LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

# FINAL YEAR PROJECT FINAL REPORT

## 2019-20 Autumn


**Student Name : Riya Shakya**

**London Met ID : 17031225**

**College ID : np01cp4a170134**


**Assignment Due Date : June 5th, 2020**

**Assignment Submission Date : June 5th, 2020**

**Word Count : 8,300**

**External Supervisor : Ishwor Shrestha**

**Internal Supervisor : Subeksha Shrestha**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

# Acknowledgement

It is always a pleasure to remind the people in the Islington College for their sincere guidance; I received to validate the Final Year Project (FYP). I am grateful to Islington College for such an opportunity to undergo a project which helps me in self-growth.

The success of the outcome of this project entailed a fair amount of guidance, co-operation and encouragement from many people and I am tremendously privileged to have got their support throughout the completion of the project. All that I have accomplished is only due to such supervision and assistance for which we will be for thankful.

First, a wholehearted thank you to my parents for the encouragement, enthusiasm, invaluable assistance and support towards me endlessly. This project could not be completed without all this.

Second, I would like to respect and thank Mr Ishwor Shrestha as my external supervisor for FYP who helped me a lot in dealing with the problems of the project and suggesting suitable solutions to the problem. I would also like to express my deepest thanks to Miss. Subeksha Shrestha as my internal supervisor for FYP who guided me through the completion of FYP. I would not forget to remember her encouragement and for their timely support and guidance.

Third, I would like to thank Mr Monil Adhikari has supported me by clearing any confusion or misunderstanding related to the project. I would also like to thank Mr Sukrit Shakya for his help to clear out any confusion related to the FYP topic. I am extremely thankful to him for providing such constant support and guidance although he had a busy schedule.

Besides, I am thankful to and fortunate for all teaching staff of Islington for the encouragement, support and guidance which motivated me in completing my project work.

Finally, I would like to apologize to all other unnamed who supported and guided me in various ways in the completion of the task.

## Abstract

Recent studies have shown that Sentiment Analysis of reviews is being implemented across the globe as one of the few arising AI. However, even though being widely used worldwide the result of the Sentiment Analysis can be taken as a final statement without the supervision of human as it is not yet accurate due to the use of sarcasm in reviews which is yet to be understood by the machine. Being a part of AI, it gets more accurate with practice. This report is about Sentiment Analysis on data mined from Trip advisor. The report covers the introduction, background, development, testing and future work.

The report is on the design of sentiment analysis, extracting and training reviews of vast amount. The result classifies the reviews of the customer of the hotel. It contains chapters, among which Introduction which gives a summary of the project, the Background gives conceptional understand the person related to their project topic, Development of the project, Testing and Analysis proves the smooth workflow of the system, and Future work includes all the work that will be done in the future.

## Table of Abbreviation

| S. No. | Abbreviation | Full Forms |
|---|---|---|
| **1.** | AI | Artificial Intelligence |
| **2.** | ML | Machine Learning |
| **3.** | SEO | Search Engine Optimization |
| **4.** | NLP | Natural Language Processing |
| **5.** | POS | Part of Speech |
| **6.** | BESAHOT | Basic functionalities of Opinion Mining as Classification and Extraction Rating from Texts |
| **7.** | GWT | Google Web Toolkit |
| **8.** | MALLET | Machine Learning for LanguagE Toolkit |
| **9.** | NB | Naïve Bayes |
| **10.** | DT | Decision Tree |
| **11.** | SVM | Support Vector Machine |
| **12.** | TF-IDF | Term Frequency – Inverse Document Frequency |
| **13.** | SDLC | Software Development Life Cycle |
| **14.** | XSLT | eXtensible Stylesheet Language Transformations |
| **15.** | SProUT | Shallow text Processing with Unification and Typed feature structure |
| **16.** | SDLC | Software Development Life Cycle |
| **17.** | DNS | Domain Name System |
| **18.** | nltk | Natural Language toolkit |

