SENTIMENT ANALYSIS USING LOGISTIC REGRESSION

PROJECT REPORT

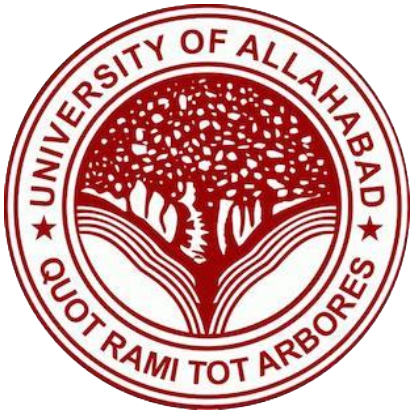
OF MAJOR PROJECT

MASTER OF COMPUTER APPLICATIONS

SUBMITTED BY

TANU JAISWAL
Batch Year– 2019-22
Enrolment No.-U1949014

PROJECT SUPERVISOR- ER. SHREYA AGARWAL



Centre of Computer Education &
Training Institute of Professional Studies
University of Allahabad, Prayagraj,
Uttar Pradesh

ACKNOWLEDGEMENT

This project report is based on '**SENTIMENT ANALYSIS USING LOGISTIC REGRESSION**'. I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to my supervisor **Er. Shreya Agarwal** for the guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project. I would like to express my gratitude towards my parents & members of Institute of Professional Studies (IPS), University of Allahabad for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to my project in-charge **Dr. Sarika Yadav** and coordinator **Prof. Ashish Khare Sir** for providing necessary guidance and giving me such attention and time.

My thanks and appreciations also go to my Friends in developing the project and people who have willingly helped me out with their abilities.

Tanu Jaiswal
MCA 6th Semester
Enrolment No.- U1949014

CERTIFICATE

This is to certify that **Tanu Jaiswal,** Student of **MCA 3RD Year** of **Institute of Professional Studies, University of Allahabad** has completed her project entitled **SENTIMENT ANALYSIS USING LOGISTIC REGRESSION** under my guidance in the academic year (2019-22).

She has taken proper care and shown utmost sincerity in completing this project. Her Work is satisfactory. I wish her all the best for her bright future. I certify that this project is up to my expectations and as per the guidelines.

This application package is the original one and is never submitted somewhere for the same purpose.

ER. SHREYA AGARWAL (PROJECT SUPERVISOR)

DECLARATION BY DISCIPLE

I, **TANU JAISWAL,** solemnly declare that the project report **SENTIMENT ANALYSIS USING LOGISTIC REGRESSION** is based on my own work carried out during the course of our study under the supervision of **ER. SHREYA AGARWAL**.

I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

I. The work contained in the report is original and has been done by me under the general supervision of my supervisor.

II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

III. We have followed the guidelines provided by the university in writing the report.

IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Date: **Tanu Jaiswal**

Place: Prayagraj **MCA 6th Semester**

ABSTRACT

Sentiment Analysis also known as Opinion Mining refers to the use of natural language processing, text analysis to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.
Sentiment Analysis is a method for judging somebody's sentiment or feeling with respect to a specific thing written in a piece of text. It is used to recognize and arrange the sentiments communicated in writings. The web-based social networking sites like twitter draws in a huge number of clients that are online for imparting their insights in the form of tweets or comments. The tweets can be then classified into positive or negative. In the proposed work, logistic regression classification is used as a classifier.

Keywords: Sentiment analysis, Opinion mining, Text classification, Twitter, Polarity, Machine learning, Logistic regression, Natural Language Processing.

SYNOPSIS

1. INTRODUCTION

1. Sentiment Analysis is a method for judging somebody's sentiment or feeling with respect to a specific thing. It is utilized to recognize and arrange the sentiments communicated in writings.

2. Textual information in the world can be broadly categorized into two main types: *facts* and *opinions*. Facts are objective about entities, events and their properties. Opinions are usually subjective expressions that describe people's sentiments, appraisals or feelings toward entities, events and their properties.

3. Before the Web, when an individual needed to make a decision, he/she typically asked for opinions from friends and families. When an organization wanted to find the opinions or sentiments of the general public about its product and services, it conducted opinion polls, surveys, and focus groups.

4. Now, people can post reviews of the product at merchant sites and express their views on almost anything in Internet forums, discussion groups, and blogs, which are collectively called the *user-generated content*.

5. It is difficult for a human reader to find the relevant sources, extract related sentences with opinions, read them, summarize them, and organized them into usable forms. Thus, automated opinion discovery and summarization systems are needed. *Sentiment analysis*, also known as *Opinion Mining*, grows out of this. It is a challenging natural language processing or text mining problem.

6. Sentiment Analysis can be used to analyse web material from social media platform, online products, companies, events and personnel.

7. Sentiment analysis employs a variety of methodologies to determine a text's or sentence's sentiment. The concept of opinion is very broad. We focus word based sentiment analysis that conveys people's positive or negative sentiments.

8. Role of Logistic Regression: It is used as a statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation. This type of analysis can help to predict the likelihood of an event happening or a choice being made.

2. PROBLEM STATEMENT

1. A basic task in sentiment analysis is classifying the *polarity* of a given text at the document, sentence, or feature/aspect level- whether the expressed opinion in a document, a sentence or an entity feature/ aspect is positive and negative.

2. Given a sentence, two sub-tasks can be performed:
    i. Subjectivity Classification: Determine whether the sentence is a subjective (opinion) or an objective (fact) sentence.
    ii. Sentence-level sentiment classification: If a sentence is subjective, determine whether it expresses a positive or a negative opinion.

3. Sentiment classification aims to determine the overall intention of a written text which can be of admiration or criticism type. This can be achieved by using machine learning algorithms such as Naïve Bayes, Support Vector Machine, etc. So, the problem that is going to be investigated in the project is as follow:

Which machine learning approach performs better in terms of accuracy on the Amazon beauty products reviews?

3. MOTIVATION

1. According to Ramteke et al. (2012) motivation for Sentiment Analysis is two-fold. Both consumers and producers highly value "customer's opinion" about products and services. Thus, Sentiment Analysis has seen a considerable effort from industry as well as academia.

2. The Consumer's Perspective: From a consumer's point of view extracting opinions about a particular entity is very important. Trying to go through such a vast amount of information to understand the general opinion is impossible for users just by the sheer volume of this data. Hence, the need of a system that differentiates between good reviews and bad reviews. Further, labelling these documents with their sentiment would provide a succinct summary to the readers about the general opinion regarding an entity.

3. The Producer's Perspective: With the explosion of Web 2.0 platforms such as blogs, discussion forums, etc., consumers have at their disposal, a platform to share their brand experiences and opinions, positive or negative regarding any product or service. According to Pang and Lee (2008) these consumer voices can wield enormous influence in shaping the opinions of other customers and, ultimately, their brand loyalties, their purchase decisions, and their own brand advocacy.
    a. Since the consumers have started using the power of the internet to expand their horizons, there has been a surge of review sites and blogs, where user can perceive a product's or service's advantages and faults. These opinions this shape the future of the product or the service. The vendors need a system that can identify trends in customer reviews and use the, to improve their product or service and also identify the requirements of the future.

4. The Societies' Perspective: Recently, certain events, which affected Government, have been triggered using the Internet. The social networks are being used to bring together people so as to organize mass gatherings and oppose oppression. On the darker side, the

social; networks are being used to insinuate people against an ethnic group or class of people, which has resulted in a serios loss of life. Thus, there is a need for Sentiment Analysis systems that can identify such phenomena and curtail them if needed.

4. OBJECTIVE

1. The fundamental objective of sentiment analysis is to classify and determine the polarity of material on the Internet by using logistic regression for the effective accuracy and the prediction of the chosen data set.

2. This primarily aims to the exploratory data analysis of the chosen data set.

3. The classification of given sentence into two possible sentiment that whether it is a positive or negative.

5. REQUIREMENTS ANALYSIS

Technology

1. Programming Language: Python

Software Requirements

1. Operating system:

    i. Windows OS-
    Windows XP, Windows 7 (Ultimate, & Enterprise), Windows 8 (or 8.1), Windows 10 and later on versions.

    ii. MacOS-
    MacOS X (10.2-10.15), MacOS 11 (Big Sur), MacOS 11.5 (Air M1, Big Sur), MacOS 12 (Monterey Version: 12.2.1) and later On versions.

2. IDE: Jupyter Notebook

3. Tools and Libraries:

    i. Natural Language ToolKit
    ii. Pandas
    iii. NumPy
    iv. TextBlob
    v. SpaCy
    vi. Matplotlib
    vii. TensorFlow, etc.

Minimum Hardware Components

1) Windows (minimum preferred)
    i) Processor: i3
    ii) Hard Disk: 6 GB or more
    iii) Memory: 1 GB RAM

1) MacOS (minimum preferred)
    i) Processor: intel or apple chip
    ii) Hard Disk: 6 GB or more
    iii) Memory: 4 GB RAM

6. SOFTWARE ANALYSIS

The implementation of this project consists of Python as programming language, Jupyter Notebook as IDE (Integrated Development Environment), and libraries such as NLTK (Natural Language ToolKit), Pandas, NumPy, SpaCy, etc.
Following are the descriptions about the factors that are included:

1. Python: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.
Features of python:

i. Large standard library
ii. Dynamic Typed Language
iii. Portable Language

2. Jupyter Notebook: The Jupyter notebook combines two components:

i **A web application**: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.
ii **Notebook documents**: a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

3. Libraries:

- **NLTK:** NLTK is one of the best Python libraries for any task based on natural language processing. Some of the applications where NLTK is best to use are:

  1. Sentiment Analysis
  2. Named Entity Recognition
  3. Part of Speech Tagging
  4. Topic Modelling

  The SentimentIntensityAnalyzer function of this library is very useful for the task of analysing sentiments in a few lines of code. When I work on any task based on sentiment analysis, NLTK is always my first choice.

- **SpaCy:** SpaCy is an industry-standard library that provides extensive functionality for natural language processing applications. Some of the applications where SpaCy can be used are:

  1. Sentiment Analysis
  2. Named Entity Recognition
  3. Model Packaging and deployment
  4. Part of speech tagging

  Although Spacy claims this is an industry force for natural language processing in Python, I will still prefer TextBlob and NLTK for sentiment analysis. Without

a doubt, this is one of the best libraries for sentiment analysis, but I would still prefer SpaCy for named entity recognition only.

- **Pandas:** Pandas is a Python library used for working with data sets. It has functions for analysing, cleaning, exploring, and manipulating data. We can analyse the data in pandas with:

    1. Series: Series is one dimensional (1-D) array defines in pandas that can be used to store any data type. The axis labels are collectively called indexes. Pandas Series is nothing but a column in an excel sheet. Labels need not be unique but must be a hashable type. The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index.

       Data can be:
        a. A Scalar value which can be integer Value, String
        b. A Python Dictionary which can be Key, Value pair
        c. A Ndarray

    2. Data Frames: DataFrames is two dimensional (2-D) size-mutable, potentially heterogeneous tabular data stricture with labelled axes (rows and columns). A data frame is 2-D data structure i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components: The data, the rows and the columns. defines in pandas which consists of rows and columns.

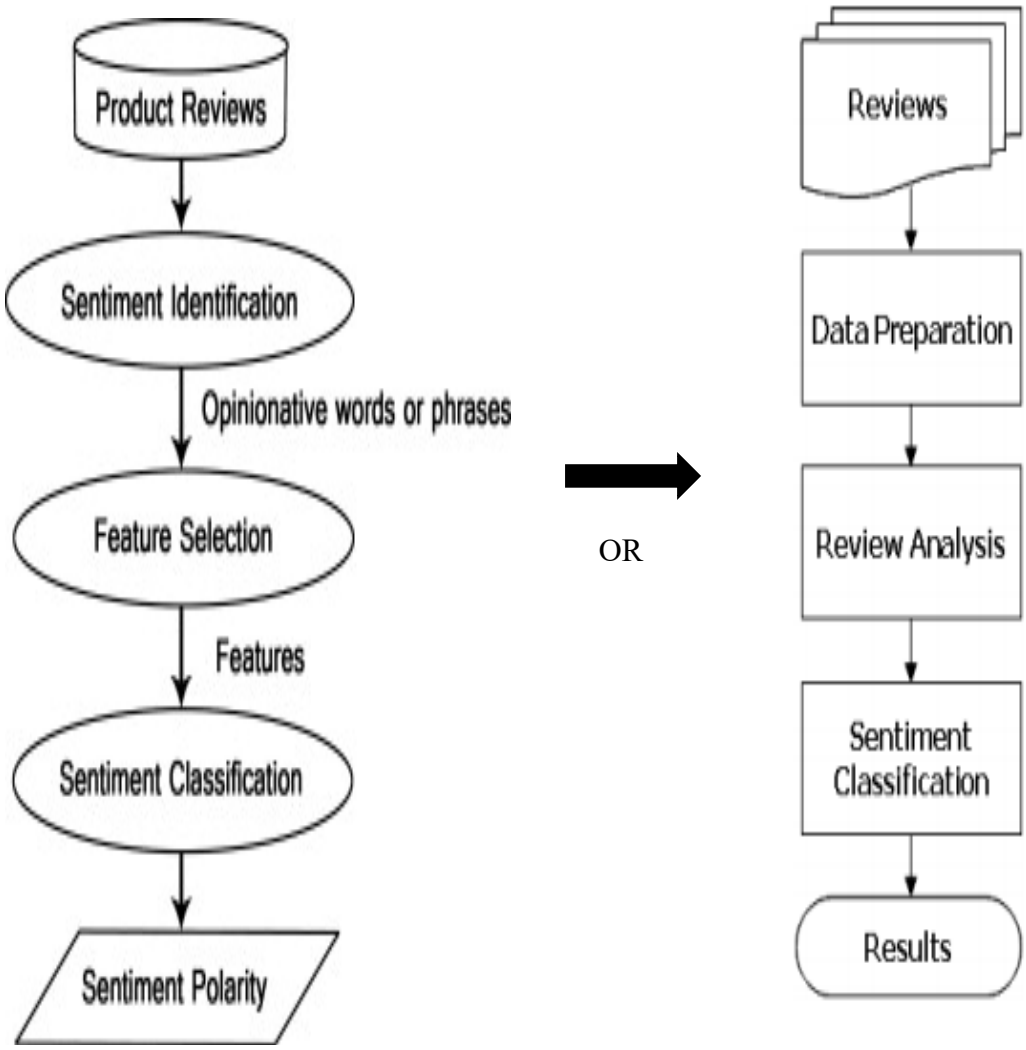
       Here, data can be:

        a. One or more dictionaries
        b. One or mor Series
        c. 2D-numpy Ndarray

7. STUDY DESIGN

To build a machine learning model to accurately classify whether customers are saying positive or negative. To build sentiment analysis text classifier following are the steps to proceed:

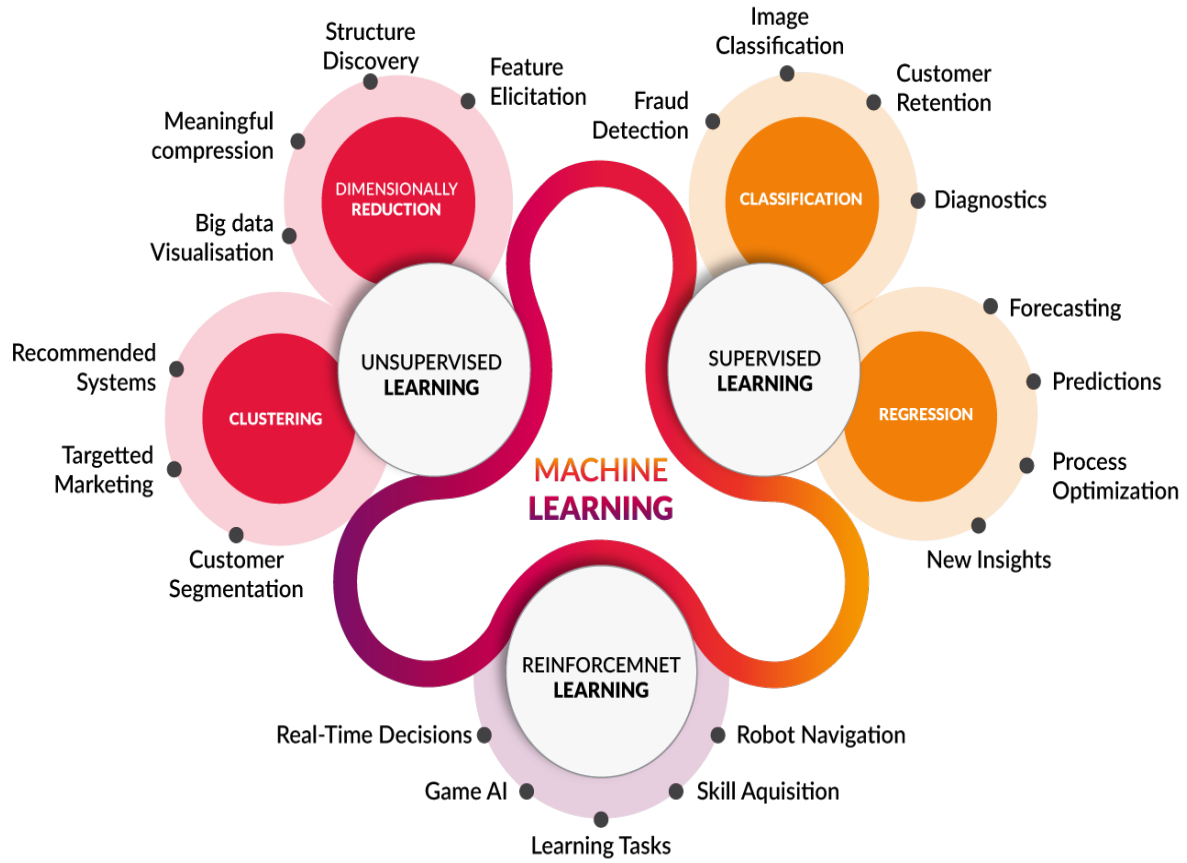
1. Problem definition and solution approach
2. Data Pre-processing of dataset
3. Exploratory data analysis
4. Build The Text Classifier
5. Train the Sentiment Analysis Model



8. TYPE OF PROBLEM & DOMAIN

Machine learning implementations are classified into three major categories, depending on the nature of the learning "signal" or "response" available to a learning system which is as follows:-

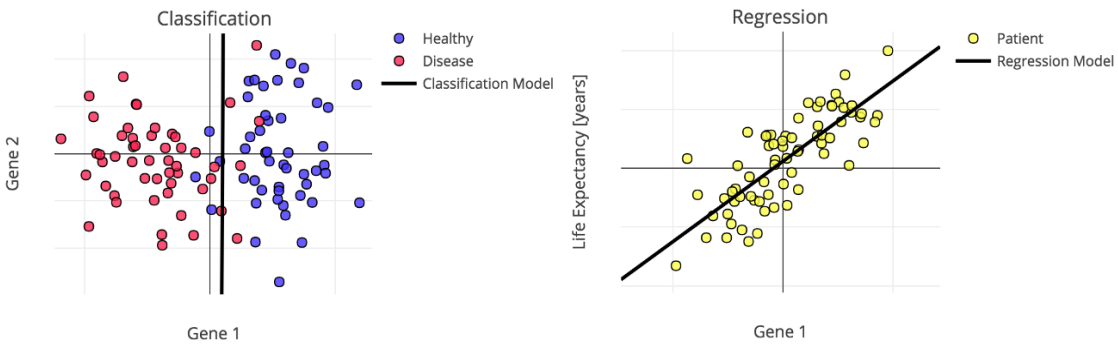
1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Reinforcement Learning



From the above figure we can conclude that the classification and regression are part of supervised learning. Therefore, the supervised learning is being exploited here. There are two main problems that can be solved with Supervised Learning:

1. **Classification** — process of *assigning category to input* data sample. Example usages: predicting whether a person is ill or not, detecting fraudulent transactions, face classifier.

