SENTIMENT ANALYSIS

In partial fulfilment of the requirements for the Bachelor of Technology degree, a project report is submitted.

in

Computer Science and Engineering

By

Shriya Singh (181455)

Under the supervision of

Dr.Pradeep Kumar Gupta

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh

DECLARATION

I hereby declare that the work presented in this project entitled **"Sentiment Analysis"** under the supervision of **Dr.Pradeep Kumar Gupta** (Associate Professor), Jaypee University of Information Technology. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Dr.Pradeep Kumar Gupta

Associate Professor

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology

Submitted by:

Shriya Singh (181455)

Computer Science & Engineering Department

Jaypee University of Information Technology

CERTIFICATE

This is to confirm that the work given in the "**Sentiment Analysis**" project report is in partial fulfilment of the requirements for the degree award. of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Shriya Singh (181455) during the period from January 2022 to May 2022 under the supervision of **Dr.Pradeep Kumar Gupta**, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Shriya Singh

(181455)

To the best of my knowledge, the preceding statement is correct.

Dr.Pradeep Kumar Gupta

Associate Professor

Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat

AKCNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I really grateful and wish my profound my indebtedness to Supervisor **Dr.Pradeep Kumar Gupta**, Associate Professor Department of CSE Jaypee University of Information Technology, Waknaghat. Deep knowledge and keen interest of my supervisor in the field of "**Research Area**" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr.Pradeep Kumar Gupta Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I want to express my gratitude to my parents for their unwavering support and patience.

Shriya Singh

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Sentiment analysis is the computational study of the sentiments, opinions, attitudes, and emotions of people expressed in written language. It is one of the most active research areas in natural language processing and data mining in recent years. Its popularity is mainly due to two reasons. First, it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behavior. Whenever we need to decide, we tend to hear other's opinion. Second, it presents many challenging research problems, which never had been attempted before the year 2000. One of the major reasons for the lack of study earlier is that there was way too little opinionated text in digital form. Hence, it is not really a surprise that the inception and the rapid growth of the field coincide with those of the social media on the web. In fact, the research has also spread out of computer science to management sciences due to its importance to business and a society. In this talk, we will start with the discussion of the mainstream sentiment analysis research and then move on to describe some recent work on modelling comments, discussions and debates, which give an overview of a whole different kind of analysis of sentiments and opinions.

Sentiment classification is a way to analyze and process the subjective information in the text and then label those opinions as positive, negative, or neutral. It is the procedure by which information is extracted from the opinion's appraisals and emotions of people regarding entities, events and their attributes. In decision making, the opinions of others have a significant effect on customer ease, making choices with regards to online shopping, choosing events, products, entities. The approaches of sentiment analysis of a text typically work only at a particular level, for example - phrase, sentence, or document level. This report basically aims at analyzing a solution for the sentiment classification at a fine-grained level, namely the sentence level in which clarity of the sentence can be given by three categories as positive negative and neutral.

CHAPTER 1 : INTRODUCTION

1. Introduction

Sentiment analysis is a kind of data mining where you measure the inclination of people's opinions by using NLP (natural language processing) and text analysis. In this project we'll be working on Fine-Grained system that analyses the sentiment and gives precise results to what the public opinion is about the subject i.e., positive, negative or neutral.

By the end of the project, we will train our data to analyze the sentiments of a customer buying products on E-Commerce website by modelling and then deploying the model using Heroku and GitHub.

Sentiment analysis refers to the utilization of natural language processing, text analysis and linguistics to spot and extract subjective information from the source materials. Sentiment analysis aims to analyze the attitude of a speaker or a writer with respect to some topic or the general contextual polarity of a document. The attitude could also be his or her judgment or evaluation affective state, or the intended emotional communication. It is that process of detecting a piece of writing for positive, negative, or neutral feelings constrained to it. Humans have the innate ability to figure out the sentiment; however, this process is time consuming, inconsistent, and expensive when it comes to the context of business. It's just not realistic to make people read tens of thousands of user customer reviews individually and then score them for sentiment.
For example, if we consider the cloud-based sentiment analysis program by Semantria. The following processes are used by Semantria's cloud-based sentiment analysis program to extract the sentiment of a document and its components:

- A document is broken in its basic parts of speech, called POS tags, which identify the structural elements of a document, paragraph, or sentence (ie Nouns, adjectives, verbs, and adverbs).
- Sentiment-bearing phrases, like "terrible service", are identified via using specifically designed algorithms.
- Each sentiment-bearing phrase in a document scored based on a logarithmic scale ranging between -10 and 10.
- Finally, the scores are combined to ascertain the overall sentiment of the sentence document, ranging between -2 and 2 .

Semantria's cloud-based sentiment analysis software is based on Natural Language Processing and delivers you more consistent results than two humans. Using automated sentiment analysis, Semantria analyses each document and its components based on sophisticated algorithms developed to extract sentiment from your content in a similar manner as a human – only 60,000 times faster.

Existing approaches to sentiment analysis are often grouped into three main categories:

- Keyword spotting
- Lexical affinity
- Statistical methods

Keyword spotting is probably the naivest approach and might also be the most popular one because of its accessibility and economy. Text is labelled into effect categories based on the presence of unambiguous affect words like ‘happy’, ‘sad’, ‘afraid’, and ‘bored’. The weakness of this approach is mainly found in two areas: poor recognition of the affect when negation is involved and reliance on surface features. The approach's first flaw is that it can accurately classify the sentence "today was a cheerful day" as being happy, it’s likely to fail on a sentence like “today wasn’t a cheerful day at all”. About its second weakness, the approach relies on the presence of obvious affect words that are only surface features of the prose. In practice, a lot of sentences convey affect through underlying meaning rather than affect adjectives. For example, the text document "My spouse has filed for divorce, and he wants to take custody of my children away from me" elicits intense emotions, but it contains no effect keywords., and therefore, fails when the keyword spotting approach is applied.

Lexical affinity is slightly more futuristic than keyword spotting as, rather than plainly detecting the affect words, it assigns arbitrary words a probabilistic ‘affinity’ for a particular emotion. For example, ‘accident’ could be assigned a 75% probability of being indicating a negative effect, as in ‘car accident’ or ‘hurt by accident’. These probabilities are usually trained via linguistic corpora. Although it frequently outperforms pure keyword identification, the method has two major flaws. First, statements like "I managed to escape an accident" (negation) and "I met my lover by accident" (negation) can readily fool lexical affinity, which operates simply on the word level (other word senses). Second, the source of the linguistic corpora often biases lexical affinity probabilities towards material of a given genre. This makes creating a reusable, domain-independent model challenging. Statistical methods, like Bayesian inference and support vector machines, are popular for affect classification of texts. By providing a machine learning algorithm a huge training corpus of affectively annotated texts, it’s possible for the system to not only learn the affective valence of affect keywords (as in the keyword spotting approach), but also to consider the valence of other arbitrary keywords (like lexical affinity), punctuation, and word co- occurrence frequencies. However, the traditional statistical methods are semantically weak, as in, except for obvious affect keywords, other lexical or co-occurrence elements in a statistical model have little predictive value individually. As a result, statistical text classifiers only seem to work with acceptable accuracy when given a text input that is sufficiently large. Even though these methods might somewhat be able to affectively classify the user’s text on a page or paragraph level, they do not work very well for smaller text units such as sentences or phrases.

1.2. Objective

Sentiment classification is a method of analyzing the subjective information present in the text and then determining the opinion. On the other hand, sentiment analysis is an approach by which information is drawn out from the opinions, appraisals, and emotions of people in regard to entities, events and their attributes. While deciding, the opinions of other people tend to have a significant effect on customers ease, while making choices with regards to online shopping, choosing events, products, entities.

Our main objective here is that when a review is given by the customer, our model must predict the sentiment of the review to Positive or Negative or Neutral.

1.3. Motivation

Nowadays, everything is getting digital. Gone are those days where everyone used to go to shopping malls to buy products, now everything is just a click away. While shopping online, I got curious about how quickly the company reads all it's reviews, instantaneously knows if the product is a success and acts fast to improve its products and services. While discussing about this with my friend we concluded to work on such a project to understand the inner workings of how a data is classified and the product becomes a hit or not.

Since we are extremely interested in everything having a relation with Machine Learning, this independent project was a great opportunity to give me the time to learn and confirm my interest for this field. The sole fact that one can make estimations, predictions and give the ability for machines to learn by themselves is powerful as well as limitless in terms of application possibilities.

1.4. Language Used

All the code of our project is written only using Python.
Python Machine Learning is one of the most common approaches to create a recommendation system. Python offers probably the most popular and powerful interpreted language, which means that when we build our recommendation system, we will be able to work with others. Python is currently being utilized in production systems all over the world. Once we become familiar with how it works, we can continue using it for real projects instead of having to learn an entirely new language. Knowing Python is a huge competitive advantage to anyone seeking to work in the data science industry.

Python Machine Learning oftentimes goes hand in hand with getting to know AI – one of the top five key trends shaping business in 2020. Python Machine Learning makes AI less intimidating

by simplifying it. This allows us to build more complicated recommendation systems more efficiently and with less stress.

The three concrete ways that this language helped us are:

- Code – With Python, we can write and test code in the easiest way possible. Dealing with algorithms becomes a lot easier because of this. Plus, Python is very malleable when applying to new operating systems and is handy when gluing together different types of data.
- Libraries – A Python library, is a collection of functions and methods that allows us to perform lots of actions without writing our own code. Python has a vast number of libraries to choose from, covering anything from scientific computing to machine learning.

