SENTIMENTAL ANALYSIS
UISNG ML

A mini-project report submitted to the
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

Submitted By
Ramya Achanta (17071A1261)
Vipparthy Niharika (17071A12B6)

Under the Guidance Of
Dr. B.V. SESHU KUMARI
(Associate Professor, IT dept, VNRVJIET)

DEPARTMENT OF INFORMATION TECHNOLOGY

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institute, NAAC Accredited With ‘A++’ Grade, NBA Accredited, Approved by AICTE, New Delhi, Affiliated to JNTUH)
MAY 2020

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute)

Hyderabad-500090

TAMASOMA JYOTHIRGAMAYA

CERTIFICATE

This is to certify that Ramya Achanta (17071A1261) and Vipparthy Niharika (17071A12B6) have successfully completed their project work at IT Department of VNR VJIET, Hyderabad entitled **SENTIMENTAL ANALYSIS - USING ML,** in partial fulfilment of the requirements for the award of B. Tech degree during the academic year 2019-2020.

Project Guide	**Head of the department**
Dr. B.V. SESHU KUMARI	***Dr. G. Suresh Reddy***
Associate Professor	***Head of the Department***
Department of IT	***Department of IT***
VNRVJIET	***VNRVJIET***

DECLARATION

This is to certify that the project work entitled **" SENTIMENTAL ANALYSIS USING ML"** submitted in VNR Vignana Jyothi Institute of Engineering & Technology in partial fulfilment of requirement for the award of Bachelor of Technology in the department of Information Technology, is a bonafide report of the work carried out by us under the guidance and supervision of Dr. B.V. Seshu Kumari (Associate Professor), Department of IT, VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any university or institution for the award of any degree or diploma.

Ramya Achanta	**Vipparthy Niharika**
(17071A1261)	(17071A12B6)
III B.Tech-IT	III B.Tech-IT
VNR VJIET	VNR VJIET

Project Guide
Dr. B.V. SESHU KUMARI
Associate Professor
Department of IT
VNRVJIET

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who activated it, without it would ever never have come into existence. To them we lay the words of gratitude imprinting within us.

We are indebted to our venerable principal **Dr. C. D. Naidu** for this unflinching devotion, which lead us to complete this project. The support, encouragement given by him and his motivation lead us to complete this project.

We express our thanks to internal guide **Dr. B.V. Seshu Kumari** and also Head of the department **Dr. G. Suresh Reddy** for having provided us a lot of facilities to undertake the project work and guide us to complete the project.

We express our sincere thanks to our faculty of the department of **Information Technology** and the remaining members of our college **VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY** who extended their valuable support in helping us to complete the project in time.

Ramya Achanta	Vipparthy Niharika
(17071A1261)	(17071A12B6)

Principal
Dr. C. D. Naidu

Head of the department
Dr. G. Suresh Reddy
Head of the Department (IT)
VNR VJIET

Project Guide
Dr. B.V. SESHU KUMARI
Associate Professor
Department of IT
VNRVJIET

ABSTRACT

The field of sentiment analysis is an exciting new research direction due to large number of real-world applications where discovering people’s opinion is important in better decision-making. The development of techniques for the document-level sentiment analysis is one of the significant components of this area. Recently, people have started expressing their opinions on the Web that increased the need of analysing the opinionated online content for various real-world applications. A lot of research is present in literature for detecting sentiment from the text. Still, there is a huge scope of improvement of these existing sentiment analysis models. Existing sentiment analysis models can be improved further with more semantic and common-sense knowledge.

Sentiment analysis systems are being applied in almost every business and social domain because opinions are central to almost all human activities and are key influencers of our behaviours. Our beliefs and perceptions of reality, and the choices we make, are largely conditioned on how others see and evaluate the world. For this reason, when we need to make a decision, we often seek out the opinions of others. This is true not only for individuals but also for organizations.

INDEX

Contents Page.No

CHAPTER 1: INTRODUCTION 9

1.1 Introduction to Machine Learning 9

1.2 Machine Learning Models 9

1.3 Classification 10

1.4 Proposed System 11

CHAPTER 2: FEASIBILITY STUDY 12

2.1Technical Feasibility 12

2.2 Economic Feasibility 12

2.3 Legal Feasibility 13

2.4 Operational Feasibility 13

2.5 Scheduling Feasibility 13

CHAPTER 3: LITERATURE SURVEY 14

3.1 Python API’s & Libraries required 14

3.1.1Matplotlib 14

3.1.2 Scikit-learn 14

3.1.3 Numpy 14

3.1.4 Pandas 14

3.1.5 Pickle 15

3.1.6 Re 15

3.1.7 Nltk 15

3.2 Python IDE 15

CHAPTER 4: ALGORITHM DESCRIPTION 16

4.1 Regression 16

4.2 Logistic Regression 16

4.2.1 Types of Logistic Regression 16

4.2.2 Logistic Regression Assumptions 16

4.2.3 Regression Models 17

4.3 Advantages 17

4.4 Disadvantage 18

4.5 Applications 19

4.6 Shortcoming of Train-test-split 20

CHAPTER 5: SYSTEM ANALYSIS 21

5 System Requirements 21

5.1 Software Requirements 21

5.2 Hardware Requirements 22

CHAPTER 6: SOFTWARE DESIGN 23

6.1 UML Diagrams 24

6.1.1 Use-Case Diagram 25

6.1.2 Class Diagram 26

6.1.3 Sequence Diagram 28

6.1.4 Activity Diagram 29

CHAPTER 7: IMPLEMENTATION 34

7.1 Code for loading dataset and Importing Libraries 34

7.2 Data Ratio 35

7.3 Data Pre-Processing 36

7.4 Cleaning Data 37

7.5 Information Retrieval 39

7.6 Predicting the accuracy of the model 38

7.7 Training and Testing on other algorithms 39

7.7.1 Logistic Regression 39

7.8 Prediction 40

7.9 Data Visualization 41

CHAPTER 8: TESTING 44

8.1 Testing plan 44

8.1.1 Test Data 44

8.1.2 Unit testing 44

8.1.3 Test Report 44

8.1.4 Error Report 44

8.1.5 Accuracy 44

CHAPTER 9: CONCLUSION 45

7.1 Conclusion 45

7.2 Future scope 45

CHAPTER 8: BIBILOGRAPHY 46

8.1 References

CHAPTER 1 INTRODUCTION

1.1 Introduction to Machine Learning

Machine Learning is a tender which offers frameworks with capability to perfunctorily take in and enhance from information it secures without really being modified to do as such. This idea accentuates on the advancement of projects that can consequently breakdown information and utilize it to absorb for themselves to improve decisions. This idea is critical to the area of Artificial Intelligence.

Learning initiates with data and impression of this data, for instance, coordinate finding, or direction, or representations, to search for designs, present regularly in material and at last distinguish between poor and enhanced judgments later with respect to cases that we give. The primary goal is to allow the PCs to learn ordinarily without human interpolation and modify exercises as indicated by the required.

ML enables programming applications to wind up more precise in anticipating results without being expressly modified. The essential introduce of ML is to fabricate calculations that can get input information and utilize measurable going-over to get ahead a yield an incentive inside a satisfactory range.

1.2 Some machine learning methods

ML algorithms are mainly categorized as supervised and unsupervised.

i. Supervised Learning calculations can remove those that have been recognized previously to fresh information utilizing named cases to foresee future occasions. Beginning since examination of a preparation dataset, the calculation delivers a gathered capacity to make forecasts about the yield esteems. The framework can give efforts to any first-hand contribution after suitable preparing. The learning calculation can likewise contrast its yield and the right, expected income and realise blunders with a specific end objective to adjust the model as needs be.

ii. In differentiate, unsupervised ML deviousness are developed when the data used to prepare is neither arranged nor marked. Unsupervised learning envisages how frameworks can surmise a capacity to portray a concealed structure from unlabelled information.

1.3 Classification

Classification is a task that requires the use of machine learning algorithms that learn how to assign a class label to examples from the problem domain. Classification predictive modelling involves assigning a class label to input examples

Binary classification refers to predicting one of two classes and multi-class classification involves predicting one of more than two classes. An easy to understand example is classifying email as “*spam*” or “*not spam*.”

Multi-label classification involves predicting one or more classes for each example and imbalanced classification refers to classification tasks where the distribution of examples across the classes is not equal.

From a modelling perspective, classification requires a training dataset with many examples of inputs and outputs from which to learn.

A model will use the training dataset and will calculate how to best map examples of input data to specific class labels. As such, the training dataset must be sufficiently representative of the problem and have many examples of each class label.

Class labels are often string values, e.g. “*spam*,” “*not spam*,” and must be mapped to numeric values before being provided to an algorithm for modelling. This is often referred to as label encoding, where a unique integer is assigned to each class label, e.g. “*spam*” = 0, “*no spam*” = 1.

There are many different types of classification algorithms for modelling classification predictive modelling problems.

Classification predictive modelling algorithms are evaluated based on their results. Classification accuracy is a popular metric used to evaluate the performance of a model based on the predicted class labels. Classification accuracy is not perfect but is a good starting point for many classification tasks.

Instead of class labels, some tasks may require the prediction of a probability of class membership for each example. This provides additional uncertainty in the prediction that an application or user can then interpret. A popular diagnostic for evaluating predicted probabilities is the ROC Curve.

Popular algorithms that can be used for classification include:

- Logistic Regression
- k-Nearest Neighbors
- Decision Trees
- Support Vector Machine
- Random Forest Classifier

Some algorithms do not natively support more than two classes; examples include Logistic Regression and Support Vector Machines.

In case of K-Nearest Neighbors, finding the most suitable K is difficult and is time consuming.

In the project we have used Random Forest Classifier for simplicity and accuracy.

1.4 Proposed System:

Sentiment analysis, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive, neutral or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling object, or predict stock markets for a given company like, if most people think positive about it, possibly its stock markets will increase, and so on. Sentiment analysis is actually far from to be solved since the language is very complex (objectivity/subjectivity, negation, vocabulary and grammar) but it is also why it is very interesting to working on. In this project I choose to try to classify tweets from Twitter into “positive” or “negative” sentiment by building a model based on probabilities. While writing a review in IMDB website, people can share their feelings quickly and spontaneously. They can address it to someone by adding the target sign “@” or participate to a topic by adding a hashtag “#”. Because of the different views people have about films, IMDB reviews is a perfect source of data to determine the current overall opinion about anything.

CHAPTER 2 FEASIBILITY STUDY

A feasibility study involves taking a judgment call on whether a project is doable. The two criteria to judge feasibility are cost required and value to be delivered. A well designed study should offer a historical background of the business or project, a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, such studies precede technical development and project implementation

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions.

2.1 Technical Feasibility

Technical feasibility involves evaluation of the hardware and the software requirements of the proposed system. In this project, the technology involved is text mining. The language that is used to implement this concept Python.

2.2 Economic Feasibility

Economic Feasibility helps in assessing the viability, cost, and benefits associated with projects before financial resources are allocated. This assessment typically involves a cost/ benefits analysis of the project.

The application is so designed that it requires minimal cost and eliminates costs as there would minimal need for manual work. The technologies used helps in understanding the user without any investment. As the machine will be trained it reduces the cost that is required to deploy the man power and also eliminates the problem of time consumption.

2.3 Legal Feasibility

The proposed system doesn’t conflict with legal requirements like data protection acts or social media laws. It ensures legal data access and gives prominence to data security.

2.4 Operational Feasibility

The application involves design-dependent parameters such as reliability, maintainability, supportability, usability, disposability, sustainability, affordability, and others. It minimizes the drawbacks of the current system by building an application that automatically resolves the user queries and helps to analyses the user data.

