Maynooth University
National University
of Ireland Maynooth

National University of Ireland, Maynooth
Maynooth, Co. Kildare, Ireland.

Sentimental Analysis of Opinions and their Level of Diversity in Survey Data Gathered for Killkenny Bridge Project

Kannadhasan Babu
August 2018
Master's in Data Science & Analytics,
Department of Mathematics & Statistics

PROJECT REPORT

Department of Computer Science
National University of Ireland, Maynooth
Co Kildare, Ireland

Supervisor: Joseph Timoney, Brian Davis

August 2018

Declaration

I hereby certify that this dissertation, which I now submit for assessment of the program of study leading to the award of Master of Science in Data Science & Analytics, is entirely my own work and has not been taken from the work of other save and to the extent that such work has been cited and acknowledged within the text of my work. This dissertation contains 15000 words.

Kannadhasan Babu
August, 2018

Acknowledgements

I would like to express my profound gratitude to **Dr. Joseph Timoney**, at Department of Computer Science, National University of Ireland, Maynooth for providing me valuable guidance and support to this project. His friendly nature and thoughtful ideas driven me to go extra mile. I also would like to extend my sincere gratitude to **Dr. Brian Davis** for his valuable support throughout development life cycle. He always steered me right direction whenever help needed and provided innovative ideas for implementation.

I would also like to extend my sincere thanks to all the faculty member of Mathematics and Statistics Department, National University of Ireland, Maynooth.

Abstract

Digitalization and predicting the future from public opinions is one of the modern developments in text processing and analysis. An important case is when there is infrastructural developments in a town or city, and opinions from the public are much needed to identify the levels of positive and negative sentiment and the variety of topical issues. This is so that the government bodies can be aware of all stakeholder requirements and can make decisions to obtain the broadest levels of satisfaction. This can be achieved by using algorithms for sentiment analysis which facilitates a process of automatically extracting information from the set of gathered stakeholder data. The output should then be visualized in such a way that illustrates the prominent features of the sentiments.

This thesis deals with analyzing the sentiments of recent project to construct a bridge in Kilkenny city. The first goal is to examine a database obtained that holds opinions sourced from the public. These opinions span many aspects with regard to impacts it would have, reflecting the many benefits and drawbacks of the project. A second aim is to find groups of similar opinions in relation to aspects of the activity using Machine learning and Clustering techniques. A related goal is then to ensure that a sufficient breadth of opinions are discovered, and particularly that a representative set from the total that would be deemed to be diverse could be obtained. The results are presented in the form of visualizations that should reflect these features of the views of stakeholders.

Table of Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

The Internet is used by people across all over the world. On a yearly basis the number of users keeps on increasing in which the proportion of social media users is also an increasing trend. Figure 1.1 below shows the growth of social media users from the year 2010 including a prediction until 2021 [13].

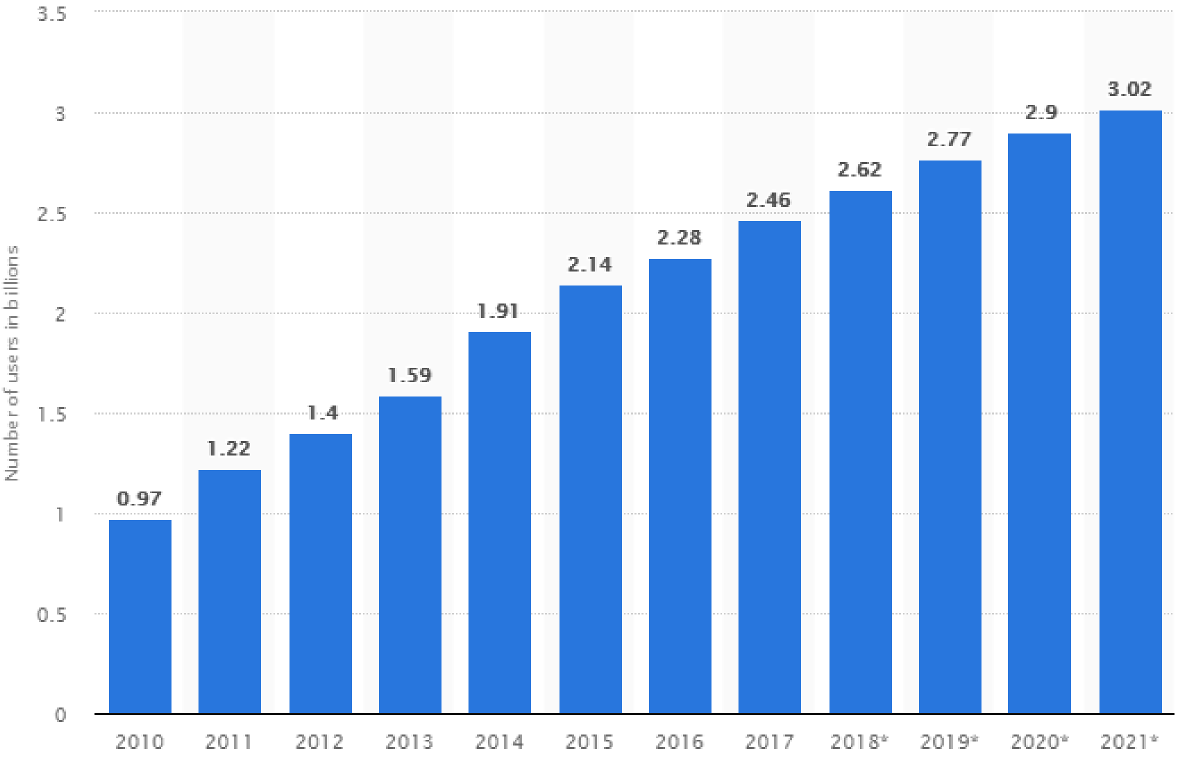
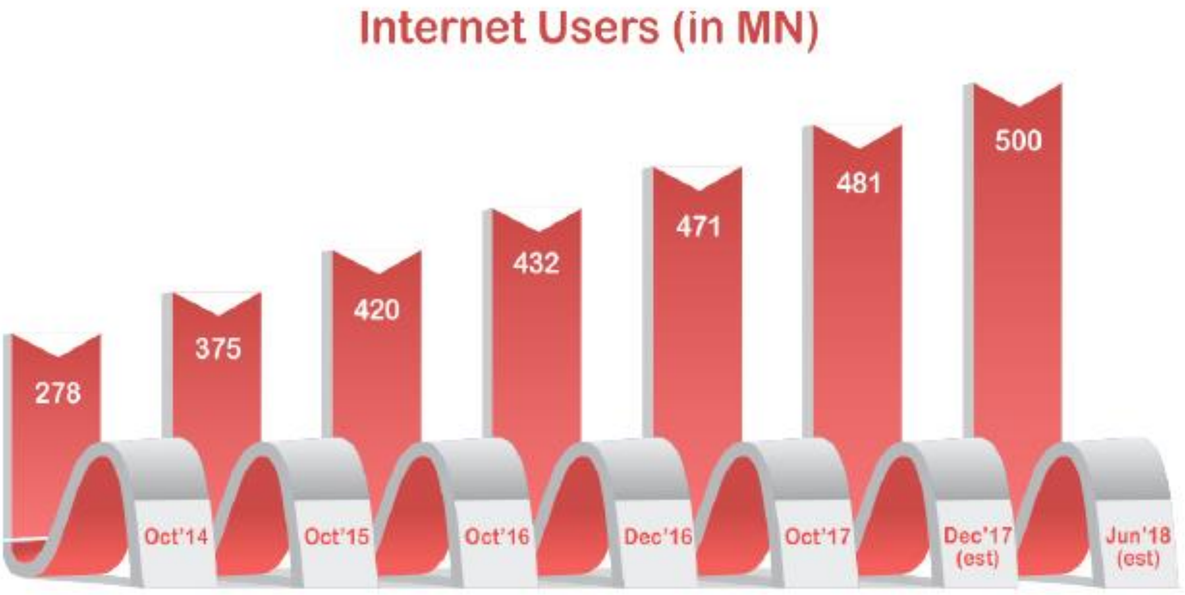


Figure 1: Number of Social Media Users across the world [13]

The increasing numbers implies that social networks such as Facebook, Twitter, Instagram, google plus etc., contains huge volumes of information about people's opinion on every issue in the society. Using this information, we can do analysis on trending topics across any platform.

Many organizations are taking advantage of these opinions to identify the users' expectations on particular products and to improve their customer service in their business. A practical example of this is digital marketing giants like Amazon, eBay, Flipkart, and Alibaba have invested significantly into sentiment analysis to predict the future of any product they have and the associated revenue. There are other topics that appear which are also important for analysis such as election campaigns, promotion of celebrities or any political leader, and advertising movies and stage shows. Also, it also became a medium to debate on any ongoing issues which will provide views of people from different part of the world at one place.



Source: IAMAI & Kantar IMRB I-CUBE 207, All India Users Estimates, October 2017

Figure 2: Internet users in India [14]

The above discussion conveys how opinions on a topic can identify what people think, and is one of the primary requirements when making any decision. If we look at the past before the evolution of internet, the way of knowing about opinions to buy any new bike, we consult with friends; to know the favor of people political party, we have to talk to each other to get the information. The processing of extracting the information is quite challenging so we cannot predict anything unless if we use or if we come across the fact in a journal. This has totally changed after World Wide Web came into picture. The Internet has become the main source of opinions from a large group of people to know their real experience about the product which is unknown to us. So, we can see that there is a significant difference from the past to present. If we see the internet usage in country like India, which has the second largest number of users in the world, there are over 500 million of users out of which 226 million active users are using social media. This is very clear from figures 2 and 3 respectively. The numbers are quite large, so they are providing opinions to a large group of strangers across the web [14] [15].

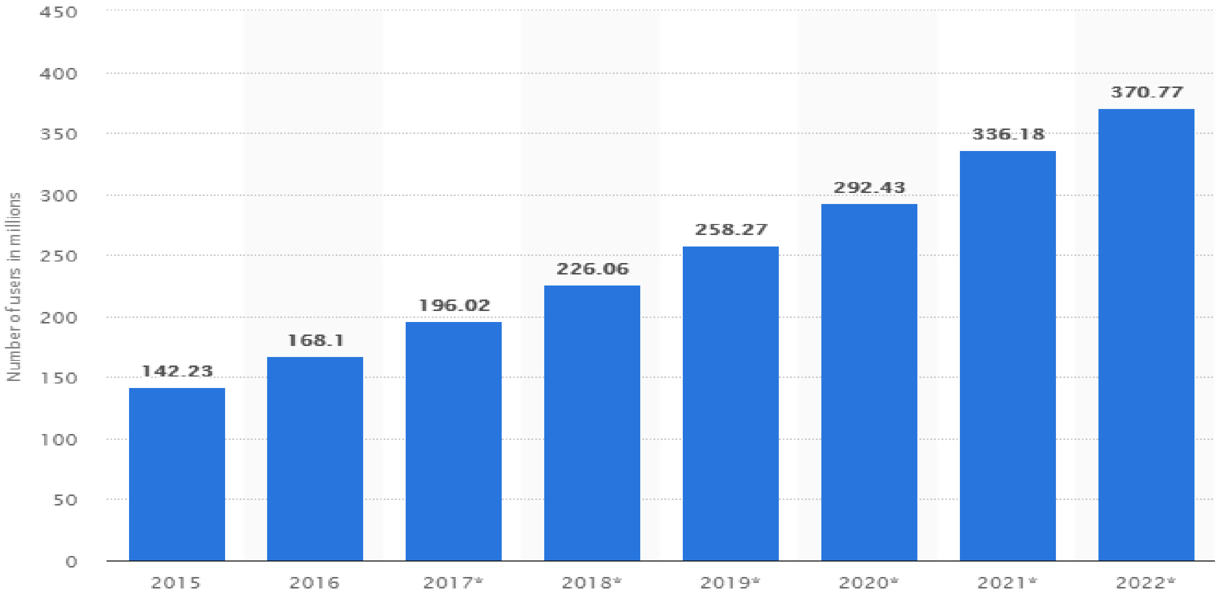


Figure 3: Social Media users in India [15]

In this thesis we have taken a real issue that took place in Kilkenny about the Central Access Scheme (CAS) [45] plan to construct the bridge in the town. There were many people opposing this plan as it impacts on natural resources and makes the city less attractive to tourists. People who are living in the Kilkenny have expressed their opinions about the bridge plan on various social media platforms. We have gathered those data from Facebook, newspaper, journals, radio, change.org and from other sources with the help of the CIVIQ company. Using this data, I have identified the topics which convey the opinions that people hold about the plan. Topic modelling is an effective way to identify topics from a large set of data and produces a set of topics with their level frequency. It is important too to know the type of opinions expressed to check whether they are positive or negative or neutral. I have carried out sentimental analysis using Natural language processing which provides the result about the sentiments of the opinion based on machine learning algorithms. Topic modelling outputs are visualized in graphical form so that it is easy to interpret the results.

1.1 Motivation

Many large projects can have multiple stakeholders. Each group can have a variety of sentiments or opinions regarding the project. Gathering and analyzing this is a very important activity in the project planning. As the more they can be incorporated into the final decisions, the more 'buy-in' and enthusiasm for the project can be generated, and the more opposition can be minimized. However, the amount of data collected can be large, so much so that manual analysis is very time consuming or almost impossible. In such scenarios having computerized tools that can assist is of vital importance. Fortunately, in the field of Machine learning there are many text analysis tools that could be configured to this task. The aim of this research is to investigate how such a tool could be created that can analyze the opinions collected surrounding a Bridge replacement project that happened in Kilkenny city in 2015. The aim of the data collection was to identify a diversity of opinions that represented the many aspects of the perceived impacts of the project on the stakeholders. A large data set was collected using surveys and social media. This set was digitized so it was ideal for machine learning testing. The questions arising were which techniques work best and how should the results be presented that will highlight their most important features.

What is Opinion?

Opinion is basically a subjective statement describing what a person believes or thinks about something. Each context can be proved either right or wrong. It depends upon the person and their background, culture, and context so each person thinks everything in a different way. The right way of doing analysis is by having factual information about the particular topic and should decide the sentiments based on the facts. There are multiple elements that we need to take into consideration in order to classify the opinion. [21]

Basic opinion representation:

Opinion holder: Whose opinion is this?

Opinion Target: What is this opinion about?

Opinion Content: What exactly is the opinion?

If we have answer to the above three questions, then it gives a complete idea about the opinion [21].

When a person reviews about a product, we know each part of opinion from the review.

My own review: "iPhone is one of the best smartphone I have ever used"

From the above review we can easily identify the part of opinion. The parts are:

Reviewer: Myself

Product: iPhone 7plus

Content: My own review

Situation: After using iPhone for a year

Time: 2018

Sentiment: Positive

In the above example the context of the opinion is so obvious, and all the information is straight forward because we know about opinion holder and opinion target. So, it is very easy to analyze and interpret the results.

Let's take another example from the news [21]:

"To alert people about the **flood** that is forecasted from weather people, **Metrology department head** came forward and said that this will be **worst** historical **storm and flood** ever in the rain history of **India** – **El Nino flood,2015**"

Opinion Holder: Metrology Department head

Opinion target: Flood, El Nino, Rain history, Storm

Sentiment: Negative

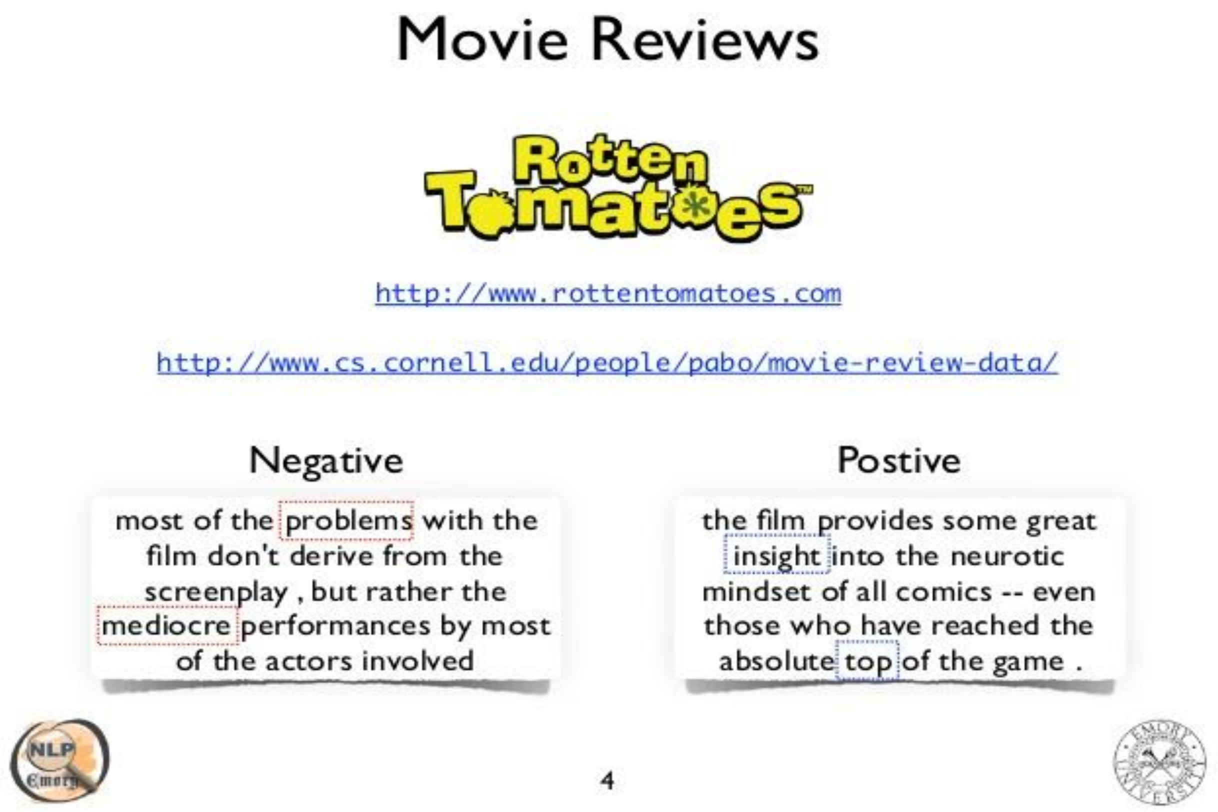
From the above example we can see that here the opinion is very harder to mine and analyze. For this type of opinions, need deeper Natural language processing. It is more difficult the analyze the opinions from the news since it has many targets, so it makes harder.