# Table of Contents


Chapter 1. Introduction ..... 1

1.1 Project Description ..... 1

1.2 Current Scenario ..... 2

1.3 Problem Domain and Project as a Solution ..... 3

1.4 Aim and Objective ..... 4

1.5 Structure of the report ..... 5

1.5.1 Background ..... 5

1.5.2 Development ..... 5

1.5.3 Testing and Analysis ..... 5

1.5.4 Conclusion ..... 5

Chapter 2. Background ..... 6

2.1 About the end users ..... 6

2.2 Understanding the solution ..... 7

2.3 Similar projects ..... 8

2.4 Comparisons of similar system with our system ..... 12

Chapter 3. Development ..... 14

3.1 Considered Methodologies ..... 14

3.2 Selected Methodology ..... 19

3.3 Phases of Methodology ..... 20

3.4 Survey Results ..... 22

3.4.1 Pre-Survey Results ..... 22

3.1.1 Post-Survey Results ..... 26

3.5 Requirement Analysis ..... 31

3.6 Design ..... 32

3.6.1 Use Case of the system ..... 32
3.6.2 High Level Use case ..... 33
3.6.3 Expanded Use Case ..... 36
3.7 Implementation ..... 41
Chapter 4. Testing and Analysis ..... 46
4.1 Test Plan ..... 46
4.1.1 Unit Testing, Test Plan ..... 46
4.1.2 System Testing, Test Plan ..... 47
4.2 Unit Testing ..... 48
4.3 System Testing ..... 65
4.4 Critical Analysis ..... 69
Chapter 5. Conclusion ..... 70
5.1 Legal, Social and Ethical Issues ..... 70
5.1.1 Legal Issue ..... 70
5.1.2 Social Issues ..... 71
5.1.3 Ethical Issues ..... 71
5.2 Advantages ..... 72
5.3 Limitations ..... 73
5.4 Future Work ..... 74
Bibliography ..... 75
Chapter 6. References ..... 78
Chapter 7. Appendix ..... 81
7.1 Appendix A: Pre-survey ..... 81
7.1.1 Pre-Survey Form ..... 81
7.1.2 Sample of Filled Pre-Survey Forms ..... 84

7.1.3 Pre-Survey Result ..... 87
7.2 Appendix B: Post-Survey ..... 91
7.2.1 Post-Survey Form ..... 91
7.2.2 Sample of Filled Post-Survey Forms ..... 94
7.2.3 Post-Survey Result ..... 97
7.3 Appendix C: Sample Codes ..... 102
7.3.1 Sample Code of the front-end ..... 102
7.3.2 Sample Code for the back end ..... 104
7.4 Appendix D: Designs ..... 107
7.4.1 Gantt Chart ..... 107
7.4.2 Work breakdown Structure ..... 108
7.4.3 Algorithms and Flowchart ..... 109
7.4.4 Data Flow Diagrams ..... 111
7.4.5 Individual Use Case ..... 112
7.4.6 Sequence Diagram ..... 116
7.4.7 Activity Diagram ..... 117
7.4.8 Wireframe ..... 118
7.5. Appendix E: Screenshots of the system ..... 121
7.6 Appendix F: User Manual ..... 122

## List of figures:

Figure 1. Aspect based Sentiment Oriented Summarization of Hotel Reviews. ..... 8
Figure 2. Multi-Language Sentiment Analysis for Hotel Reviews. ..... 9
Figure 3. Opinion Mining and Sentiment Analysis of Trip Advisor in for Hotel Reviews. ..... 10
Figure 4. Sentiment Analysis for Hotel Review. ..... 11
Figure 5. Prototype Methodology. ..... 15
Figure 6. Rational Unified Process methodology. ..... 16
Figure 7. Iterative methodology. ..... 17
Figure 8. Waterfall methodology ..... 18
Figure 9. Phases of methodology. ..... 21
Figure 10. Pre-survey result. fig(a) ..... 22
Figure 11. Pre-survey result. fig(b) ..... 22
Figure 12. Pre-survey result. fig(c) ..... 23
Figure 13. Pre-survey result. fig(d) ..... 23
Figure 14. Pre-survey result. fig(e) ..... 24
Figure 15. Pre-survey result. fig(f) ..... 24
Figure 16. Pre-survey result. fig(g) ..... 25
Figure 17. Pre-survey result. fig(h) ..... 25
Figure 18. Post-survey result. fig(a) ..... 26
Figure 19.Post-survey result. fig(b) ..... 26
Figure 20. Post-survey result. fig(c) ..... 27
Figure 21. Post-survey result. fig(d) ..... 27
Figure 22. Post-survey result. fig(e) ..... 28
Figure 23. Post-survey result. fig(f) ..... 28
Figure 24. Post-survey result. fig(g) ..... 29
Figure 25.Post-survey result. fig(h) ..... 29
Figure 26.Post-survey result. fig(i) ..... 30
Figure 27. Use of the system. ..... 32
Figure 28. Main page of the system. ..... 41
Figure 29. Result page of the system. ..... 42

