***Sentiment Analysis using product review data***

A Project Report

Submitted by

## G.ANURADHA (Regd.No.173J1A0541)

**K.SAI YESWANTH (Regd.No.173J1A0553)**

**CH.SAI SANDEEP (Regd.No.173J1A0530)**

**CH.VENKATA PAVAN (Regd.No.173J1A0527)**

*In partial fulfilment for the award of degree*

OF

# BACHELOR OF TECHNOLOGY IN

**COMPUTER SCIENCE AND ENGINEERING**

Under The Esteemed Guidance Of

**Mr.S.Om Prakash**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING RAGHU INSTITUTE OF TECHNOLOGY (AUTONOMOUS)**

(APPROVED BY AICTE, AFFILIATED TO JNTUK & ACCREDITED BY NBA & NAAC)

# DAKAMARI, VISAKHAPATNAM JULY 2021

**RAGHU INSTITUTE OF TECHNOLOGY::VISAKHAPATNAM (AUTONOMOUS)**

(APPROVED BY AICTE, AFFILIATED TO JNTUK & ACCREDITED BY NBA & NAAC)

# Department Of Computer Science and Engineering



**CERTIFICATE**

This is to certify that the project report entitled **“SENTIMENTAL ANALYSIS USING PRODUCT REVIEW DATA”** is the bonafide work of **“G.ANURADHA (173J1A0541), K.SAI YESWANTH (173J1A0553), CH. SAI SANDEEP (173J1A0530), and CH.VENKAT PAVAN (173J1A0527)”** that carried out the project work under my supervision.

## PROJECT GUIDE HEAD OF THE DEPARTMENT

**Mr.S.OM PRAKASH. Dr.S.ADINARAYANA, M.E.,PH.D.**

## ASSISTANT PROFESSOR PROFESSOR

**Department Of CSE Department Of CSE**

**Raghu Institute of Technology Raghu Institute of Technology**

**External Examiner**

**DECLARATION**

We hereby declare that this project entitled **“SENTIMENTAL ANALYSIS USING PRODUCT REVIEW DATA”** is the original work done by us in partial fulfilment of the requirement for the award of the Degree of Bachelor of Technology in Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada. This project works/project report has not been previously submitted to any other University/Institution for the award of other degree.

## G.ANURADHA (Regd.No.173J1A0541)

**K.SAI YESWANTH (Regd.No.173J1A0553)**

**CH.SAI SANDEEP (Regd.No.173J1A0530)**

**CH.VENKAT PAVAN (Regd.No.173J1A0527)**

**ACKNOWLEDGEMENT**

I would like to thank all the people who helped us in successful completion of the **SENTIMENTAL ANALYSIS USING PRODUCT REVIEW DATA**

I would like to thank **Sri Kalidindi Raghu Garu** Chairman of Raghu Institute of Technology, for providing necessary facilities.

I also thank **Mr. S. Satyanarayana,** Principal of Raghu Institute of Technology for extending his support and co-operation in providing the entire requirement for successful completion of the project.

I also pleased to thank **Dr. S. Adinarayana,** Professor and Head of the Department of Computer Science and Engineering for his constant inspiration, extensive help and valuable support in our step.

I would like to express our sincere gratitude to our project guide, **Mr. S.Om Prakash,** Professor in the Department of Computer Science and Engineering for his whole hearted guidance, encouragement and continuous support throughout this work.

We sincerely thank all the staff members of the Computer Science Department for their sustained guidance during our work.

We thank all those who contributed directly or indirectly in successfully carrying out this project work.

# ABSTRACT

Sentiment analysis or opinion mining is one of the major tasks of NLP (Natural Language Processing). Sentiment analysis has gained much attention in recent years. In this paper, we aim to tackle the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis. A general process for sentiment polarity categorization is proposed with detailed process descriptions.

Data used in this study are online product reviews collected from Amazon.com. Experiments for both sentence-level categorization and review-level categorization are performed with promising outcomes. At last, we also give insight into our future work on sentiment analysis. Sentiment analysis is defined as the process of mining of data, view, review or sentence to predict the emotion of the sentence through natural language processing (NLP). The sentiment analysis involve classification of text into three phase “Positive”, “Negative” or “Neutral”.

It analyzes the data and labels the ‘better’ and ‘worse’ sentiment as positive and negative respectively. Thus, in the past years, the World Wide Web (WWW) has become a huge source of raw data generated custom or user. Using social media, e-commerce website, movies reviews such as Facebook, twitter, Amazon, Flipkart etc. user share their views, feelings in a convenient way. In WWW, where millions of people express their views in their daily interaction, either in the social media or in e-commence which can be their sentiments and opinions about particular thing.

These growing raw data are an extremely high source of information for any kind of decision making process either positive or negative. To analysis of such huge data automatically, the field of sentiment analysis has turn up. The main aim of sentiment analysis is to identifying polarity of the data in the Web and classifying them.

Sentiment analysis is text based analysis, but there are certain challenges to find the accurate polarity of the sentence. This states that there is need to find the better solution to get much better results than the previous approach or technique used to find polarity of sentence.

Therefore, to find polarity or sentiment of, user or customer there is a demand for automated data analysis techniques. In this paper, a detailed survey of different techniques or approach is used in sentiment analysis and a new technique which is proposed in this paper.

Sentiment analysis or opinion mining is the computational study of people’s emotions, opinions, sentiments by considering their reviews in the form of text in recent years. This is the most active research area in the field of natural language processing and text mining. Since it is based on the opinions and as all the humans make decisions dependent on other opinions its popularity is increasing day by day. In this paper, our ultimate goal is to tackle the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis.

A complete process for sentiment polarity categorization is proposed with detailed process descriptions. The datasets that are used here comprise of online product reviews collected from Amazon.com. Necessary experiments for both sentence-level categorization and review-level categorization are performed with the promising outcomes. At last, we also give insight into the future scope of sentiment analysis.

Therefore, this project deals with analysis of Products data getting from the UCI Depositories, DataWorld and Jmcauley where we need products reviews and their customers names

OBJECTIVES:

1. Loading the Data

2. Exploring the Data

3. Preparing the Text Data

4. Splitting the Data

5. Creating Word Dictionary

6. Feature Extraction Process

7. Training the Model or Classifier

8. Generating the Model

9. Model Evaluation

## SCOPE OF THE PROJECT:

## • Sentiment analysis or opinion mining is the computational study of people’s emotions, opinions, sentiments by considering their reviews in the form of text.

## • To build a predictive model which can be used to forecast the prediction of the text and give us a good result.

## • Necessary experiments for both sentence-level categorization and review-level categorization are performed with the promising outcomes.

## TECHNIQUES AND METHODS:

• Preprocessing techniques

• Feature selections

• Classification algorithms in ML

• Prediction based on attributes and performance

## SYSTEM CONFIGURATIONS:

**Hardware requirements:**

* + RAM - Minimum 4 GB
  + Hard disk - Minimum 100 GB

## Software requirements:

* + 1.Language - Python 3.6
  + 2.IDE – Python 3.6

# INDEX

|  |  |
| --- | --- |
| DESCRIPTION |  |
| ACKNOWLEDGEMENT |  |
| ABSTRACT |  |
| **LIST OF FIGURES** |  |
| **INTRODUCTION** |  |
| **LITERATURE SURVEY** |  |
| **SOFTWARE REQUIREMENT ANALYSIS** |  |
| **SYSTEM DESIGN** |  |
| **CODING** |  |
| **TESTING** |  |
| **OUTPUT** |  |
| **CONCLUSION** |  |
| **REFRENCES** |  |

# CHAPTER-1 INTRODUCTION

Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people’s sentiments towards certain entities. Internet is a resourceful place with respect to sentiment information. From a user’s perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites.