2.5 Scheduling Feasibility

The project development took place in timely process by understanding time schedules of the project and maintains good time line for project development.

CHAPTER 3 LITERATURE SURVEY

3.1 Python API's & Libraries required

3.1.1 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code.

3.1.2 Scikit-learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. The library is built upon the SciPy.

3.1.3 NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, basic linear algebra, basic statistical operations, random simulation etc

3.1.4 Pandas

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labelled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.

3.1.5 Pickle

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.)

3.1.6 Re

Regular expressions use the backslash character ('\') to indicate special forms or to allow special characters to be used without invoking their special meaning. This collides with Python’s usage of the same character for the same purpose in string literals; for example, to match a literal backslash, one might have to write '\\\\' as the pattern string, because the regular expression must be \\, and each backslash must be expressed as \\ inside a regular Python string literal.

3.1.7 Nltk

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analysing linguistic structure, and more.

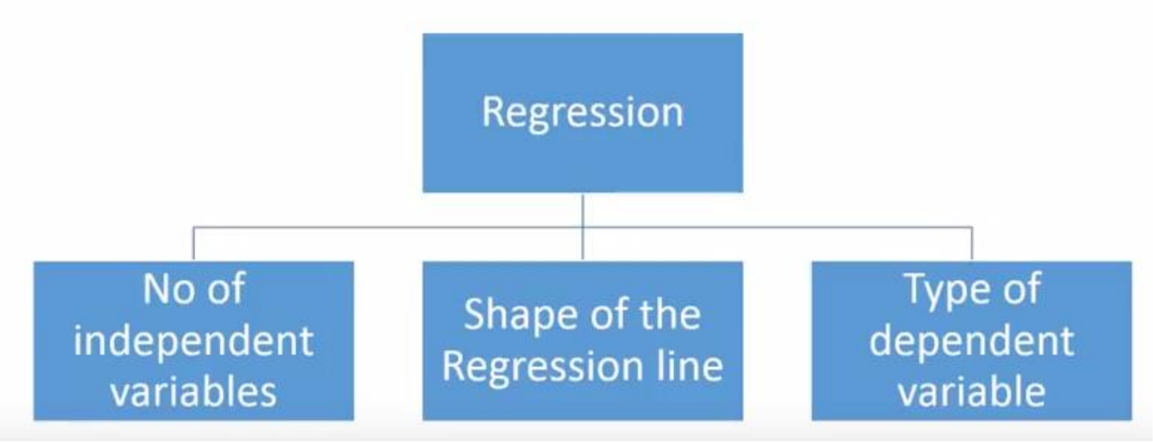
3.2 Python IDE

Integrated Development Environment (IDE) for Python has been bundled with the default implementation of the language. Its main features are – Multi window text editor with syntax highlighting, autocompletion, smart indent and others, python shell with syntax highlighting, integrated debugger with stepping, persistent breakpoints, and call stack visibility.

CHAPTER 4 ALGORITHM DESCRIPTION

4.1 Regression:

Regression analysis is a form of predictive modelling technique which investigates the relationship between a **dependent** (target) and **independent variable (s)** (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression.


Regression
No of independent variables
Shape of the Regression line
Type of dependent variable

Types:

- Linear Regression
- Logistic Regression
- Polynomial Regression
- Stepwise Regression
- Ridge Regression
- Lasso Regression
- ElasticNet Regression

4.2 Logistic Regression:

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

4.2.1 Types of Logistic Regression:

<u>Binomial</u>

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

<u>Multinomial</u>

In such a kind of classification, dependent variable can have 3 or more possible *unordered* types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

<u>Ordinal</u>

In such a kind of classification, dependent variable can have 3 or more possible *ordered* types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

4.2.2 Logistic Regression Assumptions:

Before diving into the implementation of logistic regression, we must be aware of the following assumptions about the same –

- In case of binary logistic regression, the target variables must be binary always and the desired outcome is represented by the factor level 1.
- There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other.
- We must include meaningful variables in our model.

- We should choose a large sample size for logistic regression.

4.2.3 Regression Models:

- Binary Logistic Regression Model − The simplest form of logistic regression is binary or binomial logistic regression in which the target or dependent variable can have only 2 possible types either 1 or 0.

- Multinomial Logistic Regression Model − Another useful form of logistic regression is multinomial logistic regression in which the target or dependent variable can have 3 or more possible ***unordered*** types i.e. the types having no quantitative significance.

4.3 Advantages of Logistic Regression:

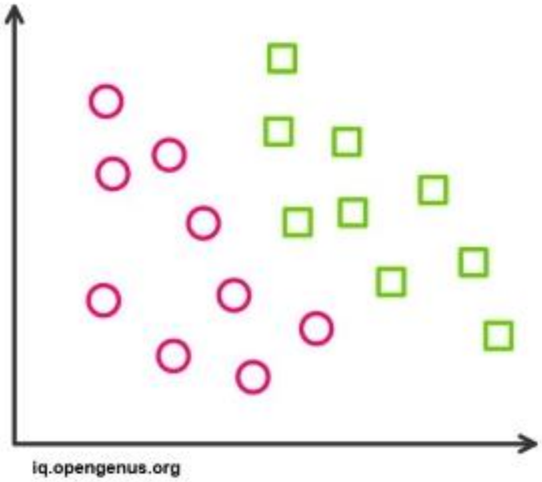
1. Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.

2. The predicted parameters (trained weights) give inference about the importance of each feature. The direction of association i.e. positive or negative is also given. So we can use logistic regression to find out the relationship between the features.

3. This algorithm allows models to be updated easily to reflect new data, unlike decision trees or support vector machines. The update can be done using stochastic gradient descent.

4. Logistic Regression outputs well-calibrated probabilities along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.
5. In a low dimensional dataset having a sufficient number of training examples, logistic regression is less prone to over-fitting.

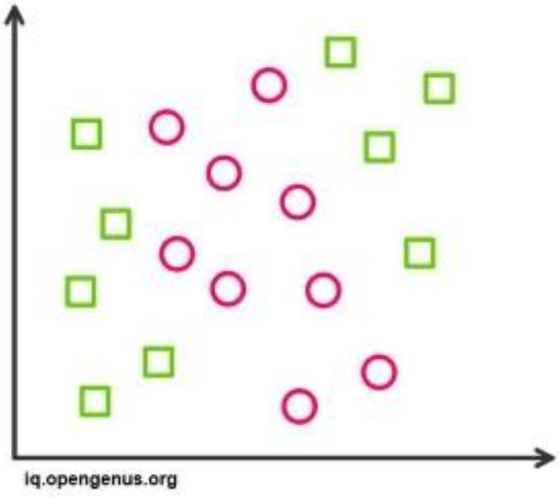
6. Rather than straight away starting with a complex model, logistic regression is sometimes used as a benchmark model to measure performance, as it is relatively quick and easy to implement.
7. Logistic Regression proves to be very efficient when the dataset has features that are linearly separable.



4.4 Disadvantages of Logistic Regression:

1. Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On high dimensional datasets, this may lead to the model being over-fit on the training set, which means overstating the accuracy of predictions on the training set and thus the model may not be able to predict accurate results on the test set. This usually happens in the case when the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid over-fitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.

2. Nonlinear problems can't be solved with logistic regression since it has a linear decision surface. Linearly separable data is rarely found in real world scenarios. So, the transformation of nonlinear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.



4.5 Logistic Regression Applications

Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences.

1. In the Trauma and Injury Severity Score (TRISS), it is widely used to predict mortality in injured patients, was originally developed by Boyd *et al.* using logistic regression. Many other medical scales used to assess severity of a patient have been developed using logistic regression.
2. Logistic regression may be used to predict the risk of developing a given disease (e.g. diabetes; coronary heart disease), based on observed characteristics of the patient (age, sex, body mass index, results of various blood tests, etc.).
3. To predict whether a Nepalese voter will vote Nepali Congress or Communist Party of Nepal or Any Other Party, based on age, income, sex, race, state of residence, votes in previous elections, etc.
4. The technique can also be used in engineering, especially for predicting the probability of failure of a given process, system or product.

5. It is also used in marketing applications such as prediction of a customer's propensity to purchase a product or halt a subscription, etc.

6. In economics it can be used to predict the likelihood of a person's choosing to be in the labour force, and a business application would be to predict the likelihood of a homeowner defaulting on a mortgage. Conditional random fields, an extension of logistic regression to sequential data, are used in natural language processing.

4.6 The Shortcoming of Train-Test Split

Imagine you have a dataset with 5000 rows. The train_test_split function has an argument for test size that you can use to decide how many rows go to the training set and how many go to the test set. The larger the test set, the more reliable your measures of model quality will be. At an extreme, you could imagine having only 1 row of data in the test set. If you compare alternative models, which one makes the best predictions on a single data point will be mostly a matter of luck.

You will typically keep about 20% as a test dataset. But even with 1000 rows in the test set, there's some random chance in determining model scores. A model might do well on one set of 1000 rows, even if it would be inaccurate on a different 1000 rows. The larger the test set, the less randomness (aka "noise") there is in our measure of model quality.

But we can only get a large test set by removing data from our training data, and smaller training datasets mean worse models. In fact, the ideal modelling decisions on a small dataset typically aren't the best modelling decisions on large dataset

CHAPTER 5 SYSTEM ANALYSIS

5.1 System Requirements

a) Python

Python is a deciphered language Guido van Rossum, Python has a diagram h ypothesis that complements code decipherability, and a sentence structure that empowers programming architects to express thoughts in less lines of code noticeably using imperative whitespace. It gives builds up that engage little immense. Incorporates a kind modified organization. Reinforces different perfect models, masterminded, essential, useful and, and a huge and exhaustive.

Fig 4.1 Python

b) Jupyter Notebook

The Jupyter Note pad is an open-source web application that allows you to make and offer reports that cover live code, conditions, observations and account content. Utilizations include: information cleaning and change, numerical reproduction, factual demonstrating, information representation, machine learning, and significantly more. The Scratch pad has bolster for more than 40 programming dialects, including Python, R, Julia, and Scala. Note pads can be imparted to others using email, Dropbox, GitHub and the Jupyter Note pad Watcher.

- Project and code navigation
- Python refactoring

- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Support for scientific tools like matplotlib, numpy and nltk [professional edition only]

5.2 Hardware Requirements

- RAM: 4GB or Higher
- Processor: Intel i3 and above
- Hard Disk: 5 GB minimum

a) Importing Dataset

In this we need to Load the dataset (question) which contains the required information and store it in a data frame.

b) Train the dataset

We split our dataset into trained and test dataset. We train our data using classification technique.

c) Apply Classification technique

Logistic regression is named for the function used at the core of the method, the logistic function.

The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

1 / (1 + e^-value)

d)Test dataset

We test the data over trained model it predicts whether the question pairs are duplicate or not.

e)Display results

We plot our confusion matrix which shows the efficiency of algorithm.

