## SENTIMENTAL PREDICTIONS ON REVIEW USING MULTIPLE ALGORITHMS

#### A PROJECT REPORT

***Submitted by***

**AARTHI.R 211418104003**

**DEEPIKA.M 211418104045**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

#### COMPUTER SCIENCE AND ENGINEERING

#### 

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

#### MAY 2022

**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

#### Certified that this project report “SENTIMENTAL PREDICTIONS ON REVIEW USING MULTIPLE ALGORITHMS” is the bonafide work of “ AARTHI.R(211418104003), DEEPIKA.M(211418104045)” who carried out the project work under “ DR.JOSEPHINE LEELA” supervision.

**SIGNATURE SIGNATURE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dr.S.MURUGAVALLI,M.E.,Ph.D**  **HEAD OF THE DEPARTMENT** | **.,** | **Dr.R.JOSEPHINE LEELA, SUPERVISOR** | **M.E .,Ph.D.** |  |
|  |  | **PROFESSOR** |  |  |

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE PANIMALAR ENGINEERING COLLEGE NASARATHPETTAI, NASARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above students were examined in the End Semester project viva-voice held on 25/05/2022.

#### INTERNAL EXAMINER EXTERNAL EXAMINER

**DECLARATION BY THE STUDENT**

We **AARTHI.R (211418104003) AND DEEPIKA.M (211418104045)**

hereby declare that this project report titled “**SENTIMENTAL PREDICTIONS ON REVIEW USING MULTIPLE ALGORITHMS”**, under the guidance of **Dr. R. JOSEPHINE LEELA M.E .,Ph.D.**, is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

1. **AARTHI.R**

**2. DEEPIKA.M**

**ACKNOWLEDGEMENT**

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI**

**KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **DR.R.JOSEPHINE LEELA, M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

**AARTHI.R [211418104003]**

**DEEPIKA.M [211418104045]**

#### ABSTRACT

Numerous consumer reviews of products are now available on the Internet. Consumer reviews contain rich and valuable knowledge for both firms and users. However, the reviews are often disorganized, leading to difficulties in information navigation and knowledge acquisition. This article proposes a product aspect ranking framework, which automatically identifies the important aspects of products from online consumer reviews, aiming at improving the usability of the numerous reviews. The important product aspects are identified based on two observations the important aspects are usually commented on by a large number of consumers and consumer opinions on the important aspects greatly influence their overall opinions on the product. In particular, given the consumer reviews of a product, we first identify product aspects by a shallow dependency parser and determine consumer opinions on these aspects via a sentiment classifier. We then develop a probabilistic aspect ranking algorithm to infer the importance of aspects by simultaneously considering aspect frequency and the influence of consumer opinions given to each aspect over their overall opinions. The experimental results on a review corpus of 21 popular products in eight domains demonstrate the effectiveness of the proposed approach.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | v |
|  | **LIST OF TABLES** | vi |
|  | **LIST OF FIGURES** | ix |
|  | **LIST OF SYMBOLS, ABBREVIATIONS** | ix |
| **1.** | **INTRODUCTION** |  |
|  | 1.1 Overview | 1 |
|  | 1.2 Problem Definition | 2 |
| **2.** | **LITERATURE SURVEY** | 3 |
| **3.** | **SYSTEM ANALYSIS** |  |
|  | 3.1 Existing System | 9 |
|  | 3.2 Proposed system | 10 |
|  | 3.3 Feasibility Study | 11 |
|  | 3.4 Hardware Environment | 12 |
|  | 3.5 Software Environment | 12 |
| **4.** | **SYSTEM DESIGN** |  |
|  | 4.1. ER diagram | 13 |
|  | 4.2 Data Flow Diagram | 13 |
|  | 4.3 UML Diagrams | 16 |
| **5.** | **SYSTEM ARCHITECTURE** |  |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | 5.1 System Architecture  5.2 Module Design Specification | 23    23 |
|  | 5.3 Algorithms | 28 |
| **6.** | **SYSTEM IMPLEMENTATION**  6.1 Sentimental Predictions on review using multiple algorithms | 29 |
| **7.** | **PERFORMANCE ANALYSIS** |  |
|  | 7.1 Test Cases & Reports | 45 |
| **8.** | **CONCLUSION** |  |
|  | 8.1 Results & Discussion | 48 |
|  | 8.2 Conclusion and Future Enhancements | 49 |
|  | **APPENDICES** |  |
|  | A.1 Sample Screens | 50 |
|  | **REFERENCES** | |  | | --- | | 52 | |
|  |  |  |

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **NAME OF THE TABLES** | **Page no** |
| 7.1.1 | Predictions rating for new review | 45 |
| 7.1.2 | Analyze each model | 46 |
|  |  |  |
| 7.1.3 | Analyze each model (precision, recall, f-measure) | 47 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **Page no** |
| 4.1 | Entity Relationship diagram | 13 |
| 4.2 | Data flow Diagram level-0 | 14 |
| 4.3 | Data flow Diagram level-1 | 14 |
| 4.4 | Data flow Diagram level-2 | 14 |
| 4.5 | Data flow Diagram level-3 | 15 |
| 4.6 | Overall Data flow Diagram | 15 |
| 4.7 | Use Case Diagram | 16 |
| 4.8 | Class Diagram | 17 |
| 4.9 | Activity diagram | 18 |
| 4.10 | Sequence diagram | 19 |
| 4.11 | Deployment diagram | 20 |
| 4.12 | Collaboration diagram | 21 |
| 4.13 | Component diagram | 22 |
| 5.1 | System Architecture Diagram | 23 |
| A.1 | Analyzing between models | 50 |
| A.2 | Finding the accuracy using ROC curve | 50 |
| A.4 | Comparing between the models and their accuracy | 51 |

**LIST OF ABBREVATIONS**

1. **KNN-** K-Nearest Neighbor

#### SVM - Support Vector Machine

1. **SVD** - singular value decomposition
2. **RMSE** - Root Mean Square Error

**5. MAE** - Mean Absolute Error

**CHAPTER-1**

**INTRODUCTION**

#### 1.1 OVERVIEW

Sentiment Analysis is a widely used text classification technique. It breaks down any given text or comments and classify the text either as positive or negative based on the views conveyed in it. Previous works done on sentiment classification used either lexicon based approach or machine learning techniques. the major drawback of the existing systems was the focus on only binary classification of review such as positive or negative. Ignorance of the neutral review will result in misinterpretation of a customer’s opinion about a product or movie, which will degrade the business or trend. In case of using only approach, the system highly depends on the selection of lexicon resource and dictionary. system built only using machine learning approach, the performance of the system depends on the algorithms chosen. This work presents a hybrid model to resolve the neutral class too. The proposed work combines a lexical approach (SentiWordNet) with the machine learning algorithms such as Support Vector Machine, Decision Tree, Logistic Regression and Naive Bayes for sentiment analysis to resolve the neutral opinions beyond the binary categorization of the customer’s review. We have also compared the performance of these four machine learning algorithms along with the approach. The results proved that Support Vector Machine and Logistic Regression algorithms outperform the other two algorithms with an accuracy of about 80% which is on average differs by 6% to 10% when compared to other algorithms.

#### PROBLEM DEFINITION

This corpus is publicly available by request. Experimental results have demonstrated the effectiveness of the proposed approaches. Moreover, we applied product aspect ranking to facilitate two real-world applications. Document-level sentiment classification and extractive review summarization. Significant performance improvements have been obtained with the help of product aspect ranking. Consumers commonly seek quality information from online reviews prior to purchasing a product, while many firms use online reviews as important feed backs in their product development, marketing and consumer relationship management. Bing Shopping has indexed more than five million products. Amazon.com archives a total of more than 36 million products. Shopper.com records more than five million products from over 3,000 merchants. Most retail Websites encourage consumers to write reviews to express their opinions on various aspectsof the products. Here an aspect also called feature in literatures, refers to a component or an attribute of a certain product. Moreover, we applied product aspect ranking to facilitate two real-world applications. document-level sentiment classification and extractive review summarization. Significant performance improvements have been obtained with the help of product aspect ranking.

#### CHAPTER 2

#### LITERATURE SURVEY

#### 2.1 Identification of Malicious Injection Attacks in Dense Rating and Co-visitation Behaviors, Zhihai Yang, Qindong Sun, Yaling Zhang, and Wei Wang, IEEE 2021

Divide-and-conquer strategy is used to detect profile injection attacks and co- visitation injection attacks for online recommender systems The detection performance of IMIA-HCRF can achieve an improvement of 7.8% for mixed profile injection attacks as well as 6% for mixed co-visitation injection attacks while keeping the highest detection rate. It cannot eliminate the disturbed data .

#### Fake review detection, Rami Mohawesh , Shuxiang Xu , Son N. Tran , Robert Ollington , Matthew Springer , Yaser Jararweh , Sumbal Maqsood ,IEEE 2021



It proposes a machine learning approach to identify fake reviews .It compares the performance of several experiments done on a real Yelp dataset of restaurants reviews with andwithout features extracted from users behaviors. The results reveal that KNN(K=7) outperforms the rest of classifiers in terms of f-score achieving best f-score 82.40%. The results show that the f-score has increased by 3.80% when taking the extracted reviewers behavioral features into consideration. This dataset is that it is equally balanced between positive and negative cases. Results indicate that current text generation methods yield fake reviews that appear so realistic that it is challenging for a human to detect them.