From a researcher’s perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers.

For instance, Twitter currently has three different versions of APIs available, namely the REST API, the Search API, and the Streaming API. With the REST API, developers are able to gather status data and user information; the Search API allows developers to query specific Twitter content, whereas the Streaming API is able to collect Twitter content in real-time. Moreover, developers can mix those APIs to create their own applications. Hence, sentiment analysis seems having a strong fundament with the support of massive online data.

However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. For example, instead of sharing topic-related opinions, online spammers post spam on forums. Some spams are meaningless at all, while others have irrelevant opinions also known as fake opinions.

The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral. The Stanford Sentiment 140 Tweet Corpus is one of the datasets that has ground truth and is also public available. The corpus contains 1.6 million machine-tagged Twitter messages. Each message is tagged based on the emoticons (☺as positive, ☹as negative) discovered inside the message).

Data used in this paper is a set of product reviews collected from Amazon, between February and April, 2014. The aforementioned flaws have been somewhat overcome in the following two ways: First, each product review receives inspections before it can be posted.

Second, each review must have a rating on it that can be used as the ground truth. The rating is based on a star-scaled system, where the highest rating has 5 stars and the lowest rating has only 1 star.



This paper tackles a fundamental problem of sentiment analysis, namely sentiment polarity categorization. Figure 2 is a flowchart that depicts our proposed process for categorization as well as the outline of this paper. Our contributions mainly fall into Phase 2 and 3.

In Phase 2:

1) An algorithm is proposed and implemented for negation phrases identification.

2) A mathematical approach is proposed for sentiment score computation.

3) A feature vector generation method is presented for sentiment polarity categorization.

In Phase 3:

1) Two sentiment polarity categorization experiments are respectively performed based on sentence level and review level.

2) Performance of three classification models are evaluated and compared based on their experimental results.

**MOTIVATION**

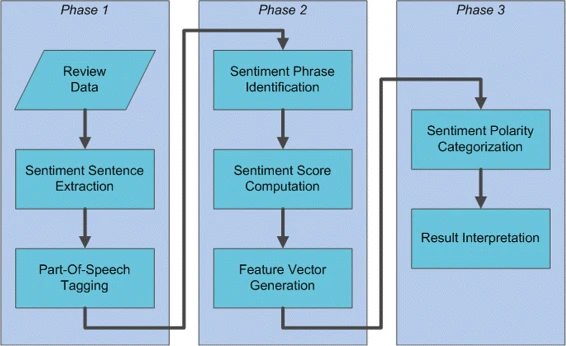
Opinions of others on a particular product can have influence on our decision. In olden days people used to collect opinions on a product from direct sources like friends, relatives, consumer reports and strangers. Now-a-days we have several ways to have a look at opinions of others (ex: Internet). Internet helps to collect opinions from different people around the world. People look to different e-commerce sites like amazon, eBay, flipkart, snapdeal etc., to know reviews and feedback about a particular product and will buy the product according to the reviews. As social media has a great influence, now-a-days those are also helping to know about a product.

Some Organizations use surveys, opinion polls, and social media as the mechanism to obtain feedback on their products. Sentimental analysis deals with the collective study of sentiments, opinions and emotions expressed in the form of text.

As there can be thousands of reviews and different feedback on a particular product, one cannot go through all those reviews to make decision so the concept of sentimental analysis arises. Sentimental analysis helps to characterize the reviews into different levels like good, bad, worst, average by considering and analysing the words present in those reviews.

It helps to:

* Characterize reviews into different types based on the words used to describe the product.
* Outputs a pictographically representation that gives a clear idea about the product.
* It considers thousands of reviews. So, our decision will be correct to maximum extent.
* One can get out of confusion regarding the product.
* The company can know the overall opinion on the product and change according to the reviews
* The customers can also know the quality of the product and will buy the best product



* + Every single day huge amount of information, reviews or opinions are getting stored in the websites of social media or Eservices in the form of raw data. To work with those raw data proper methods required.
  + Most of the methods either focus on verbs, nouns, adverbs or adjectives. Although a recent study has shown that combination of adverbs and adjectives in sentiment analysis is better than adjectives alone.
  + But no work has focused on all the possible combinations of adverbs, adjectives and verbs. This paper presents the Theoretical analysis of some well-known methods or proposal of Sentiment Analysis.
  + Both the advantages and disadvantages of the discussed methods are considered to add new features in the proposed approach.
  + The new approach follows machine learning technique at document level with combination of adjectives, adverbs, and verbs.
  + The following combinations are taken into for analysis, adverbs-adjectives.
  + The Standard classifier like Naive Bayes (NB), Support Vector Machine (SVM) are used to deduct result and for analysis.
  + This section presents the classification of Sentiment Analysis followed by detailed revision of the existing methods related to sentiment analysis.

**OBJECTIVE OF THE PROJECT**

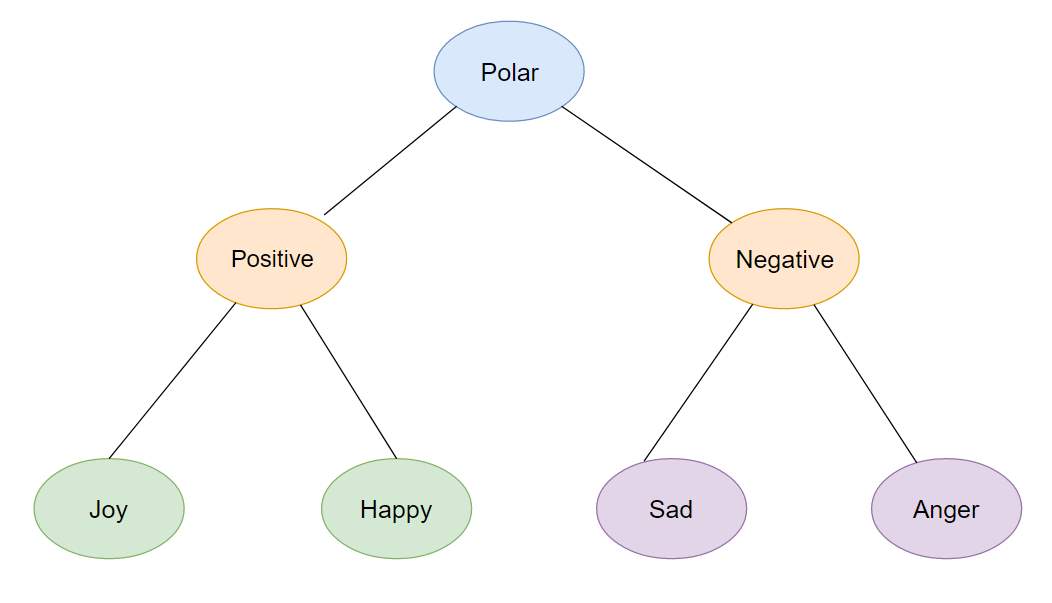
The main objective of this project is to go about an extra mile to provide the users with an output that is the analysis of thousands and thousands of reviews.

To save time by analysing thousands of reviews in short period and if those reviews were analysed manually it may take up to decades.

The overall output is achieved by several steps like:

* Step 1: Gathering the raw data from different repositories.
* Step 2: Pre-processing the raw data in a usable format.
* Step 3: Fetching the data set file using R language.
* Step 4: Identifying the sentiment words in the reviews.
* Step 5: Maintaining a count of different type of words.
* Step 6: Data visualization in a bar graph.