2. **Regression** - process of *predicting a continuous, numerical value* for input data sample. Example usages: assessing the house price, forecasting grocery store food demand, temperature forecasting.



Example of Classification and Regression models.

9. ALGORITHM

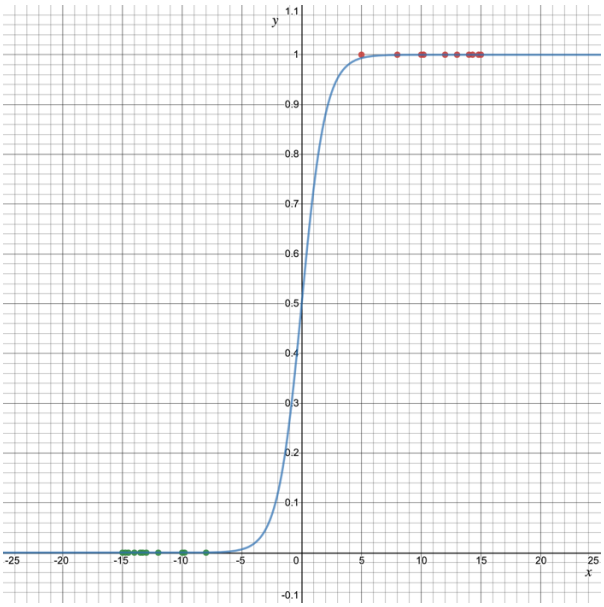
This project focuses on the algorithm called Logistic Regression of supervised learning of machine learning.

Logistic Regression:

This projects includes the Logistic Regression as the type of problem. **Logistic regression** is a supervised learning algorithm which is mostly used to solve binary "**classification"** tasks although it contains the word "**regression**" .
"Regression" contradicts with "classification" but the focus of logistic regression is on the word "logistic" referring to **logistic function** which actually does the classification task in the algorithm.
Logistic regression is a simple yet very effective classification algorithm so it is commonly used for many binary classification tasks. Customer churn, spam email, website or ad click predictions are some examples of the areas where logistic regression offers a powerful solution. It is even used as an activation function for neural network layers.
The basis of logistic regression is the logistic function, also called the **sigmoid function**. The sigmoid function/logistic function is a function that resembles an "S" shaped curve when plotted on a graph. It takes values between 0 and 1 and "squishes" them towards the margins at the top and bottom, labelling them as 0 or 1.

$$Sigmoid\ Function\colon\ y = \frac{1}{1 + e^{-x}}$$

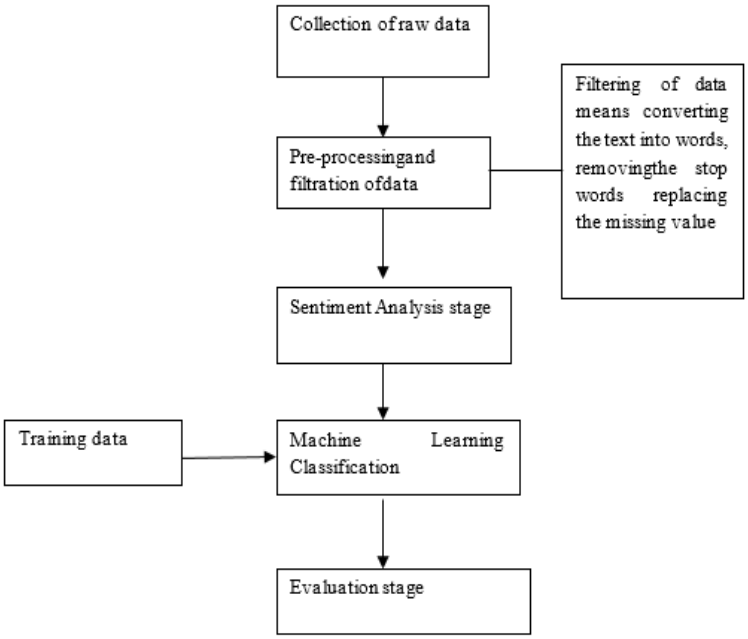
The e represents the exponential function or exponential constant, and it has a value of approximately 2.71828. Logistic regression model takes a linear equation as input and use logistic function and log odds to perform a binary classification task. Let's see how the sigmoid function represent the given dataset.



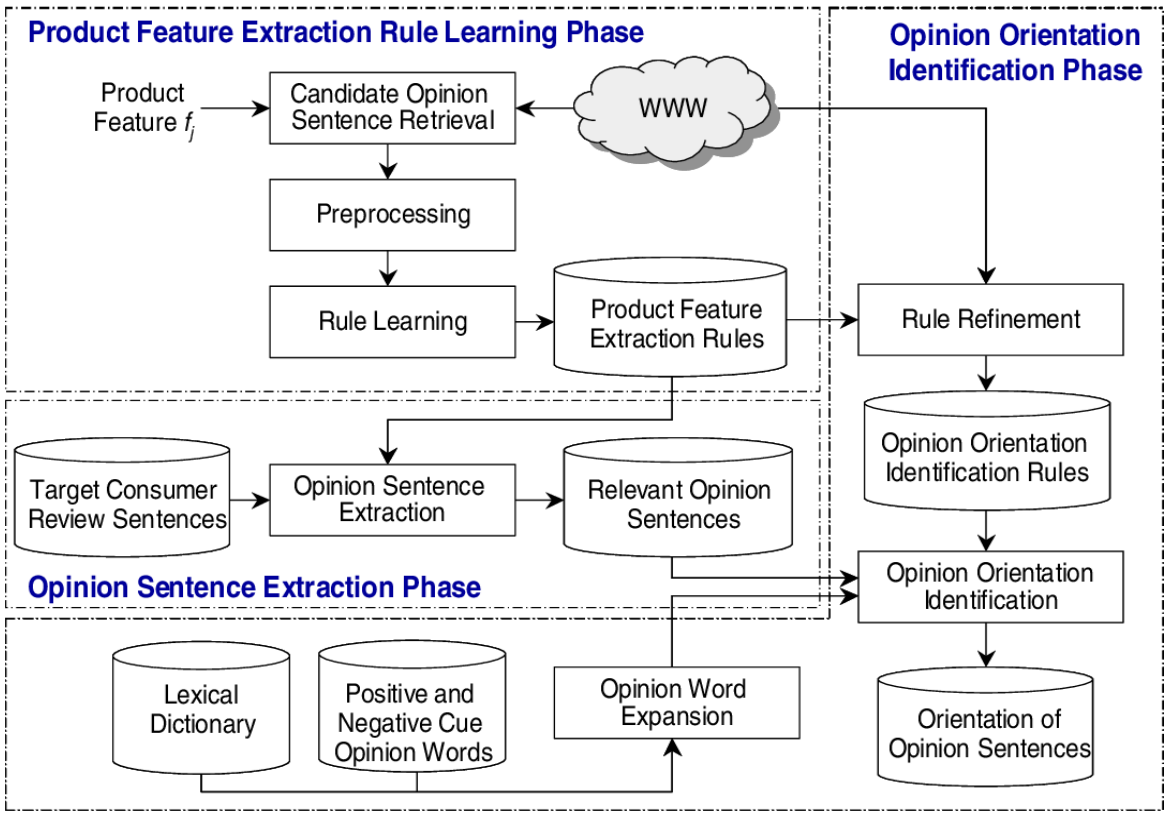
This gives a value y that is extremely close to 0 if x is a large negative value and close to 1 if x is a large positive value. After the input value has been squeezed towards 0 or 1, the input can be run through a typical linear function, but the inputs can now be put into distinct categories.

10. DFDs &/or ER Diagrams &/or Class Diagrams

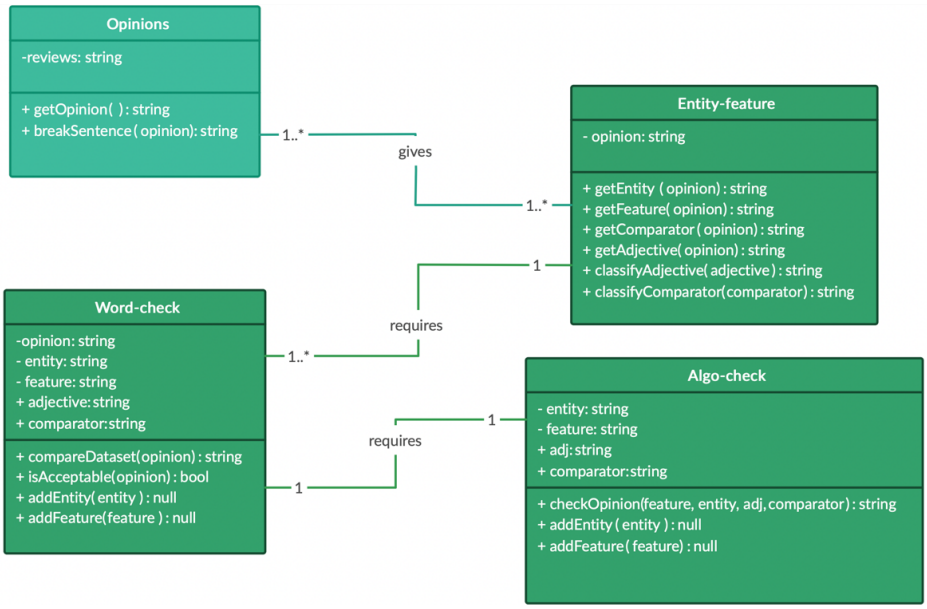
1. Basic Block diagram of sentiment analysis



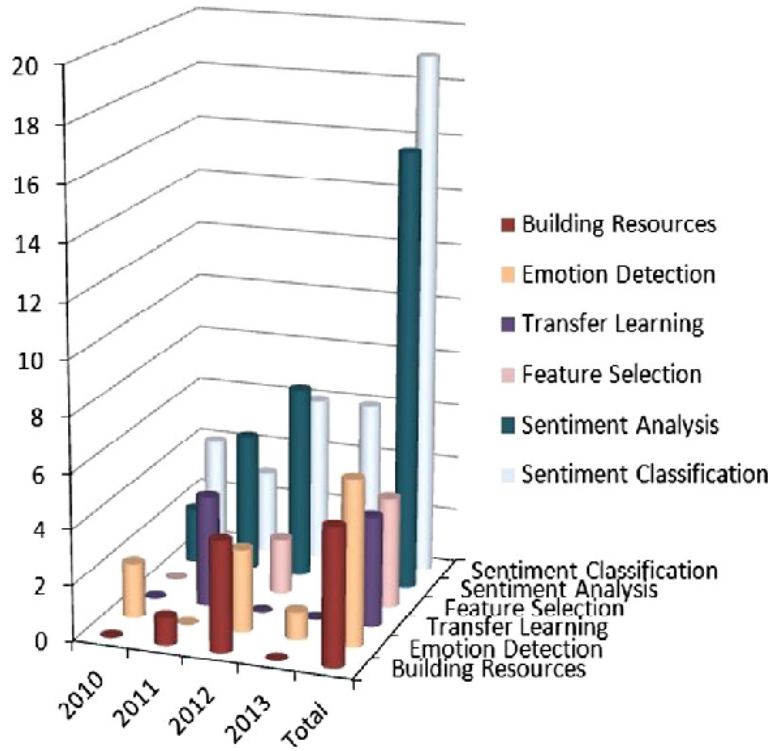
2. Design of the Sentiment Analysis Technique



3. Class Diagram of Opinion Mining



4. Number of articles tar different sentiment analysis tasks over years.



11. MILESTONE

S.No.	Project Activity	Estimated Start Date	Estimated End Date
1.	Synopsis submission	16-03-2022	24-03-2022

12. MEETING WITH THE SUPERVISOR

Date of the meet	Mode	Comments by the supervisor	Signature of the Supervisor
3rd February, 2022	Google Meet		
3rd March, 2022	Offline		
21st March, 2022	Offline		
23rd March, 2022	Offline		
25rd April, 2022	Offline		
10th May, 2022	Offline		

13. BIBLIOGRAPHY & REFERNCES

1. Classifying Text-Based Emotions using logistic regression: Fahad Mazaed Alotaibi Faculty of Computing and Information Technology in Rabigh (FCITR) King Abdul Aziz University (KAU) Jeddah Saudi Arabia.
2. Sentiment analysis system for movie review in Bahasa Indonesia using naive bayes classifier method To cite this article: Yanuar Nurdiansyah et al 2018 J. Phys.: Conf. Ser. 1008 012011
3. Article Sentiment Analysis Based Requirement Evolution Prediction Lingling Zhao 1 and Anping Zhao 2,*
4. Sentiment Analysis Using Multinomial Logistic Regression Ramadhan WP1 , Astri Novianty S.T.,M.T2 ,Casi Setianingsih S.T.,M.T3 Department of Computer Engineering, Telkom University Bandung, Indonesia
5. Sentiment analysis: What is the end user's requirement? Article · June 2012 DOI: 10.1145/2254129.2254173
6. Degree Project In Technology, First Sycle, 15 Credits Stockholm, Sweden, Sentiment classification on Amazon reviews using machine learning approaches Sepideh Paknehad
7. Sentiment Analysis of Product-Based Reviews Using Machine Learning Approaches By Anusuya Dhara (CSE/2014/041) Arkadeb Saha (CSE/2014/048) Sourish Sen Gupta (CSE/2014/049) Pranit Bose (CSE/2014/060)

PROJECT REPORT

CONTENTS

1. INTRODUCTION

1. Sentiment Analysis is a method for judging somebody's sentiment or feeling with respect to a specific thing. It is utilized to recognize and arrange the sentiments communicated in writings.

2. Textual information in the world can be broadly categorized into two main types: *facts* and *opinions*. Facts are objective about entities, events and their properties. Opinions are usually subjective expressions that describe people's sentiments, appraisals or feelings toward entities, events and their properties.

3. Before the Web, when an individual needed to make a decision, he/she typically asked for opinions from friends and families. When an organization wanted to find the opinions or sentiments of the general public about its product and services, it conducted opinion polls, surveys, and focus groups.

4. Now, people can post reviews of the product at merchant sites and express their views on almost anything in Internet forums, discussion groups, and blogs, which are collectively called the *user-generated content*.

5. It is difficult for a human reader to find the relevant sources, extract related sentences with opinions, read them, summarize them, and organized them into usable forms. Thus, automated opinion discovery and summarization systems are needed. *Sentiment analysis*, also known as *Opinion Mining*, grows out of this. It is a challenging natural language processing or text mining problem.

6. Sentiment Analysis can be used to analyse web material from social media platform, online products, companies, events and personnel.

7. Sentiment analysis employs a variety of methodologies to determine a text's or sentence's sentiment. The concept of opinion is very broad. We focus word based sentiment analysis that conveys people's positive or negative sentiments.

8. Role of Logistic Regression: It is used as a statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation. This type of analysis can help to predict the likelihood of an event happening or a choice being made.

2. PROBLEM STATEMENT

1. A basic task in sentiment analysis is classifying the *polarity* of a given text at the document, sentence, or feature/aspect level- whether the expressed opinion in a document, a sentence or an entity feature/ aspect is positive and negative.

2. Given a sentence, two sub-tasks can be performed:
    i. Subjectivity Classification: Determine whether the sentence is a subjective (opinion) or an objective (fact) sentence.
    ii. Sentence-level sentiment classification: If a sentence is subjective, determine whether it expresses a positive or a negative opinion.

