COMPARATIVE STUDY ON FAKE NEWS ACCURACY PREDICTION USING NAÏVE BAYES, SVM AND ANN

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

By

Balaga Susmitha
16JG1A0515

Dasari Poornima
16JG1A0524

Ayyalasomayajula Akshata
16JG1A0511

Koppala Likhita
16JG1A0560

Under the esteemed guidance of

MRS. K. SUNEETHA

(Assistant Professor)

CSE Department

Department of Computer Science and Engineering

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

2016-20

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project report titled “Comparative Study on Fake News Accuracy Prediction using Naïve Bayes, SVM and ANN” is a bonafide work of following IV/IV B.Tech students in the Department of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering for Women affiliated to JNT University, Kakinada during the academic year 2019-20, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology of this university.

Ms. B. SUSMITHA (16JG1A0515)	Ms. D. POORNIMA (16JG1A0524)
Ms. A. AKSHATA (16JG1A0511)	Ms. K. LIKHITA (16JG1A0560)

Mrs. K. Suneetha	Mrs.P.V.S. Lakshmi Jagadamba
Assistant Professor	Associate Professor
Internal Guide	**Head of the Department**

External Examiner

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude to our esteemed institute "**Gayatri Vidya Parishad College of Engineering for Women",** which has provided us an opportunity to fulfill our cherished desire.

Our sincere thanks to our guide **Mrs. K.Suneetha**, Assistant Professor of Computer Science and Engineering for her stimulating guidance and profuse assistance from the beginning of the project.

Our sincere thanks to **Mrs.Dr. P.V.S.L.Jagadamba**, Head of the Department of Computer Science and Engineering for her motivation.

We would like to take this opportunity to express our profound sense of gratitude to our Vice Principal **Prof. G. Sudheer** for allowing us to utilize the college resources there by facilitating the successful completion of our project.

Our sincere thanks to our beloved principal **Prof. DR. K.V.S.RAO** for providing the best faculty and lab facility throughout these academic years.

Our sincere thanks to our beloved Director **Prof. DR. E.V. PRASAD** for providing the best faculty and lab facility through these academic years.

We would like to thank all the members of teaching and lab technicians of the Computer Science and Engineering for all their support in completion of our project.

CONTENTS

Abstract **i**

List of Figures **ii**

List of Screenshots **iii**

1. Introduction……………………………………………………………………**1**

    1.1 Problem definition……………………………………………………….....1

    1.2 Motivation……………………………………………………………………1

    1.3 Objective of Project……………………………………………………........1

    1.4 Limitations of the Project………………………………………………......1

2. LITERATURE SURVEY…………………………………………………………**2**

    2.1 Introduction……………………………………………………………….....2

    2.2 Existing System…………………………………………………………….....2

    2.3 Disadvantages of the Existing system……………………………………....3

    2.4 Proposed System………………………………………………………….....3

    2.5 Applications of the Project………………………………………………….3

    2.6 Advantages of the proposed system……………………………………….....3

3. REQUIREMENT ANALYSIS………………………………………………….**4**

    3.1 Introduction……………………………………………………………………..4

    3.2 Software Requirement Specification……………………………………….....4

        3.2.1 Software requirement……………………………………………….....5

        3.2.2 Hardware requirement……………………………………………….....5

    3.3 Content Diagram of Project…………………………………………………...6

    3.4 Algorithms ad Flowcharts……………………………………………………7

        3.4.1 Naïve Bayes Algorithm………………………………………………7

        3.4.2 Support Vector Machine………………………………………………9

        3.4.3 Artificial Neural Networks……………………………………………10

    3.5 Sample Code……………………………………………………………….…13

    3.6 Conclusion…………………………………………………………………….20

4. DESIGN…………………………………………………………………………………**21**

4.1 Introduction………………………………………………………………………21

4.2 UML diagrams……………………………………………………………………22

4.2.1 Usecase Diagram…………………………………………………………23

4.2.2 Class Diagram……………………………………………………………..25

4.2.3 Sequence Diagram…………………………………………………………...26

4.2.4 Activity Diagram…………………………………………………………..28

4.2.5 State Chart Diagram………………………………………………………….30

4.3 Conclusion…………………………………………………………………………31

5. IMPLEMENTATION & RESULTS……………………………………………….**32**

5.1 Introduction…………………………………………………………………………..32

5.2 Explanation of Key functions………………………………………………………...32

5.3 Method of Implementation……………………………………………………….....34

5.3.1Python Programming Language………………………………………………34

5.3.2 Python Programming Features………………………………………………35

5.3.3 Python Packages………………………………………………………………...36

5.4 Output Screens………………………………………………………………………....37

6. TESTING & VALIDATION………………………………………………………**43**

6.1 Introduction…………………………………………………………………………….43

6.2 Design of test cases and scenarios………………………………………………….....44

6.3 Acceptance Testing…………………………………………………………………….46

6.4 Validation………………………………………………………………………………47

7. CONCLUSION…………………………………………………………………...**48**

8. REFERENCES……………………………………………………………………**49**

ABSTRACT

Fake news is one of the biggest banes in our digitally connected world. Fake news spreads like wildfire and is impacting millions of people everyday .With the advancement of digitized society, information is free for everyone and it can be manipulated by anyone. This lead to the enlargement in the spread of fake news and misinformation leading to many problems. In order to filter the information and to verify the trustworthiness of a news, we need a freely available tool. Our dataset is initially preprocessed and then it is tested using Machine Learning algorithms such as Naïve Bayes, SVM and ANN to predict the accuracy and compare them.

LIST OF FIGURES

S.No.	Fig No.	Figure Name	Page No.
1	Figure 1	Content Flow Diagram	6
2	Figure 2	Support Vector Machine Output Model	9
3	Figure 3	Hyper planes in 2D and 3D feature space	9
4	Figure 4	Neural Networks	11
5	Figure 5	Usecase Diagram	25
6	Figure 6	Class Diagram	26
7	Figure 7	Sequence Diagram	27
8	Figure 8	Activity Diagram	29
9	Figure 9	State Chart Diagram	30
10	Figure 10	Dataset	32

LIST OF SCREENSHOTS

S.No.	Screen No.	Screenshot Name	Page Number
1	Screen 1	Accuracy using Naïve Bayes Algorithm	37
2	Screen 2	Accuracy using Support Vector Machine	38
3	Screen 3	Accuracy using Artificial Neural Networks	38
4	Screen 4	Confusion Matrix for Naïve Bayes	39
5	Screen 5	Confusion Matrix for SVM	39
6	Screen 6	Confusion Matrix for ANN	40
7	Screen 7	Front Page	40
8	Screen 8	Naïve Bayes Result	41
9	Screen 9	SVM Result	41
10	Screen 10	ANN Result	42
11	Screen 11	Accuracy Plot	42

1. INTRODUCTION

1.1 Problem Definition

Develop a machine learning program to identify when a news source may be producing fake news. We aim to use a corpus of labeled real and fake news articles to build a classifier that can make decisions about information based on the content from the corpus. The model will focus on identifying fake news sources, based on multiple articles originating from a source. Once a source is labeled as a producer of fake news, we can predict with high confidence that any future articles from that source will also be fake news. Focusing on sources widens our article misclassification tolerance, because we will have multiple data points coming from each source.

1.2 Motivation

Fake news is an inaccurate, sometimes sensationalistic report that is created to gain attention, mislead, deceive or damage a reputation. Unlike misinformation, which is inaccurate because a reporter has confused facts, fake news is created with the intent to manipulate someone or something. Fake news can spread quickly. Social media websites in particular have proved to be an easy venue for distributing fake news. When fake news is used to spread propaganda, it can be dangerous. In addition to shaping public opinion and behaviour, it can also cause mistrust, encourage dissent and deflect attention from real news. In response to this spread of fake news, here we are going to implement a project on detecting fake news to protect ourselves from wrong information.

1.3 Objective of the Project

The topic of fake news detection on social media has recently attracted tremendous attention. The basic countermeasure of comparing websites against a list of labeled fake news sources is inflexible, and so a machine learning approach is desirable. Our project aims to use Naïve Bayes, Support Vector Machine and Artificial Neural Networks to obtain the accuracy and compare with each other.

1.4 Limitations of the Project

- Number of observations are less and corpus of words/vocabulary is created only on the basis of these articles.
- Model might not perform properly in case of news articles which doesn’t contain the words on which it is trained.
- Model works only on ‘proper’ English language articles.

2. LITERATURE SURVEY

2.1 Introduction

Fake news is the deliberate spread of misinformation via traditional news media or via social media. False information spreads extraordinarily fast. This is demonstrated by the fact that, when one fake news site is taken down, another will promptly take its place. In addition, fake news can become indistinguishable from accurate reporting since it spreads so fast. People can download articles from sites, share the information, re-share from others and by the end of the day the false information has gone so far from its original site that it becomes indistinguishable from real news. This is an advancement in human history, but at the same time it blurs the line between true media and maliciously fabricated generated media. A freely available tool to verify the trustworthiness of a news is needed to filter the information we receive everyday. With this motivation, through building our knowledge of python and machine learning, we worked on our final project. Given the text of a news article and it's headline as input, we have developed an algorithm that can litigate the difference between the fake and true . This project can consists of numerous binary classification (true/fake) and output is generated based on algorithms.There are 5 different types of fake news as classified below based on its nature.

- Satire
- Clickbait
- Conspiracy Theory
- Misleading or out-of-context information
- Propaganda

2.2 Existing System

BS Detector

BS Detector is a plug-in used by Mozilla and Chrome browsers to detect the presence of fake news sources and to alert the user accordingly. It works by searching through web pages references of links which have already been flagged unreliable in their database. BS Detector has been used by Facebook to solve their proliferation of fake news problem. But lately, they blocked the extension stating that they have been working on their own technique to curb the problem. BS Detector just states a warning message if the article is found to be fake. It does not specify the percentage of error and neither does it classify news into levels of "truthfulness" or "fakeness".

There exists a large body of research on the topic of machine learning methods for deception detection, most of it has been focusing on classifying online reviews and publicly available social media posts. Particularly since late 2016 during the American Presidential election, the question of determining 'fake news' has also been the subject of particular attention within the literature. Facebook has a very real fake news problem. To help combat this, B.S. Detector will show a little red warning when you're about to click a link that comes from a questionable, "satirical" or fake news source. While most of us are already programmed to ignore obviously fake tabloids at the supermarket, we're still adjusting to the era of fake news online. It's just as easy to share an article from a site created by a Macedonian teen as it is to share from the Washington Post. If you're not used to checking the reliability of news sources, it might be easy to get duped. B.S. Detector aims to help make it easier to distinguish the wheat from the chaff. The extension is relatively simple, using a basic list of fake news sources as its reference point. You can find many similar lists online. As with anything, you probably shouldn't take B.S. Detector as sacred even the extension can have false positives-but if you get the warning while using it, maybe look for confirmation from another outlet before trusting what you read. Or do that no matter what the extension says. It's just good practice.

2.3 Disadvantages of the Existing System

- The main disadvantage of B.S.Detector is that it only displays the pop-up message to the user .
- It doesn't predict the accuracy of the fake news dataset.