Ex:



# CHAPTER-2 LITERATURE SURVEY

One fundamental problem in sentiment analysis is categorization of sentiment polarity. Given a piece of written text, the problem is to categorize the text into one specific sentiment polarity, positive or negative (or neutral).

Based on the scope of the text, there are three levels of sentiment polarity categorization, namely the document level, the sentence level, and the entity and aspect level. The document level concerns whether a document, as a whole, expresses negative or positive sentiment, while the sentence level deals with each sentence’s sentiment categorization; The entity and aspect level then targets on what exactly people like or dislike from their opinions.

Since reviews of much work on sentiment analysis have already been included in, in this section, we will only review some previous work, upon which our research is essentially based. Hu and Liu summarized a list of positive words and a list of negative words, respectively, based on customer reviews. The positive list contains 2006 words and the negative list has 4783 words. Both lists also include some misspelled words that are frequently present in social media content.

Sentiment categorization is essentially a classification problem, where features that contain opinions or sentiment information should be identified before the classification. For feature selection, Pang and Lee suggested to remove objective sentences by extracting subjective ones.

They proposed a text-categorization technique that is able to identify subjective content using minimum cut. Gann et al. selected 6,799 tokens based on Twitter data, where each token is assigned a sentiment score, namely TSI (Total Sentiment Index), featuring itself as a positive token or a negative token. Specifically, a TSI for a certain token is computed as:

𝑇𝑆𝐼=𝑝−𝑡𝑝𝑡𝑛9×𝑛𝑝+𝑡𝑝𝑡𝑛∗𝑛TSI = p−tptn×np+tptn∗n

Where *p* is the number of times a token appears in positive tweets and *n* is the number of times a token appears in negative tweets. tptn is the ratio of total number of positive tweets over total number of negative tweets.

Feilong Tang suggested ‘2’ generative model, MaxEnt–JABST as well as JABST, that extracted typically the fine-grained opinions along with aspects as of reviews (online).

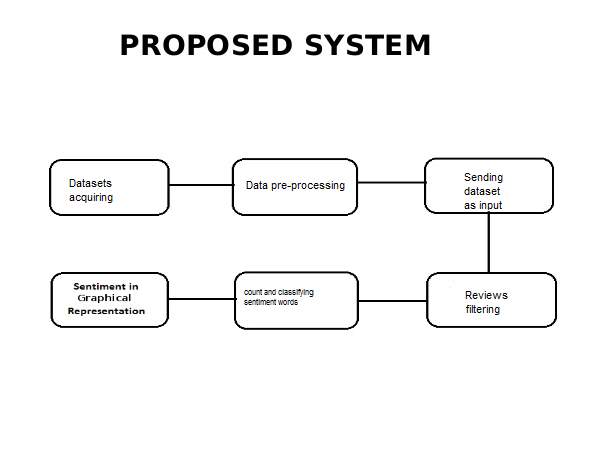
The JABST model extracted particular and general opinions and aspects together with the sentiment polarity (SP).

In addition, the MaxEnt–JABST design added a maximal entropy classifier for separating aspects or opinion words more precisely. Those designs were assessed on review regarding restaurants and electronic devices quantitatively as well as qualitatively.

The experiential outcomes evinced that the designs outperformed existent baselines and were competent to recognize fine-grained aspects and opinions but the improvement was still needed.

**Proposed System:**

* Rajkumar rendered a ‘2’ ML approaches say Naïve Bayes (NB) and SVM for performing SA on reviews of a specific product.
* In those approaches, the dataset was gathered as of Amazon, which comprised reviews regarding Laptops, Cameras, Mobiles, Tablets, video surveillance, and TVs. Subsequently, stemming, stop word removal, and also punctuation marks removal were executed and it was transmuted into a bag of words.
* This dataset was contrasted to opinion lexicons, that is, 4783 negative and 2006 positive words with sentiment scores intended for every sentence were evaluated.
* Utilizing score and disparate features, the NB along with SVM were employed and diverse accurateness was computed.



***PROPOSED ALGORITHM FOR SENTIMENT ANALYSIS***

* ***Data Filtration***:

Data filtration importing all positive and negative datasets from file and combining them into a single file. The data sets may contain lots of unwanted symbols, and number. These factors need to be corrected or solved to increase the efficiency. Therefore, in this process the unwanted symbols and number are removed

* ***Training Model***:

Fetching the datasets from the file and extracting all the corresponding words (feature words) like adjective, adverb and verb. Then datasets are labelled a respectively as “pos” for positive and “neg” for negative.

Then performing frequency distribution over collected words and selecting 5000 words for training. Again, the shuffling of data is performed using random seed for better training. Here, the labeled datasets are divided into the percentile of 70-30% for training and testing, respectively.

Training dataset to classification algorithms like Naïve Bayes classification algorithm, SVM algorithm.

* ***Testing Model***:

Here user can test and analysis the respective model by performing preprocessing over the input data. The preprocessing contains the removal of the symbol and number.

Mapping to user input using saved featured (based on training dataset). Then feed to saved model for prediction.

# CHAPTER-3

# SOFTWARE REQUIREMENT ANALYSIS

## PYTHON

**History of Python:**

Guido van Rossum developed Python in the early nineties at the National Research Institute for Mathematics and Computer Science, Netherlands. Python is a derivation of many other languages, namely ABC, Modula-3, C, C++, Algol68, Smalltalk, and Unix shell and few other scripting languages. Python has been copyrighted immediately after its establishment. Python source code is open to all with the GNU General Public License (GPL). A dedicated technical team works for the development of Python while the major decisions regarding the technology are still taken by Mr Guido van Rossum.

## Operations using NumPy:

NumPy (Numerical Python) is a python library that provides high level functions to operate mathematically on arrays. It consists of multidimensional array objects and a compilation of methods for processing array.

Using NumPy, the following operations can be performed − Mathematical and logical operations.

Fourier transforms and routines, which are used for shape manipulation. Operations related to linear algebra. NumPy also has preprocessed functions for linear algebra and random number generation.

## NAIVE BAYES ALGORITHM

Naive Bayes classifier is an algorithm that is based on the Bayes theorem. It has a substantial independence assumption. It is also called "independent feature model". It presupposes the presence or absence of a particular feature of a class is irrelevant to the presence or absence of any other feature in a supposed class. Naïve Bayes classifier can be trained in a supervised learning model too. It applies a method of highest similarity. It has worked in a complicated real-world problem. It needs only a small quantity of training data. It determines parameters for classification. The variance of the variable needs to be calculated for each class and not the entire matrix. Naïve Bayes is mainly used when the inputs are high. It gives output in a better sophisticated form. Every attribute's probability is displayed in the Prediction table. Machine learning and data mining methods are based on naïve bayes classification.