What is Sentiment Analysis?

The process of extraction of feelings of an opinion, and its attitude with associated emotions is called sentiment analysis. To analyze these sentiments from a body of text is the challenging part as the opinions varies from one person to another and each person’s way of expression is particular to them. In order to get such information with accuracy, there are Natural Language Processing techniques available that can be used.

How sentiment analysis address problems?

A steady growth for any company is an important aspect for the future. The Retail industry, customer service industry, buying and selling goods mainly depends upon the sales of products. A shortfall in the sales would be difficult to understand the factors behind it and what drives to happen such fall is unpredictable. If we take an entertainment industry, gaining box office success is a goal for a movie. But the real scenario is that only very few achieve a significant success and most of them fail to recoup the invested money. To predict such problems before it occurs, we can design a model of sentimental analysis which can be applied to forecasts in business. Bing Liu, a professor in University of Chicago, developed a model which predicts the future performance of sales using blogs. There are variety of blogs which provides huge amount of information for various sectors. All these data can be analyzed which will provide huge amount of information and we can find a way predict sales performance. Similarly, for movies box office prediction, we can utilize the same genre of movie’s reviews from the move rating boards like IMDB, rotten tomatoes and other websites. These are very popular for providing reviews. Figure 4 explains about positive and negative sentiment reviews.



Movie Reviews

http://www.rottentomatoes.com

http://www.cs.cornell.edu/people/pabo/movie-review-data/

Negative

most of the problems with the film don't derive from the screenplay , but rather the mediocre performances by most of the actors involved

Postive

the film provides some great insight into the neurotic mindset of all comics -- even those who have reached the absolute top of the game .

4

Figure 4: Movie review sentiment analysis

In figure 4, there are two types of comments expressed about a movie. Reading such comments provides insights into the sentiments. Since it is just two comments reading and analyzing is an easy way but when we have larger volumes of feedback about a movie, then reading all such comments would be very challenging. Such analysis with larger volumes of data can be done in a fraction of time using sentiment analysis concepts in python programming with machine learning algorithms. In contrast to the other mode of expressing opinions, there is video recording opinions. Any contents which came from humans are basically subjective and opinion rich. The task is how we going to mine and analyze the opinions buried in the text.

1.2 Problem Definition

As described in the motivation, sentiment analysis can able to derive solutions from the opinions expressed by people. Now the question is how it provides solution.

- Using topic modelling, can we able to identify the topics of the larger set of opinions in Python using Latent Dirichlet Analysis (LDA) method? How well are the topics found related to the actual topics? Can we find sentiments for each topic?
- Using machine learning algorithms, how can we find the overall sentiments of the given opinion and how accurate it is in explaining the sentiments and what are all the ways to compute the error?
- How useful is deep learning a solution for machine learning?

1.3 Project Objective and Solution

The main aim of this project is finding sentiments about the Kilkenny bridge plan scheme. People in Kilkenny have provided their feedback about the scheme and from this group of opinions, the main motive is to find sentiments using machine learning algorithms. The scope of this project is working on real time data and analyzing those results which provides insights on taking decisions. The developed model should be able to process the any large volumes of opinion data predominantly works well for infrastructure improvement-based opinions.

The sentiments can be classified into three types: positive, negative and neutral. For example, **holder1** says this bridge is beneficial for traffic reduction, **holder2** says the scheme is to make corruption and **holder3** says the plan is moderate. If we label the sentiments then holder1 opinion is positive, holder2 opinion is negative and holder3 opinion is neutral. The result of this project aimed to provide sentiments for each opinion for the bridge scheme using natural language processing technique and finding topics of the opinion based on the frequency value calculated for number of times the topic repeated in the data set.

1.4 Thesis Structure

This thesis has for four chapters which explains all the process, work carried out from starting of the project till the completion. The chapters are: Introduction, Background work, Analysis and Solution.

- ***Introduction (Chapter 1):*** It provides brief idea about why sentiment analysis is required for this project and in general how it is useful for all the sectors. And then in motivation section I have described about the different types of opinions and about sentiment analysis. This led a way to explain what motivated to do this project.
- ***Background Work (Chapter 2):*** In this chapter I explained about background of natural language processing techniques and various approach of opinion mining. I have done few researches on previous topic-based sentiment analysis and also on civil based domain in the literature review section.
- ***Design and Methodology (Chapter 3):*** This chapter explains about the tools and technologies and different machine learning techniques used in this thesis.
- ***Implementation (Chapter 4):*** This chapter explains about the implantation steps algorithms stated in chapter 3.
- ***Analysis and Solution (Chapter 5):*** Discussed about analysis and result.
- ***Conclusion (Chapter 6):*** This chapter concluded the analysis, summarized about over all work done in this project and future work for enhancements and further developments
- ***References:*** I have listed out all the books, journals, internet materials referred.

Chapter 2

2. Background Work

In this chapter I intend to describe about the past work in sentiment analysis using Natural Language Processing Techniques, machine learning, and civil based sentiment analysis.

2.1 Natural Language Processing

Natural Language Processing is a technique of making computers able to understand and automatically manipulate human languages. It is an interaction between the computers and humans through programming and artificial intelligence. There are lots of text that can be accessed through World Wide Web; because the availability of electronic documents like newspapers, journals, articles and books has increased a lot, and additionally there are plenty of opinions conveyed by people for a variety of products and issues that are available in written text format in social media platforms. All these documents are written in different languages. The task of making the computer to understand those language is called natural language processing. Steps and guidelines are there to be followed that automates the process using computer programming languages.

2.1.1 NLP Approach

There are two main approaches to process the languages: Rule based, and a machine learning approach that includes deep learning.[1]

- **Knowledge Based:** This is one of the traditional approaches of NLP. It works based on the hand-written rules by a NLP specialist [1]. To do linguistic processing with this approach, a grammar knowledge of that language is required along with human intuition. This is one of the very useful methods if the task is defined by rules. If we exclude the rule-based method in this approach, then it is problematic, and it is difficult the figure out the errors. The main advantage of this method is debugging is very easy because the error can occur only for the rules so if we identify the rules that are creating the problem, then resolving the problem is quite easy. But defining the rules is a time-consuming task and, if the requirement changes, then developers have to rewrite the rules again. That’s one of the primary drawbacks [1].

- **Machine Learning Approach:** This is a very advanced method in comparison to the Knowledge-based approach because of its ability to processing large quantities of data in a fraction of the time and problems are less compared to the previous approach. Unlike the knowledge-based approach, it does not consume much time in evaluating the result and also knowledge of linguistic expertise is not required. If we have the required training data set in place, setting up a system is very easy. However, creating a training set is quite a challenging task because of the manual work involved in the creation process. This approach has a dependency on the training data that must be adapt to the text, the domain, and to the new languages. In rule-based approach it can easily adapt to the new text and languages. This type of problem can be handled by having unsupervised learning methods though the accuracy in the results may not be precise. The Machine learning approach have a history of performing well for sentiment analysis. Frequently used algorithms like Naïve Bayes classifier, Support Vector machine have provided good results [46]. We will see in detail about those algorithms in the design section [1].
- **Deep learning approach:** This approach one of the recent developments in Natural language processing. It is performs well when compared to the traditional and machine learning approach. Deep learning is a part of machine learning and uses neural networks which are inspired by human brain. This type of approach has tendency to learn on its own about each object that we use in the model. Like machine learning, the data needs to be trained well to use it in the algorithm.

2.2 Opinion Mining

As I have already explained about opinion in the motivation section of Introduction chapter, it is important to know what others think on anything to make decisions. Social media has become one of the powerful platforms which contain a huge quantity of opinion data [5]. This platform provides freedom to express their opinion on any matters like movie, smartphone, restaurants, automobiles, and infrastructure. If the number of reviews for each product is large, then it is difficult for customer to take decisions based on the review. Mining the data and retrieving the information from each opinion using NLP makes easier to understand about the product which will provide clear insights of the reviews. Finally, we have to determine the sentiment of each opinion. The sentiment can be classified as Positive, Negative and neutral [3].

2.2.1 Document Level Opinion Mining

In this type of opinion mining, the overall sentiments associated with the whole documents should be discovered. We should assume one single entity and based on that we have to do analysis. As we described there are huge quantity of web documents are available in the internet [5]. All the news articles, journals, papers, project reports are available in the form of electronic documents in the web. To compute and to do linguistic processing of whole document, single entity-based approach provide the clear view of how analysis should be done. Opinions that are expressed in the whole document should be considered as single entity and then we can apply machine learning algorithm which will provide polarity of sentiments.

2.2.2 Sentence Level Opinion Mining

From the name itself the meaning is derived; sentiment analysis is done on sentences in the opinion instead of analyzing the whole document. This can do more in depth analysis of text irrespective positive, negative and neutral sentences which will help to find the root cause of whole document. It can able to identify the strong opinions and can retrieve most of the information in all levels of the data from the group of opinions.

2.2.3 Word Level Opinion Mining

This is very low-level granularity analysis which refers to single entity and it expresses one polarity of sentiments for that entity. Human readers can easily identify the sentiment of the word. In the process of attempting to capture the sentiments will lead to express the sentiment of each entity. Sentence are splinted into words and each word are considered as an entity [8]. Let's see the example below:

"iPhone has very good display and all the features are useful, but price is very expensive"

The above sentence focused on three entities: iPhone display, iPhone features, iPhone price. For any given sentence, in the word level mining it splits the sentence into smaller segment; the segments are referring to single entity and for each segment conveys sentiment towards it.

2.*3 Topic Based Sentiment*

Topic modelling is one of the well-known machine learning technology widely used for opinion mining, twitter sentiment analysis, and other domains. Topic models are important to discover the structure of a specific entity in an enormous amount of data. It is the process to finding a list of topics that occurs within a given set of opinions. It cannot interpret the meaning of words and concepts in the text. Instead, it can mingle the related words together in a group within the model. Each group holds words that correspond to the topic. This process will repeat again and again until all the possible distribution of words in the group is done. That group is considered as one topic. After extraction of topic from the opinions, then score should be calculated for each topic and based on that sentiment is discovered [9].

2.3.1 Latent Drichlet Allocation method:

This is simplest topic modeling approach in machine learning. It is very popular for topic based sentiment analysis. It is applied to documents that has random mixture of topics.

2.4 Literature Review

As I have outlined in the introduction part, I have done sentimental analysis in this thesis and I have explained brief idea of it in the chapter 1 motivation part. The domain of the data is about civil based, infrastructure development, urban planning and smart cities. In this section I am going to explain about the sentiment analysis work that has been done in the past using Natural language processing and previous sentiment analysis in civil domain.

Sentiment analysis is one of the popular techniques now to identify the feelings associated but topic based sentiment is one of the rare approaches in the opinion mining that too in civil domain.

- Jianfeng Si Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, Xiaotie DengExploiting [38] *Topic based Twitter Sentiment for Stock Prediction.* This project is about doing topic based sentiment analysis for stock market prediction. Using LDA method they have identified the topic, and for each topic they find sentiment based on the word distribution which helps to build a sentiment time series. They did daily stock prediction based on topic characteristic acquired from twitter data. using non-parametric approach analyzed topic-based sentiment for twitter data. In 2013 this was one of the new approaches in topic based sentiment for stock market prediction in autoregressive framework.
- Kaoutar Ben Ahmed1 , Atanas Radenski2 , Mohammed Bouhorma1 Mohamed Ben Ahmed [39] *Sentiment Analysis for Smart Cities: State of the Art and Opportunities*. This paper is related to sentiment analysis in civil based domain since it involved in smart cities. This article provides information on challenges in sentiment analysis for smart city platforms and applications of smart city domain. This article provided clear insights of sentiment analysis is very useful for smart city development using machine learning algorithm with supporting papers. For smart city development, sentiment analysis is one of the major contributions.
- Pierre Ficamos, Yan Liu [42] did a project on *Topic based approach for Sentimental Analysis on Twitter Data*. The idea was to split the test and train data set. From train data set extracted the topics using topic modelling algorithms and then trained the model using those topics. This approach was to increase the accuracy of prediction rather than having single model for every topic. The method is first the collected

tweet is preprocessed and then features are collected in bag of words. From this, using topic modelling technique probability distribution is estimated. Threshold is fixed. Topics are selected if the probability is greater than threshold, and those selected topics are trained to estimate the sentiment. The topics that attained highest probability was the final estimation [42].

- Bin Lu, Myle Ott, Claire Cardie and Benjamin Tsou [43] did multi-aspect sentiment analysis with topic models. There are two new approaches of topic model was used: Unsupervised and weakly supervised topic modelling are applied to multi aspect sentiment analysis to find which performs better. Multi aspect sentiment is first label the data using Multi aspect sentence labelling and second predict the star for each reviews using Multi aspect rating prediction. Features were extracted using topic models and predicted the result with and without topic. Among the two topic models weakly supervised topic approach was performed well and also for multi aspect rating prediction.
- Chenghua Lin, Yulan He [44], proposed Joint Sentiment/ Topic Model for Sentiment Analysis. In this project, unlike the usual approach of supervised learning technique which requires labelling the opinion for test and train, unsupervised learning is used along with LDA (Latent Dirichlet Allocation) algorithm for topic modelling. This model is analyzed with movie review data set for sentiment classification. It detected both sentiments and topics simultaneously. This was focused on document level sentiment classification and results were produced sentiment analysis on extracted topic. JST predicted based on the filtered subjectivity lexicon and, with subjective and without subjective MR dataset. This new approach actually performed well but compared to SVM classifier results were slightly less.

Chapter 3

3. Design and Methodologies

This chapter provides clear view about design and Methodologies that I have used in this thesis. In the first section I have explained about tools and technologies that I have used in order to provide clear understanding to the users about environment. In the second section explained about NLP preprocessing and machine learning algorithms to provide basic understanding of those algorithm and how it is useful in sentiment analysis. This chapter will be helpful to understand the implementation.

3.1 Tools and Technologies

3.1.1 Anaconda [49]

We use anaconda platform for any programming related work in this thesis. It is a free open source software. It facilitates for application development in R and Python programming languages for data science and machine learning related work. Inbuilt package in the Anaconda supervised by package management system conda.

- ***Anoconda GUI:*** Within the anaconda package there are many built-in applications are available which act as a graphical user interface for programming. We are using Jupyter notebook and Spyder for development.
- ***Anoconda prompt*** is there for any command line work. This prompt is useful to install any packages and to update the existing package to the latest version. Anaconda is very easy to manage since it supports multiple versions of python.
- ***Jupyter Notebook:*** This is one of the open source software web applications that allows us to create and execute the python code. It is built-in application of anaconda. It has a markdown which is a markup language. We can easily visualize all the results together in a single page. One of the advantages is the documentation of Jupyter. There is an option to export the code with results in various file format like HTML, notebook (ipynb), PDF, markup, and python(.py).

