A Project Report on

Sentiment Analysis Framework for Social Media

Submitted in partial fulfillment of the requirements for the award
of the degree of

Bachelor of Engineering

in

Information Technology

by

Riddhi Prajapati(15104015)
Pratiksha Patil(16204011)
Aafreen Shaikh(16204009)

Under the Guidance of

Prof.Sunil A. Sushir

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W),400615
UNIVERSITY OF MUMBAI

Academic Year 2018-2019

Approval Sheet

This Project Report entitled ***"Sentiment Analysis Framework for Social Media"*** Submitted by ***"Riddhi Prajapati"(15104015),"Pratiksha Patil"(16204011), "Aafreen Shaikh"(16204009)***is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

(Prof.Sunil A. Sushir)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Sentiment Analysis Framework for Social Media"*** submitted by ***"Riddhi Prajapati" (15104015),"Pratiksha Patil" (16204011),"Aafreen shaikh" (16204009)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2018-2019.

(Prof.Sunil A. Sushir)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Signature)

_____

(Riddhi Prajapati and 15104015)
(Pratiksha Patil and 16204011)
(Aafreen Shaikh and 16204009)

Date:

Acknowledgement

We have great pleasure in presenting the report on **Sentiment Analysis Framework for Social Media.** We take this opportunity to express our sincere thanks towards our guide **Prof.Sunil A. Sushir** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name1:Riddhi Prajapati
Student ID1:15104015

Student Name2:Pratiksha Patil
Student ID2:16204011

Student Name3:Aafreen Shaikh
Student ID3:16204009

Abstract

In today's world, Social Media is one such platform where people openly express their emotions, reviews, feedback or even personal experiences publicly. This feedback is regarding products, posts, services which they find proper or not. This is quiet useful as it lets everyone know about the quality and other measures which is good enough. Currently microblogging websites like Twitter, Facebook are filled with data and opinions. Twitter is one amongst most frequently used microblogging website where users tweet about a particular topic. Hence an individual might require hours to figure out whether the users are giving a positive or negative response for that particular topic. Sentiment Analysis is an automated process which identifies and classifies opinions or sentiments from a piece of text. Sentiments are of two categories Positive, Negative or Neutral. Sentiment Analysis in Twitter is difficult as compared to the general ones due to presence of large number of misspellings, hashtags, etc. Various Machine Learning algorithms are adopted for classification of data set into Positive or Negative classes based on their sentiments such as Naive Bayes, Decision Tree, etc. This paper contains implementation of Naive Bayes, Random Forest and Logistic Regression. The output of which are displayed in terms of Word cloud and confusion matrix. The main emphasis of this research is on the classification of emotions of tweets' data gathered from Kaggle. For actual implementation of this system Python along with Anaconda Navigator is been used.

Keywords-Twitter, Sentiment Analysis, Machine learning, Positive Tweets, Negative Tweets, Word cloud, Python.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Currently, social networking sites are at an all-time high, so from there, a large amount of data is generated. Social Networking websites and micro blogging websites in todays world has become the biggest web destinations for people to communicate with each other, to express their thoughts about products or movies, share their daily experience and communicate their opinion about real time and upcoming events, such as sports or political elections etc. To achieve a large, diverse data set of current public opinions or sentiments, Twitter could be used as a valuable resource which allows users to send and read small messages called Tweets. Twitter is a reservoir for a large amount of data. So this data is extremely useful for predicting results of political activities, new initiatives led by government, or research and deciding on what content to share with the audience. Input to our model is the raw data extracted from tweets. For the same, we automate the process of tweet extraction and categorizing it into two categories i.e. positive or negative. The content in twitter generated by the user is with the advent of social media over the last decade. The efforts to determine peoples attitudes with respect to a specific topic or event have garnered a wide research interest in natural language processing and introduced Sentiment Analysis.

Sentiment Analysis is systematic method of gaining knowledge from opinions or emotions. It's application has shown significance in business and marketing field. As social media gained its importance in recent days, Sentiment analysis turned out to be one of the best era for research. The sentiment analysis of customer's social media data is very important in the present day business scenarios. Customers share their reviews and their comfort towards the products on social media. This information can be used for various application such as market research, product feedback and analysing customer service effectiveness. The analysis of the sentiment could lead to many interesting results.

Machine Learning is a prediction technique in which the present data classification is predicted based on past observations. There are many machine learning algorithms that are in use today. These algorithms can also be successfully applied for classification of text efficiently. This is achieved by using Naive Bayes, Logistic Regression and Random Forest. In the Sentiment analysis literature, the best classification accuracy is achieved with Logistic Regression. We have used Jupyter Notebook in Anaconda version 4.2-3 and it's coding is implemented in Python by importing libraries for processes. Also there are diagrammatic representations in the form of Bar Graph, Word cloud, Confusion Matrix. Finally, all of those classification algorithms are compared and the best which fits the data set is obtained.

Sentiment Analysis Dataset Twitter has a number of applications:

Business: Companies use Twitter Sentiment Analysis to develop their business strategies, to assess customers feelings towards products or brand, how people respond to their campaigns or product launches and also why consumers are not buying certain products.

Politics: In politics Sentiment Analysis Dataset Twitter is used to keep track of political views, to detect consistency and inconsistency between statements and actions at the government level. Sentiment Analysis Dataset Twitter is also used for analyzing election results.

Public Actions: Twitter Sentiment Analysis also is used for monitoring and analyzing social phenomena, for predicting potentially dangerous situations and determining the general mood of the blogosphere. Furthermore in this paper we present the results of our experiments and ideas on how to further improve the obtained results.