3. Sentiment classification aims to determine the overall intention of a written text which can be of admiration or criticism type. This can be achieved by using machine learning algorithms such as Naïve Bayes, Support Vector Machine, etc. So, the problem that is going to be investigated in the project is as follow:

Which machine learning approach performs better in terms of accuracy on the Amazon beauty products reviews?

3. MOTIVATION

1. According to Ramteke et al. (2012) motivation for Sentiment Analysis is two-fold. Both consumers and producers highly value "customer's opinion" about products and services. Thus, Sentiment Analysis has seen a considerable effort from industry as well as academia.

2. The Consumer's Perspective: From a consumer's point of view extracting opinions about a particular entity is very important. Trying to go through such a vast amount of information to understand the general opinion is impossible for users just by the sheer volume of this data. Hence, the need of a system that differentiates between good reviews and bad reviews. Further, labelling these documents with their sentiment would provide a succinct summary to the readers about the general opinion regarding an entity.

3. The Producer's Perspective: With the explosion of Web 2.0 platforms such as blogs, discussion forums, etc., consumers have at their disposal, a platform to share their brand experiences and opinions, positive or negative regarding any product or service. According to Pang and Lee (2008) these consumer voices can wield enormous influence in shaping the opinions of other customers and, ultimately, their brand loyalties, their purchase decisions, and their own brand advocacy.

4. Since the consumers have started using the power of the internet to expand their horizons, there has been a surge of review sites and blogs, where user can perceive a product's or service's advantages and faults. These opinions this shape the future of the product or the service. The vendors need a system that can identify trends in customer reviews and use the, to improve their product or service and also identify the requirements of the future.

5. The Societies' Perspective: Recently, certain events, which affected Government, have been triggered using the Internet. The social networks are being used to bring together people so as to organize mass gatherings and oppose oppression. On the darker side, the social; networks are being used to insinuate people against an ethnic group or class of people, which has resulted in a serios loss of life. Thus, there is a need for Sentiment Analysis systems that can identify such phenomena and curtail them if needed.

4. OBJECTIVE(S)

1. The fundamental objective of sentiment analysis is to classify and determine the polarity of material on the Internet by using logistic regression for the effective accuracy and the prediction of the chosen data set.

2. This primarily aims to the exploratory data analysis of the chosen data set.

3. The classification of given sentence into two possible sentiment that whether it is a positive or negative.

5. PROPOSED SYSTEM

In our proposed system we will be using some machine learning algorithms based on the attributes of the Amazon Reviews Dataset to predict the positive and negative sentiments.

ADVANTAGES OF PROPOSED SYSTEM

1. Normalization: Abbreviated content is normalized by using a dictionary to map the content to frequently used Internet slang words.

2. Stemming: To further facilitate word matching, words in user comments are converted to their root word using the tm_ map function in Python's Snowball C package. For example, "moving," "moved," and "movement" are all converted to "move.".

6. FUTURE SCOPE

1. Sentiment Analysis of Twitter Data Using Logistic Regression Algorithm can be used by people who want to know the sentiment of their tweets. It can be used by company to find out the opinion about their products. It can also be used by many organization to know opinion about the event they have organized or going to be organized.

2. The ability to exploit public sentiment in social media is increasingly considered as an important tool for market understanding, customer segmentation and stock price prediction for strategic marketing planning and manoeuvring. This evolution of technology adoption is energised by the healthy growth in big data framework, which caused applications based on Sentiment Analysis (SA) in big data to become common for businesses. However, scarce works have studied the gaps of SA application in big data

3. The major research scope areas in sentiment analysis are:
    i Spam Detection Sentiment Analysis;
    ii Sentiment Analysis on short Sentence like abbreviations;
    iii Improving sentiment word identification algorithm;
    iv Developing fully automatic analysing tool;
    v Effective Analysis of policy opinionated content;
    vi Successful handling of bi polar sentiments;
    vii Generation of highly content lexicon database.

4. Some of future scopes that can be included in research work are:
    i Use of parser can be embedded into system to improve results.
    ii A web-based application can be made for our work in future.
    iii We can improve our system that can deal with sentences of multiple meanings.
    iv We can also increase the classification categories so that we can get better results.
    v We can start work on multi languages like Hindi, Spanish, and Arabic to provide sentiment analysis to more local.

7. REQUIREMENT ANALYSIS & SPECIFICATION

Technology

1. Programming Language: Python 3.x

Software Requirements

1. Operating system:

    i. Windows OS-
    Windows XP, Windows 7 (Ultimate, & Enterprise), Windows 8 (or 8.1), Windows 10 and later on versions.

    ii. MacOS-
    MacOS X (10.2-10.15), MacOS 11 (Big Sur), MacOS 11.5 (Air M1, Big Sur), MacOS 12 (Monterey Version: 12.2.1) and later On versions.

2. IDE: Jupyter Notebook

3. Tools and Libraries:
    i Anaconda Navigator
    ii Natural Language ToolKit
    iii Pandas
    iv NumPy, etc

Minimum Hardware Components

1. Windows (minimum preferred)
Processor: i5/I7
Hard Disk: 60 GB or more
Memory: 8 GB RAM

2. MacOS (minimum preferred)
Processor: intel or apple chip
Hard Disk: 60 GB or more
Memory: 8 GB RAM

8. FEASIBILITY ANALYSIS

A feasibility study is a preliminary study which investigates the information of prospective users and determines the resources requirements, costs, benefits and feasibility of proposed system. A feasibility study takes into account various constraints within which the system should be implemented and operated. In this stage, the resource needed for the implementation such computing equipment. manpower and costs are estimated. The estimated are compared with available resources and a cost benefit analysis of the system is made. The feasibility analysis. activity involves the analysis of the problem and collection of all relevant information relating to the project. The main objectives of the feasibility study are to determine whether the project would be feasible in terms of economic feasibility, technical feasibility and operational feasibility and schedule feasibility or not. It is to make sure that the input data which are required for the project are available.

Thus we evaluated the feasibility of the system in terms of the following categories:

1. Technical feasibility
2. Operational feasibility
3. Economic feasibility
4. Schedule feasibility

8.1 Technical Feasibility: Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at the point in time there is no any detailed designed of the system, making it difficult to access issues like performance, costs (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis; understand the different technologies involved in the proposed system. Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system. Is the required technology available? Our system is technically feasible since all the required tools are easily available. Python and PHP with JavaScript can be easily handled. Although all tools seems to be easily available there are challenges too.

8.2 Operational Feasibility: Sentiment Analysis Using Logistic Regression has a simple design and is easy to use. It can be easily accessed from anywhere having the internet if we host it on cloud. Hence Sentiment Analysis Using the Logistic Regression was determined operationally feasible.

8.3 Economic Feasibility: Economic feasibility attempts to weigh the costs of developing and implementing a new system, agar Benefits that would accrue from having the new system in place. This feasibility gives the top management the economic justification for the new system A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be useful point of reference to compare actual costs as the project progresses. There could be various types of intangible

benefits on account of automation. These could increase improvement in product quality, better decision making, and timeliness of information, expediting activities, improved accuracy of operations. better documentation and record keeping, faster retrieval of information.
This is a web based application. Creation of application is not costly.

8.4 Schedule Feasibility: A project will fail if it takes too long to be completed before it is useful Typically. this means estimating how long the system will take to develop, and if it can be completed in a given period of time using some methods like payback period. Schedule feasibility is a measure how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some project is initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable.
A minor deviation can be encountered in the original schedule decided at the beginning of the project. The application development is feasible in terms of schedule.

9. TECHNOLOGIES USED

The implementation of this project consists of Python as programming language, Jupyter Notebook as IDE (Integrated Development Environment), and libraries(or modules) such as NLTK (Natural Language ToolKit), Pandas, string, re, and packages such as NLTK: corpus, stem ,tokenize, etc.
Following are the descriptions about the factors that are included:

1. Python: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.
Features of python:

i Large standard library
ii Dynamic Typed Language
iii Portable Language

2. Jupyter Notebook: The Jupyter notebook combines two components:

i **A web application**: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.
ii **Notebook documents**: a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

3. Anaconda Navigator: Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux. The following applications are available by default in Navigator:

i JupyterLab
ii Jupyter Notebook
iii Qt Console
iv Sypder
v Glue
vi Orange
vii RStudio
viii Visual Studio Code
ix Data lore
x IBM Watson Studio Cloud
xi PyCharm Community
xii PyCharm Professional

4. Libraries and Modules:

i. NLTK: NLTK is one of the best Python libraries for any task based on natural language processing. Some of the applications where NLTK is best to use are:

5. Sentiment Analysis
6. Named Entity Recognition
7. Part of Speech Tagging
8. Topic Modelling

The SentimentIntensityAnalyzer function of this library is very useful for the task of analysing sentiments in a few lines of code. When I work on any task based on sentiment analysis, NLTK is always my first choice.

ii. Pandas: Pandas is a Python library used for working with data sets. It has functions for analysing, cleaning, exploring, and manipulating data. We can analyse the data in pandas with:

3. Series: Series is one dimensional (1-D) array defines in pandas that can be used to store any data type. The axis labels are collectively called indexes. Pandas Series is nothing but a column in an excel sheet. Labels need not be unique but must be a hashable type. The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index. Data can be:
    a. A Scalar value which can be integer Value, String
    b. A Python Dictionary which can be Key, Value pair
    c. A Ndarray
4. Data Frames: DataFrames is two dimensional (2-D) size-mutable, potentially heterogeneous tabular data stricture with labelled axes (rows and columns). Pandas DataFrame consists of three principal components: The data, the rows and the columns. defines in pandas which consists of rows and columns. Here, data can be:
    a. One or more dictionaries
    b. One or mor Series
    c. 2D-numpy Ndarray

iii. Re: The re module provides operations for regular expression matching, useful for pattern and string search.

iv. String: String module contains some constants, utility function, and classes for string manipulation. It's a built-in module and we have to import it before using any of its constants and classes.

5. Packages:

i. nltk:corpus

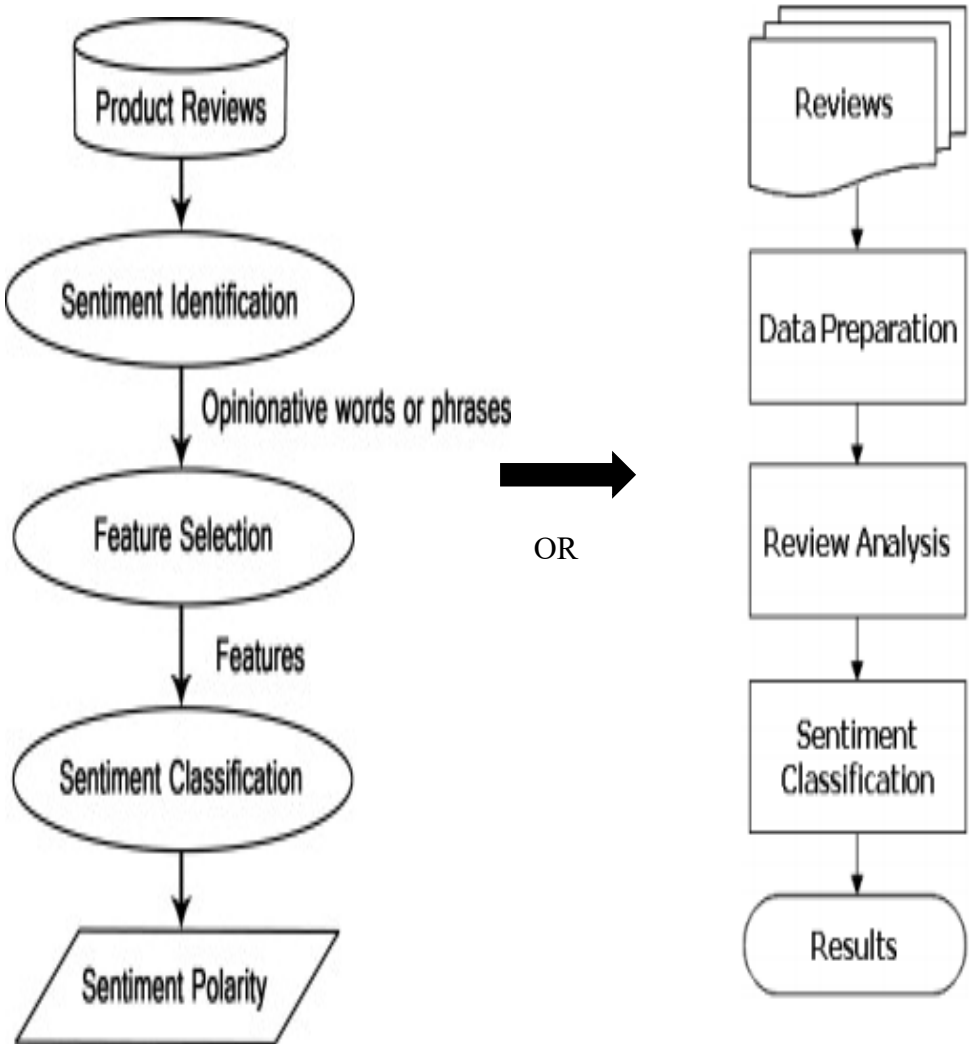
ii. nltk:stem

iii. nltk:tokenize

10. SYSTEM DESIGN

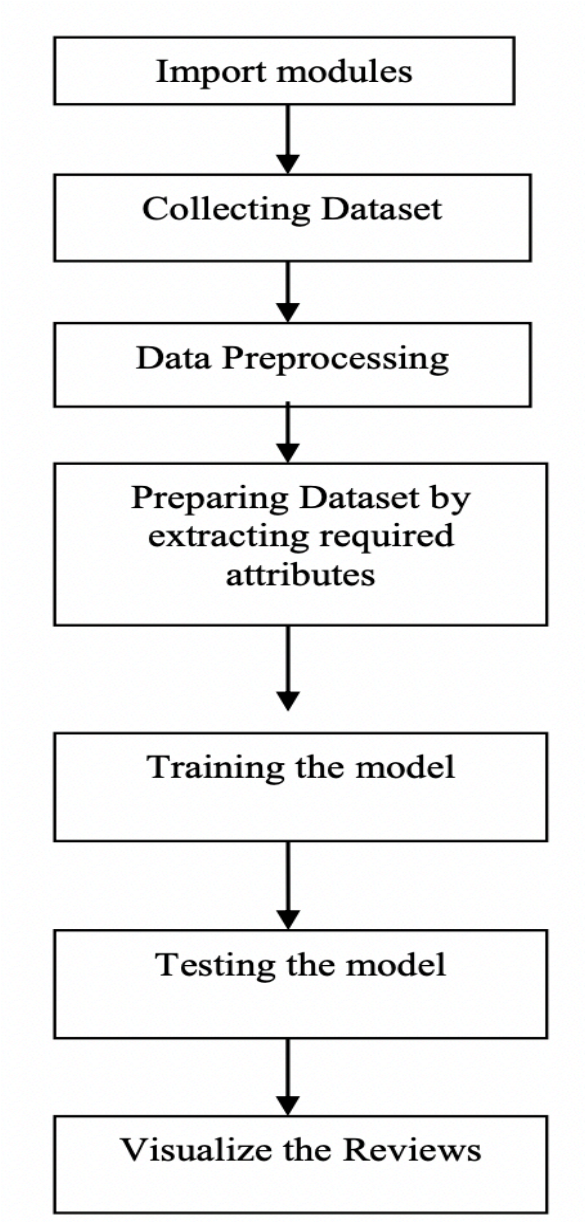
To build a machine learning model to accurately classify whether customers are saying positive or negative. To build sentiment analysis text classifier following are the steps to proceed:

6. Problem definition and solution approach
7. Data Pre-processing of dataset
8. Exploratory data analysis
9. Build The Text Classifier
10. Train the Sentiment Analysis Model



11. SYSTEM ARCHITECTURE

Below diagram depicts the whole system architecture of the Sentiment Analysis Using Logistic Regression of Twitter's Tweets:

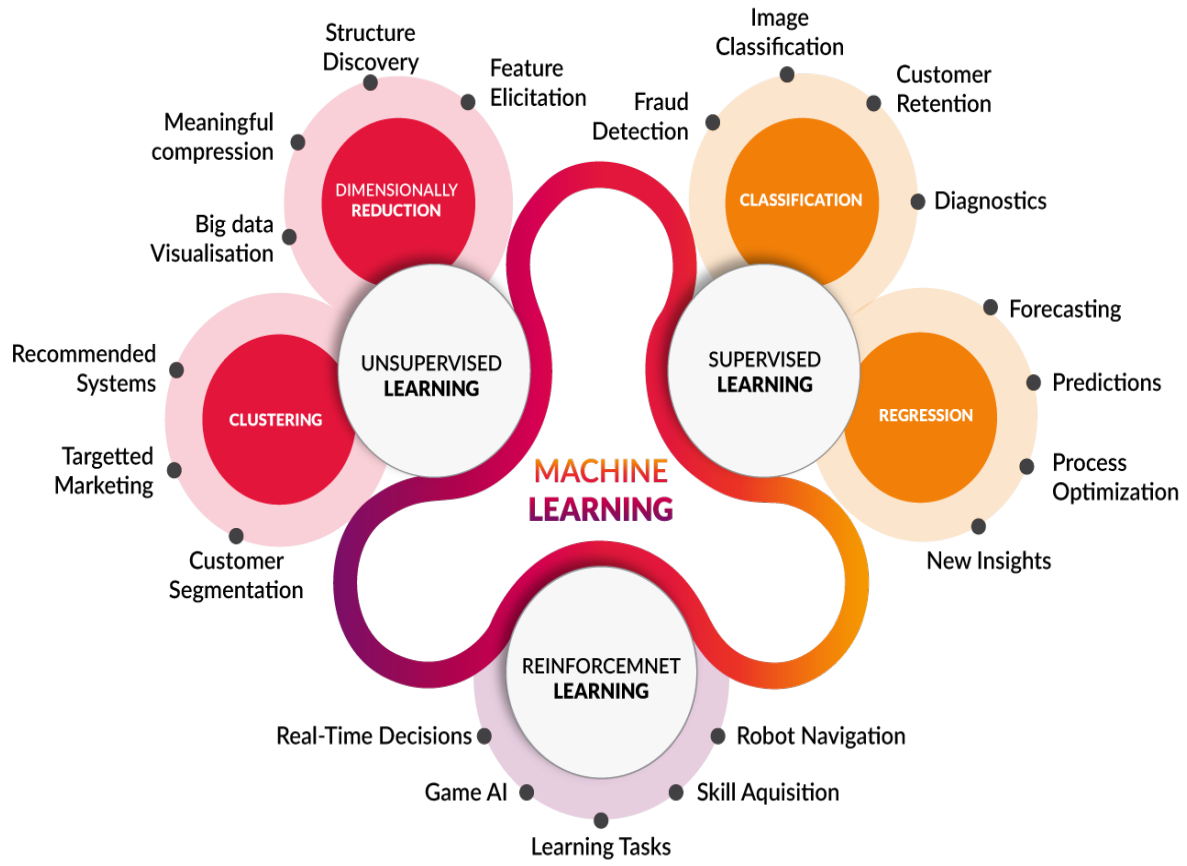


System Architecture

12. TYPE OF PROBLEM AND DOMAIN

Machine learning implementations are classified into three major categories, depending on the nature of the learning "signal" or "response" available to a learning system which is as follows:-

