in the sample?
  - actual yes/total = 105/165 = 0.64

VII. SCREENSHOTS

The following are the screenshots of train and test data

	A	B	C	D
1	Statement	Label		
2	Says the Annies List political group supports third-trimester abortions on demand.	FALSE		
3	When did the decline of coal start? It started when natural gas took off that started to begin in (President George W.) Bushs administration.	TRUE		
4	Hillary Clinton agrees with John McCain "by voting to give George Bush the benefit of the doubt on Iran."	TRUE		
5	Health care reform legislation is likely to mandate free sex change surgeries.	FALSE		
6	The economic turnaround started at the end of my term.	TRUE		
7	The Chicago Bears have had more starting quarterbacks in the last 10 years than the total number of tenured (UW) faculty fired during the last two decades.	TRUE		
8	Jim Dunnam has not lived in the district he represents for years now.	FALSE		
9	I'm the only person on this stage who has worked actively just last year passing, along with Russ Feingold, some of the toughest ethics reform since Watergate.	TRUE		
10	However, it took $19.5 million in Oregon Lottery funds for the Port of Newport to eventually land the new NOAA Marine Operations Center-Pacific.	TRUE		
11	Says GOP primary opponents Glenn Grothman and Joe Leibham cast a compromise vote that cost $788 million in higher electricity costs.	TRUE		
12	For the first time in history, the share of the national popular vote margin is smaller than the Latino vote margin.	TRUE		
13	Since 2000, nearly 12 million Americans have slipped out of the middle class and into poverty.	TRUE		
14	When Mitt Romney was governor of Massachusetts, we didnt just slow the rate of growth of our government, we actually cut it.	FALSE		
15	The economy bled $24 billion due to the government shutdown.	TRUE		
16	Most of the (Affordable Care Act) has already in some sense been waived or otherwise suspended.	FALSE		
17	In this last election in November, ... 63 percent of the American people chose not to vote, ... 80 percent of young people, (and) 75 percent of low-income workers	TRUE		
18	McCain opposed a requirement that the government buy American-made motorcycles. And he said all buy-American provisions were quote 'disgraceful.'	TRUE		
19	U.S. Rep. Ron Kind, D-Wis., and his fellow Democrats went on a spending spree and now their credit card is maxed out	FALSE		
20	Water rates in Manila, Philippines, were raised up to 845 percent when a subsidiary of the World Bank became a partial owner.	TRUE		
21	Almost 100,000 people left Puerto Rico last year.	TRUE		
22	Women and men both are making less when you adjust for inflation than when John Kitzhaber was first elected governor.	FALSE		
23	The United States has the highest corporate tax rate in the free world.	TRUE		

