A Project Report

on

Sentiment Analysis of Flipkart Product Review

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Engineering

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Swati Sharma
Assistant Professor

Submitted By

Parteek Asiwal 18SCSE1010648
Manav Raj 18SCSE1010425

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

December,2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **Sentiment Analysis of Flipkart Product Review** in partial fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of August, 2021 to December and 2021, under the supervision of Swati Sharma Assistant Professor Department of Computer Science and Engineering of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Parteek Asiwal 18SCSE1010648
Manav Raj 18SCSE1010425

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Ms. Swati Sharma
Assistant Professor

CERTIFICATE

The Final Project Viva-Voce examination of Parteek Asiwal, Manav Raj has been held on DATE and his/her work is recommended for the award of B.Tech

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021
Place: Greater Noida

Acknowledgement

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide **Ms. Swati Sharma** for the guidance, inspiration and constructive suggestions that helpful us in the preparation of this project. We also thank our colleagues who have helped in successful completion of the project.

Abstract

With the rise of online shopping the requirement for product review categorization is very important and it can be done using sentiment analysis and with the help of NLP. Sentiment Analysis analyses an incoming message and determines if the underlying sentiment is positive, negative, or neutral product . User’s comments are useful information to estimate product quality. Customers may communicate their opinions and feelings more openly than ever before, understanding people's emotions is critical for businesses. It is really difficult for a human to look through each line and find the emotion that represents the user experience. With the advancement of technology, we can now analyse consumer feedback automatically, from survey replies to social media chats, allowing firms to listen to their customers and adjust products and services to match their demands. Natural Language Processing or NLP is a field of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages. Through this analysis the company can get a full idea of customer feedback and can look after these particular areas. Increased business, fame and brand value profits make the company's customers more loyal. As Flipkart is one of the biggest ecommerce website of India so we have used it for the sentiment analysis. In this work , we propose a sentiment based rating expectation technique to take care of this issue

Contents

List of Table

List of Figures

Chapter 1
Introduction

As online marketplaces have been popular during the past decades, the online sellers and merchants ask their purchasers to share their opinions about the products they have bought. Everyday millions of reviews are generated all over the Internet about different products, services and places. This has made the Internet the most important source of getting ideas and opinions about a product or a service. However, as the number of reviews available for a product grows, it is becoming more diffcult for a potential consumer to make a good decision on whether to buy the product. Different opinions about the same product on one hand and ambiguous reviews on the other hand makes customers more confused to get the right decision. Here the need for analyzing this contents seems crucial for all e-commerce businesses. Sentiment analysis and classification is a computational study which attempts to address this problem by extracting subjective information from the given texts in natural language, such as opinions and sentiments. Different approaches have used to tackle this problem from natural language processing, text analysis, computational linguistics, and biometrics. In recent years, Machine learning methods have got popular in the semantic and review analysis for their simplicity and accuracy. Flipkart is one of the e-commerce giants that people are using every day for online purchases where they can read thousands of reviews dropped by other customers about their desired products. These reviews provide valuable opinions about a product such as its property, quality and recommendations which helps the purchasers to understand almost every detail of a product. This is not only beneficial for consumers but also helps sellers who are manufacturing their own products to understand the consumers and their needs better. This project is considering the sentiment classification problem for online reviews using supervised approaches to determine the overall semantic of customer reviews by classifying them into positive and negative sentiment. The data used in this study is a set of beauty product reviews from Flipkart that is collected from Snap dataset.

Machine language

The machine language is sometimes referred to as machine code or object code which is set of binary digits 0 and 1. These binary digits are understood and read by a computer system and interpret it easily. It is considered a native language as it can be directly understood by a central processing unit (CPU). The machine language is not so easy to understand, as the language uses the binary system in which the commands are written in

1 and 0 form which is not easy to interpret. There is only one language which is understood by computer language which is machine language. The operating system of the computer system is used to identify the exact machine language used for that particular system.

The operating system defines how the program should write so that it can be converted to machine language and the system takes appropriate action. The computer programs and scripts can also be written in other programming languages like C, C++, and JAVA. However, these languages cannot be directly understood by a computer system so there is a need for a program that can convert these computer programs to machine language. The compiler is used to convert the programs to machine language which can be easily understood by computer systems. The compiler generates the binary file and executable file.

Example of machine language for the text "Hello World".

01001000 0110101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 011001

How machine learning works

UC Berkeley (link resides outside IBM) breaks out the learning system of a machine learning algorithm into three main parts.

1. **A Decision Process**: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabeled, your algorithm will produce an estimate about a pattern in the data.

2. **An Error Function**: An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.

3. **An Model Optimization Process**: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this evaluate and optimize process, updating weights autonomously until a threshold of accuracy has been met.

Machine learning methods

Machine learning classifiers fall into three primary categories.

Supervised machine learning

Supervised learning, also known as supervised machine learning, is defined by its use of labelled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately. This occurs as part of the cross validation process to ensure that the model avoids overfitting or underfitting. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning include neural networks, naïve bayes, linear regression, logistic regression, random forest, support vector machine (SVM), and more.

Unsupervised machine learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, image and pattern recognition. It’s also used to reduce the number of features in a model through the process of dimensionality reduction; principal component analysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include neural networks, k-means clustering, probabilistic clustering methods, and more.

Semi-supervised learning

Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labeled data set to guide classification and feature extraction from a larger, unlabeled data set. Semi-supervised learning can solve the problem of having not enough labeled data (or not being able to afford to label enough data) to train a supervised learning algorithm.

Reinforcement machine learning

Reinforcement machine learning is a behavioral machine learning model that is similar to supervised learning, but the algorithm isn't trained using sample data. This model learns as it goes by using trial and error. A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem.

Models

Performing machine learning involves creating a model, which is trained on some training data and then can process additional data to make predictions. Various types of models have been used and researched for machine learning systems.

Artificial neural networks

An artificial neural network is an interconnected group of nodes, akin to the vast network of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

Artificial neural networks (ANNs), or connectionist systems, are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

An ANN is a model based on a collection of connected units or nodes called "artificial neurons", which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit information, a "signal", from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called "edges". Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

Deep learning consists of multiple hidden layers in an artificial neural network. This approach tries to model the way the human brain processes light and sound into vision and hearing. Some successful applications of deep learning are computer vision and speech recognition.

Decision trees

Decision tree learning uses a decision tree as a predictive model to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining, and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data, but the resulting classification tree can be an input for decision making.

Support-vector machine

Support-vector machines (SVMs), also known as support-vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.An SVM training algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use SVM in a probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Regression analysis

Regression analysis encompasses a large variety of statistical methods to estimate the relationship between input variables and their associated features. Its most common form is linear regression, where a single line is drawn to best fit the given data according to a mathematical criterion such as ordinary least squares. The latter is often extended by regularization (mathematics) methods to mitigate overfitting and bias, as in ridge regression. When dealing with non-linear problems, go-to models include polynomial regression (for example, used for trendline fitting in Microsoft Excel), logistic regression (often used in statistical classification) or even kernel regression, which introduces non-linearity by taking advantage of the kernel trick to implicitly map input variables to higher-dimensional space.