## Bayes theorem:

P(H|X) = P(X|H) P(H)

P(X)

* + Where P(H|X ) is the posterior probability of H conditioned on X
  + P(X|H) is the posterior probability of X conditioned on H
  + P(H)is the prior probability of H
  + P(X) is the prior probability of X

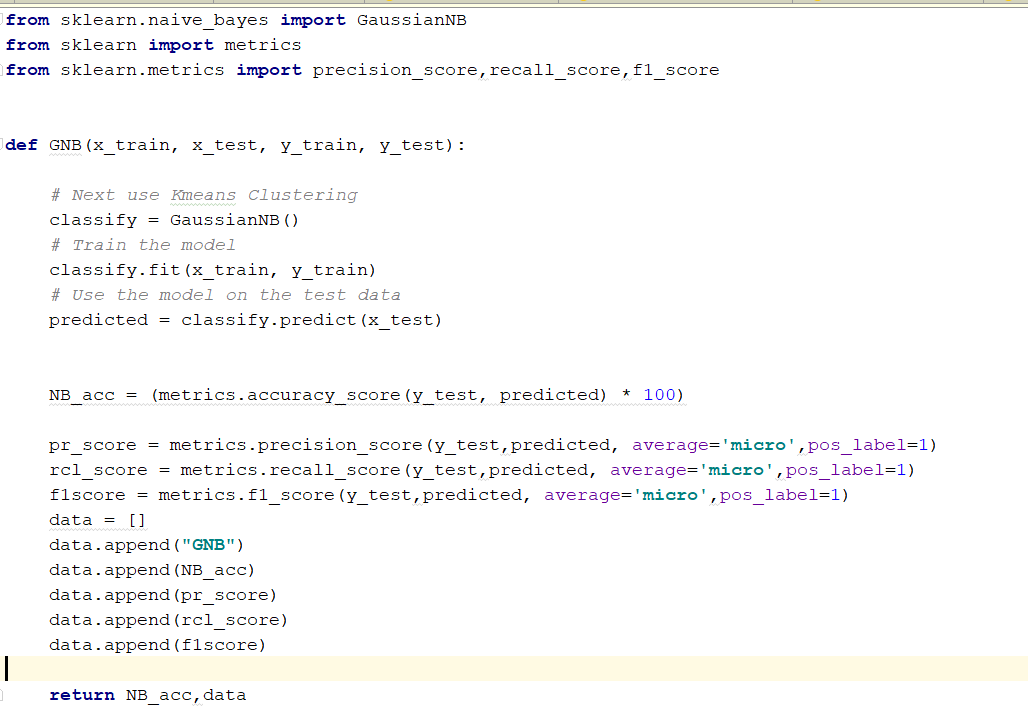
**Advantages and Disadvantages:**

**Advantages:**

* It is fast when compared to other algorithms.
* Simple to understand.
* Simple to build.
* Not sensitive to other (irrelevant) features.

**Disadvantages:**

* Since it considers each and every feature as independent, it is not correct in all cases.



*Image 1 Gaussian Naive Bayes code*

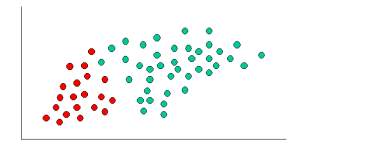
***NAIVE BAYES CLASSIFIERS***

##### Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.

* It is mainly used in *text classification* that includes a high-dimensional training dataset.
* It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
* Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

**How it works?**

* Generally all the machine learning algorithms should be trained for supervised learning tasks like classification, prediction etc and even for unsupervised learning tasks like clustering.
* The word training it specifies that train them on some particular inputs so that we can test them for new inputs for which they may classify or predict based on the learning. This is the technique used by most of the machine learning algorithms.
* So in a machine learning project we have to divide our input set into two sets, one is development set which consists of training and development test data and the other set is test set or evaluation set for which we have to make classification or prediction.
* The test set need to have the format that is same as training set. And if the use the same test set as training set then the scores will be obviously high which is not correct. Most of the test set should be different from training set, so that we get genuine results.
* **Simple demonstration:**



* As indicated by the fig  the objects can be classified as either red or green. Our aim to classify new cases as they arrive i.e., either they belong to red or they belong to green.
* Since there are more green objects than red generally one thinks that the probability of new object is green. This belief is treated as prior probability in Bayesian Theorem. Prior probabilities are based on previous experiences; in this case we consider the percentages of red and green objects present.
* **Prior probability of green:**
* Number of green objects/total number of objects.
* **Prior probability of red:**
* Number of red objects/total number of objects.
* Since there are total number of 60 objects in which red are 20 and green ae 40, our prior probabilities will look like this:
* **Prior probability of green:**40/60
* **Prior probability of red:**20/60

HOW DOES NAIVE BAYES CLASSIFIER WORKS

#### Convert the given dataset into frequency tables.

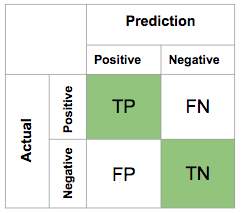
* 1. Generate Likelihood table by finding the probabilities of given features.
  2. Now, use Bayes theorem to calculate the posterior probability.
  3. **For example:**

A fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter.

* 1. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

**Confusion Matrix:**

A confusion matrix is a representation of table which is used to describe the performance of a classification model like naïve bayes classifier on a set of test data for which the true values are known like trained data. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.



# CHAPTER-4 SYSTEM DESIGN

**System Design**

We propose the following system design (shown in Figure 1). Our design has two major steps: extracting an initial set of features and performing feature selection. These steps are used to carry out sentiment classification of forum messages.