#### Fake review detection on online E-commerce platforms: a systematic literature Review, Himangshu Paul and Alexander Nikolaev, IEEE 2021

In addition, this study has identified different performance metrics that are commonly used to evaluate the accuracy of the review spam detection models. The first comprehensive review of existing studies in the domain of review detection using SLR process. Supervised machine learning methods mostly used AUC evaluation measures. There is a need for in-depth research on the detection of fake review in multilingual reviews, such feedback or comments on given reviews have not been exploited as features for detection of fake reviews.

#### A Comparative study to recognize fake ratings in recommendation system using classification techniques, Sundari.P.Shanmuga, Subaji.M, IJERCSE 2021

Focusing on Collaborative Filtering, Content-Based Filtering and Hybrid recommendation system by using the well-known MovieLens dataset. It proposed approach is compared with well-known machine learning approaches namely k nearest neighbor (K-NN), singular value decomposition (SVD) and Co-clustering. It verified using MovieLens 100 K datasets and error of the RS is measured using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The result shows that the proposed approach gives a lesser error rate with RMSE (0.9201) and MAE (0.7219). Accuracy alone is not suﬃcient for the selection of related algorithm. The users like diverse recommendations as compared to more accurate recommendations.

#### 2.5 Creating and detecting fake reviews of online products,Joni Saliminen, Chandhrasekhar Kandpal, Ahmed Mohamed kamel,Soon-gyo Jung, Bernard J. Jansen, 2021

Creating and detecting the fake reviews by using ULMFit and GPT-2.Fake reviews are created by based on amazon’s e-commerce datset. GPT-2 model classifies the fake reviews. It achieves an accuracy of 97%. When the review is descriptive its easy to classify and take decisions accordingly.When the product review is short it is difficult to determine

#### 2.6 Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics, Anindya Ghose, Panagiotis G. Ipeirotis, 2021

With the rapid growth of the Internet, the ability of users to create and publish content has created active electronic communities that provide a wealth of product information. However, the high volume of reviews that are typically published for a single product makes harder for individuals as well as manufacturers to locate the best reviews and understand the true underlying quality of a product.  The impact of reviews on economic outcomes like product sales and see how different factors affect social outcomes such as their perceived usefulness. Our approach explores multiple aspects of review text, such as subjectivity levels, various measures of readability and extent of spelling errors to identify important text-based features. Multiple reviewer-level features such as average usefulness of past reviews and the self-disclosed identity measures of reviewers that are displayed next to a review.

#### 2.7 Document-Word Co-regularization for Semi-supervised Sentiment Analysis, Vikas Sindhwani and Prem Melville,2020

The goal of sentiment prediction is to automatically identify whether a given piece of text expresses positive or negative opinion towards a topic of interest.  sentiment prediction as a standard text categorization problem, but gathering labeled data turns out to be a bottleneck. Moreover, in many applications abundant unlabeled data is also available. In this paper, we propose a novel semi-supervised sentiment prediction algorithm that utilizes lexical prior knowledge in conjunction with unlabeled examples. We present an empirical study on a diverse collection of sentiment prediction problems which confirms that our semi-supervised lexical models significantly outperform purely.

#### 2.8 Hot item mining and summarization from multiple auction Web site Tak-lam, Wai lam,2020

Develop a two-phase framework which aims at mining and summarizing hot items from multiple auction Web sites to assist decision making. The objective of the first phase is to automatically extract the product features and product feature values of the items from the descriptions provided by the sellers. We design a HMM- based learning method to train an extended HMM model which can adapt to the unseen Web page from which the information is extracted. The summary of the hot items is then generated by considering the frequency and the position of the product features being mentioned in the descriptions. We have conducted extensive experiments from several real-world auction Web sites to demonstrate the effectiveness of our framework.

**2.9 Modeling and Predicting the Helpfulness of Online Reviews, Yang Liu1, Xiangji Huang2, Aijun An1 and Xiaohui Yu2,2019**

Online reviews provide a valuable resource for potential customers to make purchase decisions. However, the sheer volume of available reviews as well as the large variations in the review quality present a big impediment to the effective use of the reviews, as the most helpful reviews may be buried in the large amount of low quality reviews. The goal of this paper is to develop models and algorithms for predicting the helpfulness of reviews.

* 1. **A survey on opinion mining and sentiment analysis Tasks, approaches and applications, Kumar Ravi,Vadlamani Ravi in Knowledge-Based Systems,2019.**

 Evolution of social media has also contributed immensely to these activities, thereby providing us a transparent platform to share views across the world. These electronic Word of Mouth (eWOM) statements expressed on the web are much prevalent in business and service industry to enable customer to share his/her point of view. In the last one and half decades, research communities, academia, public and service industries are working rigorously on sentiment analysis, also known as, opinion mining, to extract and analyze public mood and views. In this regard, this paper presents a rigorous survey on sentiment analysis, which portrays views presented by over one hundred articles published in the last decade regarding necessary tasks, approaches, and applications of sentiment analysis. Several sub-tasks need to be performed for sentiment analysis which in turn can be accomplished using various approaches and techniques.

# 

# CHAPTER 3

# SYSTEM ANALYSIS

#### EXISTING SYSTEM

Most retail Websites encourage consumers to write reviews to express their opinions on various aspects of the products. Here an aspect also called feature in literatures refers to a component or an attribute of a certain product A sample review The battery life of Nokia N95 is amazing. reveals positive opinion on the aspect battery life of product Nokia N95.Besides the retail Websites, any forum Websites also provide a platform for consumers to post reviews on millions of products. For example,CNet.com involves more than seven million product reviews; whereas Price grabber.com contains millions of reviews on more than 32 million products in 20 distinct categories over 11,000 merchants. Such numerous consumer reviews contain rich and valuable  knowledge and have become an important resource for both consumers and firms. Consumers commonly seek quality information from online reviews prior to purchasing a product, while many firms use online reviews as important feed backs in their product development, marketing and consumer relationship management. We argue that some aspects are more important than the others, and have greater impact on the eventual consumers decision making as well as firms product development strategies. Some aspects of iPhone usability and “battery are concerned by most consumer and are more important than the others such as USB and button.

#### PROPOSED SYSTEM

Propose a product aspect ranking framework to automatically identify the important aspects of products from online consumer reviews. Our assumption is that the important aspects of a product possess the following characteristics they are frequently commented in consumer reviews; and consumers opinions on these aspects greatly influence their overall opinions on the product. A straightforward frequency-based solution is to regard the aspects that are frequently commented in consumer reviews as important. However, consumers opinions on the frequent aspects may not influence their overall opinions on the product, and would not influence their purchasing decisions.This method simply assumes that an overall rating was derived from the specific opinions on different aspects individually, and cannot precisely characterize the correlation between the specific opinions and the overall rating. Hence, we go beyond these methods and propose an effective aspect ranking approach to infer the importance of product aspects. on specific aspects over their overall ratings on the product is to count the cases where their opinions on specific aspects and their overall ratings are consistent, and then ranks the aspects according to the number of the consistent cases. Most consumers frequently criticize the bad “signal connection" of iPhone 4, but they may still give high overall ratings to iPhone 4. Therefore, the frequency-based solution is not able to identify the truly important aspects.

#### FEASIBILITY STUDY TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system. This project is build on jupyter notebook in which we use python as programming language because it is dynamic and has vast library support.

#### SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

#### ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased

#### HARDWARE ENVIRONMENT

* + - Processor - i3, i5, i7
    - Speed - 2.3 GHz
    - RAM - 4 Gb
    - Hard Disk - 260 GB

#### SOFTWARE ENVIRONMENT

* + - Operating System - Windows 7/8/10
    - Front End - Html, Css.
    - Back End - MySQL.
    - Scripts - Java.

# CHAPTER 4

# SYSTEM DESIGN

#### ENTITY RELATIONSHIP DIAGRAM

An **entity relationship diagram** (**ERD**) shows the relationships of entity sets stored in a database. By defining the entities, their attributes, and showing the relationships between them, an ER Diagram illustrates the logical structure of databases.ER Diagrams are used to sketch out the design of a database.