    1. NLTK: Language processing.
    2. Numpy: Scientific computing.
    3. Pandas: Data manipulation and analysis.
    4. Scikit-learn: Machine learning and data mining.
    5. SpaCy: Large scale extracting and analysing of textual information. Support deep learning.
    6. matplotlib: For data visualisation.
    7. Seaborn: Also for data visualisation. Also support for pandas and Numpy.
    8. Plotly: For making publication-quality plots and graphs. Widely used in finance and geospatial industries.
    9. Regular expression specifies a set of strings that matches it
    10. Secure Sockets Layer (SSL) and is designed to create secure connection between client and server.
    11. Imblearn number of re-sampling techniques commonly used in datasets showing strong between-class imbalance.
    12. Pickle: store ML model

- Community – Python has a huge community made up largely of young and ambitious programmers, many of which are more than happy to help each other out on different projects and issues. In addition, Python is completely open source and there is a fair amount of material available online that can teach us all the tips and tricks we need to master it.

MOVING FORWARD WITH PYTHON

Python Machine Learning is not only the leading way to learn how to build a recommendation system, but also one of the best ways to build a recommendation system in general. Knowing a fabulously easy language is a talent that will serve you well throughout your life.
It's a mistake to think about Python as the first step toward sophisticated coding. Python is the current industry standard. Yes, understanding all the sophisticated coding 'ins' and 'outs' is remarkable, but coding doesn't have to be such a time-consuming process. This is particularly true if your primary purpose is to collect data rather than learn to code. Of course, no matter what machine learning system you plan to use, it will take a significant investment of your time.

Machine Learning:

Machine learning is the idea that machines can learn from the preexisting data and work in a way that they need not be explicitly programmed every time for a new dataset. It is a branch of Artificial Intelligence that aims at automating analytical models. Machine learning builds models from test inputs. Machine learning is done where masterminding and programming express computations is unthinkable. Models incorporate spam filtering.

Natural Language Processing:

Sentiment Analysis is a well-known NLP technique that is used to analyze and then classify text information or spoken human language information into specific classes. The Sentiment Analysis technique is used to categorize public opinions by classifying them as positive, neutral, or negative based on polarity.

Since python offers such a large collection of NLP tools and libraries to choose from, we generally use ***Python*** to perform NLP tasks. Other than this, there are other features of Python which makes it one of the best programming language choices for such NLP tasks and projects like the structure of syntax, which is so simple and the transparent semantics of python results in making it such a great choice of tasks that include Natural Language Processing tasks.

Python has so many amazing features that make it so versatile and overall, such a great technology for working on tasks involving machines processing natural languages. We know that Python provides programmers with a vast array of tools that can aid in the performance of natural language processing tasks. One of them is POS tagging, which will be used in the project's sentiment analysis phase. It also allowed coders to create classifications of documents and models.

Since python has such a vast collection of modulus and libraries, it has been used in so many tasks and has therefore replaced many programming languages and has become one of the most popular programming languages, for performing the nlp related sentiment analysis task. Some of the Python libraries that used in performing Machine Learning tasks are:

- Numpy
- Scipy
- Pandas
- Matplotlib
- Seaborn

NumPy:

Another open-source Python library is Numpy which is used for scientific computing and helps users and programmers to work with the mathematical functions, around which the concept of machine learning revolves. Not only this, numpy also allows python to work with most efficient arrays and matrices.

Pandas:

Pandas is a package which helps in data manipulation and is a very important tool when it comes to cleaning of data, exploration of data and visualization tasks as well as data manipulation.

1.5. Technical Requirements

For this project we require data set for E-commerce reviews which is provided by Kaggle and after doing computational work on it will produce an interesting result.

1. Hardware Configuration

Table 1: Hardware Configuration

Processor	Apple M1 chip, 8-core CPU
RAM	8 GB
Hard Disk	256 GB SSD
Monitor	13”

2. Software Configuration

Table 2: Software Configuration

Operating System	Windows 10, Mac OS
Language	Python
Database	CSV, TSV on MsExcel
Tools Used	Jupyter Notebook, GoogleCollab

There can be several technical issues that can be faced:

- Perhaps the biggest issue faced by review systems is that they need a lot of data to effectively decide upon reviews. It’s no coincidence that the companies most identified with having excellent customer reviews are those with a lot of consumer user data.

- Reviews don’t work because there are simply too many product attributes/features, and each attribute has a different level of importance at different times for the same consumer.

1.6.Deliverables

Firstly, we analyze customer reviews and label them as positive or negative i.e., check whether our customer is happy with our products or not. By this we aim to achieve the popularity of our services and further predict the changes required in the for upcoming projects.

Then we feature vectorize and train various ML models to predict how accurate is our prediction. By applying the train and test features on different models we aim to find optimal model best suited for future prediction and purposes.

CHAPTER 2: LITERATURE SURVEY

- Paper: E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework

  Authors: Feng Xu, Zhenchun Pan, Rui Xia

  Xu et al. [1] have introduced a NB method for multi-domain and large-scale E-commerce platform product review classification of sentiment. Consequently, the parameter evaluation method was extended in NB for continuous learning fashion. Many methods were afterwards created for fine-tuning the learnt distribution based on three sorts of assumptions to obtain the optimal performance. The results have shown that the suggested model has high accuracy in Amazon product and movie review sentiment datasets.

- Paper: Twitter Sentiment Analysis: The Good the Bad and the OMG!

  Authors: Efthymios Kouloumpis, Theresa Wilson, Johanna Moore

  Kouloumpis et al. [2] highlighted the efficacy of existing lexical resources and linguistic features for conduct SA on Twitter messages and similar micro-blogging posts. In comparison to part-of-speech traits and those pertaining to an established sentiment lexicon, the researchers believe that microblogging is more relevant and appropriate. The authors concluded that inclusion of microblogging features is not likely to augment training data. Hybrid classification was also found to be more relevant and showed promising results in another research involving rule-based classification and supervised learning processes. In the context of natural language processing (NLP), SA has grown in stature as one of the most researched topics since the turn of the century. Researchers have been constantly examining sentences and assorted types to streamline SA methods.

- Paper: A Literature Survey on Sentiment Analysis Techniques Involving Social Media And Online Platforms

  Authors: Raktim Kumar Dey, Debabrata Sarddar, Indranil Sarkar, Rajesh Bose, Sandip Roy

  Tripathy et. al. [3] proposed in their research, four ML algorithms for sentiment classification. These were Naïve Bayes, Maximum Entropy, Stochastic Gradient Descent and Support Vector Machine. Their research demonstrated that accuracy can be achieved through progressive classification. Their research was conducted on popular movie review website – IMDB. Deploying an n-gram approach, the authors were able to produce consistent high accuracy using a combination of TF-IDF and count vectorizer technique. To address the issue of words and punctuation symbols that, while humanly understandable, lack a formal definition in the English lexicon, the authors created a new list of such words to aid with SA.

- Paper: Open Domain Targeted Sentiment

  Authors: M. Mitchell, J. Aguilar, T. Wilson, and B. Van Durme

  Mitchel et al. [4] modelled sentiment detection in their research paper. The goal of their research was to demonstrate the applicability of sentiment detection as a sequence tagging problem. [56]. Their work was expanded upon by later research work [57] that examined embedding of words and automated combination of features through neural networks. Arun et. al. proceeded to conduct SA on tweets involving demonetization in Indian economy. Their research approach [58] was to extract data from Twitter and convert such into text. This text was to act as input dataset. SA was then performed following removal of stop words so that determination of polarity of the words could be carried out and the actual tweets, therefore, could be identified as either positive or negative.

- Paper: Tourism Mobile App with Aspect-Based Sentiment Classification Framework for Tourist Reviews

  Authors: Muhammad Afzal, Muhammad Usman, Alvis Fong

  Afzaal et al. [5] have recommended a novel approach of aspect-based sentiment classification, which recognized the features in a precise manner and attained the best classification

accuracy. Moreover, the scheme was developed as a mobile application, which assisted the tourists in identifying the best hotel in the town, and the proposed model was analyzed using the real-world data sets. The findings revealed that the proposed model was capable of both recognition and classification.

- Paper: Fuzzy Rule based Unsupervised Sentiment Analysis from Social Media Posts

  Authors: Srishti Vashishtha, Seba Susan

  Vashishtha and Susan [6] have calculated the sentiment related to social media posts by a new set of fuzzy rules consisting of many datasets and lexicons. The developed model combined Word Sense Disambiguation and NLP models with a new unsupervised fuzzy rule-based model for categorising the comments into negative, neutral, and positive sentiment class. The experiments were performed on 3 sentiment lexicons, four existing models, and nine freely available twitter datasets. The results showed that the new method produced the best results.

- Paper: Semi-supervised distributed representations of documents for sentiment analysis

  Authors: Saerom Park, Jaewook Lee, Kyoungok Kim

  Park et al. [7] have developed a semi-supervised sentiment-discriminative objective for resolving the issue by documents partial sentiment data. The suggested model not only reflected the partial data, but also secured the local structures obtained from real data. On real-time datasets, the proposed model was tested. The results have shown that the suggested model was performing well.