CHAPTER 6 SOFTWARE DEISGN

6.1 UML Diagrams

Unified Modelling Language is a tool that helps a designer to present his ideas about the project to his client and his developer. Modelling plays a crucial role in designing a software. A poorly designed model can lead to a poorly developed software. A UML system has using five different views that help in describing systems from different perspectives. Each view has a set of diagrams that and components that represent the real time objects.

a. User Model View:

i. It models the user behaviour in a system context.
ii. All the diagrams are drawn keeping in mind the user's response and reaction towards a system.

b. Structural Model View

i. This view consists of class diagram and object diagram which is used to model the static structures.
ii. It uses objects, attributes, operations and relationships.

c. Behavioural Model View

i. It mainly consists of the sequence diagram, collaboration diagram, state chart diagram and activity diagram. They mainly represent flow of actions between different objects involved in the system
ii. They are used to visualize various dynamic aspects of the system architecture.

d. Implementation Model View

i. This view consists of component diagrams and deployment diagrams. This view models the static software modules for an organization.
ii. This usually contains the data files, documentation, the executables and source code.
iii. These are the physically replaceable components of the system. They are modelled using component

6.1.1 Use Case Diagram

The basic representation for the interaction of the user with the system is represented using the use case diagram. It involves the relationship between the user and various use cases with the actors being involved. There are different kinds of relationships that are involved between the use cases and the actors. They include:

a) Association relationship
b) Generalization
c) Dependency
d) Realizations
e) Transitions

The following represents the use case diagram of the proposed system:

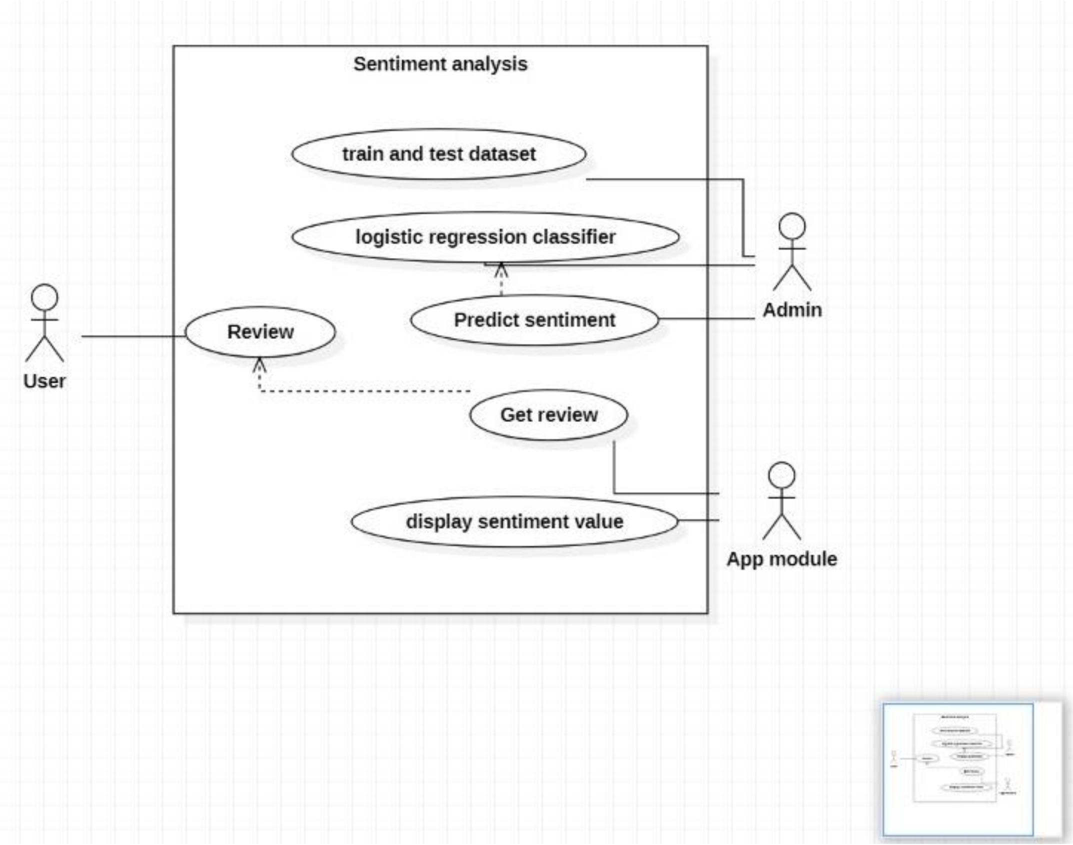


Fig 6.1: Use Case Diagram for Developed Model

6.1.2 Class Diagram

They are static representation of an application. Only the class diagrams have the capability to be directly mapped with the OOP Languages because in OOPs everything is model in the form of classes and objects. Because of this reason these diagrams are used widely at the time of construction. This is one of the most popularly used UML diagram in the designer community. A class diagram plays an essential role in forward and reverse engineering.

a) It acts as a base for the component and deployment diagrams.
b) It mainly describes and defines the basic responsibilities of a system's application.
c) It implements the analysis and design view for a static application.

In a class diagram, each object is modelled as a class. Each class consists of section or compartments.

1. Class name
2. Attributes of a class or operations
3. Methods or functions
4. Documentation (optional section)

The following points ought to be recollected while drawing a class diagram:

a) The name of the class diagram must be meaningful to portray the aspect of the framework.
b) Each component and their connections must be distinguished ahead of time.
c) Each class has a responsibility (attributes and methods) that must be identified clearly.
d) Number of properties for each class must be minimum. Since pointless properties will make the diagram convoluted.
e) At whatever point required to depict some part of the diagram use notes Since toward the finish of the diagram it must be justifiable to the designer/coder.
f) Before finalising the last version, the diagram must be drawn on plain paper and revise whatever number circumstances as would be prudent to make it redress.

Scopes:
The UML diagrams have two different types of scopes for class members:

i) instance members scope and
ii) classifier members scope

1. Classifier members are "static" members of a class in many programming languages. The scope is the class itself.
i) Static attributes are common to all other objects that invoke the class.
ii) Static methods are not instantiated.

2. **Instance members** are nothing but the members that are local to an object.

i) The main purpose of instance members is to allow the objects to store their states.

ii) Declarations outside the methods are usually known as instance members.

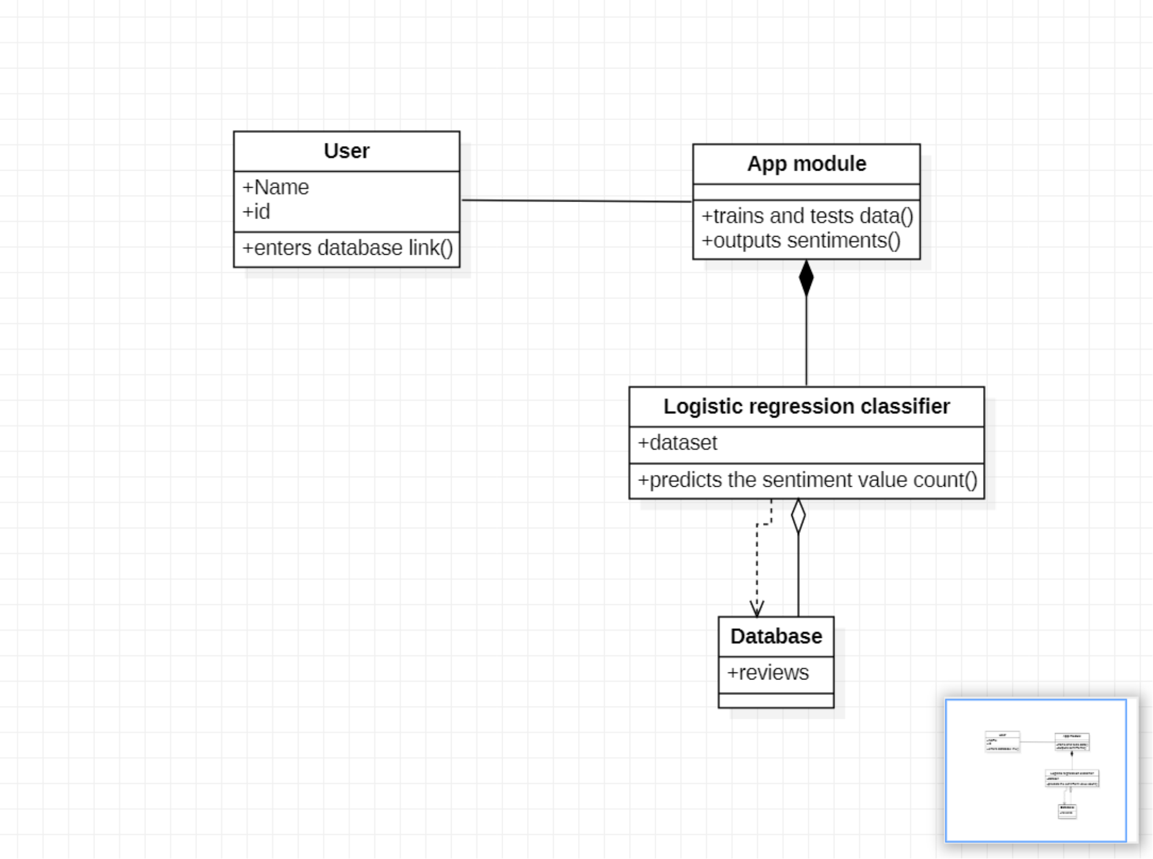


Fig 6.2**:** Class Diagram for Developed Model

6.1.3 Sequence Diagram

The Sequence Diagram depicts the time sequence among various objects in an application. It depicts the sequence of messages with which objects communicate with each other so that they carry out the required functionality.

It consists of the lifelines which are usually parallel vertical lines. It consists of horizontal arrows which indicates the direction of the messages that are exchanged in a proper order which makes the user easy to understand.

The lifeline for a given object represents a role. The synchronous calls are represented with the help of a solid arrow head whereas the asynchronous messages are represented with the help of open arrow heads.

All objects are represented according to their time ordering. Timing of messages plays a major role in sequence diagrams. An object is killed immediately after its use in sequence diagrams.

1. Common Properties

Arrangement graph is much the same as unique sort of diagram and offers some in distinguishable properties from other diagrams. In any case, it varies from every single other diagram in its content.

2. Contents

Objects are normally named or unknown instances of class, however may likewise speak to occurrences of different things, for example components, collaboration and nodes. Graphically, object is represented as a rectangle by underlying its name.

3. Links

A link is a semantic association among objects i.e., an object of an affiliation
is called as a connection. It is represented as a line.

4. **Messages**

A message is a determination of a correspondence between objects that passes on the data with the desire that the action will follow.

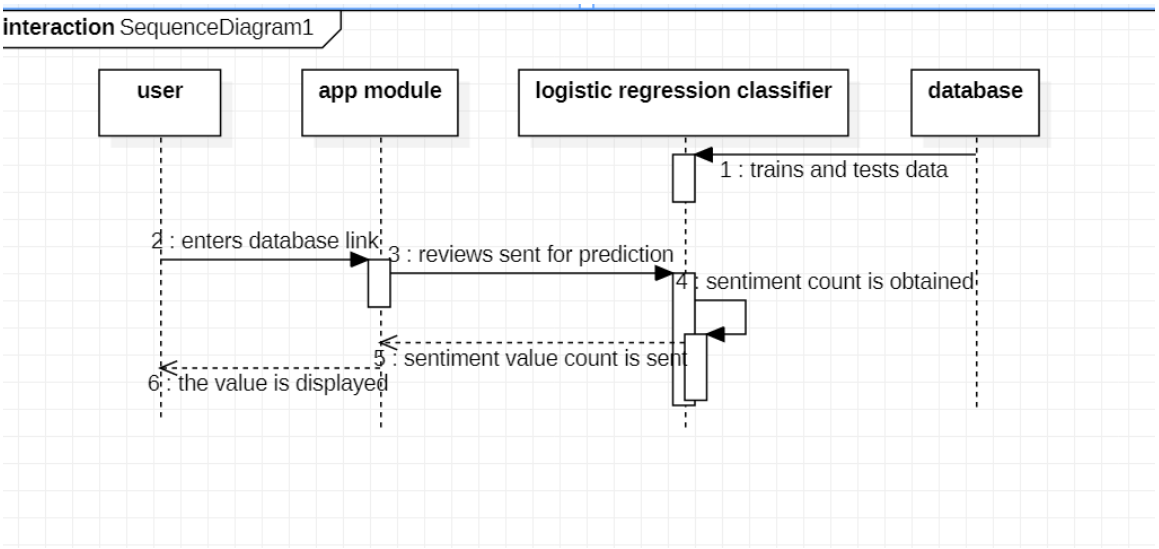

interaction SequenceDiagram1
user
app module
logistic regression classifier
database
1 : trains and tests data
2 : enters database link
3 : reviews sent for prediction
4 : sentiment count is obtained
5 : sentiment value count is sent
6 : the value is displayed

Fig 6.3 Sequence Diagram for Developed Model

6.1.4 Activity Diagram

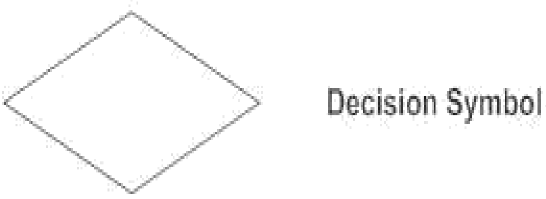
The flow from one activity to another activity can be represented in the form of a flow chart which is usually an activity diagram. It forms a backbone for the UML diagrams. It depicts the dynamic aspects for all the objects within the system.The control flow from one object to another object is drawn which shows the basic operations that are to be performed.