Figure 30. Back-end code of the system. fig(a) ..... 43
Figure 31. Back-end code of the system. fig(b) ..... 44
Figure 32. Back-end code of the system. fig(c) ..... 44
Figure 33. Back-end code of the system. fig(d) ..... 45
Figure 34. Back-end code of the system. fig(e) ..... 45
Figure 35. Unit test 1: Check if the raw extracted data is successfully saved in csv file. ..... 49
Figure 36. Unit test 2: Check if the csv file containing raw extracted data is opened successfully. ..... 50
Figure 37. Unit test 3: Check if the graph “No. of reviews per week” is displayed successfully. ..... 51
Figure 38. Unit test 4: Check if the reviews are filtered successfully. ..... 52
Figure 39. Unit test 5: Check if the filtered reviews are successfully saved in a csv file. ..... 53
Figure 40. Unit test 6: Check if the reviews are label as per their rating successfully. .. 54
Figure 41. Unit test 7: Check if the labelled reviews are successfully saved in a csv file. ..... 55
Figure 42. Unit test 8: Check if the positive reviews are listed after classification. ..... 56
Figure 43. Unit test 9: Check if the negative reviews after classification, display “No Negative Reviews”. ..... 57
Figure 44. Unit test 10: Check if the frequent words are displayed in the diagram. ..... 58
Figure 45. Unit test 11: Check if the positive reviews are displayed in a graph. ..... 59
Figure 46. Unit test 12: Check if there are no negative reviews, display “No Negative Reviews”. ..... 60
Figure 47. Unit test 13: Check if the graph to compare occurrence of negative and positive reviews is displayed ..... 61
Figure 48. Unit test 14: Check if the graph to compare the rating is displayed. ..... 62
Figure 49. Unit test 15: Check if the graph of bigrams is displayed. ..... 63
Figure 50. Unit test 16: Check if the graph of trigrams is displayed. ..... 64
Figure 51. System test 1: Check if the reviews, reviews rating and date of reviews are successfully extracted from web scrapper. ..... 65

Figure 52. System Test 2: Check if the model is trained and gives high accuracy. ..... 66
Figure 53. System Test 3: Check if the graphs are displayed in the web page successfully. ..... 67
Figure 54. System Test 4: Check if the data of web page can be downloaded successfully. ..... 68
Figure 55. Pre-survey form. fig(a) ..... 81
Figure 56. Pre-survey form. fig(b) ..... 82
Figure 57. Pre-survey form. fig(c) ..... 83
Figure 58. Sample of filled pre-survey. Fig(a) ..... 84
Figure 59. Sample of filled pre-survey. Fig(b) ..... 85
Figure 60. Sample of filled pre-survey. Fig(c) ..... 86
Figure 61. Pre-survey result. fig(a) ..... 87
Figure 62. Pre-survey result. fig(b) ..... 87
Figure 63. Pre-survey result. fig(c) ..... 88
Figure 64. Pre-survey result. fig(d) ..... 88
Figure 65. Pre-survey result. fig(e) ..... 89
Figure 66. Pre-survey result. fig(f) ..... 89
Figure 67. Pre-survey result. fig(g) ..... 90
Figure 68. Pre-survey result. fig(h) ..... 90
Figure 69.Post-survey form. fig(a) ..... 91
Figure 70. Post-survey form. fig(b) ..... 92
Figure 71. Post-survey form. fig(c) ..... 93
Figure 72. Sample of filled post-survey. Fig(a) ..... 94
Figure 73. Sample of filled post-survey. Fig(b) ..... 95
Figure 74. Sample of filled post-survey. Fig(c) ..... 96
Figure 75. Post-survey result. fig(a) ..... 97
Figure 76. Post-survey result. fig(b) ..... 97
Figure 77. Post-survey result. fig(c) ..... 98
Figure 78. Post-survey result. fig(d) ..... 98
Figure 79. Post-survey result. fig(e) ..... 99
Figure 80. Post-survey result. fig(f) ..... 99