3.1.2 Python:

Python is object oriented, high-level programing language with dynamic semantic. For any rapid application development this is most suitable programming language because of its data structures and dynamic binding. This is widely used for data science research activities. In the course of time, there are many tools and packages have been built in python especially for data science. The features of python assist us to analyze any type of data. [27]

- **Why python:** In recent years, data science gained a lot of attention in the community of marketing, medicine, engineering, banking, and computer science. This is mainly dealing with converting enormous quantity of data into meaningful strategies. The records are available in the internet provided by users expressing their choices on anything. These records are collected together, and data scientist performs the research to bring a logical conclusion. Python is the one tool that helps to achieve this strategy for any analytical and computational work in the data. It is well known for its immense of libraries for data manipulation. It can easily adapt to the existing environment. This tool furnishes solution to the most difficult problems [23].

3.1.3 Python tools and libraries:

As I mentioned in the section 3.1.2 in recent times there are plenty of tools have been developed in python which automates most of the functions. I have used the following tools in this project to achieve the result:

- **Pandas:** Pandas library is one of the powerful data analysis tools which provide simple data structures. Python is well known for data cleaning and preparing the data but does lack in modelling and analysis. To fill this gap, pandas was developed which enables us to perform data analysis in python program without depending upon the domain. [25] The main features [25] [26] of pandas are
    - Converting the data into a data frame object for data manipulation
    - It is an easy way of managing the missing data which is also represented as NaN (Not a number) in floating and non-floating-point data

  - Adding and deleting the columns in data frame object and higher dimensional objects
  - Automatic and manual data alignment
  - Group by functionality helps to combine the data, split the data
  - Reading the different data's like flat files, CSV, excel, and database files
  - Joining the datasets together.

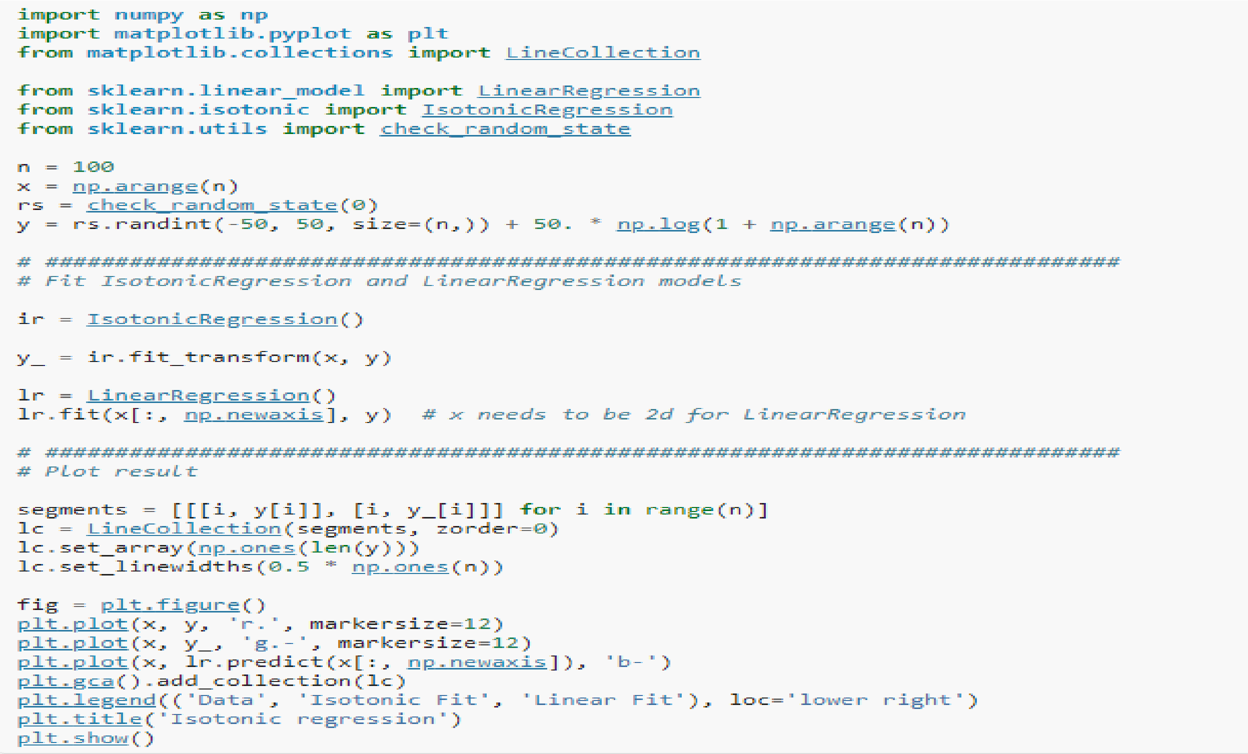
- **Numpy:** It is a widely used useful package of python that mainly does scientific computation. It supports multi-dimensional arrays and matrices. It has a collection of mathematical functions. One of the core functionalities of numpy is ndarray which deals with n-dimensional arrays. [50]

- **Keras: [51]** It is library developed in python focused on constructing deep learning models which runs on the top of either Theano or TensorFlow. It is useful for most of the **n**eural network research and development activities. [51] It has four main principles, they are:

  - **Modularity:** Any model that we develop using Keras can easily interpret the meaning as a sequence of fully-configurable modules that can be mingled with any necessary restrictions as possible. To be precise, neural layers, cost functions, optimizers, initialization schemes, activation functions, regularization schemes are all standalone modules that you can combine to create new models. [51]
  - **Python**: Since the model can be developed in python using this tool, it is easier to debug, update which makes Keras more suitable.
  - **User Friendly**: It is designed in such a way that provide a user-centric service which reduces cognitive loads so user action is minimal and also error handling is easier as it provides clear feedback to take an action to resolve the error.
  - **Extensibility:** In the existing model we can able to add new model. It is easy to update the models for any additional functions and classes. Using Keras, it is very simple to create a new model.

- **Neural Network:** It is machine learning technique which can learn the objects on its own and it can add meaning to it after inputting the data to the model. This type of model can be developed efficiently using the Keras library. It is outperformed well for natural language processing that are aimed to find sentiments.

- **Natural Language ToolKit (NLTK):** It is a very popular platform for building programs to analyze human language data. It provides cool features for text preprocessing such as tokenization, stemming, lemmatization, tagging, parsing, normalization, and a sentence splitter. It is useful for data scientists who are doing research in linguistic processing. It is available in all platforms such as Windows, Linux, and Mac OS. [52]
- **SKlearn:** It is one of the largest libraries which has inbuilt functionality of machine learning algorithms and techniques of data science that can be applied in real-time for analysis. SK learn facilitates algorithms such as classification, clustering, regression, SVM (Support Vector Machine), random forest, gradient boosting, k-means, Naïve Bayes classifier. [54]



```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.collections import LineCollection

from sklearn.linear_model import LinearRegression
from sklearn.isotonic import IsotonicRegression
from sklearn.utils import check_random_state

n = 100
x = np.arange(n)
rs = check_random_state(0)
y = rs.randint(-50, 50, size=(n,)) + 50. * np.log(1 + np.arange(n))

# #############################################################################
# Fit IsotonicRegression and LinearRegression models

ir = IsotonicRegression()

y_ = ir.fit_transform(x, y)

lr = LinearRegression()
lr.fit(x[:, np.newaxis], y)  # x needs to be 2d for LinearRegression

# #############################################################################
# Plot result

segments = [[[i, y[i]], [i, y_[i]]] for i in range(n)]
lc = LineCollection(segments, zorder=0)
lc.set_array(np.ones(len(y)))
lc.set_linewidths(0.5 * np.ones(n))

fig = plt.figure()
plt.plot(x, y, 'r.', markersize=12)
plt.plot(x, y_, 'g.-', markersize=12)
plt.plot(x, lr.predict(x[:, np.newaxis]), 'b-')
plt.gca().add_collection(lc)
plt.legend(('Data', 'Isotonic Fit', 'Linear Fit'), loc='lower right')
plt.title('Isotonic regression')
plt.show()
```

Figure 5: Example of SKLearn [53]

Figure 5 shows how to import the SKlearn library to fit the model for linear and isotonic regression. Linear regression is used for predictive analysis to find which predictor variables are good for predicting the outcome. To be precise, we have to find the significant predictors. To fit both the models, input parameters x and y are required to predict the response y for a given predictor variable. A graph is plotted for the obtained result. Similarly, there are simple linear regression, multiple linear regression, ANOVA, and many other models are there that can be fitted using the SKLearn library in python.

- **Gensim**: [27] It is designed in a way to extract the topic from an unstructured document. The main feature of this library is it can able to convert words to vectors using the Word2Vec algorithm, fast text. The Latent Drichlet Allocation method is very popular for topic modelling. These algorithms help to find the semantic structure of the given data set by inspecting statistical patterns within the body of the document. Since the algorithms are unsupervised, it doesn't require any special inputs just the text document that contains the group of opinions is enough for the analysis and to fit the model. Some of the features are:

  - **Memory**: The training data is not required to be populated in the RAM one at a time. The data can reside in the local disk and can be loaded through mmap. The loaded data can be availed by multiple processes where it is needed so it reduces the usage of RAM. Thus, it can process a large web scale corpora.
  - It is very popular for vector space algorithms; for a vector-based approach this tool is the best option because it supports the algorithms that help to extract the topics from the opinion.

3.1.4 NLP preprocessing:

As described above in the data cleaning process, an initial process of cleaning of data has to be done manually to correct the basic errors. Before proceeding to the next step of cleaning, in the first process itself it is important to remove the disturbing sentences or text for data preprocessing. These are the following steps that I have taken to preprocess the data. [1]

- **Tokenization**

  It is the process of splitting the sentence into individual text, which contains words, letters, numbers, and special characters. It is recommended for any text preprocessing, since in many machine learning algorithms and POS tagging an individual word is required as input. This tokenizer should perform in a way that it should split all the words in any type of sentences so that it does not affect the results. [1]

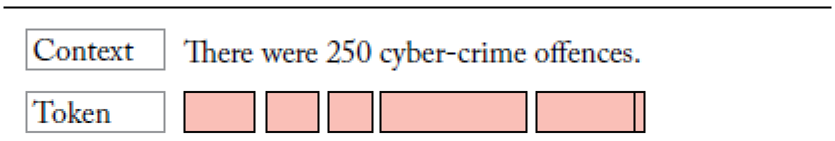


Figure 6 Example of tokenized sentence [1]

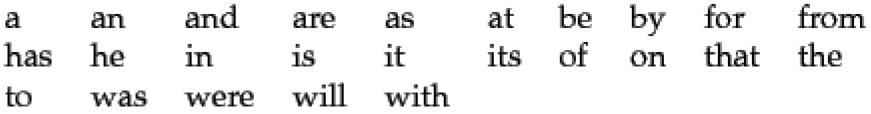
Once sentences are split into words, then we can add features to each token. This depends upon type of token and it should be reusable and adaptable irrespective of the domains; it should be able to handle mathematical and chemical formula's. At this stage, if any error remains in the data set, then it will lead to problems in the later state of analysis. Some of the errors in the text are spelling because in tokenization text aren't scanned for spell check, unwanted and irrelevant characters. So, the tokenizer should be customized in such a way that it should be handle all the errors.

- **Challenges:** Tokenization challenges rely on the type of language that we use. Each language way of scripting is different from one another. If we take English, words are separated by space but Chinese and Thai are different as it doesn't follow any boundaries in the structure of the words. This will affect tokenizing. To tokenize such language words additional lexical and morphological information is required. After reviewing the collective of words from various languages, the structure is categorized into three types:
- ***Agglutinative:*** When the words can break into small parts are Agglutinative type of words. Tamil and Japnese are examples of such words.
- ***Inflectional:*** There are no proper boundaries in the words. Example: Latin

- ***Isolating:*** The structure of the words is different. Words are not segmented into smaller parts. One example is the Chinese Mandarin language

- ***Stopwords:***

  This is one of the text preprocessing steps to exclude it from using the most frequent meaningless words in the opinions. The method of collecting the stop word from the data set is by calculating the frequency of meaningless words and collecting the words which have a greater frequency value which is irrelevant to the context or the domain on which opinions are expressed [17].



a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

Figure 7: Most common stop words [17]

The main use of preventing stop words is it reduces the storage space in the system. But sometimes if stop words are not indexed, then it tends to cause minor harm to the results because for a few contexts it actually completes the meaning of the sentence. Let’s take an example: “The next train from Dublin to Cork is at 5 PM”. Here, the stop word TO makes more sense and gives meaning to sentences. If we remove the stop word ‘TO’ from the sentence, then the meaning is lost. So, stop words should work in a way without affecting the meaning of the sentence.

- ***Sentence Splitting:*** it is the process of separating the running text into sentences by detecting the punctuations such as comma, full stop, question and exclamation mark to determine whether it’s the end of sentence or it represents something else. The common rule in the splitter is usually full stop denotes it’s end of sentence unless if it’s Mr or Mrs. Issues arises when the line break is used for an address. In such cases recognizing the sentence structure is challenging. There is more difficult situation as well especially when we use table, mathematical formulae, or titles. These types of text make the splitter complex. This is one of the places where it easily prone to error but because of its structure there is the possibility to skip it while processing. Let’s discuss about challenges [28] that exist in sentence splitting:

- ***Language Dependence:*** To analyze any human language the highlight has always been on the script of the language since computers can more easily understand the text transcript rather than recognizing the actual speech. Language always differs from place to place because each society of humans speaks different language. According to David D Palmer [28] approximately there are thousands of unique languages and dialects, but at the present time only few of the languages are digitalized in the system which represents the elements of it.

  The speech and pronunciation of a language has features; similarly writing also possess set of unique features. The writing system can be divided into three types depends on the way of writing: [28]

  Logographic: substantial amount of individual symbol depicts words

  Syllabic: each symbol depicts syllables

  Alphabetic: Each symbol depicts sound

  Apart from the different type of symbol that we use in the writing, there are orthographic conventions which implies the boundaries between each unit such as words, sentence, letters or syllables. Not all language writing has explicit mention of boundaries in words and sentence, some doesn't have. For example, Thai text have neither of them. If we take English, there is white space for each word and sentences but that doesn't enough to split the text unambiguously. It is important to study the language orthographic conventions and structure of it so that while visualizing the elements of such language in the system an explicit program is required to make changes in software level and sometimes even hardware level of changes might be required. The following table provides details about the list of languages boundaries for end of a word and sentence.

Table 1: Orthographic conventions of languages [28, Page 3, Language Dependence]

Language	Boundaries for words	Boundaries for Sentence
Thai	No	No

Tibetan	No	No
Korean	No	No
Chinese	No	Yes
Japanese	No	Yes
English	White space	White space

The first step for sentence splitting is understanding the writing system of a language. Hence, the writing methodology of each language in the world result in language and orthographic specific. For a successful text segmentation all these features must be considered.

- ***Character-set Dependence: [28]*** The document which contains letters, words, running text, paragraph used in a computer is hold by its memory in the form of bits. From the bits we have to interpret the characters of natural language where each bit is a character set of writing. There were some limitations in the interpretation. The text that we use in the computer encoded in the ASCII 7 bit character set, in which only 128 characters can be stored in roman alphabet format. This tend to conversion of normal characters into roman is called romanization. These conversions required for the characters which are not defined in the ASCII character set. Languages that are different from English requires much more romanization to permit ASCII processing.

  To extend the encoding of characters, 8-bit encoding is used where it allows to encode 256 unique characters. As there are thousands of characters are remaining in most of the language, overlap the characters in the present system. Most of the words can be encoded in the single bit system but languages like Chinese and Japanese contains a greater number of character set, which required 2 byte system. Each character depicts by a pair of 8-bytes. Since the same range of words and number are represented in a different encoding makes complex for tokenization. Tokenizers objective is performing an action focused on to a particular language with specific encoding.