Bayesian networks

A simple Bayesian network. Rain influences whether the sprinkler is activated, and both rain and the sprinkler influence whether the grass is wet.

A Bayesian network, belief network, or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independence with a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning. Bayesian networks that model sequences of variables, like speech signals or protein sequences, are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

Genetic algorithms

A genetic algorithm (GA) is a search algorithm and heuristic technique that mimics the process of natural selection, using methods such as mutation and crossover to generate new genotypes in the hope of finding good solutions to a given problem. In machine learning, genetic algorithms were used in the 1980s and 1990s. Conversely, machine learning techniques have been used to improve the performance of genetic and evolutionary algorithms.

Natural language processing (NLP)

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural language generation.

The following is a list of some of the most commonly researched tasks in natural language processing. Some of these tasks have direct real-world applications, while others more commonly serve as subtasks that are used to aid in solving larger tasks.

Though natural language processing tasks are closely intertwined, they can be subdivided into categories for convenience. A coarse division is given below.

Text and speech processing
Optical character recognition (OCR)
Given an image representing printed text, determine the corresponding text.

Speech recognition
Given a sound clip of a person or people speaking, determine the textual representation of the speech. This is the opposite of text to speech and is one of the extremely difficult problems colloquially termed "AI-complete" (see above). In natural speech there are hardly any pauses between successive words, and thus speech segmentation is a necessary subtask of speech recognition (see below). In most spoken languages, the sounds representing

successive letters blend into each other in a process termed coarticulation, so the conversion of the analog signal to discrete characters can be a very difficult process. Also, given that words in the same language are spoken by people with different accents, the speech recognition software must be able to recognize the wide variety of input as being identical to each other in terms of its textual equivalent.

Speech segmentation

Given a sound clip of a person or people speaking, separate it into words. A subtask of speech recognition and typically grouped with it.

Text-to-speech

Given a text, transform those units and produce a spoken representation. Text-to-speech can be used to aid the visually impaired.

Word segmentation (Tokenization)

Separate a chunk of continuous text into separate words. For a language like English, this is fairly trivial, since words are usually separated by spaces. However, some written languages like Chinese, Japanese and Thai do not mark word boundaries in such a fashion, and in those languages text segmentation is a significant task requiring knowledge of the vocabulary and morphology of words in the language. Sometimes this process is also used in cases like bag of words (BOW) creation in data mining.

Morphological analysis

Lemmatization

The task of removing inflectional endings only and to return the base dictionary form of a word which is also known as a lemma. Lemmatization is another technique for reducing words to their normalized form. But in this case, the transformation actually uses a dictionary to map words to their actual form.

Morphological segmentation

Separate words into individual morphemes and identify the class of the morphemes. The difficulty of this task depends greatly on the complexity of the morphology (i.e., the structure of words) of the language being considered. English has fairly simple morphology, especially inflectional morphology, and thus it is often possible to ignore this task entirely and simply model all possible forms of a word (e.g., "open, opens, opened, opening") as separate words. In languages such as Turkish or Meitei, a highly agglutinated Indian language, however, such an approach is not possible, as each dictionary entry has thousands of possible word forms.

Part-of-speech tagging

Given a sentence, determine the part of speech (POS) for each word. Many words, especially common ones, can serve as multiple parts of speech. For example, "book" can be a noun ("the book on the table") or verb ("to book a flight"); "set" can be a noun, verb or adjective; and "out" can be any of at least five different parts of speech.

Stemming

The process of reducing inflected (or sometimes derived) words to a base form (e.g., "close" will be the root for "closed", "closing", "close", "closer" etc.). Stemming yields similar results as lemmatization, but does so on grounds of rules, not a dictionary.

Syntactic analysis

Grammar induction

Generate a formal grammar that describes a language's syntax.

Sentence breaking (also known as "sentence boundary disambiguation")

Natural language processing strives to build machines that understand and respond to text or voice data—and respond with text or speech of their own—in much the same way humans do.

What is natural language processing?

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There’s a good chance you’ve interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

NLP tasks

Human language is filled with ambiguities that make it incredibly difficult to write software that accurately determines the intended meaning of text or voice data. Homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, variations in sentence structure—these just a few of the irregularities of human language that take humans years to learn, but that programmers must teach natural language-driven applications to recognize and understand accurately from the start, if those applications are going to be useful.

Several NLP tasks break down human text and voice data in ways that help the computer make sense of what it's ingesting. Some of these tasks include the following:

Speech recognition, also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions. What makes speech recognition especially challenging is the way people talk—quickly, slurring words together, with varying emphasis and intonation, in different accents, and often using incorrect grammar.

Part of speech tagging, also called grammatical tagging, is the process of determining the part of speech of a particular word or piece of text based on its use and context. Part of speech identifies ‘make’ as a verb in ‘I can make a paper plane,’ and as a noun in ‘What make of car do you own?’

Word sense disambiguation is the selection of the meaning of a word with multiple meanings through a process of semantic analysis that determine the word that makes the most sense in the given context. For example, word sense disambiguation helps distinguish the meaning of the verb 'make' in ‘make the grade’ (achieve) vs. ‘make a bet’ (place).

Named entity recognition, or NEM, identifies words or phrases as useful entities. NEM identifies ‘Kentucky’ as a location or ‘Fred’ as a man's name.

Co-reference resolution is the task of identifying if and when two words refer to the same entity. The most common example is determining the person or object to which a certain pronoun refers (e.g., ‘she’ = ‘Mary’), but it can also involve identifying a metaphor or an idiom in the text (e.g., an instance in which 'bear' isn't an animal but a large hairy person).

Sentiment analysis attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text.

Natural language generation is sometimes described as the opposite of speech recognition or speech-to-text; it's the task of putting structured information into human language.

See the blog post “NLP vs. NLU vs. NLG: the differences between three natural language processing concepts” for a deeper look into how these concepts relate.

NLP tools and approaches

Python and the Natural Language Toolkit (NLTK)

The Python programing language provides a wide range of tools and libraries for attacking specific NLP tasks. Many of these are found in the Natural Language Toolkit, or NLTK, an open source collection of libraries, programs, and education resources for building NLP programs.

The NLTK includes libraries for many of the NLP tasks listed above, plus libraries for subtasks, such as sentence parsing, word segmentation, stemming and lemmatization (methods of trimming words down to their roots), and tokenization (for breaking phrases, sentences, paragraphs and passages into tokens that help the computer better understand the text). It also includes libraries for implementing capabilities such as semantic reasoning, the ability to reach logical conclusions based on facts extracted from text.

Sentiment analysis

Sentiment analysis is a machine learning tool that analyzes texts for polarity, from positive to negative. By training machine learning tools with examples of emotions in text, machines automatically learn how to detect sentiment without human input.

To put it simply, machine learning allows computers to learn new tasks without being expressly programmed to perform them. Sentiment analysis models can be trained to read beyond mere definitions, to understand things like, context, sarcasm, and misapplied words. For example:

“Super user-friendly interface. Yeah right. An engineering degree would be helpful.”