Figure 81. Post-survey result. fig(g) ..... 100
Figure 82. Post-survey result. fig(h) ..... 100
Figure 83. Post-survey result. fig(i)..... 101
Figure 84. The front-end code of the system. fig(a) ..... 102
Figure 85. The front-end code of the system. fig(b) ..... 103
Figure 86. The front-end code of the system. fig(c)..... 103
Figure 87. The back-end code of the system. fig(a) ..... 104
Figure 88.The back -end code of the system. fig(b) ..... 104
Figure 89. The back-end code of the system. fig(c) ..... 105
Figure 90. The back-end code of the system. fig(d) ..... 105
Figure 91. The back-end code of the system. fig(e) ..... 106
Figure 92. The back-end code of the system. fig(f) ..... 106
Figure 93. Gantt Chart of the project. ..... 107
Figure 94. Work break down structure of the system. ..... 108
Figure 95. Flowchart of the system. ..... 110
Figure 96. DFD of the system. ..... 111
Figure 97. Individual Use case: Input Link..... 112
Figure 98. Individual Use case: Data Extraction..... 112
Figure 99. Individual Use case: Filter Reviews..... 113
Figure 100.Individual Use case: Label Review..... 113
Figure 101. Individual Use case: Classify Report. ..... 114
Figure 102. Individual Use case: Create Report..... 114
Figure 103. Individual Use case: View Report..... 115
Figure 104. Individual Use case: Download Report. ..... 115
Figure 105. Sequence diagram of the system..... 116
Figure 106 Activity diagram of the system..... 117
Figure 107. Wireframe: Main page..... 118
Figure 108. Wireframe: Result page fig (a). ..... 119
Figure 109. Wireframe: Result page fig (b). ..... 120
Figure 110. Wireframe: Result page fig (c)..... 120
Figure 111. The system UI. fig(a)..... 121

Figure 112.  The system UI. fig(b) ..... 121
Figure 113. User manual: Drop the link of hotel from trip advisor website. ..... 122
Figure 114 User manual: Start sentiment analysis. ..... 123
Figure 115. User Manual: View result of sentiment analysis. ..... 123
Figure 116. User Manual: Download the sentiment report. ..... 124

## List of tables:

Table 1. Comparison of similar system with our system..... 13
Table 2 Unit Test Plan..... 46
Table 3. System Test Plan. ..... 47
Table 4. Unit test 1: Check if the raw extracted data is successfully saved in csv file... 48
Table 5. Unit test 2: Check if the csv file containing raw extracted data is opened successfully..... 50
Table 6. Unit test 3: Check if the graph “No. of reviews per week” is displayed successfully..... 51
Table 7. Unit test 4: Check if the reviews are filtered successfully. ..... 52
Table 8. Unit test 5: Check if the filtered reviews are successfully saved in a csv file... 53
Table 9. Unit test 6: Check if the reviews are label as per their rating successfully. ..... 54
Table 10. Unit test 7: Check if the labelled reviews are successfully saved in a csv file. ..... 55
Table 11. Unit test 8: Check if the positive reviews are listed after classification. ..... 56
Table 12. Unit test 9: Check if the negative reviews after classification, display “No Negative Reviews”. ..... 57
Table 13. Unit test 10: Check if the frequent words are displayed in the diagram. ..... 58
Table 14. Unit test 11: Check if the positive reviews are displayed in a graph. ..... 59
Table 15. Unit test 12: Check if there are no negative reviews, display “No Negative Reviews”. ..... 60
Table 16. Unit test 13: Check if the graph to compare occurrence of negative and positive reviews is displayed. ..... 61
Table 17. Unit test 14: Check if the graph to compare the rating is displayed..... 62
Table 18. Unit test 15: Check if the graph of bigrams is displayed..... 63
Table 19. Unit test 16: Check if the graph of trigrams is displayed..... 64
Table 20. System test 1: Check if the reviews, reviews rating and date of reviews are successfully extracted from web scrapper..... 65
Table 21. System Test 2: Check if the model is trained and gives high accuracy. ..... 66
Table 22. System Test 3: Check if the graphs are displayed in the web page successfully. ..... 67

Table 23. System Test 4: Check if the data of web page can be downloaded successfully. ..... 68

# Chapter 1. Introduction

## 1.1 Project Description

In today's world, the internet has made a huge impact on human life. The introduction of the internet has changed society, driving it forward from the industrial age to the network era. The internet has changed the business, education, government, healthcare, and even ways in which we interact with our loved one, becoming one of the key drivers of social evolution. The speed of the information transfer increases with new technologies.

The internet provides the platform of e-commerce which has possibilities for buying content, new and leisure products making it a major distribution of channel for goods and service. The security provided to the business transactions by new applications allows new commercial business. The consumer's gain the upper hand as the access to information multiplies, and their reviews of their experiences with various products and services to take centre stage. The facility to compare products and ranks, user reviews and comments and recommendations from bloggers with large followings have given a new shape for consumer behaviour, retail trade, and the economy in general. These reasons make consumers prefer online shopping more. The online business owners must not only make sure that their products are of the best quality but also make sure the reviews from the consumers are good to attract more consumers.