CHAPTER 3: SDLC

1. Feasibility Study

A feasibility study is said to be a preliminary study which investigates the details of prospective users and determines the resources requirements, costs, welfare, and feasibility of the proposed system. It takes under consideration, all the constraints within which the implementation and operation of the system takes place. In this stage, the resources needed for the implementation like computing equipment, manpower and costs are estimated. The estimated resources are compared with the available resources and a cost benefit analysis of the system is done. The feasibility analysis activity involves the analysis of the issue and then collecting all relevant information related to the project. The main objective of the feasibility study is to determine whether the project would be feasible in terms of economic feasibility, technical feasibility and operational feasibility and schedule feasibility or not. Also, to make sure that the input data which is required for the project isavailable. Hence, we have evaluated the feasibility of the system into the following categories:

- Technical feasibility
- Operational feasibility
- Economic feasibility
- Schedule feasibility

1.1. Technical Feasibility

The evaluation of technical feasibility is a tricky part it comes to of a feasibility study. This is because, now there isn't any detailed designed of the system, making it difficult to access issues like performance, costs (on account of the kind of technology to be deployed) etc. Several issues need to be considered while doing a technical analysis; understand the various technologies used within the proposed system. Before commencing the project, we must be very clear about what are the technologies that are required for the development of the new system.
Whether the required technology is available or not? Because all the necessary tools are readily available, our approach is theoretically possible. Although all tools that seem to be easily available come with challenges too.

1.2. Operational Feasibility

Proposed project is useful if it can be turned into information systems that will meet the requirements of the operation. In simpler terms, this test of feasibility asks whether the system will work when it is developed and installed. Are there major barriers to Implementation? The proposed is to form a simplified web application. It is simpler to work upon and can be utilized any webpages. It is free and not costly to operate.

1.1.3. Economic Feasibility

It attempts to weigh the costs of developing and implementing a new system, against the benefits that would increase over a period from having the new system in place. This feasibility study done for the new system gives the top management the economic justification. A simple economic analysis which provides the true comparison of costs and benefits are far more meaningful in this case. Furthermore, this proves to be a useful point of reference to differentiate actual costs as the project progresses. There could be several intangible advantages on account of automation. These could increase improvement in product quality, better decision-making ability, and timeliness of data, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of the data. It is a web-based application. Creating the application is not costly.

1.1.4. Schedule Feasibility

A project is basically of no use if it takes too long to be completed before it becomes useful. Typically, this is about estimating how long the system will take to develop, and whether it can be completed in each period using some methods such as payback period. The feasibility of a schedule is a metric for determining how feasible a project's schedule is. Given our technical expertise, are the deadlines given for the project, reasonable? Some project is initiate with specific deadlines. It is necessary to figure out whether the deadlines are mandatory or desirable. A minor deviation can be encountered in the original schedule decided at the commencement of the project. The application development is possible in terms of schedule.

1.2.Requirement Definition

After the extensive analysis of the issues faced within the system, we are familiarized with the needs and requirements of the current system. The system's requirements are divided into two categories: functional and non-functional requirements. These requirements are listed below:

1.2.1. Functional Requirements

Functional requirement are those functions or features that must be included in any system to fulfil the business needs and be acceptable to the users. Based on this definition, the functional requirements needed by the system are as follows:

- System should be able to process new reviews stored in database after retrieval
- System should be able to analyze data and classify the polarity of each review

1.2.2. Non-Functional Requirements

Non-functional requirements are a description of features, characteristics, and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. The non-functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. The following are the non-functional needs based on these:

- User friendly
- System should provide better accuracy
- To perform with efficient throughput and response time

To further elaborate, we can state the nonfunctional requirements based on the following needs:

- Performance
  The response time of the application aimed not to exceed 10 seconds for each interaction. The first compilation of this much large dataset takes much time but over the time after compiling for several times, the system gets faster.

- Reliability
  For any transaction, the system shall respond and should not result in failure. In the case of any failure, the user shall reattempt connection to the system and pass the reviews again.

- Availability
  The system shall be available for the user for any reviews that the user tries to make with the review system. If the system doesn't provide polarity, the user shall reattempt to run the polarity code again.

- Security
  The data that we work on should be secured. The identity of the customer should be remained enclosed.

- Maintainability
  The system should be low maintenance. It must not break down regularly or when big data is being processed.

- Portability
  The application will be independent of the underlying OS or hardware. The application only requires the presence of any incorporated middleware and the databases used on the machine on which the application is being ported.