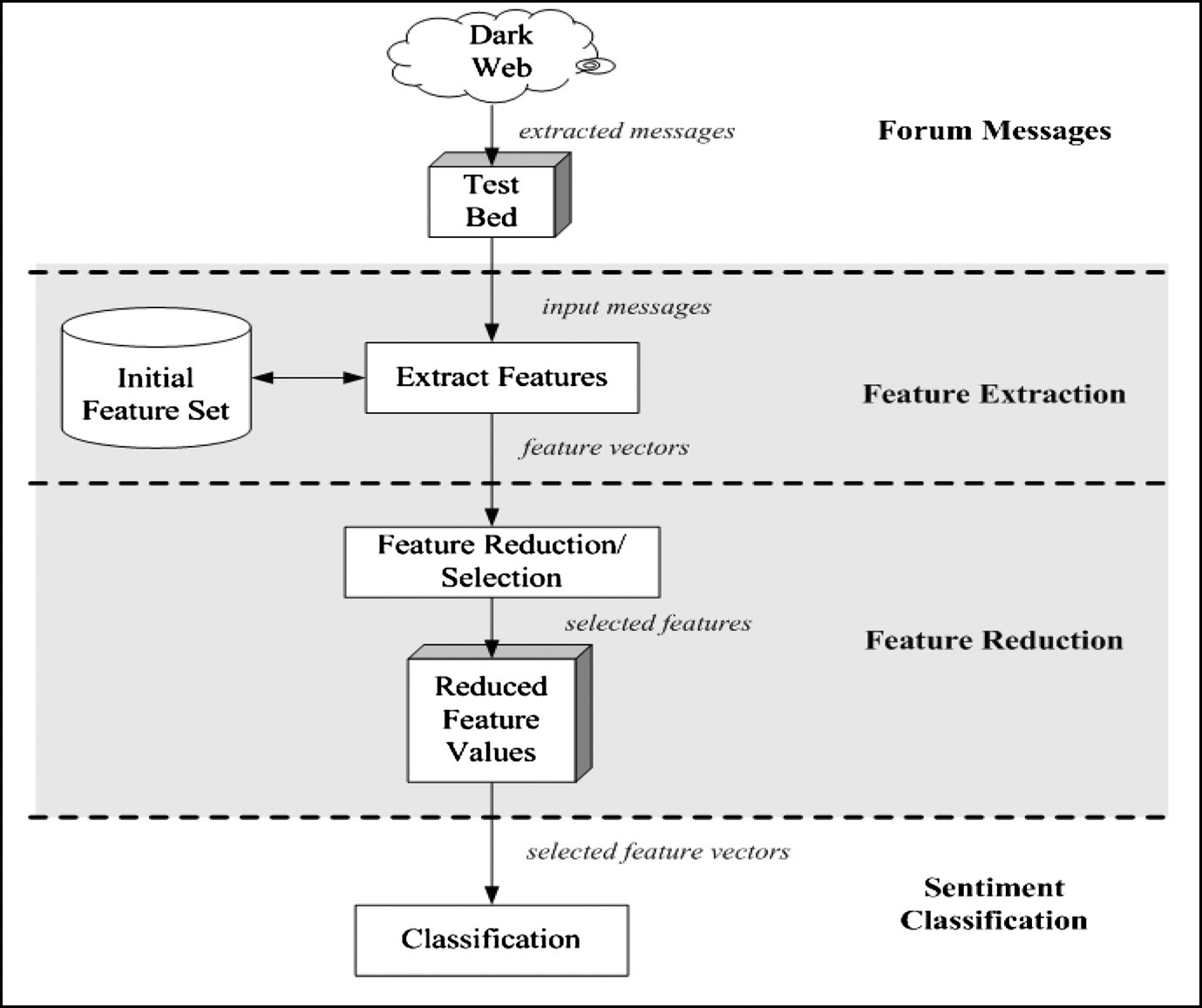
**Feature Extraction:**

We incorporated syntactic and stylistic features in our sentiment classification attribute set. These features are more generic and applicable across languages. For instance, syntactic, lexical, and structural features have been successfully used in stylometric analysis studies applied to English, Chinese [Peng et al. 2003; Zheng et al. 2006], Greek [Stamamatos et al. 2003], and Arabic [Abbasi and Chen 2006, 2005].

Link-based features were not included, since our messages were not in sequential order (insufficient cross-message references). These types of features are only effective where the testbed consists of entire threads of messages and message referencing information is available. Semantic features were not used since these attributes are heavily context dependent [Pang et al. 2002]. Such features are topic- and language specific. For example, the set of positive-polarity words describing a good movie may not be applicable to discussions about racism.

Unlike stylistic and syntactic features, semantic features such as manually crafted lexicons incorporate an inherent feature selection element via human involvement. Such human involvement makes semantic features (e.g., lexicons and dictionaries) very powerful for sentiment analysis.

Lexicon developers will only include features that are considered important and will weight these features based on their significance, thereby reducing the need for feature selection.

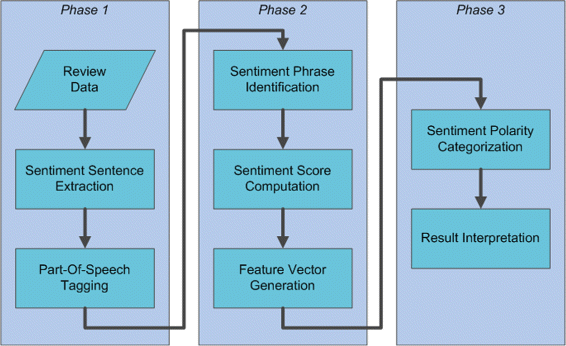


For example, Whitelaw et al. used WordNet to construct an initial set of features which were manually filtered and weighted to create the lexicon. Unfortunately, the language specificity of semantic features is particularly problematic for application to the Dark Web, which contains text in dozens of languages [Chen 2006].

We hope to overcome the lack of semantic features by incorporating feature selection methods intended to isolate the important subset of stylistic and syntactic features and remove noise.

Determining size of initial feature set. Our initial feature set consisted of 14 different feature categories which included POS tag n-grams (for English), word roots (for Arabic), word n-grams, and punctuation for syntactic features. Style markers included word- and character-level lexical features, word-length distributions, special characters, letters, character n-grams, structural features, vocabulary richness measures, digit n-grams, and function words. The word-length distribution includes the frequency of 1- to 20-letter words.

**Sentiment Polarity Categorization Process.**



**OUR APPROACH:**

**Sentimental Analysis using Supervised Learning**

Step 1:

Data Pre-processing: After the data has been selected, it needs to be pre-processed using the given steps:

1. Formatting the data to make it suitable for ML (structured format).

2. Cleaning the data to remove incomplete variables.

3. Sampling the data reduces the run time for algorithms and memory requirements.

Step 2:

Tokenization: The process of breaking a stream of text up into phrases, words, symbols, or other meaningful elements called tokens. The goal of the tokenization is the exploration of the words in a sentence.

Step 3:

Stop-word Elimination: The most common words that unlikely to help text mining such as prepositions, articles, and pro-nouns can be considered as stop words. Since every text document deals with these words which are not necessary for application of text mining.

All these words are eliminated. A new list of stop words was created that eliminated only those words that did not contribute to opinion mining.

Step 4:

Bag-of-words Model: The bag-of-words model is one of the simplest language models used in NLP. It makes a unigram model of the text by keeping track of the number of occurrences of each word. This can later be used as features for Text Classifiers.

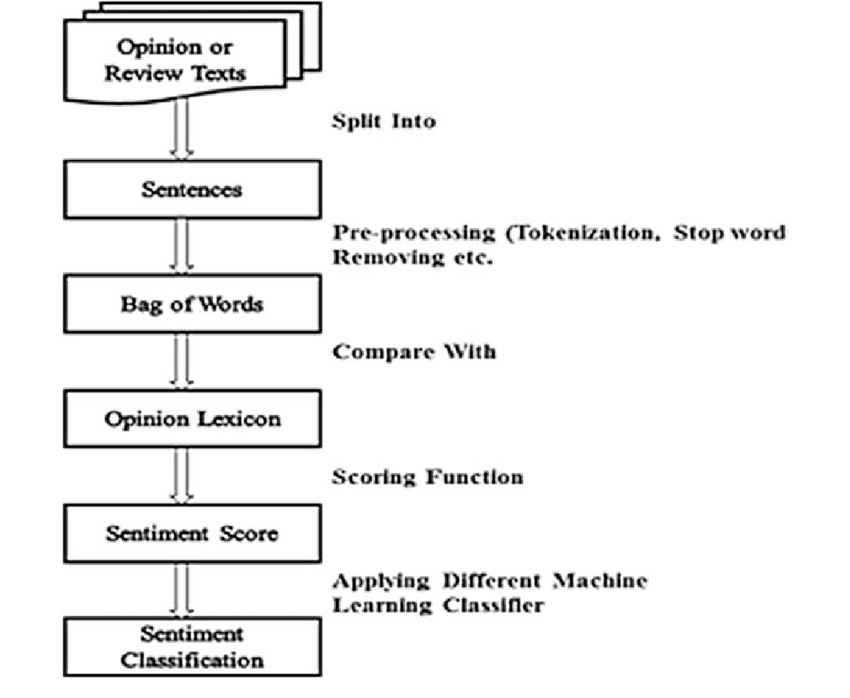
In this bag-of-words model you only take individual words into account and give each word a specific subjectivity score.

Step 5:

Training the classifier: We are training the classifier using the Features Extracted using the Bag-of Words Model. The Features of both the training and test dataset are com- pared. And this is giving to the classifier to give the predictions on the test data.

Step 6:

Sentimental Analysis: For sentimental analysis we are using the Decision tree classifier and Naive Bayes and comparing the results. We also see which classifier has the most accuracy.

****

**System Design**

The System Design Document specifies the requirements of the system, operating system environment, subsystem architecture, files and design of the database, input and output formats, output layouts, human-machine interfaces, detailed design of the system, processing logic, and other external interfaces.

**USE CASE DIAGRAM:**

Use case: A use case can be described as a specific way of using the system from a user’s (actor’s) perspective Actor: Actor symbolizes the role that a user plays in the complete system.

An actor associate with the use cases, but has no control over them.

**An actor is someone who**

• interacts with the system.

• Gives input to the system and receives information from the system.

• Is external and has completely no control over the use cases, whatsoever.

**FLOW OF EVENTS:**

The flow of events is a chain of transactions executed by the system. They include complete information, addressed in terms of what the system does and not how the system performs the task.

The flow of events is formulated as separate documents in your preferred text editor and can be then joined to a use case utilizing the Files tab of a model component.