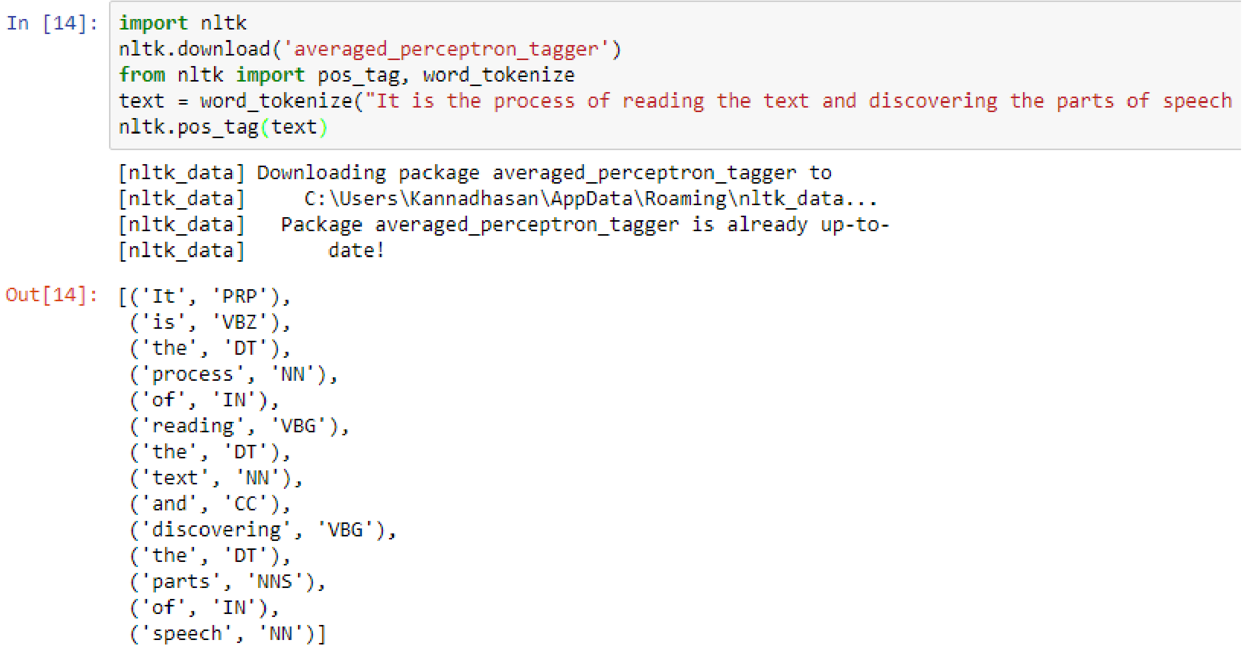
For English or Spanish language uses ISO 8859-1 encoder in which, special characters such as ¡, ¿, £, $ etc., and other symbols are represented in the bytes range 161-191. The encoder TIS 620 is used by Thai language; the same bytes range 161 - 191 represents consonants. For each encoder that we change for language, tokenizer should be able to handle. So, the necessary rules should be added to handle different language for a successful tokenization.

- ***Corpus Dependence:*** Corpora is group of text document written in some language. Due to huge volumes of availability of corpora with variety of languages surrounded by wide range of data types, NLP should perform vigorously to rectify the errors such as misspelling, unwanted spacing, punctuation misplace and irregularities in the sentence. David [28] had mentioned in the corpus dependency section in page 5 that algorithms are not good enough to handle the Input text that comes after set of conventions that are hand written. It is challenging to define the rules for such language. It is also impossible to make people to follow the rules of writing.

  While writing, traditionally defined rules of writing are generally ignored. This one prominent point that should be taken into consideration for the present discussion of sentence splitting because the segmentation mainly depends upon the space between words and sentence, and punctuation. Present algorithm of sentence splitting are depend upon nature of corpora and specific to language are designed to detect and recognize the ambiguities. Text splitting algorithm designed to handle such corpora and it should have an ability to distinguish the irregular documents. [28]

- ***Parts of Speech tagging:*** It is the process of reading the text and discovering the parts of speech (Noun, pronoun, verb, adverb, adjective etc.,) of each word in the given data and tagging them. It is one of the most popular frequently used preprocessing techniques of Natural language processing for text analytics. In Natural Language Processing for Semantic web [1] book, in the 15th page, pos tagging is explained that different categories of words are distinguished according to tenses for words, singular and plural.

There are different tag sets for labelling the POS for words which varies for each system and language. For English PTP (Penn TreeBank) tag set is used.



```
In [14]: import nltk
         nltk.download('averaged_perceptron_tagger')
         from nltk import pos_tag, word_tokenize
         text = word_tokenize("It is the process of reading the text and discovering the parts of speech
         nltk.pos_tag(text)

         [nltk_data] Downloading package averaged_perceptron_tagger to
         [nltk_data]     C:\Users\Kannadhasan\AppData\Roaming\nltk_data...
         [nltk_data]   Package averaged_perceptron_tagger is already up-to-
         [nltk_data]       date!

Out[14]: [('It', 'PRP'),
          ('is', 'VBZ'),
          ('the', 'DT'),
          ('process', 'NN'),
          ('of', 'IN'),
          ('reading', 'VBG'),
          ('the', 'DT'),
          ('text', 'NN'),
          ('and', 'CC'),
          ('discovering', 'VBG'),
          ('the', 'DT'),
          ('parts', 'NNS'),
          ('of', 'IN'),
          ('speech', 'NN')]
```

Figure 8: PTP pos tag representation [Own example]

Figure 8 explains how parts of speech tagged for each word in a sentence. Each tag representation has a meaning.

PRP: Personal Pronoun

VBZ: Verb (3rd person singular)

DT: Determiner

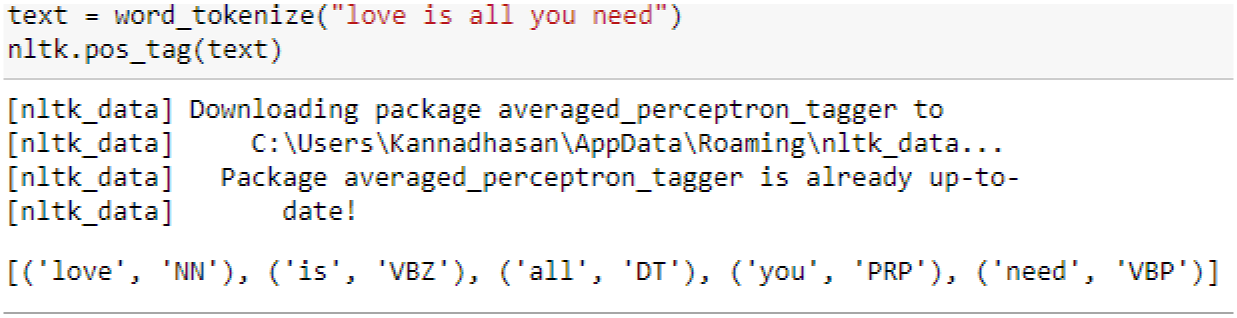
NN: Noun

IN: Preposition

CC: Coordinating Conjunction

NNS: Noun, Plural

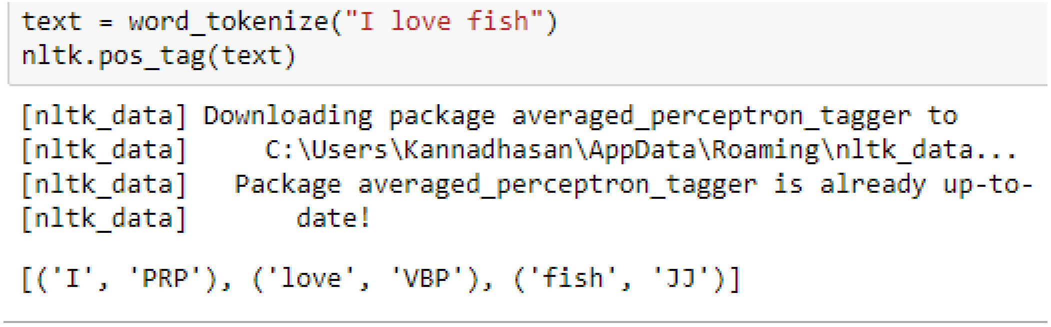
Pos taggers considers entire context of the sentence to tag a word. The reason is t avoid ambiguity of words. Let’s see an example below:



```
text = word_tokenize("love is all you need")
nltk.pos_tag(text)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Kannadhasan\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!

[('love', 'NN'), ('is', 'VBZ'), ('all', 'DT'), ('you', 'PRP'), ('need', 'VBP')]
```

Figure 9: Ambiguity example 1



```
text = word_tokenize("I love fish")
nltk.pos_tag(text)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Kannadhasan\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!

[('I', 'PRP'), ('love', 'VBP'), ('fish', 'JJ')]
```

Figure 10: Ambuigity example 2

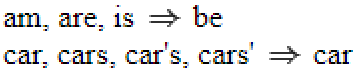
In the figure 9 and 10, we can see how pos taggers avoid ambiguity. In two sentences the word ‘Love’ is common but the usage is different which means a different grammar. This can be correctly identified only when we read the whole sentence. In most of the taggers machine learning algorithms like Markov Models, maximum entropy is used. In most of the NLP toolkits Brill tagger along with PTB tag set. Most of the NLP toolkits has own tagger. Some of the taggers are:

Table 2: POS Taggers [1, page 16, first paragraph]

S. No	**POS Tagger**	**Algorithm/Approach**
1	Brill Transformation	Machine learning
2	Open NLP POS Tagger	Beam Search Algorithm
3	Stanford POS Tagger	Maximum Entrophy
4	TNT (Trigram’s tags)	Viterbi Algorithm

- ***Lemmatization:***

  It is the process of removing the inflections from the opinions and producing the meaningful words without changing grammar based on the wordnet dictionary. The wordnet dictionary lemmatizes the words on its own by using the dictionary without any explicit mention of conversion of words. For example, the word amusement, amuses, amusing will convert it as amuse. So, Lemmatization algorithm identifies lemma for each word. It does the process by understanding the whole context of the sentence with parts of speech tagging for each words and then produces the lemma of those words. [11][12][13]



am, are, is ⇒ be
car, cars, car's, cars' ⇒ car

Figure 11: Lemmatization [10]

In figure 9, if we see the words “am, are, is” and “car, cars, car’s, cars’” has only one meaning, that is, be and car. Here, be and car are leamma for the respective group of similar words. It does morphological analysis in all the words and removes the inflected words and then returns the words which is in the wordnet dictionary. It is an additional text preprocessing for indexing and such process exist in for both commercial and open source analysis[13].

3.2 Machine learning

Shai Shalev-Shwartz and Shai Ben-David Machine [29] have explained that machine learning is discovering the meaning full patterns in the data set. We can notice there is a huge development in this field and this is mostly often used technique to extract any information from a large set of data. We are within the boundaries of this technique every day: smart search engines, preventing frauds in banking, detecting spam in the email etc. Likewise, there are a number of applications that have used this this technique. In this section, machine learning algorithms concepts are explained to provides a basic understanding how it works.

3.2.1 Supervised and Unsupervised learning:

- **Supervised:** It is one of the machine learning algorithms which is used the predict the output for a given dataset. In this algorithm we know what we have to predict from the input. It mainly addresses the classification and regression problems. The basic idea is first we have to train the certain amount of data and we have to fit the model. While training the data it extracts the features from it. To test the fitted model, we have to pass the test data to the algorithm. Using already extracted features from the training set, model runs the algorithm and predicts the output. Now to calculate the precision, we have to test data that are excluded from training set. From this result we can interpret how well the model is performed and we can use metrics to categorize the result of classes. Some of the supervised learning algorithms are Naïve Bayes classifier, Support Vector machine, neural networks, random forest.
- **Unsupervised learning** is contrast to the supervised learning because here we don't have predictors. From the given uncertain data, the task is to identify similar groups by omitting the dissimilar one. Some of the algorithms are clustering, k-Means algorithm, Latent Dirichlet Allocation. The main goals of this algorithm are to find the unusual observations and patterns and to cluster the group of similar observation. For any unstructured data with lots of information they can be exposed to the algorithm. It will learn the data on its own and will provide the results. When the structure of the data frequently varies, then algorithm should also be capable to adapt to such data and also when the volume of data increase, for supervise algorithm it can't learn the data. We have to feed the rules to extract the features whereas in unsupervised, it has the capability to handle it. Supervised learning is suitable only for structured data. When the pattern of data changes frequently, then unsupervised learning is the more appropriate algorithm to deal such data.

3.2.2 Naïve Bayes Classifier:

It is one of the machine learning algorithms that uses Bayes theorem for any classification problems. The main use of this algorithm is sentiment Analysis/Opinion

Mining, medical diagnosis, and spam filters. Bayes theorem is finding the probability of event happening which is connected to other events. For example, probability of getting admission into a college is decided by the time of application, the chosen course, number of other students applied. So, the final aim is it gives conditional probability of an event based upon the knowledge derived from test (other events).

Both event and test are different. If we take a medical test for heart disease, that is not an event but identifying the probability of an event heart disease. From the information we got in a test, we have to identify the false positive rate. Just because the test result is positive that doesn't mean that the patient is suffering from the disease. There are dependent factors that we need to take into consideration. If the test has taken for a rare disease, then the chance false positive rate is higher. This applies for spam detectors in the email as well.

- **Naïve Bayes for text Classification:** It is one of the mostly used algorithm for text classification, sentiment analysis and opinion mining. For sentiment analysis we have a class positive, negative and neutral. Jiangtao Ren , Sau Dan Lee , Xianlu Chen , Ben Kao , Reynold Cheng and David Cheung presented a paper on *Naïve Bayes classification for uncertain data* [30] have explained that based on the estimated density from of a class with prior probability , the posterior class probability of a test data point can be derived and the test data will be assigned to the class with the maximum posterior class probability. Dealing with uncertain data is a problem while density estimation of a class. From a probability distribution of a uncertain data, we should learn the conditional density of a class.

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

The formula is finding the conditional probability for a class A, with feature variable B. Class for sentiment could be Positive, negative and neutral and the variables would be test set data. The variable usually requires a larger number of samples. The model is able to calculate the conditional probability

3.2.3 ***Support Vector Machine:*** Using the knowledge derived in the previous classification, the objects are classified into two groups. It is similar to labelling the sentiment of the opinion by knowing the emotion learned from a previous experience. This type of problem is called binary classification. SVM mainly supports such problems.

- **How it works:** To categorize an object into separate group, SVM create vector space with two dimension. Based on the object property which already learned, it falls into vector space. Now vector space is divided by linear partition. This is called a hyperplane between the two different features. We can also call it as a boundary between those two categories.

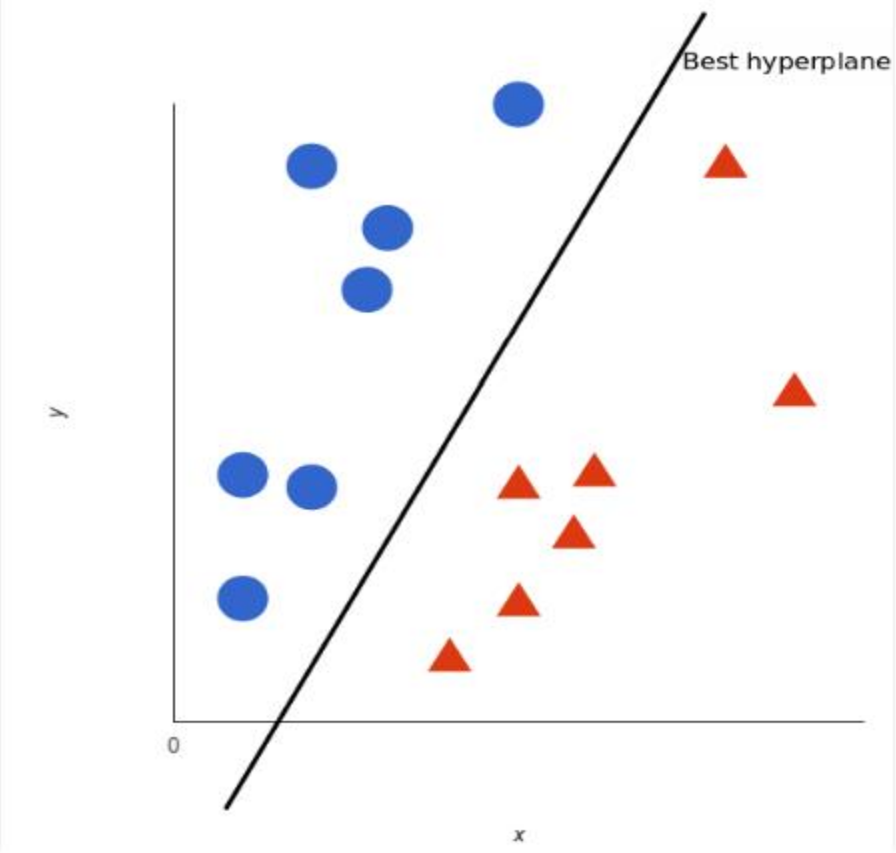


Figure 12: SVM 2D Hyperplane example [31]

Figure 12 clearly illustrates how a hyperplane separates the two features. In x,y coordinates blue and red are two unique properties. In the plot it is separated by a linear hyperplane.

- **Maximum Margin Classifier:** We can have n number of hyperplanes that separate the unique features. The boundary of the hyperplane keeps on changing depending upon the points that are closer to it because the boundary is decided by the distance between the points and hyperplane. Points that are closer to the plane are considered as support vector because it controls the hyperplane. When the data increases, this classifier is able to classify the data precisely.