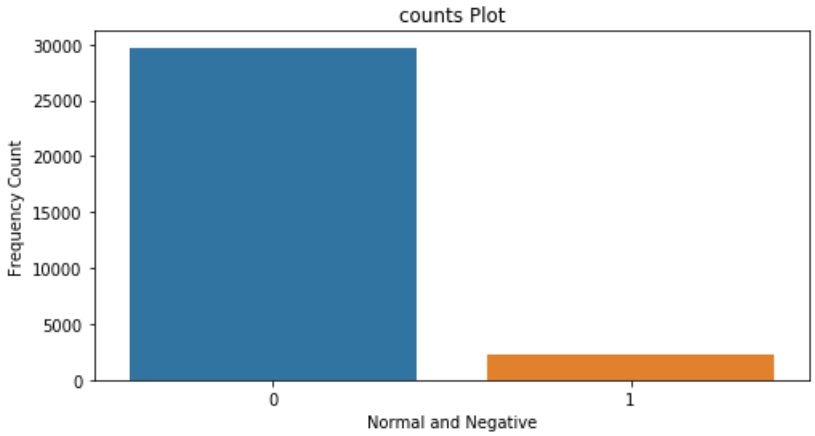


Figure 1.1: Graphical representation of Tweets

1.1 Problem Statement

The problem in Sentiment analysis is classifying the polarity or accuracy of given text at the document, sentence, or feature/aspect level. Whether the expressed opinion is a document, a sentence or an entity feature/aspect is positive, negative or neutral.

1.2 Scope

Sentiment Analysis is becoming better as Social media is rising up day by day. This project will be quite helpful to the political parties, common people, businesses, etc. It can be used to give valuable insights to businesses regarding how common people feel about a particular brand or in case of social media it can briefly summarize on how people think about a certain topic. Similarly companies can get reviews regarding their newly released software or hardware. By analyzing the tweets we can get result on how positive or negative people are about it.

1.3 Objectives

The objectives of this project are as follows :

- To Preprocess the data set and classification of the same into positive and negative.
- Sentiment analysis to determine the attitude of the mass. Whether it is positive or negative towards its subject of interest.
- Graphical representation of the sentiments in form of Bar graph and Word cloud.
- To implement algorithms for classification of data set and display the same using Confusion matrix
- To determine the algorithm with highest accuracy on data set.

1.4 Overview

This proposal entitled "Sentiment Analysis Framework on Social Media" is a system which is used to analyze data set of tweets. We will be performing sentiment analysis in twitter data set which is classified as positive or negative. This system can be used by political leaders to review regarding their contribution in their time period, companies to review about their products or services, etc. The misspellings, hashtags, slang words, stop words will be removed in preprocessing part. The graphical representation will show percentage of positive and negative content. Based on entire data set, separate Word clouds for positive and negative will be displayed. Then the given data set is split into training and testing data. We will be applying classification algorithms to data set and the one which will be fitting best will show highest accuracy. Also the Confusion Matrix plotted will display the number of classified and miss-classified tweets. And above all, this entire system will be GUI-based output for better results.

1.5 Motivation

- An aspect of Social media data such as Twitter messages is that it includes rich structured information about the individuals involved in the communication.
- It can lead to more accurate tools for extracting semantic information.
- It provides means for empirically studying properties of social interactions.
- Freely available, annotated corpus, Pre-written Classifier Codes in Python using NLTK (Natural Language Toolkit) that can be used in NLP in order to promote research that will lead to a better understand of how sentiment is conveyed in tweets and texts.

Chapter 2

Literature Review

Ms. Farha Nausheen and Ms. Sayyada Hajera Begum have presented a paper at 2016 "Sentiment Analysis to Predict Election Results Using Python". They have proposed lexicon based sentiment analyzer which classifies the tweets based on its sentiment value. They have considered the tweets from US presidential elections 2016. They have performed classification based on polarity and subjectivity measures. The proposed system retrieves tweets and performs political sentiment analysis. Tools like Twython, NLTK, TextBlob are employed. They present comparison between the top candidates for presidential elections.

Anukur Goel, Jyoti Gautam and Sitesh Kumar have presented a paper at 2016 "Real Time Sentiment Analysis of Tweets Using Naive Bayes" They have proposed the use of SentiWordNet2 for Opinion Mining providing sentiment score to each synset of WordNet. For implementation of the same they have used python with NLTK along with python-twitter APIs. The scores are of three types: positivity score, negativity score and objectivity score. So instead of using Naive Bayes in traditional way, they have used scores so that more accuracy can be attained. Also they have implemented Naive Bayes using sentiment140 training data using twitter database.

Ms.K.Saranya and Dr.S.Jayanthy have presented a paper at 2017 "Onto-based sentiment classification using Machine Learning Techniques" They proposed the use of semantics and ontology for text classification combining the same with machine learning techniques for getting better results. They employed SVM (Support Vector Machine), NB (Naive Bayes), kNN (k nearest neighbors) classifiers. They identified affective class hierarchy in WordNet with extracting it. Later assigning emotions to semantic roles and creating an emotion hierarchy.

Megha Rathi, Aditya Malik, Daksh Varshney, Rachita Sharma, Sarthak Mendiratta have presented a paper at 2018 "Sentiment Analysis of Tweets Using Machine Learning Approach" They have proposed Support Vector Machine, Adaboosted Decision Tree and Decision Tree based hybrid sentiment classification model for improving the overall accuracy of the classifier in the classification of tweets. For analytical evaluation of the proposed classifier they have employed accuracy and f-measure. They have provided comparative results which prove that hybrid model improved the overall classification accuracy and f-measure of sentiment prediction as compared to traditional existing techniques for classification. Their proposed approach classified the tweets as Positive and Negative.

Chapter 3

Proposed System

The proposed system retrieves tweets and performs Sentiment analysis. It consists of four main modules as mentioned below :
1. Retrieval module
2. Pre-processing module
3. Analysis module
4. Result visualization module

3.1 Retrival Module

Twitter allows its users to retrieve tweets. Here we are using offline data set. User tweets are collected from Kaggle i.e. in the form of a data set. The data set is processed using python code in Jupyter Notebook of Anaconda Navigator. These tweets are then filtered and classified based on the sentiments i.e. Positive or Negative.