2.4 Proposed System

The project comes up with a proposed work on pre-processing a dataset of both fake and real news and employ a Machine Learning Algorithm or Artificial Neural Network in order to create a model that predicts the accuracy based on its words and phrases.

2.5 Applications of the Project

This project has its applications in various fields as follows

1) Business

2) Social Networks

3) Security Agencies

2.6 Advantages of the proposed System

The main advantage of this system is to explore how artificial intelligence technologies, particularly Machine Learning and Artificial Neural Networks might be used to detect the fake news accuracy.

3. ANALYSIS

3.1 Introduction

The project comes up with a proposed work on assembling a dataset of both fake and real news and employ a Naive Bayes Classifier, Support vector machine, Artificial Neural networks in order to create a model to classify a news into fake or real and thus the accuracy is generated.

3.2 Software Requirement Specification

Specific requirements

The following list offers a brief outline and description of the main features and functionalities of the Fake News Accuracy Prediction. The requirements may be functional or non- functional.

Functional Requirements

This project detects accuracy based on whether the news is fake or real. After that we apply models like Naive Bayes, SVM and Artificial Neural Networks.

The response/stimulus for different class of news are:

1. Input the data

2. Apply the algorithm

3. Visualize the model

4. Get output

The Fake News Accuracy Prediction is useful for many users. The system should detect news based on the model we train.

Non-Functional Requirements

- **Usability**

  1. It is used in many fields.

  2. It is used in Investigations.

  3. It is used in Crime Prediction.

- **Reliability**

  1. It should be accurate as much as possible.
  2. It should predict the news as simple as possible.

- **Performance**

  1. It must be able to perform in adverse conditions.
  2. Must have high data transfer rate.

- **Supportability**

  1. It should be flexible for Testing.
  2. It should be adaptable to any type of news.
  3. It should be Robust.

3.2.1 Software Requirements

- LANGUAGE: Python 3.7
- TOOL: Rational Rose
- PACKAGES : Numpy, Pandas, Util, Gensim

3.2.2 Hardware Requirements

- OPERATING SYSTEM: Windows 8.1
- RAM: 4GB

3.3 CONTENT FLOW DIAGRAM

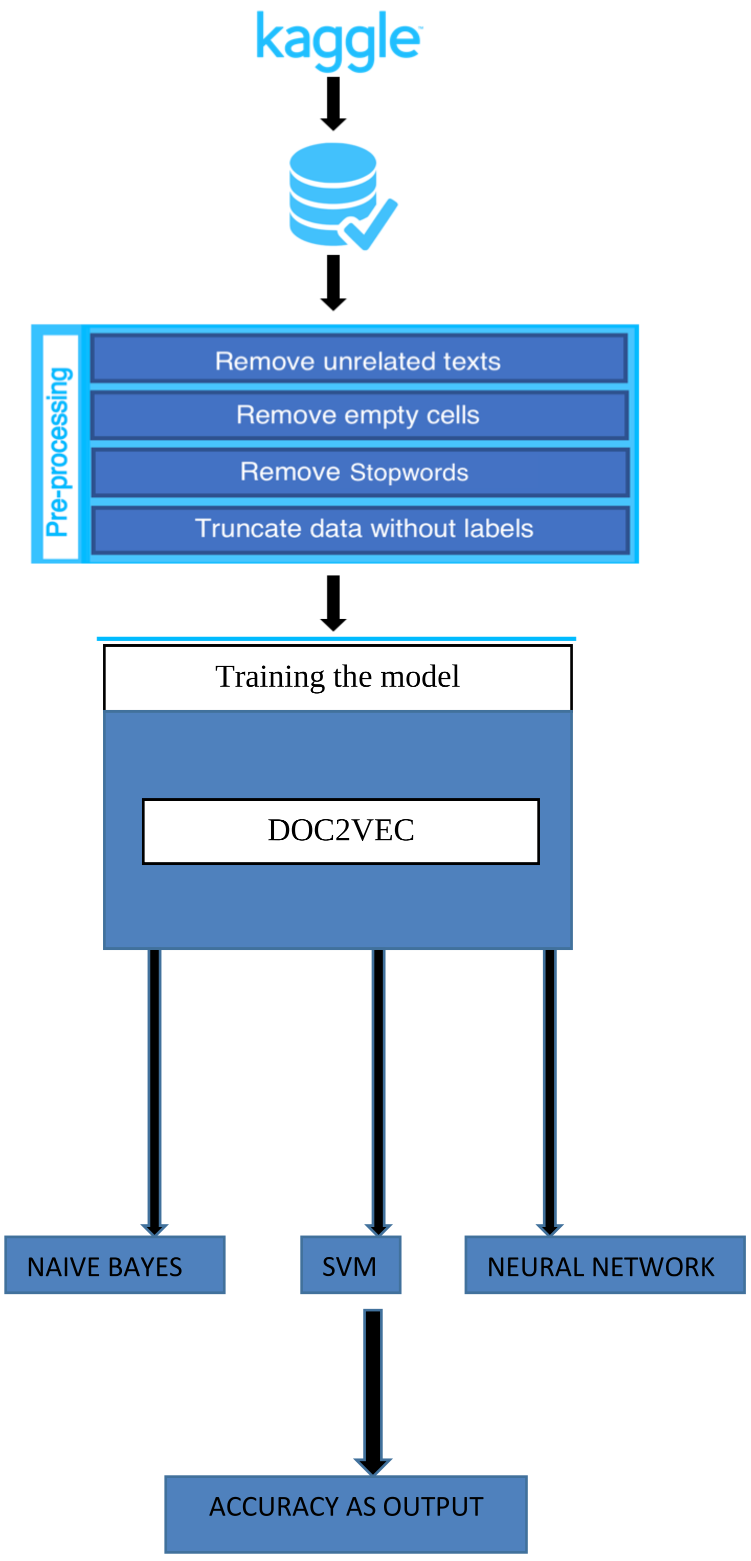


Fig 1 : Content Flow Diagram

3.4 ALGORITHMS AND FLOWCHART

- NAÏVE BAYES ALGORITHM
- SUPPORT VECTOR MACHINE ALGORITHM
- ARTIFICIAL NEURAL NETWORKS

3.4.1 NAIVE BAYES ALGORITHM

Bayes Theorem:

Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. It is based on the Bayes theorem.Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier. Naive Bayes algorithm can be defined as a supervised classification algorithm which is based on Bayes theorem with an assumption of independence among features.

Bayes Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where A and B are events and P(B) is not equal to 0.

- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.
- P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance (here, it is event B).

- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Now, with regards to our dataset, we can apply Bayes' theorem in following way: where, y is class variable and X is a dependent feature vector (of size *n*) where:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Gaussian Naive Bayes:

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.

Pros and Cons of Naive Bayes Algorithm:

Pros :

- It is easy to understand.
- It can also be trained on small dataset.

Cons :

- It has a 'Zero conditional probability Problem', for features having zero frequency the total probability also becomes zero.There are several sample correction techniques to fix this problem such as "Laplacian Correction".
- Another disadvantage is the very strong assumption of independence class features that it makes. It is near to impossible to find such data sets in real life.

Applications of Naive Bayes Algorithm :

1. Naive Bayes is widely used for text classification.

2. Another example where Naive Bayes is mostly used is Spam Filtering in Emails.

3. Real time Prediction

4. Multi class Prediction

5. Spam Filtering

3.4.2 **SUPPORT VECTOR MACHINE**

The objective of the support vector machine algorithm is to find a hyper plane in an N-dimensional space (N is the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyper planes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyper planes are decision boundaries that help classify the data points. Data points falling on either side of the hyper plane can be attributed to different classes. Also, the dimension of the hyper plane depends upon the number of features. If the number of input features is 2, then the hyper plane is just a line. If the number of input features is 3, then the hyper plane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

Support vectors are data points that are closer to the hyper plane and influence the position and orientation of the hyper plane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyper plane.

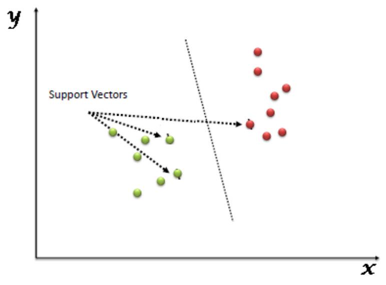


Fig 2 : SVM Output Model

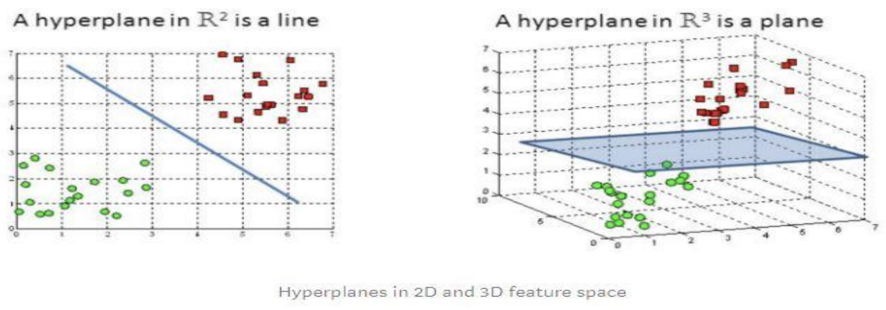


Fig 3: Hyper planes in 2D and 3D feature space

In Python, scikit-learn is a widely used library for implementing machine learning algorithms. SVM is also available in the scikit-learn library and we follow the same structure for using it.

The two main advantages of support vector machines are:

1. They’re accurate in high dimensional spaces.
2. They use a subset of training points in the decision function (called support vectors), so it’s also memory efficient.

The disadvantages of support vector machines are:

1. The algorithm is prone for over-fitting, if the number of features is much greater than the number of samples.
2. Also, SVMs do not directly provide probability estimates, which are desirable in most classification problems.
3. And finally, SVMs are not very efficient computationally, if your dataset is very big, such as when you have more than one thousand rows.

Applications of SVM:

- Face Detection
- Text and Hypertext Categorization
- Classification Of Images
- BioInformatics

3.4.3 Artificial Neural Networks :

Artificial neural networks are one of the various data mining techniques used to forecast the power output of a wind farm using meteorological information predicted by NWP models.
ANNs attempt to copy the behavior of biological neural networks. In analogy to the structure of the brain, ANNs consist of single processing units called neurons. In the network structure, the neurons are arranged in layers. Each of the neurons in the input layer receives one of the variables (e.g., wind speed and direction, humidity, temperature, and atmospheric pressure) on which the variables that we wish to forecast depend.

The neurons of the output layer return the values of the variables that we wish to forecast (e.g., the power output of the wind farm at subsequent instants). There can also be a series of intermediate layers, called hidden layers. The manner in which the neurons interconnect is known as the connectivity pattern or architecture of the network.