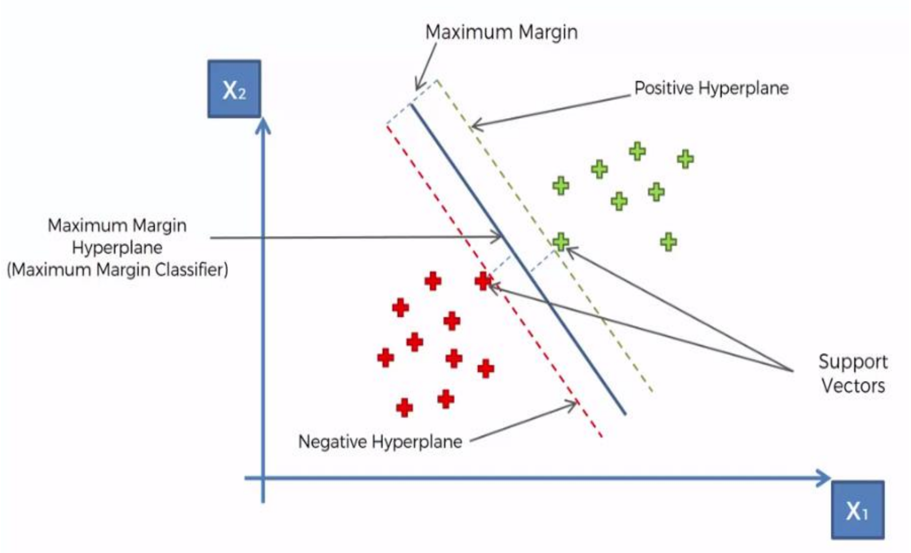


Figure 13: Example of Maximal Margin classifier

- ***Support Vector Classifier:*** Sometimes separating hyperplane might not be possible. There can be misplacement of the data points in the wrong category. It is violation of training data which falls in the opposite side. To some extent model permits this because this parameter helps to minimize the error. There is limitations for having observations on the wrong side of the plan. This is controlled by tuning parameter C. depends on the value of C, the observations can fall on the wrong side. [34]

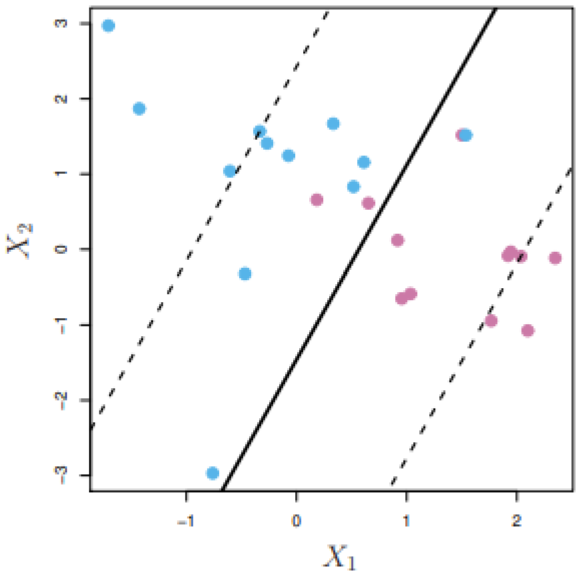


Figure 14: SVM Classifier [34, page no 5]

In figure 14 we can see that red color points are misclassified in blue class. The permitted misclassification is 2. This value of C is derived from cross validation. The observation that lies close the plane are the support vectors. Based on these vectors, some misclassification happens.

- **Support Vector machine:** In support vector classification hyperplane or decision boundary is linear. This limits to do non-linear classification. To overcome such problems Support vector machine is used, in which there are different kernals introduced to decide the linearity.

3.2.4 Deep Learning Neural Network

Neural is network is one of the supervised machine learning algorithm technique mainly used to recognize patterns in the data. The format of data can be audio, text, images, numerical – it can able to recognize all such data. it takes those data in the form of vectors. This algorithm clusters the unlabeled data based on the similarity and also it classifies the labelled data. This is helpful for text classification to extract the features that are given as a input data. [35]

- **Classification:** As described above labelled data can be classified. In order to classify the data we have feed the data as input to the algorithm. This will lead to find the correlation between the labels and data. It can recognize the patterns in the faces, expressions and also any objects in the images. Using this algorithm, we can also translate the speech into text.

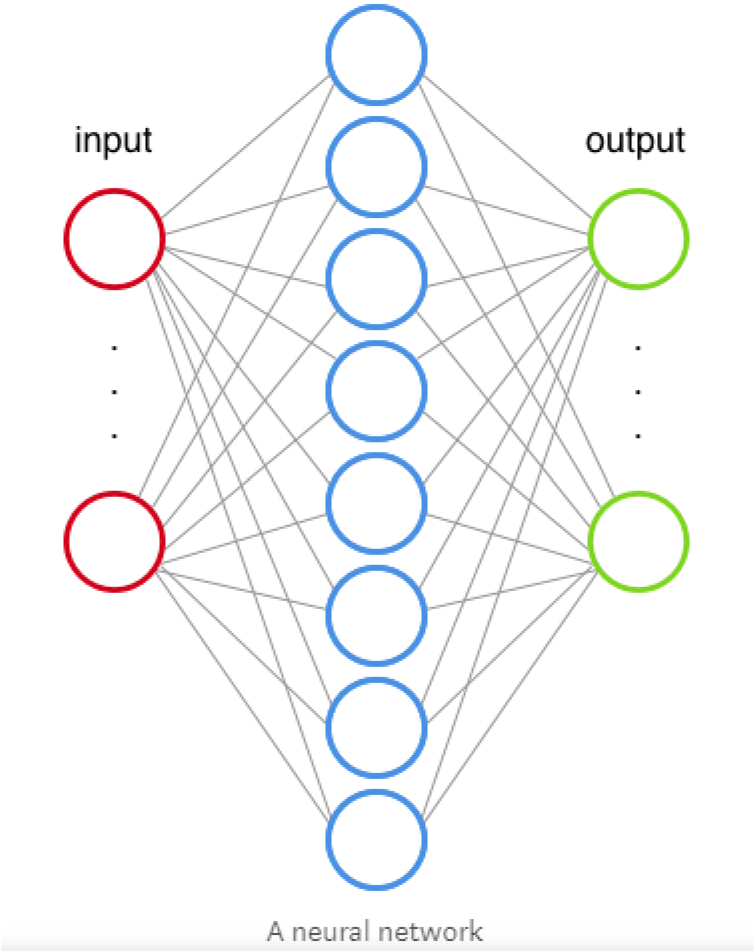


Figure 15: Neural Network [36]

Hidden Layer: The network has input, output and hidden layer. The hidden layer processes the input and provides output. Number of nodes required in the hidden layer needs to be defined. Each node contains weight value. While training the data, network will adjust the values to provide correct output. Nodes also has bias value. Once the input values multiplied by weights and added the values to bias, we can pass the data to the activator. Activator is a function that helps to process the output. [36]

Output Layer: This is the last layer. Number of nodes in this layer is decided by number of classes used in the input layer. Even for the output values weight is multiplied and bias is added but the activation function different from hidden layer. [36]

- ***How neural network learns:*** The data that needs to be learned are provided in input layer which will form the hidden layer and provides output. This flow is called feedforward network. The algorithm woks the same way how brain works. Let's take an example of bowling. If we play the game for a first time, picked up the ball and rolled it down, we would have noticed how fast we bowled and how it crossed the line to hit the goal. Again, while playing next time, we can change the bowling style to target the goal in a better way. We can use the knowledge learned from previous attempt. Neural network also learns in a same way is called backpropagation. It compares the output that network produced and the expected output. Using the differences of the output it modifies and adds weight and gets the expected output. [37]
- **Activator**: It is used to determine the output of neural network. The output values is lies between 0 to 1. It converts the input value in function to an output signal. In that function. For each input it adds weight and applies function to get the output. [47]
    - **Softmax:** This function is used for classification problem where we have multiple classes to get the result.

      $$S(y_i) = \frac{e^{yi}}{\sum_{=1}^{i} e^{y_i}}$$

      For the given input the function returns probability. The sum of the output is always equals to 1.
    - **ReLu:** ReLu is Rectifier Linear unit is one of the most widely used activation in deep learning. This function gets the data from input layer and then it converts the data by adding weights and functions to output layer. If the result is positive then it returns the actual value but if the result is negative, then output is 0. [47]

3.2.5 Topic Modelling

It is process of discovering the relevant topic based on the pattern and occurrences within the document. The document can contain opinions, product reviews, feedbacks, feelings about religion, views on politics e tc., it can be from different source and media. All these are collected together in a document [40]. This modeling is recent popular for text mining and opinion mining. the document for topic modeling will contain mixture of words which will considered as “bag of words”. This model works in a way to group the words that related each other in a group and that forms a topic. For example, if we take the word automobile, then it will consider car, bus, bike, scooter and other automotive. These group of words clustered together finally that considered as a topic. [41] There are many topic models such as Latent Dirichlet Allocation, Latent Semantic Analysis, Probabilistic Latent Semantic Analysis, Correlated Topic Model are used for classification which provided great results for text/opinion mining field.

- ***Latent Dirichlet Allocation:*** There are huge number of collections of documents available in internet in the form of web, blogs, micro blogging, articles and literature are accessed and used for text analytics. The need of automation is increased to analyze, extract and summarize the result. LDA method is one of the topic models provides accurate results especially for large number of documents. This is one of the unsupervised machine learning algorithms widely used in the field of opinion mining and text analytics. It is generative model which tries to replicate the human writing [40].

  The idea is given set of opinions are modelled as a collective of topic together and each topic cooccurrences is calculated by probability distribution defines the chances of appearances in a topic. Set of opinions are considered as bag of words.

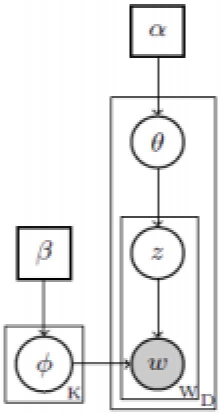


Figure 16: Topic Modelling representation [41, page 3, figure 3]

D – Documents, **K**- Topics, **W** - Word

LDA fit model for a given document D which contains opinions which contains mixture of topics K that explains the distribution of words W. There are models based on LDA: Emotion topic, Role Discovery, Automatic Essay Grading, Anti-Phishing, and Correlated topic model.

Correlated Topic Model is one of the LDA model mainly used for Natural Language Processing. This model extracts the topic from a group of documents. The words are distributed in the logistic normal distribution. Using this distribution, we can create the relationship in the topics. This calculate the frequency of word occurrences within the topic and that plotted as a graph. The limitation of this method is it requires lots of calculation and also we identify the general irrelevant words to omit it from modelling so that those words won't be account for a topic.

Chapter 4

4 Implementation

Analyzing the data and experimenting with it is the pathway to get the solution. In this section I have described about the data set that I have used in this project and what are all the techniques of natural Language processing that were applied, and how the machine learning algorithms were applied to achieve the results.

4.1 Bridge plan Dataset

Kilkenny is one of the oldest towns in Ireland. It is a well-known as a tourist destination because of its medieval heritage. However, as the roads are very narrow throughout the town, particularly during the peak hours of school and offices the city is frequently congested with traffic and it becomes difficult for the people. As a part of infrastructure development in the town, the government introduced a central access scheme which had a plan to construct the bridge to facilitate commuters traveling to the city center and other parts of the town. One of the main aims of the project was to reduce the levels of traffic at peak times. However, the town people were divided about the plan. Although, the plan was believed to address all the existing problems but still some group of people were against it because it affects the natural environment. People who are living in and around Kilkenny have expressed their opinion about the bridge plan proposed by government. The data has three main columns, which is, opinions from people, source of opinion, and on which social media they have posted the opinion. There are totally 620 opinions collected from the people of Kilkenny though social media.

4.2Data Cleaning

Next step after gathering data together is finding the inconsistencies and errors which exist in the form of text in the collected data set. This process is called data cleaning. If any inaccuracy is identified in the data set, then immediately we have to resolve it so that it improves the quality of data. After reviewing the data, I have identified that for some of the opinions source is left empty and I have named it as Anonymous. Since the goal of this project is about finding sentiments, as a part of data processing I have labelled the data. The label is about manually entering the sentiments of each opinions in the data set so that we can consider the data for training, testing and cross validation. Once it is labelled, there are steps to clean the data even more before applying the machine learning algorithm.

4.3Data reading and Initial Validation

As a first step in the implementation I have imported the libraries such as Pandas, numpy, and keras. I have also imported the tools – Tokenizer, pad sequences, and 'to categorical' within which there is the keras library. As the data set is in CSV format I read and converted the data as a data frame object because of its convenience for accessing the data in column and row levels, and additionally it is easier to explore and manipulate the data.

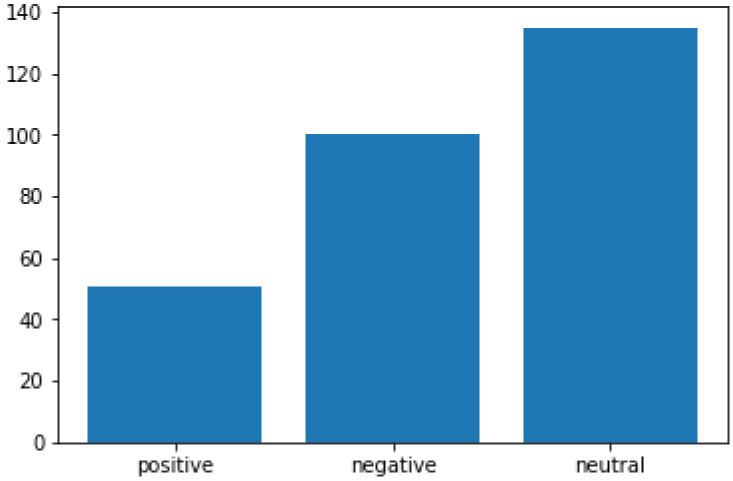
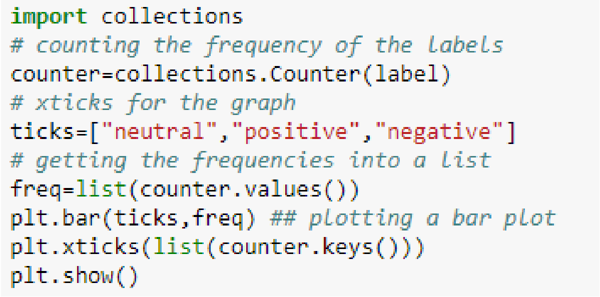


Figure 17: Barplot for frequency of sentiments

The next step was to check for any discrepancies and unavailability of data. I have dropped a row if it is empty. I gathered all the opinion and sentiments into two different columns and each is represented by a different variable name. The Sentiment classes of Neutral, Positive and Negative are classified as 0, 1 and 2 so that it is easy to interpret the results at a class level with the various metrics that we use. Before starting the preprocessing of the opinions, I have calculated the frequency of the three sentiment classes to know which has the maximum frequency.

Figure 14 provides the steps to calculate the frequency using the collection library and figure 15 is barplot of the output. This provides an initial impression of the sentiments underlying the opinions.

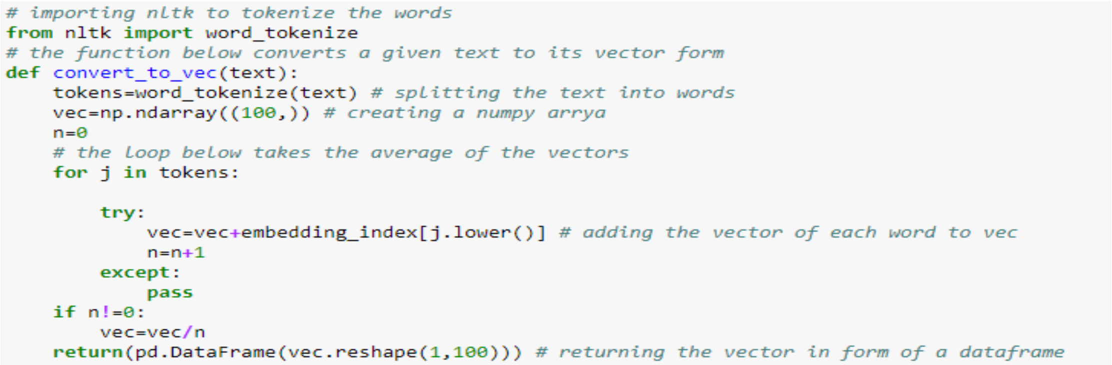


```
import collections
# counting the frequency of the labels
counter=collections.Counter(label)
# xticks for the graph
ticks=["neutral","positive","negative"]
# getting the frequencies into a list
freq=list(counter.values())
plt.bar(ticks,freq) ## plotting a bar plot
plt.xticks(list(counter.keys()))
plt.show()
```

Figure 18: Steps to calculate frequency

4.4Data Preprocessing in Python

Data preprocessing is a mandatory step for any natural language processing. In this stage, we have to modify the data so that it should meet the requirements about the input to the classification algorithm. Since, we are going to use supervised learning algorithms like Naïve bayes, Support Vector Machine and Neural Network, it expects word level opinions as input data and also if there are any irrelevant data, appropriate steps should be taken to omit those words. To do this, first we have to split the opinions into words using a tokenizer.



```
# importing nltk to tokenize the words
from nltk import word_tokenize
# the function below converts a given text to its vector form
def convert_to_vec(text):
    tokens=word_tokenize(text) # splitting the text into words
    vec=np.ndarray((100,)) # creating a numpy arrya
    n=0
    # the loop below takes the average of the vectors
    for j in tokens:

        try:
            vec=vec+embedding_index[j.lower()] # adding the vector of each word to vec
            n=n+1
        except:
            pass
    if n!=0:
        vec=vec/n
    return(pd.DataFrame(vec.reshape(1,100))) # returning the vector in form of a dataframe
```

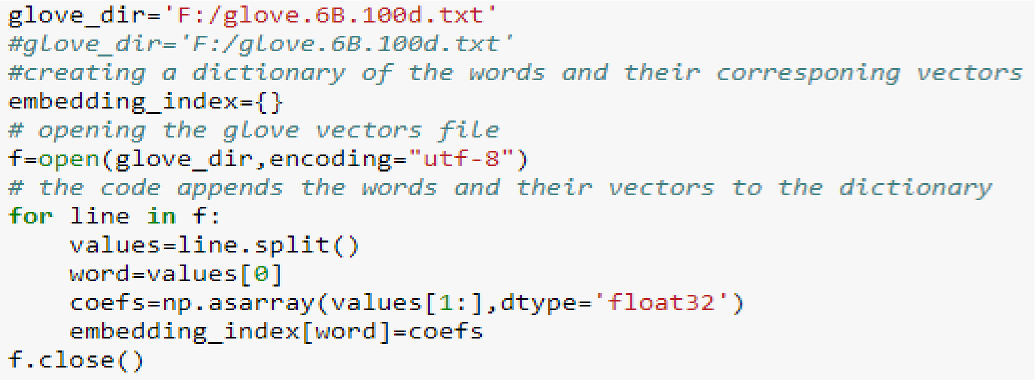
Figure 19: Tokenization implementation

In Figure 16 above, I have imported the NLTK library of python to tokenize the opinions. I have written a function that tokenizes the text and then it converts it into a vector form. So, the first step is to split the sentences with the opinions into words. Then an array is formed and which is used in the 'for' loop to combine the vectors of each word. The vector values are converted into data frame format. This is the scope of tokenization in the project. Vectorization here is to transform the data into a multidimensional array which is of the same data type. This is helpful for indexing operations on the data.