1.3. Use Case Diagram

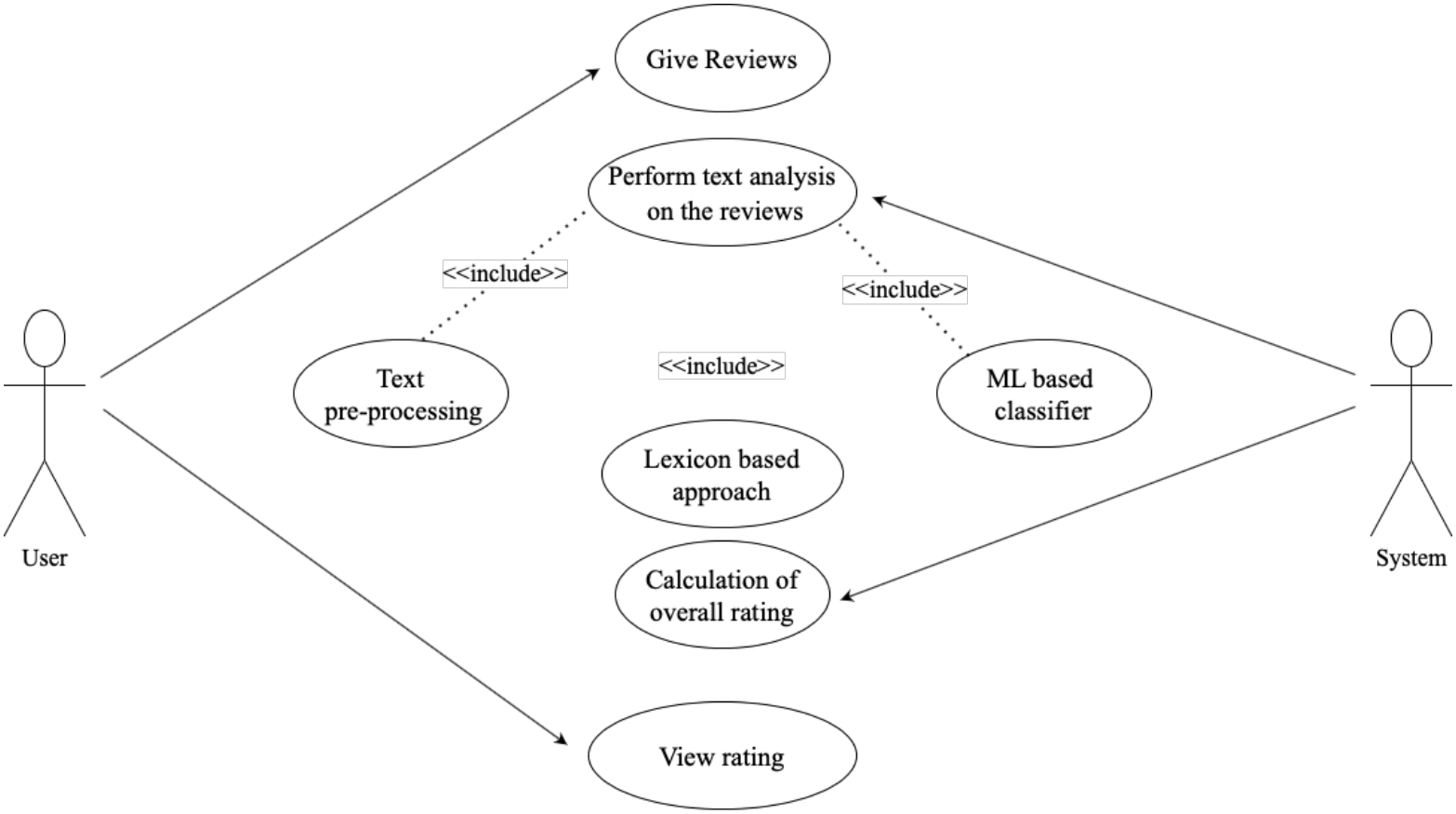

Figure 1: Use Case Diagram

1.4. DFD Diagram

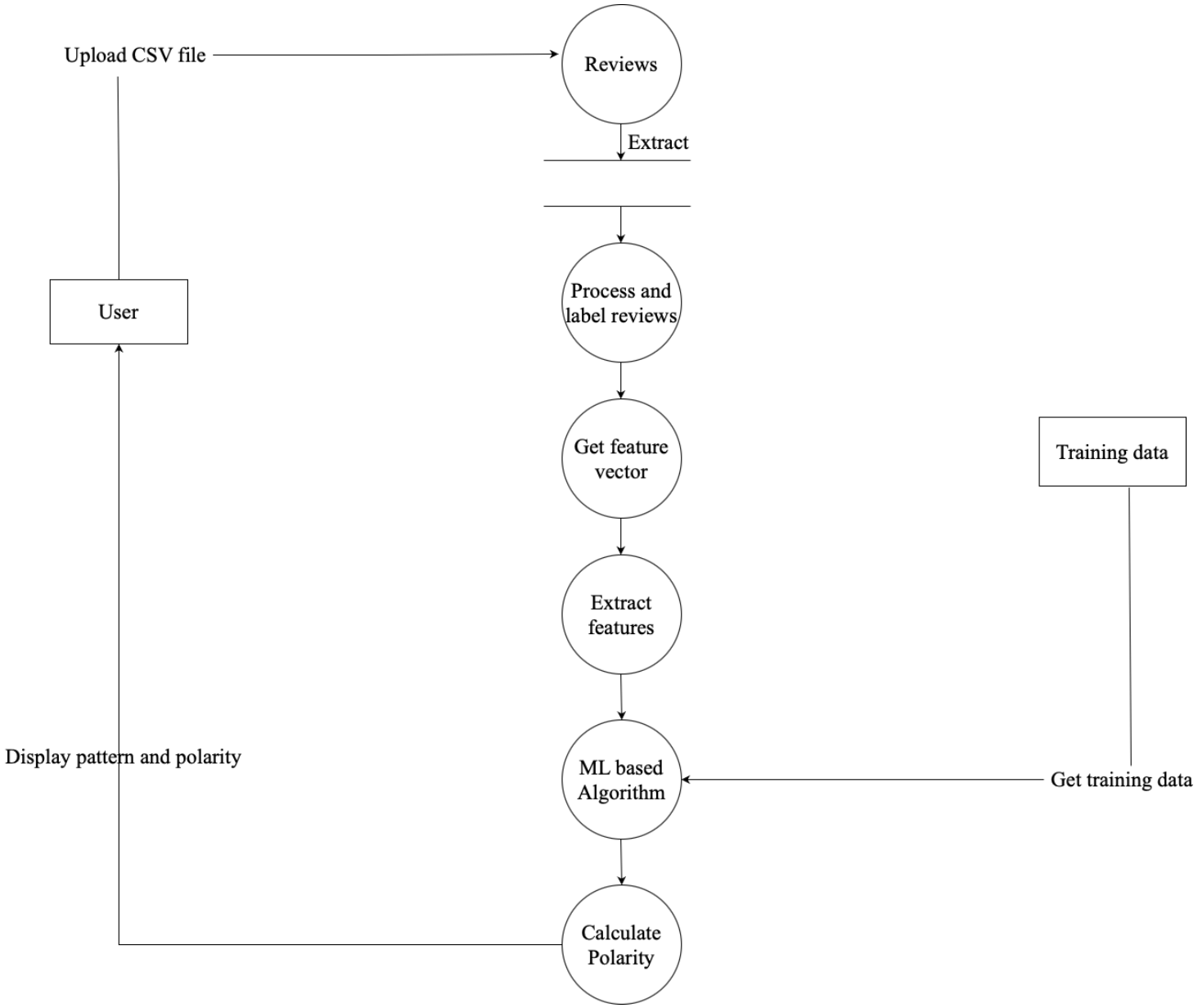

Figure 2: Data Flow Diagram

1.5. State Transition Diagram

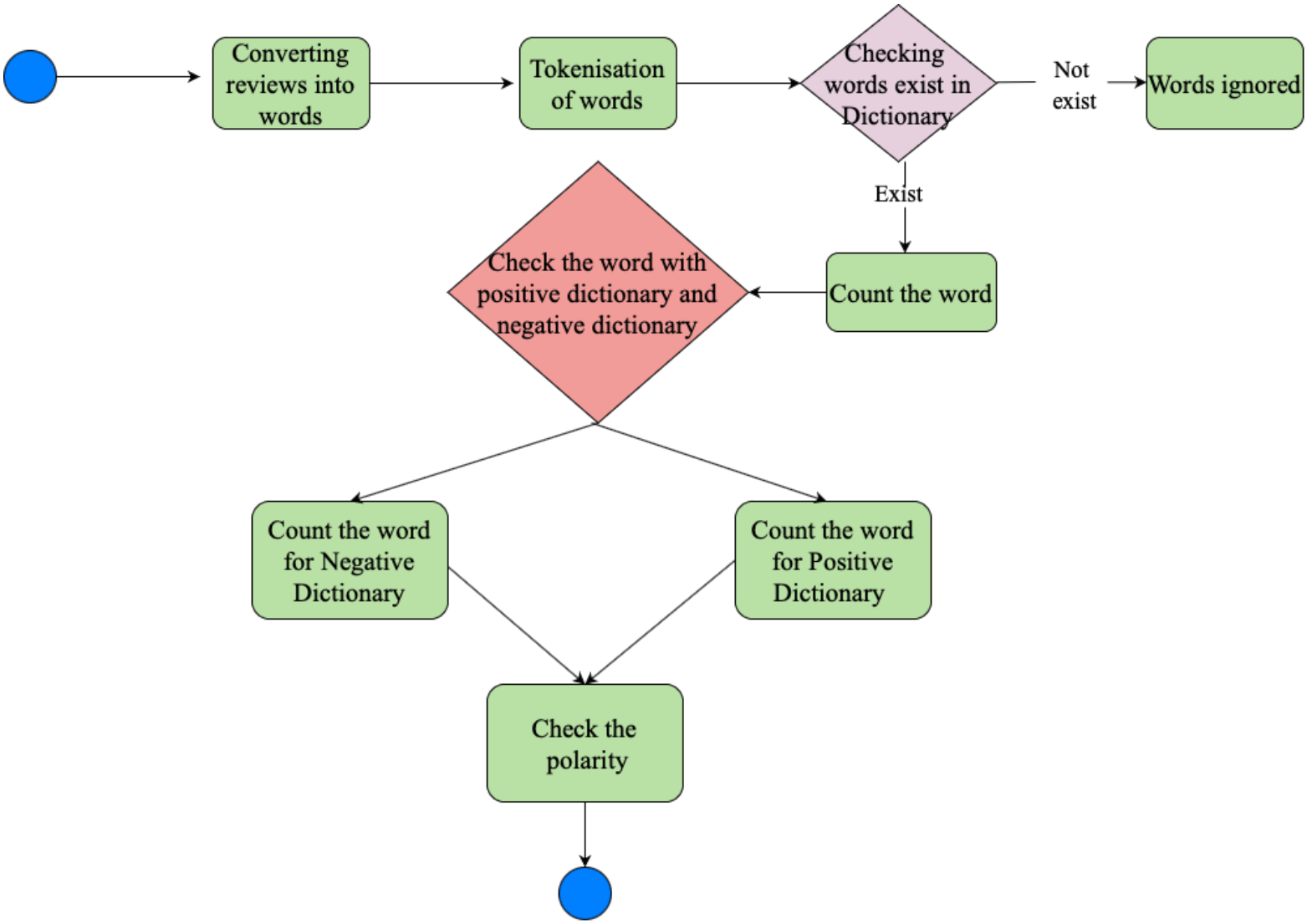

Figure 3: State Transition Diagram

CHAPTER 4: IMPLEMENTATION

1. Data Set Used

The dataset used is a Women's Clothing E-Commerce dataset from Kaggle revolving around the reviews written by customers. Its supportive features offer a great environment to parse out the text through its multiple dimensions. Because this is real commercial data, it has been anonymized, and references to the company in the review text and body have been replaced with "retailer".

2. Data Set Features

There are 23486 rows and 10 feature variables in this dataset. Each row corresponds to a customer review.

2.1. Types of Data Set

Women Clothing E-Commerce Reviews

	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	767	33		Absolutely wonderful - silky and sexy and comfortable	4	1	0	Initmates	Intimate	Intimates
1	1080	34		Love this dress! it's sooo pretty. i happened to find it in a store,	5	1	4	General	Dresses	Dresses
2	1077	60	Some major design flaws	I had such high hopes for this dress and really wanted it to work	3	0	0	General	Dresses	Dresses
3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, flirty, and fabulous! every	5	1	0	General Petite	Bottoms	Pants
4	847	47	Flattering shirt	This shirt is very flattering to all due to the adjustable front tie. it	5	1	6	General	Tops	Blouses
5	1080	49	Not for the very petite	I love tracy reese dresses, but this one is not for the very petite.	2	0	4	General	Dresses	Dresses
6	858	39	Cagrcoal shimmer fun	I aded this in my basket at hte last mintue to see what it would l	5	1	1	General Petite	Tops	Knits
7	858	39	Shimmer, surprisingly goes with lots	I ordered this in carbon for store pick up, and had a ton of stuff (	4	1	4	General Petite	Tops	Knits
8	1077	24	Flattering	I love this dress. i usually get an xs but it runs a little snug in bus	5	1	0	General	Dresses	Dresses
9	1077	34	Such a fun dress!	I'm 5"5' and 125 lbs. i ordered the s petite to make sure the leng	5	1	0	General	Dresses	Dresses
10	1077	53	Dress looks like it's made of cheap material	Dress runs small esp where the zipper area runs. i ordered the s	3	0	14	General	Dresses	Dresses

Figure 4: Data set used (part1)

.....

Figure 5: Data set used (part 2)

965	1140	36		This skirt is not what i was expecting at all. for starters it runs lar	3	0	3	General Petite	Trend	Trend
966	936	55		I really did not need another coat at all, but i couldn't resist. got	5	1	5	General	Tops	Sweaters
967	1047	43	The fabric is heavy	I wanted to love these pants, but they were larger than i expecte	3	0	0	General	Bottoms	Pants
968	936	32		It pills a little under the arms and picks up lint, but it's totally wo	5	1	0	General	Tops	Sweaters
969	868	55		Omg - wish it came in more colors,,	5	1	2	General	Tops	Knits
970	1094	54	Beautiful fit	This dress has a beautiful fit; eyed it for awhile--snatched it up a	5	1	0	General	Dresses	Dresses
971	936	58		Absolutely beautiful. as soon as i saw the color i had to have it. I	5	1	0	General	Tops	Sweaters
972	1094	67		Very cheap looking material. looks cheap to cost $158.	1	0	1	General	Dresses	Dresses
973	1033	47	Finally jeans for curves!	I've been a fan of the pilcro chinos for years but the jeans have r	5	1	1	General Petite	Bottoms	Jeans
974	936	31			5	1	0	General	Tops	Sweaters
975	1094	36	A winner!	What a well-designed dress! the cut is a slight a-line without add	5	1	6	General	Dresses	Dresses
976	154	48	Cozy comfy beautiful top	I was initially attracted to the colors	5	1	1	Initmates	Intimate	Sleep
977	936	21	Warm and cozy	This coat is a perfect fall to winter transition coat. it is surprising	4	1	3	General	Tops	Sweaters

.....