Out of context, the words ‘super user-friendly’ and ‘helpful’ could be read as positive, but this is clearly a negative comment. Using sentiment analysis, computers can automatically process text data and understand it just as a human would, saving hundreds of employee hours.

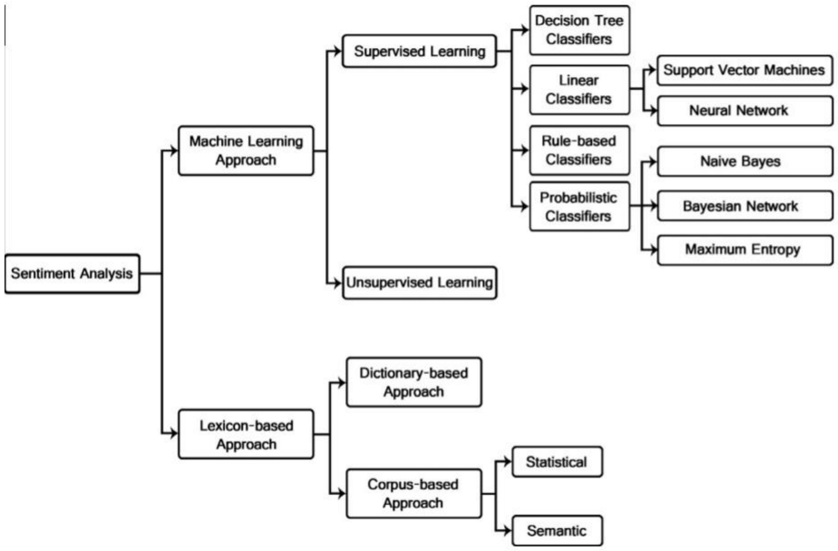
Imagine using machine learning to process customer service tickets, categorize them in order of urgency, and automatically route them to the correct department or employee. Or, to analyze thousands of product reviews and social media posts to gauge brand sentiment.

Sentiment Analysis, or Opinion Mining, is a subfield of NLP (Natural Language Processing) that aims to extract attitudes, appraisals, opinions, and emotions from text. Inspired by the rapid migration of customer interactions to digital formats e.g. emails, chat rooms, social media posts, comments, reviews, and surveys, Sentiment Analysis has become an integral part of analytics organizations must perform to understand how they are positioned in the market. To be clear, Sentiment Analysis isn’t a novel concept. In fact, it has always been an important part of CRM (Customer Relationship Management) and Market Research — companies rely on knowing their customers better to evolve and innovate. The more recent rise is driven largely by the availability/accessibility of customer interaction records and well as improved computing capabilities to process these data. This advancement has really benefited consumers in meaningful ways. More than ever, organizations are listening to their constituents to improve. There are numerous approaches for Sentiment Analysis. In this article, we'll explore three such approaches: 1) Naive Bayes, 2) Deep Learning LSTM, and 3) Pre-Trained Rule-Based VADER Models. We will focus on comparing simple out-of-the-box version of the models with the recognition that each approach can be tuned to improve performance. The intention is not to go into great details about how each methodology works but rather a conceptual study on how they compare to help determine when one should be preferred over another.

Background on Sentiment Analysis

The objective of Sentiment Analysis ranges on the positive to negative spectrum. As with other NLP efforts, it is generally considered a classification problem, though it can be viewed as a regression problem when precision is important. Sentiment Analysis used to be accomplished by having a large labor force to read through and manually assess texts. This approach is costly and prone to human error. In an effort to automate this process, companies look to advanced analytical methods for solving this problem. The challenge with Sentiment Analysis is that people express and interpret sentiment polarity and intensity differently. Furthermore, words and sentences can have multiple meanings based on the context (known as polysemy). While some of these issues can be mitigated, there is almost always a trade-off between speed and performance like any analytical tasks. We review three general methodologies, each with its own strengths and drawbacks:

There are two approaches to achieve Sentimental Analysis



How Does Sentiment Analysis with Machine Learning Work?
There are a number of techniques and complex algorithms used to command and train machines to perform sentiment analysis. There are pros and cons to each. But, used together, they can provide exceptional results. Below are some of the most used algorithms.

How machine learning in sentiment analysis works in both the training and prediction phases

Naive Bayes
Naive Bayes is a fairly simple group of probabilistic algorithms that, for sentiment analysis classification, assigns a probability that a given word or phrase should be considered positive or negative.

Essentially, this is how Bayes' theorem works. The probability of A, if B is true, is equal to the probability of B, if A is true, times the probability of A being true, divided by the probability of B being true:

Formula for Bayes' theorem: The probability of A, if B is true, is equal to the probability of B, if A is true, times the probability of A being true, divided by the probability of B being true

But that's a lot of math! Basically, Naive Bayes calculates words against each other. So, with machine learning models trained for word polarity, we can calculate the likelihood that a word, phrase, or text is positive or negative.

When techniques like lemmatization, stopword removal, and TF-IDF are implemented, Naive Bayes becomes more and more predictively accurate.

Linear Regression
Linear regression is a statistical algorithm used to predict a Y value, given X features. Using machine learning, the data sets are examined to show a relationship. The relationships are then placed along the X/Y axis, with a straight line running through them to predict further relationships.

Linear regression calculates how the X input (words and phrases) relates to the Y output (polarity). This will determine where words and phrases fall on a scale of polarity from "really positive" to "really negative" and everywhere in between.

Support Vector Machines (SVM)
A support vector machine is another supervised machine learning model, similar to linear regression but more advanced. SVM uses algorithms to train and classify text within our sentiment polarity model, taking it a step beyond X/Y prediction.

For a simple visual explanation, we'll use two tags: red and blue, with two data features: X and Y. We'll train our classifier to output an X/Y coordinate as either red or blue.

How SVM works: red and blue shapes represent two data features: X and Y
The SVM then assigns a hyperplane that best separates the tags. In two dimensions this is simply a line (like in linear regression). Anything on one side of the line is red and anything on the other side is blue. For sentiment analysis this would be positive and negative.

In order to maximize machine learning, the best hyperplane is the one with the largest distance between each tag:

SVM assigns a hyperplane that best separates the tags or red and blue shapes
However, as data sets become more complex, it may not be possible to draw a single line to classify the data into two camps:

two-dimensional hyperplane explaining how SVM works
Using SVM, the more complex the data, the more accurate the predictor will become. Imagine the above in three dimensions, with a Z axis added, so it becomes a circle.

Mapped back to two dimensions with the best hyperplane, it looks like this:

Showing the best hyperplane for SVM

Chapter 2
Literature Survey

In this 21st century, people are more social in social media, internet, online shopping etc. Thus directly or indirectly online judgments, opinions are eventually gaining great attention. But the real deal is analysis or mining of opinions. Below is the review of some existing solutions available for SA. These methods are also briefly tabulated in Table.1.

OPINE, an unsupervised, web-based information extraction system proposed by Propescu et al. [5] extracted product feature and opinions from reviews. It identifies product feature, opinion regarding product feature, determines polarity of opinions and then ranks product accordingly [9]. In feature identification, nouns from dataset or reviews are extracted. Frequencies higher than the threshold frequency are kept else discarded. OPINE's feature assessor is used to extract explicit features (occurrence offrequent features) [4]. Researchers have used manual extraction rule to extract data [4]. Advancement of OPINE is its domain independency. But fails to find its real life uses as OPINE system is not easily available.

Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone, is a linguistic approach of sentiment analysis atdocument level, proposed by Benamara et al. [8] in the year 2006.This research work began with measuring the intensity of degreeof adverbs (using Linguistic Classifiers) and adverb-adjective combinations (using Scoring Methods). Variable Priority Scoring,Adjective Priority Scoring and Adverb First Scoring are the said Scoring methods used herein [8]. The goal of all these methods are nothing but to add a relative weight (in a variable, on a scale of 0 to 1) of score of adverb relative to the score of adjective. This paper aim to determine which weight most closely matches human assignments of opinions. Experimenting on about 200 documents of news resources it shows that analysis that best matches the human sentiments must comprise of 35% of adverbsalong with adjectives. Produces Pearson correlation (correlation between human sentiment and Sentiment Analysis Algorithms) and of about 0.47 (ranging in between -1 and 1) [8]. Though this approach shows higher Pearson correlation but considered very few dataset.

One of the solutions to Sentiment Analysis namely Opinion Digger was introduced by Moghaddam and Ester [1]. This unsupervised Machine Learning methodology works at Sentence level. Correlates and compares product aspect and standard rating guidelines (used in Amazon, Snapdeal, flipkart1 etc). This proposed work is divided into two sub methods. At first, input information is fragmented into sentences. Repeated nouns in the sentences are coined as aspects. Aspect (repeated nouns) if forms any pattern, are stored. Secondly, aspects are compared to the rating guideline (like 4 means "Good", 3 means "Average", etc) and accordingly labeled as "Good" ,"Average" and "Bad" [1]. Major advantage is its high performance in product rating at aspect level with a loss of 0.49 only. Demanding guidelines and known data to rate are its major drawbacks and it was compared with very few methodologies. Therefore lacks more number of performance comparisons.

Sentiment Classification from Online Customer reviews Using Lexical Contextual Sentence Structure was proposed by Khan et al. [2] is a semantic or Rule-based (Dictionary Polarity) approach of analyzing customer reviews [2]. Firstly, input is fragmented into sentences and using method “POS” each word is stored. Secondly, based on the context and structure of the sentence polarity of the given sentence is calculated. Nouns are coined as “aspects”. Concept of semantic score of words available in SentiWordNet are used to label the sentence as either positive or negative [6]. Accuracy of 86% is produced. Said to be domain independent (subject of review), advantage but the author collected few data (about 3600). Major drawback is it full dependency on WordNet [2].

Interdependent Latent Dirichlet Allocation presented by Moghaddam and Ester [1] is a probabilistic graphical model of rating product at aspect level [6]. Majority of the review sites considers number of stars as the tool to rate a product. This proposed work also does the same assuming interdependency between aspect (feature) and its matching rating. This model tries to generate and showcase cluster head terms into aspects and reviews into ratings in the form of multinomial distributions [10]. Each item in the pool of discrete data is represented as a finite mixture over some latent variables. Found to gain a rating accuracy of about 73%. Since graphical representation suffers from chances of having errors and mistakes in representation of data, this technique might not produce expected output always.

A Joint Model of Feature Mining and Sentiment Analysis for Product Review Rating was presented by de Albornoz et al. [11]. This machine learning method rates product at global level considering whole opinion at once. This approach is basically carried out in four steps. At first important features in the document or review are marked. Secondly, sentences containing features (aspects) are identified. Very next, polarity and strength of those sentences are calculated. At last, products are rated globally at aspect level. Feature weights are calculated automatically. Researchers have used the concept of Vector Feature Intensity Graph (VFIG) to represent the reviews [6]. Though use of WordNet is the major disadvantage of this work, it produces an average prediction accuracy of 71% (3 categories) and 46.9% (5 categories) [11].

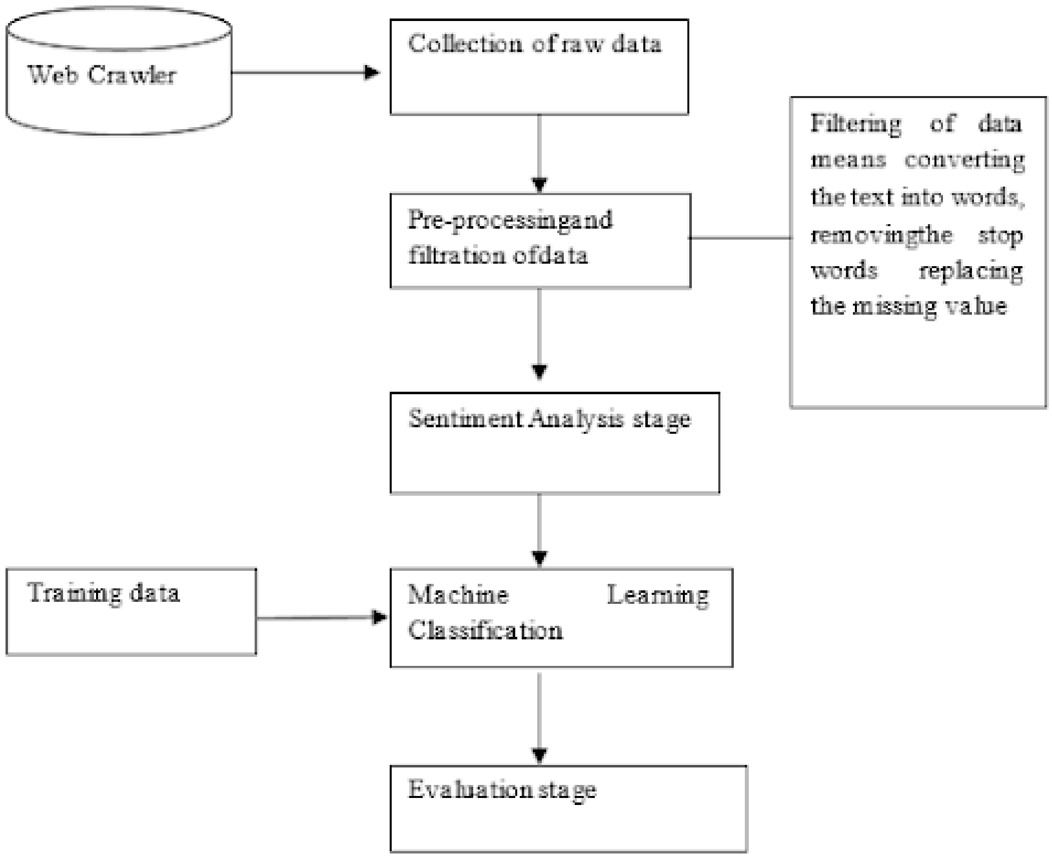
Method	**Year of proposal**	**Classification**	**Text Level**	**Prediction Accuracy**	**Pros**	**Cons**
OPINE	2005	Unsupervise drule-based approach	Word	87%	Domain independent	Difficulty in availing OPINE system, thus rare to get applied in real life.
Sentiment Analysis: Adjectives and Adverbs are better than AdjectivesAlone	2006	Linguistic approach	Document	Pearson correlation of0.47	Adjectives are given more priority(adjectives expresses human sentiments better than adverbs alone)	None
Opinion Digger	2010	Unsupervised machine learning method	Sentence	51%	Rates product at aspect level	Requires rating guidelines to rate. Works only on known data.
Sentiment Classification Using Lexical Contextual Sentence Structure	2011	Rule based approach	Sentence	86%	Said to be domain independent [6]	Depends solely on wordNet
Interdependent Latent Dirichlet Allocation	2011	Probabilistic graphical model	Document	73%	Faster in comparing and correlating sentiment andrating	Correlation between identified clusters and feature or ratings are not explicit always[6]
A Joint Model of Feature Mining and Sentiment Analysis for Product Review Rating	2011	Machine Learning	Document	71% (in 3 categories) 46.9% (in 5 categories)	Automatic calculation offeature vector	Use of WordNet