With an increasing number of consumers, it is impossible to keep track of reviews of the product manually. The problem of numerous reviews can be handled with the help of Artificial Intelligence (AI) and Machine Learning (ML). The reviews can be identified as positive and negative using Sentiment Analysis. There are various web applications which analysis the sentiment of reviews of the product, tweets, and such post available on the internet to solve this issue.

## 1.2 Current Scenario

The e-commerce platform introduced has introduced many online retailers. The e-commerce facility is often used to refer to the sale of physical products online, but it also includes wholesaler, drop-shipping, crowdfunding, subscriptions, physical products, digitals products and services. With every shop at their disposal, the consumers prefer online shopping. The online is easy but does not lead to a blind purchase. In a local consumer review survey, it was proofed that 93% of consumer spend time reading the reviews before making the purchase, 91% of consumers are more likely to contact the business after reading positive reviews and 82% of consumers are less likely to use a business after seeing negative reviews (Murphy, 2020)

The top industries where online reviews matter is restaurants, grocery stores, medical, clothing stores, hotels, entertainment, automotive, hair and beauty, pet services and car dealerships. The business must ensure the reviews they are getting are broadly positive for the achievement of success in the field of e-commerce. It does not mean focusing on just a happy customer, but all the feedback was given by the customers to build a better business. The negative feedback strongly impacts on the decision of the customer while choosing a business. The online reviews are critical tools as good reviews make the business more trustable. The reviews are not only important for the consumer but also important for the business owners.

The reviews in the new concept of word-of-mouth for business. The good reviews help to improve search engine optimization (SEO) of business by creating unique, up-to-date content on ongoing business (Podium, 2020). The business owners get to know what the consumer thinks of their products or service and how to improve their bank ranks on different platforms. The positive reviews from social influencer attract customer. The negative reviews point out the areas for improvement as it points out the product so we can solve and analyse the demand of customer and keep an eye on the competition. Keeping track of reviews shows the customers that the business owners care about their experience.

## 1.3 Problem Domain and Project as a Solution

**Problem Domain:**

Hotels are an imperative part of the tourism industry. The success of a hospitality business like a hotel depends on the client whether the client enjoys the hotel's services or not. The consumers' opinions on hotel facilities can be known through customer reviews. Trip Advisor is one of the largest online travel review sites with a strong network effect, where guest reviews can be found. It influences 40-50% of all online travel with an annual growth rate of 43% (Tripadvisor, 2017). The reviews of Trip Advisor greatly impact the business as it is the foundation to rating, influence booking, and evolvement of business all over. Reviews provide a strong value for the hospitality business as more reviews are more engagement and the mistake made by many hospitality businesses is not actively collecting guest reviews on sites like Trip Advisor. Trip Advisor has more than 280 traveller reviews and opinion submitted to the site per minute, making it difficult to go through the hotel reviews manually (Advisor, 2019). Therefore, it is hard for hotel business owner to determine what customer loves and hates based on huge data of reviews.

**Project as a solution:**

A web application will be developed to tackle the problem mentioned above. The application will have an input area where the user can input Trip Advisor's hotel link. The web application will analyse the reviews of Trip Advisor and give proper sentiment analysis of reviews in the form of textual and virtualization. The website will help the company to know whether the product is preferred by the consumers or there is a weakness that needs change or perhaps marketing policy is not practical and many other factors.

## 1.4 Aim and Objective

This project aims to apply sentiment analysis on the enormous amount of reviews of hotels in Trip Advisor website and provide the sentiment of the reviews to the hotel owners which help to manage their reputation by improving their services offered and gauge the general customer attitude to their hotel.

The objectives of this project are listed below:

- To gain knowledge about Natural Language Processing (NLP) and Machine Learning (ML) to create a web application.
- To develop a web application which will reflect the learning outcome of NLP and ML.
- To give a textual and virtual presentation of the reviews of Trip Advisor through the application which will give the hotel owner a clear idea of the customer review.
- The project will deliver a structured review of the clients to the hospitality.
- The reviews of the customers are mainly unstructured reviews which are difficult, time-consuming and expensive to analyse, understand and sort through manually but this project helps make sense of the unstructured text by automating business processes, getting actionable insights, and saving hours of manual data.