Activity diagrams are constructed using the following:

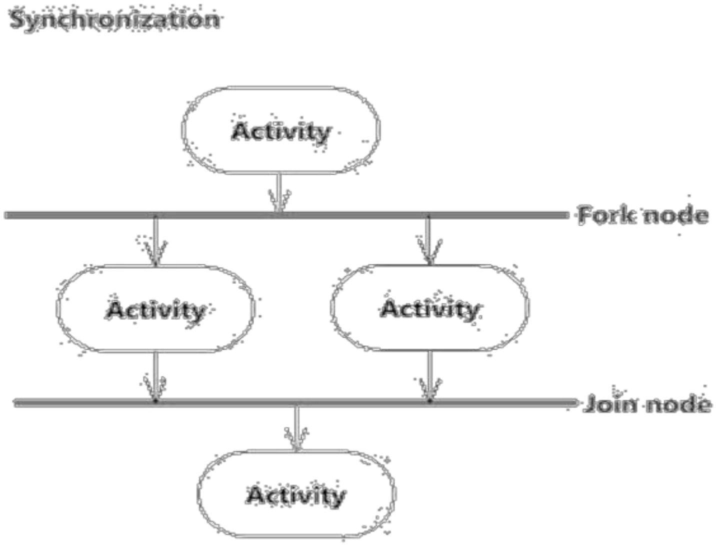
1. Actions are represented using rounded rectangles

Activity
Activity

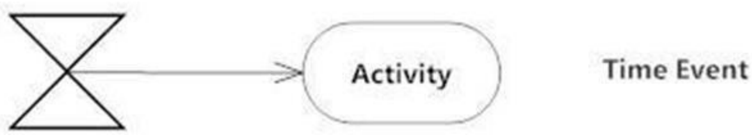
2. decisions are represented using diamonds



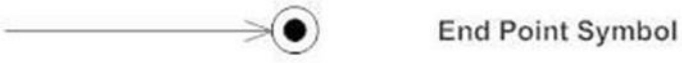
3. concurrent activities bars are represented using the start (split) or end(join);



4. Time event is represented as



5. final state is represented using encircled black circle.



The basic purpose of an activity diagram is same as that of other UML diagrams. The dynamic behaviour of the system is viewed by the activity diagram. They are used to construct a system using the backward and forward engineering mechanisms.

The purpose of an activity diagram is as follows:

1) For drawing the flow (i.e. activity) in a system.

2) For showing the flow of sequence from one activity to another activity.

3) For showing the concurrent and parallel flow of actions in the system. The elements that are used in an activity diagram are as follows:

i) Association relationship

ii) Activities

iii) Conditions and Constraints.

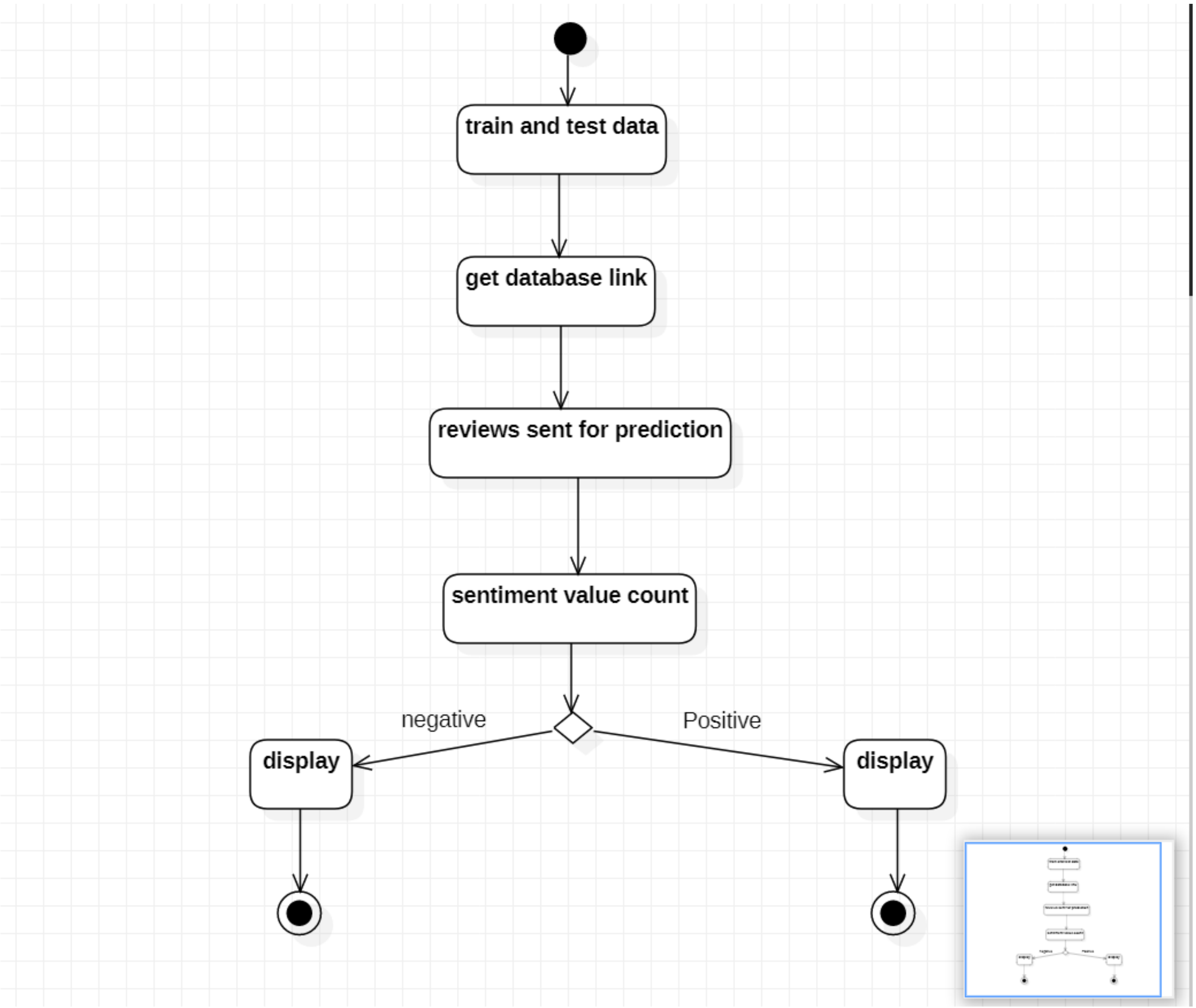
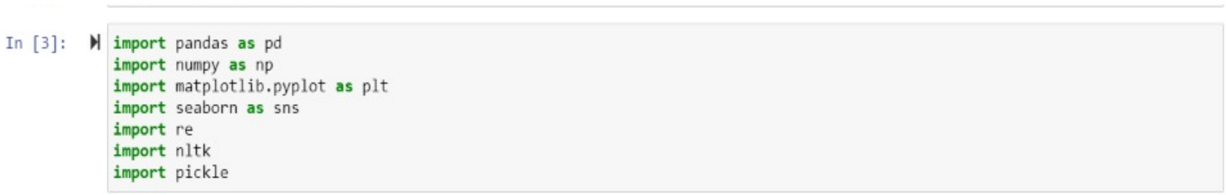


Fig 6.4 Activity Diagram

CHAPTER 7 IMPLEMENTATION

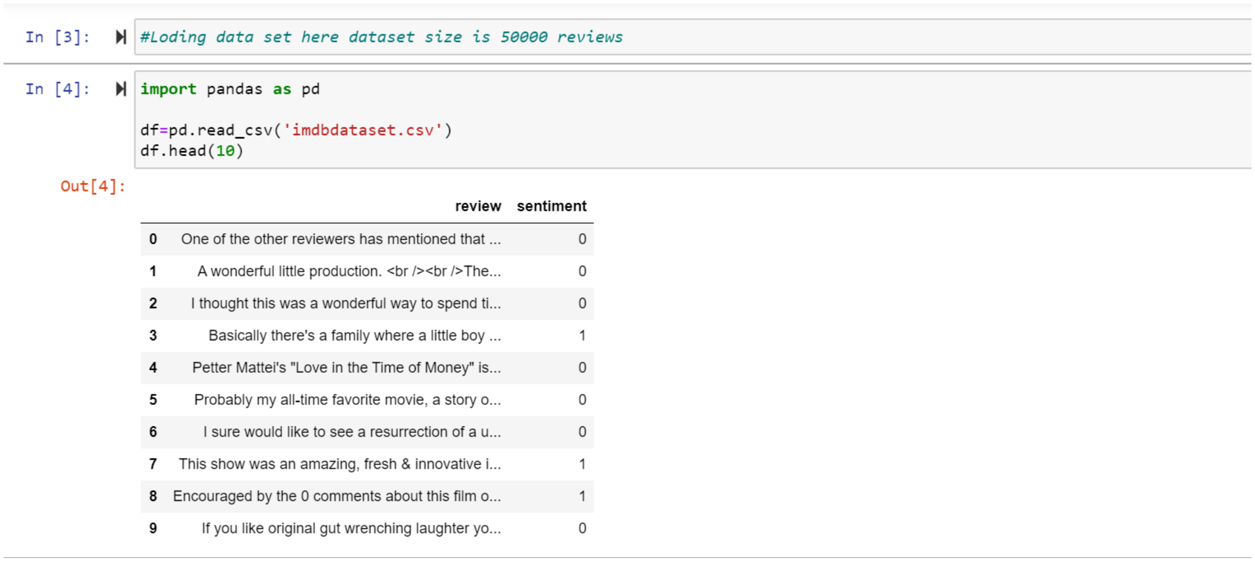
7.1 Code for loading dataset and Importing Libraries

Numpy, Pandas, pickle and matplotlib, seaborn and nltk libraries are imported. Pickle library reduces the task of loading datasets and training algorithm every time the program is executed.



```
In [3]:  import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns
         import re
         import nltk
         import pickle
```

Loading data:



```
In [3]:  #Loding data set here dataset size is 50000 reviews
```

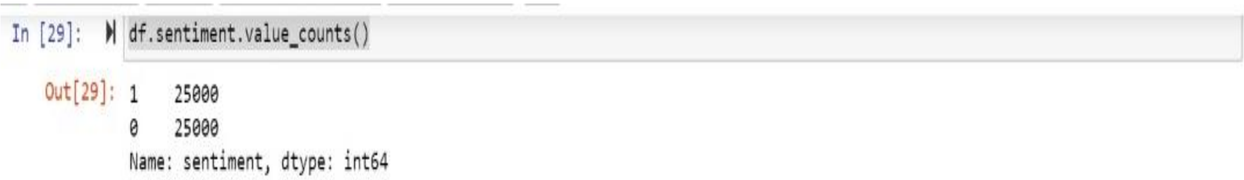
```
In [4]:  import pandas as pd

         df=pd.read_csv('imdbdataset.csv')
         df.head(10)
```