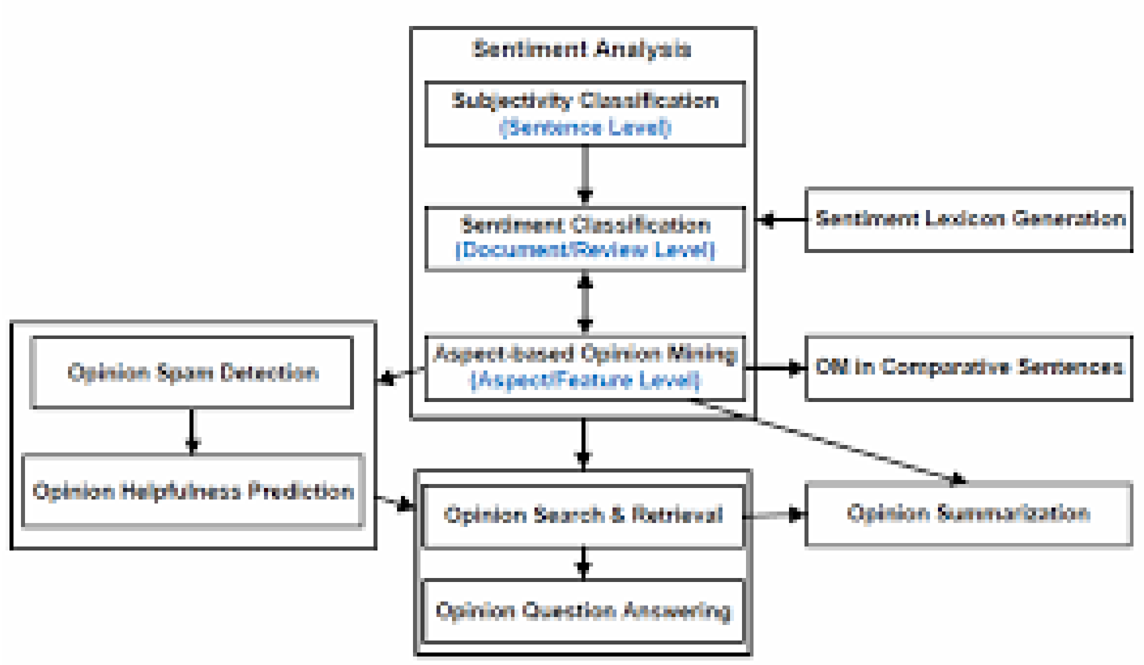
Comparison Table of Existing Techniques

2.1 Project Design

"AI" redirects here. For other uses, see AI (disambiguation) and Artificial intelligence (disambiguation).
Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to natural intelligence displayed by animals including humans. Leading AI textbooks define the field as the study of "intelligent agents": any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term "artificial intelligence" to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving", however, this definition is rejected by major AI researchers.

AI applications include advanced web search engines (e.g., Google), recommendation systems (used by YouTube, Flipkart and Netflix), understanding human speech (such as Siri and Alexa), self-driving cars (e.g., Tesla), automated decision-making and competing at the highest level in strategic game systems (such as chess and Go). As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the AI effect. For instance, optical character recognition is frequently excluded from things considered to be AI,having become a routine technology.


Web Crawler
Collection of raw data
Pre-processingand filtration ofdata
Filtering of data means converting the text into words, removingthe stop words replacing the missing value
Sentiment Analysis stage
Training data
Machine Learning Classification
Evaluation stage


Sentiment Analysis
Subjectivity Classification (Sentence Level)
Sentiment Classification (Document/Review Level)
Aspect-based Opinion Mining (Aspect/Feature Level)
Sentiment Lexicon Generation
OM in Comparative Sentences
Opinion Spam Detection
Opinion Helpfulness Prediction
Opinion Search & Retrieval
Opinion Question Answering
Opinion Summarization

Chapter 3

Working of Project

This section illustrates the proposed algorithm for sentiment analysis. This proposed algorithm is divided into three phases as shown in Fig.2.

- Data Filtration
- Training model
- Testing model

The detailed algorithms of all phases are discussed below. The data filtration flow diagram is given in Fig.3, for training model flow diagram is given in Fig.4 and for testing model flow diagramis given in Fig.5.