3.2 Pre-Processing Module

Huge amount of data is already present related to different problems, but using this given data as it is, may not produce desired output as data contains many irrelevant things which makes it tough to handle. So it is always suggested to remove irrelevant portion of data or make it relevant which can further increase efficiency of the system. The Pre-processing module performs the filtration process using python libraries for retrieving the meaningful parts of the tweet by removing the unnecessary content. The collected tweets may contain hashtags , URLs, @tags, emoticons, trailing whitespaces and newline characters. Each of these have their own significance during the Sentiment analysis and some are irrelevant which do not have any significant effect so it is suggested to omit these data while performing sentiment analysis. User name in tweets always starts with @ symbol, this is to tell who tweeted this particular tweet , while doing sentiment analysis there is no significant effect of user name so by applying filters, user name is excluded from training as well testing data.

3.3 Analysis Module

The original sentiments are used to check two factors:

- **Polarity:**It is the sentiment associated with the entity i.e. it may be positive or negative?
- **Subjectivity:** How much sentiment (of any polarity) does the entity gather?

Polarity indicates percentage of positive sentiments among overall sentiment references and percentage of negative sentiments among overall sentiment references. Subjectivity indicates proportion of sentiment to frequency of occurrence. Average polarity and subjectivity measures are calculated as in the proposed algorithms used.

3.4 Result Visualization Module

The final step of this process is to take in the filtered tweets and generate Bar graph after pre-processing. Also it generates Word Cloud of positive and negative tweets separately of entire data set. Confusion Matrix will be displayed of each classifiers depicting classified and miss-classified tweets.