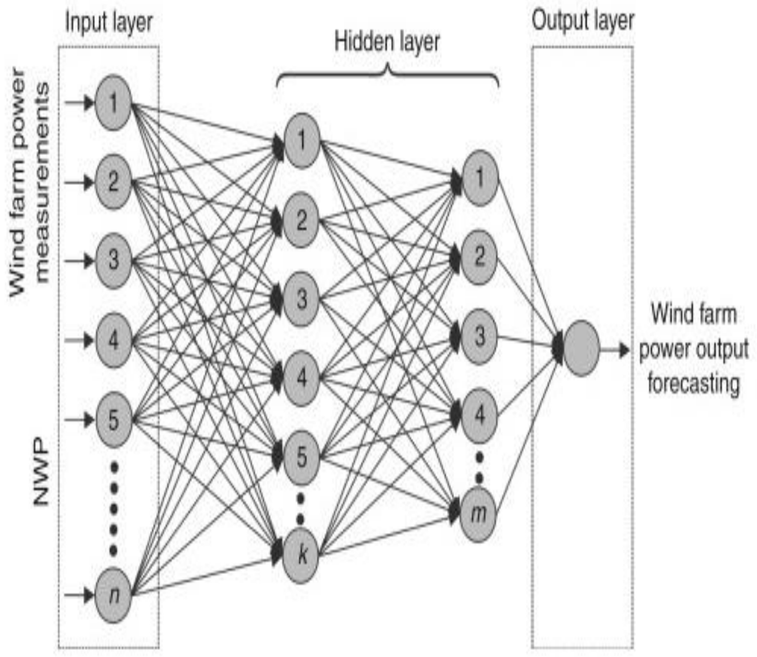

Input layer
Hidden layer
Output layer
Wind farm power measurements
NWP
1
2
3
4
5
n
1
2
3
4
5
k
1
2
3
4
m
Wind farm power output forecasting

Fig 4 : Neural Networks

Pros:

- Storing information on the entire network: Information such as in traditional programming is stored on the entire network, not on a database. The disappearance of a few pieces of information in one place does not restrict the network from functioning.
- The ability to work with inadequate knowledge: After ANN training, the data may produce output even with incomplete information. The lack of performance here depends on the importance of the missing information.
- It has fault tolerance: Corruption of one or more cells of ANN does not prevent it from generating output. This feature makes the networks fault-tolerant.

Cons:

- Hardware dependence: Artificial neural networks require processors with parallel processing power, by their structure. For this reason, the realization of the equipment is dependent.
- Unexplained functioning of the network: This is the most important problem of ANN. When ANN gives a probing solution, it does not give a clue as to why and how. This reduces trust in the network.
- Assurance of proper network structure: There is no specific rule for determining the structure of artificial neural networks. The appropriate network structure is achieved through experience and trial and error.

Applications:

- Image Processing and Character recognition
- Forecasting
- Text Classification and Categorization
- Named Entity Recognition
- Speech Recognition

3.5 SAMPLE CODE

<u>Preprocessing Code</u>

```
import numpy as np

import re

import string

import pandas as pd

from gensim.models import Doc2Vec

from gensim.models.doc2vec import LabeledSentence

from gensim import utils

from nltk.corpus import stopwords

def textClean(text):

    text = re.sub(r"[^A-Za-z0-9^,!.\/'+-=]", " ", text)

    text = text.lower().split()

    stops = set(stopwords.words("english"))

    text = [w for w in text if not w in stops]

    text = " ".join(text)

    return (text)


def cleanup(text):

    text = textClean(text)

    text = text.translate(str.maketrans("", "", string.punctuation))

  return text
```

```
def constructLabeledSentences(data):
sentences = []
    for index, row in data.iteritems():
        sentences.append(LabeledSentence(utils.to_unicode(row).split(), ['Text' + '_%s' % str(index)]))
    return sentences


def getEmbeddings(path,vector_dimension=300):
    data = pd.read_csv(path)
    missing_rows = []
    for i in range(len(data)):
        if data.loc[i, 'text'] != data.loc[i, 'text']:
            missing_rows.append(i)
    data = data.drop(missing_rows).reset_index().drop(['index','id'],axis=1)
    for i in range(len(data)):
        data.loc[i, 'text'] = cleanup(data.loc[i,'text'])
    x = constructLabeledSentences(data['text'])
    y = data['label'].values


    text_model = Doc2Vec(min_count=1, window=5, vector_size=vector_dimension, sample=1e-4,
negative=5, workers=7, epochs=10, seed=1)
    text_model.build_vocab(x)
    text_model.train(x, total_examples=text_model.corpus_count, epochs=text_model.iter)


    train_size = int(0.8 * len(x))
    test_size = len(x) - train_size
    text_train_arrays = np.zeros((train_size, vector_dimension))

    text_test_arrays = np.zeros((test_size, vector_dimension))

    train_labels = np.zeros(train_size)

    test_labels = np.zeros(test_size)

    for i in range(train_size):

        text_train_arrays[i] = text_model.docvecs['Text_' + str(i)]

        train_labels[i] = y[i]

    j = 0

    for i in range(train_size, train_size + test_size):

        text_test_arrays[j] = text_model.docvecs['Text_' + str(i)]

        test_labels[j] = y[i]

        j = j + 1

    return text_train_arrays, text_test_arrays, train_labels, test_labels
```

Naive Bayes Code

```
from getEmbeddings import getEmbeddings

from sklearn.naive_bayes import GaussianNB

import numpy as np

import matplotlib.pyplot as plt

import scikitplot.plotters as skplt

def plot_cmat(yte, ypred):

    '''Plotting confusion matrix'''

    skplt.plot_confusion_matrix(yte,ypred)

    plt.show()
```

```
xtr,xte,ytr,yte = getEmbeddings("datasets/train.csv")
    np.save('./xtr', xtr)
    np.save('./xte', xte)
    np.save('./ytr', ytr)
    np.save('./yte', yte)
    xtr = np.load('./xtr.npy')
    xte = np.load('./xte.npy')
    ytr = np.load('./ytr.npy')
    yte = np.load('./yte.npy')
    gnb = GaussianNB()
    gnb.fit(xtr,ytr)
    y_pred = gnb.predict(xte)
    m = yte.shape[0]
    n = (yte != y_pred).sum()
    print("Accuracy = " + format((m-n)/m*100, '.2f') + "%")
    plot_cmat(yte, y_pred)
```

Support Vector Machine Code

```
from getEmbeddings import getEmbeddings
import numpy as np
from sklearn.svm import SVC
import matplotlib.pyplot as plt
import scikitplot.plotters as skplt
```

```
def plot_cmat(yte, ypred):
    skplt.plot_confusion_matrix(yte,ypred)
    plt.show()


xtr,xte,ytr,yte = getEmbeddings("datasets/train.csv")
np.save('./xtr', xtr)
np.save('./xte', xte)
np.save('./ytr', ytr)
np.save('./yte', yte)


xtr = np.load('./xtr.npy')
xte = np.load('./xte.npy')
ytr = np.load('./ytr.npy')
yte = np.load('./yte.npy')


clf = SVC()
clf.fit(xtr, ytr)
y_pred = clf.predict(xte)
m = yte.shape[0]
n = (yte != y_pred).sum()
print("Accuracy = " + format((m-n)/m*100, '.2f') + "%")
plot_cmat(yte, y_pred)
```

Artificial Neural Networks Code

```
from getEmbeddings import getEmbeddings

import matplotlib.pyplot as plt

import numpy as np

import keras

from keras import backend as K

from keras.utils import np_utils

from keras.models import Sequential

from keras.layers import Dense, Dropout, LSTM, Embedding, Input, RepeatVector

from keras.optimizers import SGD

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

import scikitplot.plotters as skplt


def plot_cmat(yte, ypred):

    skplt.plot_confusion_matrix(yte, ypred)

    plt.show()

xtr,xte,ytr,yte = getEmbeddings("datasets/train.csv")

np.save('./xtr', xtr)

np.save('./xte', xte)

np.save('./ytr', ytr)

np.save('./yte', yte)


xtr = np.load('./xtr.npy')
```

```
xte = np.load('./xte.npy')

ytr = np.load('./ytr.npy')

yte = np.load('./yte.npy')

def baseline_model():

   model = Sequential()

   model.add(Dense(256, input_dim=300, activation='relu', kernel_initializer='normal'))

   model.add(Dropout(0.3))

   model.add(Dense(256, activation='relu', kernel_initializer='normal'))

   model.add(Dropout(0.5))

   model.add(Dense(80, activation='relu', kernel_initializer='normal'))

   model.add(Dense(2, activation="softmax", kernel_initializer='normal'))


   sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)


   model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])

   return model

model = baseline_model()

model.summary()

x_train, x_test, y_train, y_test = train_test_split(xtr, ytr, test_size=0.2, random_state=42)

label_encoder = LabelEncoder()

label_encoder.fit(y_train)

encoded_y = np_utils.to_categorical((label_encoder.transform(y_train)))

label_encoder.fit(y_test)

encoded_y_test = np_utils.to_categorical((label_encoder.transform(y_test)))

estimator = model.fit(x_train, encoded_y, epochs=20, batch_size=64)
```

```
print("Model Trained!")

score = model.evaluate(x_test, encoded_y_test)

print("")

print("Accuracy = " + format(score[1]*100, '.2f') + "%")

probabs = model.predict_proba(x_test)

y_pred = np.argmax(probabs, axis=1)

plot_cmat(y_test, y_pred)
```

3.6 CONCLUSION :

To detect the best accurate algorithm that can be used in detection of fake news i.e., the accuracy of each algorithm is determined and compared , the algorithm with highest accuracy rate is the best served algorithm.

4.DESIGN

4.1 INTRODUCTION

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design. It was developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in 1994–1995, with further development led by them through 1996.

In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005, UML was also published by the International Organization for Standardization (ISO) as an approved ISO standard. Since then the standard has been periodically revised to cover the latest revision of UML.

Basic building blocks of UML

As UML describes the real-time systems, it is very important to make a conceptual model and then proceed gradually. The conceptual model of UML can be mastered by learning the following three major elements −

- UML building blocks
- Rules to connect the building blocks
- Common mechanisms of UML

This chapter describes all the UML building blocks. The building blocks of UML can be defined as −

- Things
- Relationships
- Diagrams

UML includes the following nine diagrams :

- Class diagram
- Object diagram
- Use Case diagram

- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2 UML DIAGRAMS

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

UML Relationships depict a connection between several things, such as structural, behavioral, or grouping things in the unified modeling language. Since it is termed as a link, it demonstrates how things are interrelated to each other at the time of system execution. It constitutes four types of relationships, i.e., dependency, association, generalization, and realization.

Dependency

Whenever there is a change in either the structure or the behavior of the class that affects the other class, such a relationship is termed as a dependency. Or, simply, we can say a class contained in other class is known as dependency. It is a unidirectional relationship.

Association

Association is a structural relationship that represents how two entities are linked or connected to each other within a system. It can form several types of associations, such as one-to-one, one-to-many, many-to-one, and many-to-many. A ternary association is one that constitutes three links. It portrays the static relationship between the entities of two classes.

Aggregation

An aggregation is a special form of association. It forms a binary relationship, which means it cannot include more than two classes. It is also known as Has-a relationship. It specifies the direction of an object contained in another object. In aggregation, a child can exist independent of a parent.

Composition

In a composition relationship, the child depends on the parent. It forms a two-way relationship. It is a special case of aggregation. It is known as Part-of relationship.