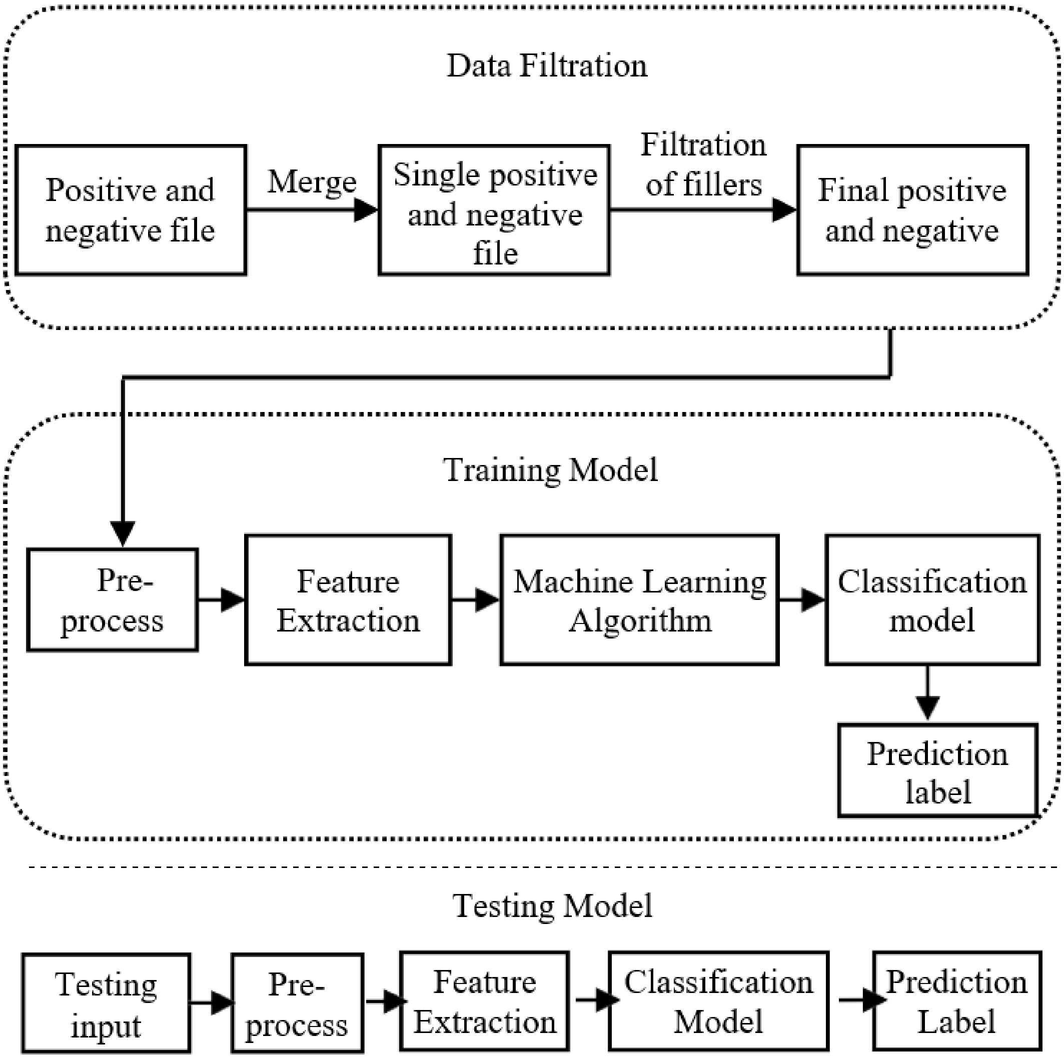
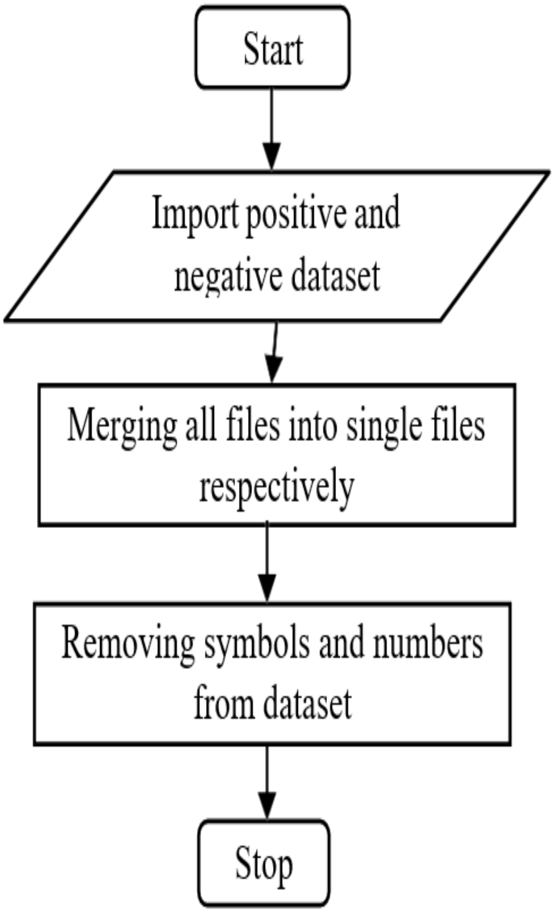


Fig.2. Illustration of the proposed model for sentiment analysis

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 andtesting, respectively. Training dataset to classification algorithms like Naïve Bayes classification algorithm


Start
Import positive and negative dataset
Merging all files into single files respectively
Removing symbols and numbers from dataset
Stop

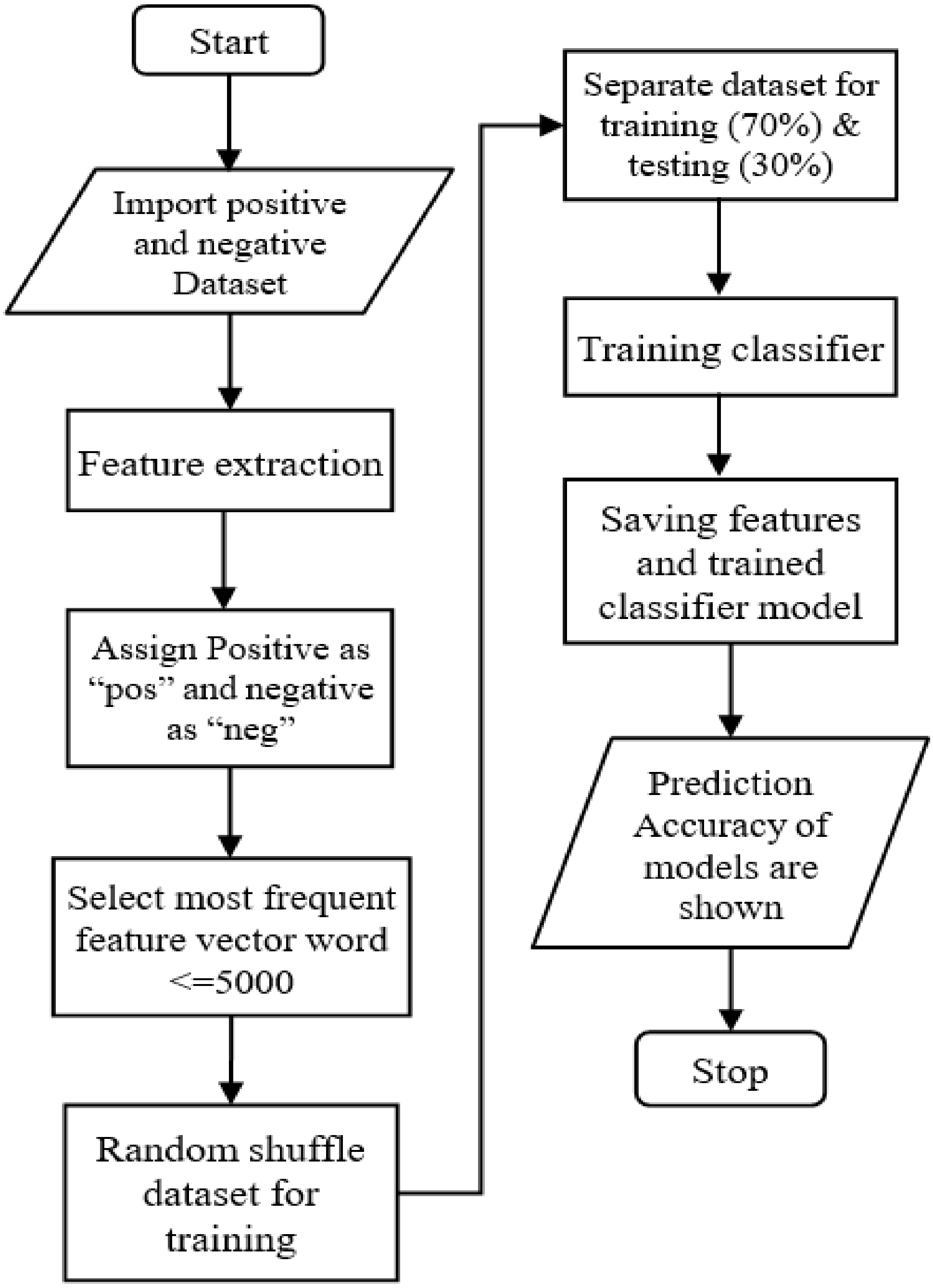

Start
Import positive and negative Dataset
Feature extraction
Assign Positive as "pos" and negative as "neg"
Select most frequent feature vector word <=5000
Random shuffle dataset for training
Separate dataset for training (70%) & testing (30%)
Training classifier
Saving features and trained classifier model
Prediction Accuracy of models are shown
Stop