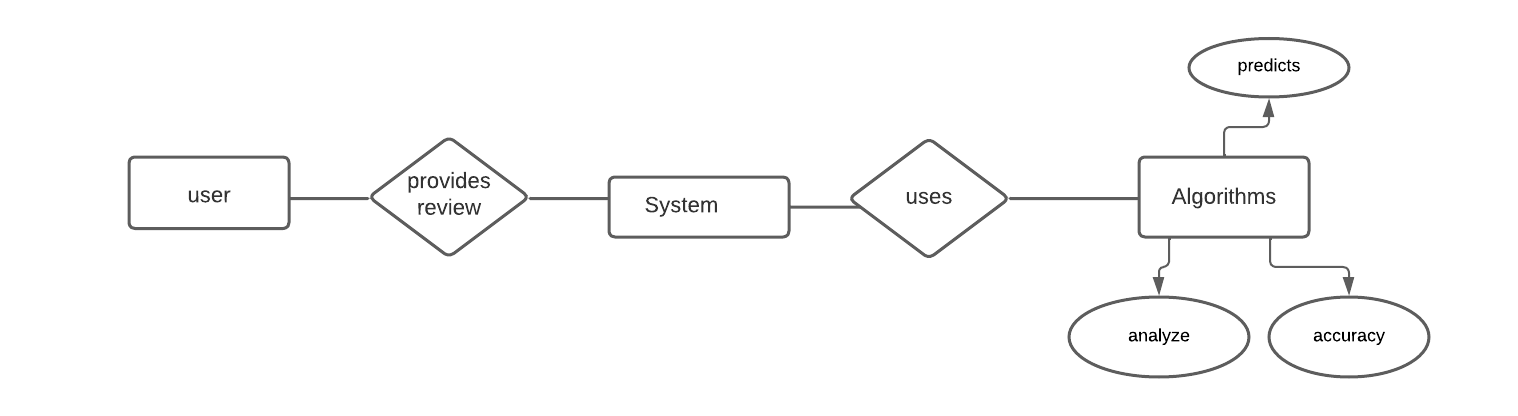


Figure 4.1 – Entity Relationship Diagram of Sentimental prediction

**4.2 DATA FLOW DIAGRAM**

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business.



Figure 4.2 -Data flow diagram level-0

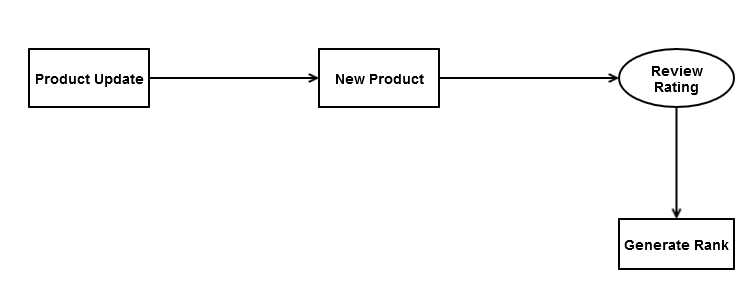


Figure 4.3- Data flow diagram level-1

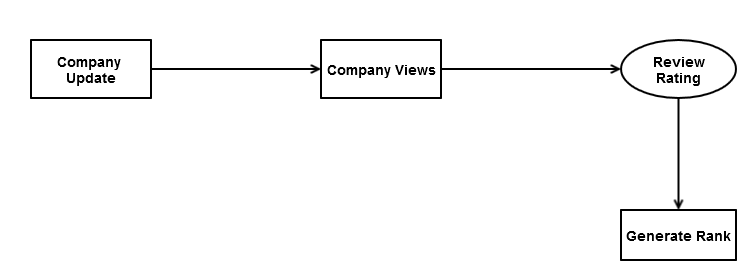


Figure 4.4 - Data flow diagram level-2

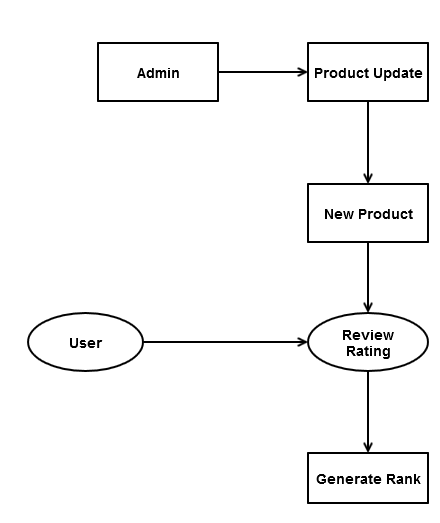


Figure 4.5- Data flow diagram level-3

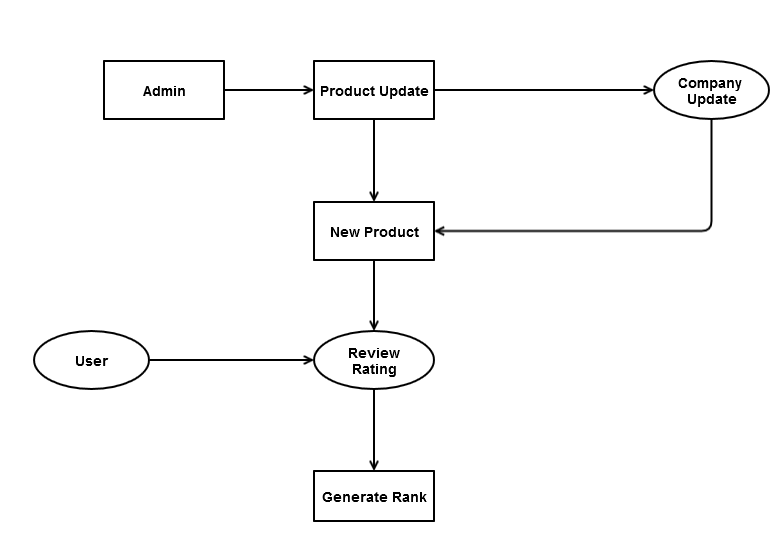


Figure 4.6 – Overall Data flow diagram

#### 4.3UML DIAGRAMS

* + 1. **USE CASE DIAGRAM**

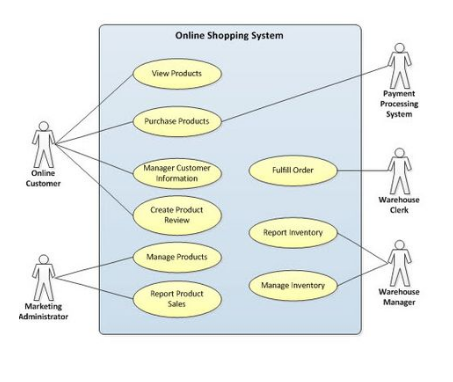


Figure 4.7 – Use Case Diagram of Sentimental predictions.