Chapter 4

Design

4.1 Use-case Diagram

4.1.1 Use case 1

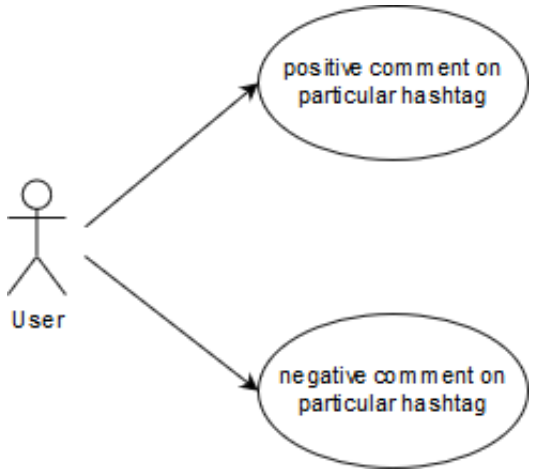


Figure 4.1: Use-Case for Users

4.1.2 Use case 2

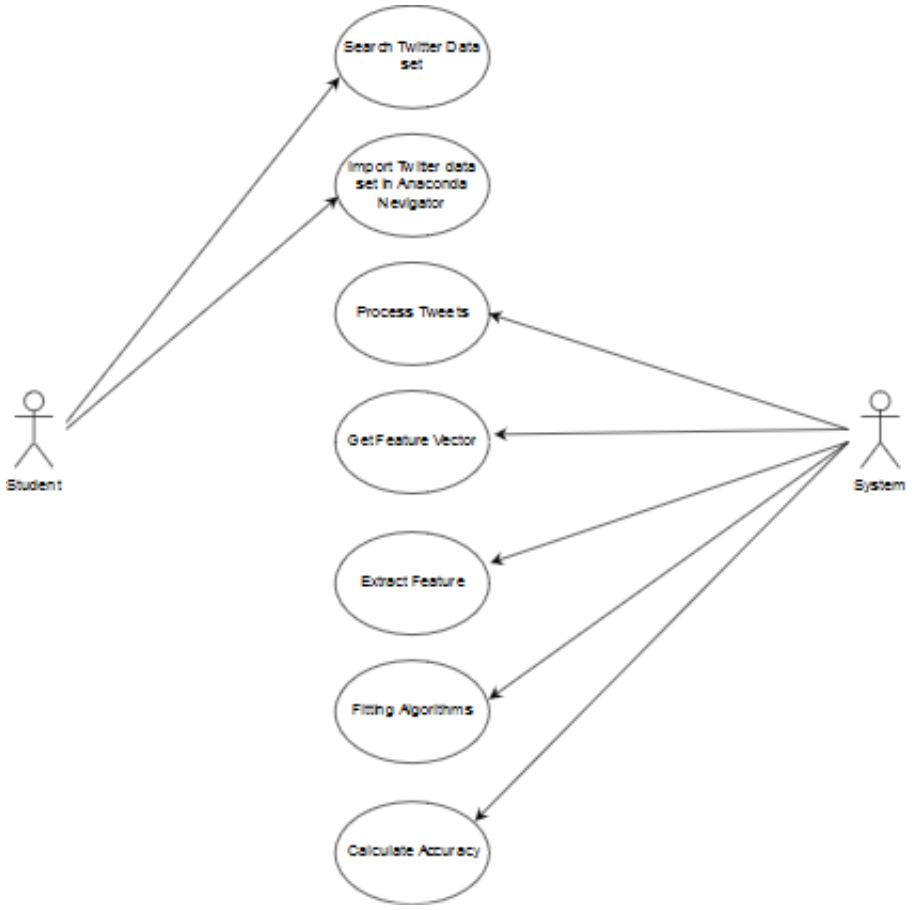


Figure 4.2: Use-Case for Student and system

4.2 Activity Diagram

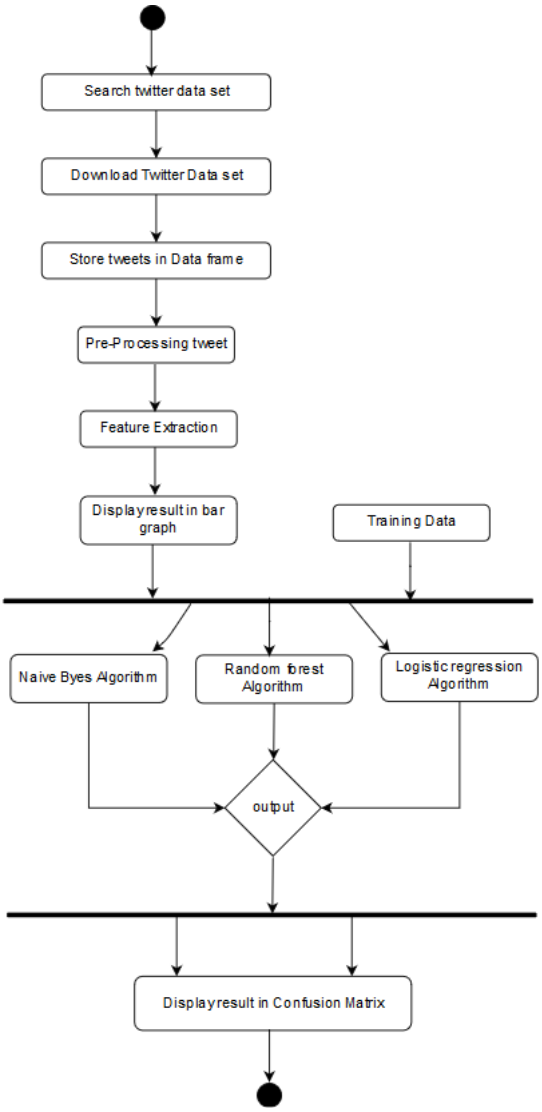


Figure 4.3: Activity Diagram

4.3 Data Flow Diagram

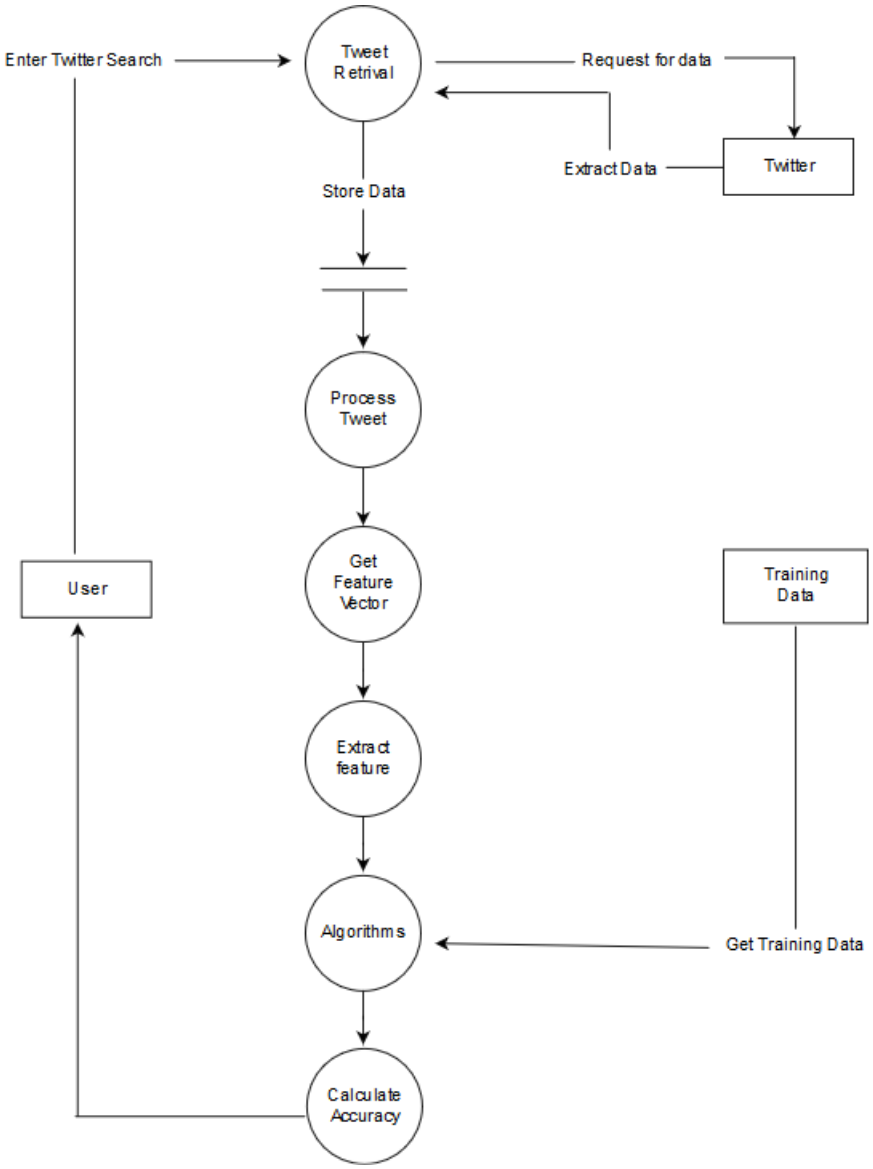


Figure 4.4: Data Flow Diagram

4.4 System Flow Diagram

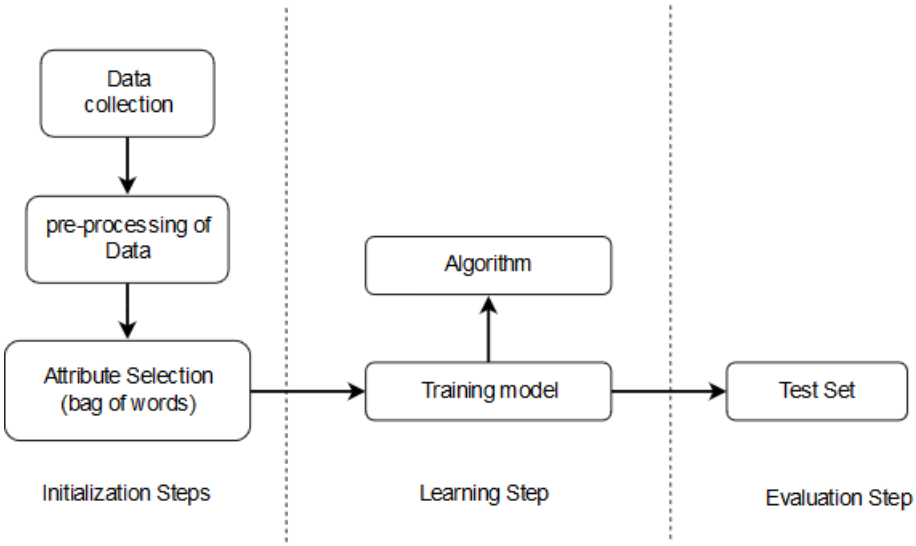


Figure 4.5: System Flow Diagram

Chapter 5

Implementation

5.1 libraries

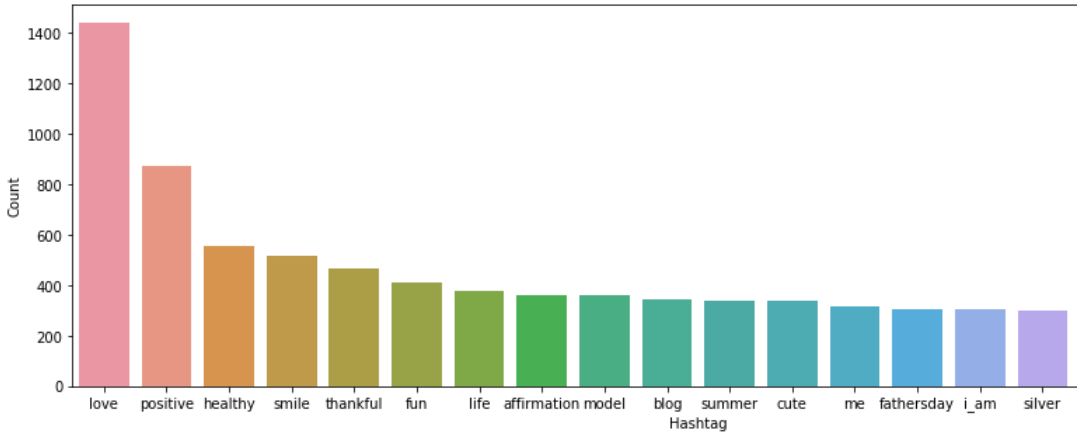
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#for sentiment
import nltk
from wordcloud import WordCloud,STOPWORDS
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer, ENGLISH_STOP_WORDS

#For Model
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import f1_score, roc_auc_score