Fig.4. Schematic diagram for implementing machine learning algorithms

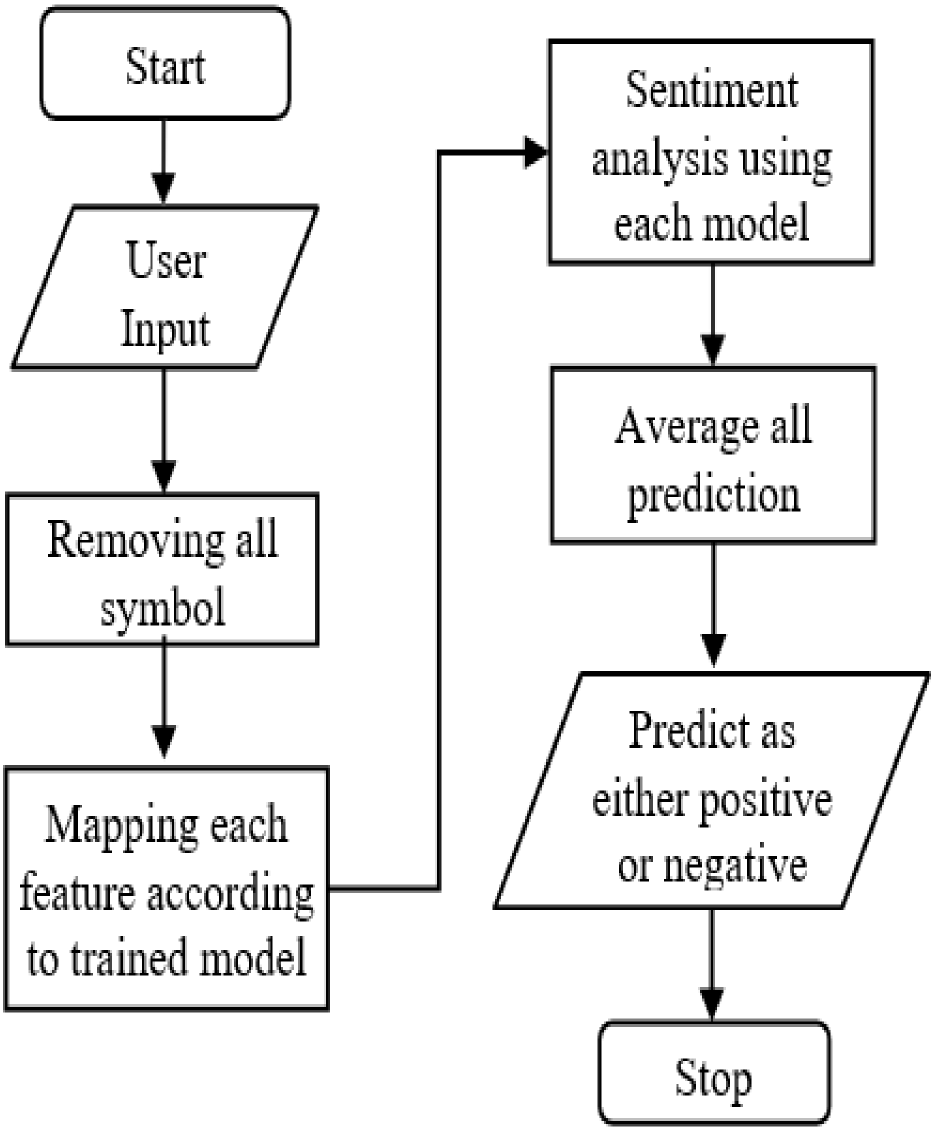

Start
User Input
Removing all symbol
Mapping each feature according to trained model
Sentiment analysis using each model
Average all prediction
Predict as either positive or negative
Stop

Fig.5. Diagram for testing the proposed model on datasets

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.

Algorithm 1: Data Filtration Algorithm

Step 1: Importing both positive and negative files and combining them into single file

Step 2: Removal of punctuations and numbers from the dataset

Step 3: Output (Filtered data)

Algorithm 2: Algorithm for Machine Learning Implementation

Step 1: Fetching text paragraph from dataset

Step 2: Feature Extraction phase: Extracting words corresponds to adjective, adverb and verb.

Step 3: All the positive sentences are labeled as "pos" and all the negative ones are labeled as "neg".

Step 4: Most frequent feature vector word is set to 5000 words.

Step 5: Random shuffling the dataset for training

Step 6: Dividing dataset into 70% training and 30% testing dataset

Step 7: Training dataset to classification algorithms like Naïve Bayes classification algorithm [5], Linear Model algorithm [16], SVM algorithm [17]

Step 8: Save the outputs of step 2, and step 7

Step 9: Output (Representation of Accuracy of each model)

Algorithm 3: Proposed algorithm to perform Sentiment Analysis

Step 1: User Input

Step 2: Preprocessing:

- a. Removal of “ ’ ” symbol from the text
- b. Mapping to user input using saved featured (based on training dataset)

Step 3: Feeding Mapped data to different model for sentiment analysis

Step 4: Output (Averaging all the models)

Implementation :

Phase 1: Importing the packages for data analysis

At first sight, we will use four main packages: pandas,numpy, matplotlib and seaborn. Let's import these packages using the keyword import. We will change the name
from pandas to pd,numpy to np,matplotlib to pltand seaborn to sns, using the keyword as.

We will be using the nltk, sklearn, collections and wordcloud packages for processing our text component. While we analyze the text, we will be using sklearn package again to model our text features.

Phase 2: Reading and performing basic analysis of the data

As usual the first step is to read the available data and perform some high-level analysis on it:

```
Flipkart_reviews = pd.read_csv('https://media.githubusercontent.com/media/juliandariomirandacalle/NLP_Notebooks/master/01-Introduction_NLP/Customer_Reviews.csv')
Flipkart_reviews.head(3)
```

We will be working with a .csv file that contains information about tens of thousands of customers writing reviews on Amazon products every day. Each review contains textual feedback along with a 1-to-5 star rating system (1 being least satisfied and 5 being most satisfied). In this way, the following attributes are available in the data:

1. **Id (numerical):** start and end date of the attack in *timestamp* format.
2. **ProductId (categorical):** ID of the referenced product by the customer.
3. **UserId (categorical):** registered user ID.
4. **ProfileName (text):** registered user profile name.
5. **HelpfulnessNumerator (numerical):** number of users who found the review helpful.
6. **HelpfulnessDenominator (numerical):** Number of users who voted whether the review was helpful or not.
7. **Score (ordinal):** rating between 1 and 5.
8. **Time (numerical):** timestamp of the review.
9. **Summary (text):** brief summary of the review.
10. **Text (text):** text of the review.