* + 1. **CLASS DIAGRAM**

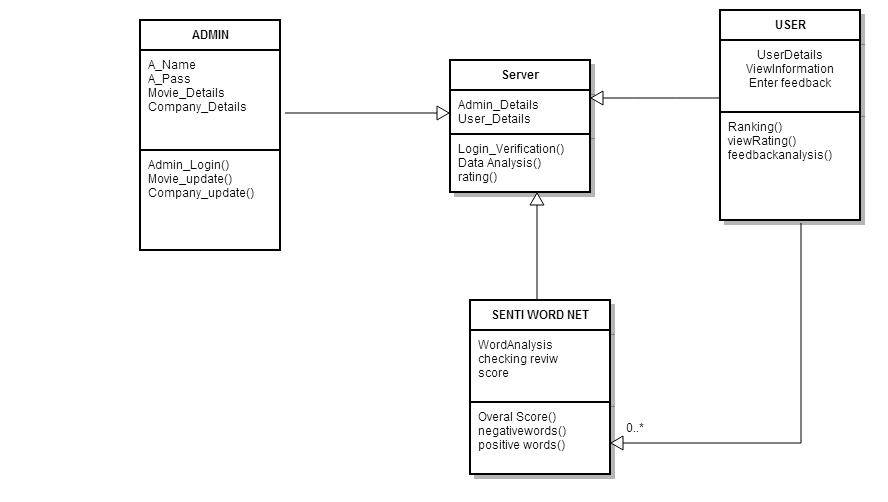
****

Figure 4.8 – Class Diagram of Sentimental prediction

**4.3.3 ACTIVITY DIAGRAM**

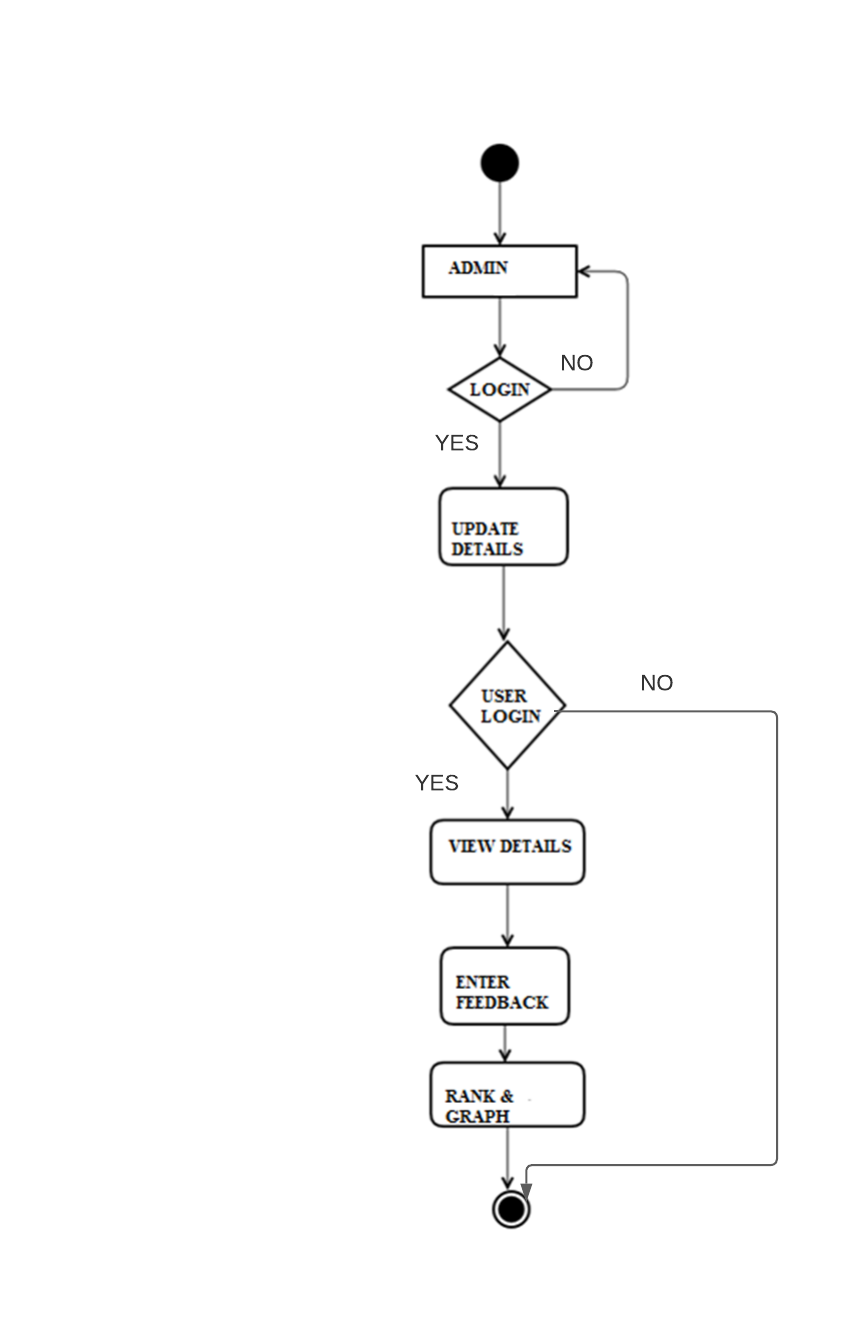


Figure 4.9 – Activity Diagram of Sentimental predictions

* + 1. **SEQUENCE DIAGRAM**

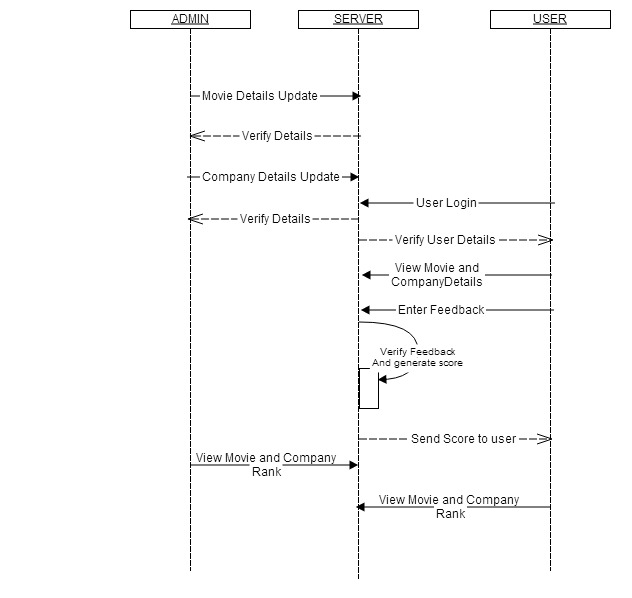


Figure 4.10– Sequence Diagram of Sentimental Predictions

**4.3.5 DEPLOYMENT DIAGRAM**

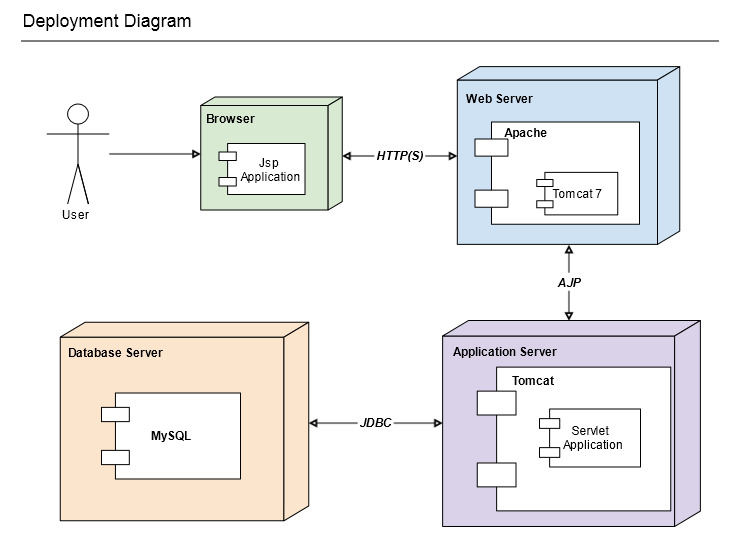


Figure 4.11 – Deployment Diagram of Sentimental Predictions

**4.3.6 COLLABORATION DIAGRAM**

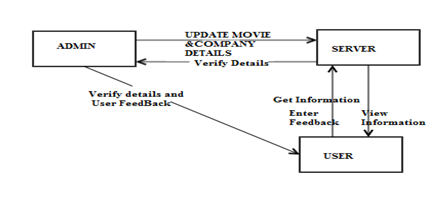
****

Figure 4.12– Collaboration Diagram of Sentimental Predictions

* + 1. **COMPONENT DIAGRAM**

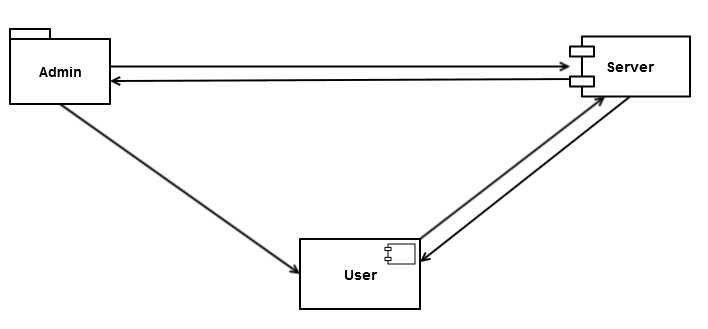
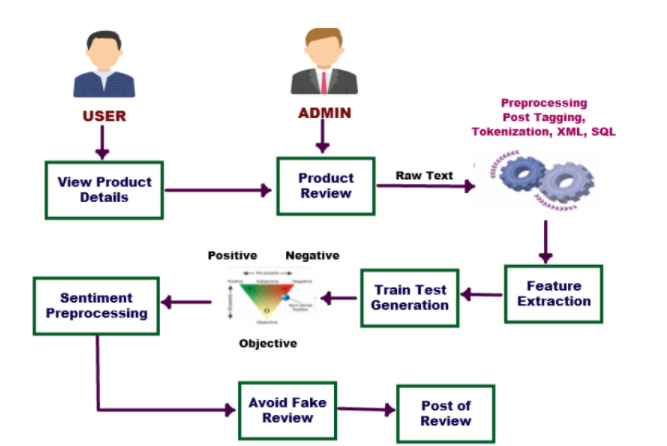


Figure 4.13 – Component Diagram of Sentimental Predictions

# CHAPTER 5

**SYSTEM ARCHITECTURE**

#### 5.1 ARCHITECTURE DIAGRAM



**Figure 5.1 - System Architecture Diagram**

This architecture diagram depicts the work flow, it starts with the pre-processing step and by extracting the features training the models are done, through sentimental processing the reviews are classified.

**5.2 MODULE DESIGN SPECIFICATION**

* Product aspect identification
* Product aspect ranking
* Extractive review
* Sentimental classification
* Consumer review

#### MODULE DESCRIPTION

**5.2.1 Product aspect identification**

* The Websites such as CNet.com require consumers to give an overall rating on the product, describe concise positive and negative opinions on some product aspects, as well as write a paragraph of detailed review in free text.
* Websites Viewpoints.com only ask for an overall rating and a paragraph of free-text review.
* Websites Reevoo.com just require an overall rating and some concise positive and negative opinions on certain aspects.
* An overall rating, a consumer review consists of Pros and Cons reviews, free text review, or both.
* That aspects are usually nouns or noun phrases and we can obtain highly accurate aspects by extracting frequent noun terms from the Pros and Cons reviews.
* For identifying aspects in the free text reviews a straightforward solution is to employ an existing aspect identification approach.
  + 1. **Product aspect Ranking**
* Overview of its pipeline consisting of three main components aspect identification sentiment classification on aspects probabilistic aspect ranking.
* we first identify the aspects in the reviews and then analyze consumer opinions on the aspects via a sentiment classifier.
* Finally, we propose a probabilistic aspect ranking algorithm to infer the importance of the aspects by simultaneously taking into account aspect frequency and the influence of consumers opinions given to each aspect over their overall opinions.
* Finally assigns an overall rating is a numerical score that indicates different levels of overall opinion in the review are the minimum and maximum ratings respectively
  + 1. **Extractive Review**
* A particular product there is an abundance of consumer reviews available on the internet.
* It is impractical for user to grasp the overview of consumer reviews and opinions on various aspects of a product from such enormous reviews.
* Hence there is a compelling need for automatic review summarization which aims to condense the source reviews into a shorter version preserving its information content and overall meaning.
* Existing review summarization methods can be classified into abstractive and extractive summarization.
* An abstractive summarization attempts to develop an understanding of the main topics in the source reviews and then express those topics in clear natural language.