```

5.2 Hashtags

```
#Select all words from normal tweet
normal_words = ' '.join([word for word in train['cleaned_tweet'][train['label'] == 0]])
#Collect all hashtags
pos_htag = [htag for htag in normal_words.split() if htag.startswith('#')]
#Remove hashtag symbol (#)
pos_htag = [pos_htag[i][1:] for i in range(len(pos_htag))]
#Count frequency of each word
pos_htag_freqcount = nltk.FreqDist(pos_htag)
pos_htag_df = pd.DataFrame({'Hashtag' : list(pos_htag_freqcount.keys()),
                            'Count' : list(pos_htag_freqcount.values())})
```

5.3 Positive Hashtag

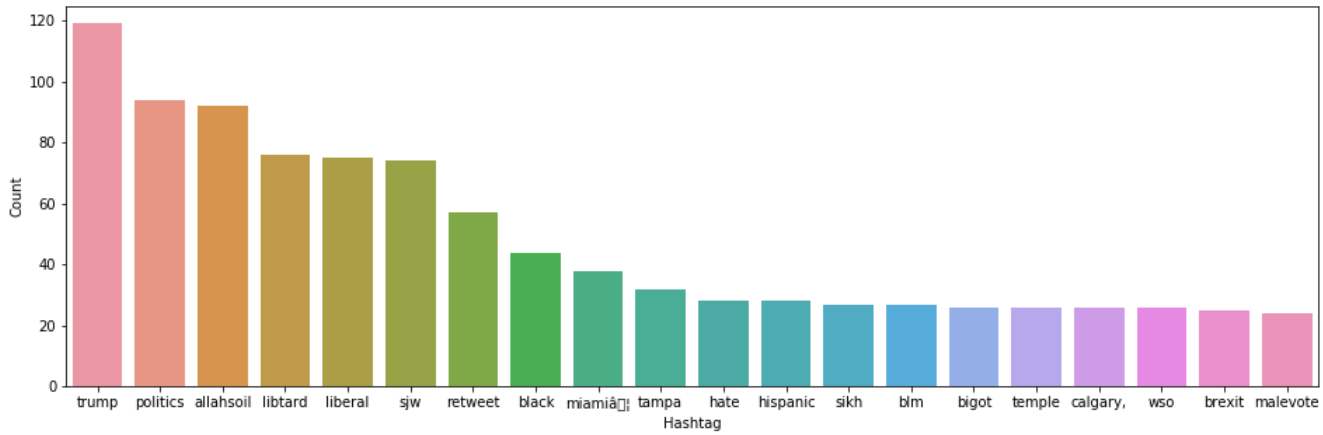
```
#Select top 20 most frequent hashtags and plot them
most_frequent = pos_htag_df.nlargest(columns="Count", n = 20)
plt.figure(figsize=(16,5))
ax = sns.barplot(data=most_frequent, x= "Hashtag", y = "Count")
ax.set(ylabel = 'Count')
plt.show()
```



5.4 Negative Hashtag