1507	1020	47	Versatile, great fit, pockets!	It's rare to find such a great skirt -- flattering, unique design, sub	5	1	0	General Petite	Bottoms	Skirts
1508	863	60	Sleeves stretch out	I really liked this top when i tried it on but as i wore it the bottom	4	0	3	General	Tops	Knits
1509	1020	38	Great fit for curves	This skirt fits my curves so well. the adjustable waist is so helpfu	5	1	1	General Petite	Bottoms	Skirts
1510	977	31	Enormous!!	I know this product description says oversized but that's an und	1	0	0	General Petite	Jackets	Jackets
1511	1080	39	Red "evanthe" dress	Agreed with the previous reviewer, this is the same dress as the	4	1	1	General	Dresses	Dresses
1512	912	46	Haven't washed well	I bought three of these this summer--a mint, ivory, and yellow. w	3	0	0	General Petite	Tops	Fine gauge
1513	865	63	Beautiful, flattering top	Ordered this top online, color is a little more gray than green but	5	1	2	General	Tops	Knits
1514	850	33	I wear my sunglasses	I love this blouse! the colors of the little sunglasses are vibrant a	5	1	0	General	Tops	Blouses
1515	1080	56	Does not fit well but is beautiful	This is such a beautiful dress. but have to return for the reasons	2	0	0	General	Dresses	Dresses
1516	1080	40	Small-chested women need not apply	Girls, this dress is gorgeous. but if you are an a/b cup, there is j	5	1	0	General	Dresses	Dresses
1517	947	48	Beautiful sweater but not flattering on me	Really wanted this to worked for me. ordered the s in ivory (the c	3	1	0	General	Tops	Sweaters
1518	1020	37	Pop of color	This skirt is pleasantly more dynamic in person. the print and co	5	1	8	General Petite	Bottoms	Skirts
1519	850	32	Dress up or down	This top can be dressed up or down, very good quality fabric an	5	1	7	General	Tops	Blouses

Figure 6: Data set used (part 3)

1.2.2.Number of Attributes, Fields, Description of the data set

This data set has ten feature variables, which are described below-

1. Clothing ID: Integer Categorical variable that refers to the specific piece being reviewed.
2. Age: Positive Integer variable of the reviewer's age.
3. Title: String variable for the title of the review.
4. Review Text: String variable for the review body.
5. Rating: Positive Ordinal Integer variable for the product score granted by the customer from 1 Worst to 5 Best.
6. Recommended IND: Binary variable stating where the customer recommends the product where 1 is recommended, 0 is not recommended.
7. Positive Feedback Count: Positive Integer documenting the number of other customers who found this review positive.
8. Division Name: Categorical name of the product high level division.
9. Department Name: Categorical name of the product department name.
10. Class Name: Categorical name of the product class name.

Furthermore, every single row gives us the customer review for a particular clothing item.

1.3. Design of Problem Statement

The designing of a problem statement is one of the key steps before starting a project. If one doesn't clearly understand what they really want, then they will come across many problems that they did not even account for.
This is a digital era. Those days are long gone when everyone used to go to shopping marts to buy products. Everything is just a click away in today's world.. With the boom in internet companies, there has been a surge in the competition in the industry, and hence to retain their customers, companies need to analyze the feedback given by the customers and then evolve over time. Since there are millions of customers, it is almost impossible to manually review each feedback or sentiment. That is exactly where our problem lies. Our job is to find the best fitting classifier which tells us about the sentiment of the customer based on the review they have given.

1.4. Algorithm / Pseudo Code

1. Read the CSV file.
2. Select Features
3. Clean dataset
   3.1. Remove missing values using deletion
   3.2. Drop non-selected features
4. Data pre-processing
   4.1. Remove special characters, hashtags, links, extra spaces, and social media handles
   4.2. Tokenize and remove stop words
   4.3. Lemmatise words
5. Labelling data using SentiWord, where 1 if positive and 0 otherwise.
   SentiWordNet is a lexical resource for opinion mining. It assigns to each synsetof WordNet three sentiment scores- positivity, negativity, and objectivity.
6. Balancing the unbalanced dataset
7. Featuring via test-train-split
8. Applying the various Machine Learning Models such as Logistic regression, SVM, Random Forest, etc.
   Metrics used:
   F1_score(weighted), Confusion Matrix.
   F1 = 2 * (precision * recall) / (precision + recall)
   The F1_score can be interpreted as a harmonic average of precision and recall.
9. Evaluation: Comparing and finding optimal model.

1.5. Flow Graph

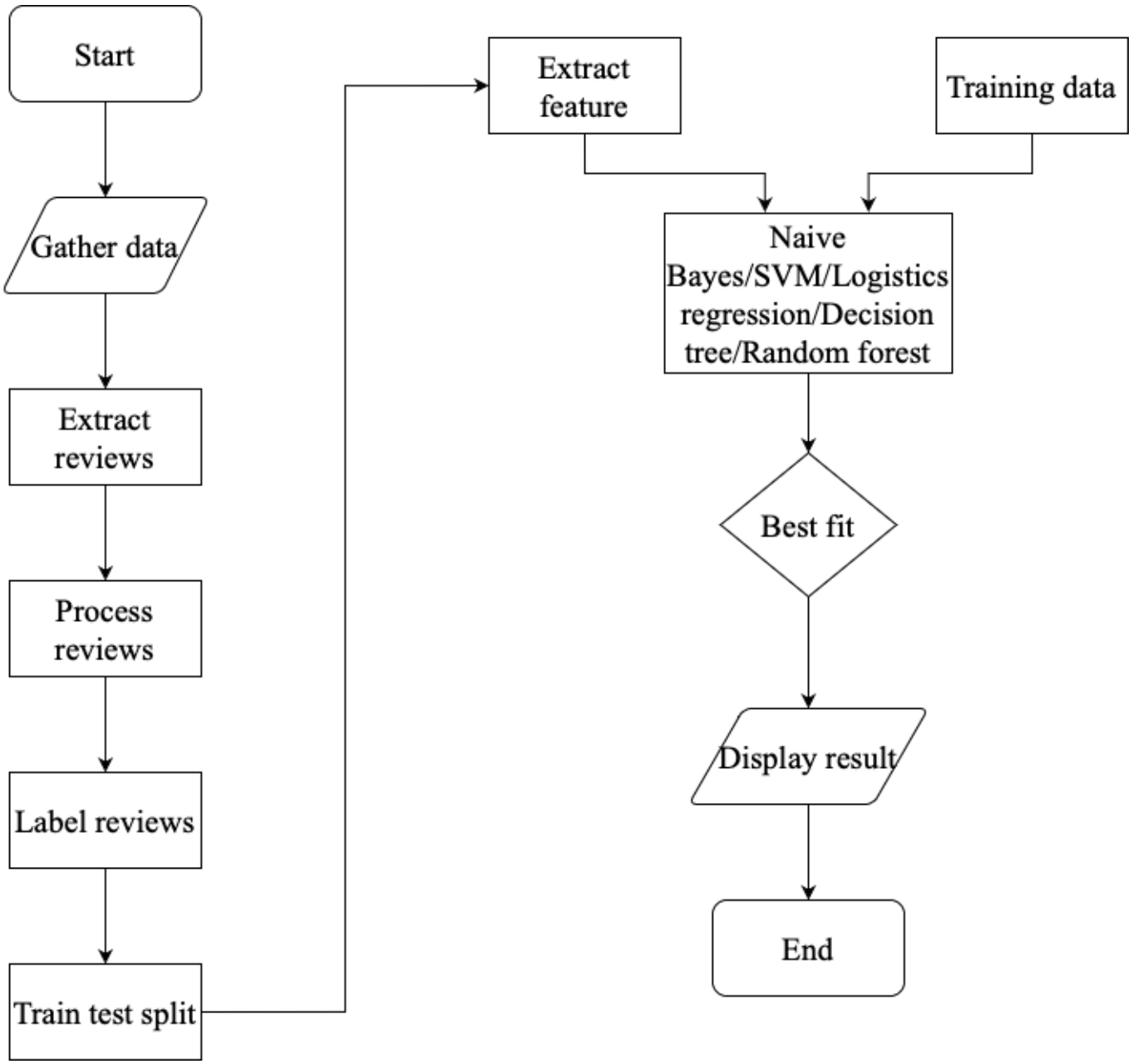

Figure 7: Flow Graph

1.6. Screenshots of Various

Stages

Following mentioned are the various stages involved in this project:

1.6.1. Data Cleaning

In this stage we handle missing values and remove unwanted features.