## dd4.png

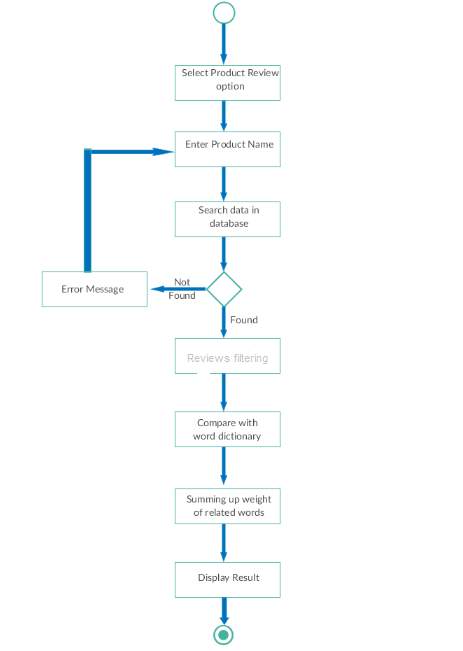
## SEQUENCE DIAGRAM:

A sequence diagram is a graphical view of a situation that shows the interaction of the object in a time-based sequence of what happens in chronological order. Sequence diagrams set the role of an object and support in providing fundamental information to define the class capacities and interfaces

An object has three main basic features, namely, state, behavior, and identity. The common class will define the structure and behavior of similar objects. Each object in a diagram symbolizes some instance of the corresponding class. An object with no name is mentioned as a class instance.

A message forms the communication which is carried between two objects that initialize an event.

A link subsists between two objects, including the class utilities, only if there is a connection between their corresponding classes.



## COLLABORATION DIAGRAM:

Collaboration diagrams show how objects associate with each other. It is also called the communication diagram, which is modified into a simpler version of the collaboration diagram in the newer versions.

## DFD DIAGRAM:

A data flow diagram is graphical tool used to explain and interpret the flow of data through a system.

It is a central tool and the source from which all other elements are developed. The conversion of data from input to output can be described as physical components associated with the system which is logical and independent. These diagrams are called as the logical data flow diagrams.

The concept behind the discharge of a process into several processes understands that at a level of detail is discharged into higher detail at the next level.

This is performed until further discharge is required and a sufficient amount of detail is defined for analyst to know the process.

A DFD is also referred to as the “bubble Chart”, which has a purpose of making clear the system requirements and to identify significant transformations that will be programs in the system design.

# CHAPTER-5 CODING

**Main.py**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.metrics import accuracy\_score, confusion\_matrix

import string

from sklearn.feature\_extraction.text import TfidfVectorizer

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer

from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import MultinomialNB

import warnings

warnings.filterwarnings('ignore')

import pickle

def plot\_confusion\_matrix(y\_true, y\_pred):

cm = confusion\_matrix(y\_true, y\_pred)

plt.figure(figsize=(4,4))

sns.heatmap(cm, cbar=False, cmap='viridis', annot=True, fmt='.0f')

plt.title("Confusion matrix")

plt.xlabel("Predicted Label")

plt.ylabel("True Label")

return plt.show()

**# Accuracy Scorer**

def get\_accuracy\_score(model,X\_train, y\_train, X\_test, y\_test, return\_model=False):

model = model.fit(X\_train, y\_train)

y\_preds\_train = model.predict(X\_train)

y\_preds = model.predict(X\_test)

print("Train accuracy:", accuracy\_score(y\_train, y\_preds\_train))

print("Test accuracy:", accuracy\_score(y\_test, y\_preds))

print()

return model if return\_model==True else None

data = pd.read\_csv('Reviews.csv')

data.head(3)

data.isna().sum().to\_frame(name='# Missing values')

total\_rows = data.shape[0]

data.dropna(how='any',inplace=True)

remaining\_rows= data.shape[0]

removed\_rows = total\_rows-remaining\_rows

a = data.shape[0]

data = data.loc[data.Score != 3]

b = data.shape[0]

data.loc[:, "Sentiment"] = data.Score.apply(lambda x : 1 if x > 3 else 0)

data = data[["Text", "Sentiment"]]

a = data.shape[0]

data.drop\_duplicates(inplace=True)

b = data.shape[0]

neg\_data = data.loc[data.Sentiment == 0]

**# positive reviews**

pos\_data = data.loc[data.Sentiment == 1][:neg\_data.shape[0]]

**# balanced data**

a = data.shape[0]

data = pd.concat([pos\_data, neg\_data])

data = data.sample(frac=1, random\_state=1)

b = data.shape[0]

total\_stopwords = set(stopwords.words('english'))

negative\_stop\_words = set(word for word in total\_stopwords if "n't" in word or 'no' in word)

final\_stopwords = total\_stopwords.symmetric\_difference(negative\_stop\_words)

stemmer = PorterStemmer()

def preprocessor(review):

HTMLTAGS = re.compile('<.\*?>')

review = HTMLTAGS.sub(r'', review)

table = str.maketrans(dict.fromkeys(string.punctuation))

review = review.translate(table)

remove\_digits = str.maketrans('', '', string.digits)

review = review.translate(remove\_digits)

review = review.lower()

MULTIPLE\_WHITESPACE = re.compile(r"\s+")

review = MULTIPLE\_WHITESPACE.sub(" ", review).strip()

review = [word for word in review.split() if word not in final\_stopwords]

review = ' '.join([stemmer.stem(word) for word in review])

return review

def create\_model():

X = data['Text']

y = data['Sentiment']

X = X.apply(preprocessor)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=1)

vectorizer = TfidfVectorizer(max\_features=15000)

vectorizer.fit(X\_train)

tfidf\_X\_train = vectorizer.transform(X\_train)

tfidf\_X\_test = vectorizer.transform(X\_test)

final\_model= get\_accuracy\_score(MultinomialNB(),tfidf\_X\_train, y\_train, tfidf\_X\_test, y\_test,return\_model=True)

y\_pred = final\_model.predict(tfidf\_X\_test)

plot\_confusion\_matrix(y\_test, y\_pred)

with open("tfidf\_vectorizer.pkl", "wb") as f:

pickle.dump(vectorizer, f)

with open("model.pkl", "wb") as f:

pickle.dump(final\_model, f)

**#predict review**

import pickle

from nlp import preprocessor

with open("tfidf\_vectorizer.pkl", "rb") as f:

vectorizer = pickle.load(f)

with open("model.pkl", "rb") as f:

model = pickle.load(f)

def pred(text):

review = preprocessor(text)

x = vectorizer.transform([review])

y = model.predict(x)[0]

return y

while(1):

s = input("Enter review: ")

if(s == "exit"):

break

print(pred(s))

# CHAPTER-6 TESTING

Testing is a debugging process, which is one of the most significant aspects of the triggers in programming. Without a proper working program, the system will never present the desired output. Testing is most beneficial when user development assists in the process of identifying all errors and bugs.

A part of the sample data is used for testing, which is called the testing data. It is not quantity but the quality of the data used that matters in the testing. Testing is aimed at guaranteeing that the system is accurate and efficient.

## Purpose of Testing

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn’t work. A primary purpose of testing is to detect software failures so that defects may be discovered and corrected.

Testing cannot establish that a product functions properly under all conditions, but only that it does not function properly under specific conditions. The scope of software testing may include the examination of code as well as the execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do.

In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

## Testing Objectives

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. To evaluate the work products such as requirements, design, user stories, and code.To verify the fulfillment of all specified requirements.To validate if the test object is complete and works as per the expectation of the users and the stakeholders.