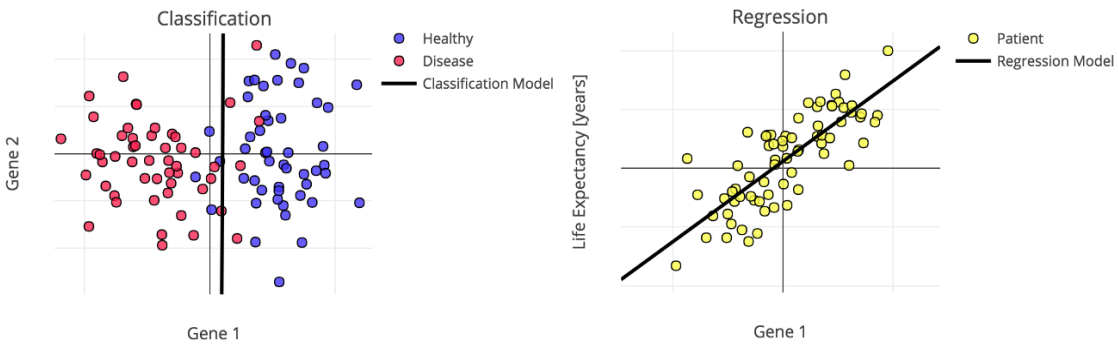
4. Supervised Machine Learning
5. Unsupervised Machine Learning
6. Reinforcement Learning



From the above figure we can conclude that the classification and regression are part of supervised learning. Therefore, the supervised learning is being exploited here. There are two main problems that can be solved with Supervised Learning:

3. **Classification** — process of *assigning category to input* data sample. Example usages: predicting whether a person is ill or not, detecting fraudulent transactions, face classifier.

4. **Regression** - process of *predicting a continuous, numerical value* for input data sample. Example usages: assessing the house price, forecasting grocery store food demand, temperature forecasting.



Example of Classification and Regression models.

13. METHODOLOGY

This project use supervised machine learning classifier which is Logistic Regression. Logistic Regression requires labelled data for training the classifier.

13.1 Data collection

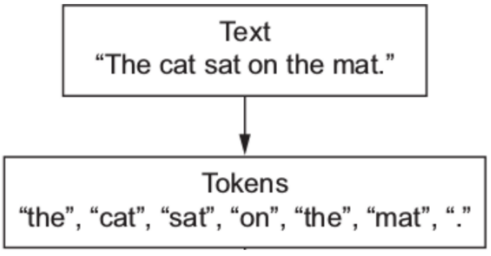
Data used by the Sentiment Analysis Using Logistic Regression Algorithm was collected from the publicly available data which is already being placed by NLTK. It is collected from the nltk:corpus, called as, twitter_samples. The dataset consists of combination of 10 to 20 thousands training and testing data in csv format and is labelled 0 for negative and 1 for positive. So this project uses only the positive and negative datasets.

13.2 Data Pre-processing

The twitter data consist of different properties in which most of it is not useful for sentiment analysis.

Data pre-processing includes various step.

1. **Usernames:** Twitter consists of username which consist of symbol @ at the beginning.eg @sparkingroshan. It is replaced by the word AT_USER in data sets which is started by @ in the datasets.
2. **Usages Link:** User includes the link in the tweets for the more detail information which is not useful for sentiment analysis. The link is replaced by the word „URL".
3. **Stop Words:** Stop word are those filler words which are not useful in for sentiment. These words includes most repeated word like a, an, the, for, etc. These words does not give any sentiment hence they are filtered out form the datasets.
4. **Removing Hash-tags**: Hash-tag in a tweet is used to emphasize a particular word or sentence, for example #thisisgood. Removal of these hash-tags is important because these hash-tags do not define any sentiment. Thus, pre-processing is done and hash-tags before any word are removed.
5. **Tokenization:** Tokenization is the process of converting text as a string into processable elements called tokens. In the context of a tweet, these elements can be words, emoticons, URL links, hashtags or punctuations. Tokenization of reviews after removal of STOP words which mean nothing related to sentiment is the basic requirement for POS tagging.



After proper removal of STOP words like "am, is, are, the, but" and so on the remaining sentences are converted in tokens. These tokens take part in POS tagging. In natural language processing, part-of-speech (POS) taggers have been developed to classify words based on their parts of speech.

For sentiment analysis, a POS tagger is very useful because of the following two reasons:

i. Words like nouns and pronouns usually do not contain any sentiment. It is able to filter out such words with the help of a POS tagger;
ii. A POS tagger can also be used to distinguish words that can be used in different parts of speech.

6. **Repeated Letters:** Tweets contain the very causal language (Walker, Wadhwani, Pooja, & Kola, 2016) so the word such as hurrayyy is replaced with actual word hurray. The letter repeated more time reduced to the one.
7. **Stemming:** Change a word in the text into its base term or root term. Example, happiness to happy.

13.3 Feature Extraction

After pre-processing the tweets, tweets is converted into feature vector. Feature vector are the most important concept in implementing classifier. Feature vector is used for building the model and is used to train the model which is further used to classify the unseen data. Feature vector is the n-dimensional vector of numerical features that represent the some object. In tweets we can consider the presence or absence of words that appear in the tweets. The tweets in training data is split into words and each words into feature words. The feature words may consist of words unigram or bigrams. This project consider unigram as feature words. For eg. This is the ball is represented as this, is, the, ball as unigrams. The entire feature vector will be the combination of each of this feature words.

13.4 Classifier

This class implements the Logistic Regression algorithm which take feature from Feature extractor.
Input: It takes input from the feature extractor.
Process: It classify the tweets as positive or negative and return the list of tweets with its sentiment value.
Output: It gives classified tweets with positive and negative tweets value

14. ALGORITHM

This project focuses on the algorithm called Logistic Regression of supervised learning of machine learning.

Logistic Regression:

This projects includes the Logistic Regression as the type of problem. **Logistic regression** is a supervised learning algorithm which is mostly used to solve binary "**classification"** tasks although it contains the word "**regression**" .

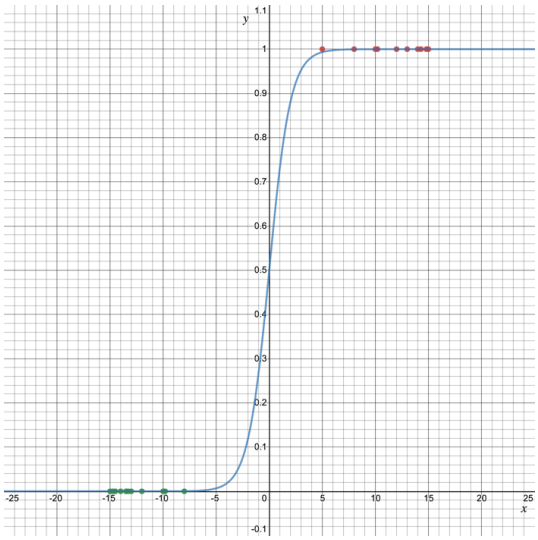
"Regression" contradicts with "classification" but the focus of logistic regression is on the word "logistic" referring to **logistic function** which actually does the classification task in the algorithm.

Logistic regression is a simple yet very effective classification algorithm so it is commonly used for many binary classification tasks. Customer churn, spam email, website or ad click predictions are some examples of the areas where logistic regression offers a powerful solution. It is even used as an activation function for neural network layers.

The basis of logistic regression is the logistic function, also called the **sigmoid function**. The sigmoid function/logistic function is a function that resembles an "S" shaped curve when plotted on a graph. It takes values between 0 and 1 and "squishes" them towards the margins at the top and bottom, labelling them as 0 or 1.

$$Sigmoid\ Function\colon\ y = \frac{1}{1 + e^{-x}}$$

The e represents the exponential function or exponential constant, and it has a value of approximately 2.71828. Logistic regression model takes a linear equation as input and use logistic function and log odds to perform a binary classification task. Let's see how the sigmoid function represent the given dataset.



This gives a value y that is extremely close to 0 if x is a large negative value and close to 1 if x is a large positive value. After the input value has been squeezed towards 0 or 1, the input can be run through a typical linear function, but the inputs can now be put into distinct categories.

Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical). A linear regression is not appropriate for predicting the value of a binary variable for two reasons:

1. A linear regression will predict values outside the acceptable range (e.g. predicting probabilities outside the range 0 to 1)
2. Since the dichotomous experiments can only have one of two possible values for each experiment, the residuals will not be normally distributed about the predicted line.

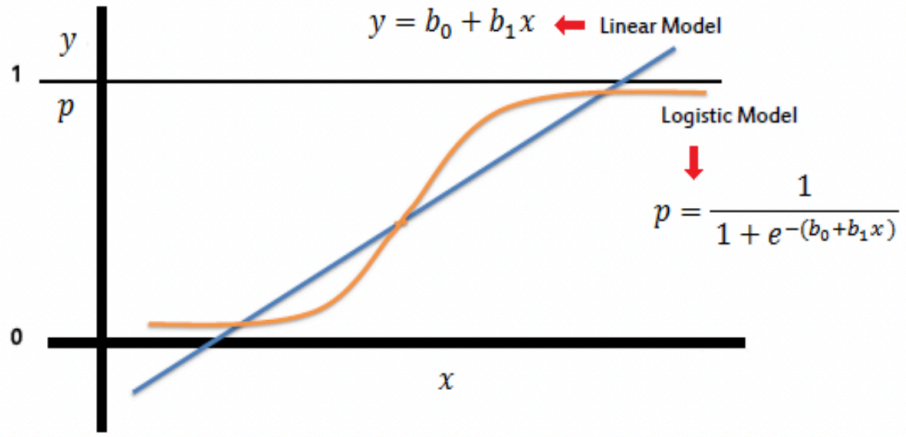
On the other hand, a logistic regression produces a logistic curve, which is limited to values between 0 and 1. Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the "odds" of the target variable, rather than the probability. Moreover, the predictors do not have to be normally distributed or have equal variance in each group. Logistic regression uses maximum likelihood estimation (MLE) to obtain the model coefficients that relate predictors to the target. After this initial function is estimated, the process is repeated until LL (Log Likelihood) does not change significantly.

$$\boldsymbol{\beta^1 = \beta^0 + [X^T W X]^{-1}.X^T(y-\mu)}$$

$\boldsymbol{\beta}$ *is a vector of the logistic regression coefficients.*

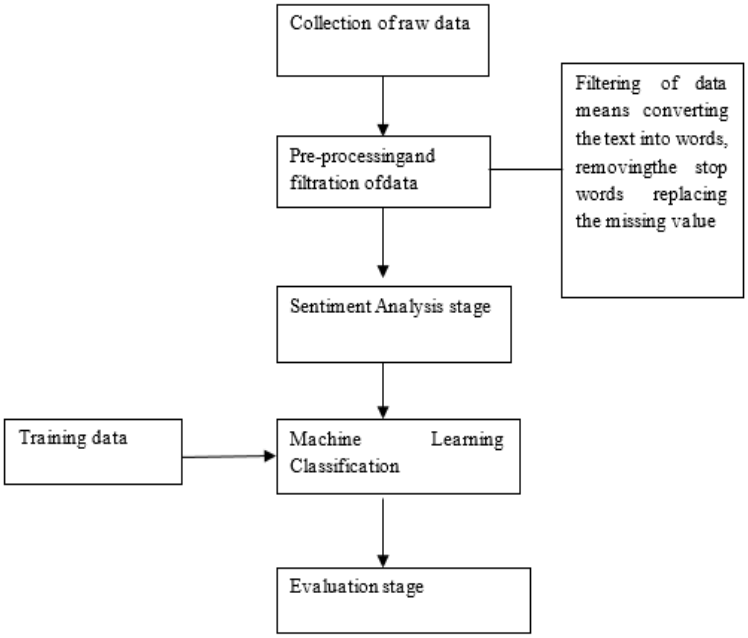
$\boldsymbol{W}$ *is a square matrix of order N with elements* $n_i\pi_i(1-\pi_i)$ *on the diagonal and zeros everywhere else.*

$\boldsymbol{\mu}$ *is a vector of length N with elements* $\mu_i = n_i\pi_i$.

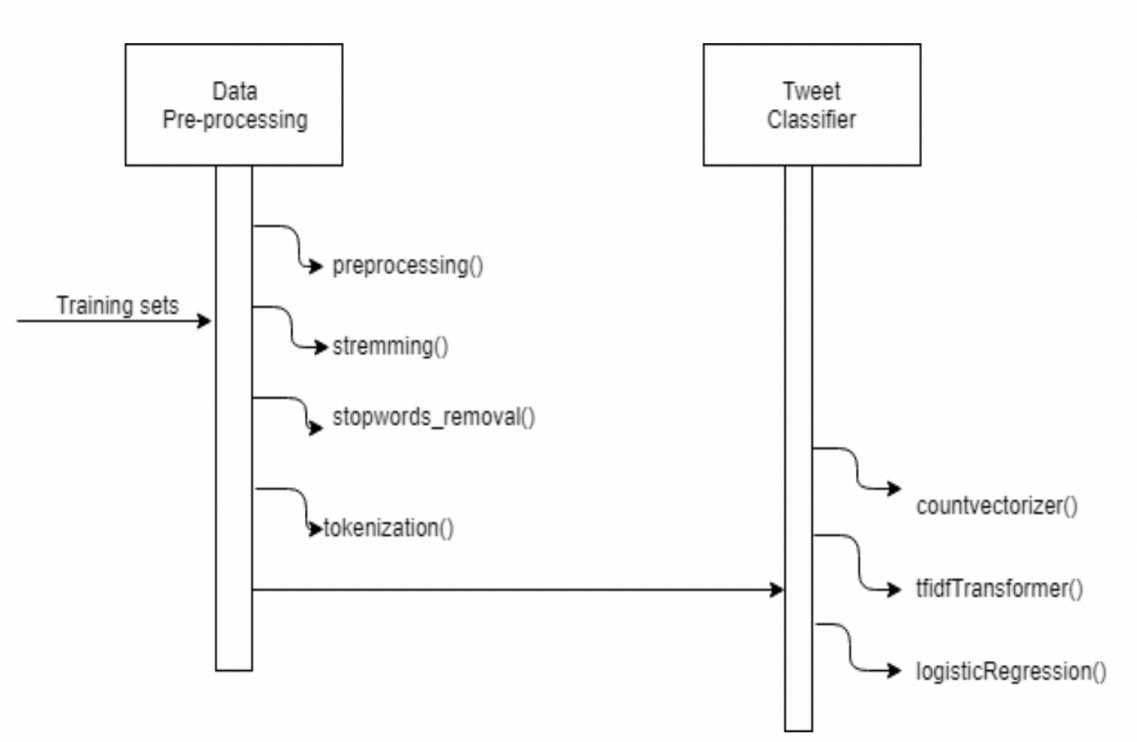


15. STUDY DESIGN

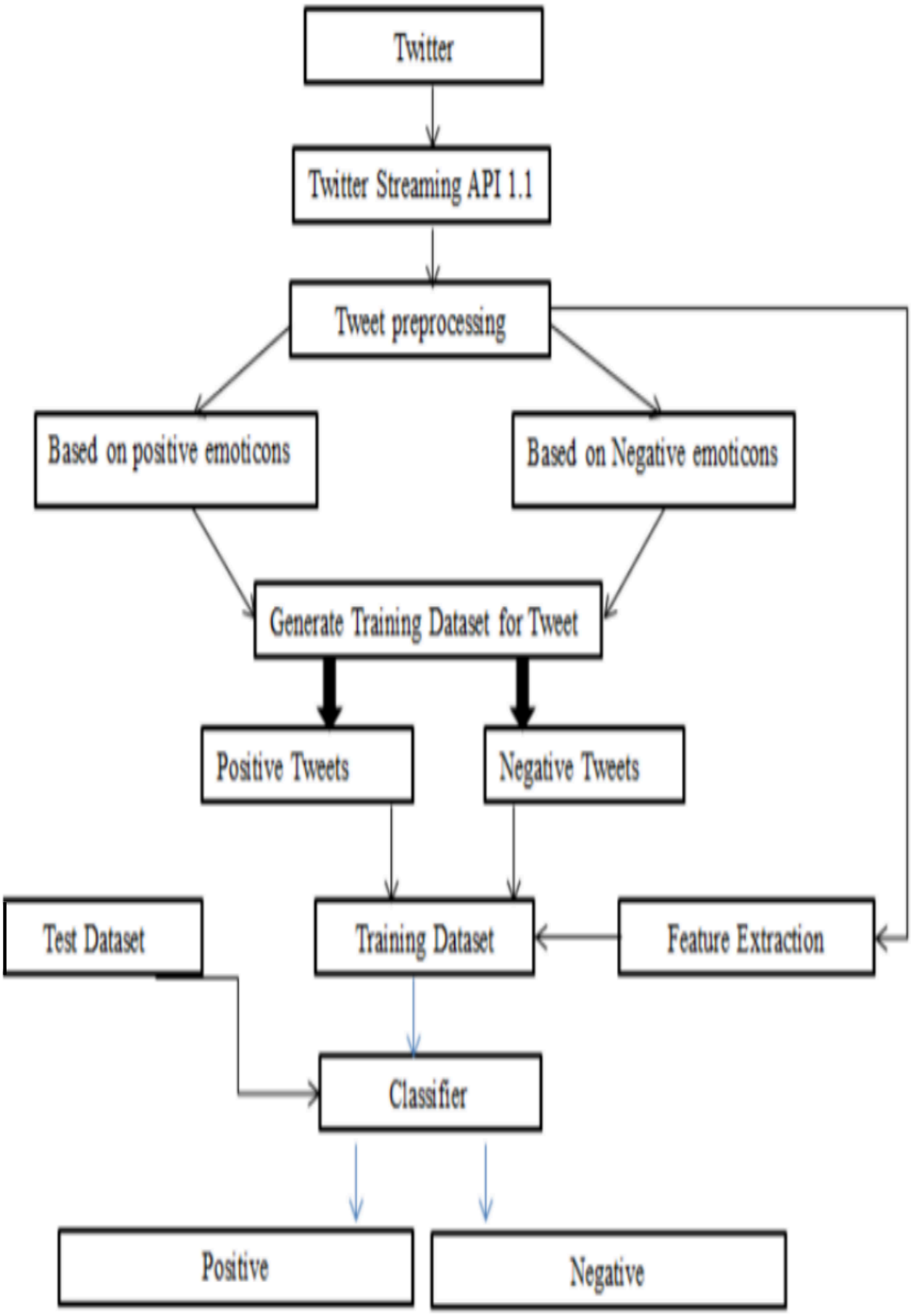
15.1 Basic Block diagram of sentiment analysis