Out[4]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	0
1	A wonderful little production. <br /><br />The...	0
2	I thought this was a wonderful way to spend ti...	0
3	Basically there's a family where a little boy ...	1
4	Petter Mattei's "Love in the Time of Money" is...	0
5	Probably my all-time favorite movie, a story o...	0
6	I sure would like to see a resurrection of a u...	0
7	This show was an amazing, fresh & innovative i...	1
8	Encouraged by the 0 comments about this film o...	1
9	If you like original gut wrenching laughter yo...	0

7.2 Data Ratio

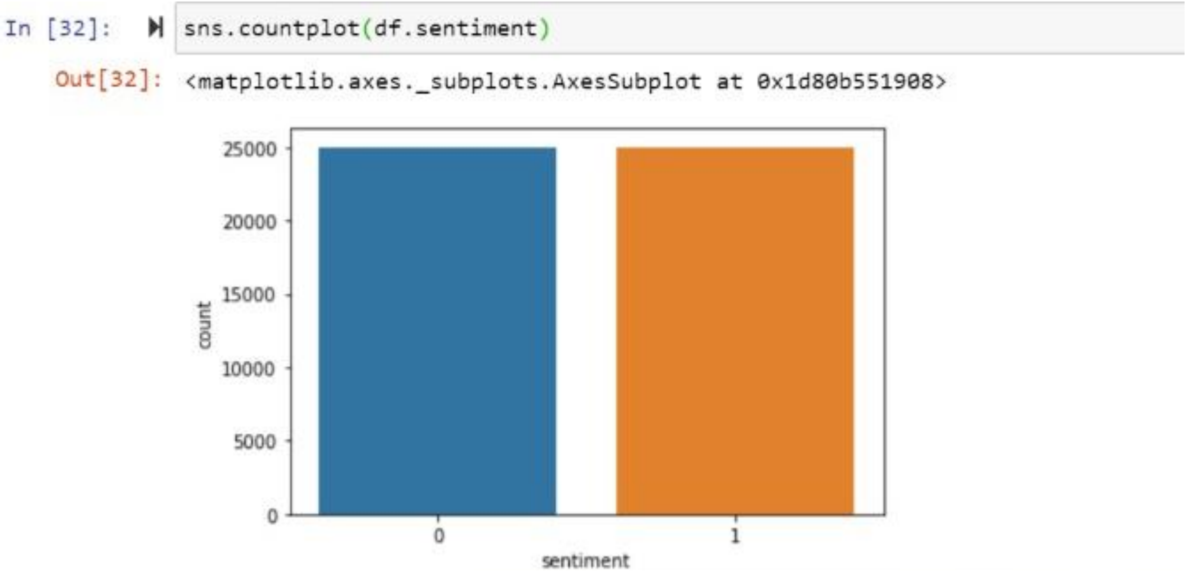
We use the following command to check the sentiment count value of each sentiment in the dataset.



```
In [29]:  df.sentiment.value_counts()

 Out[29]: 1    25000
          0    25000
          Name: sentiment, dtype: int64
```

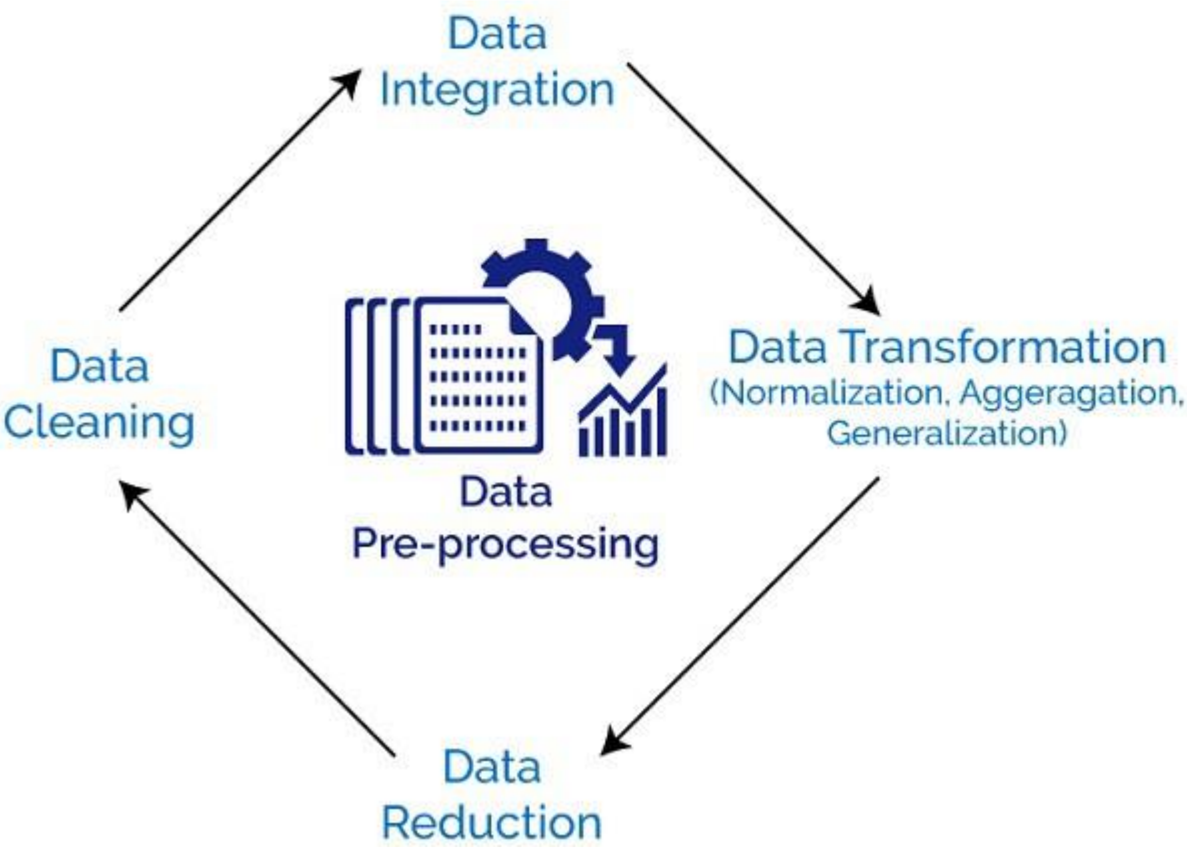
As we can see, the data is equally balanced between positive and negative sentiments.

Graphical representation of the sentiment value count.

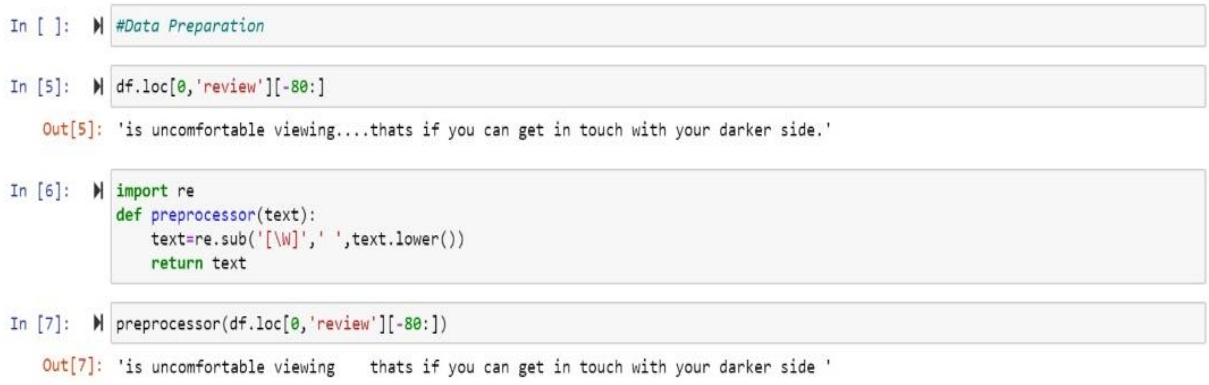


```
In [32]:  sns.countplot(df.sentiment)

 Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x1d80b551908>
```

7.3 Data Pre-processing

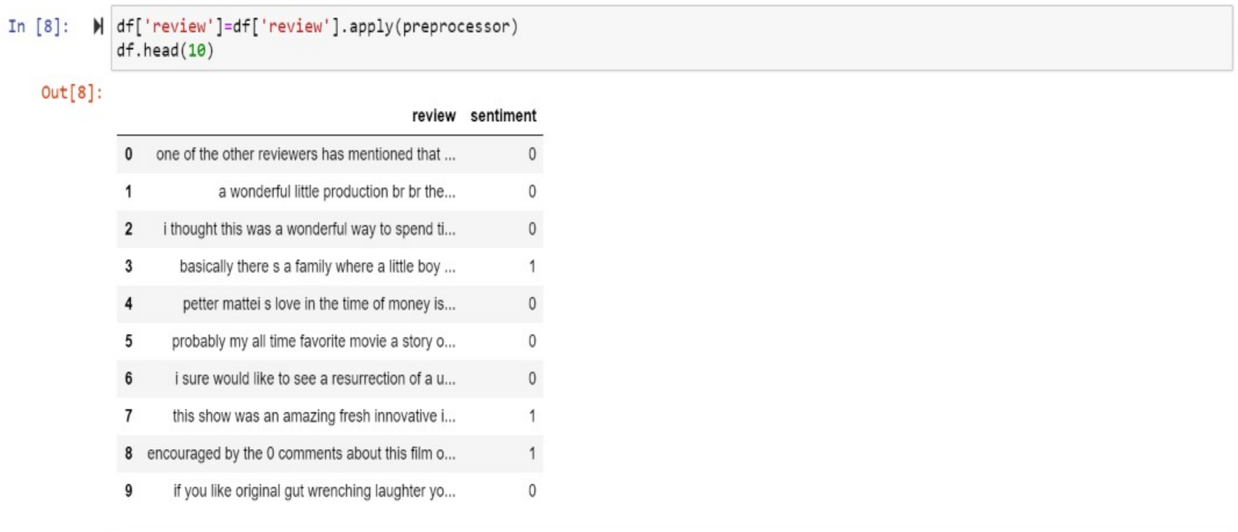


The data is prepared and unwanted symbols are removed using the user-defined function called pre-processor.



```
In [ ]:   #Data Preparation

In [5]:   df.loc[0,'review'][-80:]

Out[5]:  'is uncomfortable viewing....thats if you can get in touch with your darker side.'

In [6]:   import re
          def preprocessor(text):
              text=re.sub('[\W]',' ',text.lower())
              return text

In [7]:   preprocessor(df.loc[0,'review'][-80:])

Out[7]:  'is uncomfortable viewing    thats if you can get in touch with your darker side '
```

The data is printed after applying the pre-processor function.



```
In [8]:  df['review']=df['review'].apply(preprocessor)
         df.head(10)
```

Out[8]:

	review	sentiment
0	one of the other reviewers has mentioned that ...	0
1	a wonderful little production br br the...	0
2	i thought this was a wonderful way to spend ti...	0
3	basically there s a family where a little boy ...	1
4	petter mattei s love in the time of money is...	0
5	probably my all time favorite movie a story o...	0
6	i sure would like to see a resurrection of a u...	0
7	this show was an amazing fresh innovative i...	1
8	encouraged by the 0 comments about this film o...	1
9	if you like original gut wrenching laughter yo...	0

7.4 Cleaning Data

We use a user defined function called clean in which we are tokenizing the data, removing the stop words and stemming the data.

