

Text Classification using Machine Learning Algorithms

Project report submitted in partial fulfillment of the
Requirements for the
Award of the Degree of B.Tech in
Computer Science and Engineering

BY

Jasmeet Kaur 2007466

Surbhi Goyal 2007565

Ashutosh Verma 2007440

Under the Guidance of

Dr. Bhaskar Pant (Associate Professor (CSE))

Dr. Devesh Pratap Singh (Head of Department (CSE))



Department of Computer Science and Engineering

Graphic Era University

(Under Section 3 of UGC Act, 1956)

Dehradun-248002

2017



GraphicEra

UNIVERSITY

Deemed University under section 3 of UGC Act, 1956
Accredited by NAAC with Grade 'A'

CERTIFICATE

This is to certify that the project report entitled for **Text Classification using machine Learning Algorithms** is being submitted by

Jasmeet Kaur (2007466)

Surbhi Goyal (2007565)

Ashutosh Verma (2007440)

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Graphic Era University is a record of bonafied work carried out under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Dr. Bhaskar Pant
(Associate Professor (CSE))

Date –

Dr. Devesh Pratap Singh
(Head of the Department)

ACKNOWLEDGEMENT

The completion of any project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge. We are grateful to **Dr. Bhaskar Pant** and **Dr. Devesh Pratap Singh** for their even willingness to give us valuable advice and direction; whenever we approached them with a problem. We are thankful to them for providing immense guidance for this project.

We would also express our gratitude towards our parents for their kind cooperation and encouragement which helped us in the completion of the project. We are also highly thankful to Graphic Era University for their support whenever needed. GEU has provided us with the best faculty mentors which are very helpful for our growth into a good human being.

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

ABSTRACT

It has always been a basic human tendency to classify all sorts of things. We are living in the twenty first century where, nearly everything is digitized. We've been classifying different forms of texts into various categories from centuries. Classifying documents makes searching and sorting of data faster. Considering the huge amount of text data produced every day, classifying it manually is nearly impossible. For this purpose, automated text classification has always been considered as a vital method to manage and process a vast amount of documents in digital forms that are increasing by leaps and bounds. This text data is a part of big data.

By using pre-defined data to train our classifying algorithms i.e. Support Vector Machine, Decision Trees, Random Forest, Extra Trees Classifier, we can simplify the task of text classification. Using relevant results we will prove that SVM is one of the better algorithms in providing higher accuracy over the other three algorithms i.e. Decision Tree, Extra Trees and Random Forest.

Table of contents

Chapter 1 Introduction.....	1
Chapter 2 Literature Survey.....	2
2.1 Approach for the problem	
Chapter 3 Software Requirement Analysis.....	4
3.1 Problem	
3.2 Keywords and Abbreviations	
3.3 Functional Requirements	
3.4 Non Functional Requirements	
3.5 Major Challenges Faced	
3.5.1 Filtering Data	
3.5.2 Filtering Stop Words	
3.6 Modules and their Functionalities	
3.6.1 Data Streaming Module	
3.6.2 Data Filtering Module	
3.6.3 Final Processing	
Chapter 4 Software Design.....	11
Chapter 5 Software and Hardware Requirements.....	16
5.1 Hardware Requirements	
5.2 Software Requirements	
Chapter 6 Coding.....	17
6.1 Data Crawling	
6.2 Data Cleaning	
6.3 Data Analysis	

Chapter 7 Testing.....	29
Chapter 8 Output Screens.....	30
8.1 Crawling Output	
8.2 Performance Metrics and Accuracy	
8.3 Graph	
8.3.1 Code for generating graph	
Chapter 9 Conclusions.....	37
Chapter 10 Future Enhancements.....	38
Chapter 11 References.....	39

List of Figures

3.1	Decision tree for concept of PlayTennis.....	7
3.2	Demonstration of Random Forest classifier.....	8
3.3	Demonstration of Extra Trees Classifier.....	9
3.4	Demonstration of SVM classifier.....	10
4.1	Component Diagram.....	11
4.2	Use Case Diagram.....	12
4.3	Activity Diagram.....	13
4.4	Sequence Diagram.....	14
4.5	Working of Web Crawler.....	15
8.2	Graph representing analysis of classifiers.....	36

List of Screenshots

7.1	Screenshot of files used for testing.....	29
8.1	Screenshot of Data Crawling.....	30

List of Tables

8.1	Performance metrics of SVM.....	31
8.2	Performance metrics of Random Forest.....	32
8.3	Performance metrics of Decision Tree.....	33
8.4	Performance metrics of Extra Trees.....	34

1. INTRODUCTION

Text mining involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the structured data, and finally evaluation and interpretation of the output.

Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics.

Our project on news Categorization classifies the news based on the category they may correspond to, like murder related news, awards related news etc. This helps the users to look for information faster by searching only in the categories they want to, rather than searching the entire information space. The importance of classifying text becomes even more apparent, when the information is too big in terms of volume. Google directory is one such web classification system. In our project, classification methods based on machine learning have been used. In these techniques, classifiers are built (or trained) with a set of training documents. The trained classifiers are then used to assign testing documents to their suitable categories.

The project primarily had three phases - first was to extract (through a web crawler). Second was to preprocess and filter the collected data. The third phase worked on categorizing the data based on different domains of news (e.g. acid attack, education etc) from various websites produced by Bing search engine.

2. LITERATURE SURVEY

The purpose of this project is to classify the news in multiple categories using different machine learning classifiers and comparing their respective accuracies. To accomplish the aforesaid, the following publications were taken into consideration:

- “Extremely randomized trees” by- Pierre Geurts · Damien Ernst · Louis Wehenkel
This paper proposes a new tree-based ensemble method for supervised classification. It consists of randomizing strongly both attribute and cut-point choice while splitting a tree node.
- “Random Forest” by- Leo Breiman
This paper shows how random forests are an effective tool in prediction. Injecting the right kind of randomness makes them accurate classifiers and regressors.
- “A Survey of Various Machine Learning Techniques for Text Classification” by- Gaurav S. Chavan, Sagar Manjare, Parikshit Hegde and Amruta Sankhe
This paper compares the accuracies of Decision Tree, SVM and Naïve Bayes by classifying text and using re-defined data for training.
- “A Comparative Study on Different Types of Approaches to Text Categorization” by- Pratiksha Y. Pawar and S. H. Gawande
The documents can be classified by three ways unsupervised, supervised and semi supervised methods. Text categorization refers to the process of assign a category or some categories among predefined ones to each document, automatically. This paper presents a comparative study on different types of approaches to text categorization.
- “A Review on Multi-Label Learning Algorithms”
The multi-label learning algorithms are scrutinized under common notations with relevant analyses and discussions by – Min-Ling Zhang and Zhi-Hua Zhou.