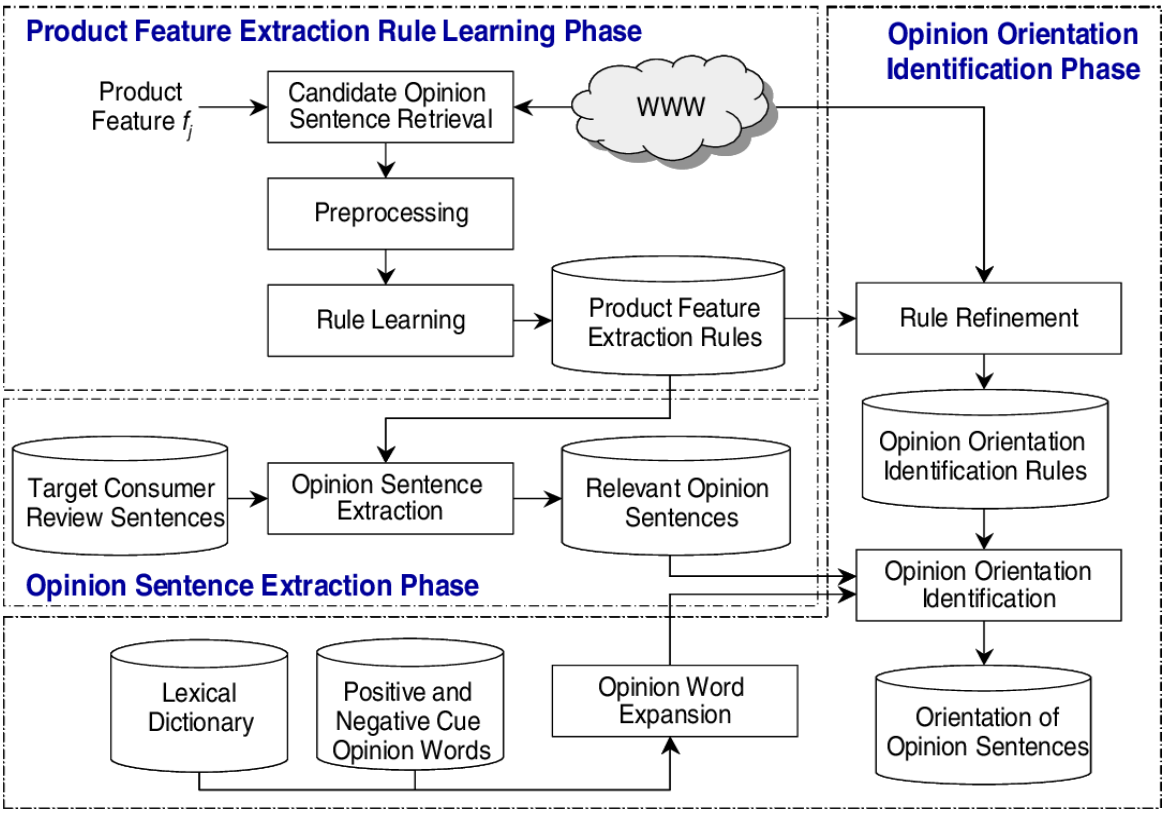
15.2 Sequence Diagram



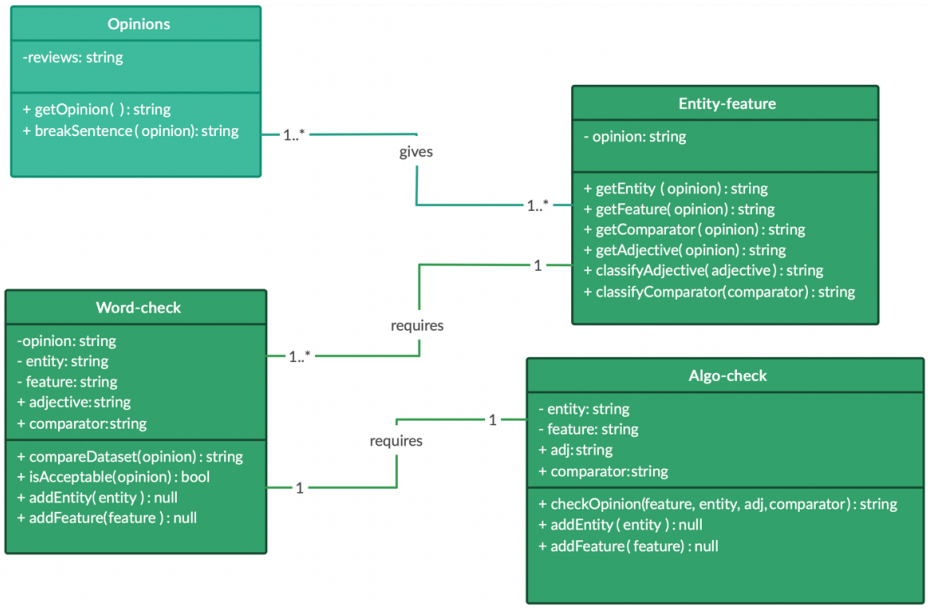
15.3 Structure Chart



15.4 Design of the Sentiment Analysis Technique



15.5 Class Diagram of Opinion Mining



16. IMPLEMENTATION

User can access the application through a browser and see the interface.

16.1 Tools used

Server Side:

a) Python programming language is used to implement the core program logic.
b) NumPy is to manipulate the large multidimensional arrays and matrices.
c) Pandas is used for data manipulation and analysis.
d) NLTK is used for natural language processing which contains text processing libraries.

All the algorithms for the application are written in Python. Algorithms used in Sentiment Analysis of Twitter Data Using Logistic Regression is Predictive analysis model. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. The algorithm is codded in python programming language

16.2 Description of Major Function

The major function in the application are:

Pre-processing :
This is the function which is run for processing the tweets.
Input: It takes the inputs as tweets
Process: It call other function like remove URL, filter stop words, etc.
Output: It gives the list of the process tweets.

Feature extractor :
This function is implemented after the pre-processing data.
Input: This takes pre-processed data as in input.
Process: It then uses the method, extract feature to process the taken input, process and extract the feature.

Classifier:
This class implements the Logistic Regression algorithm which take feature from Feature extractor.
Input: It takes input from the feature extractor.
Process: It classify the tweets as positive or negative and return the list of tweets with its sentiment value.
Output: It gives classified tweets with positive and negative tweets value.

16.3 Testing

Among the total data 80% of the data is used for training and 20% is used for testing.

17. SAMPLE CODE

main.ipynb

```
import nltk
from os import getcwd

nltk.download('twitter_samples')
nltk.download('stopwords')

filePath = f"{getcwd()}/../tmp2/"
nltk.data.path.append(filePath)

import numpy as np
import pandas as pd
from nltk.corpus import twitter_samples

from utils import process_tweet, build_freqs

all_positive_tweets = twitter_samples.strings('positive_tweets.json')
all_negative_tweets = twitter_samples.strings('negative_tweets.json')

test_pos = all_positive_tweets[4000:]
train_pos = all_positive_tweets[:4000]
test_neg = all_negative_tweets[4000:]
train_neg = all_negative_tweets[:4000]


train_y = np.append(np.ones((len(train_pos), 1)), np.zeros((len(train_neg), 1)), axis=0)
test_y = np.append(np.ones((len(test_pos), 1)), np.zeros((len(test_neg), 1)), axis=0)

print("train_y.shape = " + str(train_y.shape))
print("test_y.shape = " + str(test_y.shape))

freqs = build_freqs(train_x, train_y)

print("type(freqs) = " + str(type(freqs)))
print("len(freqs) = " + str(len(freqs.keys())))

print('This is an example of a positive tweet: \n', train_x[0])
print('\nThis is an example of the processed version of the tweet: \n',
process_tweet(train_x[0]))

def sigmoid(z):

    h = 1/(1+np.exp(-z))

    return h
```

```
if (sigmoid(0) == 0.5):
    print('SUCCESS!')
else:
    print('Oops!')

if (sigmoid(4.92) == 0.9927537604041685):
    print('CORRECT!')
else:
    print('Oops again!')

-1 * (1 - 0) * np.log(1 - 0.9999)

-1 * np.log(0.0001)

def gradientDescent(x, y, theta, alpha, num_iters):

    m = len(x)

    for i in range(0, num_iters):

        z = np.dot(x,theta)

        h = sigmoid(z)

        J = -(1/m) * (np.dot(np.transpose(y), np.log(h)) +  np.dot(np.transpose(1-y), np.log(1-
h)))

        theta = theta - (alpha/m)*(np.dot(np.transpose(x),(h-y)))

    J = float(J)
    return J, theta

np.random.seed(1)

tmp_X = np.append(np.ones((10, 1)), np.random.rand(10, 2) * 2000, axis=1)

tmp_Y = (np.random.rand(10, 1) > 0.35).astype(float)


tmp_J, tmp_theta = gradientDescent(tmp_X, tmp_Y, np.zeros((3, 1)), 1e-8, 700)
print(f"The cost after training is {tmp_J:.8f}.")
print(f"The resulting vector of weights is {[round(t, 8) for t in np.squeeze(tmp_theta)]}")

def extract_features(tweet, freqs):

    word_l = process_tweet(tweet)
    x = np.zeros((1, 3))
    x[0,0] = 1
```

```
for word in word_l:
        if (word, 1) in freqs:
            x[0,1] += freqs[(word, 1)]

        if (word, 0) in freqs:
            x[0,2] += freqs[(word, 0)]

    assert(x.shape == (1, 3))
    return x

tmp1 = extract_features(train_x[0], freqs)
print(tmp1)

tmp2 = extract_features('blorb bleeeeb bloooob', freqs)
print(tmp2)

X = np.zeros((len(train_x), 3))
for i in range(len(train_x)):
    X[i, :]= extract_features(train_x[i], freqs)

Y = train_y

J, theta = gradientDescent(X, Y, np.zeros((3, 1)), 1e-9, 1500)
print(f"The cost after training is {J:.8f}.")
print(f"The resulting vector of weights is {[round(t, 8) for t in np.squeeze(theta)]}")

def predict_tweet(tweet, freqs, theta):
    x = extract_features(tweet, freqs)
    y_pred = sigmoid(np.dot(x,theta))
    return y_pred

for tweet in ['I am happy', 'I am bad', 'this movie should have been great.', 'great', 'great great',
'great great great', 'great great great great']:
    print( '%s -> %f' % (tweet, predict_tweet(tweet, freqs, theta)))

my_tweet = 'I am learning :)'
predict_tweet(my_tweet, freqs, theta)

def test_logistic_regression(test_x, test_y, freqs, theta):
    y_hat = []
    for tweet in test_x:
        y_pred = predict_tweet(tweet, freqs, theta)
        if y_pred > 0.5:
            y_hat.append(1)
        else:
            y_hat.append(0)
    y_hat = np.asarray(y_hat)
    test_y = np.squeeze(test_y)
```

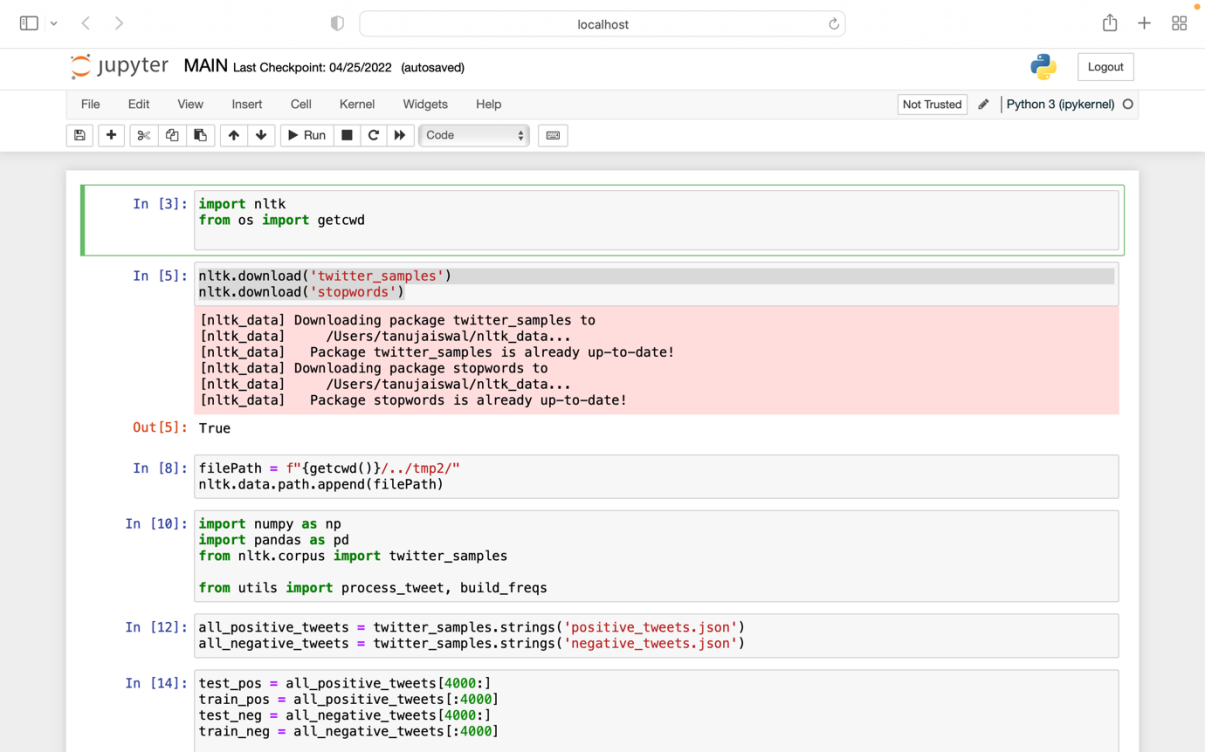
```
    count =0
    for i in range(len(test_y)):
        if (test_y[i] == y_hat[i]):
            count = count+ 1

        else:
            count

    accuracy = count/(len(test_y))
    return accuracy

tmp_accuracy = test_logistic_regression(test_x, test_y, freqs, theta)
print(f"Logistic regression model's accuracy = {tmp_accuracy:.4f}")

print('Label Predicted Tweet')
for x,y in zip(test_x,test_y):
    y_hat = predict_tweet(x, freqs, theta)

    if np.abs(y - (y_hat > 0.5)) > 0:
        print('THE TWEET IS:', x)
        print('THE PROCESSED TWEET IS:', process_tweet(x))
        print('%d\t%0.8f\t%s' % (y, y_hat, ' '.join(process_tweet(x)).encode('ascii', 'ignore')))

#my_tweet_1 = 'xyz'
my_tweet_2 = 'Ridiculous'
print(process_tweet(my_tweet_2))
y_hat = predict_tweet(my_tweet_2, freqs, theta)
print(y_hat)
if y_hat > 0.5:
    print('Positive sentiment')
else:
    print('Negative sentiment')
['ridicul']
[[0.49958364]]
Negative sentiment
```