## The basic levels of Testing:

Client needs acceptance testing

Requirements system testing

Design integration testing

Code unit testing

## Unit Testing

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an objectoriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. A Unit is a smallest testable portion of system or application which can be compiled, liked, loaded, and executed.

This kind of testing helps to test each module separately. The aim is to test each part of the software by separating it. It checks that component is fulfilling functionalities or not. This kind of testing is performed by developers.

## Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed.

Integration means combining. For Example, In this testing phase, different software modules are combined and tested as a group to make sure that integrated system is ready for system testing.

Integrating testing checks the data flow from one module to other modules. This kind of testing is performed by testers.

## System Testing

System testing tests a completely integrated system to verify that the system meets its requirements. System testing is performed on a complete, integrated system. It allows checking system's compliance as per the requirements.

It tests the overall interaction of components. It involves load, performance, reliability and security testing. System testing most often the final test to verify that the system meets the specification. It evaluates both functional and non-functional need for the testing.

## Acceptance Testing

Acceptance Testing is a test conducted to determine if the requirements of a specification or contract are met.

Acceptance testing is a test conducted to find if the requirements of a specification or contract are met as per its delivery. Acceptance testing is basically done by the user or customer. However, other stockholders can be involved in this process.

Each Module can be tested using the following two approaches:

* + Black Box Testing
  + White Box Testing

## Black Box Testing:

Black box testing is a software testing technique in which the functionality of the Software under Test (SUT) is examined without studying the internal code, implementation details and information of inner paths of the software. This type of testing is done on the basis completely of the software requirements and specifications. In Black Box Testing we just concentrate on inputs and output of the software system without fretting about inner knowledge of the software program.

## White Box Testing:

The white box testing is one of two components of the "box testing" approach of software testing. Its counterpart, black box testing, includes testing from an external or end-user type prospect. While Whitebox testing is on the basis of the internal workings of an application and rotates around internal testing. The term "white box" was used because of the see-through box theory. The clear box or white box name signifies the capability to see through the software's outer shell (or "box") into its interior functioning.

## Importance of Software Testing

Few can argue against the need for quality control when developing software. Late delivery or software defects can damage a brand’s reputation — leading to frustrated and lost customers. In extreme cases, a bug or defect can degrade interconnected systems or cause serious malfunctions.

Though testing itself costs money, companies can save millions per year in development and support if they have a good testing technique and QA processes in place. Early software testing uncovers problems before a product goes to market. The sooner development teams receive test feedback, the sooner they can address issues such as:

* Architectural flaws
* Poor design decisions
* Invalid or incorrect functionality
* Security vulnerabilities
* Scalability issues

When development leaves ample room for testing, it improves software reliability and high-quality applications are delivered with few errors. A system that meets or even exceeds customer expectations leads to potentially more sales and greater market share.

## DATA SETS USED FOR THE PROJECT:

An informational index is an accumulation of information that depicts quality qualities (factors) of various true protests (units). With information that are in fact amend, we comprehend an informational index where each esteem

1. Can be straightforwardly perceived as having a place with a specific variable;

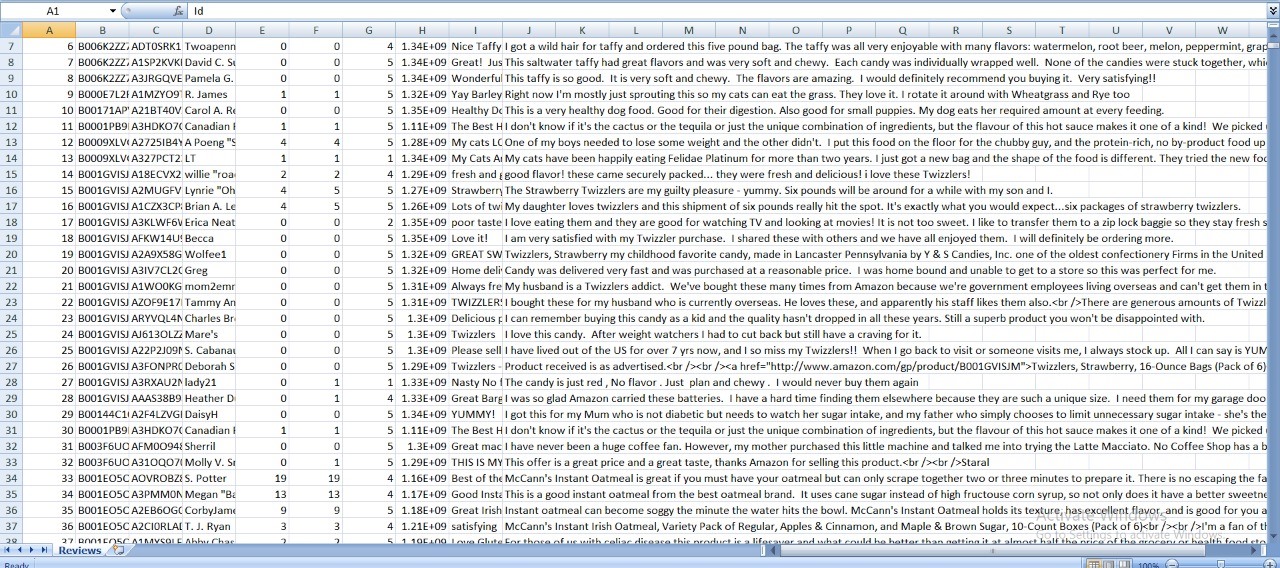
2. Is put away in an information sort that speaks to the esteem space of this present reality variable.

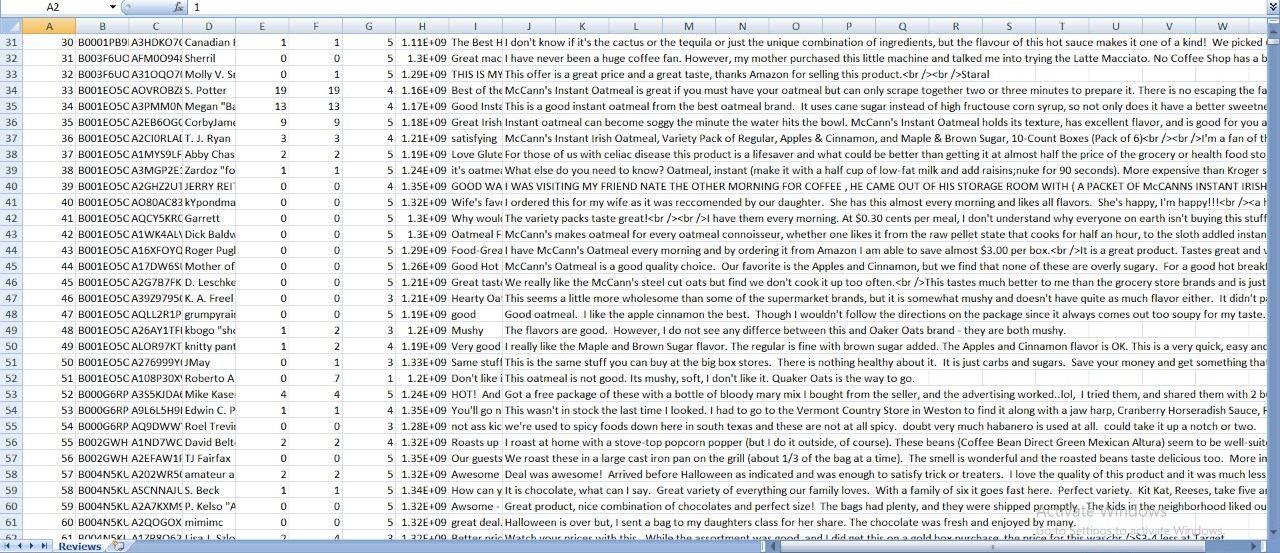
First, we got several datasets from the UCI depositories, Data world and Jmcauley Websites. The dataset contains reviews about the Cell phone and Musical Products of the Websites. The dataset contains about 50,000 to 90,000 of reviews but in our project we have taken only 2000 of reviews and made analysis on it.