Generalization

The generalization relationship implements the object-oriented concept called inheritance or is-a relationship. It exists between two objects (things or entities), such that one entity is a parent (super class or base class), and the other one is a child (subclass or derived class). These are represented in terms of inheritance. Any child can access, update, or inherit the functionality, structure, and behavior of the parent.

Realization

It is a kind of relationship in which one thing specifies the behavior or a responsibility to be carried out, and the other thing carries out that behavior. It can be represented on a class diagram or component diagrams. The realization relationship is constituted between interfaces, classes, packages, and components to link a client element to the supplier element.

4.2.1 Use Case Diagram

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.
When the initial task is complete, use case diagrams are modelled to present the outside view.
In brief, the purposes of use case diagrams can be said to be as follows −

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities.

They also help identify any internal or external factors that may influence the system and should be taken into consideration. They provide a good high level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

- Functionalities to be represented as use case
- Actors
- Relationships among the use cases and actors.

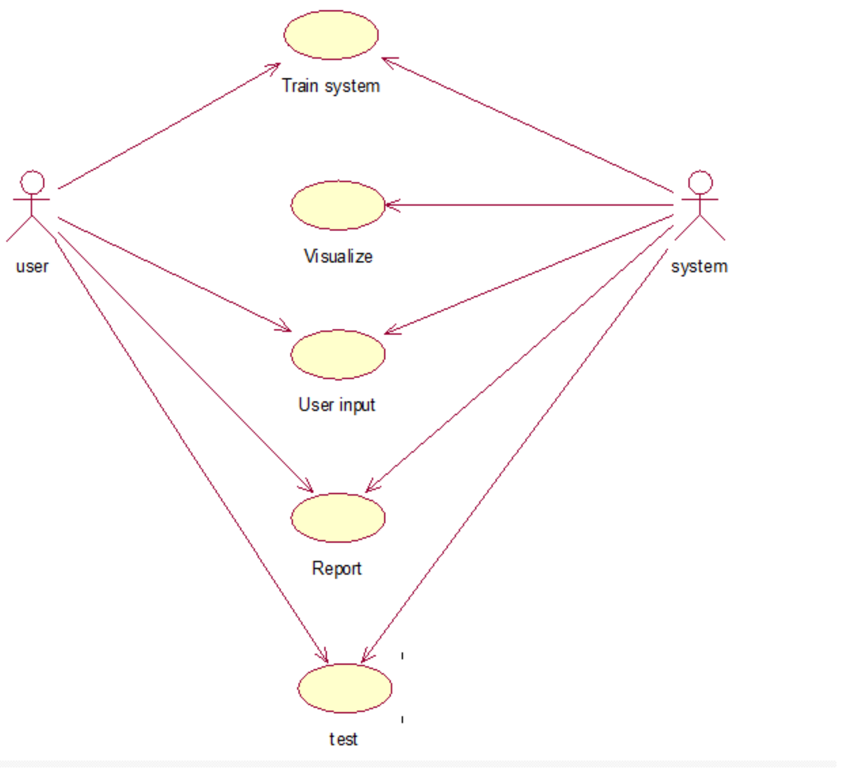


Fig 5: Use Case Diagram

4.2.2 Class Diagram

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

The purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

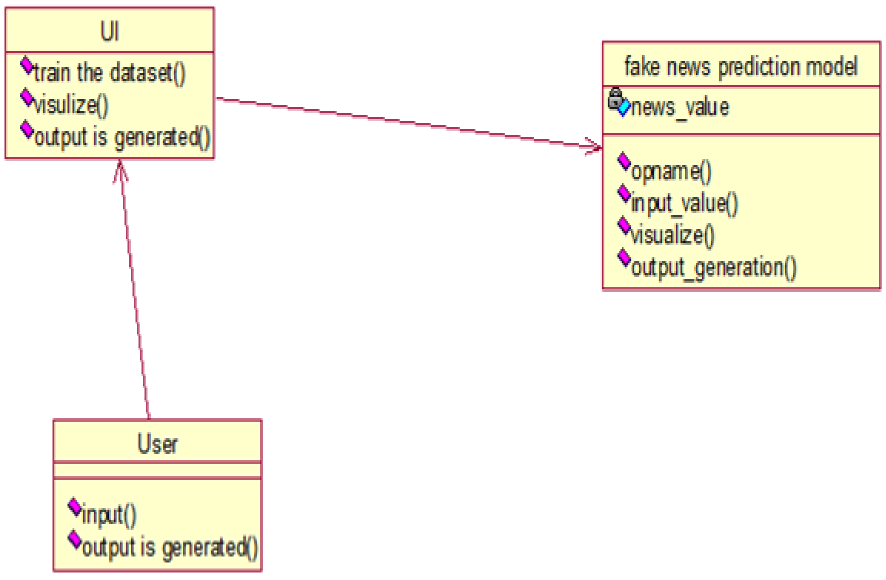


Fig 6 : Class Diagram

4.2.3 Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

- To model high-level interaction among active objects within a system.
- To model interaction among objects inside a collaboration realizing a use case.
- It either models generic interactions or some certain instances of interaction.

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

Uses of sequence diagrams

- Used to model and visualise the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.

- Visualize how messages and tasks move between objects or components in a system.

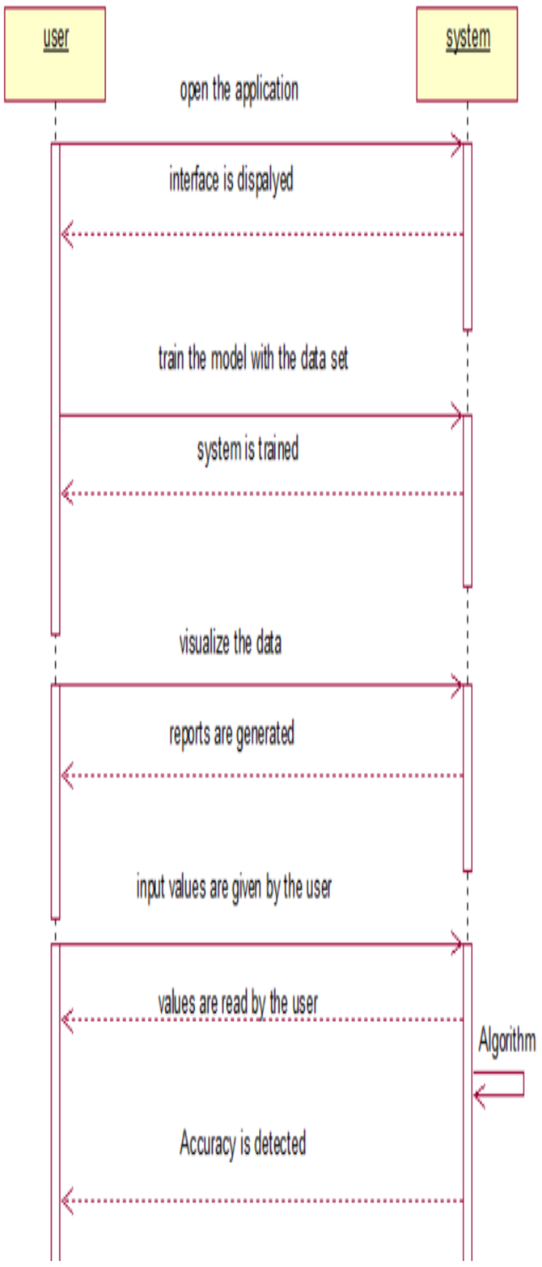


Fig 7 : Sequence Diagram

4.2.4 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join.

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

How to Draw an Activity Diagram?

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.

Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements −

- Activities

- Association
- Conditions
- Constraints

Once the above-mentioned parameters are identified, we need to make a mental layout of the entire flow. This mental layout is then transformed into an activity diagram.

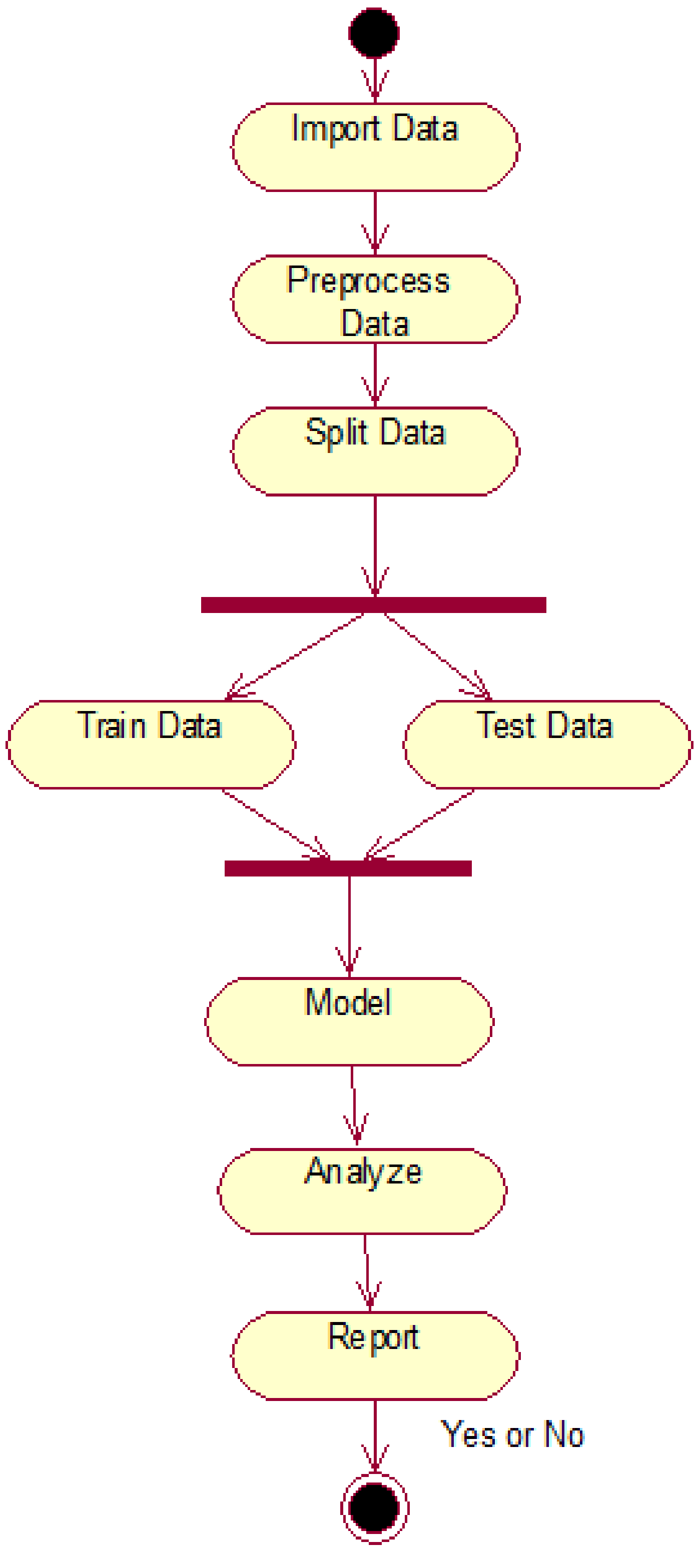


Fig 8 : Activity Diagram

4.2.5 State Chart Diagram

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using State chart diagrams −