## 1.5 Structure of the report

### 1.5.1 Background

The main theme of this chapter is to introduce the result obtained by the end-users, understanding solution of the problem, review of a similar web application, various libraries used by the application and comparison between this application and other existing application.

### 1.5.2 Development

The main theme of this chapter is to introduce the considered methodologies, selected methodology, and phases of methodology with a brief explanation of the activities of each phase. The chapter includes pre-survey and post-survey result done for the development of the application. The chapter also includes requirement analysis which analyses the feature of the application, tests to minimize the occurrence of errors and removal of errors, and conclusion.

### 1.5.3 Testing and Analysis

The main theme of this chapter is to list out the tests carried out during the development of the application such as unit testing and system testing and critical analysis of the application.

### 1.5.4 Conclusion

The main theme of this chapter is to conclude the report with legal, social, and ethical issues regarding the application, the advantages, and limitations of the application, along with the future work to be carried out in future.

# Chapter 2. Background

## 2.1 About the end users

The end-users of this application can be anyone interested to know the sentiment of the reviews available on the trip advisor website. The main targeted end users are the hotel's owners who need to go through the reviews of the customers for the improvement of their business. The users will get the sentiment summary of the reviews with some positive and negative reviews with a graphical presentation for easier understanding. The hotel owners can use positive reviews as the means of attraction for new customers whereas the negative reviews can be used as a room for improvement.

## 2.2 Understanding the solution

The project aims to apply sentiment analysis on the reviews of the hotel in trip advisor website to give a summarise report through graphs to the owners of the hotel. The summarise report helps to know the market value of their hotel. Sentiment analysis interprets and classifies the text as positive or negative with the help of text analysis techniques. It uses an algorithm to train the model which is used to classify the text. The quality of data used to train the model decides the accuracy of the model.

In this project, the data is extracted from the trip advisor website using beatifulsoup4. The extracted data is saved in a CSV file. The data of the CSV file is filtered using the nltk of NLP. The nltk platform consists of more than 50 corpora and lexical resources along with text processing libraries for classification, tokenization, stemming, tagging, parsing and semantic reasoning and more. It was used for classification, tokenization, and stemming the reviews. The filtered data is labelled with the help of the rating. The model of sentiment analysis is made with the help of the ML algorithm, Naïve Bayes algorithm. There are other ML algorithms which can be used for sentiment analysis such as DT, Logical Regression and SVM. The Naïve Bayes algorithm is simple and powerful. It is a probabilities theorem based on Bayes Theorem. The algorithm is used to train the model and classify the reviews. The accuracy of the model is tested, the accuracy of the model changes with the data used to train the model.

The analysis of the reviews is presented with the help of the matplotlib and plotly. The framework Flask is used to display the result of the analysis in the webpage. The flask framework display analysis with the help of graph and reviews listed for easier understanding of the summary. The analysis of the report can be downloaded with the help of the download button. The project analyse the reviews of the hotel selected by the user and displays the analysis report through graphs.

## 2.3 Similar projects

- **Aspect based Sentiment Oriented Summarization of Hotel Reviews**

  The propose of this system is for the betterment of the customers. This system was created so the customer can find the hotel they are looking as the system summarizes the review and provides understanding for the customer. It only deals with German reviews from German websites. In this system, the tools used are python libraries 'beautiful' and 'urlib', OpenNER, Basic functionalities of opinion mining as classification and extraction rating from texts (BESAHOT), Google Web Toolkit (GWT) framework, Machine Learning for LanguagE Toolkit (MALLET) and Senti-Word net corpus (Nadeem Akhtara, 2017).