#### Sentimental classification

#### The task of analyzing the sentiments expressed on aspects is called aspect-level sentiment classification in literature.

#### Exiting techniques include the supervised learning approaches and the lexicon-based approaches which are typically unsupervised The lexicon-based methods utilize a sentiment lexicon consisting of a list of sentiment words phrases and idioms to determine the sentiment orientation on each aspect

#### The classifier is then used to predict the sentiment on each aspect.Many learning-based classification models are applicable for example Support Vector Machine (SVM),Naive Bayes,and Maximum Entropy (ME) model Supervised learning is dependent on the training data and cannot perform well without sufficient training samples.

#### However,labeling training data is laborintensive and time-consuming.In this work,the Pros and Cons reviews have explicitly categorized positive and negative opinions on the aspects.

**5.2.5 Consumer Review**

* The opinions on different aspects might be in contrast to each other and have different degree of impacts on the overall opinion of the review document a sample review document of iPhone 4.
* It expresses positive opinions on some aspects such as reliability easy to use and simultaneously criticizes some other aspects such as touch screen,quirk,music play.
* Finally,it assigns an high overall rating (positive opinion) on iPhone 4 due to that the important aspects are with positive opinions.
* Hence,identifying important aspects can naturally facilitate the estimation of the overall opinions on review documents.This observation motivates us to utilize the aspect ranking results to assist document-level sentiment Classification.
* Specifically,we randomly sampled 100 reviews of each product as testing samples and used the remaining reviews for training.

#### ALGORITHM

**5.3.1 PROBABILISTIC ASPECT RANKING ALGORITHM**

Customer review corpus R, each review r Ɛ R is associated with an overall rating Or , and a vector of opinions or on specific aspects.

Output: Importance scores wr |k=1 m for all the m aspects.

While

not converged do Update { }r=1 to |R| according to ŵr = (oror T /σ2 + Σ-1 ) -1 (Or or /σ2 + Σ-1µ) Update {µ, Σ, σ2 } according to σˆ2 =1/|R|Σr∈R (Or − ω T r or)2.

End while Compute aspect importance scores {ῶk}k=1

**Input**: Consumer review corpus R, each review r ε R is associated with an overall rating Or and a vector of opinions Or on specific aspects.

**Output**: Importance scores m k k 1 ϖ| = for all m aspects.

while not converged

do Update | | 1 { } =r rωR

Update { , , } 2 σ ∑ μ

End While Compute aspect importance scores m k k 1 ϖ| =

# CHAPTER 6

**SYSTEM IMPLEMENTATION**

#### 6.1 Sentimental Predictions on review using multiple algorthim.ipynb

# 6.1.1 Import Libraries

# import pandas as pd

# import numpy as np

# import matplotlib.pyplot as plt

# import seaborn as sns

# %matplotlib inline

# %config InlineBackend.figure\_format = 'retina'

# import warnings

# warnings.filterwarnings('ignore')

# 6.1.2 Import Data

amazon\_data **=** pd.read\_csv("C:/Users/Aarthi/amazon review (1)/amazon review/Amazon\_Unlocked\_Mobile.csv")

amazon\_data **=** amazon\_data.dropna(axis **=** 0)

amazon\_data.shape

| **Product Name** | **Brand Name** | **Price** | **Rating** | **Reviews** | **Review Votes** |
| --- | --- | --- | --- | --- | --- |
| "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 5 | I feel so LUCKY to have found this used (phone... | 1.0 |
| "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 4 | nice phone, nice up grade from my pantach revu... | 0.0 |
| "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 5 | Very pleased | 0.0 |
| "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 4 | It works good but it goes slow sometimes but i... | 0.0 |
| "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 4 | Great phone to replace my lost phone. The only... | 0.0 |

amazon\_data**=**amazon\_data[["Reviews","Rating"]]

amazon\_data.head()

| **Reviews** | **Rating** |
| --- | --- |
| I feel so LUCKY to have found this used (phone... | 5 |
| nice phone, nice up grade from my pantach revu... | 4 |
| Very pleased | 5 |
| It works good but it goes slow sometimes but i... | 4 |
| Great phone to replace my lost phone. The only... | 4 |

amazon\_data\_pos**=**amazon\_data[amazon\_data["Rating"].isin([4,5])]

amazon\_data\_pos.head()

| **Reviews** | **Rating** |
| --- | --- |
| I feel so LUCKY to have found this used (phone... | 5 |
| nice phone, nice up grade from my pantach revu... | 4 |
| Very pleased | 5 |
| It works good but it goes slow sometimes but i... | 4 |
| Great phone to replace my lost phone. The only... | 4 |

|  | **Reviews** | **Rating** |
| --- | --- | --- |
| **413830** | LOVE IT | 5 |
| **413832** | good rugged phone that has a long-lasting batt... | 4 |
| **413835** | another great deal great price | 5 |
| **413837** | Passes every drop test onto porcelain tile! | 5 |
| **413839** | Only downside is that apparently Verizon no lo... | 4 |
|  |  |  |

amazon\_data\_neg**=**amazon\_data[amazon\_data["Rating"].isin([1,2])]

amazon\_data\_neg.head()

amazon\_data\_neg.tail()

amazon\_data.Rating.value\_counts()

sns.barplot(x**=**amazon\_data.Rating.value\_counts().index,y**=**amazon\_data.Rating.value\_counts().values)

amazon\_data\_filtered**=**pd.concat([amazon\_data\_pos[:20000],amazon\_data\_neg[:20000]])

amazon\_data\_filtered.shape

sns.barplot(x**=**amazon\_data\_filtered.Rating.value\_counts().index,y**=**amazon\_data\_filtered.Rating.value\_counts().values)

amazon\_data\_filtered["r"]**=**1

amazon\_data\_filtered["r"][amazon\_data\_filtered["Rating"].isin([1,2])]**=** 0

amazon\_data\_filtered.head()

amazon\_data\_filtered.tail()

amazon\_data\_filtered.r.value\_counts()

**6.1.3 Spilt Train And Test data**

**from** sklearn.model\_selection **import** train\_test\_split

X\_train\_data,x\_test\_data,Y\_train\_data,y\_test\_data**=**train\_test\_split(amazon\_data\_filtered["Reviews"],amazon\_data\_filtered["r"],test\_size**=**0.2)Y\_train\_data.head()

X\_train\_data.head()

**6.1.4Text Transformation**

**from** sklearn.feature\_extraction.text **import** CountVectorizer,TfidfVectorizer

tfidf\_vector **=** TfidfVectorizer(stop\_words**=**"english")

tfidf\_vector.fit(X\_train\_data)

print(tfidf\_vector.get\_feature\_names()[0:20])

print(tfidf\_vector.get\_feature\_names()[**-**20:])

['00', '000', '0000', '000mah', '002', '00emotional', '00now', '00pm', '00so', '00time', '01', '013435003182980', '018633051660f', '02', '03', '04', '04th', '06', '07', '0780']

['zentalk', 'zenui', 'zero', 'zf2', 'zillion', 'zip', 'ziploc', 'ziplock', 'zippy', 'zmax', 'zone', 'zones', 'zoom', 'zoomed', 'zte', 'ítem', 'óptico', 'ünlocked', 'ýn', 'śo']

X\_train\_data\_new**=**tfidf\_vector.transform(X\_train\_data)

X\_train\_data\_new.shape

x\_test\_data\_new**=**tfidf\_vector.transform(x\_test\_data)

**6.1.5 Logistic Regression**

predictions **=** dict()

**from** sklearn.linear\_model **import** LogisticRegression

lr\_model **=** LogisticRegression()

lr\_model.fit(X\_train\_data\_new,Y\_train\_data)

predictions["LogisticRegression"] **=** lr\_model.predict(x\_test\_data\_new)

**from** sklearn.metrics

**import** classification\_report,accuracy\_score,confusion\_matrix

**6.1.6 SVM**

**from** sklearn.svm **import** SVC

svm\_model **=** SVC()

svm\_model.fit(X\_train\_data\_new,Y\_train\_data)

predictions["SVM"]**=**svm\_model.predict(x\_test\_data\_new)

accuracy\_score(y\_test\_data,predictions["SVM"])