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

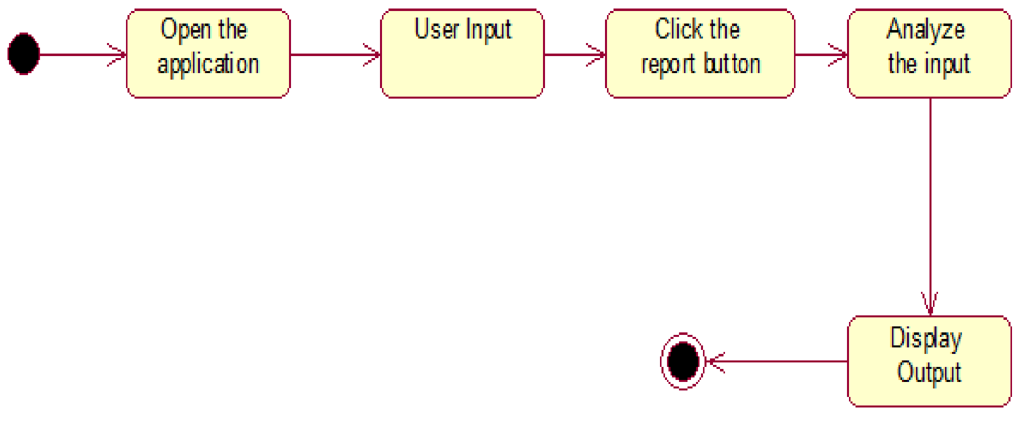


Fig 9: State Chart Diagram

The first state is an idle state from where the process starts. The next states are arrived for events like send request, confirm request, and dispatch order. These events are responsible for the state changes of order object.

During the life cycle of an object (here order object) it goes through the following states and there may be some abnormal exits. This abnormal exit may occur due to some problem in the system. When the entire life cycle is complete, it is considered as a complete transaction as shown in the following figure.

4.3 CONCLUSION

UML diagrams are not only made for developers but also for business users, common people, and anybody interested to understand the system. The system can be a software or non-software system. Thus it must be clear that UML is not a development method rather it accompanies with processes to make it a successful system. In conclusion, the goal of UML can be defined as a simple modeling mechanism to model all possible practical systems in today’s complex environment.

5. IMPLEMENTATION & RESULTS

5.1 INTRODUCTION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

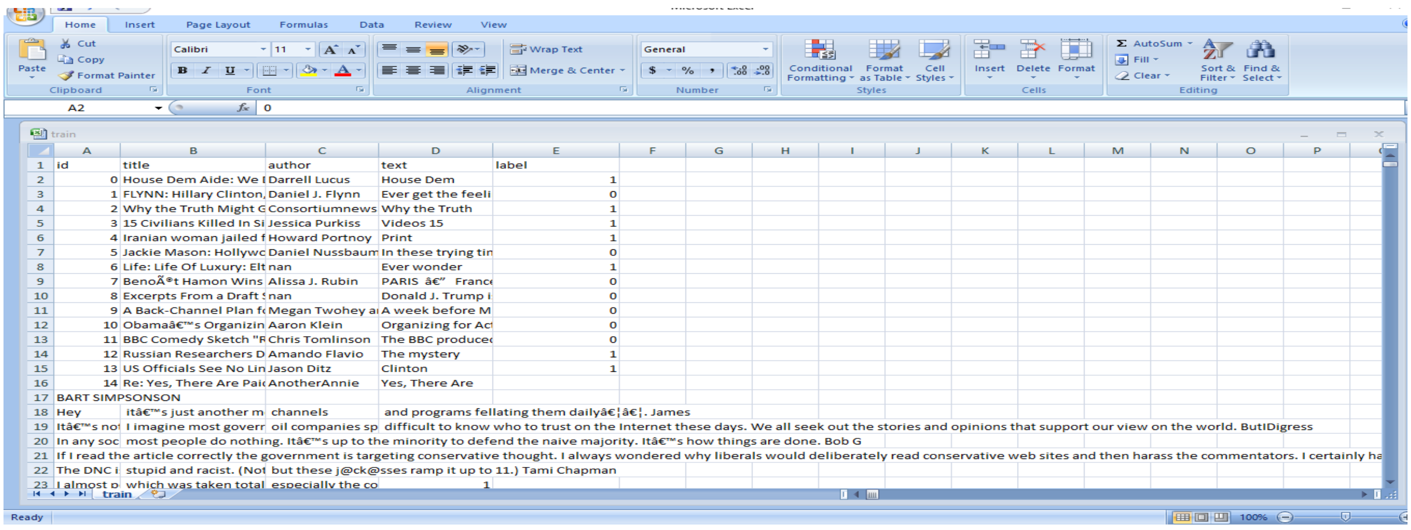
5.2 EXPLANATION OF KEY FUNCTIONS :

The modules in our project are:

- Getting the dataset
- Data pre-processing
- Polarity dataset
- Applying the Algorithms (Naïve Bayes ,Support Vector Machine, Artificial Neural Networks)
- Predicting and Visualizing the results.

Getting the dataset

In this step we download a existing fake news detection dataset from kaggle . This dataset consists of the following attributes- id, author, title, text, label



	A	B	C	D	E
1	id	title	author	text	label
2	0	House Dem Aide: We I	Darrell Lucus	House Dem	1
3	1	FLYNN: Hillary Clinton,	Daniel J. Flynn	Ever get the feeli	0
4	2	Why the Truth Might G	Consortiumnews	Why the Truth	1
5	3	15 Civilians Killed In Si	Jessica Purkiss	Videos 15	1
6	4	Iranian woman jailed f	Howard Portnoy	Print	1
7	5	Jackie Mason: Hollywc	Daniel Nussbaum	In these trying tin	0
8	6	Life: Life Of Luxury: Elt	nan	Ever wonder	1
9	7	BenoÃ®t Hamon Wins	Alissa J. Rubin	PARIS â€" France	0
10	8	Excerpts From a Draft S	nan	Donald J. Trump i	0
11	9	A Back-Channel Plan fc	Megan Twohey ai	A week before M	0
12	10	Obamaâ€™s Organizin	Aaron Klein	Organizing for Act	0
13	11	BBC Comedy Sketch "R	Chris Tomlinson	The BBC producec	0
14	12	Russian Researchers D	Amando Flavio	The mystery	1
15	13	US Officials See No Lin	Jason Ditz	Clinton	1
16	14	Re: Yes, There Are Paic	AnotherAnnie	Yes, There Are	
17	BART SIMPSONSON				
18	Hey	itâ€™s just another m	channels	and programs fellating them dailyâ€¦â€¦. James	
19	Itâ€™s not	I imagine most goverr	oil companies sp	difficult to know who to trust on the Internet these days. We all seek out the stories and opinions that support our view on the world. ButIDigress	
20	In any soc	most people do nothing. Itâ€™s up to the minority to defend the naive majority. Itâ€™s how things are done. Bob G			
21	If I read the article correctly the government is targeting conservative thought. I always wondered why liberals would deliberately read conservative web sites and then harass the commentators. I certainly ha				
22	The DNC i	stupid and racist. (Not	but these j@ck@sses ramp it up to 11.) Tami Chapman		
23	I almost p	which was taken total	especially the co	1	

Fig 10 : Dataset

Data pre-processing

Data Pre-processing in Machine learning is a technique refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format.

Data pre-processing includes:

1. Analyzing the dataset.
2. Identifying the categorical data
3. Encoding the categorical data

In the dataset we divide the data set into training set and testing set in 8:2 ratio respectively.

- **Training set**

  The training data set in Machine Learning is the actual dataset used to train the model for performing various actions. This is the actual data the ongoing development process models learn with various API and algorithm to train the machine to work automatically.

- **Testing set**

  A test dataset is a dataset that is independent of the training dataset, but that follows the same probability distribution as the training dataset. It is used to test the trained model and predict the results.

Applying the Algorithms :

In this step the predicted results and accuracy will be displayed on the screen. To Calculate Gaussian Naïve bayes , We used the Gaussian function to estimate the probability of a given attribute value, given the known mean and standard deviation for the attribute estimated from the training data**.** The following formula is used for calculation of the accuracy.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

To Calculate the Accuracy using support vector machine , We used the support vector machine algorithm to estimate the probability of a given attribute value

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Predicting and Visualizing the results :

Here , the output values are generated.

5.3 METHOD OF IMPLEMENTATION

INSTALLING PYTHON

- Step 1: Download the Python 3 Installer
  Open a browser window and navigate to the Download page for Windows at python.org.
  Underneath the heading at the top that says Python Releases for Windows, click on the link for the Latest Python 3 Release - Python 3.x.x. (As of this writing, the latest is Python 3.6.5.)
  Scroll to the bottom and select either Windows x86-64 executable installer for 64-bit or Windows x86 executable installer for 32-bit.
- Step 2: Run the Installer
  Once you have chosen and downloaded an installer, simply run it by double-clicking on the downloaded file. Then just click Install Now

5.3.1 Python Programming Language

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major Python versions**:** Python 2 and Python 3. Both are quite different.

Python is used for:

- Web Development (server-side)
- Software Development
- Mathematics
- System scripting.

5.3.2 Python Programming Features

- **Easy to code**

  Python is high level programming language.Python is very easy to learn language as compared to other language like c, c#, java script, java etc.It is very easy to code in python language and anybody can learn python basic in few hours or days.It is also developer-friendly language.

- **Free and Open Source**

  Python language is freely available at official website and you can download it from the given download link below click on the Download Python keyword.
  Download Python
  Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

- **Object-Oriented Language**

  One of the key features of python is Object-Oriented programming Python supports object oriented language and concepts of classes, objects encapsulation etc

- **GUI Programming Support**

  Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, Python or Tk in python.
  PyQt5 is the most popular option for creating graphical apps with Python.

- **High-Level Language**

  Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

- **Extensible feature**

  Python is a Extensible language. We can write our some python code into C/C ++ language and also we can compile that code in C/C++ language.

- **Python is Portable language**

  Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we need not change it, we can run this code on any platform.

- **Python is Integrated language**

  Python is also an integrated language because we can easily integrated python with other language like C/C ++ etc

- **Interpreted Language**

  Python is an interpreted language because python code is executed line by line at a time like other language C, C++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

- **Large Standard Library**

  Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing.There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

- **Dynamically Typed Language**

  Python is dynamically typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance because of this feature we don’t need to specify the type of variable.