2.1. Approach for the problem

The complexity of the problems varies from high to low. So some problems are easily solvable like World Knowledge and some are difficult like Negation. For this purpose various algorithms like Decision Tree, Random Forest, Support Vector Machine and Extra trees are available at our disposal.

Steps -

1. Firstly we need to crawl the appropriate data for training.
2. Preprocess the data with the help of libraries such as NLTK.
3. Prepare the data for training.
4. Train the classifier.
5. Make predictions by giving new test data to the trained classifier.
6. Obtain accuracy and performance metrics.

3. SOFTWARE REQUIREMENT ANALYSIS

3.1 Problem

To design a modular program that performs Text Classification on data fed via file and represent the same in various graphical forms.

3.2 Keywords and Abbreviations

1. Topic - This is the subject of discussion. In this project it is generally the name of the labels in which the data will fall.
2. Keywords - Words which help in defining or describing important facts about a topic.
3. Stop Words - These are irrelevant words that do not carry any sentiment example - punctuations , articles and conjunctions.

3.3 Functional Requirements

Major requirements related to functional aspects of the software are -

- The program must categorize the news accurately.
- The program should be able to automatically filter unnecessary and garbage data from the input data.

3.4 Non- Functional Requirements

The non-functional requirements can be broadly categorized as -

- i. Performance - The system's performance will rely on the crawler and the internet connection. These are responsible for fetching news from the internet; hence they play a major role in the system's performance.
- ii. Reliability - The system is expected to fulfill all the functional requirements without any unexpected behavior.
- iii. Availability - The software will be available at all times on the user's device desktop or laptop, as long as the device is in proper working order.

The functionality of the software will depend on any external services such as internet access that are required.

- iv. Maintainability - The system is designed in a modular fashion. These modules help in reducing the complexity of the code and it makes fault tolerance easier.

3.5 Major Challenges Faced

3.5.1 Filtering Data

The crawled data has too much noise. It has to be cleaned or pre-processed for further stages. Python script is used to extract the desired clean data.

3.5.2 Filtering Stop Words

Stop words are sentiment less words. These words are intended to join sentences or act as fillers, punctuations like on, in, under are common examples of stop words. It was a cumbersome challenge to identify and remove stop words using python.

3.6 Modules and their functionalities

The system carries the following modules -

3.6.1 Data Streaming Module

The project made use of a web crawler for extracting news of different domain from various sites. **WEB CRAWLER:** Web crawling is the practice of gathering data by writing an automated program that queries a web server, requests data (usually in the form of the HTML and other files that comprise web pages), and then parses that data to extract needed information.

WebCrawler is a metasearch engine that blends the search results from Google Search or Bing Search or Yahoo! Search. WebCrawler also provides users the option to search for images, audio, video or news. A metasearch engine (or

aggregator) is a search tool that uses another search engine's data to produce their own results from the Internet. Metasearch engines take input from a user and simultaneously send out queries to third party search engines for results. Sufficient data is gathered, formatted by their ranks and presented to the users.

3.6.2 Data Filtering Module (Coded in Python)

The data collected is present in raw form it is disorganized and carries a lot of noise (useless content like hash tags, various symbols and advertisements). This data is then fed into the data filtering module (coded in python) this module filters the data and saves the useful content in a separate file.

The output file of the above module contains well filtered data. This file is fed to the next module.

3.6.3 Final Processing (Classification)

Different classifiers have been used to train the cleaned data. The performance metrics along with the accuracy of different classifier have been obtained. The models used are Decision tree, Random Forest, Support Vector Machine and Extra trees. Comparative study is carried out and SVM is found to be performing the best.

DECISION TREE –

Decision tree is a method for approximating discrete-valued target functions, in which the learned function is represented by decision tree. Learned trees can also be re-represented as sets of if-then rules to improve human readability. These learning methods are among the most popular of inductive inference algorithms and have been successfully applied to a broad range of tasks from learning to diagnose medical cases to learning to assess credit risk of loan applicants.

Decision tree representation –

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

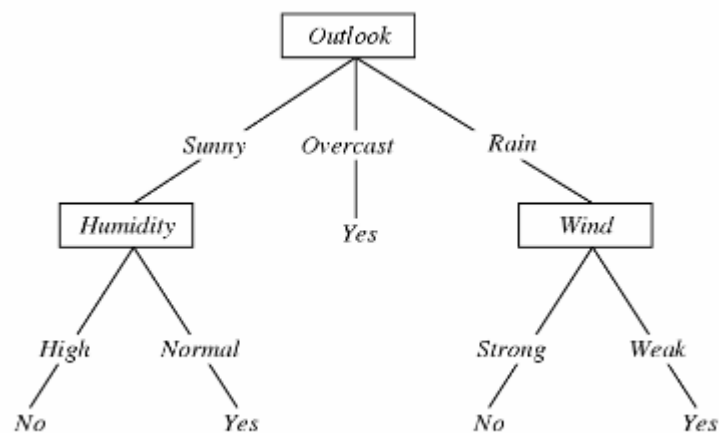


Fig.3.1 A decision tree for the concept PlayTennis. An example is classified by sorting it through tree to the appropriate leaf node, then returning the classification associated with this leaf(in this case Yes or No) .This tree classifies Saturday mornings acc. to whether or not they are suitable for playing tennis.

RANDOM FOREST –

Random Forest is a machine learning algorithm used for classification, regression,and feature selection. It's an ensemble technique, meaning it combines the output of one weaker technique in order to get a stronger result. The weaker technique in this case is a decision tree. Decision trees work by splitting and re-

splitting the data by features. If a decision tree is split along good features, it can give a decent predictive output.

Random Forest works by averaging decision tree output, but it's a bit more complicated than that. It also ranks an individual tree's output, by comparing it to the known output from the training data. This allows it to rank features. Some of the decision trees will perform better, and so the features within the tree will be deemed more important.

Random forest works by averaging decision tree output, but it's a bit more complicated than that classification and regression would be the actual output of the model. A good RF (meaning one that generalizes well) will have higher accuracy by each tree, and higher diversity among its trees.

One downfall of random forest is it can fail with higher dimensional data, because the trees will often be split by less relevant features.