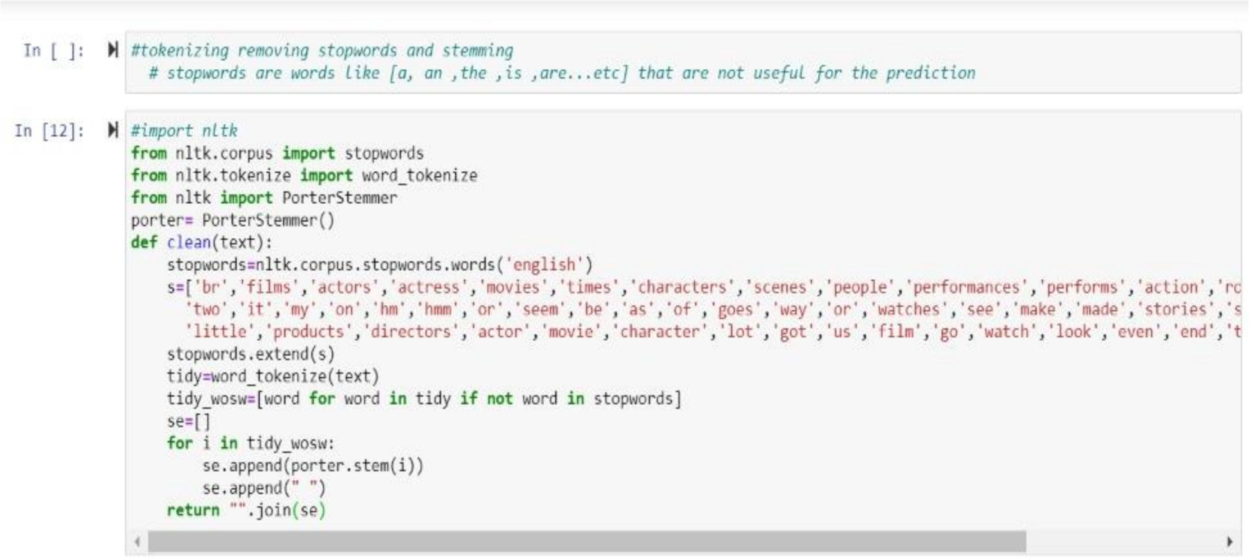
Tokenizing - we will tokenize all the cleaned reviews in our dataset. Tokens are individual terms or words, and tokenization is the process of splitting a string of text into tokens using word_tokenize in built function.

Stemming - is a rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc.) from a word. For example — “play”, “player”, “played”, “plays” and “playing” are the different variations of the word — “play”

Stopwords - are the English words which do not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For example, the words like the, he, have etc. Such words are already captured this in corpus named corpus.

The aforementioned methods are applied using nltk (natural language took kit)

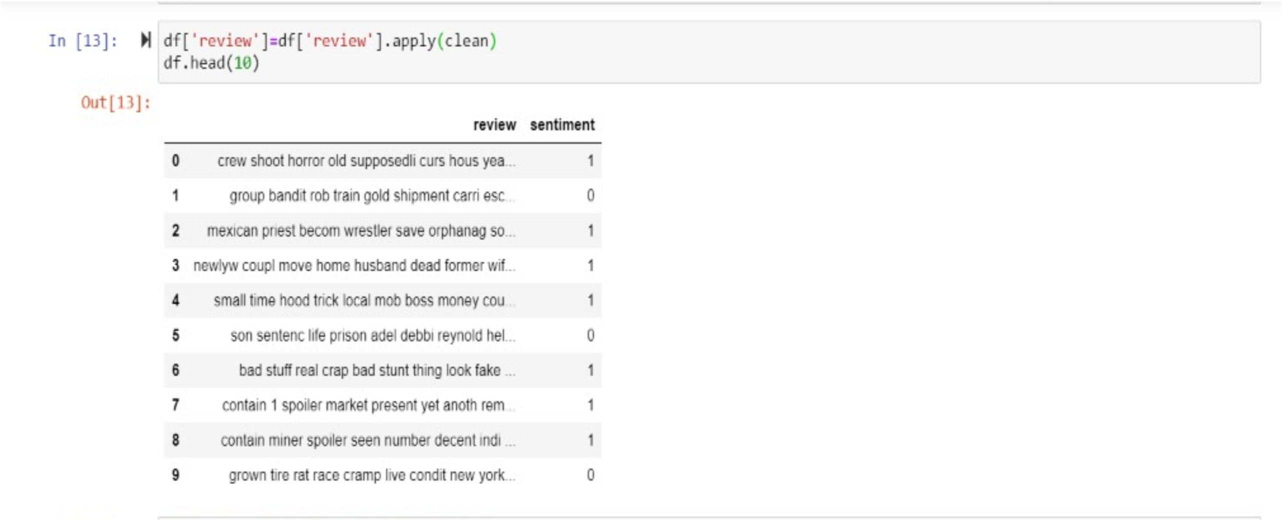
Input:



```
In [ ]:  #tokenizing removing stopwords and stemming
           # stopwords are words like [a, an ,the ,is ,are...etc] that are not useful for the prediction
```

```
In [12]: #import nltk
         from nltk.corpus import stopwords
         from nltk.tokenize import word_tokenize
         from nltk import PorterStemmer
         porter= PorterStemmer()
         def clean(text):
             stopwords=nltk.corpus.stopwords.words('english')
             s=['br','films','actors','actress','movies','times','characters','scenes','people','performances','performs','action','ro
               'two','it','my','on','hm','hmm','or','seem','be','as','of','goes','way','or','watches','see','make','made','stories','s
               'little','products','directors','actor','movie','character','lot','got','us','film','go','watch','look','even','end','t
             stopwords.extend(s)
             tidy=word_tokenize(text)
             tidy_wosw=[word for word in tidy if not word in stopwords]
             se=[]
             for i in tidy_wosw:
                 se.append(porter.stem(i))
                 se.append(" ")
             return "".join(se)
```

Output:



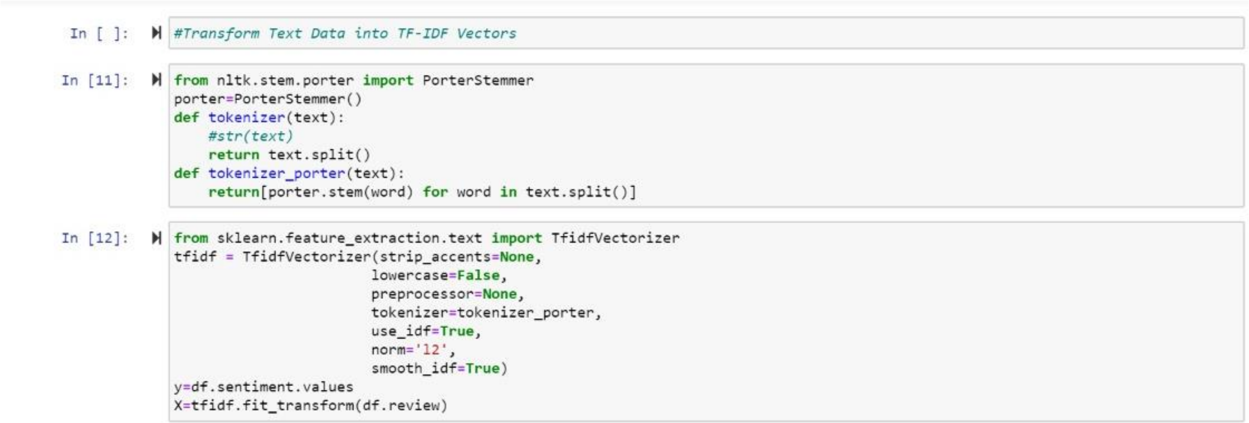
```
In [13]: df['review']=df['review'].apply(clean)
         df.head(10)
```

Out[13]:

	review	sentiment
0	crew shoot horror old supposedli curs hous yea...	1
1	group bandit rob train gold shipment carri esc...	0
2	mexican priest becom wrestler save orphanag so...	1
3	newlyw coupl move home husband dead former wif...	1
4	small time hood trick local mob boss money cou...	1
5	son sentenc life prison adel debbi reynold hel...	0
6	bad stuff real crap bad stunt thing look fake ...	1
7	contain 1 spoiler market present yet anoth rem...	1
8	contain miner spoiler seen number decent indi ...	1
9	grown tire rat race cramp live condit new york...	0

7.5 Information Retrieval:

In information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.



```
In [ ]:   #Transform Text Data into TF-IDF Vectors
```

```
In [11]:  from nltk.stem.porter import PorterStemmer
          porter=PorterStemmer()
          def tokenizer(text):
              #str(text)
              return text.split()
          def tokenizer_porter(text):
              return[porter.stem(word) for word in text.split()]
```

```
In [12]:  from sklearn.feature_extraction.text import TfidfVectorizer
          tfidf = TfidfVectorizer(strip_accents=None,
                                  lowercase=False,
                                  preprocessor=None,
                                  tokenizer=tokenizer_porter,
                                  use_idf=True,
                                  norm='l2',
                                  smooth_idf=True)
          y=df.sentiment.values
          X=tfidf.fit_transform(df.review)
```

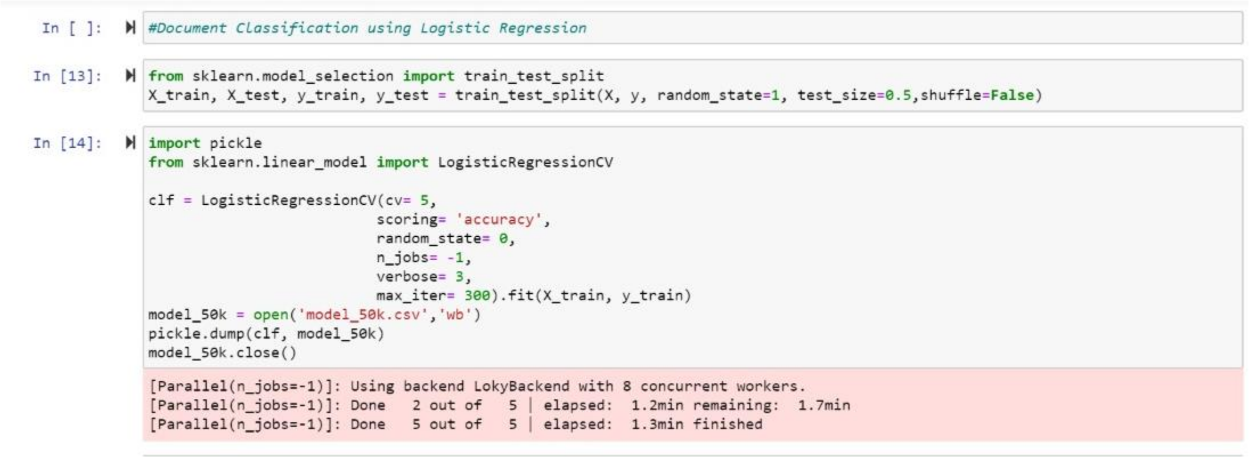
Datasets are imported and then the data set is split into two sets X and Y.

X contains the set of reviews and Y contains sentiment values for the corresponding review.