5.3.3 Python Packages

A package is basically a directory with Python files and a file with the name __init__.py. This means that every directory inside of the Python path, which contains a file named __init__.py, will be treated as a package by Python. It's possible to put several modules into a Package.
Python supports various packages like numpy, pandas ,genism ,sklearn etc
You can install packages using pip i.e., pip install (package_name)

The advantages of packages are :

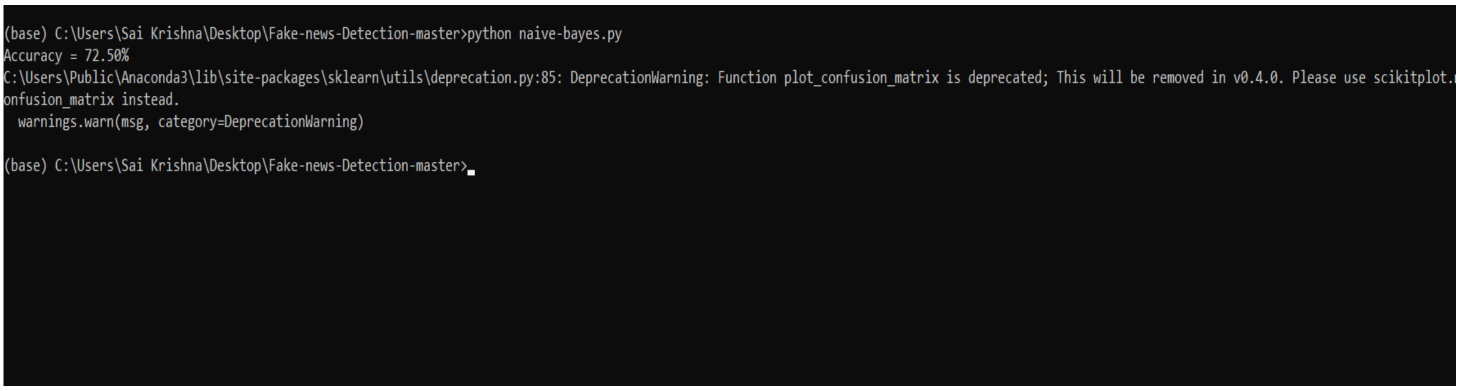
- **Simplicity:** Rather than focusing on the entire problem at hand, a module typically focuses on one relatively small portion of the problem. If you’re working on a single module, you’ll have a smaller problem domain to wrap your head around. This makes development easier and less error-prone.

- **Maintainability:** Modules are typically designed so that they enforce logical boundaries between different problem domains. If modules are written in a way that minimizes interdependency, there is decreased likelihood that modifications to a single module will have an impact on other parts of the program. (You may even be able to make changes to a module without having any knowledge of the application outside that module.) This makes it more viable for a team of many programmers to work collaboratively on a large application.

- **Reusability:** Functionality defined in a single module can be easily reused (through an appropriately defined interface) by other parts of the application. This eliminates the need to recreate duplicate code.

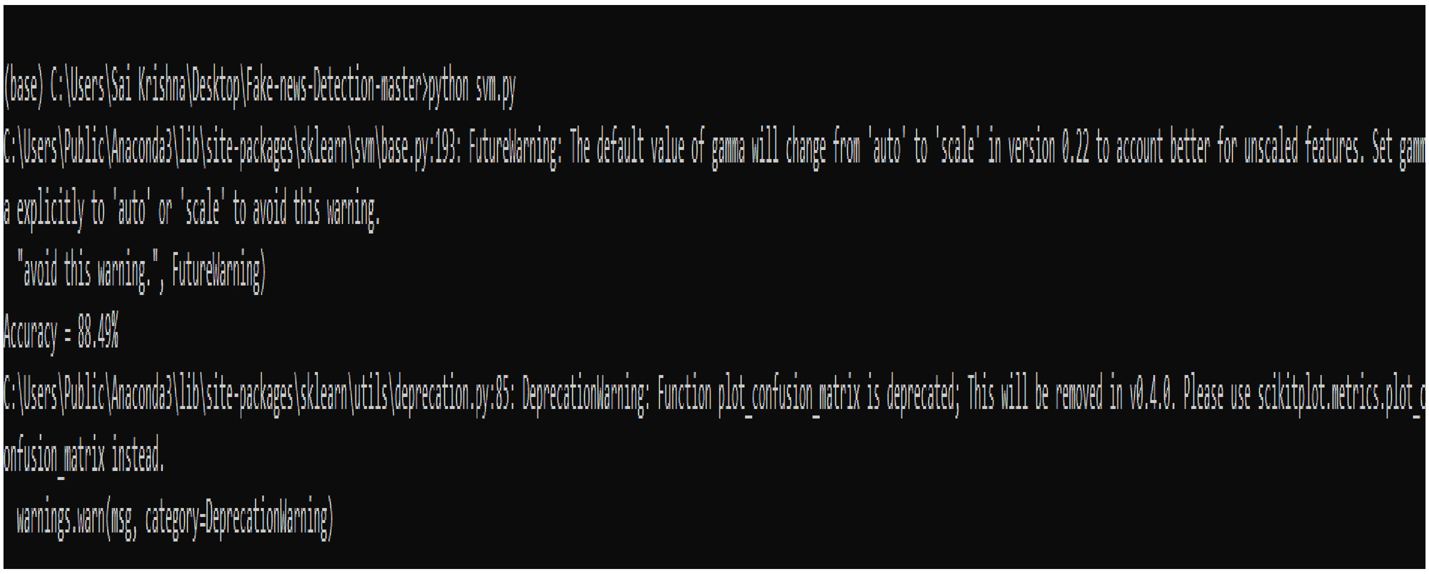
- **Scoping:** Modules typically define a separate namespace, which helps avoid collisions between identifiers in different areas of a program.

5.4 Output Screens



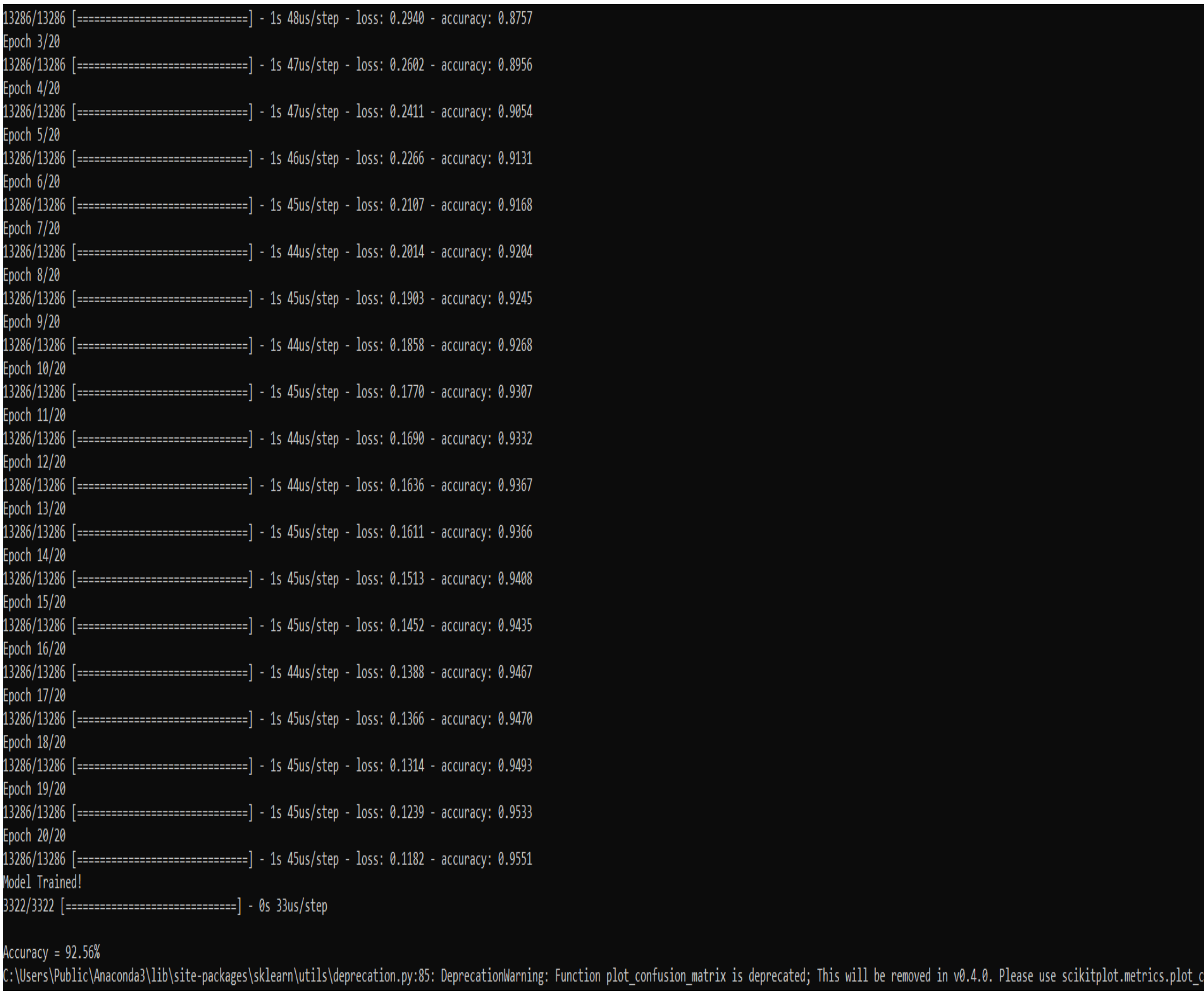
```
(base) C:\Users\Sai Krishna\Desktop\Fake-news-Detection-master>python naive-bayes.py
Accuracy = 72.50%
C:\Users\Public\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:85: DeprecationWarning: Function plot_confusion_matrix is deprecated; This will be removed in v0.4.0. Please use scikitplot.
onfusion_matrix instead.
  warnings.warn(msg, category=DeprecationWarning)

(base) C:\Users\Sai Krishna\Desktop\Fake-news-Detection-master>
```

Screen 1 : Accuracy using Naïve Bayes Algorithm



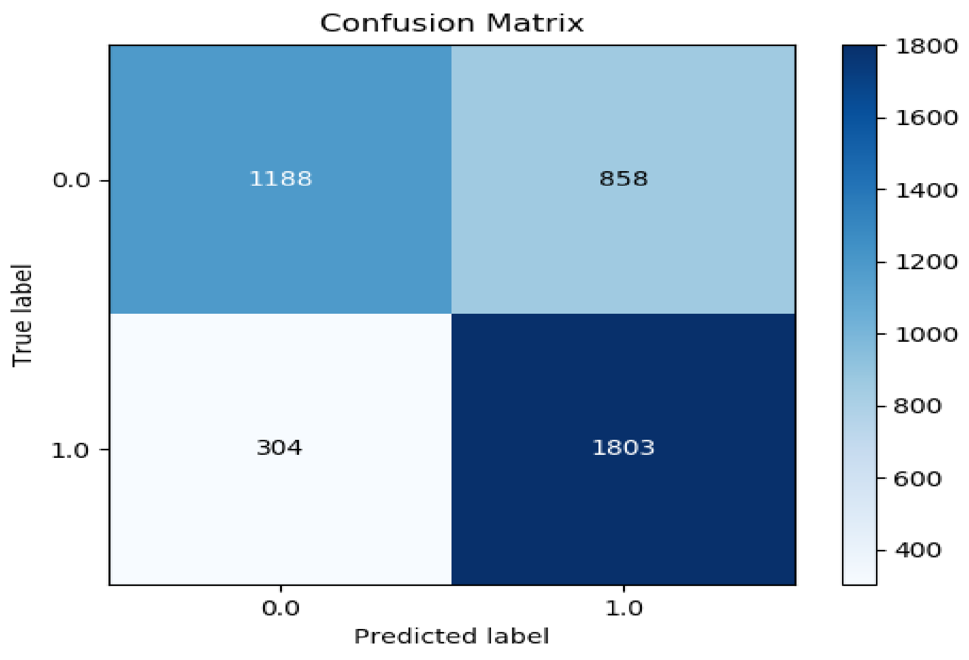
```
(base) C:\Users\Sai Krishna\Desktop\Fake-news-Detection-master>python svm.py
C:\Users\Public\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamm
a explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
Accuracy = 88.49%
C:\Users\Public\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:85: DeprecationWarning: Function plot_confusion_matrix is deprecated; This will be removed in v0.4.0. Please use scikitplot.metrics.plot_c
onfusion_matrix instead.
  warnings.warn(msg, category=DeprecationWarning)
```