train

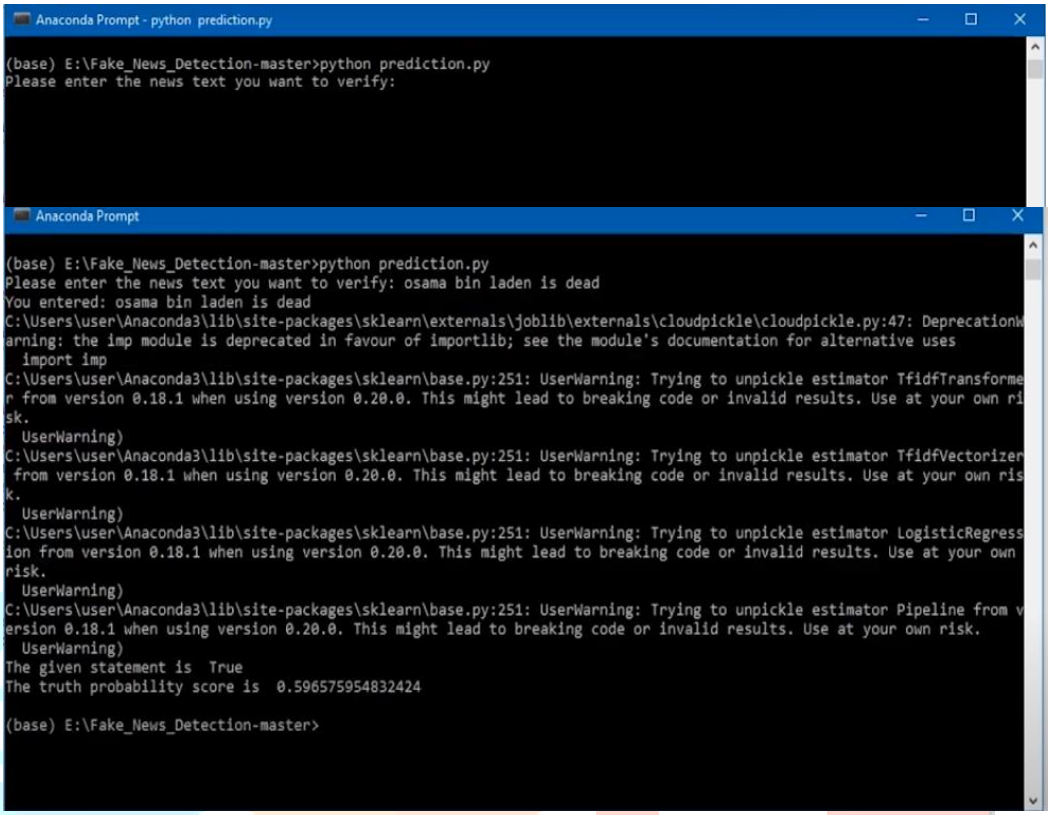
	A	B
804	Says the initial Portland plastic bag ban represented only a modest share of total single-use checkout bag use.	FALSE
805	The Bundy Ranch deal is all about Nevada Sen. Harry Reid using federal violence to take peoples land in his state so he can package it to re-sell it to the Chine	FALSE
806	There's no evidence anywhere that offshore drilling has hurt tourism in any area where it has been allowed.	FALSE
807	Says Wisconsin women facing pay discrimination cant do something about it under bill passed by Republicans.	FALSE
808	Under the new health care law, if a landscaper wants to buy a new lawnmower, or a restaurant needs a new ice-maker, they have to report that to the feds.	TRUE
809	Says that President Obama said if Congress passed the economic stimulus bill, we would have unemployment at 8 percent and no higher. And it went higher.	FALSE
810	Says Charlie Crist implemented Jeb Bushs A+ Plan.	TRUE
811	We have 650 people who move to Texas every day.	TRUE
812	By some estimates, as few as 2 percent of the 50,000 (Central American) children who have crossed the border illegally this year have been sent home.	TRUE
813	Says city of Portland has a one-time $22 million surplus	TRUE
814	I never gave up custody of my children. I never lost custody of my children.	TRUE
815	The New England Journal of Medicine released a survey the week that President Obama signed Obamacare stating that over 30 percent of American physician	FALSE
816	You can buy lobster with food stamps.	TRUE
817	I cut more as a percentage out of government than any state in the country this past decade. And where is Michigan in terms of its economic growth? Cutting	TRUE
818	The Obama administration gave Iran $400 million in ransom payment cash.	FALSE
819	I remember one of [Curt Schillings] teammates said he painted his sock, the bloody sock.	FALSE
820	Not one dime gets added to the deficit because of Social Security.	FALSE
821	The United States of America, right now, has the strongest, most durable economy in the world.	TRUE
822	Canada sets aside 36 percent of their visas for people with skills they think their country needs. We set aside 6 percent.	FALSE
823	The federal health care law is probably the biggest tax increase ever in the history of our country.	FALSE
824	There are 60,000 fewer jobs today in this state than we had in 2008.	TRUE
825	On toll roads.	TRUE
826	Says Republicans supported legislation on early voting and in-person voting in 2005.	TRUE

test

VIII. OBSERVED RESULTS:

The following are the results obtained in terms of metrics like Accuracy, Recall and Precision.

METRICS	SVM model
Accuracy	97%
Precision	0.98
Recall	0.97



IX. CONCLUSION

With the increasing popularity of social media, more and more people are interested and attracted to the news from social media rather than traditional news media. Nevertheless, social media has always been a source to spread slanting, fake and misleading news, which has strong negative impacts on not only the individual users but also on the broader society. The rise of fake news has become a global problem that even the leading Tech companies like Facebook and Google are struggling to solve. It is not so easy to determine whether a text is factual or not without the additional context and the human judgment. This proposed Project helps in identifying the misinformation.

X. REFERENCES

[1]. Srijan Kumar and Neil Shah. 2018. False Information on Web and Social Media: A Survey. 1, 1 (April 2018)
[2]. Zhang, J., Dong, B., & Yu, P. S. (2020). FakeDetector: Effective Fake News Detection with Deep Diffusive Neural Network. 2020 IEEE 36th International Conference on Data Engineering (ICDE). doi:10.1109/icde48307.2020.00180