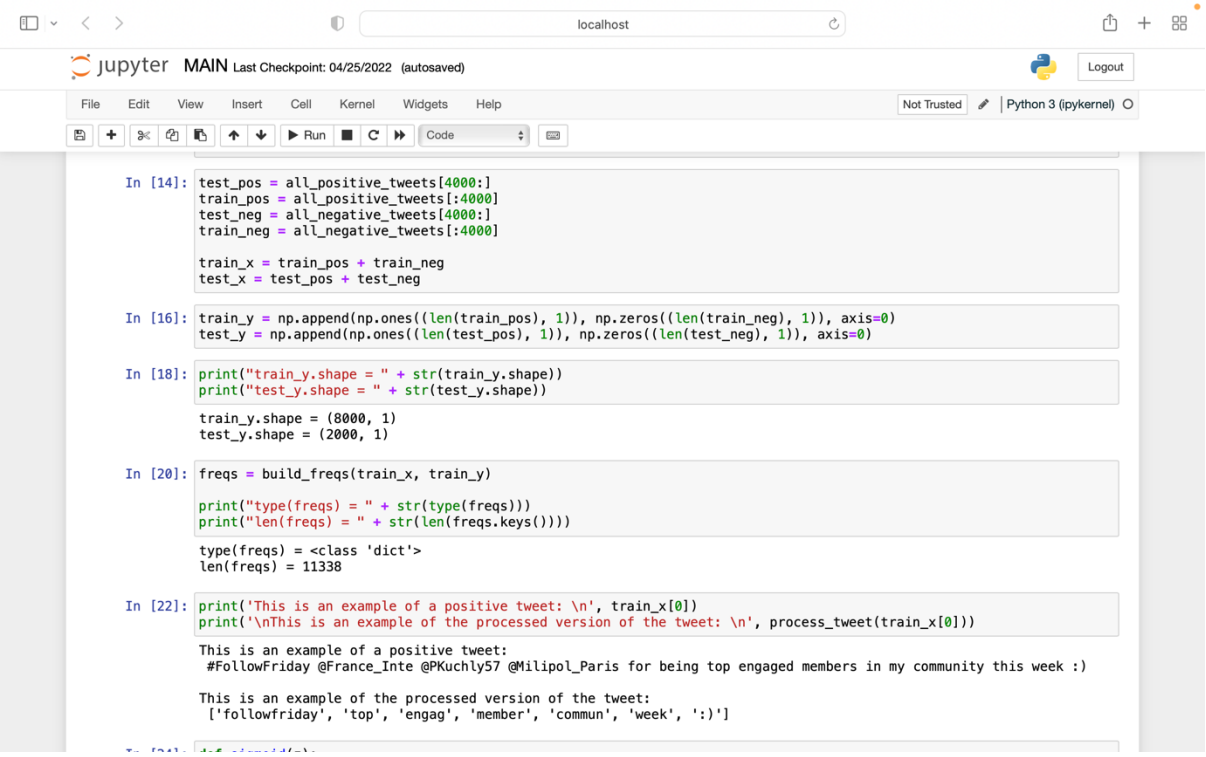
utils.py

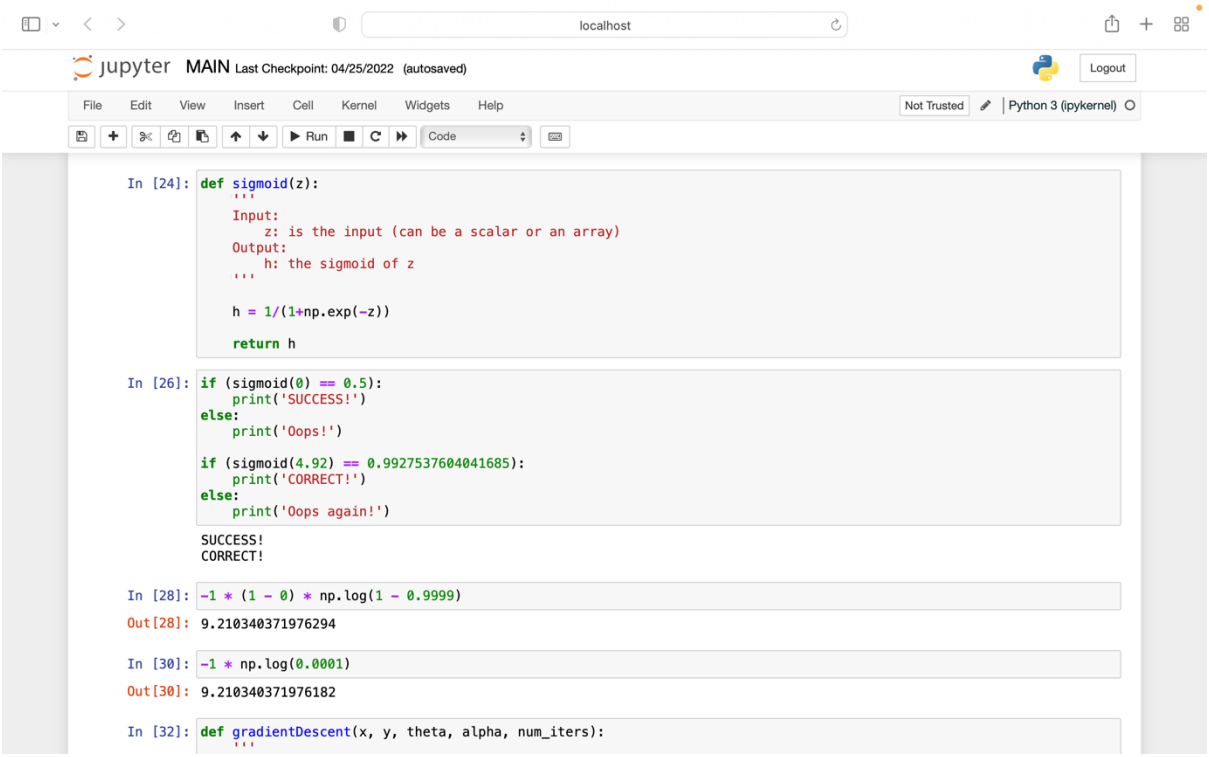
```
import re
import string
import numpy as np

from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import TweetTokenizer


def process_tweet(tweet):
    stemmer = PorterStemmer()
    stopwords_english = stopwords.words('english')

    tweet = re.sub(r'\$\w*', '', tweet)

    tweet = re.sub(r'^RT[\s]+', '', tweet)

    tweet = re.sub(r'https?:\/\/.*[\r\n]*', '', tweet)

    tweet = re.sub(r'#', '', tweet)

    tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True,
                               reduce_len=True)
    tweet_tokens = tokenizer.tokenize(tweet)

    tweets_clean = []
    for word in tweet_tokens:
        if (word not in stopwords_english and
                word not in string.punctuation):

            stem_word = stemmer.stem(word)
            tweets_clean.append(stem_word)

    return tweets_clean

def build_freqs(tweets, ys):

    yslist = np.squeeze(ys).tolist()

    for y, tweet in zip(yslist, tweets):
        for word in process_tweet(tweet):
            pair = (word, y)
            if pair in freqs:
                freqs[pair] += 1
            else:
                freqs[pair] = 1
    return freqs
```

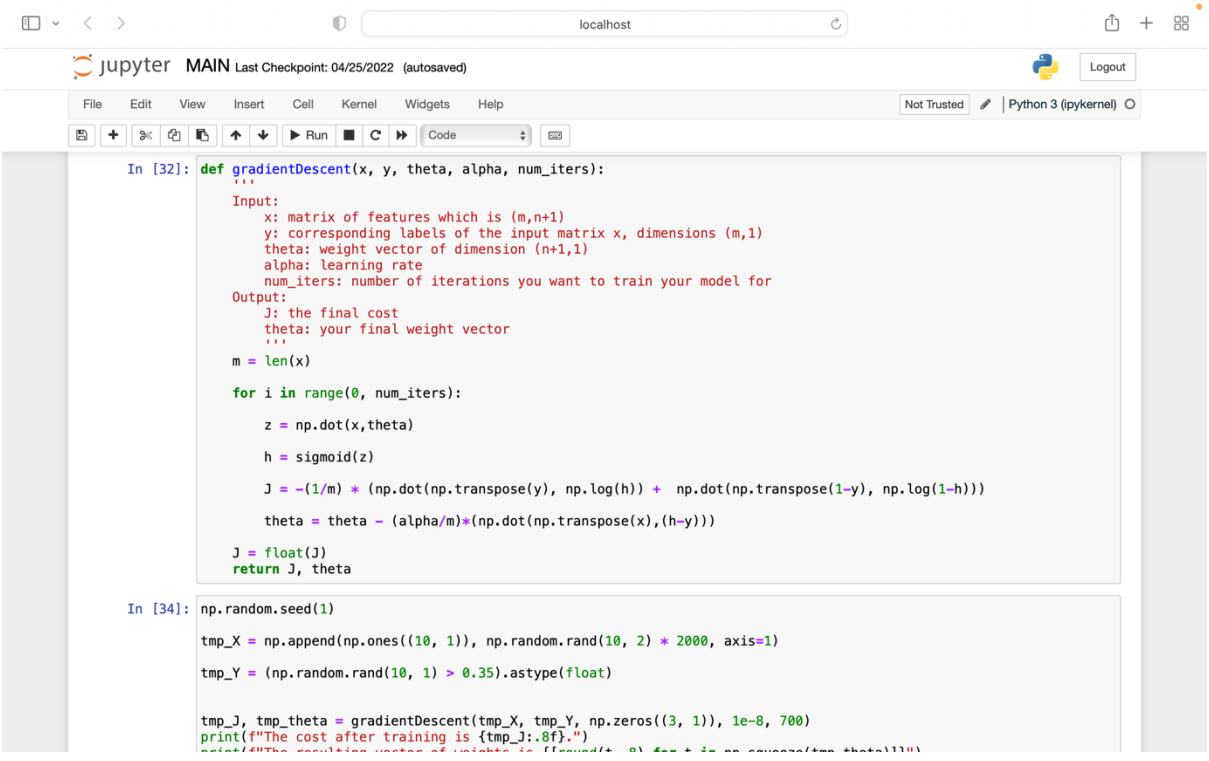
18. SAMPLE OUTPUT

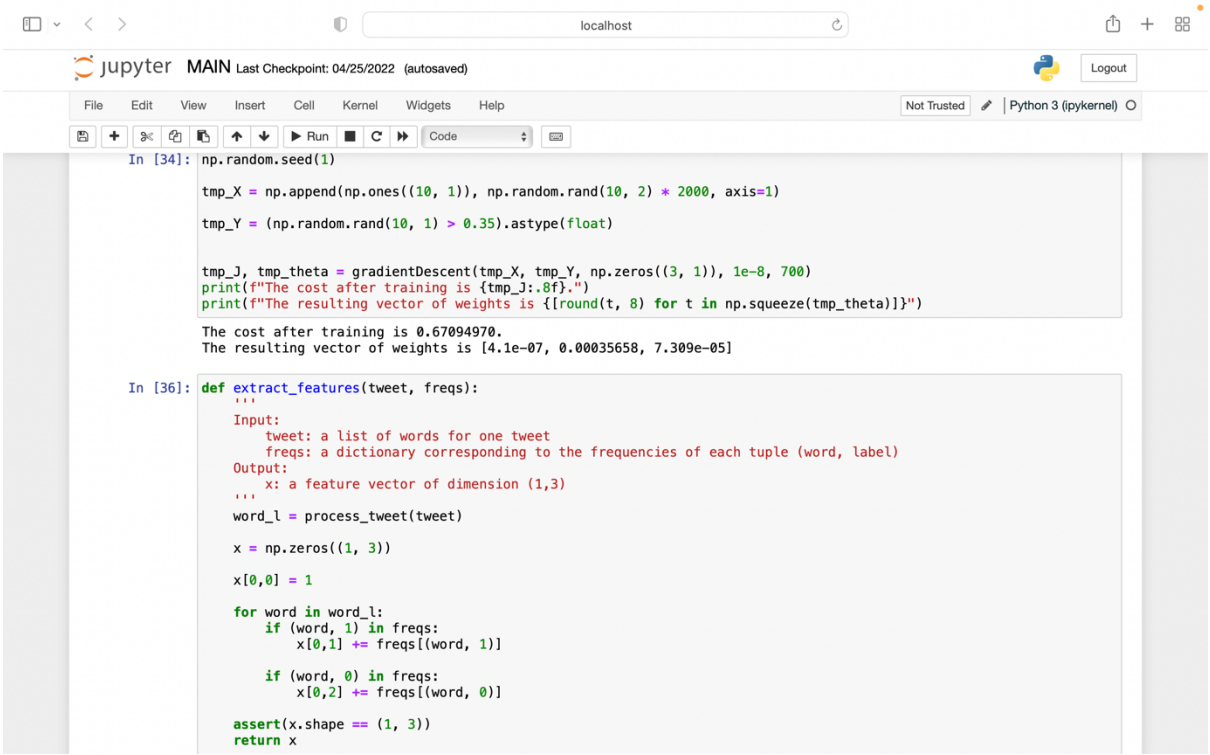
main.ipynb

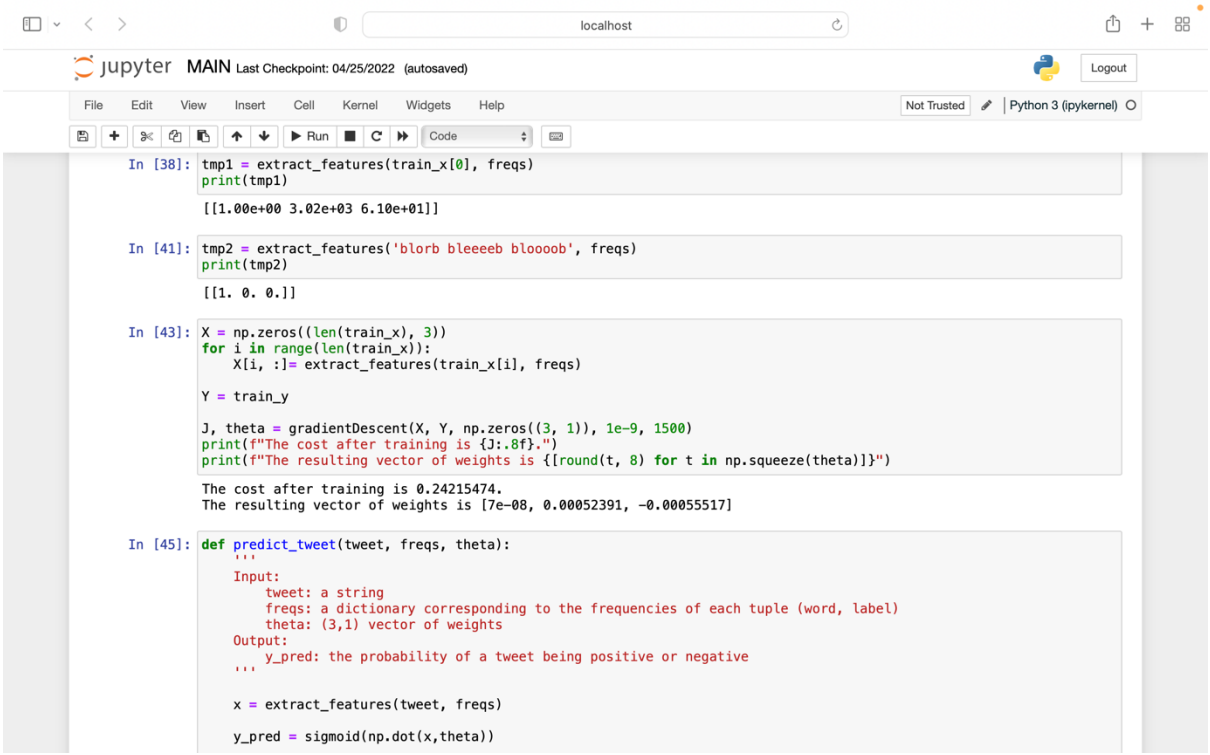


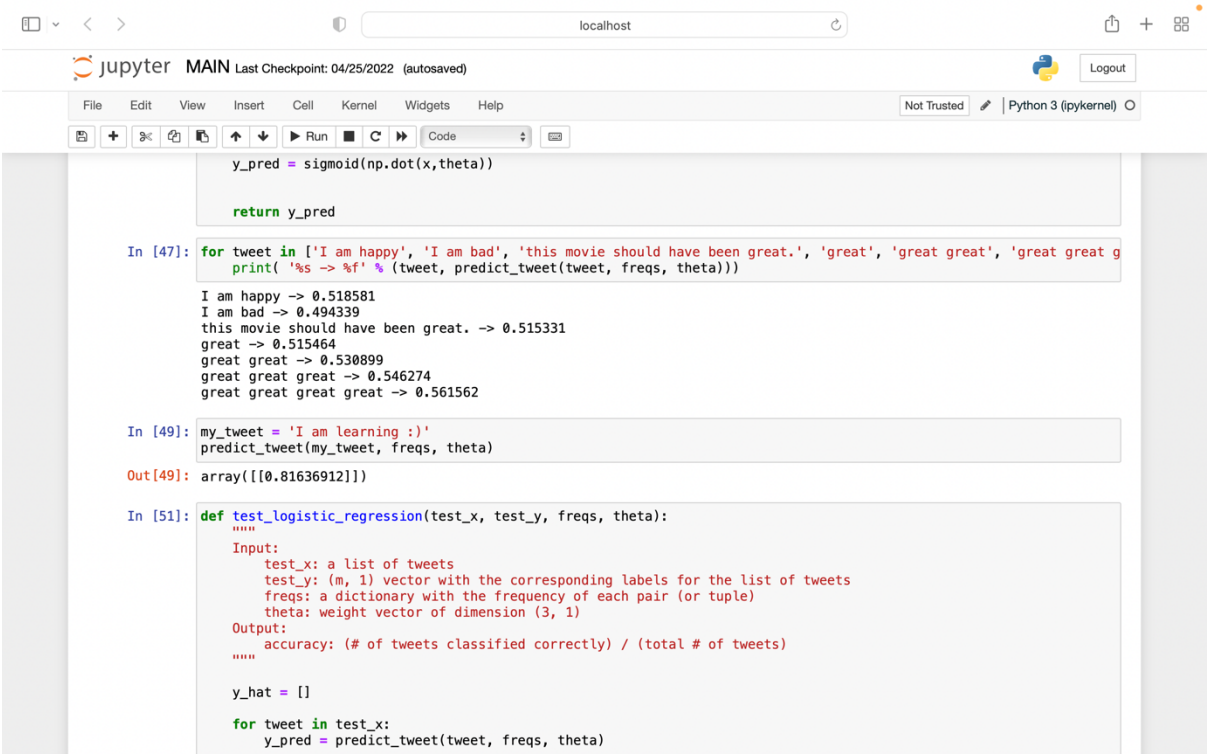



localhost
jupyter MAIN Last Checkpoint: 04/25/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [24]:
def sigmoid(z):
'''
Input:
z: is the input (can be a scalar or an array)
Output:
h: the sigmoid of z
'''
h = 1/(1+np.exp(-z))
return h
In [26]:
if (sigmoid(0) == 0.5):
print('SUCCESS!')
else:
print('Oops!')
if (sigmoid(4.92) == 0.9927537604041685):
print('CORRECT!')
else:
print('Oops again!')
SUCCESS!
CORRECT!
In [28]:
-1 * (1 - 0) * np.log(1 - 0.9999)
Out[28]:
9.210340371976294
In [30]:
-1 * np.log(0.0001)
Out[30]:
9.210340371976182
In [32]:
def gradientDescent(x, y, theta, alpha, num_iters):
'''