```
#Repeat same steps for negative tweets
negative_words = ' '.join([word for word in train['cleaned_tweet'][train['label'] == 1]])
neg_htag = [htag for htag in negative_words.split() if htag.startswith('#')]
neg_htag = [neg_htag[i][1:] for i in range(len(neg_htag))]
neg_htag_freqcount = nltk.FreqDist(neg_htag)
neg_htag_df = pd.DataFrame({'Hashtag' : list(neg_htag_freqcount.keys()),
                            'Count' : list(neg_htag_freqcount.values())})
```

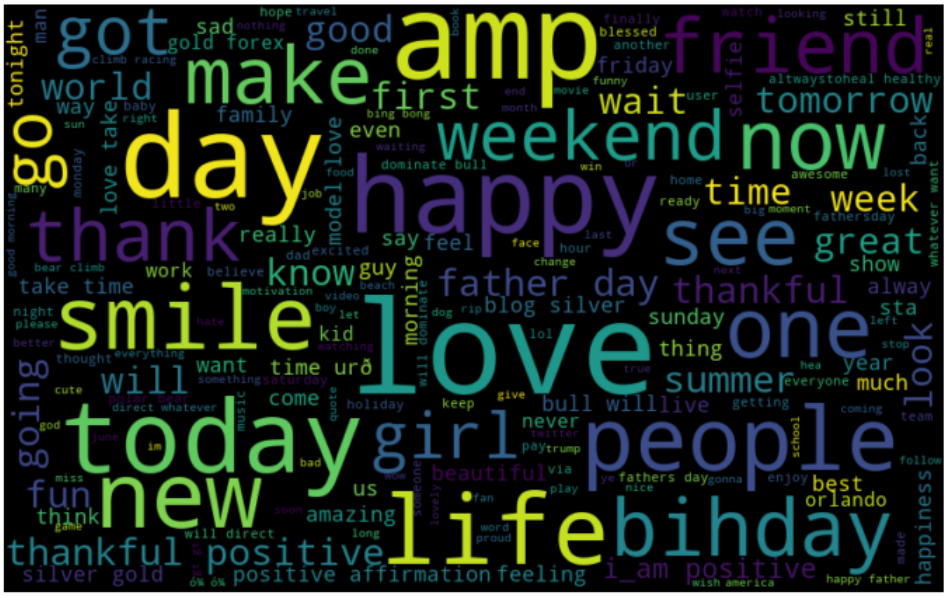
```
most_frequent = neg_htag_df.nlargest(columns="Count", n = 20)
plt.figure(figsize=(16,5))
ax = sns.barplot(data=most_frequent, x= "Hashtag", y = "Count")
plt.show()
```



5.5 Finding common words in both classes of tweets using visualization

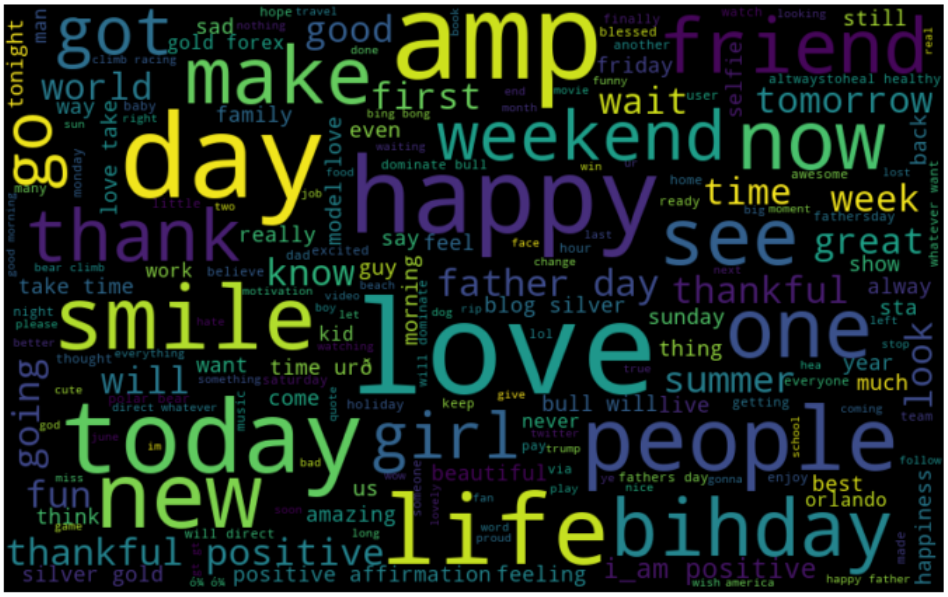
5.5.1 Positive Words

```
normal_words = ' '.join([word for word in train['cleaned_tweet'][train['label'] == 0]])
wordcloud = WordCloud(width = 800, height = 500, max_font_size = 110).generate(normal_words)
print('Normal words')
plt.figure(figsize= (12,8))
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off')
plt.show()
```



5.5.2 Negative Words

```
negative_words = ' '.join([word for word in train['cleaned_tweet'][train['label'] == 1]])
wordcloud = WordCloud(width = 800, height = 500, max_font_size = 110).generate(negative_words)
print('Negative words')
plt.figure(figsize= (12,8))
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off')
plt.show()
```



5.5.3 Applying Bag-of-Words

```
vect = CountVectorizer().fit(X_train)
vect
```

```
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
        dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
        lowercase=True, max_df=1.0, max_features=None, min_df=1,
        ngram_range=(1, 1), preprocessor=None, stop_words=None,
        strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
        tokenizer=None, vocabulary=None)
```