Supervised Learning

4.5 Deep Learning implementation:

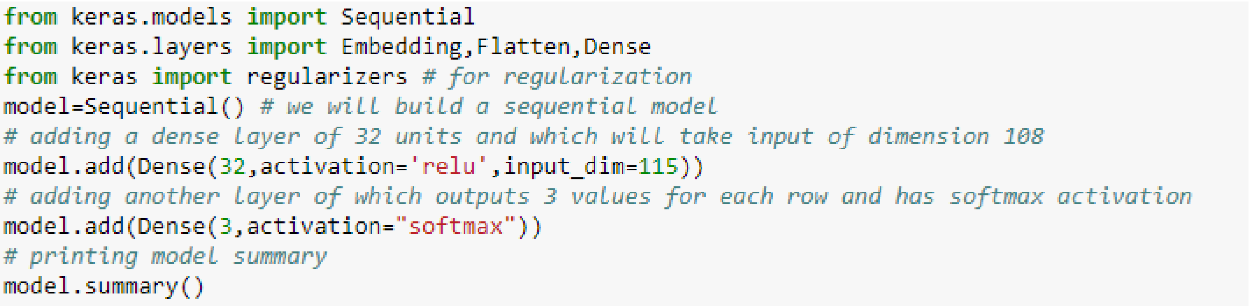
Deep learning is one of the effective machine learning algorithms. It has the ability to learn the given objects on its own by using knowledge of previously learned objects. It is built on neural networks that use input, output and hidden layers.



```
glove_dir='F:/glove.6B.100d.txt'
#glove_dir='F:/glove.6B.100d.txt'
#creating a dictionary of the words and their corresponing vectors
embedding_index={}
# opening the glove vectors file
f=open(glove_dir,encoding="utf-8")
# the code appends the words and their vectors to the dictionary
for line in f:
    values=line.split()
    word=values[0]
    coefs=np.asarray(values[1:],dtype='float32')
    embedding_index[word]=coefs
f.close()
```

Figure 20: Dictionary Creation

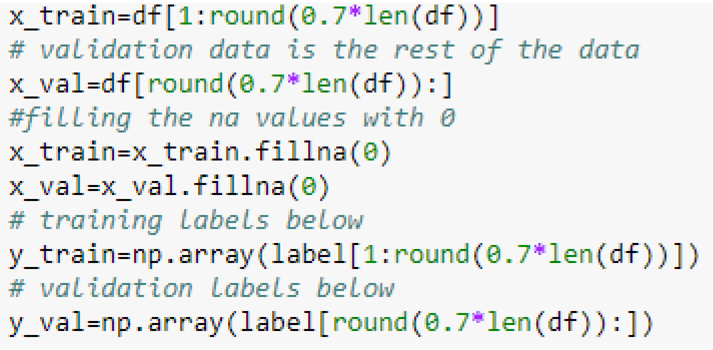
- **Glove [Figure 17]:** Glove is one type of machine learning algorithm to create a vector for each word in the corpus. For generating the vectors in python, there is already pre-trained document which contains vectors of all words. Basically, the task of glove is to find the probability of co-occurrence of a word. As a result, it forms the matrix depends upon the number of occurrences within the corpus. Whenever we use a new data or bunch of corpora together, creating a vector is computationally expensive. If we use already trained data, iterations are faster. So, I have used glove file to add vectors for each word in the corpus and created dictionary of words. The dictionary contains words and corresponding vectors.



```
from keras.models import Sequential
from keras.layers import Embedding,Flatten,Dense
from keras import regularizers # for regularization
model=Sequential() # we will build a sequential model
# adding a dense layer of 32 units and which will take input of dimension 108
model.add(Dense(32,activation='relu',input_dim=115))
# adding another layer of which outputs 3 values for each row and has softmax activation
model.add(Dense(3,activation="softmax"))
# printing model summary
model.summary()
```

Figure 21: Sequential Model

- **Creating the sequential model**: Using Keras library [Figure 18], I have used the sequential model for auto numbering of words to check its presence by matching with word dictionary that we created. It will categorize the words by numbers based on the match and un match. This happens at the backend of the model.
- **Adding Layers:** Once sequential model is built, then we have to create different layers of neural network such as Input, hidden and output layer. All the three layers are interconnected layers with artificial neurons. I have created input and output dense layers:
  - Input layers has 115 inputs with 32 neurons in the hidden layer with relu activation [Refer section 3.2.4 for ReLu and SoftMax activator]
  - Output layers has 3 classes with SoftMax activation.



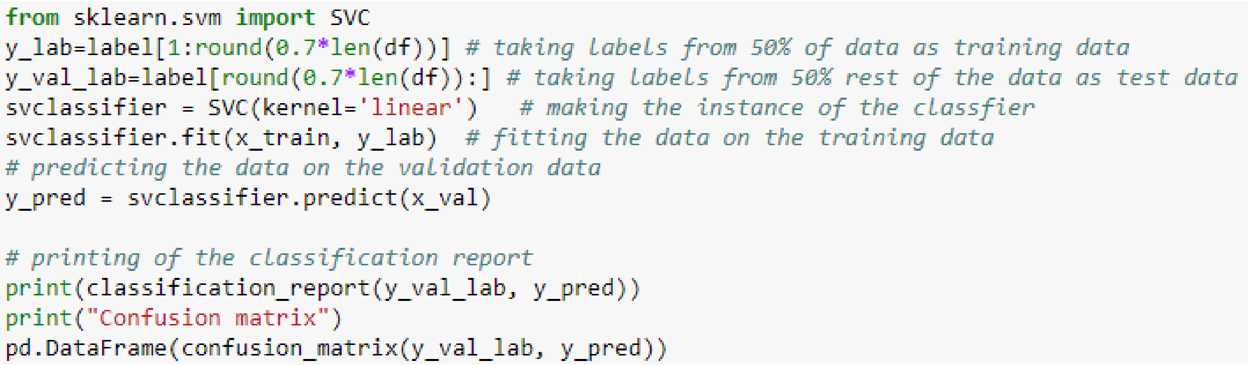
```
x_train=df[1:round(0.7*len(df))]
# validation data is the rest of the data
x_val=df[round(0.7*len(df)):]
#filling the na values with 0
x_train=x_train.fillna(0)
x_val=x_val.fillna(0)
# training labels below
y_train=np.array(label[1:round(0.7*len(df))])
# validation labels below
y_val=np.array(label[round(0.7*len(df)):])
```

Figure 22: Test and Train data Split

- **Testing and Training the model [Figure 19]:** This is an important step to get the expected output for the given input data. We have to split the data for training and test set. First training set is used to fit the model. From this, model will learn about the opinions that we have used. We already have data in the form of vector in the dataframe format which we did at the initial process. I took 70% of the data as a training set to learn the model. The remaining 30% is used as the test set to calculate the metrics of the sentiment classes that we have considered at the beginning. I have tested the model using the Keras neural network and calculated the metrics precision, F1 score, and recall for sentiment classes positive, negative and neutral.

4.6 SVM Implementation:

As I have explained in the section 3.2.3, SVM is supervised learning algorithm used for classification problems. The aim of this algorithm is finding the best hyperplane among all the plane based on the distance of vectors close the plane. SVM kernel has important role in transformation of data. If the data can be linearly separable, then we can use a linear kernel. Since we are dealing with text classification, the size of the corpus and features will be large. It is able to convert small dimension space into larger dimensional space depends upon the number of features. So, for such data the Linear kernel is also performing faster than other kernels.



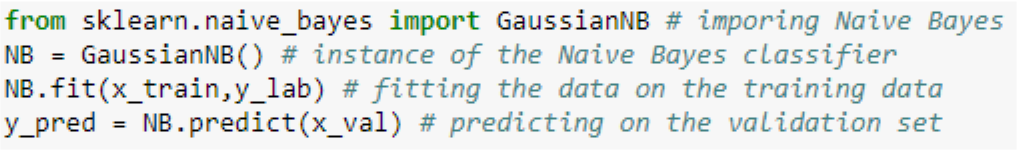
```
from sklearn.svm import SVC
y_lab=label[1:round(0.7*len(df))] # taking labels from 50% of data as training data
y_val_lab=label[round(0.7*len(df)):] # taking labels from 50% rest of the data as test data
svclassifier = SVC(kernel='linear')   # making the instance of the classfier
svclassifier.fit(x_train, y_lab)  # fitting the data on the training data
# predicting the data on the validation data
y_pred = svclassifier.predict(x_val)

# printing of the classification report
print(classification_report(y_val_lab, y_pred))
print("Confusion matrix")
pd.DataFrame(confusion_matrix(y_val_lab, y_pred))
```

Figure 23: SVM Algorithm

As described in Section 4.5 I have split the data to train and test set. I fitted the SVM model using the training data and predicted the sentiments for different metrics using the test set of data.

4.7 Naïve Bayes Implementation

This model is about finding the conditional probability of an event based on the other relational events. This algorithm is mostly used for sentiment analysis.



```
from sklearn.naive_bayes import GaussianNB # imporing Naive Bayes
NB = GaussianNB() # instance of the Naive Bayes classifier
NB.fit(x_train,y_lab) # fitting the data on the training data
y_pred = NB.predict(x_val) # predicting on the validation set
```

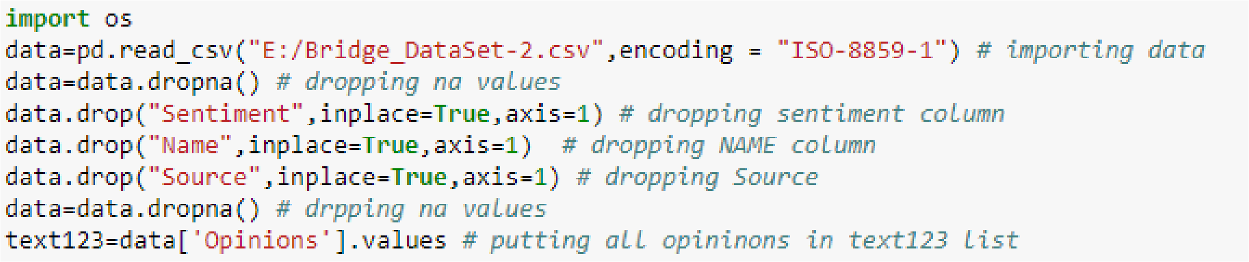
Figure 24: Naive Bayes Algorithm

I have created an instance of Naïve Bayes using Gaussian which I have imported using SKlearn library. I fitted the model using the training data. After training the model, I predicted the probability of the sentiments, that is how much percentage of the labelled sentiments are true.

Unsupervised Learning

4.8 Topic Modelling Implementation:

The idea of this modelling is to identify the discussion of relevant meaningful topics in the opinion. I automated the algorithm that can examine the whole opinions and produce the topics discussed. In this model to extract the topic I have used Latent Drichlet Allocation method with the Gensim package. This is an unsupervised machine learning technique. Here we don't use any labelled data like we used in the supervised learning algorithms.

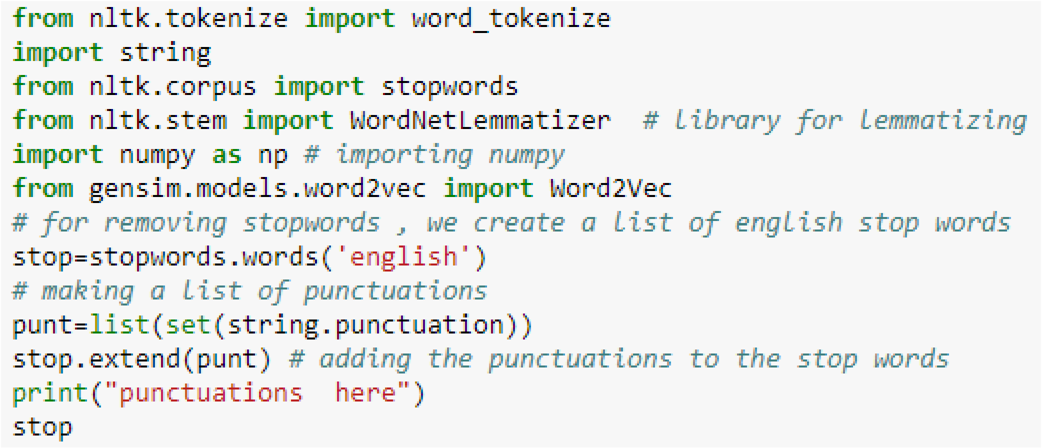


```
import os
data=pd.read_csv("E:/Bridge_DataSet-2.csv",encoding = "ISO-8859-1") # importing data
data=data.dropna() # dropping na values
data.drop("Sentiment",inplace=True,axis=1) # dropping sentiment column
data.drop("Name",inplace=True,axis=1)  # dropping NAME column
data.drop("Source",inplace=True,axis=1) # dropping Source
data=data.dropna() # drpping na values
text123=data['Opinions'].values # putting all opininons in text123 list
```

Figure 25: Data loading for Topic

As a first step of implementation I have loaded the data using the pandas library. Since the analysis is about finding the topic in the opinions, I have dropped other columns of data which are irrelevant for this model. I took the opinions alone in a variable.