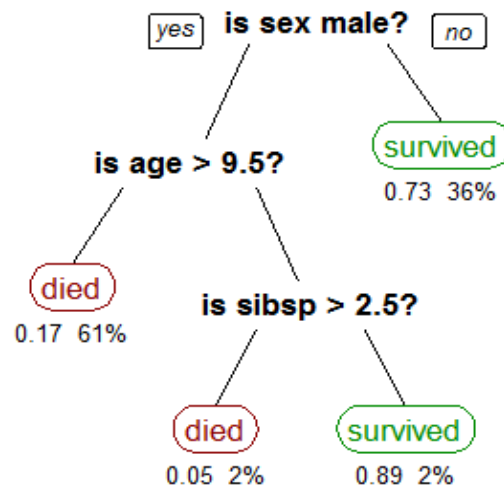


Fig. 3.2 Demonstration of Random Forest Classifier

EXTRA TREES –

An “extra trees” classifier or “Extremely randomized trees” classifier is a variant of a random forest. Unlike a random forest, at each step the entire sample is used and decision boundaries are picked at random, rather than the best one.

The Extra-Trees algorithm builds an ensemble of unpruned decision or regression trees according to the classical top-down procedure. Its two main differences with other tree-based ensemble methods are that it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees.

In real world cases, performance is comparable to an ordinary random forest, sometimes a bit better.

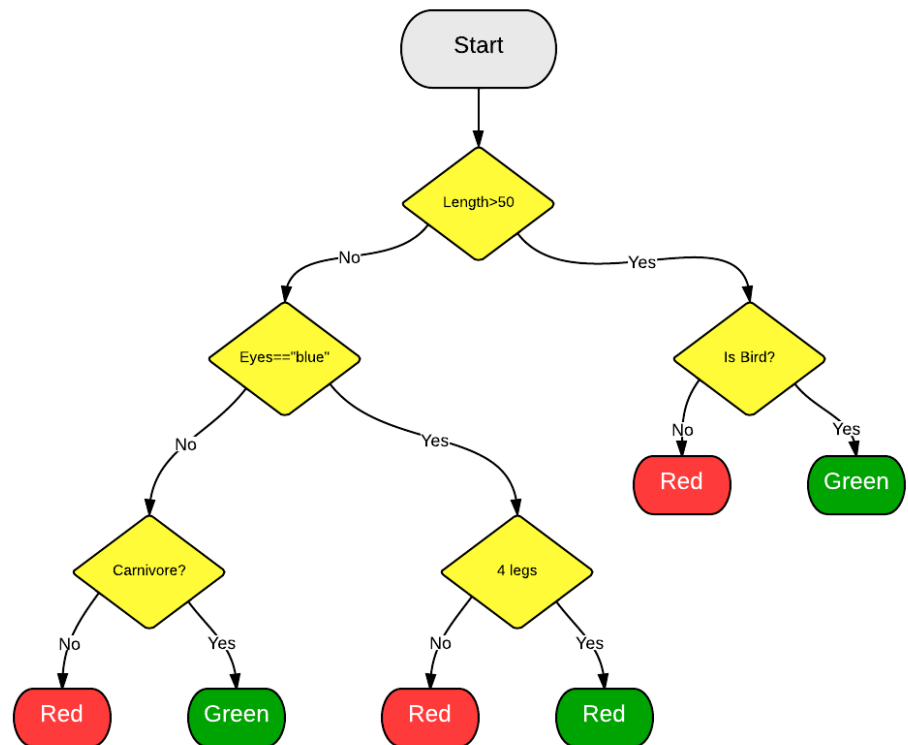


Fig 3.3 Demonstration of Extra Trees classifier

SUPPORT VECTOR MACHINE –

Support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. SVMs are helpful in text and hypertext categorization.