```
print('Total features =', len(vect.get_feature_names()))
print(vect.get_feature_names()[::5000])
```

```
Total features = 34478
['00', 'btg', 'encouragement', 'ifcarlingdidperfectdays', 'mona', 'rdoequipment', 'technology']
```

```
X_train_vectorized = vect.transform(X_train)
X_train_vectorized
```

```
<23971x34478 sparse matrix of type '<class 'numpy.int64'>'
        with 266363 stored elements in Compressed Sparse Row format>
```

Chapter 6

Methodology

In the very first stage of the proposed model, all the essential python libraries have to be imported for data set manipulation, mathematical functions, visualization, etc. Also libraries for sentiment like Natural Language Toolkit, Word cloud, etc are imported. The libraries for different models also needs to be imported. Three data sets have been incorporated for training the classifier.

The data set being used is described below and imported as a .csv file in Jupyter Notebook:

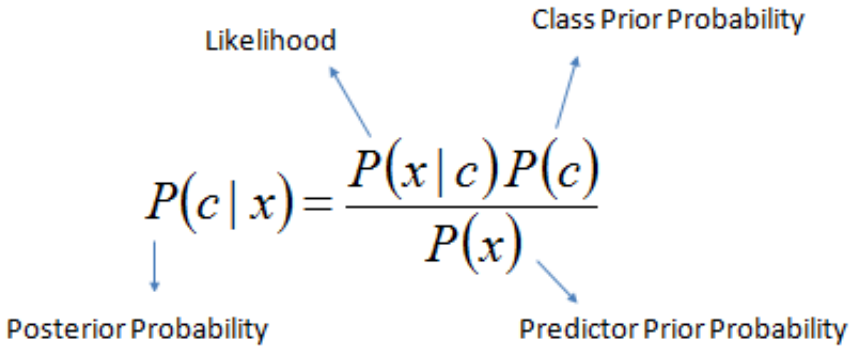
- Sentiment
- Size: 32,000 tweets

The data set is then saved into new dataframe and displayed. There are tweets which are labelled 0 are considered as positive and tweets with label 1 are considered as negative randomly.Which is followed by the PRE-PROCESSING of data: -
1. Lower Case - Tweets are converted to lower case
2. URLs - Convert www.* or https? ://* to 'URL'
3. @username - Convert username to '–HANDLE'
4. A Bar Graph is plotted for overall positive and negative tweets on x and y axis.
5. hashtag - Hashtags mostly contain useful information, so we are replacing hashtag with its word written without the hash. E.g. Black is converted to 'Black'
5. Trimming the tweet
7. hashtags will be first removed for Positive tweets and Bar Graph will be plotted for 'n' number of most frequently occurring tweets.
8. The same procedure is followed for Negative tweets and Bar Graph is plotted for the same.
9. Repeating words: In Informal conversations, a person often uses repeating characters, such as "I'm happyyyyy". We are replacing characters with a frequency greater than two with that character repeated twice so that the result for above would be "I'm happyy".
10. From the entire data set, separate Word Cloud is plotted for Positive tweets and Negative tweets.
11. The data set is then split into Training data and Testing data.
12. Three classification algorithms are used for fitting the data set and finding the accuracy of the best.

6.1 Algorithms which we are using in our project

6.1.1 Naive Bayes Classifier:

The Naive Bayes classifier is the simplest and most commonly used classifier. Naive Bayes classification model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the BOWs (Bag of Words) feature extraction which ignores the position of the word in the document. It uses Bayes Theorem to predict the probability that a given feature set belongs to a particular label. Naive Bayes is one of the most efficient and effective inductive learning algorithms for machine learning and data mining. It's competitive performance in classification is surprising, because the conditional independence assumption on which it is based, is rarely true in real-world applications. Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes theorem with the naive assumption of independence between every pair of features. Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. On the flip side, although naive Bayes is known as a decent classifier, it is known to be a bad estimator, so the probability outputs from predict probability are not to be taken too seriously.



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

6.1.2 Logistic Regression:

Logistic Regression is a classification algorithm and follows statistical method for analysing a data set in which there are one or more independent variables that determines an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. To represent binary or categorical outcome, we use dummy variables. It predicts the probability of occurrence of an event by fitting data to a logistic function. It measures the relationship between the dependent variable (Our label, what we want to predict) and the one or more independent variables (Our features). These probabilities must then be transformed into binary values in order to actually make a prediction.

$$P = \frac{e^{a+bX}}{1 + e^{a+bX}}$$

6.1.3 Random Forest:

With increase in computational power, we can now choose algorithms which perform very intensive calculations, which is discussed in our project. One such algorithm is Random Forest. Random forest tries to build multiple CART (Classification and Regression Trees) model with different samples and different initial variables. That operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of individual trees. The algorithm of Random Forest is like bootstrapping algorithm with Decision tree (CART) model. Say, we have 1000 observation in the complete population with 10 variables. Random forest tries to build multiple CART model with different sample and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction.