7.6 Training and testing on algorithms

7.6.1 Logistic Regression:

Building and evaluation

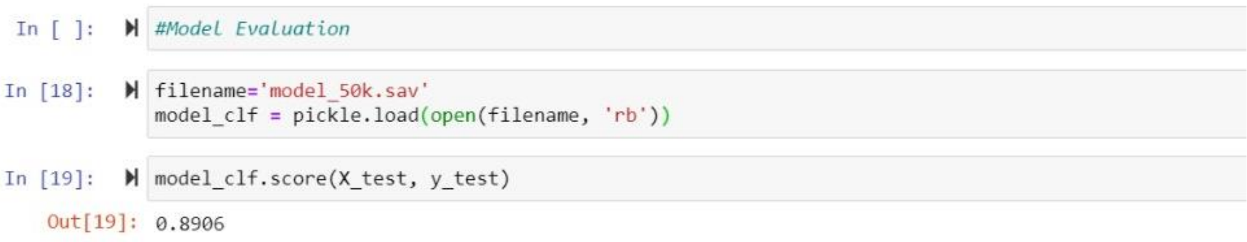


```
In [ ]:   #Document Classification using Logistic Regression
```

```
In [13]:  from sklearn.model_selection import train_test_split
          X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1, test_size=0.5,shuffle=False)
```

```
In [14]:  import pickle
          from sklearn.linear_model import LogisticRegressionCV

          clf = LogisticRegressionCV(cv= 5,
                                     scoring= 'accuracy',
                                     random_state= 0,
                                     n_jobs= -1,
                                     verbose= 3,
                                     max_iter= 300).fit(X_train, y_train)
          model_50k = open('model_50k.csv','wb')
          pickle.dump(clf, model_50k)
          model_50k.close()

          [Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
          [Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:  1.2min remaining:  1.7min
          [Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  1.3min finished
```

Calculating accuracy



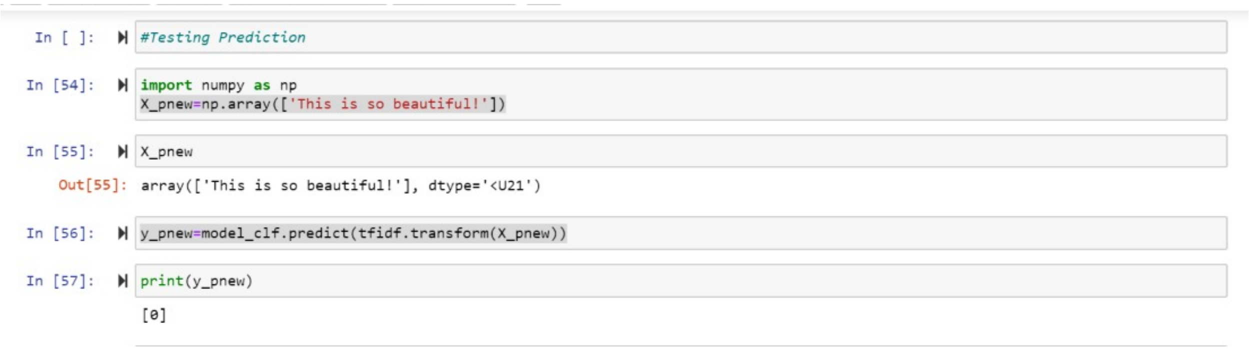
```
In [ ]:   #Model Evaluation

In [18]:  filename='model_50k.sav'
          model_clf = pickle.load(open(filename, 'rb'))

In [19]:  model_clf.score(X_test, y_test)
Out[19]:  0.8906
```

7.7 Training our classifier with the entire dataset:

Sometimes the data inside the testing dataset has useful information. If such data is not used for training the predictions might be wrong. So we use both train and test set data while predicting the diseases in the flask environment.

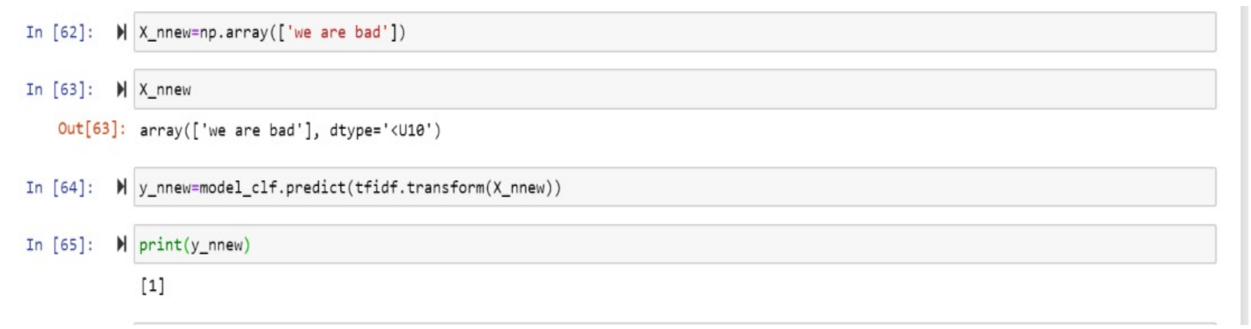
7.8 Prediction:

Predicting the sentiment value for a positive review.



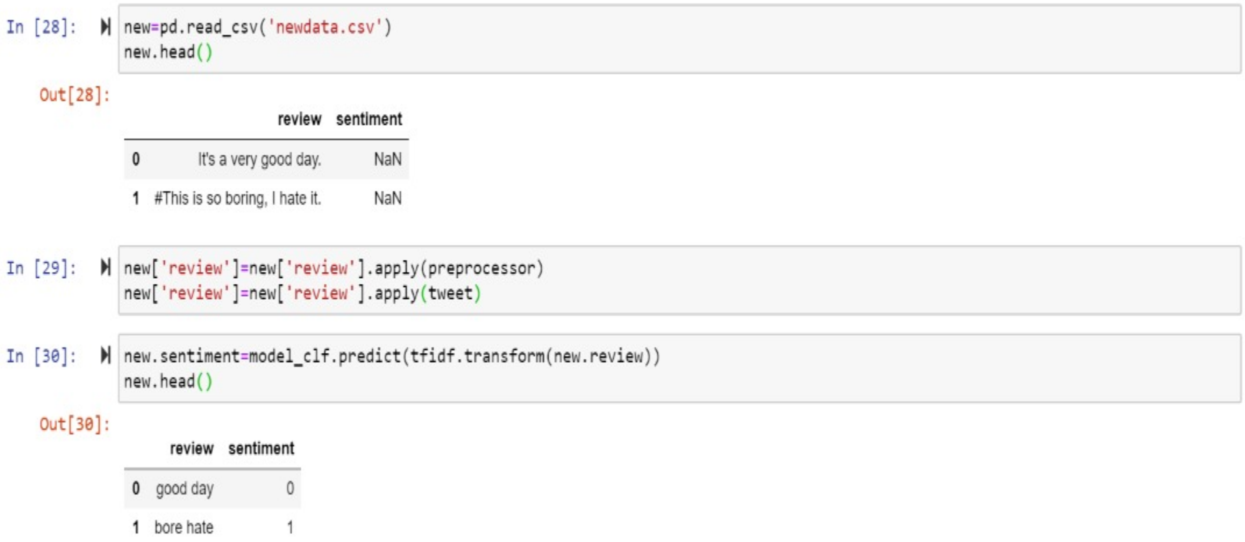
```
In [ ]:   #Testing Prediction

In [54]:  import numpy as np
          X_pnew=np.array(['This is so beautiful!'])

In [55]:  X_pnew
Out[55]:  array(['This is so beautiful!'], dtype='<U21')

In [56]:  y_pnew=model_clf.predict(tfidf.transform(X_pnew))

In [57]:  print(y_pnew)
          [0]
```

Predicting the sentiment value for a negative review.



```
In [62]:  X_nnew=np.array(['we are bad'])

In [63]:  X_nnew
Out[63]:  array(['we are bad'], dtype='<U10')

In [64]:  y_nnew=model_clf.predict(tfidf.transform(X_nnew))

In [65]:  print(y_nnew)
          [1]
```

Predicting sentiment for multiple reviews.



```
In [28]:  new=pd.read_csv('newdata.csv')
          new.head()
```

Out[28]:

	review	sentiment
0	It's a very good day.	NaN
1	#This is so boring, I hate it.	NaN

```
In [29]:  new['review']=new['review'].apply(preprocessor)
          new['review']=new['review'].apply(tweet)
```

```
In [30]:  new.sentiment=model_clf.predict(tfidf.transform(new.review))
          new.head()
```

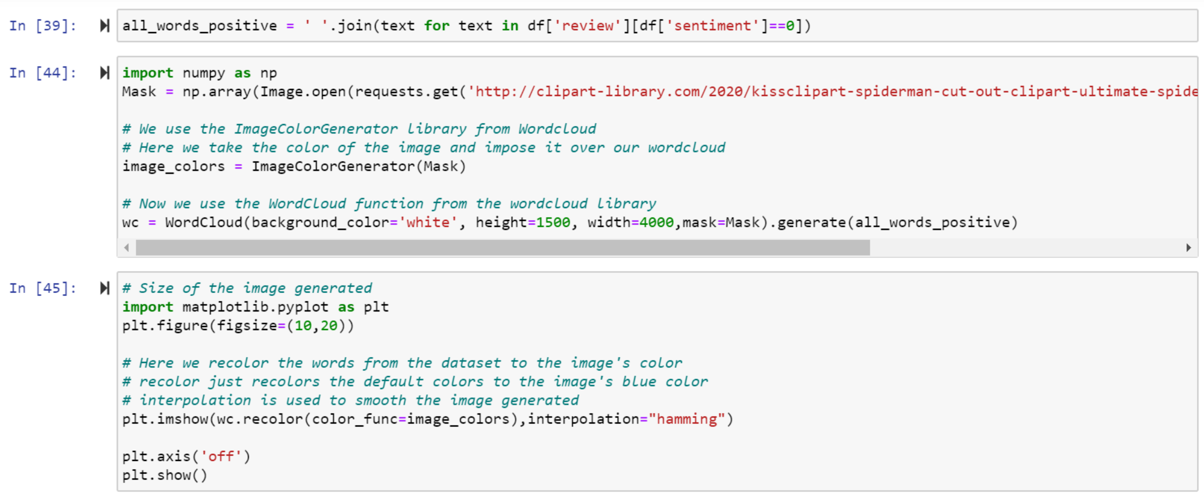
Out[30]:

	review	sentiment
0	good day	0
1	bore hate	1

7.9 Data Visualization

Data Visualization is one of the important techniques used in ML.