Data Cleaning

```
[ ]  #rename data-coloumn
     data=data.rename(columns={'Review Text': 'Reviews'})
```

```
[ ]  #remove missing values
     data=data.dropna()
     data.reset_index(drop=True, inplace=True)
     data.isna().sum()
```

```
Unnamed: 0                0
Clothing ID               0
Age                       0
Title                     0
Reviews                   0
Rating                    0
Recommended IND           0
Positive Feedback Count   0
Division Name             0
Department Name           0
Class Name                0
dtype: int64
```

```
[ ]  #droping extra columns
     data=data.drop(["Clothing ID","Division Name","Department Name","Class Name","Title","Age",
                     "Positive Feedback Count","Recommended IND","Unnamed: 0","Rating"],axis=1)
     data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19662 entries, 0 to 19661
Data columns (total 1 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
```

Figure 8: Data Cleaning

1.6.2. Data Pre-processing

In this stage we process data by first removing all extra spaces, special characters, social media handles and tags using regular expression.
After that we'll remove stopwords using stopwords library present in NTLK package. Lastly, we will lemmatise our data using WordNetLemmatizer that is removing words less than 2,numbers, etc.

```
def preprocess_Reviews_data(data,name):
    # Proprocessing the data
    data[name]=data[name].str.lower()
    # Code to remove the Hashtags from the text
    data[name]=data[name].apply(lambda x:re.sub(r'\B#\S+','',x))
    # Code to remove the links from the text
    data[name]=data[name].apply(lambda x:re.sub(r"http\S+", "", x))
    # Code to remove the Special characters from the text
    data[name]=data[name].apply(lambda x:' '.join(re.findall(r'\w+', x)))
    # Code to substitute the multiple spaces with single spaces
    data[name]=data[name].apply(lambda x:re.sub(r'\s+', ' ', x, flags=re.I))
    # Code to remove all the single characters in the text
    data[name]=data[name].apply(lambda x:re.sub(r'\s+[a-zA-Z]\s+', '', x))
    # Remove the twitter handlers
    data[name]=data[name].apply(lambda x:re.sub('@[^\s]+','',x))

# Function to tokenize and remove the stopwords
def rem_stopwords_tokenize(data,name):

    def getting(sen):
        example_sent = sen

        filtered_sentence = []

        stop_words = set(stopwords.words('english'))

        word_tokens = word_tokenize(example_sent)

        filtered_sentence = [w for w in word_tokens if not w in stop_words]

        return filtered_sentence
    # Using "getting(sen)" function to append edited sentence to data
    x=[]
    for i in data[name].values:
        x.append(getting(i))
```

Figure 9: Pre-processing data (part 1)

```
            return filtered_sentence
        # Using "getting(sen)" function to append edited sentence to data
        x=[]
        for i in data[name].values:
            x.append(getting(i))
        data[name]=x
```

+ Code + Text

```
[ ]  lemmatizer = WordNetLemmatizer()
    def Lemmatization(data,name):
        def getting2(sen):

            example = sen
            output_sentence =[]
            word_tokens2 = word_tokenize(example)
            lemmatized_output = [lemmatizer.lemmatize(w) for w in word_tokens2]

            # Remove characters which have length less than 2
            without_single_chr = [word for word in lemmatized_output if len(word) > 2]
            # Remove numbers
            cleaned_data_title = [word for word in without_single_chr if not word.isnumeric()]

            return cleaned_data_title
        # Using "getting2(sen)" function to append edited sentence to data
        x=[]
        for i in data[name].values:
            x.append(getting2(i))
        data[name]=x
```

```
[ ]  def make_sentences(data,name):
        data[name]=data[name].apply(lambda x:' '.join([i+' ' for i in x]))
        # Removing double spaces if created
        data[name]=data[name].apply(lambda x:re.sub(r'\s+', ' ', x, flags=re.I))
```

Figure 10: Pre-processing data (part 2)

1.6.3. Labelling Data

In this stage we'll label our Reviews using SentiWordNet. It is a lexical resource for opinion mining. It assigns three sentiment scores to each WordNet synset: positivity, negativity, and objectivity.

	Reviews
0	high hope dress really wanted work meinitially...
1	love love love jumpsuit itfun flirty fabulous ...
2	shirt flattering due adjustable front tie perf...
3	love tracy reese dress one petiteam foot tall ...
4	aded basket hte last mintue see would look lik...

Figure 11: Reviews

1.6.4. EDA

- Imbalance data- By plotting the following graph we concluded that our data is highly imbalanced. This can pose problems for our model's performance; we will address this issue during Modelling.
- We found top 10 words using IDF-weights. From the word cloud, the top 10 keywords are **love, well, great, petite, comfortable** etc. These words make sense. So, our data looks fine.

```
pos=neg=obj=count=0

postagging = []

for review in data['Reviews']:
    list = word_tokenize(review)
    postagging.append(nltk.pos_tag(list))

data['pos_tags'] = postagging

def penn_to_wn(tag):
    if tag.startswith('J'):
        return wn.ADJ
    elif tag.startswith('N'):
        return wn.NOUN
    elif tag.startswith('R'):
        return wn.ADV
    elif tag.startswith('V'):
        return wn.VERB
    return None


# Returns list of pos-neg and objective score. But returns empty list if not present in senti wordnet.
def get_sentiment(word,tag):
    wn_tag = penn_to_wn(tag)

    if wn_tag not in (wn.NOUN, wn.ADJ, wn.ADV):
        return []
```

Figure 12: Labelling data

```
        #Lemmatization
        lemma = lemmatizer.lemmatize(word, pos=wn_tag)
        if not lemma:
            return []

        #SentiWordNet is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet thre
        #Synset is a special kind of a simple interface that is present in NLTK to look up words in WordNet.
        #Synset instances are the groupings of synonymous words that express the same concept.
        #Some of the words have only one Synset and some have several.
        synsets = wn.synsets(word, pos=wn_tag)
        if not synsets:
            return []

        # Take the first sense, the most common
        synset = synsets[0]
        swn_synset = swn.senti_synset(synset.name())

        return [synset.name(), swn_synset.pos_score(),swn_synset.neg_score(),swn_synset.obj_score()]

        pos=neg=obj=count=0

        ###############################################################################
    senti_score = []

    for pos_val in data['pos_tags']:
        senti_val = [get_sentiment(x,y) for (x,y) in pos_val]
        for score in senti_val:
            try:
                pos = pos + score[1]  #positive score is stored at 2nd position
                neg = neg + score[2]  #negative score is stored at 3rd position
            except:
                continue
        senti_score.append(pos - neg)
        pos=neg=0
```

Figure 13: Lemmatization

	Reviews	pos_tags	senti_score	Rating
19657	happy snag dress suchgreat price itvery easy s...	[(happy, JJ), (snag, NN), (dress, NN), (suchgr...	1.250	Positive
19658	reminds maternity clothes soft stretchy shiny ...	[(reminds, NNS), (maternity, NN), (clothes, NN...	-0.625	Negative
19659	fit well top see never would worked mem gladwa...	[(fit, RB), (well, RB), (top, JJ), (see, NN), ...	0.500	Positive
19660	bought dress forweddinghave summer itso cute u...	[(bought, VBD), (dress, NN), (forweddinghave, ...	0.875	Positive
19661	dress inlovely platinum feminine fit perfectly...	[(dress, NN), (inlovely, RB), (platinum, JJ), ...	0.500	Positive

```
[ ] def convert_sentiment(sentiment):
        if sentiment=='Negative':
            return 0
        else:
          return 1;
    data.Rating = data.Rating.apply(convert_sentiment)
```

```
[ ] data=data.drop(['pos_tags','senti_score'], axis = 1)
    data.head()
```

	Reviews	Rating
0	high hope dress really wanted work meinitially...	0
1	love love love jumpsuit itfun flirty fabulous ...	1
2	shirt flattering due adjustable front tie perf...	1
3	love tracy reese dress one petiteam foot tall ...	0
4	aded basket hte last mintue see would look lik...	1

Figure 14: Reviews and Ratings

1.6.5. Balance Dataset

In this stage we balanced our dataset using oversampling technique called Smote which duplicate or create new synthetic examples in the minority class.

```
[ ]  #imbalance-dataset
     sns.set(style="darkgrid")
     bar = sns.countplot(x = 'Rating' , hue = 'Rating' , data = data, palette="flare")
     bar.set_title("Overall Sentiment", size = 20)
     plt.show()
```

Figure 15: Imbalance dataset

Figure 16: Keywords

Balancing Dataset

```
X=data["Reviews"]
Y=data["Rating"]
```

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
    tf=TfidfVectorizer(min_df=0,max_df=1,use_idf=True,ngram_range=(1,2))

    X=tf.fit_transform(X)
```

```
[ ] from imblearn.combine import SMOTETomek
    from imblearn.under_sampling import NearMiss

    smk = SMOTETomek(random_state=42 , sampling_strategy = 0.8)
    X_res,y_res= smk.fit_sample(X,Y)

    unique_elements, counts_elements = np.unique(y_res, return_counts=True)
    np.asarray((unique_elements, counts_elements))
```

```
array([[    0,     1],
       [10872, 13591]])
```

Figure 17: Balancing dataset

1.6.6. Modelling

Here we applied various ML algorithms to find out the optimal model.
So, for this, we have experimented with the following algorithms-

- SVM
- Decision trees
- Naive Bayes
- Random forests
- Logical Regression

Logistic Regression

```
[ ]  #LogisticRegression
     from sklearn.linear_model import LogisticRegression
     logistic=LogisticRegression(penalty='l2',max_iter=500,C=1,random_state=0)
     logistic.fit(x_train1,y_train1)
     #Generate predictions with the model using our X values
     y_pred_lr = logistic.predict(x_test1)
```