$$K_k^{cc}(\mathbf{x}, \mathbf{z}) = \sum_{k_1,\ldots,k_d, \sum_{j=1}^d k_j = k} \frac{k!}{k_1! \cdots k_d!} \left(\frac{1}{d}\right)^k \prod_{j=1}^d \mathbf{1}_{\lceil 2^{k_j} x_j \rceil = \lceil 2^{k_j} z_j \rceil},$$
$$\text{for all } \mathbf{x}, \mathbf{z} \in [0,1]^d.$$

Chapter 7

Result

7.1 Naive Bayes Algorithm

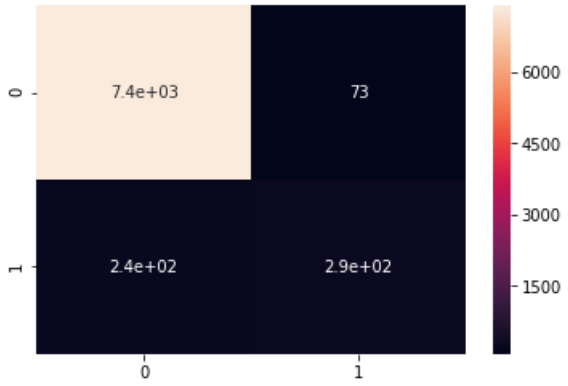


Figure 7.1: Confusion Matrix Of Naive Bayes Algorithm

7.2 Logistic Regression Algorithm

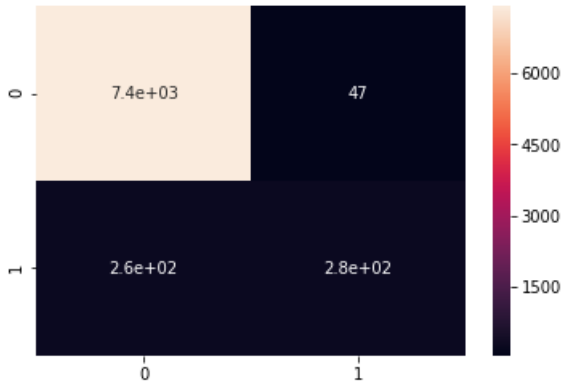


Figure 7.2: Confusion Matrix Of Logistic Regression Algorithm

7.3 Random forest Algorithm

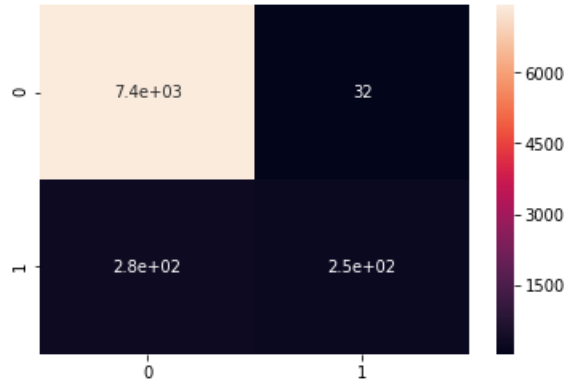


Figure 7.3: Confusion Matrix Of Random forest Algorithm

Algorithm	Model Fitting	Accuracy of Prediction
Naive Bayes	0.6502	0.9608
Logistic Regression	0.6447	0.9620
Random Forest	0.6184	0.9610

Table 7.1: Overall Calculation Of Algorithm

Chapter 8

Conclusions

We have completed our project using python as language. We have used Html and Css for output representation. Although there was a problem with merging of python with html, through number of tutorial we were able to merge it. In this paper, Naive Bayes, Random Forest and Logistic Regression based sentiment are represented for improving the overall accuracy of classifiers in the classification of tweets. We were able to determine the Positivity and Negativity of each tweet. Hence for this we apply pre-processing techniques so that proper classification is performed based on unbiased user tweets. Those were represented in the form of diagrams like Bar graph, Word cloud. Although many classifiers, data is fed as an input to the training data set, and the process we have proposed classifies the tweets into positive and negative. We have took comparative observations against the Naive Bayes, Random Forest and Logistic Regression. Here, Logistic Regression when trained 32000 tweets of data set and tested, the system shows an accuracy of 96.19 percent. Also for each algorithm, we displayed an Confusion Matrix.

8.1 Future Enhancement

We look forward to use bigger data set to improve the accuracy. Considering the emotions, determining neutrality, potential improvement can be made to our data collection and analysis method. Future research can be done with possible improvement such as more refined data and more accurate algorithms.

Bibliography

[1] Ms. Farha Nausheen,Ms. Sayyada Hajera Begum,"Sentiment Analysis to Predict Election Results Using Python",2018

[2] Ankur Goel,Jyoti Gautam,Sitesh Kumar,"Real Time Sentiment Analysis of Tweets Using Naive Bayes",2016

[3] Ms.k.Saranya,Dr.s.Jayanathy,"Onto based Sentiment Classification using Machine learning",2017

[4] Megha Rathi, Aditya Malik, Daksh Varshney, Rachita Sharma, Sarthak Mendiratta,"Sentiment Analysis of Tweets using Machine Learning Approach",2018

Appendices

Detailed information, lengthy derivations, raw experimental observations etc. are to be presented in the separate appendices, which shall be numbered in Roman Capitals (e.g. Appendix A).

Python

Python is a widely used high-level,general-purpose,interpreted,dynamic programming language.its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as c or java. The language provide constructs intended to enable writing clear programs on both small and large scale.

NLTK

Natural Language Tool Kit(NLTK) is leading platform for building python programs to work with human language data.it provides easy-to-use interfaces to over 50 corpora and lexical resources such as Word Net, along with a suite of text processing libraries for classification,tokenization, stemming, tagging, parsing and semantic reasoning.NLTK has been called "a wonderful tool for teaching, and working in computational linguistics using python." and "an amazing library to play with natural language."

Word Cloud

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.For generating word cloud in Python, modules needed are matplotlib, pandas and wordcloud.

Pandas DataFrames

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

scikit-learn

Scikit-learn is probably the most useful library for machine learning in Python. It is on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. scikit-learn is used to build models. It should not be used for reading the data, manipulating and summarizing it. There are better libraries for that (e.g. NumPy, Pandas etc.)

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits

Appendix-A: Anaconda Nevigator and libraries Installation

1.first we go to https://repo.continuum.io/archive/ here we will get all versions of anaconda that includes all versions of python.

2.Then we will download Anaconda 3.4.2 64 bit version as per out operating system that contains python 3.5 version

3.Anaconda contains many packages and one of that include a package named jupyter notebook(IPynb) that is a python interpreter and basically it is like a local html file you open in your browser.

Publication