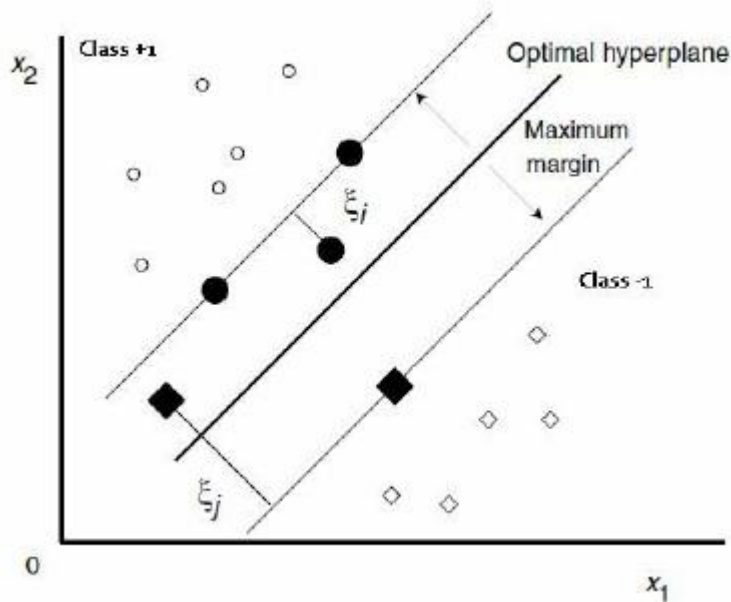


Fig 3.4 Demonstration of SVM classifier

4. SOFTWARE DESIGN

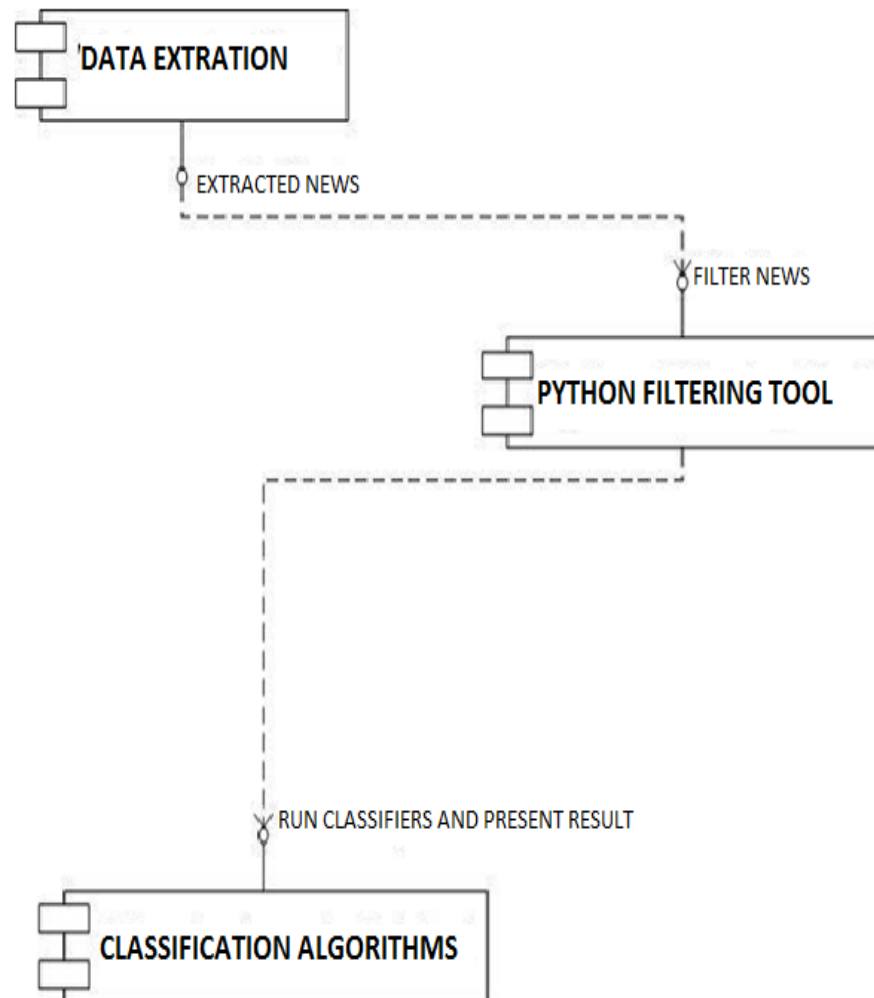


Fig 4.1 Component Diagram

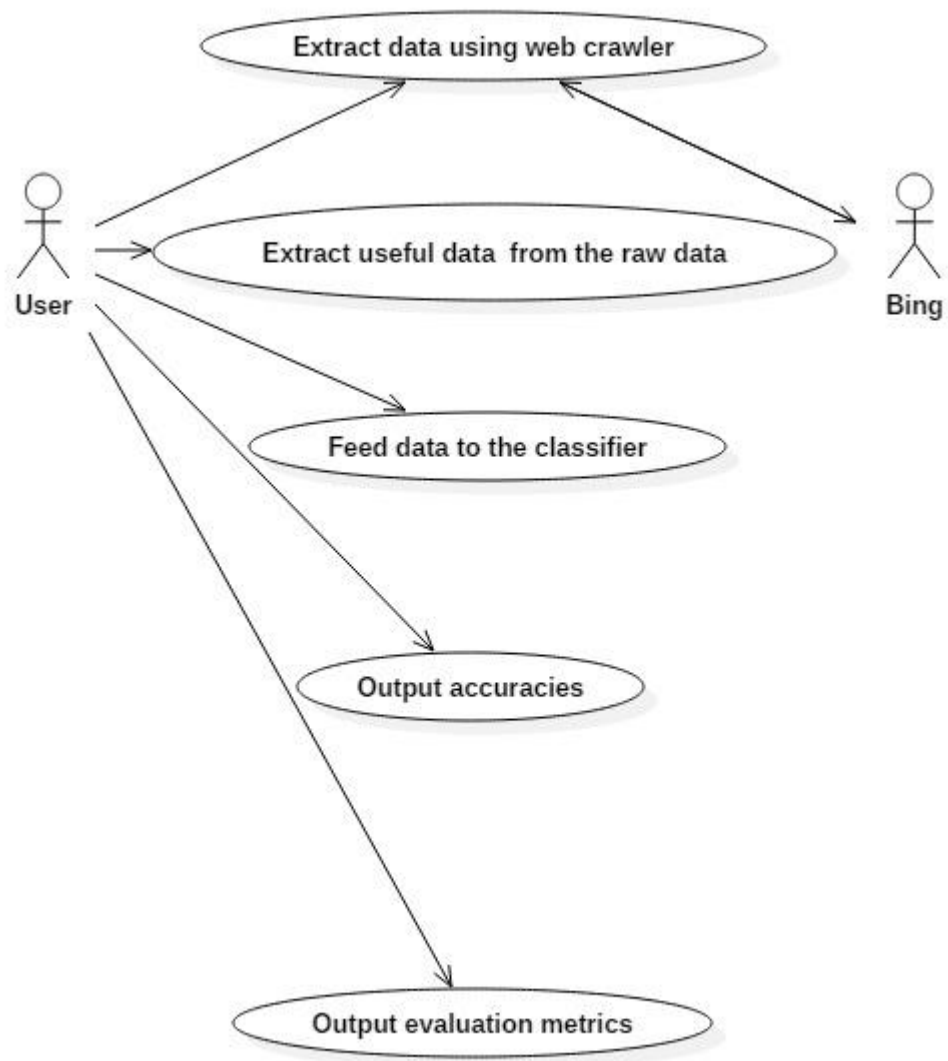


Fig 4.2 Use Case Diagram

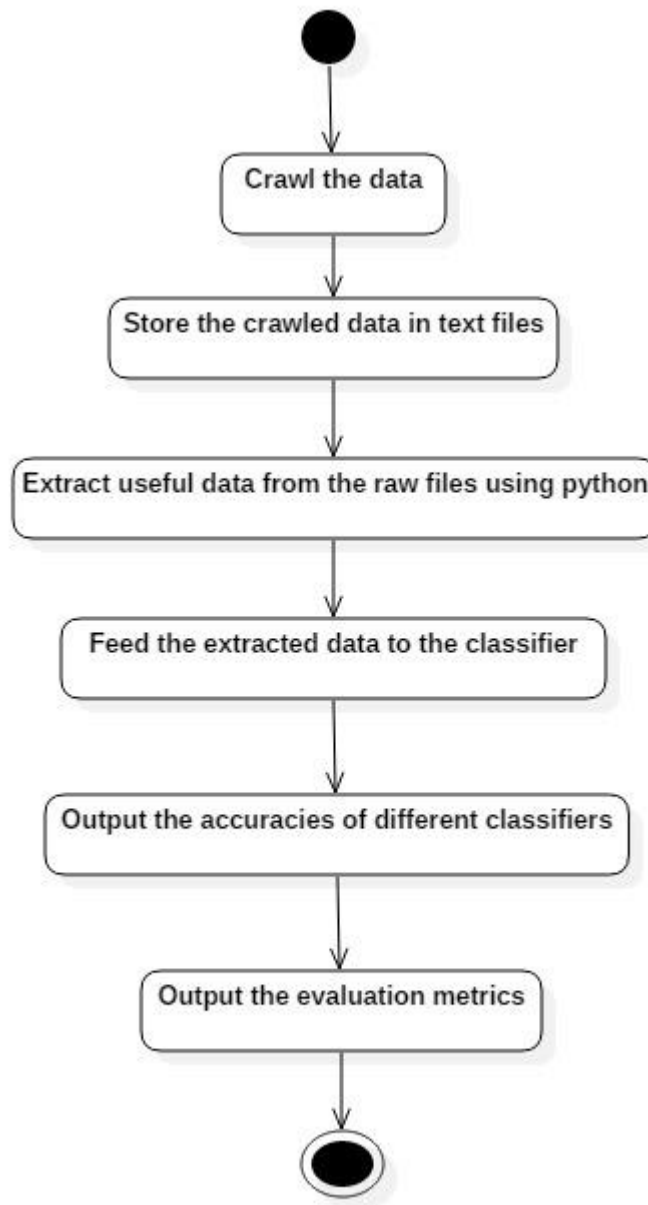


Fig 4.3 Activity Diagram

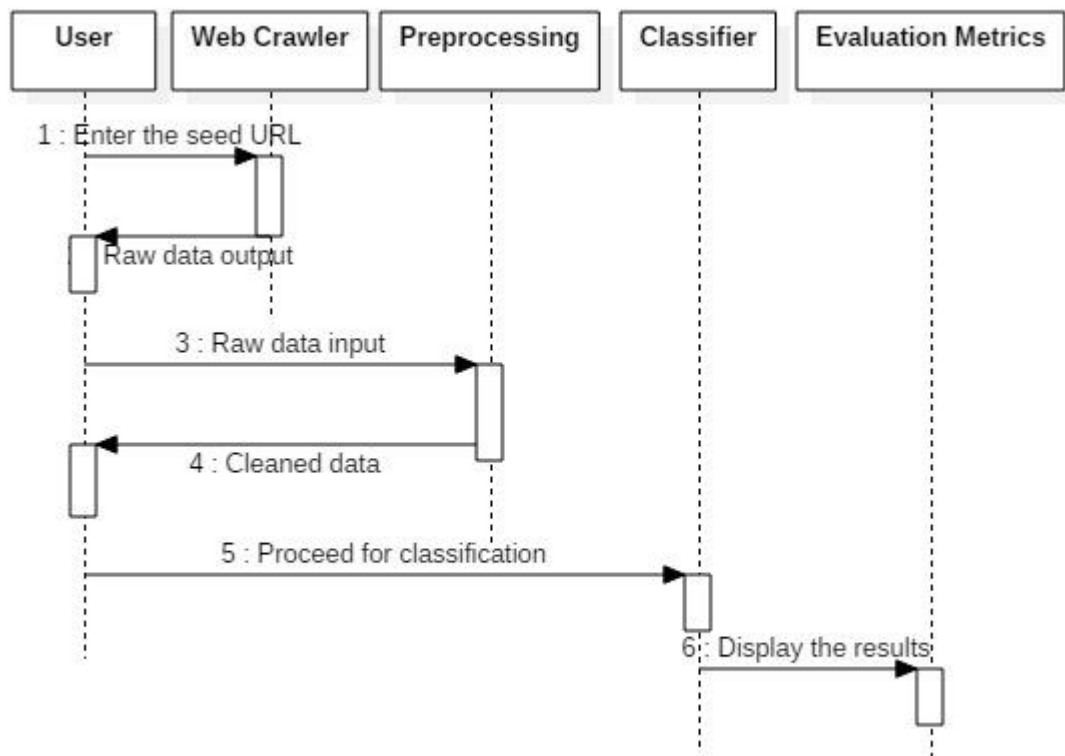


Fig 4.4 Sequence Diagram

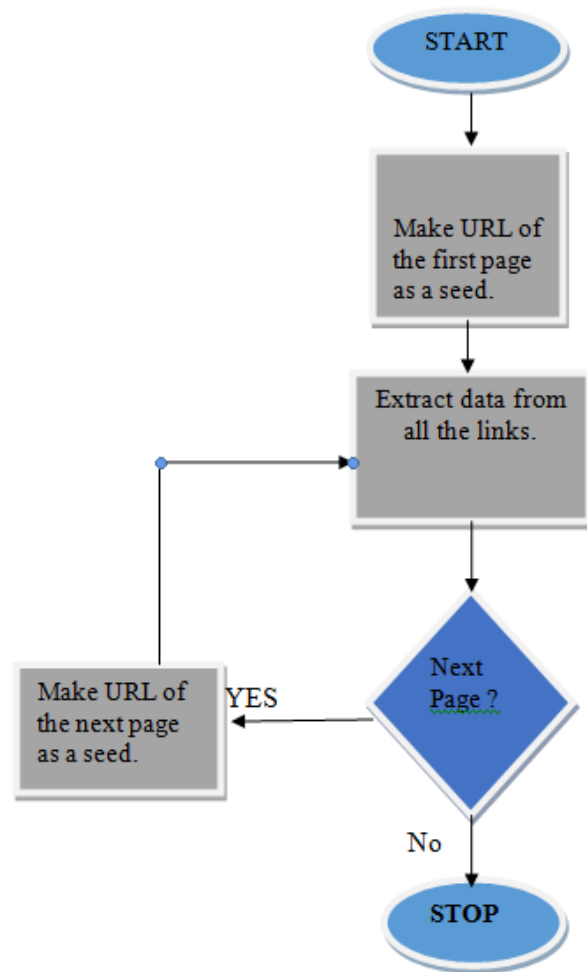


Fig 4.5 Working of Web Crawler

5. SOFTWARE AND HARDWARE REQUIREMENTS

5.1 Hardware Requirements

- Sufficient hardware required to operate Windows or Linux operating system.

5.2 Software Requirements

- Windows or Linux based operating environment.
- Python programming language environment.

6. CODING

6.1 Data Crawling

Data is extracted from web and stored into different text files. Data has been crawled from multiple links using bing search engine initiating from a seed. Python library – BeautifulSoup has been used to pull data out of HTML and XML files.

```
def getData(insideSite,fp):

    html=urlopen(insideSite)

    bsObj1 = BeautifulSoup(html,"lxml")

    dataList=bsObj1.find_all("p")

    global s

    global s1

    s=""

    s1=""

    for data in dataList:

        s=s+data.get_text()

    s1=s.encode('utf-8')

    print s1

    fp.write(s1)

    fp.write("\n\n\t\t\t*****NEWSITE*****\n\n")

    fp.write(insideSite)


def getExternalLinks(bsObj, excludeUrl):
```

```

global externalLinks

externalLinks=[]

#Finds all links that start with "http" or "www" that do not contain the current URL

for link in bsObj.findAll("a",href=re.compile(
    "^((http|www|https)((?!"+excludeUrl+"").)*$"),limit=8):
    if link.attrs['href'] is not None:
        if link.attrs['href'] not in externalLinks:
            externalLinks.append(link.attrs['href'])

return externalLinks

def splitAddress(address):
    addressParts = address.replace("https://","").split("/")
    return addressParts

seeds="https://www.google.co.in/search?q=acid+attack+news&ie=utf-8&oe=utf-8&client=firefox-b&gfe_rd=cr&ei=EXQyWcHzGPTs8AexuKyoBg "

j=0

externalLinks = []

s="

s1="