**6.1.7 Naive Bayers**

**from** sklearn.naive\_bayes **import** MultinomialNB

mul\_model **=** MultinomialNB()

mul\_model.fit(X\_train\_data\_new,Y\_train\_data)

predictions["Multinomial"] **=** mul\_model.predict(x\_test\_data\_new)

accuracy\_score(y\_test\_data, predictions["Multinomial"])

*# 2)BernoulliNB*

**from** sklearn.naive\_bayes **import** BernoulliNB

ber\_model **=** BernoulliNB()

ber\_model.fit(X\_train\_data\_new,Y\_train\_data)

predictions["BernoulliNB"]**=**ber\_model.predict(x\_test\_data\_new)

accuracy\_score(y\_test\_data,predictions["BernoulliNB"])

**6.1.7 k -NN classifier**

**from** sklearn.neighbors **import** KNeighborsClassifier

**from** sklearn.neighbors **import** KNeighborsClassifier

knn\_model **=** KNeighborsClassifier(n\_neighbors**=**5)

knn\_model.fit(X\_train\_data\_new,Y\_train\_data)

predictions["knn"] **=** knn\_model.predict(x\_test\_data\_new)

accuracy\_score(y\_test\_data,predictions["knn"])

**6.1.8 Ensemble classifier**

**from** sklearn.ensemble **import** RandomForestClassifier

ess\_model **=** RandomForestClassifier()

ess\_model.fit(X\_train\_data\_new,Y\_train\_data)

predictions["EssembleClasification"] **=** ess\_model.predict(x\_test\_data\_new)

predictions["EssembleClasification"]

**from** sklearn.metrics **import** roc\_curve, auc

**from** sklearn.metrics **import** roc\_curve, auc

false\_positive, true\_positive,\_**=** roc\_curve(y\_test\_data,predictions["EssembleClasification"])

plt.title('Receiver Operating Characteristic')

plt.plot(false\_positive, true\_positive)

​

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

accuracy\_score(y\_test\_data,predictions["EssembleClasification"])

**6.1.8 Decision Tree**

**from** sklearn **import** tree

​

tree\_model **=** tree.DecisionTreeClassifier()

tree\_model.fit(X\_train\_data\_new,Y\_train\_data)

predictions["DecisionTree"] **=** tree\_model.predict(x\_test\_data\_new)

accuracy\_score(y\_test\_data,predictions["DecisionTree"])

**6.1.9 Evaluation**

print(classification\_report(y\_test\_data, predictions['EssembleClasification'], target\_names **=** ["Good", "Bad"]))

matrix **=** confusion\_matrix(y\_test\_data, predictions['EssembleClasification'])

matrix\_normalized **=** matrix.astype('float') **/** matrix.sum(axis**=**1)[:, np.newaxis]

sns.heatmap(matrix\_normalized)

plt.ylabel('Actual')

plt.xlabel('Predicted')

print\_results **=** {}

**for** k,v **in** predictions.items():

print\_results[k] **=** accuracy\_score(y\_test\_data,v)

print\_results

result\_table**=**pd.DataFrame(list(print\_results.items()), columns**=**["Model","Accuracy"])

result\_table

**6.2 Sentimental analysis of customer reviews system using TFIDF\_Technique.ipynb**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

%config InlineBackend.figure\_format = 'retina'

import warnings

warnings.filterwarnings('ignore')

from ipywidgets import \*

from IPython.display import display

from ipywidgets import FloatProgress

import time

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

from sklearn.metrics import roc\_curve, auc

from sklearn.feature\_extraction.text import CountVectorizer,TfidfVectorizer

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

from sklearn.svm import SVC

from sklearn.naive\_bayes import MultinomialNB

from sklearn.naive\_bayes import BernoulliNB

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn import tree

from sklearn.linear\_model import LogisticRegression

#Data cleaning and preprocessing

amazon\_data = pd.read\_csv("C:/Users/Aarthi/amazon review (1)/amazon review/Amazon\_Unlocked\_Mobile.csv")

amazon\_data = amazon\_data.dropna(axis = 0)

amazon\_data=amazon\_data[["Reviews","Rating"]]

amazon\_data\_pos=amazon\_data[amazon\_data["Rating"].isin([4,5])]

amazon\_data\_neg=amazon\_data[amazon\_data["Rating"].isin([1,2])]

amazon\_data\_filtered=pd.concat([amazon\_data\_pos[:20000],amazon\_data\_neg[:20000]])

amazon\_data\_filtered["r"]=1

amazon\_data\_filtered["r"][amazon\_data\_filtered["Rating"].isin([1,2])]= 0

#Splitting Train and Test Data

X\_train\_data,x\_test\_data,Y\_train\_data,y\_test\_data=train\_test\_split(amazon\_data\_filtered["Reviews"],amazon\_data\_filtered["r"],test\_size=0.2)

#Text Transformation using TFIDF

tfidf\_vector = TfidfVectorizer(stop\_words="english")

tfidf\_vector.fit(X\_train\_data)

X\_train\_data\_new=tfidf\_vector.transform(X\_train\_data)

x\_test\_data\_new=tfidf\_vector.transform(x\_test\_data)

predictions = dict()

#Widgets

start\_button=Button(description="Start System",button\_style='danger')

drop\_down\_choice=widgets.Dropdown(options={'Analyze each model':1,'Predict rating for new review':2,'Compare Models':3},value=2)

choose\_button=Button(description="Click",button\_style='danger')

drop\_down=widgets.Dropdown(options={'Logistic Regression': 1, 'SVM': 2, 'Multinomial Naive Bayes': 3, 'Bernoulli Naive Bayes':4,'k-NN':5, 'Ensemble':6,'Decision Tree':7},value=1)

rad\_button=widgets.RadioButtons(options=['Accuracy', 'ROC Curve', 'Precision,Recall and F-Measure',],value='Accuracy',disabled=False)

button= widgets.Button(description="Submit",button\_style='info')

Inp\_text = widgets.Text(description="", width=20000)

button\_rating= widgets.Button(description="Find Rating",button\_style='info')

fp = FloatProgress(min=0,max=100,description="Calculating")

Out\_text=widgets.Text()

#1. SVM Classifier

def svm():

print("\n\nSVM Classifier")

print("Please be patient. This may take some time.")

svm\_model = SVC()

#Train Model

svm\_model.fit(X\_train\_data\_new,Y\_train\_data)

#Test Model

predictions['SVM']=svm\_model.predict(x\_test\_data\_new)

#Calculating Model Accuracy

if rad\_button.value=='Accuracy':

print("Accuracy = "+str(accuracy\_score(y\_test\_data,predictions['SVM'])))

#Generating ROC Curve

if rad\_button.value=='ROC Curve':

false\_positive, true\_positive,\_= roc\_curve(y\_test\_data,predictions['SVM'])

plt.title('Receiver Operating Characteristic')

plt.plot(false\_positive, true\_positive)

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

#Calculating Precision, Recall and F-Measure

if rad\_button.value=='Precision,Recall and F-Measure':

print(classification\_report(y\_test\_data, predictions['SVM'], target\_names = ["Good", "Bad"]))

#2. Multinomial Naive Bayes Classifier

def multinomial():

print("\n\nMultinomial Naive Bayes Classifier")

print("Please be patient. This may take some time.")

mul\_model = MultinomialNB()

#Train Model

mul\_model.fit(X\_train\_data\_new,Y\_train\_data)

#Test Model

predictions["Multinomial"] = mul\_model.predict(x\_test\_data\_new)

#Calculating Model Accuracy

if rad\_button.value=='Accuracy':

print("Accuracy= "+str(accuracy\_score(y\_test\_data,predictions["Multinomial"])))

#Generating ROC Curve

if rad\_button.value=='ROC Curve':

false\_positive, true\_positive,\_= roc\_curve(y\_test\_data,predictions['Multinomial'])

plt.title('Receiver Operating Characteristic')

plt.plot(false\_positive, true\_positive)

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

#Calculating Precision, Recall and F-Measure

if rad\_button.value=='Precision,Recall and F-Measure':

print(classification\_report(y\_test\_data, predictions['Multinomial'], target\_names = ["Good", "Bad"]))

#3. Bernoulli Naive Bayes Classifier

def bernoulli():

print("\n\nBernoulli Naive Bayes Classifier")

print("Please be patient. This may take some time.")

ber\_model = BernoulliNB()

#Train Model

ber\_model.fit(X\_train\_data\_new,Y\_train\_data)

#Test Model

predictions["BernoulliNB"]=ber\_model.predict(x\_test\_data\_new)

#Calculating Model Accuracy

if rad\_button.value=='Accuracy':

print("Accuracy = "+str(accuracy\_score(y\_test\_data,predictions["BernoulliNB"])))

#Generating ROC Curve

if rad\_button.value=='ROC Curve':

false\_positive,

true\_positive,\_= roc\_curve(y\_test\_data,predictions["BernoulliNB"])

plt.title('Receiver Operating Characteristic')

plt.plot(false\_positive, true\_positive)

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

#Calculating Precision, Recall and F-Measure

if rad\_button.value=='Precision,Recall and F-Measure':

print(classification\_report(y\_test\_data, predictions["BernoulliNB"], target\_names = ["Good", "Bad"]))