Screen 2 : Accuracy using Support Vector Machine

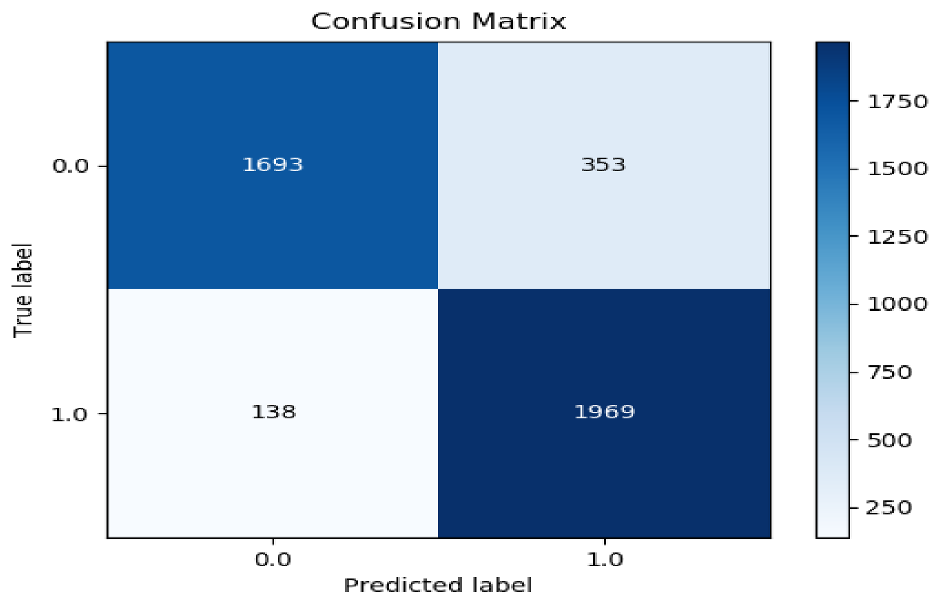


```
13286/13286 [==============================] - 1s 48us/step - loss: 0.2940 - accuracy: 0.8757
Epoch 3/20
13286/13286 [==============================] - 1s 47us/step - loss: 0.2602 - accuracy: 0.8956
Epoch 4/20
13286/13286 [==============================] - 1s 47us/step - loss: 0.2411 - accuracy: 0.9054
Epoch 5/20
13286/13286 [==============================] - 1s 46us/step - loss: 0.2266 - accuracy: 0.9131
Epoch 6/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.2107 - accuracy: 0.9168
Epoch 7/20
13286/13286 [==============================] - 1s 44us/step - loss: 0.2014 - accuracy: 0.9204
Epoch 8/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1903 - accuracy: 0.9245
Epoch 9/20
13286/13286 [==============================] - 1s 44us/step - loss: 0.1858 - accuracy: 0.9268
Epoch 10/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1770 - accuracy: 0.9307
Epoch 11/20
13286/13286 [==============================] - 1s 44us/step - loss: 0.1690 - accuracy: 0.9332
Epoch 12/20
13286/13286 [==============================] - 1s 44us/step - loss: 0.1636 - accuracy: 0.9367
Epoch 13/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1611 - accuracy: 0.9366
Epoch 14/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1513 - accuracy: 0.9408
Epoch 15/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1452 - accuracy: 0.9435
Epoch 16/20
13286/13286 [==============================] - 1s 44us/step - loss: 0.1388 - accuracy: 0.9467
Epoch 17/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1366 - accuracy: 0.9470
Epoch 18/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1314 - accuracy: 0.9493
Epoch 19/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1239 - accuracy: 0.9533
Epoch 20/20
13286/13286 [==============================] - 1s 45us/step - loss: 0.1182 - accuracy: 0.9551
Model Trained!
3322/3322 [==============================] - 0s 33us/step

Accuracy = 92.56%
C:\Users\Public\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:85: DeprecationWarning: Function plot_confusion_matrix is deprecated; This will be removed in v0.4.0. Please use scikitplot.metrics.plot_c
```

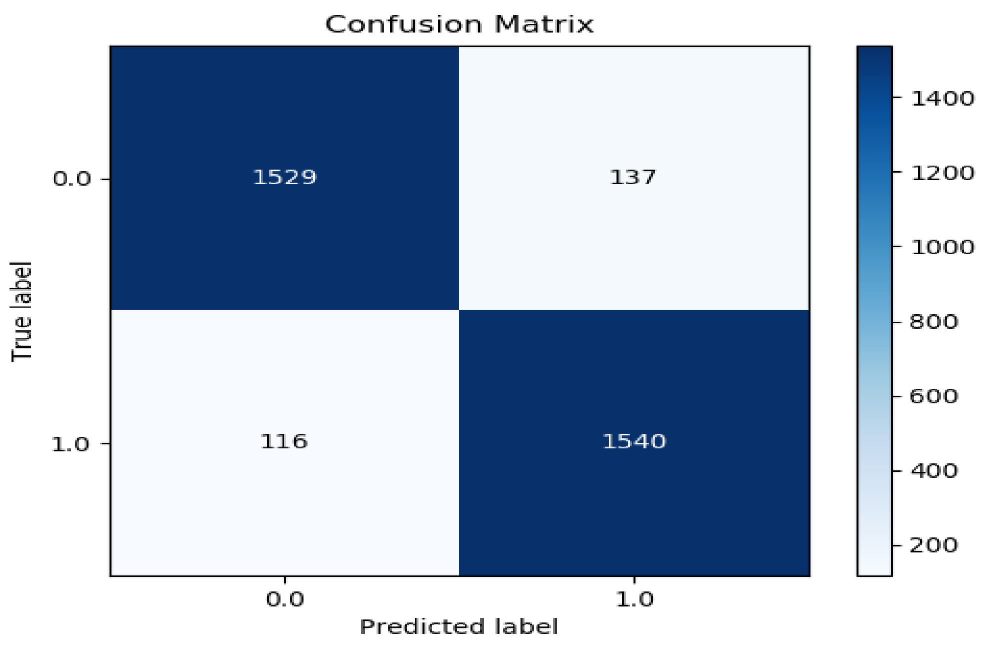
Screen 3 : Accuracy using Artifical Neural Networks



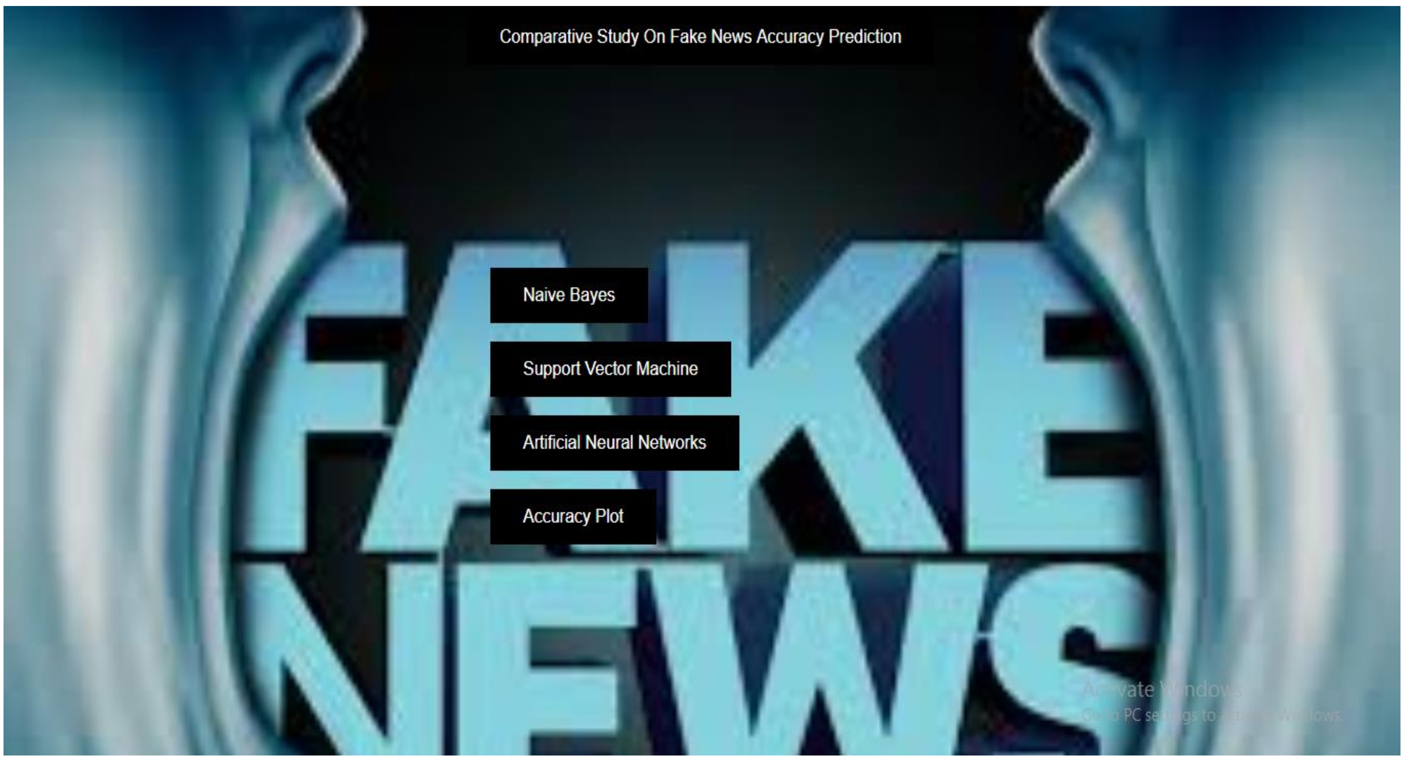
Screen 4 : Confusion Matrix for Naïve Bayes