```
[ ]  #classification_report
     lr_report1=classification_report(y_test1,y_pred_lr,target_names=['0','1'])
     print(lr_report1)
     y_pred = logistic.predict(x_train1)
     # lr_report2=classification_report(y_train1,y_pred)
     # print(lr_report2)
```

```
                precision    recall  f1-score   support

             0       1.00      0.36      0.53      2223
             1       0.65      1.00      0.79      2670

      accuracy                           0.71      4893
     macro avg       0.83      0.68      0.66      4893
  weighted avg       0.81      0.71      0.67      4893
```

Figure 18: Logical Regression

Naive Bayes Classifier - MultinomialNB

```
[ ]  #Naive Bayes
     from sklearn.naive_bayes import MultinomialNB
     mnb=MultinomialNB()
     mnb_model=mnb.fit(x_train1,y_train1)
```

```
     mnb_bow_predict=mnb_model.predict(x_test1)
     mnb_bow1=mnb_model.predict(x_train1)
```

```
[ ]  mnb_bow_report = classification_report(y_test1,mnb_bow_predict)
     print(mnb_bow_report)
     # mnb_bow_repor = classification_report(y_train1,mnb_bow1)
     # print(mnb_bow_repor)
```

```
                precision    recall  f1-score   support

           0       0.85      0.57      0.68      2223
           1       0.72      0.91      0.81      2670

    accuracy                           0.76      4893
   macro avg       0.78      0.74      0.74      4893
weighted avg       0.78      0.76      0.75      4893
```

Figure 19: Naive Bayes Classifier

Support Vector Machine

```
     #SVM
     from sklearn import svm
     model2 = svm.SVC()
     model2.fit(x_train1,y_train1)

     # testing
     y_pred_svm = model2.predict(x_test1)

     c=classification_report(y_test1,y_pred_svm)
     print(c)
```

```
                precision    recall  f1-score   support

           0       0.96      0.58      0.72      2223
           1       0.74      0.98      0.84      2670

    accuracy                           0.80      4893
   macro avg       0.85      0.78      0.78      4893
weighted avg       0.84      0.80      0.79      4893
```

Figure 20: SVM

Decision Trees Classifier

```
[ ]  #Decision Tree
     from sklearn.tree import DecisionTreeClassifier
     clf = DecisionTreeClassifier()

     #Fit train and test into the model
     clf.fit(x_train1, y_train1)

     #Predict the result
     y_pred_dt = clf.predict(x_test1)
```

```
     c=classification_report(y_test1,y_pred_dt)
     print(c)
```

```
                  precision    recall  f1-score   support

               0       1.00      0.51      0.68      2223
               1       0.71      1.00      0.83      2670

        accuracy                           0.78      4893
       macro avg       0.86      0.75      0.75      4893
    weighted avg       0.84      0.78      0.76      4893
```

Figure 21: Decision Trees Classifier

Random Forest Classifier

```
[ ]  #Random Forest
     from sklearn.ensemble import RandomForestClassifier
     random_forest = RandomForestClassifier()

     random_forest.fit(x_train1, y_train1)

     y_pred_rf = random_forest.predict(x_test1)

     c=classification_report(y_test1,y_pred_rf)
     print(c)
```

```
                  precision    recall  f1-score   support

               0       1.00      0.51      0.68      2223
               1       0.71      1.00      0.83      2670