localhost
jupyter MAIN Last Checkpoint: 04/25/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [32]:
def gradientDescent(x, y, theta, alpha, num_iters):
'''
Input:
x: matrix of features which is (m,n+1)
y: corresponding labels of the input matrix x, dimensions (m,1)
theta: weight vector of dimension (n+1,1)
alpha: learning rate
num_iters: number of iterations you want to train your model for
Output:
J: the final cost
theta: your final weight vector
'''
m = len(x)
for i in range(0, num_iters):
z = np.dot(x,theta)
h = sigmoid(z)
J = -(1/m) * (np.dot(np.transpose(y), np.log(h)) + np.dot(np.transpose(1-y), np.log(1-h)))
theta = theta - (alpha/m)*(np.dot(np.transpose(x),(h-y)))
J = float(J)
return J, theta
In [34]:
np.random.seed(1)
tmp_X = np.append(np.ones((10, 1)), np.random.rand(10, 2) * 2000, axis=1)
tmp_Y = (np.random.rand(10, 1) > 0.35).astype(float)
tmp_J, tmp_theta = gradientDescent(tmp_X, tmp_Y, np.zeros((3, 1)), 1e-8, 700)
print(f"The cost after training is {tmp_J:.8f}.")


localhost
jupyter MAIN Last Checkpoint: 04/25/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [34]:
np.random.seed(1)
tmp_X = np.append(np.ones((10, 1)), np.random.rand(10, 2) * 2000, axis=1)
tmp_Y = (np.random.rand(10, 1) > 0.35).astype(float)
tmp_J, tmp_theta = gradientDescent(tmp_X, tmp_Y, np.zeros((3, 1)), 1e-8, 700)
print(f"The cost after training is {tmp_J:.8f}.")
print(f"The resulting vector of weights is {[round(t, 8) for t in np.squeeze(tmp_theta)]}")
The cost after training is 0.67094970.
The resulting vector of weights is [4.1e-07, 0.00035658, 7.309e-05]
In [36]:
def extract_features(tweet, freqs):
'''
Input:
tweet: a list of words for one tweet
freqs: a dictionary corresponding to the frequencies of each tuple (word, label)
Output:
x: a feature vector of dimension (1,3)
'''
word_l = process_tweet(tweet)
x = np.zeros((1, 3))
x[0,0] = 1
for word in word_l:
if (word, 1) in freqs:
x[0,1] += freqs[(word, 1)]
if (word, 0) in freqs:
x[0,2] += freqs[(word, 0)]
assert(x.shape == (1, 3))
return x


localhost
jupyter MAIN Last Checkpoint: 04/25/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [38]:
tmp1 = extract_features(train_x[0], freqs)
print(tmp1)
[[1.00e+00 3.02e+03 6.10e+01]]
In [41]:
tmp2 = extract_features('blorb bleeeeb bloooob', freqs)
print(tmp2)
[[1. 0. 0.]]
In [43]:
X = np.zeros((len(train_x), 3))
for i in range(len(train_x)):
X[i, :]= extract_features(train_x[i], freqs)
Y = train_y
J, theta = gradientDescent(X, Y, np.zeros((3, 1)), 1e-9, 1500)
print(f"The cost after training is {J:.8f}.")
print(f"The resulting vector of weights is {[round(t, 8) for t in np.squeeze(theta)]}")
The cost after training is 0.24215474.
The resulting vector of weights is [7e-08, 0.00052391, -0.00055517]
In [45]:
def predict_tweet(tweet, freqs, theta):
'''
Input:
tweet: a string
freqs: a dictionary corresponding to the frequencies of each tuple (word, label)
theta: (3,1) vector of weights
Output:
y_pred: the probability of a tweet being positive or negative
'''
x = extract_features(tweet, freqs)
y_pred = sigmoid(np.dot(x,theta))




```
    y_pred = sigmoid(np.dot(x,theta))

    return y_pred
```

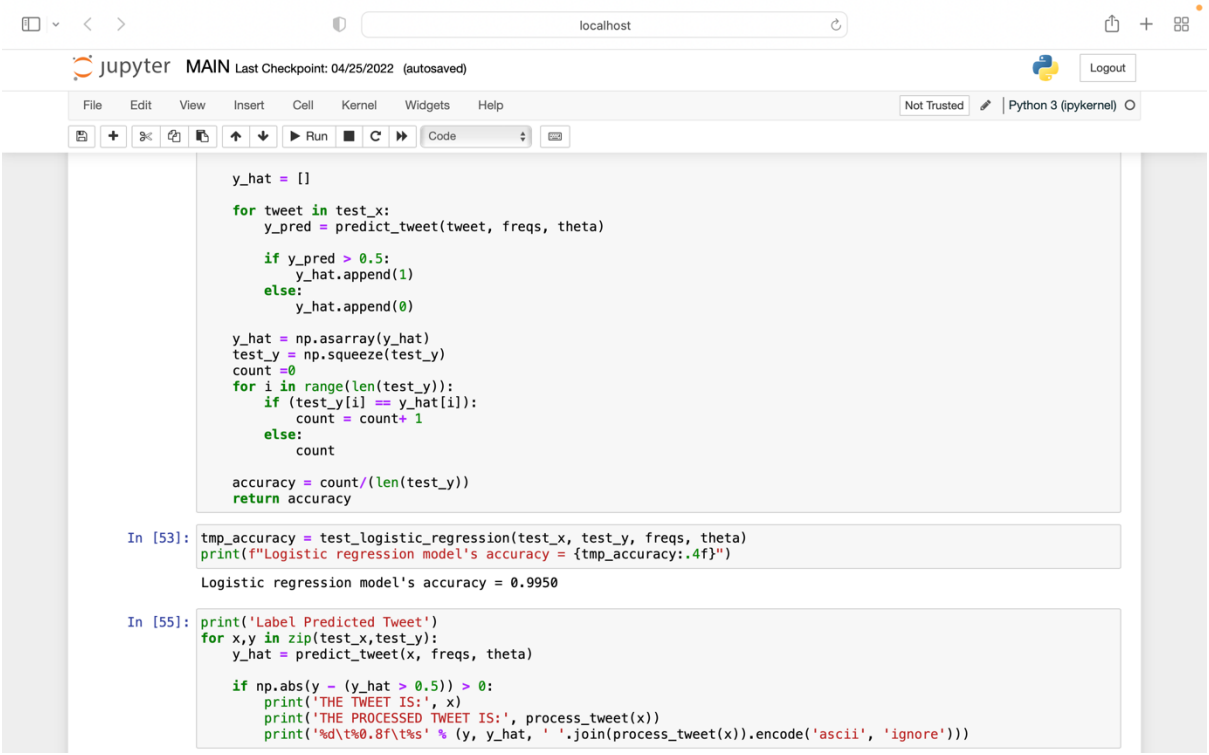
In [47]:
```
for tweet in ['I am happy', 'I am bad', 'this movie should have been great.', 'great', 'great great', 'great great g
    print( '%s -> %f' % (tweet, predict_tweet(tweet, freqs, theta)))
```

```
I am happy -> 0.518581
I am bad -> 0.494339
this movie should have been great. -> 0.515331
great -> 0.515464
great great -> 0.530899
great great great -> 0.546274
great great great great -> 0.561562
```

In [49]:
```
my_tweet = 'I am learning :)'
predict_tweet(my_tweet, freqs, theta)
```

Out[49]:
```
array([[0.81636912]])
```

In [51]:
```
def test_logistic_regression(test_x, test_y, freqs, theta):
    """
    Input:
        test_x: a list of tweets
        test_y: (m, 1) vector with the corresponding labels for the list of tweets
        freqs: a dictionary with the frequency of each pair (or tuple)
        theta: weight vector of dimension (3, 1)
    Output:
        accuracy: (# of tweets classified correctly) / (total # of tweets)
    """

    y_hat = []

    for tweet in test_x:
        y_pred = predict_tweet(tweet, freqs, theta)
```

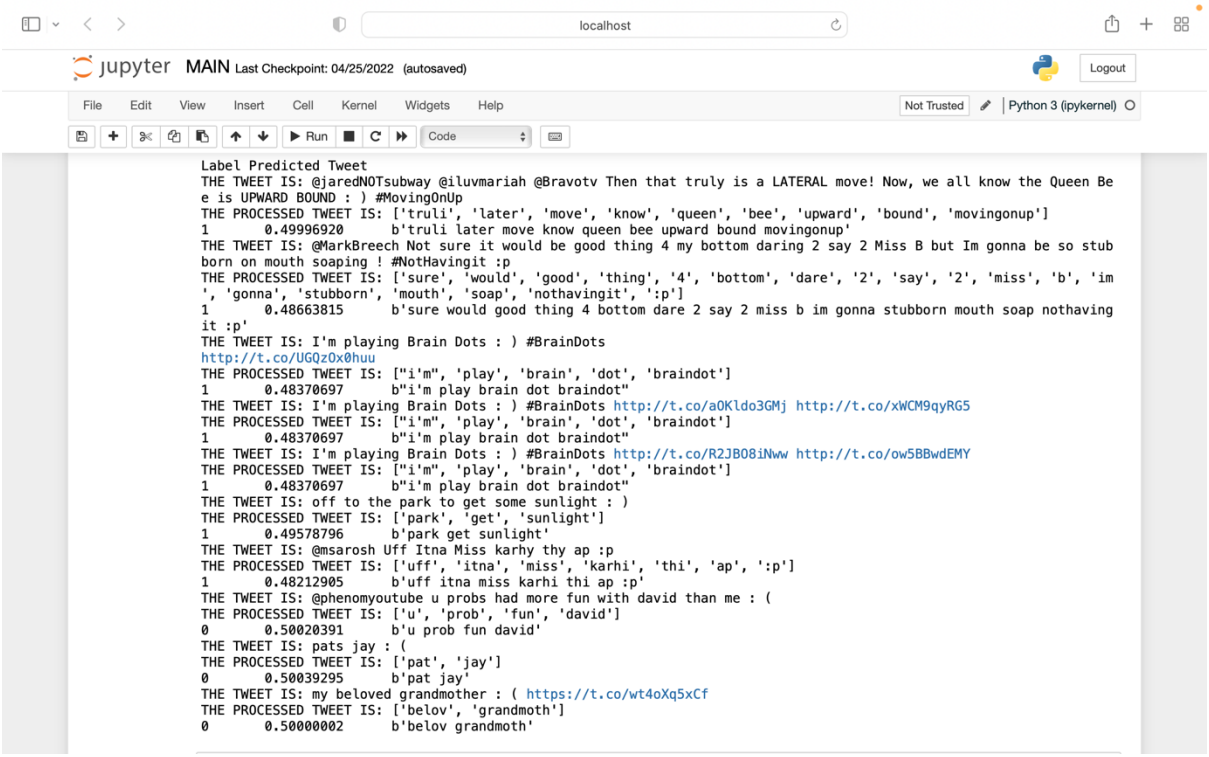



```
    y_hat = []

    for tweet in test_x:
        y_pred = predict_tweet(tweet, freqs, theta)

        if y_pred > 0.5:
            y_hat.append(1)
        else:
            y_hat.append(0)

    y_hat = np.asarray(y_hat)
    test_y = np.squeeze(test_y)
    count =0
    for i in range(len(test_y)):
        if (test_y[i] == y_hat[i]):
            count = count+ 1
        else:
            count

    accuracy = count/(len(test_y))
    return accuracy
```

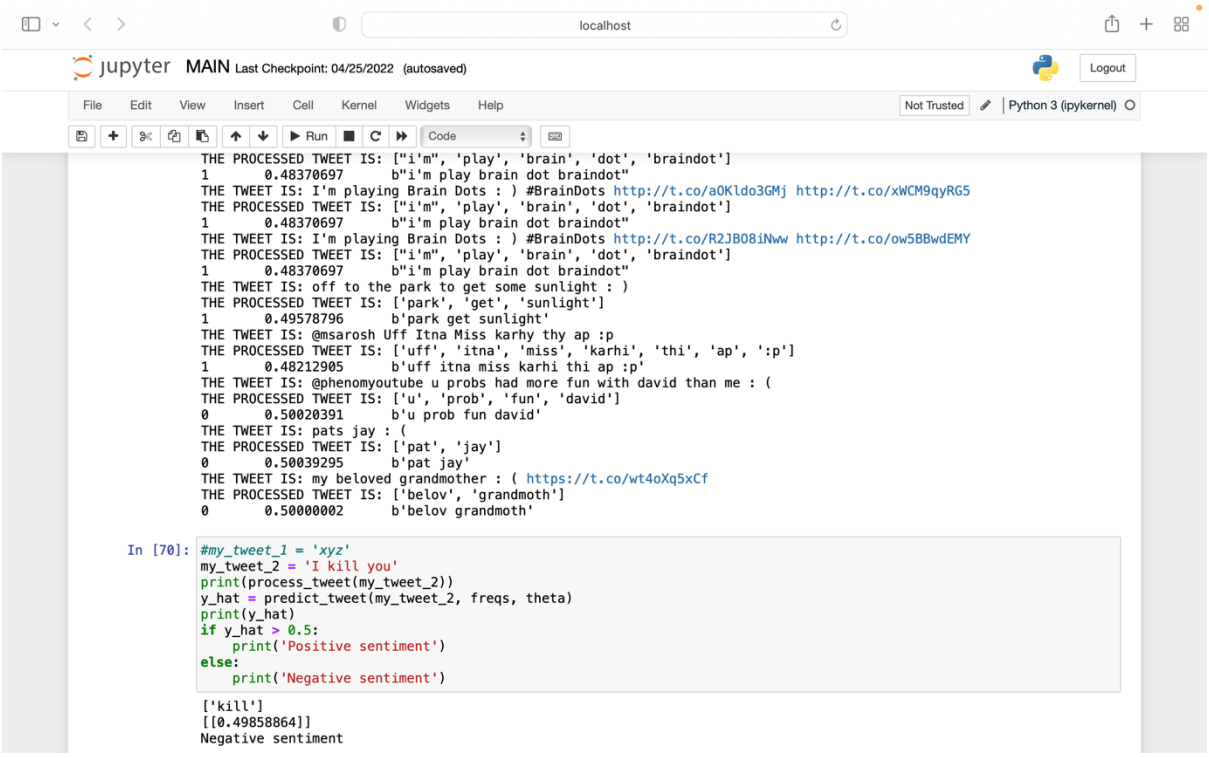
In [53]:
```
tmp_accuracy = test_logistic_regression(test_x, test_y, freqs, theta)
print(f"Logistic regression model's accuracy = {tmp_accuracy:.4f}")
```

```
Logistic regression model's accuracy = 0.9950
```