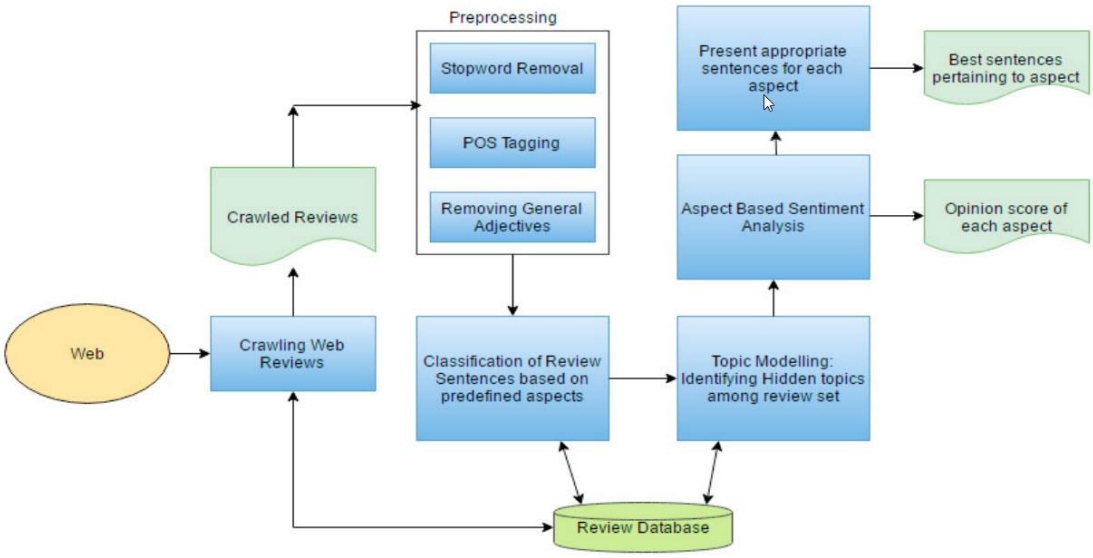



*Figure 1. Aspect based Sentiment Oriented Summarization of Hotel Reviews.*

- **Multi-Language Sentiment Analysis for Hotel Reviews**

  The purpose of the system is decision making easier for the customer. The detailed information about the hotel and a large amount of subjective information from the previous guest of their quality makes it difficult to make a sensible choice. It deals with reviews in English and Thai. In this system, the sentiment analysis is done using supervised learning algorithm: Naïve Bayes (NB), Decision Trees (DT), and Support Vector Machine (SVM) with two types of information: frequency and Term Frequency – Inverse Document Frequency (TF-IDF) (EDP Sciences, 2016).