# CHAPTER 7

# PERFORMANCE ANALYSIS

* 1. **TEST CASES AND REPORT**

**7.1.1 PREDICTIONS RATING FOR NEW REVIEW**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Customer review** | **ratings** | **Expected Result** | **Actual Result** | **Testcase: Pass/fail** |
| 1 | Excellent | Good | Good | Good | Pass |
| 2 | Worst | Bad | Bad | Bad | Pass |
| 3 | Useless product | Bad | Bad | Bad | Pass |
| 4 | Product came without sealed container | Bad | Bad | Bad | Pass |

**7.1.2 ANALYZING EACH MODEL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | **ALGORITHMS** | **ACTUAL**  **RESULT** | **EXPECTED**  **RESULT** | **TEST CASE PASS /FAIL** |
| 1 | Logistic regression | 0.943 | 0.901 | pass |
| 2 | SVM | 0.768 | 0.745 | pass |
| 3 | Multinomial Naïve Bayes | 0.987 | 0.976 | pass |
| 4 | K-NN | 0.4556 | 0.422 | pass |
| 5 | Ensemble | 0.789 | 0.752 | pass |
| **6** | Decision tree | 0.567 | 0.543 | pass |

**7.1.3 ANALYZE (Precision,Recall and F-Measure)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SNO** | **ALGORITHM** | | **CHECK** | **PRECISION** | | **RECALL** | **FI-SCORE** | | **SUPPORT** | | **TESTCASE PASS /FAIL** |
| **1.** | Logistic Regression Classifier | | GOOD  BAD | 0.94  0.95 | | 0.95  0.94 | 0.95  0.94 | | 4101  3899 | | Pass |
| **2.** | SVM | | GOOD  BAD | 0.97    0.97 | | 0.97  0.97 | 0.97  0.97 | | 4101  3899 | | Pass |
| **3.** | Multinomial Naive Bayes Classifier | | GOOD  BAD | 0.94  0.94 | | 0.94  0.94 | 0.94  0.94 | | 4101  3899 | | Pass |
| **4.** | Bernoulli Naive Bayes Classifier | | GOOD  BAD | 0.94  0.74 | | 0.68  0.96 | 0.79  0.83 | | 4101  3899 | | Pass |
| **5.** | kNN Classifier | | GOOD  BAD | 0.99  0.80 | | 0.77  0.99 | 0.86  0.89 | | 4101  3899 | | Pass |
| **6.** | Ensemble Classifier | | GOOD  BAD | 0.96  0.97 | | 0.97  0.96 | 0.97  0.96 | | 4101  3899 | | Pass |
| **7.** | Decision tree | GOOD  BAD | | 0.96  0.93 | 0.93  0.96 | | 0.95  0.95 | 4101  3899 | | Pass | |

# CHAPTER 8

# CONCLUSION

#### CONCLUSION AND FUTURE ENHANCEMENT

In this study, we have used three review dataset such as product, for the experimentation. The method is suggested for classifying neutral reviews beyond the positive and negative reviews After preprocessing the text of the review comments, the features are extracted using a bag of words model, n-gram and TF-IDF. Then, the data was split into training (80%) and testing datasets (20%). Then it is given as input to the machine learning algorithms like Naive Bayes, Linear Regression, Support vector machine and Decision tree. To evaluate the performance of these algorithms with and without Lexicon based approach, the metrics like accuracy, precision, f-measure, recall, and AUC were used. From the observational results, we conclude that the approach of using helps to solve the issue of binary classification by predicting the third class neutral. Further, by using different feature extraction models like Tf-IDF, Bag-Of-Words and N-gram it is understood that Tf-IDF yield better accuracy for all the three datasets Twitter, Movie and Product review comments. From this it is understood that Logistic Regression works better than other classifiers .In the forthcoming, we can experiment with different preprocessing techniques, heterogeneous features and other machine learning techniques for acquiring even more accurate syste

* 1. **RESULTS AND DISCUSSION**

**RESULT**:

We have proposed a product aspect ranking framework to identify the important

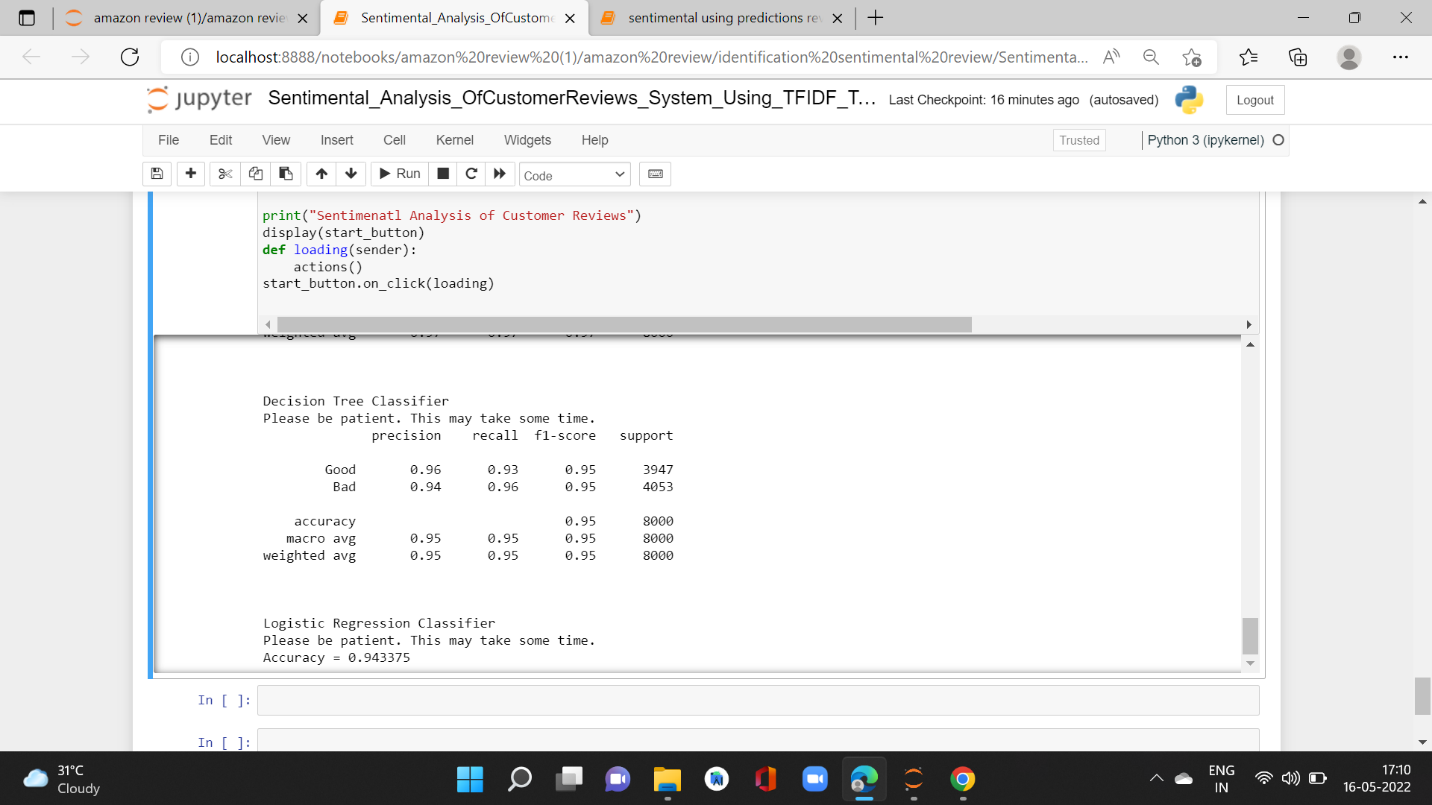
aspects of products from numerous consumer reviews. The framework contains

three main components, i.e., product aspect identification, aspect sentiment classification, and aspect ranking. First, we exploited the Pros and Cons reviews to improve aspect identification and sentiment classification on free-text reviews. We then developed a probabilistic aspect ranking algorithm to infer the importance of various aspects of a product from numerous reviews. The algorithm simultaneously explores aspect frequency and the influence of consumer opinions given to each aspect over the overall opinions.