For positive reviews



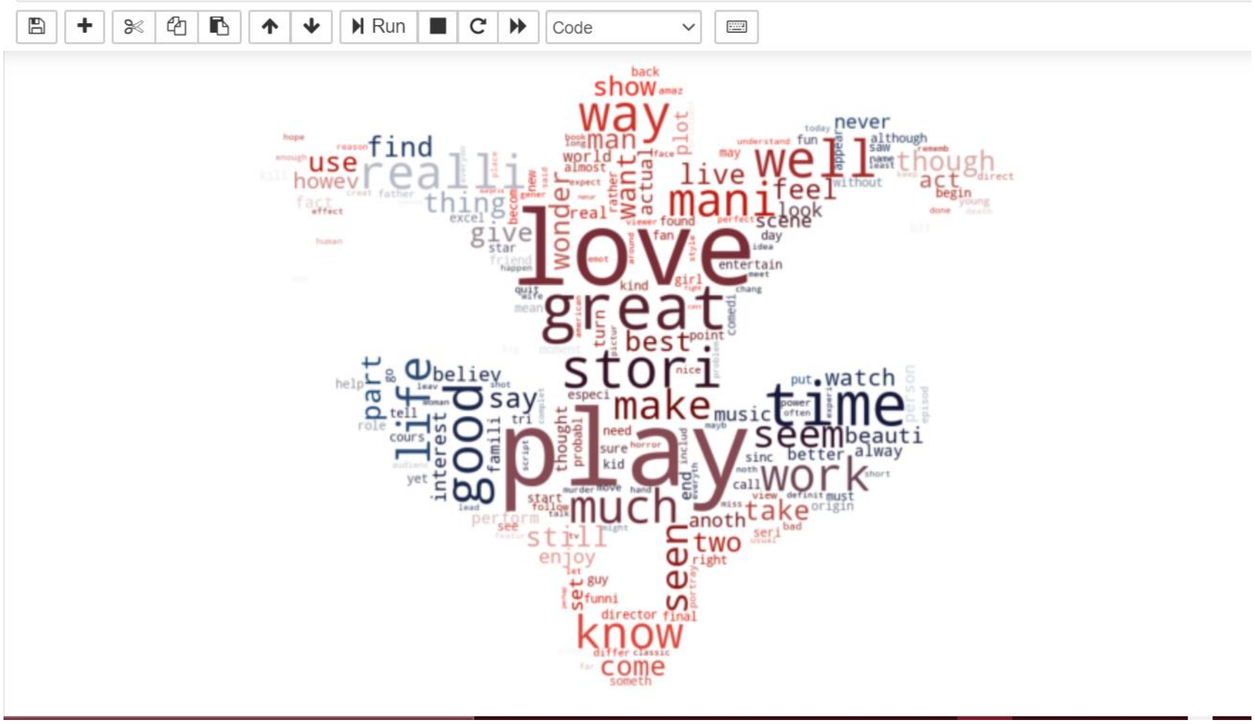
```
In [39]:  all_words_positive = ' '.join(text for text in df['review'][df['sentiment']==0])
```

```
In [44]:  import numpy as np
          Mask = np.array(Image.open(requests.get('http://clipart-library.com/2020/kissclipart-spiderman-cut-out-clipart-ultimate-spide

          # We use the ImageColorGenerator library from Wordcloud
          # Here we take the color of the image and impose it over our wordcloud
          image_colors = ImageColorGenerator(Mask)

          # Now we use the WordCloud function from the wordcloud library
          wc = WordCloud(background_color='white', height=1500, width=4000,mask=Mask).generate(all_words_positive)
```

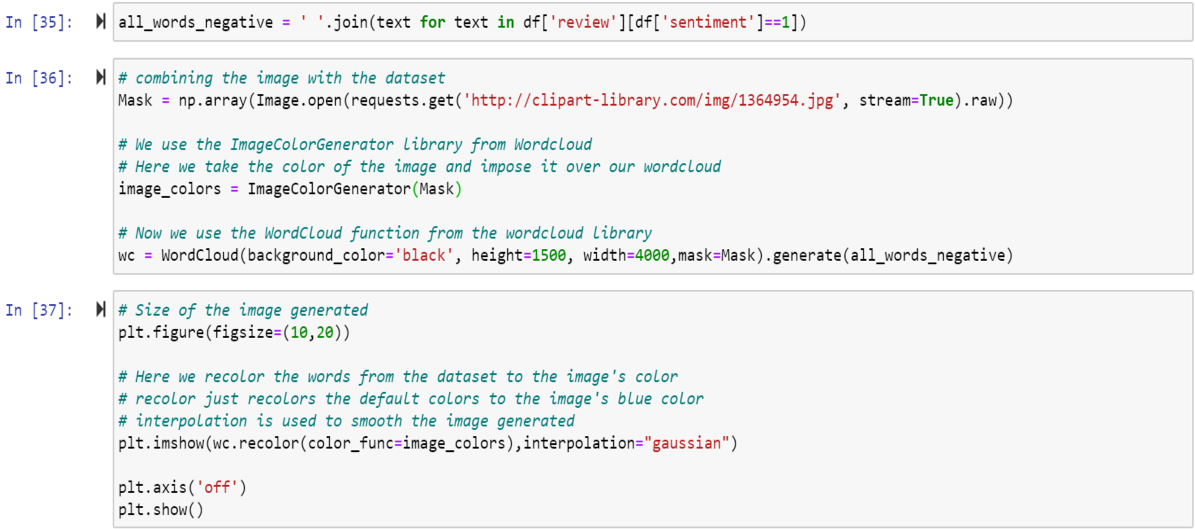
```
In [45]:  # Size of the image generated
          import matplotlib.pyplot as plt
          plt.figure(figsize=(10,20))

          # Here we recolor the words from the dataset to the image's color
          # recolor just recolors the default colors to the image's blue color
          # interpolation is used to smooth the image generated
          plt.imshow(wc.recolor(color_func=image_colors),interpolation="hamming")

          plt.axis('off')
          plt.show()
```

Output:

The words repeated maximum number of times are printed with the greatest size.



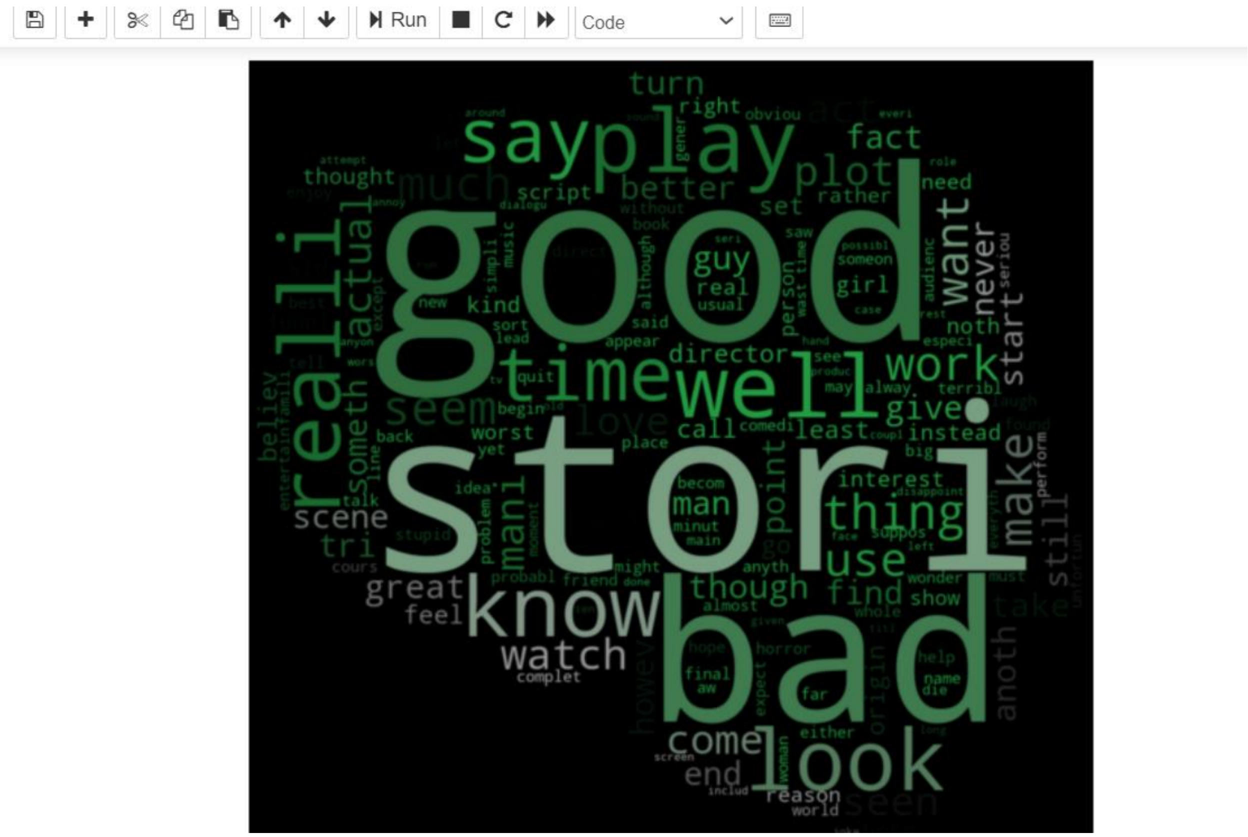
For negative reviews



```
In [35]:  all_words_negative = ' '.join(text for text in df['review'][df['sentiment']==1])

In [36]:  # combining the image with the dataset
          Mask = np.array(Image.open(requests.get('http://clipart-library.com/img/1364954.jpg', stream=True).raw))

          # We use the ImageColorGenerator library from Wordcloud
          # Here we take the color of the image and impose it over our wordcloud
          image_colors = ImageColorGenerator(Mask)

          # Now we use the WordCloud function from the wordcloud library
          wc = WordCloud(background_color='black', height=1500, width=4000,mask=Mask).generate(all_words_negative)

In [37]:  # Size of the image generated
          plt.figure(figsize=(10,20))

          # Here we recolor the words from the dataset to the image's color
          # recolor just recolors the default colors to the image's blue color
          # interpolation is used to smooth the image generated
          plt.imshow(wc.recolor(color_func=image_colors),interpolation="gaussian")

          plt.axis('off')
          plt.show()
```

Output:

The words repeated maximum number of times are printed with the greatest size.



CHAPTER 8 TESTING

8.1 Testing Plan

Testing process starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies the schedules, allocates the resources, and specified guidelines for testing. During the testing of the unit the specified test cases are executed and the actual result compared with expected output. The final output of the testing phase is the test report and the error report.

8.1.1 Test Data

Testing process begins with a test design. This arrangement recognizes all the testing related exercises that must be performed like the timetables, assigning the assets, and determining rules for testing. This testing of the unit of the predetermined experiments are executed and the genuine outcome is expected. The last part of the testing stage is the test report and the error report.

8.1.2 Unit testing

Every individual module has been tried against the necessity with some test.

8.1.3 Test Report

The module is working appropriately given the client must enter data. All information section frames have tested with indicated test cases and all information passage shapes are working properly.

8.1.4 Error Report

On the off chance that the client does not enter information in determined request, at that point the client will be incited with error messages. Error reduction is done to deal with the normal and sudden mistakes.

8.1.5 Accuracy

For every algorithm accuracy is very important and in field of medicine accuracy is given utmost importance. Our project related to text mining field is tested well and has an accuracy of 0.89

CHAPTER 9 CONCLUSION

9.1 Conclusion

Based on the results, it can be concluded that inputs i.e. reviews are very important to predict the sentiment behind the words. In conclusion of this case study, a machine learning model using Logistic Regression algorithm is built for a client whose purpose is to analyze the public response through sentiment analysis on a particular topic (on twitter platform). Sentiment analysis can be done by brands for market response on certain products, a film or book response, the sentiment around socialist movement etc.; can be observed by the client successfully and the generated reports and predictions can be effectively tackled by taking their respective domain related measures accordingly.

Example: Analysis performed on “feminist movement” in a case study taken from twitter dataset can be used to understand the extent of debatability of the topic and plan news programs and panels accordingly.

9.2 Future Scope

Future opinion-mining systems need broader and deeper common- and common-sense knowledge bases. This will lead to a better understanding of natural language opinions and will more efficiently bridge the gap between multimodal information and machine processable data. Blending scientific theories of emotion with the practical engineering goals of analysing sentiments in natural language text will lead to more bio-inspired approaches to the design of intelligent opinion-mining systems capable of handling semantic knowledge, making analogies, learning new affective knowledge, and detecting, perceiving, and “feeling” emotions.

CHAPTER 10 BIBILOGRAPHY

References

- Couresea Project on sentimental analysis using ML

- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
  https://en.wikipedia.org/wiki/Tf%E2%80%93idf#:~:text=In%20information%20retrieval%2C%20tf%E2%80%93idf,in%20a%20collection%20or%20corpus.
  https://towardsdatascience.com/social-media-sentiment-analysis-49b395771197

- https://en.wikipedia.org/wiki/Tf%E2%80%93idf#:~:text=In%20information%20retrieval%2C%20tf%E2%80%93idf,in%20a%20collection%20or%20corpus.

- https://towardsdatascience.com/social-media-sentiment-analysis-49b395771197