We can take more number of reviews for analysis but if we do that then while finding the accuracy since we use machine learning next, it takes much time to analyze more reviews. We got Dataset in variety of extension like JSON, XML, CSV, EXCEL Later we converted the files into one extension i.e.. CSV file. Csv file is Comma Separated File where data is separated by comma and saved in dot csv .The data we got is presented in irregular way so to convert the date in the structured manner we need Preprocessing technique.

In the real world applications we work on thousands and thousands of line of data so it may take even days to analyze.

This is the Dataset that we got:





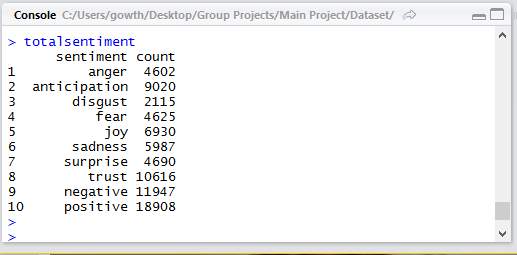
**Filtering the reviews:**

There are many numbers of reviews given for a particular product so in that reviews we need only the adjective words and the remaining nouns and pronouns are to be ignored so to ignore those words we use Stop words. Stop words is a variable consists of all the noun and pronoun words. By removing stop words from the reviews, we get only the adjective words from filtering. The remaining words are taken by having spaces between them. Later we need only v1(Verb 1) form of adjective so to do that we need remove the suffix from the adjective word and change it into the v1 form.

**Finding the count of sentiment words:**

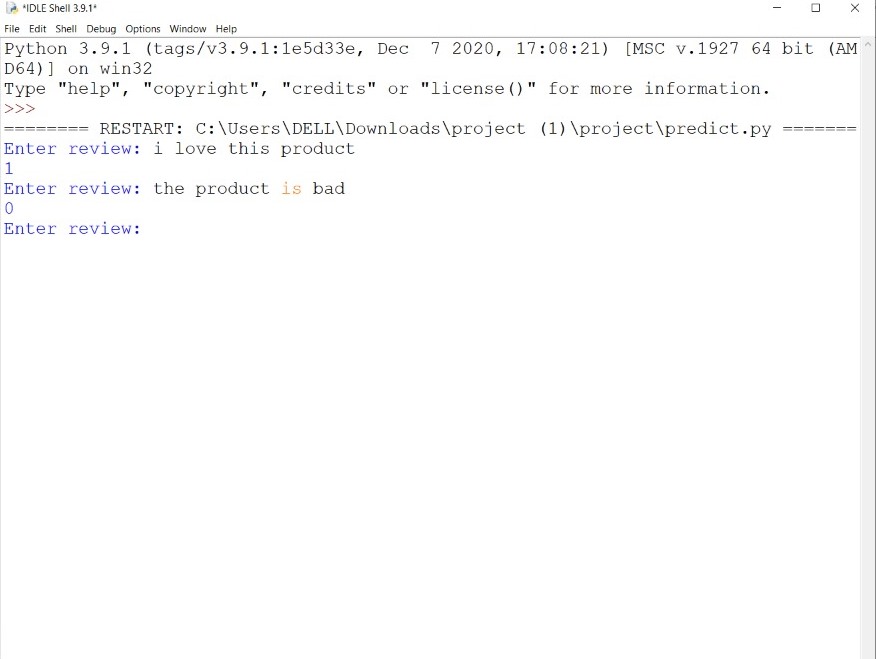
As our output is a bar-graph that outputs the emotions of customers based on the words used in their reviews, our first task is to find out count of those words. We used several packages to help with the count. In the previous step we removed all stop words and reviews are filtered so that can be used in this step.

Once we get all the necessary words that are adjectives from the previous step we classify them into positive and negative and then in those positive and negative classes we further divide them into good, best, anger, bad and other categories depending on the customer emotions that are specified by his words. We even got the number of sentiment words in each line.



**CHAPTER-7 OUTPUT**

Output:



* The output ‘1’ indicates it’s a positive review and whereas ‘0’ indicates it’s a negative review.

**EXPLANATION:**

* In our project we took different classes of sentiment on X-axis and the number of words that specify those emotions on the Y-axis.
* As we took bar graph, pie char and box plot for data visualization our output can be easily understood by the user.
* The bar corresponding to each class shows how much people are satisfied regarding the product.
* The pie chart describes how much percentage of the sentiment words is highly used.
* The box plot
* Total count on the Y-axis species the total count of such emotional words used in the reviews of our data set.
* This Is the Final Result of the Sentimental Analysis on Product Reviews

# CHAPTER-8 CONCLUSION

1. The ML approaches proffered the good outcomes to categorize product reviews. SVN got 98.170% accuracy and NB got 93.54% accuracy for Camera—related Reviews.
2. The approach utilizes the SVM, which encompasses several key parameters that are required to be set properly for attaining the best classification outcomes.
3. Thus, the SVM renders BEST accuracy in classification.
4. The key aim is to analyze a large amount of reviews by using amazon dataset which are already labeled.
5. With the variety of products increasing day by day the decision to opt for a particular product is becoming difficult. So, the need for sentimental analysis is increasing gradually. Although sentimental analysis tasks are challenging due to their natural language processing origins, much progress has been made over the last few years due to the high demand for it. Not only the consumer wants to know about the product but also the companies want to know about the condition of their product in the market.
6. The growing need for product insights – and the technical challenges currently facing the field –will keep sentiment analysis and opinion mining relevant for the foreseeable future. Next-generation opinion mining systems need a deeper bind between complete knowledge bases with reasoning methods inspired by human thought and psychology. This will lead to a better understanding of natural language opinions and will more efficiently bridge the gap between unstructured information in the form of human thoughts and structured data that can be analyzed and processed by a machine.
7. The intelligent opinion mining systems are capable of handling semantic knowledge, making analogies, continuous learning and detecting emotions that are leading to the highly efficient sentiment analysis.

.

# CHAPTER-9 REFERENCES

* + - 1. Tang Feilong, Luoyi Fu, Yao Bin, Wenchao Xu. Aspect based fine-grained sentiment analysis for online reviews. Inf Sci. 2019;488:190–204.
      2. Jagdale RS, Shirsat VS, Deshmukh SN. Sentiment analysis on product reviews using machine learning techniques. In Cognitive Informatics and Soft Computing, Springer, Singapore, pp. 639–647, 2019.
      3. Vanaja S, Belwal M. Aspect-level sentiment analysis on e-commerce data. In: international conference on inventive research in computing applications (ICIRCA), IEEE, pp. 1275–1279, 2018.
      4. Zhang W, Kong SX, Zhu YC. Sentiment classification and computing for online reviews by a hybrid SVM and LSA based approach. Cluster Computing. 2018

5. Sentiment Analysis and Opinion Mining by Bing Liu

(http://www.cs.uic.edu/~liub/FBS/SentimentAnalysisand-OpinionMining.html)

* 1. Sentiment Analysis by Professor Dan Jurafsky.

(https://web.standford.edu/class/cs124/lec/ sentiment.pdf)

* 1. S. Blair-Goldensohn, Hannan, McDonald, Neylon, Reis and Reynar 2008 – Building a Sentiment Summarizer for Local Service Review.

(http://www. ryanmcd.com/papers/local\_service\_summ.pdf)