```

```

fp=open("NewAcidData.txt","a")

while(j<100):

    html = urlopen(seeds)

    bsObj = BeautifulSoup(html,"lxml")

    externalLinks = getExternalLinks(bsObj, splitAddress(seeds)[0])

    for i in range(0,len(externalLinks)):

        if(externalLinks[i]=="http://go.microsoft.com/fwlink/?LinkId=521839&CLCID=4009"):

            break

        else:

            getData(externalLinks[i],fp)

            nextSeed="

            l=bsObj.findAll("a",{ "class":"sb_pagN"})

            if (len(l)!=0):

                for l1 in l:

                    nextSeed=l1.attrs['href']

                    seeds="https://www.bing.com"+nextSeed

            j=j+1

        else:

            break

    fp.close()

```

6.2 Data Cleaning

The extracted news contains many undesired information like hash tags, web addresses, special symbols etc. The following python script extracts useful data from the raw data and gives cleaned data which is further processed.

```
lst=[]

fp=open("NewAcidAttack.txt","r")

lst=fp.readlines()


wrd_lst=[]

new_lst=[]

for i in lst:

    wrd_lst.append(i.split(' '))


for i in wrd_lst:

    if (len(i)>8 and (len(i[0])<20 and len(i[1])<20 and len(i[2]))):

        new_lst.append(i)


s=""

for i in new_lst:

    for j in range(0,len(i)):

        s=""

        for k in i[j]:
```

```

        if(k.isalpha()):

            s=s+k

        else:

            break

    i[j]=s

fp1=open("nltk_dictionary words.txt","r")

eng=fp1.readlines()

for i in range(0,len(eng)):

    s=""

    for j in range(0,len(eng[i])-1):

        s=s+eng[i][j]

    eng[i]=s

hsh={ }

for i in eng:

    if (len(i)>1):

        hsh[i]="

final=[]

```

```

k=0

for i in new_lst:

    c=0

    for j in i:

        j=j.lower()

        try:

            hsh[j]

            c=c+1

        except:

            continue

    if (c>3):

        final.append(i)


fp2=open("AcidData.txt","w")

s=""

for i in final:

    s=s+' '.join(i)

    s=s+'\n\n'


fp2.write(s)

fp1.close()

fp2.close()

```

6.3 Data Analysis

Analysis can be performed by using various algorithms. These algorithms perform a polarity check and determine the nature of news as different labels like Acid, Charity, Cyber Crime etc. The classifiers that have been used are Decision tree, Random Forest and Extra Trees. Partial data has been trained using classifiers and the accuracy of classifiers have been obtained. Every classifier has been made to run on the same data to get their accuracies -

```
x=[]
```

```
y=[]
```

```
Q=[]
```

```
x_train=[]
```

```
y_train=[]
```

```
count=0
```

```
c=0
```

```
c1=0
```

```
c2=0
```

```
Q1=[]
```

```
neg=[]
```

```
pos=[]
```

```
xnew_train=[]
```

```
xnew_test=[]
```

```
ynew_train=[]
```



```

ynew_test=[]

y_model=[]

y1_model=[]

new_neg=[]

new_pos=[]

c=0

f=['AcidData.txt','Charity.txt','CyberCrime.txt','DowryData.txt','EducationNews.txt','Murder
_News.txt','NewAwardsNews.txt','NewNobelPrize.txt','NewScienceNews.txt','NewSmuggli
ngNews.txt','SocialService.txt']

for i in range(0,len(f)):

    x=[]

    y=[]

    f1=open(f[i],'r')

    data=f1.readlines()

    for j in data:

        x.append(j)

        y.append(c)

    xt,xtt,yt,ytt=train_test_split(x,y,test_size=0.40)

    for j in range(0,len(xt)):

        Q.append((xt[j],yt[j]))

    for j in range(0,len(xtt)):

        Q1.append((xtt[j],ytt[j]))

```

```

    c+=1

rm.shuffle(Q)

for i in range(0,len(Q1)):

    Q.append(Q1[i])

for i in range(0,len(Q)):

    x_train.append(Q[i][0])

    y_train.append(Q[i][1])


vec = CountVectorizer()

vec.fit_transform(x_train)

sm = vec.transform(x_train)

sm2=sm[:len(x_train)-len(Q1)]

sm3=sm[len(x_train)-len(Q1):]

y1_train=y_train[:len(Q1)]

y1_train=np.array(y1_train)

```

Code for decision tree -

```

acc=0

d=tree.DecisionTreeClassifier()

d.fit(sm2,y1_train)

l1=d.predict(sm3)

for i in range(0,len(l1)):

```

```

        if l1[i]==Q1[i][1]:

            acc+=1

a=[]

for i in range(0,len(Q1)):

    a.append(Q1[i][1])

a=np.array(a)

met=metrics.classification_report(a,l1)

```

Code for Random Forest -

```

acc=0

d=ensemble.RandomForestClassifier()

d.fit(sm2,y1_train)

l1=d.predict(sm3)

for i in range(0,len(l1)):

    if l1[i]==Q1[i][1]:

        acc+=1

a=[]

for i in range(0,len(Q1)):

    a.append(Q1[i][1])

a=np.array(a)

met=metrics.classification_report(a,l1)

```

Code for Extra Trees Classifier -

```
acc=0

d=ensemble.ExtraTreesClassifier()

d.fit(sm2,y1_train)

l1=d.predict(sm3)

for i in range(0,len(l1)):

    if l1[i]==Q1[i][1]:

        acc+=1

a=[]

for i in range(0,len(Q1)):

    a.append(Q1[i][1])

a=np.array(a)

met=metrics.classification_report(a,l1)
```