2.1 Standardizing the ratings for sentiment analysis (5 mts)

For the purposes of sentiment analysis, we will convert all of the ratings into binary values using the follow rule:

- Ratings of 4 or 5 will get mapped to 1 and will be reltead to positive reviews
- Ratings of 1 or 2 will get mapped to 0 and will be related to negative reviews
- Ratings of 3 will get removed since they will represent neutral reviews.

CodeText

2.2 Pre-processing

As discussed previously, text preprocessing and normalization is crucial before building a proper NLP model. Some of the important steps are:

1. Converting words to lower/upper case
2. Removing special characters
3. Removing stopwords and high/low-frequency words
4. Stemming/lemmatization

2.3 Stemming & lemmatization

Now we are ready for the last part of our pre-processing - **stemming & lemmatization**.

Different forms of a word often communicate essentially the same meaning. For example, there’s probably no difference in intent between a search for shoe and a search for shoes. The same word may also appear in different tenses; e.g. "run", "ran", and "running". These syntactic differences between word forms are called **inflections**. In general, we probably want to treat inflections identically when extracting features from the text.

Sometimes this process is nearly-reversible and quite safe (e.g. replacing verbs with their infinitive, so that "run", "runs", and "running" all become "run"). Other times it is a bit dangerous and context-dependant (e.g. replacing superlatives with their base form, so that "good", "better", and "best" all become "good"). The more aggressive you are, the greater the potential rewards and risks. For a very aggressive example, you might choose to replace "Zeus" and "Jupiter" with "Zeus" only; this might be OK if you are summarizing myths, confusing if you are working on astronomy, and disastrous if you are working on comparative mythology.

Phase 3 : Building a machine learning and model Applying machine leaning algorithm.

Now we have cleaned-up versions of two very important pieces of data – the actual review text and its corresponding sentiment rating:

CodeText

The independent variables or model features are derived from the review text. Previously, we discussed how we can use n-grams to create features, and specifically how bag-of-words is the simplest interpretation of these n-grams, disregarding order and context entirely and only focusing on frequency/count. Let's use that as a starting point.

Conversely, reading each of the reviews, it is clear that, for instance, "good" is mentioned in context like "not as good" or "sounds good". This indicates that in the world of text we cannot go by single words (also called **1-grams**) alone. The context of the sentence or the surrounding words at least are very much necessary to understand the sentiment of a sentence.

3.1 n-grams

Since 1-grams are sometimes insufficient to understand the significance of certain words in our text, it is natural to consider blocks of words, or **n-grams**.

The simplest version of the n-gram model, for $n>1$, is the **bigram** model, which looks at pairs of consecutive words. For example, the sentence "The quick brown fox jumps over the lazy dog" would have tokens "the quick", "quick brown",..., "lazy dog".

3.2 Bag-of-words

The bag-of-words procedure falls under a broader category of techniques known as **count-based representations**. These are techniques to analyze documents by indicating how frequently certain types of structures occur throughout.

Let's start with 1-grams (words). The simplest type of information would be whether a particular word occurs in particular documents. This leads to **word-document co-occurrence matrices**, where the (W,X) entry of the word-document matrix is set to 1 if word W occurs in document X, and 0 otherwise.

Chapter 4

CONCLUSION

In this paper, we offer a sentiment-based rating prediction and recommendation algorithm that may be used to predict product ratings based on user reviews. The purpose is to provide a feature-based impression of a large number of customer reviews of a product sold on the internet. To accomplish the rating prediction challenge, we combine sentiment similarity, interpersonal sentiment influence, and item reputation similarity into a single matrix factorization framework.

We will examine complicated strategies for extracting opinion and product features, as well as novel classification models that can handle the organised names property in rating prediction and sentiment lexicons to apply fine-grained sentiment analysis in the future.

Chapter 5
Reference

[1] Samaneh Moghaddam and Martin Ester, “Opinion Digger: An Unsupervised Opinion Miner from Unstructured Product Reviews”, Proceedings of 19th ACM International Conference on Information and Knowledge Management, pp. 1825-1828, 2010.

[2] Aurangzeb Khan, Baharum Baharudin and Khairullah Khan, “Sentiment Classification from Online Customer Reviews using Lexical Contextual Sentence Structure”, Proceedings of International Conference on Software Engineering and Computer Systems, pp. 317-331, 2011.

[3] M. Hu and B. Liu, “Mining and Summarizing Customer Reviews”, Proceedings of 10th ACM International Conference on Knowledge Discovery and Data Mining, pp. 166-177, 2005.

[4] A.M. Popescu and O. Etzioni, “Extracting Product Features and Opinions from Reviews”, Proceedings of International Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 339-346, 2005.

[5] G. Vinodhini and R.M. Chandrasekaram, “Sentiment Analysis and opinion Mining: A Survey”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, No. 6, pp. 28-35, 2012.

[6] A. Collomb, C. Costea, D. Joyeux, O. Hasan and L. Brunie, “A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation”, Available at: https://liris.cnrs.fr/Documents/Liris-6508.pdf.

[7] Mika V. Mantyla, Daniel Graziotin and Miikka Kuutila, “The Evolution of Sentiment Analysis-A Review of Research Topics”, Computer Science Review, Vol. 27, No. 1, pp. 16-32, 2018.

[8] F. Benamara, C. Cesarano and D. Reforgiato, “Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone”, Proceedings of International Conference on Weblogs and Social Media, pp. 1-7, 2006.

[9] R.A. Hummel and S.W. Zucker, “On the Foundations of Relaxation Labeling Processes”, Proceedings of International Conference on Computer Vision: Issues, Problems, Principles, and Paradigms, pp. 585-605, 1987.

[10] Samaneh Moghaddam and Martin Ester, “ILDA: Interdependent LDA Model for Learning Latent Aspects and their Ratings from Online Product Reviews”, Proceedings of 34th

International ACM Conference on Research and Development in Information Retrieval, pp. 665-674, 2011.

[11] Jorge Carrillo De Albornoz, Laura Plaza, Pablo Gervas and Alberto Diaz, “A Joint Model of Feature Mining and Sentiment Analysis for Product Review Rating”, Proceedings of International Conference on Advances in Information Retrieval, pp. 55-66, 2011.

[12] Sentiment Analysis, Available at: https://insightsatlas.com/sentiment-analysis/

[13] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng and Christopher Potts, “Learning Word Vectors for Sentiment Analysis”, Proceedings of 49th Annual Meeting of the Association for Computational Linguistics, pp. 1-7, 2011.

[14] Understanding Sentiment Analysis: What It Is and Why It’s Used Understanding Sentiment Analysis: What It Is and Why It’s Used, Available at: https://www.brandwatch.com/blog/understanding-sentiment-analysis/

[15] Sentiment Analysis Explained, Available at: https://www.lexalytics.com/technology/sentiment-analysis