        accuracy                           0.78      4893
       macro avg       0.86      0.75      0.75      4893
    weighted avg       0.84      0.78      0.76      4893
```

Figure 22: Random Forest Classifier

1.6.7.Comparison and Conclusion

As we can see, almost all models are having the same f1 scores while Logistic Regression is being the least one and SvC is slightly better than other models.

	MLA Model	Accuracy	Precission	F1 Score
0	LogisticRegression	0.7104	0.653291	0.790292
1	MultinomialNB	0.7584	0.719210	0.805079
2	SVC	0.7975	0.735484	0.841059
3	DecisionTreeClassifier	0.7772	0.710106	0.830482
4	RandomForestClassifier	0.7772	0.710106	0.830482

Figure 23: Comparing the 5 models

CHAPTER 5: RESULTS AND CONCLUSION

1. Results Achieved

As we compare the confusion matrices, we can clearly see that SVM predicts maximum truly positive with almost negligible false negative compared to other models.

Figure 24: Confusion Matrix of Logical Regression

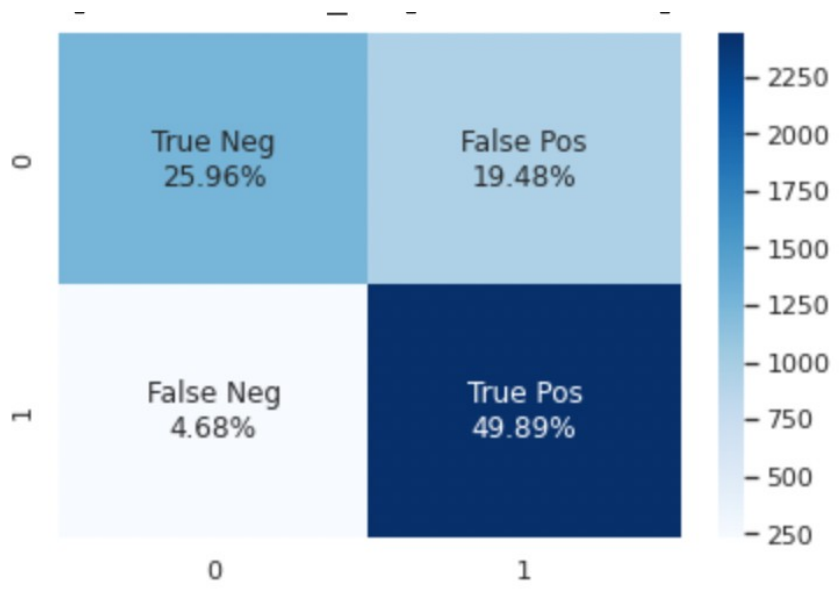

Figure 25: Confusion Matrix of Naive Bayes Classifier

Figure 26: Confusion Matrix of Decision Tree Classifier

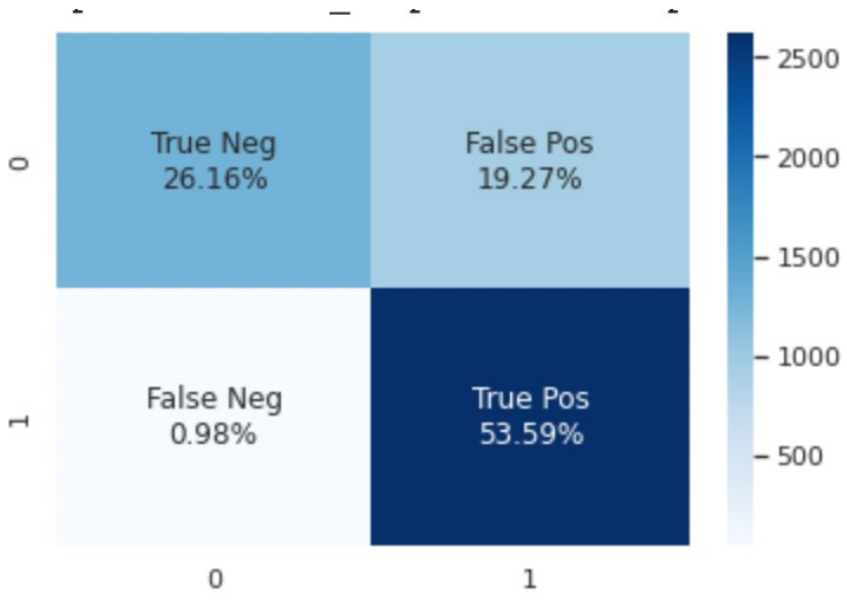

Figure 27: Confusion Matrix of SVM

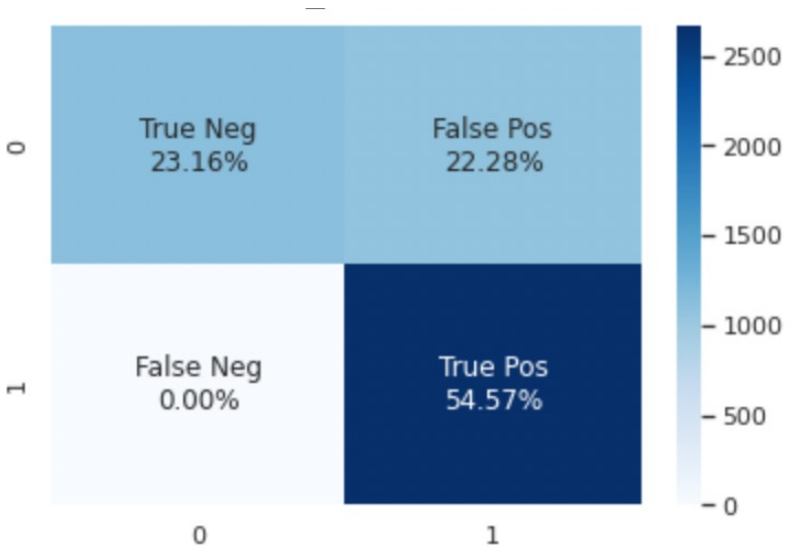

Figure 28: Confusion Matrix of Random Forest Classifier

Also in the following plot, we can see that the f1 scores for the corresponding ML models. By this we can conclude that SVM predicts slight better than other models with prediction of 84% and hence we choose it as our optimal model for deployment and future purposes.

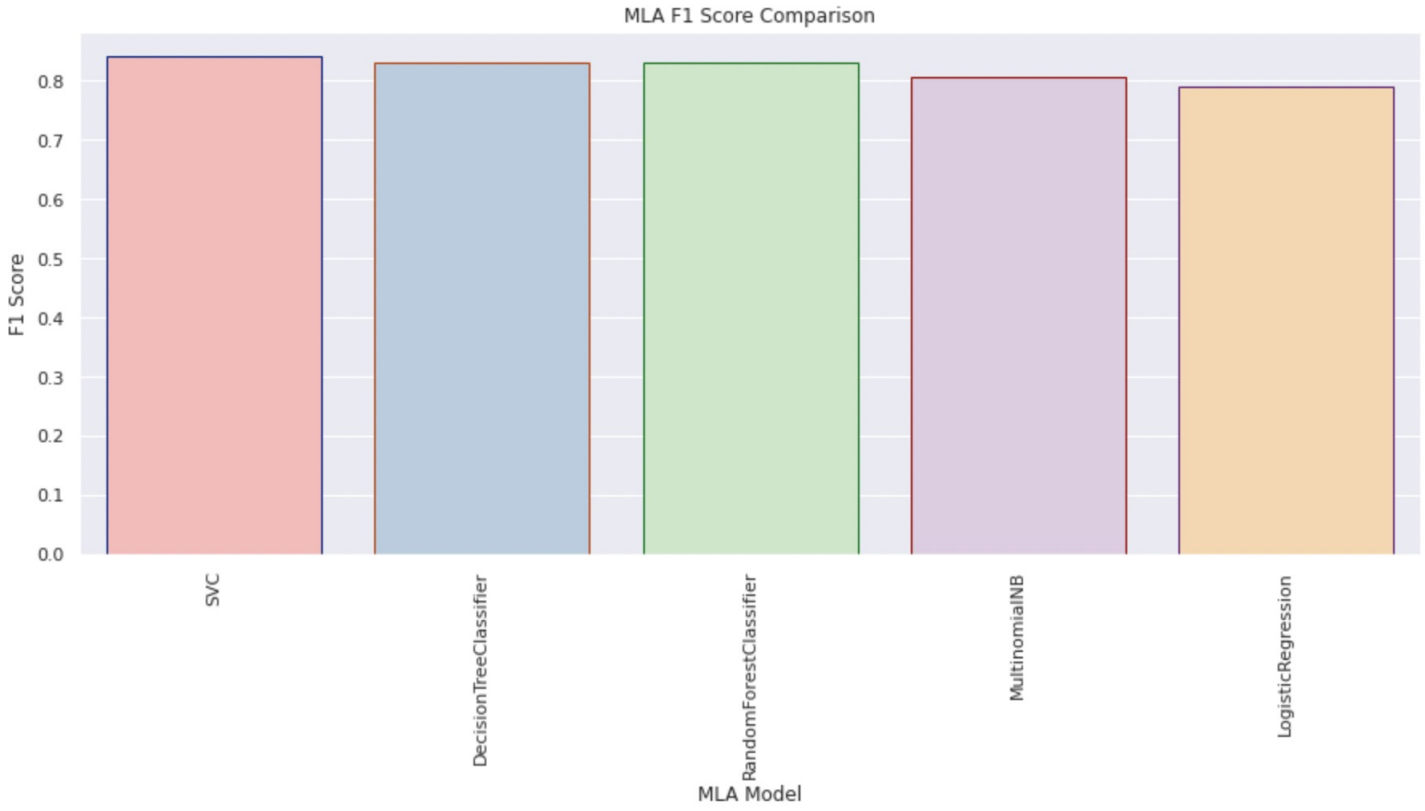

Figure 29: f1 Scores of ML Models

2. Applications

Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative our neutral.

The most typical application of sentiment research in diverse markets is brand monitoring and reputation management. No wonder - understanding how the consumers perceive your brand/ product/ service is equally useful for tech companies, marketing agencies, fashion brands, media organizations, and many others.
In essence, the sentiment analysis application brings additional flexibility and insight into the presentation of the brand and its products. It allows companies to:

- Track the perception of the brand by the customers.
- Point out the specific details about the attitude.
- Find patterns and trends.
- Keep a close eye on the presentation by the influencers.

All this allows us to adjust to the state of things accordingly and give the product a proper presentation.

Overall, sentiment analysis can be used to:

- Automate media monitoring process and the accompanying alert system
- Monitor mentions or reviews of the brand on different platforms (blogs, social media, review sites, forums, etc.)
- Categorize urgency of mentions according to the relevancy scoring (i.e., which platform, type of user is vital to the brand)

The whole operation consists of two stages:

- At the initial stage, the company reacts to the incoming results and adapts.
- Over time, sentiment analysis can transform the course of action from reacting to managing the perception.

The application of sentiment analysis in product analytics can be traced back to reputation management. Conceptually, it is very similar to brand monitoring. Instead of brand mentions, it focuses on comments and observations about the product and how well it performs in certain areas (user interface, feature performance, etc.).
This kind of knowledge is critical in the early stages of MVP development, when you need to put the product to the test (i.e., with real users) and make it as polished as possible.
The most basic approach to use sentiment analysis at this point is to collect and categorize input for future improvements. Sentiment analysis algorithm can do the dirty work and show what kind of feedback goes from which segment of the audience and at what it points.

Typically, the entire thing is separated into the following categories:

- Brand keywords
- Brand-adjacent keywords
- Customer needs
- Customer sentiment
- Competitors analysis (based on similar criteria)

As a result, this can be a significant factor in the product's successful establishment on the market.

At the later stages, the use of sentiment analysis in product analytics merges with brand monitoring and provides a multi-dimensional view of the product and its brand:

- How is the brand/product perceived by various target audience segments?
- Which elements of the product or its presentation are the points of contention and in what light?

3. Limitations

Sentiment, like all opinions, is essentially subjective from one person to the next, and can even be irrational. When attempting to evaluate sentiment, it's vital to mine a broad — and relevant — sample of data. No data point is necessarily relevant. It’s the aggregate that matters. One or more indirect causes may alter an individual's mood toward a brand or product; for example, someone may be having a terrible day and tweet a critical remark about something they otherwise had a neutral view about. With a large enough sample, outliers are diluted in the aggregate. Also, because sentiment is likely to fluctuate over time as a result of a person's mood, world events, and other factors, it's usually vital to examine data throughout time.

When it comes to sarcasm, context is important, just like it is for any other sort of natural language processing (NLP) analysis. The problem of the next 2-3 decades, in my opinion, will be analyzing natural language data. It's a difficult problem, because sarcasm and other forms of sarcastic language are naturally difficult for robots to recognize when seen separately. It’s imperative to have a sufficiently sophisticated and rigorous enough approach that relevant context can be considered. For example, knowing whether a user is sarcastic, ironic, or hyperbolic in general, or having a larger sample of natural language data that provides hints to assess whether a statement is ironic, would be required. The system we designed is used to determine the opinion of the people based on product review data. We somehow completed our project and were able to determine only positivity and negativity of the review. For neutral data, we were unable to merge dataset.

4. Future Work / Scope

- Analyzing sentiments on emoticons/smiley.

- Determining neutrality.
  We are currently only able to determine whether the sentiment is negative or positive. We intend to create the concept of neutrality by adding neutral keywords, so that the results are not biased.

- Potential improvement can be made to our data collection and analysis method.

- Future research can be done with possible improvement such as more refined data and more accurate algorithm.

- Sentiment word dictionaries for sentiment analysis is not a great way since it has low reliability if the rate of categorized good and bad words are limited. Therefore, we could have provided a criterion where, if the number of sentiment words was 5 or less, we could exclude the observations to avoid biased results.

- If the emotion is positive, the reliability should be increased to the positive side, and if it is negative, the reliability should be increased toward the negative side. However, we simply multiplied the usefulCount for reliability and did not consider this part. As a result, we should have multiplied based on the sign of usefulCount for various types of emotions.

REFRENCES

1. FengXu, ZhenchunPan, and RuiXia, "E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework", Information Processing & Management, Available online 13 February 2020.

2. Kouloumpis, T. Wilson, and J. Moore, —Twitter sentiment analysis: "The good the bad and the omg!", In Proceedings of the Fifth International AAAI conference on weblogs and social media, Barcelona, Catalonia, Spain, pp. 538-541, July 2011.

3. L. Qu, G. Ifrim, and G. Weikum,, — "The bag-of- opinions method for review rating prediction from sparse text patterns", InProceedings of the 23rd international conference on computational linguistics, Coling 2010, Beijing, Association for Computational Linguistics, pp. 913-921, August 2010.

4. M. Mitchell, J. Aguilar, T. Wilson, and B. Van Durme, —Open domain targeted sentiment‖, In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, pp. 1643-1654, October 2013.

5. M. Afzaal, M. Usman and A. Fong, "Tourism Mobile App with Aspect-Based Sentiment Classification Framework for Tourist Reviews," IEEE Transactions on Consumer Electronics, vol. 65, no. 2, pp. 233-242, May 2019.

6. SrishtiVashishtha, and SebaSusan, "Fuzzy rule based unsupervised sentiment analysisfrom social media posts", Expert Systems with Applications, vol. 138, 30 December 2019.

7. SaeromPark, JaewookLee, and KyoungokKim, "Semi-supervised distributed representations of documents for sentiment analysis", Neural Networks, vol. 119, pp. 139-150, November 2019.