Code for SVM –

```
acc=0

d=svm.SVC(kernel='linear', gamma=1)

d.fit(sm2,y1_train)

l1=d.predict(sm3)

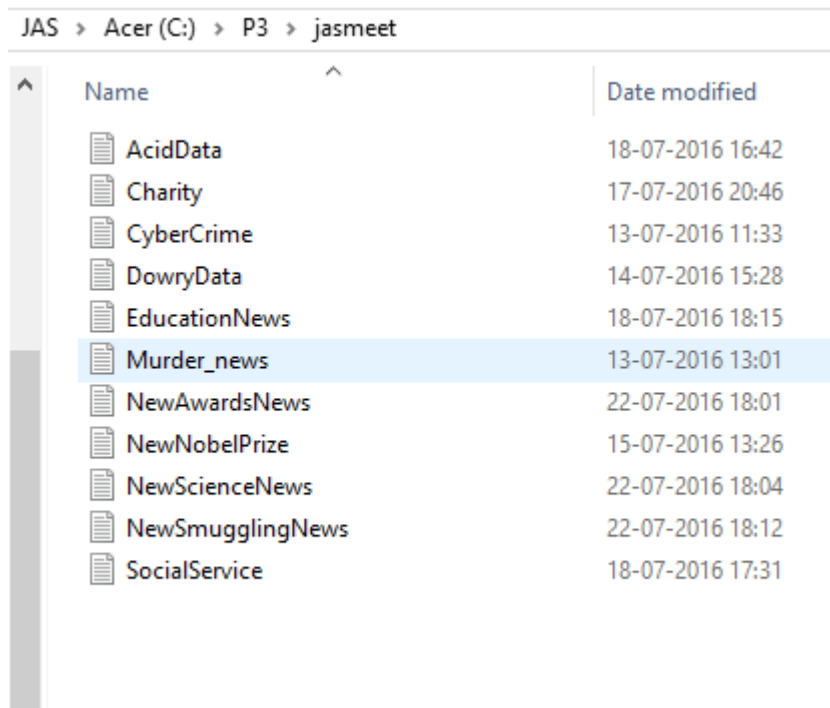
for i in range(0,len(l1)):

    if l1[i]==Q1[i][1]:

        acc+=1
```

```
a=[]  
  
for i in range(0,len(Q1)):  
    a.append(Q1[i][1])  
  
a=np.array(a)  
  
print 'acc'  
  
print acc  
  
acc=(acc*100)/1101230  
  
print acc  
  
met=metrics.classification_report(a,11)  
  
print met
```

7. TESTING














JAS > Acer (C:) > P3 > jasmeet		
^	Name	Date modified
	 AcidData	18-07-2016 16:42
	 Charity	17-07-2016 20:46
	 CyberCrime	13-07-2016 11:33
	 DownryData	14-07-2016 15:28
	 EducationNews	18-07-2016 18:15
	 Murder_news	13-07-2016 13:01
	 NewAwardsNews	22-07-2016 18:01
	 NewNobelPrize	15-07-2016 13:26
	 NewScienceNews	22-07-2016 18:04
	 NewSmugglingNews	22-07-2016 18:12
	 SocialService	18-07-2016 17:31

Fig 7.1 Screenshot of files used for testing

8. OUTPUT SCREENS

8.1 CRAWLING OUTPUT

Two men remain in custody following their arrest in connection with a nightclub attack that left two people partially blinded. A 24-year-old man was arrested at an address in north London on suspicion of grievous bodily harm on Friday evening, following the earlier arrest of a 21-year-old man on the same charges, also in north London. Officers also carried out three arrest warrants at addresses in Hertfordshire and others in Milton Keynes. They are still trying to trace 25-year-old Arthur Collins, the boyfriend of reality TV star Ferne McCann. A 22-year-old woman and a 24-year-old man were both blinded in one eye in the attack at Mangle E8 in Dalston, east London, in the early hours of Monday. Officers believe a dispute between two groups of people developed inside the venue, resulting in a noxious substance being sprayed by a male suspect directly at the pair. Some 20 people suffered burns, with 12 people attending hospital. Other people inside the venue suffered the effects of the substance.:: Victim reveals acid injuries after attack at London bar Mangle E8 Sophie Hall is the latest victim to speak about her ordeal. She told The Sun: "There was panic and shouting and I just started crying because my face felt as if it was on fire." The 21-year-old, who had been on a night out to celebrate a friend's birthday, added: "I looked in the mirror and saw how disfigured my face was." The acid had run down my cheeks and burnt into my skin. I was hysterical." Detective Inspector Lee McCullough said the investigation was "moving at great pace" and the net was "closing in on those we believe to be responsible". He said: "This incident has caused suffering to a large group of people and left a young man and a woman blinded in one eye and many others needing long-term treatment." The noxious substance used has not yet been confirmed, but samples retrieved from the scene have been sent for analysis. "If you were there and saw anyone involved inside or leaving the nightclub, please get in touch." A 33-year-old woman, who was arrested on suspicion of firearms offences following the raids on Thursday, has been bailed to return at a later date.:: Anyone with information about the attack can contact police on 101, Crimestoppers anonymously on 0800 555 111, or tweet @MetCC.

© 2017 Sky UK

<http://globalnews.ca/tag/acid-attack/>

Change Location Newscasts & Videos A woman says she was attacked with acid by a man she accused of raping her in Bulandshahr city, Uttar Pradesh province on Sunday, after she refused to withdraw her case against him. Continue reading → Police in Pakistan say they have arrested a woman for allegedly throwing acid on a man who refused to marry her. Continue reading →

Fig 8.1 Screenshot of Data Crawling

8.2 PERFORMANCE METRICS AND ACCURACY

By SVM - Accuracy 46.72923004

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.40	0.30	0.34	231747
1	0.67	0.19	0.29	25247
2	0.70	0.51	0.59	270592
3	0.43	0.82	0.57	339416
4	0.15	0.03	0.05	32516
5	0.32	0.11	0.16	9070
6	0.16	0.02	0.03	846
7	0.21	0.04	0.07	983
8	0.08	0.04	0.06	1577
9	0.33	0.16	0.22	1088
10	0.32	0.11	0.17	188148
Avg/total	0.47	0.46	0.43	1101230

Table 8.1 Performance metrics of SVM

By Random Forest - Accuracy 44.85729593

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.35	0.16	0.22	231747
1	0.83	0.19	0.31	25247
2	0.60	0.44	0.51	270592
3	0.43	0.83	0.57	339416
4	0.26	0.03	0.06	32516
5	0.29	0.12	0.17	9070
6	0.41	0.11	0.17	846
7	0.12	0.01	0.02	983
8	0.55	0.01	0.03	1577
9	0.69	0.06	0.11	1088
10	0.43	0.29	0.35	188148
Avg/total	0.45	0.43	0.41	1101230

Table 8.2 Performance metrics of Random Forest

By Decision Tree - Accuracy 44. 2899481353

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.39	0.35	0.37	231747
1	0.51	0.21	0.30	25247
2	0.59	0.45	0.51	270592
3	0.44	0.76	0.56	339416
4	0.13	0.03	0.04	32516
5	0.14	0.05	0.07	9070
6	0.85	0.41	0.55	846
7	0.04	0.01	0.01	983
8	0.11	0.01	0.02	1577
9	0.38	0.04	0.07	1088
10	0.26	0.11	0.16	188148
Avg/total	0.42	0.44	0.41	1101230

Table 8.3 Performance metrics of Decision Tree

By Extra Trees - Accuracy 45.72923004

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.34	0.19	0.24	231747
1	0.80	0.20	0.32	25247
2	0.59	0.46	0.51	270592
3	0.45	0.80	0.57	339416
4	0.22	0.04	0.07	32516
5	0.28	0.12	0.17	9070
6	0.78	0.20	0.32	846
7	0.21	0.04	0.07	983
8	0.25	0.01	0.03	1577
9	0.45	0.18	0.26	1088
10	0.41	0.30	0.35	188148
Avg/total	0.45	0.46	0.42	1101230

Table 8.4 Performance metrics of Extra Trees

Like this, for one model, results of various classifiers were compared but among all the results, the output using Random Forest was the best as shown above.