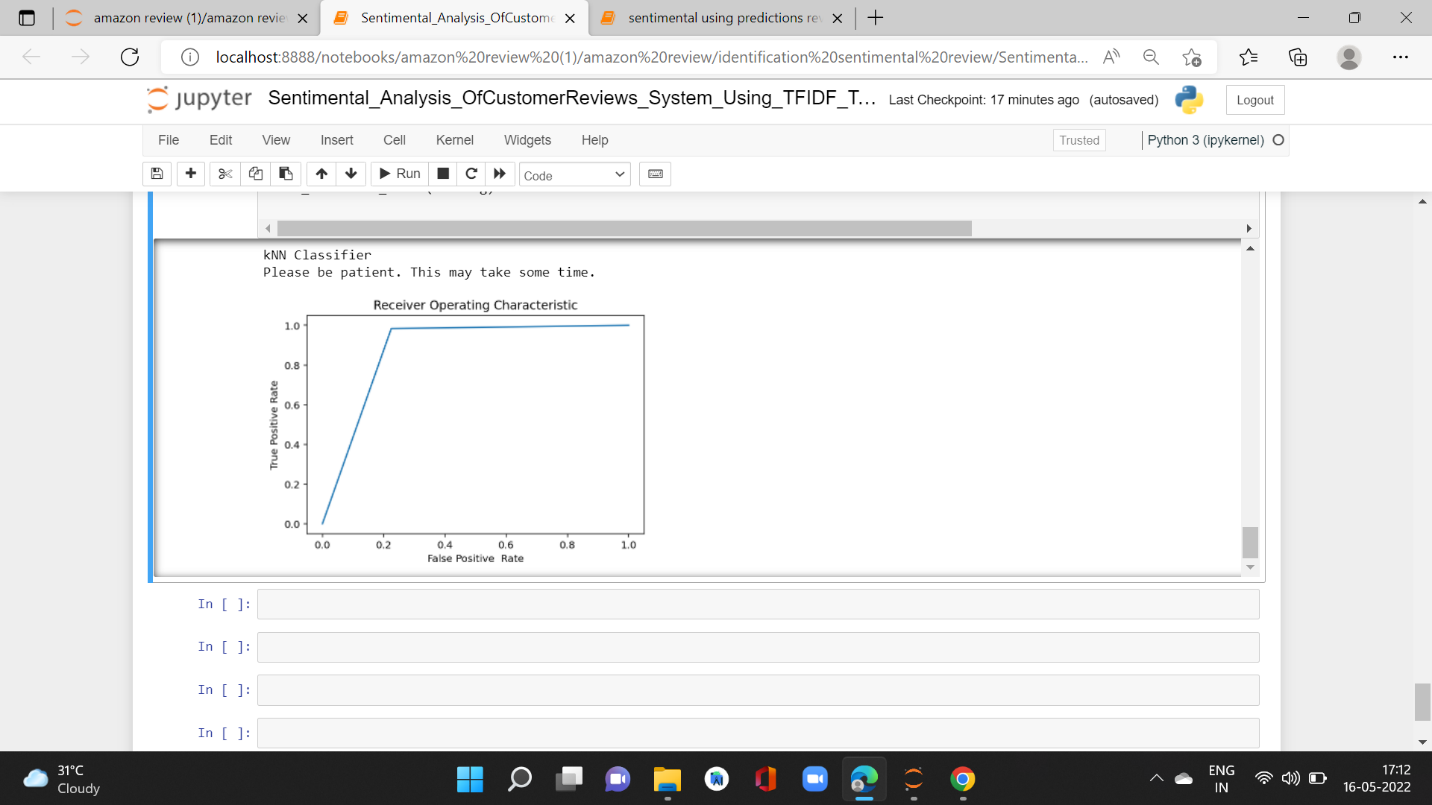
**DISCUSSION:** The product aspects are finally ranked according to their importance scores. We have conducted extensive experiments to systematically evaluate the proposed framework. The experimental corpus contains 94,560 consumer reviews of 21 popular products in eight domains. This corpus is publicly available by request.Facilitate two real-world applications, i.e., document-level sentiment classification and extractive review summarization. Significant performance improvements have been obtained with the help of product aspect ranking.

#### APPENDICES

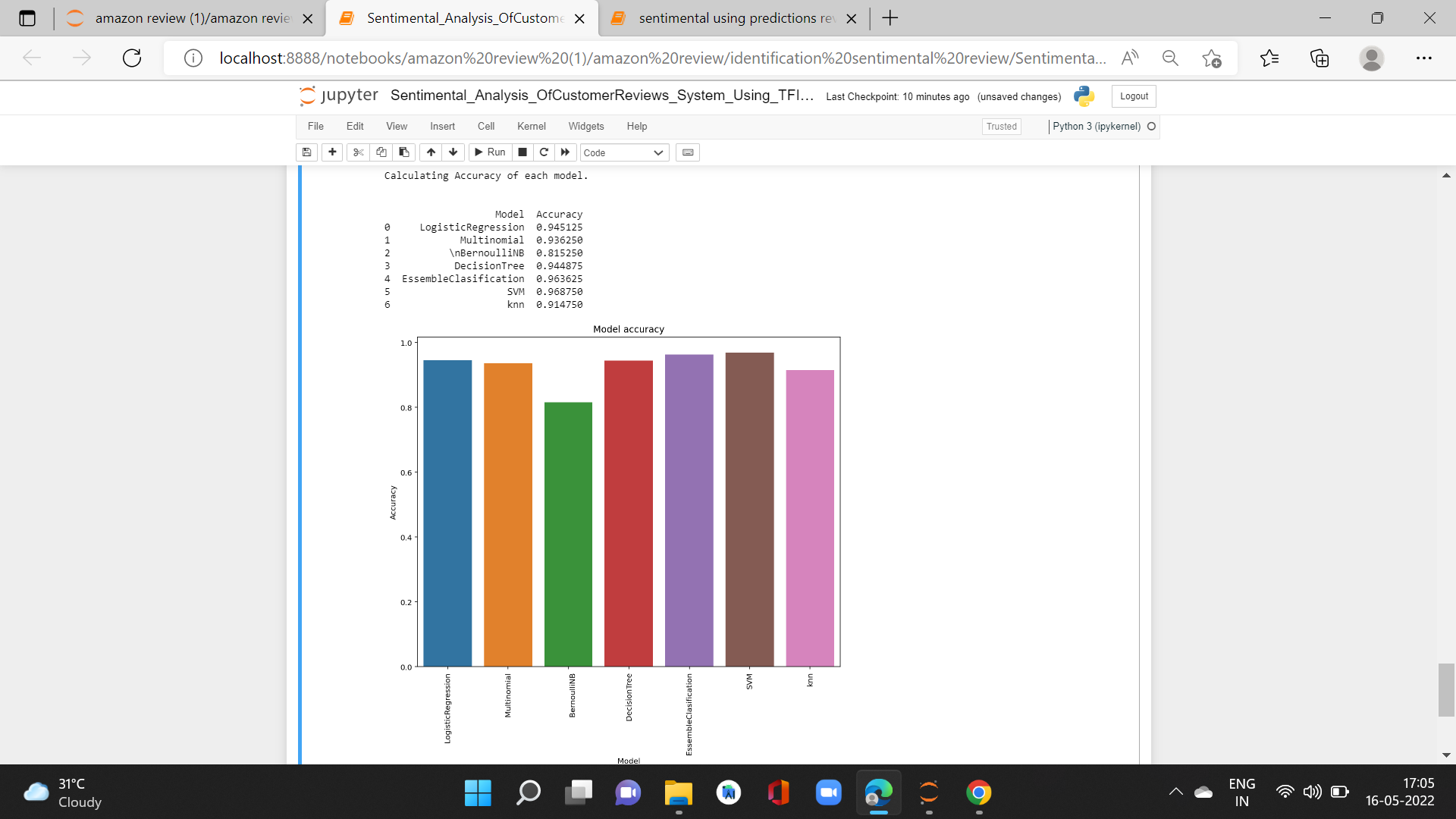
**A.1 SAMPLE SCREEN SHOTS**

****

A.1.Analyzing between models

****

A.2.Finding the accuracy using ROC curve



A.3.Comparing between the models and their accuracy

REFERENCES

[1] Zhihai Yang, Qindong Sun, Yaling Zhang, and Wei Wang,” Identification of Malicious Injection Attacks in Dense Rating and Co-visitation Behaviors”,pp 2021.

[2] Rami Mohawesh , Shuxiang Xu , Son N. Tran , Robert Ollington , Matthew Springer , Yaser Jararweh , Sumbal Maqsood,” Fake Reviews Detection”,pp 2021.

[3] Himangshu Paul andAlexander Nikolaev,” Fake review detection on online E-commerce platforms: a systematic literature review”,pp 2021.

[4] Sundari .P.Shanmuga and Subaji.M,” A comparative study to recognize fake ratings in recommendation system using classification techniques”,pp 2021.

[5] Joni Saliminen, Chandhrasekhar Kandpal, Ahmed Mohamed kamel, Soon-gyo Jun,” Creating and detecting fake reviews of online products”,pp 2021.

[6] Anindya Ghose and Panagiotis G. Ipeirotis, “Estimating the Helpfulness andEconomic Impact of Product Reviews: Mining Text and Reviewer Characteristics, pp 2022.

[7] Vikas Sindhwani and Prem Melville, “Document-Word Co-regularization

for Semi-supervised Sentiment Analysis,” IEEE, pp.Jan. 2020.

[8] Tak-Lam Wong and Wai Lam, “Hot item mining and summarization from

multiple auction Web site,” Mar.2020.

[9] Yang Liu1, Xiangji Huang2, Aijun An1 and Xiaohui Yu2, “Modeling and

Predicting the Helpfulness of Online Reviews,” pp. Sep. 2020.

[10] Kumar Ravi,Vadlamani Ravi ,”A survey on opinion mining and sentiment analysis Tasks approaches and applications in Knowledge-Based Systems”,pp.2019.