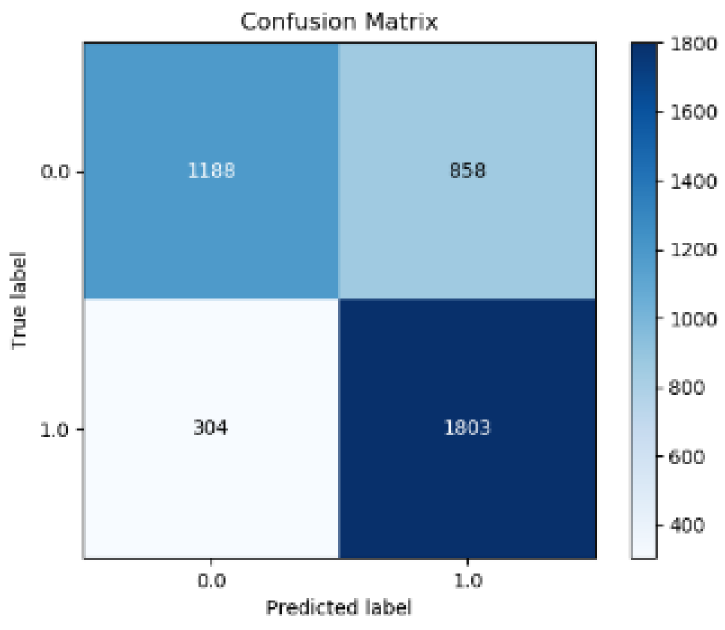
Screen 5 : Confusion Matrix for Support Vector Machine



Screen 6 : Confusion Matrix for Artificial Neural Network

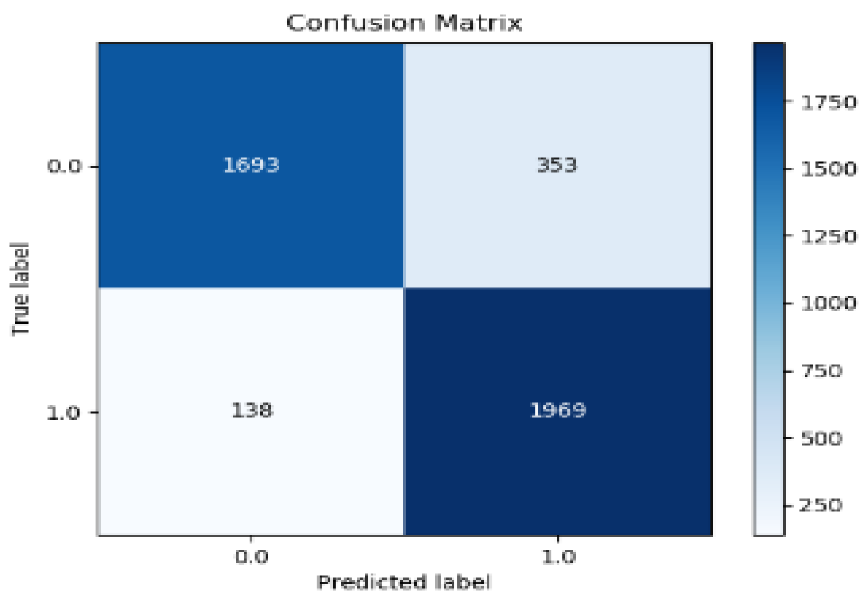

Comparative Study On Fake News Accuracy Prediction
Naive Bayes
Support Vector Machine
Artificial Neural Networks
Accuracy Plot
FAKE NEWS

Screen 7: Front Page



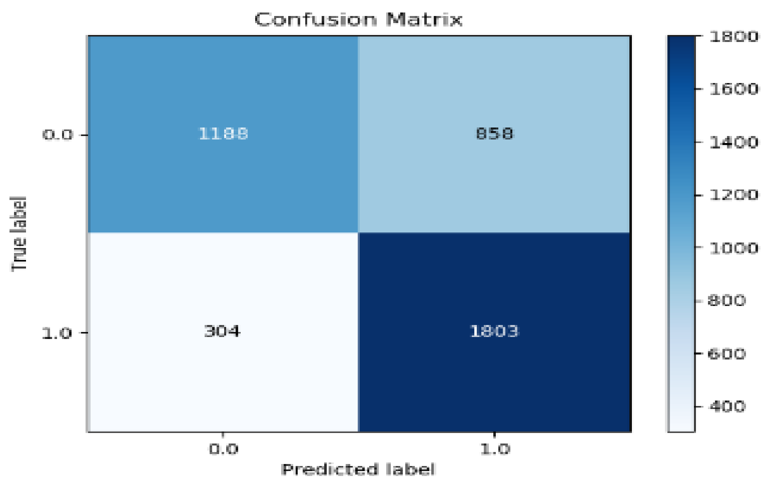
Accuracy is 72.94%

Screen 8 : Naive Bayes Result



Accuracy is 88.97%

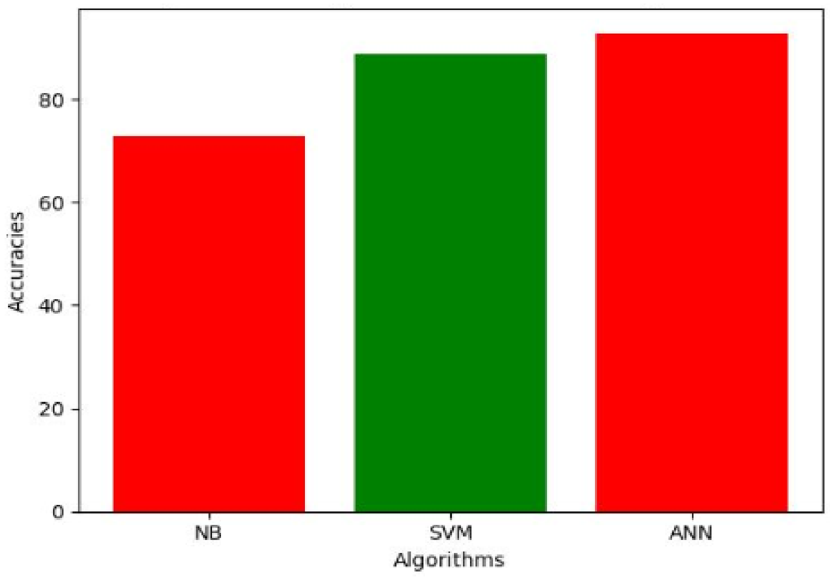
Screen 9 : SVM Result



Accuracy is 92.9715143115%

Screen 10 : ANN Result

Comparative Study On Fake News Accuracy Prediction



Screen 11 : Accuracy Plot

6. TESTING & VALIDATION

6.1 Introduction

- Software testing is really required to point out the defects and errors that were made during the development phases.
- It's essential since it makes sure that the customer finds the organization reliable and their satisfaction in the application is maintained.
    - If the customer does not find the testing organization reliable or is not satisfied with the quality of the deliverable, then they may switch to a competitor organization.
    - Sometimes contracts may also include monetary penalties with respect to the timeline and quality of the product. In such cases, if proper software testing may also prevent monetary losses.
- It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence. (Know more about Software Quality)
    - As explained in the previous point, delivering good quality product on time builds the customers confidence in the team and the organization.
- Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
    - High quality product typically has fewer defects and requires lesser maintenance effort, which in turn means reduced costs.
- Testing is required for an effective performance of software application or product.
- It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development.
    - Proper testing ensures that bugs and issues are detected early in the life cycle of the product or application.
    - If defects related to requirements or design are detected late in the life cyle, it can be very expensive to fix them since this might require redesign, re-implementation and retesting of the application.
- It's required to stay in the business.
    - Users are not inclined to use software that has bugs. They may not adopt a software if they are not happy with the stability of the application.
    - In case of a product organization or startup which has only one product, poor quality of software may result in lack of adoption of the product and this may result in losses which the business may not recover from.

6.2 Design of Test cases and Scenarios

Types of Testing

Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs.

Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design.Integration testing is testing in which a group of components are combined to produce output.

Integration testing is of four types:

(i) Top down
(ii) Bottom up
(iii) Sandwich
(iv) Big-Bang

Black Box Testing

Black box testing also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

White Box Testing

White box testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like white/transparent box inside which one clearly sees.

Regression Testing

Every time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.

Smoke Testing

This test is done to make sure that software under testing is ready or stable for further testing
It is called smoke test as testing initial pass is done to check if it did not catch the fire or smoked in the initial switch on.

Alpha Testing

This is a type of validation testing. It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.

Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment.

System Testing

In this software is tested such that it works fine for different operating system. It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working. In this we have security testing, recovery testing , stress testing and performance testing.

Stress Testing

In this we gives unfavorable conditions to the system and check how they perform in those condition.

Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system. It is used to test speed and effectiveness of program.

6.3 Acceptance Testing

It's a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

- Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.
- Acceptance Testing is the fourth and last level of software testing performed after System Testing and before making the system available for actual use.
- Usually, Black Box Testing method is used in Acceptance Testing.

Test Number	Test Case	Expected Output	Actual Output	Result
1	**Pre-Processing the database**	**Removes the noise**	**Removes the noise**	**Pass**
2	**Clicking on the Accuracy Plot**	**Displays the graph**	**Graph is displayed**	**Pass**
3	**Algorithm Output**	**Displays the accuracy**	**Accuracy is displayed**	**Pass**

6.4 Validation

This is the process of evaluating software at the end of the development process to determine whether the software meets the customer expectations and requirements.

There are two ways to perform software validation

- Internal
- External

During internal software validation, it is assumed that the goals of the stakeholders were correctly understood and that they were expressed in the requirement artifacts precisely and comprehensively. If the software meets the requirement specification, it has been internally validated. External validation happens when it is performed by asking the stakeholders if the software meets their needs.

Performance

- Identification of unstable components/functionalities
- Validation focused on Error-Handling: complementary (not concurrent!) validation regarding the one performed by the Development team (More for the Money, More for the time)
- Compliance with Software and System Requirements
- Black box testing and White box testing techniques
- Experience based techniques

7. CONCLUSION

Combination of machine learning and artificial neural networks are useful for fake news accuracy prediction as it looks like that fake news may be the most challenging area of research in the coming years. Here, we build a model for fake news accuracy prediction using machine learning algorithms. For text processing, we applied Naive Bayes Algorithm,SVM and ANN. Artificial Neural Networks gives the best accuracy out of all.

The project has a vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion data. Generally simple algorithms perform better on less variant data. Since we had less data, SVM and Naive Bayes and Neural Networks performed well.

8. REFERENCES

[1] Conroy, Niall & Rubin, Victoria & Chen, Yimin. (2015). Automatic Deception Detection: Methods for Finding Fake News. . USA

[2] Ball, L. & Elworthy, J. J Market Anal (2014) 2: 187. https://doi.org/10.1057/jma.2014.15

[3] Lu TC. Yu T., Chen SH. (2018) Information Manipulation and Web Credibility. In: Bucciarelli E., Chen SH., Corchado J. (eds) Decision Economics: In the Tradition of Herbert A. Simon's Heritage. DCAI 2017. Advances in Intelligent Systems and Computing, vol 618. Springer, Cham

[4] Rubin, Victoria & Conroy, Niall & Chen, Yimin & Cornwell, Sarah. (2016). Fake News or Truth?Using Satirical Cues to Detect Potentially Misleading News. . 10.18653/v1/W16-0802

[5] www.kaggle.com

[6] https://deeplearning4j.org/keras-supported-features

[7] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[8] https://ieeexplore.ieee.org/document/7489220

[9] https://www.kaggle.com/rksriram312/fake-news-nlp-stuff/notebook

[10] https://ieeexplore.ieee.org/document/8546944

[11] https://ieeexplore.ieee.org/document/8862770

[12] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47