8.3 GRAPH

8.3.1 CODE FOR GENERATING GRAPH

```
vals=[]

vals=met.split()

xvals=[vals[len(vals)-4], vals[len(vals)-3], vals[len(vals)-2]]

N = 3

ind = np.arange(N) # the x locations for the groups

width = 0.07      # the width of the bars

fig = plt.figure()

ax = fig.add_subplot(111)

rects1 = ax.bar(ind, xvals, width, color='r')

rects2 = ax.bar(ind+width, yvals, width, color='g')

rects3 = ax.bar(ind+width*2, zvals, width, color='b')

rects4 = ax.bar(ind+width*3, kvals, width, color='y')
```

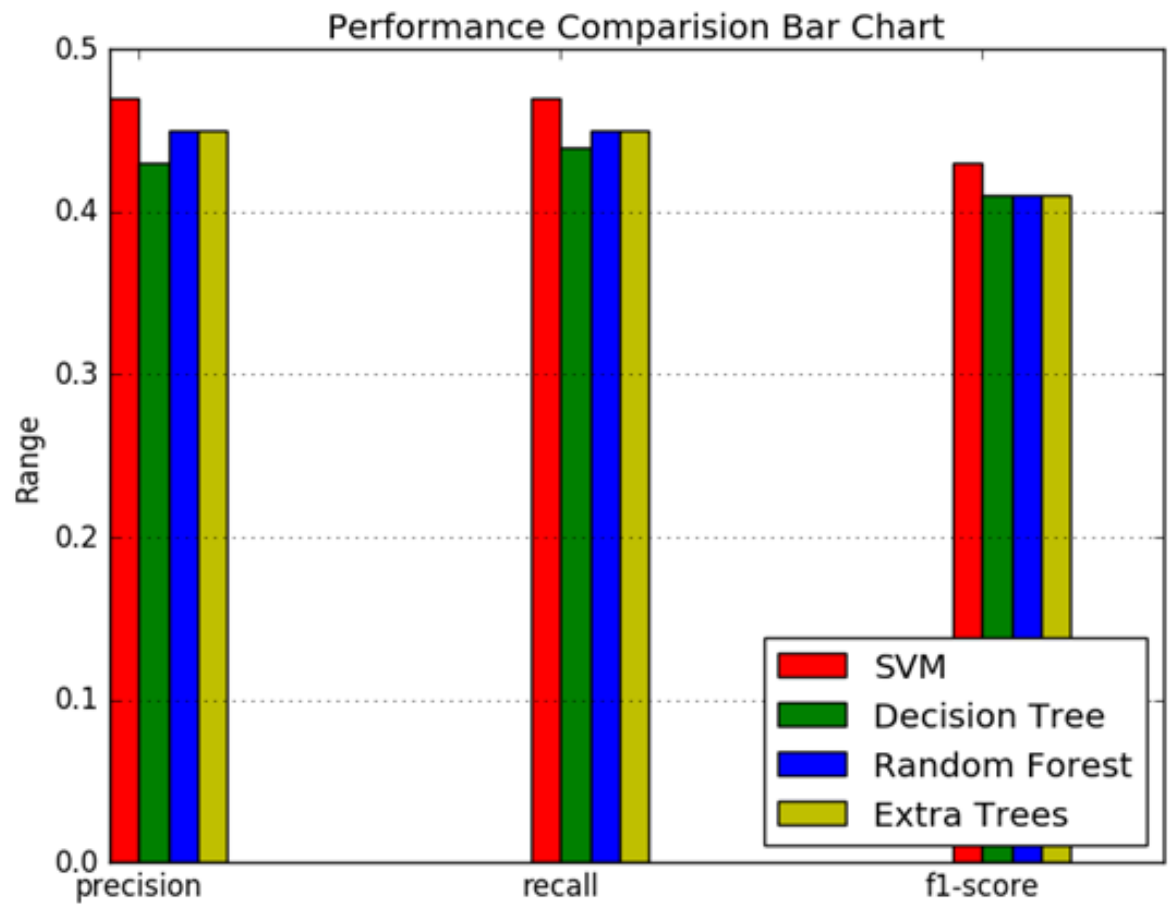


Fig 8.2 Graph representing comparative analysis of classifiers

9. CONCLUSION

News was classified into different categories such as murder, acid attack, dowry etc. After using different classifiers accuracy of the classifiers was treated as a parameter to view their performance. After comparing the results Support Vector Machine is found out to be working the best.

10. FUTURE ENHANCEMENTS

Apart from single classifier method ensemble learning can be applied. The project can be extended to build a system that categorizes news articles and helps visualize them based on the entities location, organization and person - for instance determining the count of a particular crime by location. Another aspect that can be included is the comparison of news article coverage based on their source, for example: the difference in BBC News and CNN when covering news in a particular domain.

11. REFERENCES

- [1] Ryan Mitchell (2015), Web Scraping with Python Collecting Data from the Modern Web.
- [2] Tom Mitchell's Decision Tree Teaching Material
<http://www2.cs.uh.edu/~ceick/ai/dectree.pdf>
- [3] Himay Jesal Desai, Bharat Thatavarti, Aditi Satish Mhapsekar,
CS 410 PROJECT REPORT News Article Categorization Team Members
- [4] Pierre Geurts ,Damien Ernst and Louis Wehenkel (2006), "Extremely randomized trees"
- [5] Leo Breiman (2001) , "RANDOM FORESTS"
- [6] Gaurav S. Chavan, Sagar Manjare, Parikshit Hegde and Amruta Sankhe, "A Survey of various Machine Learning Techniques for Text Classification"
- [7] Pratiksha Y. Pawar and S. H. Gawande, "A Comparative Study on Different Types of Approaches to Text Categorization"
- [8] Min-Ling Zhang and Zhi-Hua Zhou (2013), "A Review on Multi-Label Learning Algorithms"