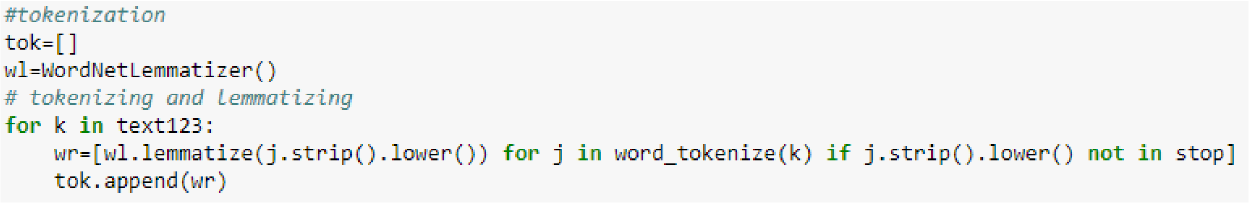
- ***Lemmatizing and Tokenizing:***



```
from nltk.tokenize import word_tokenize
import string
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer  # library for lemmatizing
import numpy as np # importing numpy
from gensim.models.word2vec import Word2Vec
# for removing stopwords , we create a list of english stop words
stop=stopwords.words('english')
# making a list of punctuations
punt=list(set(string.punctuation))
stop.extend(punt) # adding the punctuations to the stop words
print("punctuations  here")
stop
```

Figure 26: Stopwords

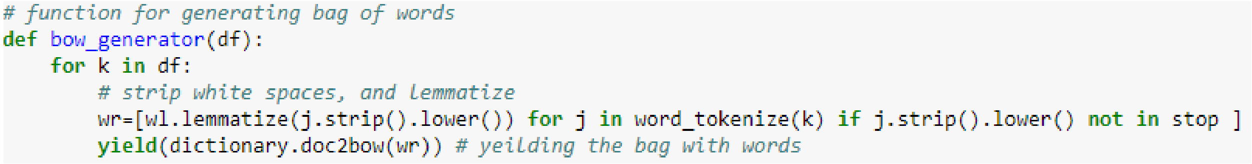
From Figure 23 we can see that I have downloaded and imported libraries NLTK, String, numpy to perform text preprocessing for topic modelling. To extract the relevant topics from the opinion, we have to remove the irrelevant data like prepositions, conjunctions, interjections. Stopwords contains those list words to remove from the opinion data. Additionally, we can also include more words in the stopwords as it can be rewritable.



```
#tokenization
tok=[]
wl=WordNetLemmatizer()
# tokenizing and lemmatizing
for k in text123:
    wr=[wl.lemmatize(j.strip().lower()) for j in word_tokenize(k) if j.strip().lower() not in stop]
    tok.append(wr)
```

Figure 27: Tokenization and Lemmatization

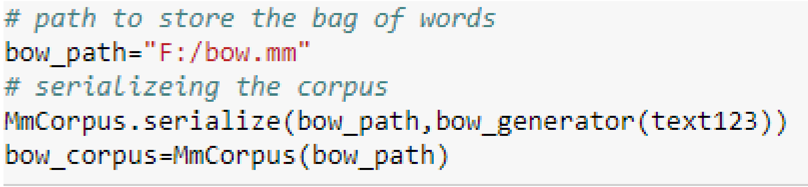
After excluding the stop words from the opinion [Figure 24], I have split the sentence into words using tokenizer and simultaneously I have lemmatized to get the lemma for most of the common words with the same meaning.



```
# function for generating bag of words
def bow_generator(df):
    for k in df:
        # strip white spaces, and lemmatize
        wr=[wl.lemmatize(j.strip().lower()) for j in word_tokenize(k) if j.strip().lower() not in stop ]
        yield(dictionary.doc2bow(wr)) # yeilding the bag with words
```

Figure 28: Generating bag of words

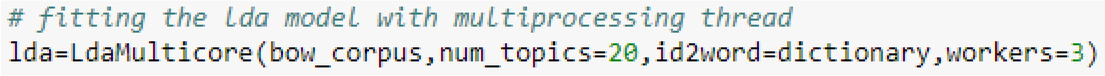
Next step is generating the bag of words [Figure 25]. After tokenizing and lemmatizing, there will be list of relevant opinion data in a variable. Those group of words I have yielded and formed a dictionary of words.



```
# path to store the bag of words
bow_path="F:/bow.mm"
# serializeing the corpus
MmCorpus.serialize(bow_path,bow_generator(text123))
bow_corpus=MmCorpus(bow_path)
```

Figure 29: Serializing

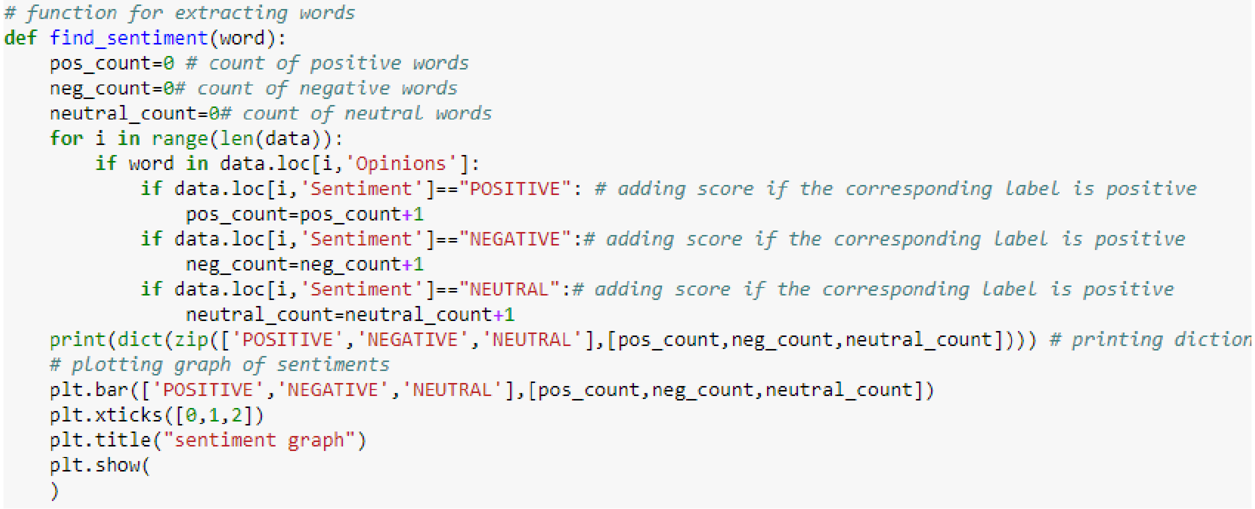
Those group of words [Figure 26] saved in a document in a local system. Each word is serialized so that now it will have an unique number. This is one the required input of LDA model.



```
# fitting the lda model with multiprocessing thread
lda=LdaMulticore(bow_corpus,num_topics=20,id2word=dictionary,workers=3)
```

Figure 30: Fitting the model

I have fitted the model [Figure27] with bag of words that was generated, and also mentioned about the number of required topics as discussed.



```
# function for extracting words
def find_sentiment(word):
    pos_count=0 # count of positive words
    neg_count=0# count of negative words
    neutral_count=0# count of neutral words
    for i in range(len(data)):
        if word in data.loc[i,'Opinions']:
            if data.loc[i,'Sentiment']=="POSITIVE": # adding score if the corresponding label is positive
                pos_count=pos_count+1
            if data.loc[i,'Sentiment']=="NEGATIVE":# adding score if the corresponding label is positive
                neg_count=neg_count+1
            if data.loc[i,'Sentiment']=="NEUTRAL":# adding score if the corresponding label is positive
                neutral_count=neutral_count+1
    print(dict(zip(['POSITIVE','NEGATIVE','NEUTRAL'],[pos_count,neg_count,neutral_count]))) # printing diction
    # plotting graph of sentiments
    plt.bar(['POSITIVE','NEGATIVE','NEUTRAL'],[pos_count,neg_count,neutral_count])
    plt.xticks([0,1,2])
    plt.title("sentiment graph")
    plt.show(
    )
```

Figure 31: Sentiments for topic

Figure 31 shows the automated function developed to find the sentiment of extracted topic in the model. It identifies the frequency of co-occurrences in the opinion document and provides overall sentiment of the topic.

Chapter 5

5 Analysis and Solution

In this chapter I have discussed about the solutions acquired from the machine learning algorithm. First, I explain about the metrics I have used to measure the results. Then, I explain about the error computation for each algorithm.

5.1 Metrics [33]:

I have used following metrics and terms in supervised machine learning algorithm to measure the results:

- ***True Positive:*** This refers to truly predicted positive value. There is an actual class and predicted values. When an actual class indicates something, predicted class should also imply the same thing.
- ***True Negative:*** This refers to truly predicted negative value. Here, both the classes are no. That is, when an actual class value is no then predicted value should also be no.
- ***False Positive:*** This refers to the predicted result is positive but actually it is negative.
- ***False Negative:*** This indicates the actual class value is positive, but the predicted class is negative.
- ***Classes:*** Actual class provides output from the real set of data which we fit before prediction and predicted results provided by machine learning algorithm. The output values are comparison of both predicted and actual values.
- ***Precision:*** It is Percentage of truly predicted positive value to the total predicted positive values. The question is out of all the opinions that are marked, how much percentage is correct? Precision provides answer to this question. [33]

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

- ***Recall:*** *It is ratio of truly predicted positive observation to all the observations that are actually positive. [33]*

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

- ***F1 Score:*** This is average of both precision and recall. This considers both false positives and false negatives. The score of f1 is considered best if the values is close to 1 and it is worst if the value is close to 0.[33]

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

- ***Support:*** It is number of actual occurrences of the class in specified data set.

5.2Results

The following results are predicted values of machine learning algorithm.

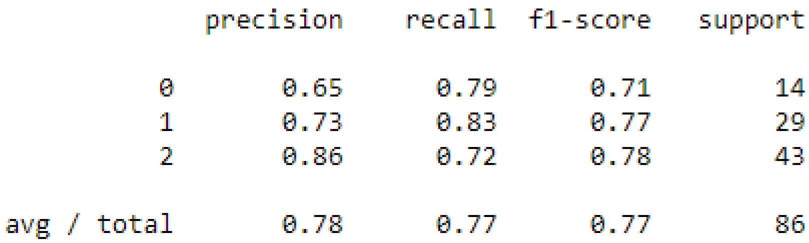
Sentiment Classes:

Neutral: 0

Positive: 1

Negative: 2

- **Deep Learning:**



```
             precision    recall  f1-score   support

          0       0.65      0.79      0.71        14
          1       0.73      0.83      0.77        29
          2       0.86      0.72      0.78        43

avg / total       0.78      0.77      0.77        86
```

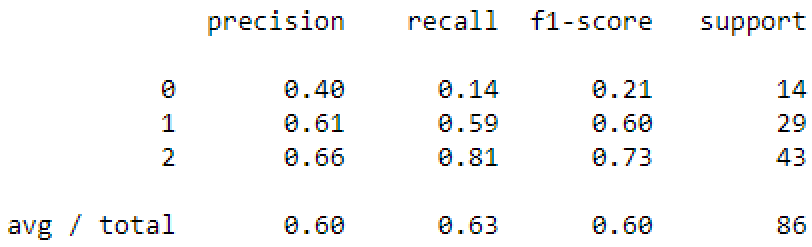
Figure 32: Deep Learning results

From figure 29 we can see the precision score which we have acquired using the deep learning neural network algorithm. Overall the precision of predicted value is 78 % accurate. For all the instances that was classified are positive sentiment, 73% is correct. Similarly, for the negative sentiment 86 % is correct and for the neutral sentiment 65% is correct.

Recall: For all the instances that were classified, 77 % were classified correctly. If we look at a class level, for all the instances that were actually positive 83% was classified correctly, and for negative 72% was classified correctly.

The overall percentage of f1 is 77% which means it is close to the best score 1 that is 100%. This is a weighted average of both precision and recall. Since the score is near to the best score, we can consider that this model prediction rate is significant.

- **Support Vector Machine:**



```
             precision    recall  f1-score   support

          0       0.40      0.14      0.21        14
          1       0.61      0.59      0.60        29
          2       0.66      0.81      0.73        43

avg / total       0.60      0.63      0.60        86
```

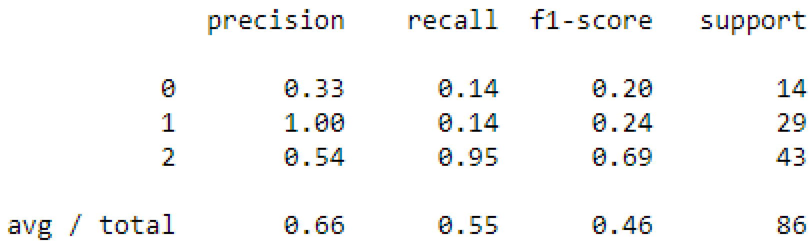
Figure 33: SVM Results

From figure 30 we can see that overall precision score for all the classes is 60 %. For all the instances of classification, 60% was correct. For all the instances classified as neutral, only 40% was correct, and for positive it is 61%, and for negative it is 66%.

If we look at the overall recall score, the results are not bad. For all the instances, 63% was correctly classified. For all the instances that were actually positive, 59% was classified correctly. For neutral only 14% was classified correctly. For the negative sentiment the score is quite high and good because 81% is classified correctly.

Overall the f1 score is 6 % which is not bad but still improvement is required.

- **Naïve Bayes:**



```
             precision    recall  f1-score   support

          0       0.33      0.14      0.20        14
          1       1.00      0.14      0.24        29
          2       0.54      0.95      0.69        43

avg / total       0.66      0.55      0.46        86
```

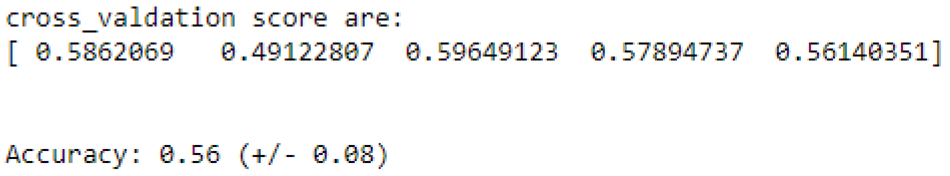
Figure 34: Naive Bayes Results

From figure 31 we can see the result of Naïve Bayes. Overall the precision rate is 66 percent, but the recall score is quite less than the precision. For all the instance, 55%was correctly classified. The f1 score is much smaller at 46%.

- **Model comparison:** If we compare the three models with f1 score, deep learning has performed well with 77%. The rule of thumb is that the model which has the score closest to 1, that is, 100 percent is the best model.

- ***5.3 Cross Validation***: [55]

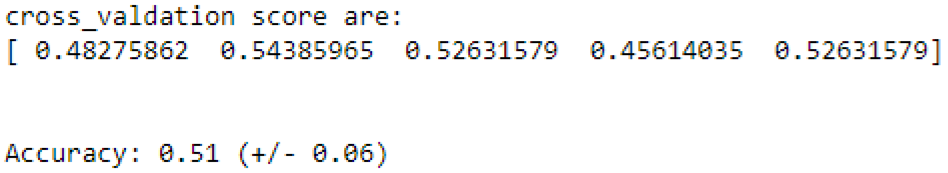
This is the process of splitting the training data into different sets to find out the best sample of data for model prediction. Using this validation, we can identify the data with a greater number of features. For each sample, an accuracy is predicted and from this we can say which sample performs best among the split. Splitting happens randomly.



```
cross_valdation score are:
[ 0.5862069   0.49122807  0.59649123  0.57894737  0.56140351]


Accuracy: 0.56 (+/- 0.08)
```

Figure 35: SVM Cross Validation

From Figure 32 cross validation result it is found out that SVM model can predict the result with 56% accuracy. The score actually is quite less because we have less data samples for model prediction.



```
cross_valdation score are:
[ 0.48275862  0.54385965  0.52631579  0.45614035  0.52631579]


Accuracy: 0.51 (+/- 0.06)
```

Figure 36:Naive Bayes Cross Validation

For Naïve Bayes it is shown that 51% the model can accurately predict the result.

If we compare these results with those for both of the models without cross validation it is obvious that without cross validation the results are better. The variation between the testing and cross validation results is because of a smaller number of sampling.

5.4 Topic Modelling

The Topic modelling algorithm using the Latent Drichlet Method extracted the top three topics discussed in the opinions. The number of topics can be chosen depending upon the requirement of the user. Figure 34 and figure 35 explains about the two different topics extracted. It shows the people's opinion about the bridge plan. Topic one has many subtopics that has a relation with each other. All those topics are clustered together and plotted.

LDA graph Visualization:

There is a pair of histograms on the right side that visualizes about actual and estimated frequency. The frequency of specific topic features overall is represented in a blue color. The red color implies that how many occurrences happened within the document related to that particular topic. The distance between the center of the circle indicates similarity between topics. [48]

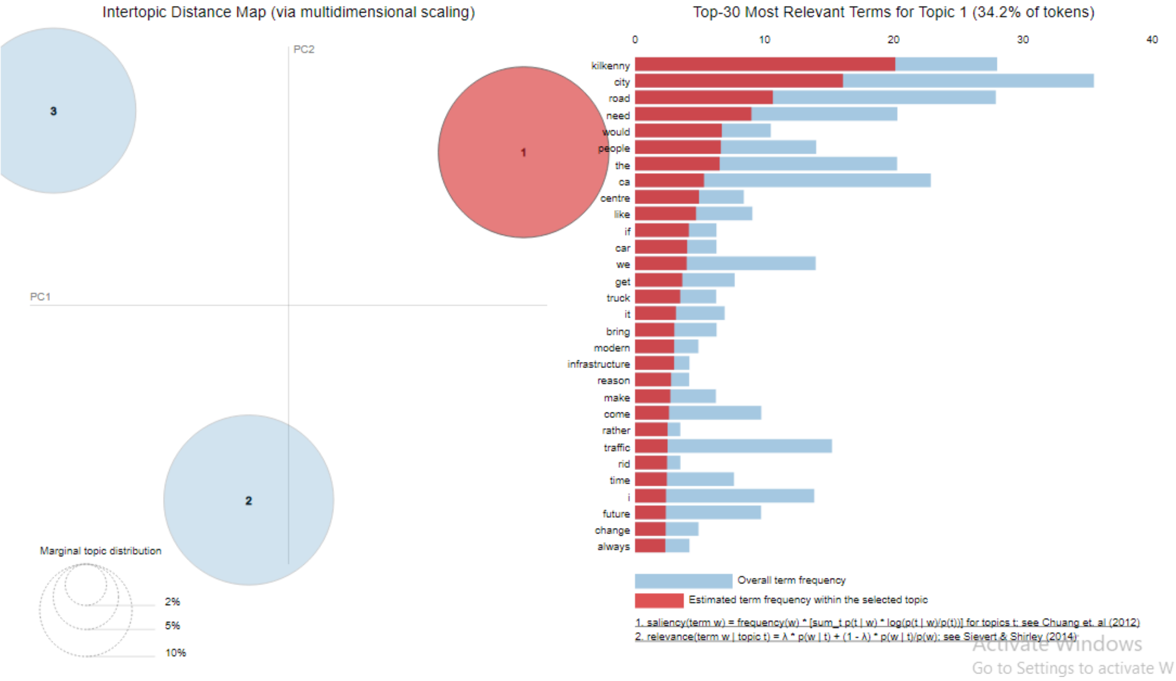


Figure 37: Topic modelling 1

Figure 34 provides information about the first topic. If we look the first topic, people have expressed about the topic featured are Killkenny road, needs, people, modern, infrastructure, traffic, rid, and future. From this result we can interpret that people have conveyed that this plan provides needed infrastructure and also it provides solution to the current traffic problems. So, it is helpful to get rid of traffic. This plan will also for future development.