In [55]:
```
print('Label Predicted Tweet')
for x,y in zip(test_x,test_y):
    y_hat = predict_tweet(x, freqs, theta)

    if np.abs(y - (y_hat > 0.5)) > 0:
        print('THE TWEET IS:', x)
        print('THE PROCESSED TWEET IS:', process_tweet(x))
        print('%d\t%0.8f\t%s' % (y, y_hat, ' '.join(process_tweet(x)).encode('ascii', 'ignore')))
```




```
Label Predicted Tweet
THE TWEET IS: @jaredNOTsubway @iluvmariah @Bravotv Then that truly is a LATERAL move! Now, we all know the Queen Be
e is UPWARD BOUND : ) #MovingOnUp
THE PROCESSED TWEET IS: ['truli', 'later', 'move', 'know', 'queen', 'bee', 'upward', 'bound', 'movingonup']
1       0.49996920      b'truli later move know queen bee upward bound movingonup'
THE TWEET IS: @MarkBreech Not sure it would be good thing 4 my bottom daring 2 say 2 Miss B but Im gonna be so stub
born on mouth soaping ! #NotHavingit :p
THE PROCESSED TWEET IS: ['sure', 'would', 'good', 'thing', '4', 'bottom', 'dare', '2', 'say', '2', 'miss', 'b', 'im
', 'gonna', 'stubborn', 'mouth', 'soap', 'nothavingit', ':p']
1       0.48663815      b'sure would good thing 4 bottom dare 2 say 2 miss b im gonna stubborn mouth soap nothaving
it :p'
THE TWEET IS: I'm playing Brain Dots : ) #BrainDots
http://t.co/UGQzOx0huu
THE PROCESSED TWEET IS: ["i'm", 'play', 'brain', 'dot', 'braindot']
1       0.48370697      b"i'm play brain dot braindot"
THE TWEET IS: I'm playing Brain Dots : ) #BrainDots http://t.co/aOKldo3GMj http://t.co/xWCM9qyRG5
THE PROCESSED TWEET IS: ["i'm", 'play', 'brain', 'dot', 'braindot']
1       0.48370697      b"i'm play brain dot braindot"
THE TWEET IS: I'm playing Brain Dots : ) #BrainDots http://t.co/R2JBO8iNww http://t.co/ow5BBwdEMY
THE PROCESSED TWEET IS: ["i'm", 'play', 'brain', 'dot', 'braindot']
1       0.48370697      b"i'm play brain dot braindot"
THE TWEET IS: off to the park to get some sunlight : )
THE PROCESSED TWEET IS: ['park', 'get', 'sunlight']
1       0.49578796      b'park get sunlight'
THE TWEET IS: @msarosh Uff Itna Miss karhy thy ap :p
THE PROCESSED TWEET IS: ['uff', 'itna', 'miss', 'karhi', 'thi', 'ap', ':p']
1       0.48212905      b'uff itna miss karhi thi ap :p'
THE TWEET IS: @phenomyoutube u probs had more fun with david than me : (
THE PROCESSED TWEET IS: ['u', 'prob', 'fun', 'david']
0       0.50020391      b'u prob fun david'
THE TWEET IS: pats jay : (
THE PROCESSED TWEET IS: ['pat', 'jay']
0       0.50039295      b'pat jay'
THE TWEET IS: my beloved grandmother : ( https://t.co/wt4oXq5xCf
THE PROCESSED TWEET IS: ['belov', 'grandmoth']
0       0.50000002      b'belov grandmoth'
```

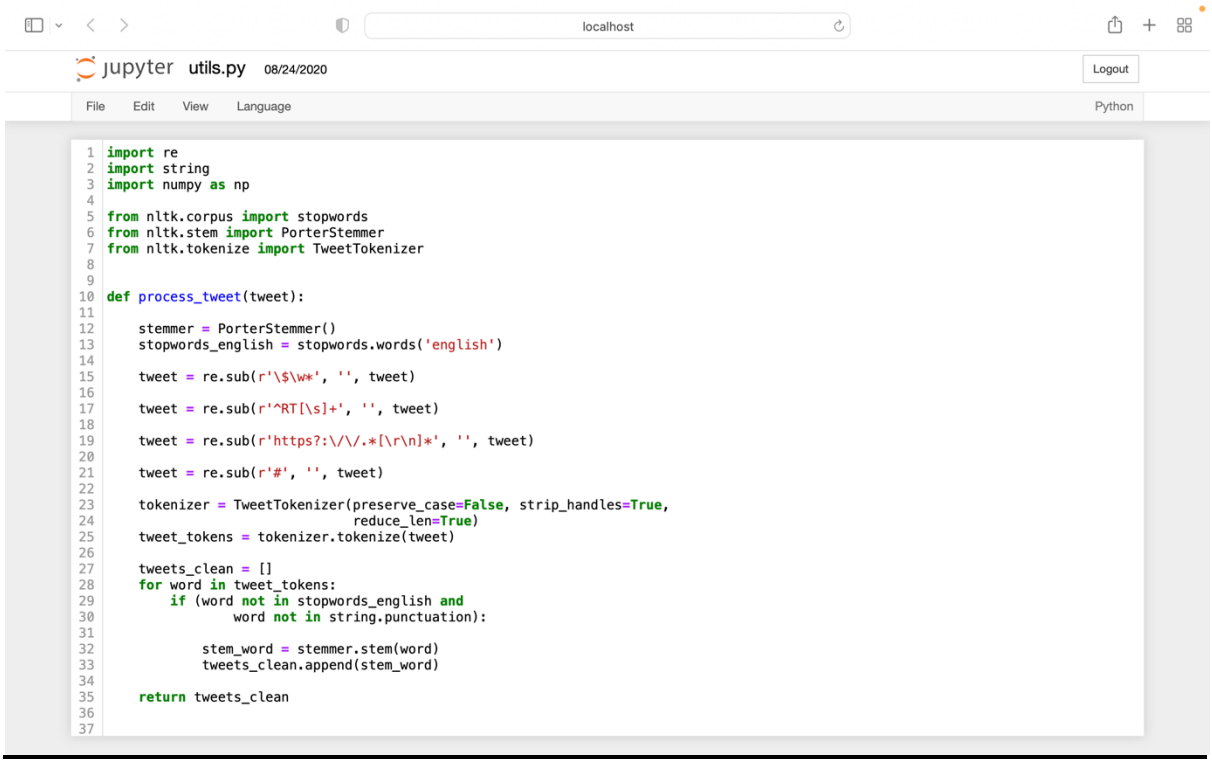



```
THE PROCESSED TWEET IS: ["i'm", 'play', 'brain', 'dot', 'braindot']
1       0.48370697      b"i'm play brain dot braindot"
THE TWEET IS: I'm playing Brain Dots : ) #BrainDots http://t.co/aOKldo3GMj http://t.co/xWCM9qyRG5
THE PROCESSED TWEET IS: ["i'm", 'play', 'brain', 'dot', 'braindot']
1       0.48370697      b"i'm play brain dot braindot"
THE TWEET IS: I'm playing Brain Dots : ) #BrainDots http://t.co/R2JBO8iNww http://t.co/ow5BBwdEMY
THE PROCESSED TWEET IS: ["i'm", 'play', 'brain', 'dot', 'braindot']
1       0.48370697      b"i'm play brain dot braindot"
THE TWEET IS: off to the park to get some sunlight : )
THE PROCESSED TWEET IS: ['park', 'get', 'sunlight']
1       0.49578796      b'park get sunlight'
THE TWEET IS: @msarosh Uff Itna Miss karhy thy ap :p
THE PROCESSED TWEET IS: ['uff', 'itna', 'miss', 'karhi', 'thi', 'ap', ':p']
1       0.48212905      b'uff itna miss karhi thi ap :p'
THE TWEET IS: @phenomyoutube u probs had more fun with david than me : (
THE PROCESSED TWEET IS: ['u', 'prob', 'fun', 'david']
0       0.50020391      b'u prob fun david'
THE TWEET IS: pats jay : (
THE PROCESSED TWEET IS: ['pat', 'jay']
0       0.50039295      b'pat jay'
THE TWEET IS: my beloved grandmother : ( https://t.co/wt4oXq5xCf
THE PROCESSED TWEET IS: ['belov', 'grandmoth']
0       0.50000002      b'belov grandmoth'
```

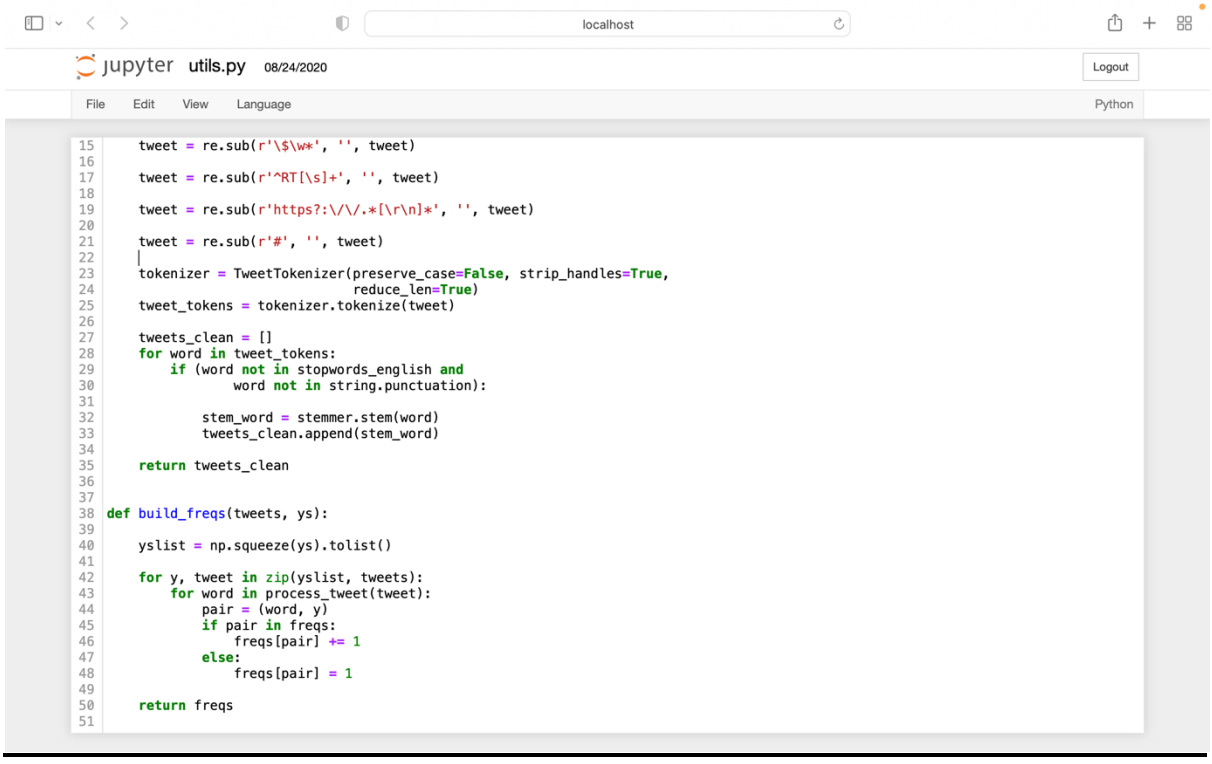
In [70]:
```
#my_tweet_1 = 'xyz'
my_tweet_2 = 'I kill you'
print(process_tweet(my_tweet_2))
y_hat = predict_tweet(my_tweet_2, freqs, theta)
print(y_hat)
if y_hat > 0.5:
    print('Positive sentiment')
else:
    print('Negative sentiment')
```

```
['kill']
[[0.49858864]]
Negative sentiment
```

utils.py



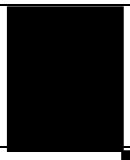
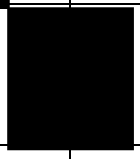

```
import re
import string
import numpy as np

from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import TweetTokenizer


def process_tweet(tweet):

    stemmer = PorterStemmer()
    stopwords_english = stopwords.words('english')

    tweet = re.sub(r'\$\w*', '', tweet)

    tweet = re.sub(r'^RT[\s]+', '', tweet)

    tweet = re.sub(r'https?:\/\/.*[\r\n]*', '', tweet)

    tweet = re.sub(r'#', '', tweet)

    tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True,
                               reduce_len=True)
    tweet_tokens = tokenizer.tokenize(tweet)

    tweets_clean = []
    for word in tweet_tokens:
        if (word not in stopwords_english and
                word not in string.punctuation):

            stem_word = stemmer.stem(word)
            tweets_clean.append(stem_word)

    return tweets_clean

```




```
    tweet = re.sub(r'\$\w*', '', tweet)

    tweet = re.sub(r'^RT[\s]+', '', tweet)

    tweet = re.sub(r'https?:\/\/.*[\r\n]*', '', tweet)

    tweet = re.sub(r'#', '', tweet)

    tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True,
                               reduce_len=True)
    tweet_tokens = tokenizer.tokenize(tweet)

    tweets_clean = []
    for word in tweet_tokens:
        if (word not in stopwords_english and
                word not in string.punctuation):

            stem_word = stemmer.stem(word)
            tweets_clean.append(stem_word)

    return tweets_clean


def build_freqs(tweets, ys):

    yslist = np.squeeze(ys).tolist()

    for y, tweet in zip(yslist, tweets):
        for word in process_tweet(tweet):
            pair = (word, y)
            if pair in freqs:
                freqs[pair] += 1
            else:
                freqs[pair] = 1

    return freqs

```

19. PROJECT SCHEDULING

An elementary timeline chart for the development of the plan is given below:

	Feb	Mar	Apr	May
Requirement Gathering				
Analysis				
Coding				
Implementation				
	W_2 W_3 W_4	W_1 W_2 W_3 W_4	W_2 W_3 W_4	W_1

Here, $W_{i's}$ represents the week of the months.

20. LIMITATION

On the surface, sentiment analysis appears to be nothing more than the process of classifying text as positive, negative, or neutral. However, there are numerous limitations to that process. Here is a list of some of them.

Context:

Anything said sometimes, by someone, by someone, by someone, etc. In context, what I mean is something Analysing feelings without context won't help you find the exact feeling in any text. Unlike humans, however, unless specifically specified, machines can detect or interpret contexts

Irony and Sarcasm:

When it comes to irony and sarcasm, people often express positive feelings with negative words. This can make classifying such textual data much harder for machines.

Negation Detection:

Negation is a way to change the true meaning of words, phrases or even phrases. In order to identify the source of the negation, users use various language approaches, but the range of words affected by the negation also needs to be analysed.

Example: "The movie was not interesting". In this sentence, negation occurs before the words interesting and only affects one word.

Word Ambiguity:

Ambiguity in words is another obstacle to the analysis of feelings. Here, the polarity of words is hard to analyse, since they depend heavily on the context of the sentence. Lexicons are created as a popular way of overcoming this hurdle. Although word polarity differs greatly in different fields, a universal lexicon for the analysis of feelings cannot be developed.

A tool like Bytes view that can easily dissect and analyse unstructured information can give you insights that are not affected by such limitations in order to avoid these limitations.

21. CONCLUSION

Sentiment analysis is used to identifying people's opinion, attitude and emotional states. The views of the people can be positive or negative. Commonly, parts of speech are used as feature to extract the sentiment of the text. An adjective plays a crucial role in identifying sentiment from parts of speech. Sometimes words having adjective and adverb are used together then it is difficult to identify sentiment and opinion.

To do the sentiment analysis of tweets, the proposed system first extracts the twitter posts from twitter by user. The system can also computes the frequency of each term in tweet. Using machine learning supervised approach help to obtain the results.

Twitter is large source of data, which make it more attractive for performing sentiment analysis. We perform analysis on around 15,000 tweets total for each party, so that we analyse the results, understand the patterns and give a review on people opinion. We saw different party have different sentiment results according to their progress and working procedure. We also saw how any social event, speech or rally cause a fluctuation in sentiment of people. We also get to know which policies are getting more support from people which are started by any of these parties. It was shown that BJP is more successful political part in present time based on people opinion. It is not necessary that our classifier can only be used for political parties. It is general classifier. It can be used for any purpose based on tweets we collect with the help of keyword. It can be used for finance, marketing, reviewing and many more

22. MILESTONE

S.No.	Project Activity	Estimated Start Date	Estimated End Date
1.	Synopsis submission	16-03-2022	24-03-2022
2.	Project Completion	25-02-2022	05-05-2022
3.	Report Submission	28-04-2022	14-05-2022

23. MEETING WITH THE SUPERVISOR

Date of the meet	Mode	Comments by the supervisor	Signature of the Supervisor
3rd February, 2022	Google Meet		
3rd March, 2022	Offline		
21st March, 2022	Offline		
23rd March, 2022	Offline		
25rd April, 2022	Offline		
10th May, 2022	Offline		

24. BIBLIOGRAPHY & REFERENCES

8. Classifying Text-Based Emotions using logistic regression: Fahad Mazaed Alotaibi Faculty of Computing and Information Technology in Rabigh (FCITR) King Abdul Aziz University (KAU) Jeddah Saudi Arabia.
9. Sentiment analysis system for movie review in Bahasa Indonesia using naive bayes classifier method To cite this article: Yanuar Nurdiansyah et al 2018 J. Phys.: Conf. Ser. 1008 012011
10. Article Sentiment Analysis Based Requirement Evolution Prediction Lingling Zhao 1 and Anping Zhao 2
11. Speech and Language Processing (3rd ed. draft) Dan Jurafsky and James H. Martin https://web.stanford.edu/~jurafsky/slp3/
12. Sentiment Analysis Using Multinomial Logistic Regression Ramadhan WP1 , Astri Novianty S.T.,M.T2 ,Casi Setianingsih S.T.,M.T3 Department of Computer Engineering, Telkom University Bandung, Indonesia
13. Sentiment analysis: What is the end user's requirement? Article · June 2012 DOI: 10.1145/2254129.2254173
14. Sentiment Analysis: What is the End User's Requirement? Amitava Das Department of Computer and Engineering Jadavpur University Kolkata-700032, India Sivaji Bandyopadhyay Department of Computer and Engineering Jadavpur University Kolkata-700032, India Björn Gambäck IDI, Norwegian University of Science and Technology (NTNU) Trondheim, Norway
15. Degree Project In Technology, First Sycle, 15 Credits Stockholm, Sweden, Sentiment classification on Amazon reviews using machine learning approaches Sepideh Paknehad
16. International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611 Volume 4 Issue 12, December 2015 www.ijsr.net Licensed Under Creative Commons Attribution CC BY A Study on Sentiment Analysis: Methods and Tools Abhishek Kaushik1 , Anchal Kaushik2 , Sudhanshu Naithani3
17. International Journal of Engineering Research and Modern Education (IJERME) ISSN (Online): 2455 - 4200 (www.rdmodernresearch.com) Volume I, Issue I, 2016 AUTOMATIC SENTIMENT ANALYSIS OF USER REVIEWS B. Kasthuri & A. Anitha
18. Using Machine Learning Techniques for Sentiment Analysis O`scarRomeroLlombart
19. Sentiment Analysis of Product-Based Reviews Using Machine Learning Approaches By Anusuya Dhara (CSE/2014/041) Arkadeb Saha (CSE/2014/048) Sourish Sen Gupta (CSE/2014/049) Pranit Bose (CSE/2014/060)