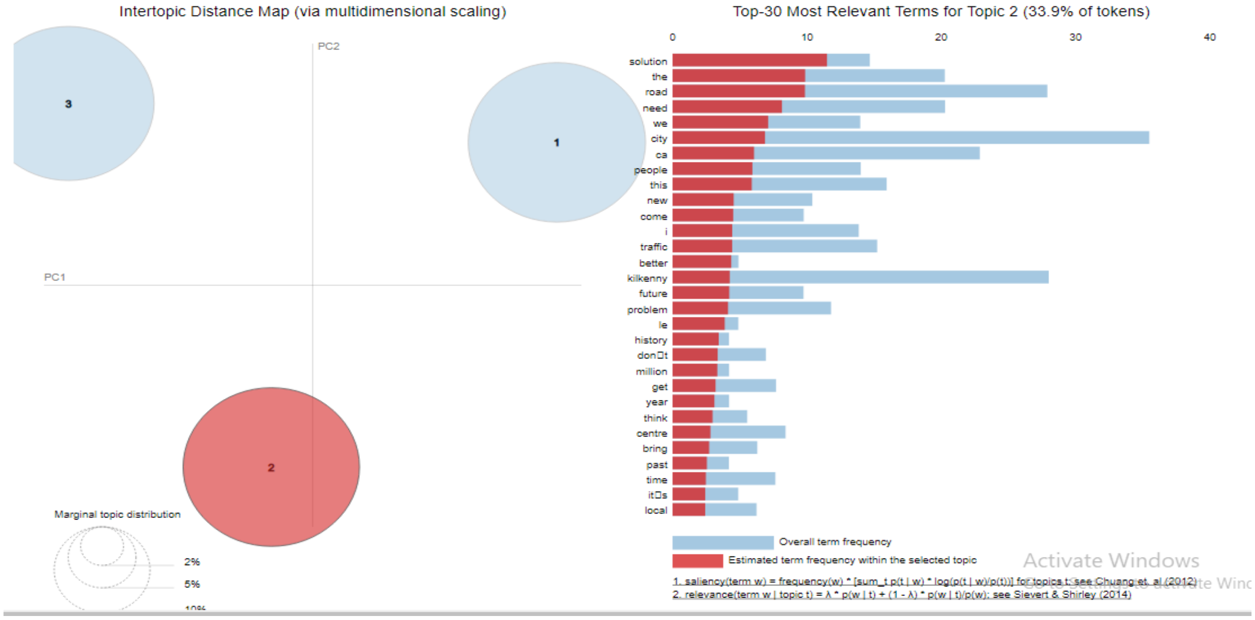
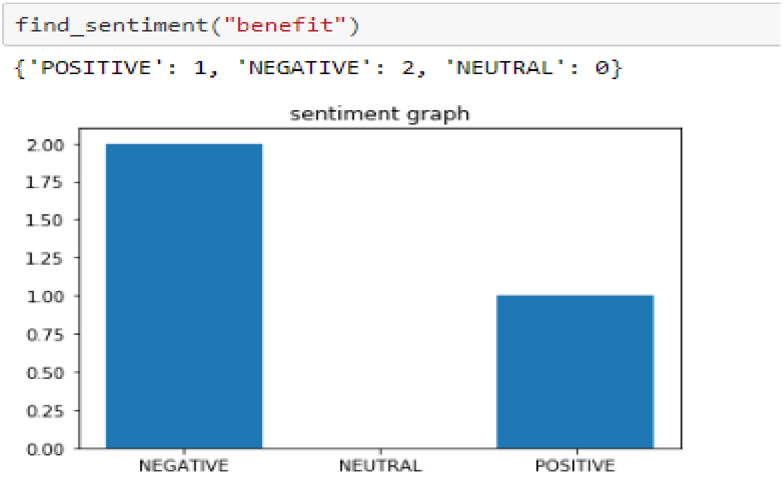

Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
3
1
2
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 2 (33.9% of tokens)
0
10
20
30
40
solution
the
road
need
we
city
ca
people
this
new
come
i
traffic
better
kilkenny
future
problem
le
history
don□t
million
get
year
think
centre
bring
past
time
it□s
local
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 38: Topic Modelling 2

Similarly, figure 35 provides the result of topic 2. In this we can see the topic features of city, people, history, future, and problem. Here they represent features about Kilkenny which is a medieval city and also a tourist attraction so since it affects the tourism people do not like the plan. Each topic conveys the people's opinion on this bridge plan.

Sentiment for topic: I have also developed a function to predict sentiments on each topic that people have discussed. It will provide a measure of the overall sentiment of the topic within each sentiment class.



```
find_sentiment("benefit")
{'POSITIVE': 1, 'NEGATIVE': 2, 'NEUTRAL': 0}
```

Figure 39: Sentiment for topic

Figure 36 is a sentiment prediction for a topic benefit. It provides overall sentiment- positive, negative and neutral frequencies. For the word benefit, the occurrences are mostly in the negative and very less than the positive sentiment. Thus, we can conclude that people’s opinion on the plan is that it will not benefit them.

Chapter 6

In this chapter I summarized about over all analysis and development done to this project. I have also explained about challenges faced on each progression, and for future work I have described about how can we enhance the project to proceed further, what could have done for better results and how to improve it.

6.1 Conclusion

Social media has become very popular for sharing opinions on any topics. Social sites like Facebook, twitter, YouTube permits users to create and share the information in the internet. Nowadays people are shopping in the digital market. They share or express their feelings on the purchased product; they also share views on politics, religion and many other topics. They can also share their opinion about any development plan of government bodies. The opinion can be either supporting the plan or against the plan or it can be feelings about other aspects. If the opinions are in form of text or in several documents but have a huge volume, then manually identifying the feelings and extracting the topic discussed in the opinions is a challenging task. In this thesis, opinions gathered for a proposed Kilkenny bridge plan by a central access scheme were analyzed. A topic-based sentiment analysis on the collected data was done.

From the result we found that in supervised learning algorithms, deep learning achieved high accuracy of 77 percent. There are differences in the accuracy result of Naïve Bayes and SVM when compared to deep learning. Initially, when the train and test split percentage was equal, result weren’t good. After increasing the percentage of train set to 70 percent and test set to 30 percent results were improved a lot, which means increasing the training set can improve the model as well. Overall negative classifiers accuracy is quite good for all the models. So, we can derive that though there are few positives, mostly negative sentiment is associated in the opinions expressed about the Kilkenny bridge plan. The automated topic modelling algorithm also performed well. It found relationship from the features and clustered those topics together in a group and formed a topic. Sentiment analysis also were able to predict for any given topic. This topic-based sentiment analysis approach for a civil domain is a new approach and it performed efficiently. Though expected results are achieved, more work could have done to improve the performance.

6.2 Summary:

I have summarized about the This thesis consists of four stages. They are:

Data Preprocessing: This is the first stage of development. In this stage I have used Natural Language Processing technique to preprocess the data gathered from social media. I have performed all the necessary text preprocessing.

Sentiment Analysis: Once data is cleaned I have classified the sentiments based on the feeling associated in it- this is manually labelling each opinion. Classified sentiments are Positive, negative and neutral.

Prediction: Using the labeled data predicted the sentiments using machine learning algorithms – Deep learning, Naïve Bayes and Support Vector Machine. Of all the algorithm, deep learning model is performed well with f1 score of 77 percent. The other two models perform fairly.

Topic based Sentiment: In this stage I have extracted the topics discussed in the opinion using Latent Dirichlet allocation method. For the extracted topic, finding overall sentiment.

6.3 Future Work

Though the targeted analysis performed for this thesis, still much more work could have been done to improve the result but due to time constraint those works are left for the future enhancement. I have listed out those works:

- One of the constraints that we had is very less data so while sentiment classification for each sentiment, data is not balanced. Because of that there were inconstancies when rerunning the data. The reason is while train and test data split, there are chances are one category of sentiment data may fall into the training set which makes no data in the test set. While predicting the model, the result can be 0. This can be avoided if we have more data. So, first step to improve the present model is more adding data to the existing opinions.

- In the text preprocessing we can also do lemmatization for sentiment prediction which will help to extract more features with high performance. Now we did tokenization, stop words but still the performance would have better if the text is preprocessed with more tools.
- In supervised learning I have used three algorithms to predict the sentiments with various metrics. Apart from deep learning, other models did not perform well. If we could have predicted with other supervised algorithms like Random Forest, decision trees, then we can compare the results and will able to identify best performed model. There are chances that other models can perform better than the models that we used.
- For cross validation we can use k-Fold method on all the algorithms with vectors and unigrams, bigrams and tfiDF. This is a different approach of fitting the model. Since the model couldn’t extract the best features, these types of approach can able to predict the result with more accuracy.

7 References

1. Diana Maynard, Kalina Bontcheva, and Isabelle Augenstein. '*Natural Language Processing for Semantic Web'*, Book revised in the year 2016
2. Bing Liu (2015). '*Sentiment Analysis: mining opinions, sentiments and emotions'* , University of Illinois, Chicago.
3. R. Kibble (2013), '*Introduction to Natural Language Processing*', Goldsmiths, University of London.
4. Bo Pang, Lillian Lee. *Opinion Mining and Sentiment Analysis, Yahoo Research, 701 First Ave, Sunnyvale, CA 94089,* Computer Science Department, Cornell University, Ithaca.
5. David Osimo, Francesco Mureddu. '*Research Challenge on Opinion Mining and Sentiment Analysis'.*
6. Nidhi Mishra, C.K. Jha, PhD. Classification of Opinion Mining Technique, *International Journal of Computer Applications (0975 – 8887) Volume 56– No.13, October 2012.*
7. Sunidhi Dwivedi, Shama Parveen. *Document Level Opinion Mining of Reviews of Mobile Phone Companies.* International Journal of Innovative Research in Science, Engineering and Technology (June 2016).
8. Nikos Engonopoulos, Angeliki Lazaridou, Georgios Paliouras, Konstantinos Chandrinos. '*A Word-Level Method for Entity-Level Sentiment Analysis'.*
9. Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng. '*Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey*'
10. Sentiment Analysis: Concept, Analysis and Applications, Shashank Gupta
https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17
11. Wikipedia: Sentimental Analysis
https://en.wikipedia.org/wiki/Sentiment_analysis
12. 7 Benefits of Sentiment Analysis You Can't Overlook, Shiho Hashimoto
https://blog.insightsatlas.com/7-benefits-of-sentiment-analysis-you-cant-overlook

13. Research from Statista website about social media users
https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/
14. Sentiment Analysis on Twitter using Apache Spark, Amandeep Kaur, Deepesh Khaneja, Khushboo Vyas, Ranjit Singh Saini, Carleton University 2016
15. Times of India article about internet user
https://timesofindia.indiatimes.com/business/india-business/number-indian-internet-users-will-reach-500-million-by-june-2018-iamai-says/articleshow/62998642.cms
16. Number of social network users from Statista
https://www.statista.com/statistics/278407/number-of-social-network-users-in-india/
17. Stopwords in NLP
https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html#fig:stoplist
18. Lemmatization
https://en.wikipedia.org/wiki/Lemmatisation
19. Stemming and Lemmatization, Daniel Tunkelang
https://queryunderstanding.com/stemming-and-lemmatization-6c086742fe45
20. Stanford online book for stemming and lemmatization
https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
21. Opinion Mining and sentiment analysis
https://www.coursera.org/lecture/text-mining/5-5-opinion-mining-and-sentiment-analysis-motivation-o93Yl
22. Brief history of NLP
https://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html
23. What is Python.
https://www.python.org/doc/essays/blurb/
24. Why is python right programming language for data science?
https://datafloq.com/read/why-python-programming-language-data-science/2426
25. Pandas python library
https://pandas.pydata.org/

26. Pandas: powerful python data analysis toolkit
https://pandas.pydata.org/pandas-docs/stable/

27. Gensim documentation
https://radimrehurek.com/gensim/intro.html

28. David D. Palmer. *Tokenization and sentence segmentation, Chapter 2*

29. Shai Shalev-Shwartz and Shai Ben-David (2014). '*Understanding Machine Learning: From theory to Algorithms'. Published in 2014 by Cambridge University press*

30. Jiangtao Ren , Sau Dan Lee , Xianlu Chen , Ben Kao , Reynold Cheng and David Cheung. 2009, *Naïve Bayes classification for uncertain data, 2009 Ninth IEEE International Conference on Data Mining.*

31. Aakash Tandel (Aug 2017*), Support vector machine: A brief overview*, Towards Data science website

32. Precision vs Recall - Demystifying Accuracy Paradox in Machine Learning ,
https://www.newgenapps.com/blog/precision-vs-recall-accuracy-paradox-machine-learning

33. Accuracy, precision, Recall, F1 score.
http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

34. Support Vector Machine, Katrina Domijan, Maynooth University

35. Guide to neural network and deep learning
https://skymind.ai/wiki/neural-network

36. Neural Network
https://medium.freecodecamp.org/big-picture-machine-learning-classifying-text-with-neural-networks-and-tensorflow-d94036ac2274

37. Neural Network
https://www.explainthatstuff.com/introduction-to-neural-networks.html

38. Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, Xiaotie Deng, (2013), *Exploiting Topic based Twitter Sentiment for Stock Prediction*

39. Kaoutar Ben Ahmed , Atanas Radenski , Mohammed Bouhorma, Mohamed Ben Ahmed, *Sentiment Analysis for Smart Cities: State of the Art and Opportunities. International Conference, internet computing and internet of things.*

40. Rubayyi Alghamdi, Khalid Alfalqi (2015), *A Survey of Topic Modeling in Text Mining*, International Journal of Advanced Computer Science and Applications.

41. Shibin Zhou, Kan Li, Yushu Liu. *Text Categorization Based on Topic Model,* International Journal of Computational Intelligence Systems vol2 no.4 (December 2009).

42. Pierre FICAMOS, Yan LIUA. *Topic based Approach for Sentiment Analysis on Twitter Data.* International Journal of Advanced Computer Science and Applications, vol. 7, 2016.

43. Bin Lu, Myle Ott, Claire Cardie and Benjamin Tsou. *Multi-aspect Sentiment Analysis with Topic Models*

44. Chenghua Lin, Yulan He '*Joint Sentiment/Topic Model for Sentiment Analysis'*

45. Kilkenny Central Access Scheme
https://en.wikipedia.org/wiki/Kilkenny_Central_Access_Scheme

46. Nabeela Altrabsheh, Mihaela Cocea, Sanaz Fallahkhair. '*Sentiment analysis: towards a tool for analysing real-time students feedback'*

47. Anish Sign Walia (2017), Activation function and it's type- which is better?
https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f

48. Xuan Qi, LDA topic modelling visualization
https://medium.com/@sherryqixuan/topic-modeling-and-pyldavis-visualization-86a543e21f58

49. Anaconda (Python Distribution)
https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)

50. Numpy python documentation
https://en.wikipedia.org/wiki/NumPy
https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.html

51. Keras: Python documentation, https://keras.io/

52. NLTK Documentation, release 3.2.5, Steven Bird, September 2017

53. SciKit learn documentation, Isotonic regression
http://scikit-learn.org/stable/auto_examples/plot_isotonic_regression.html#sphx-glr-auto-examples-plot-isotonic-regression-py

54. Scikit-learn
https://en.wikipedia.org/wiki/Scikit-learn

55. Cross Validation in machine learning
https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f

Appendix

1. Coding part of this project is available in following github Link:

   https://github.com/kannadhaasan/Projects.git