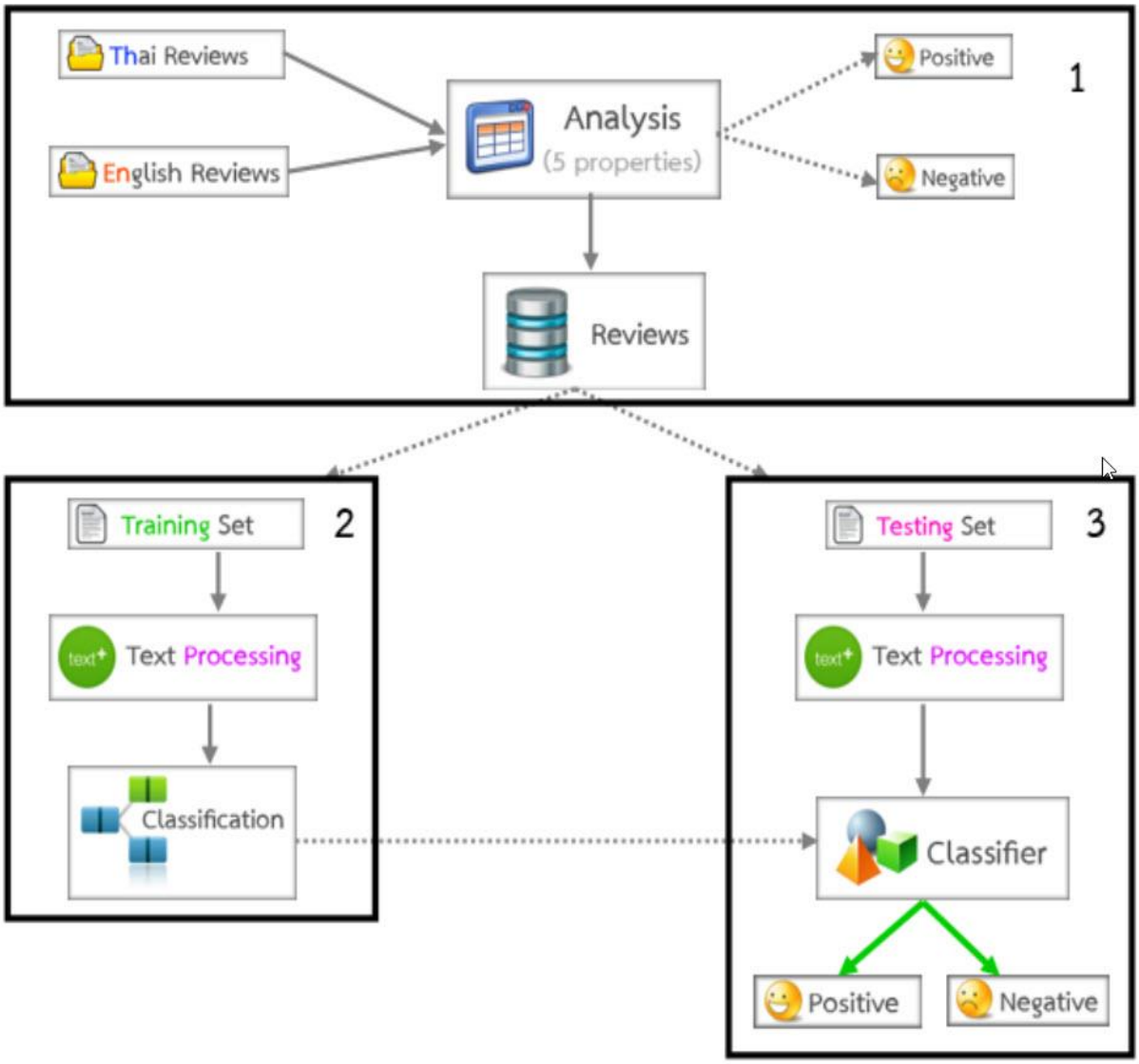



*Figure 2. Multi-Language Sentiment Analysis for Hotel Reviews.*

- **Opinion Mining and Sentiment Analysis of Trip Advisor in for Hotel Reviews**

  The purpose of this system is to provide essential and necessary information about reviews of the hotel. This system clusters the mined data, the clustered data go through classification. In this system, the sentiment analysis is done using K-nearest neighbour (KNN), Naïve Bayes, Sequential Minimal Optimization (SMO), Partition Around Medoids (PAM), and j48 algorithm for Decision Tree (Divyashree N, Santhosh Kumar K L, Jharna Majumdar, 2017).

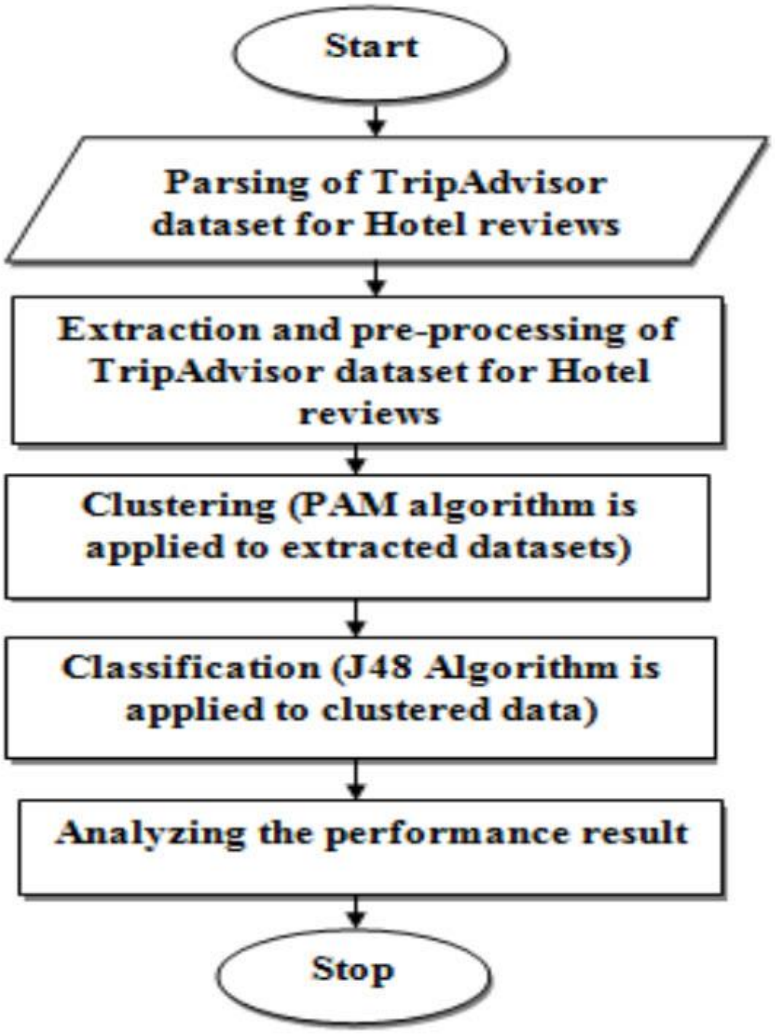



*Figure 3. Opinion Mining and Sentiment Analysis of Trip Advisor in for Hotel Reviews.*

- **Sentiment Analysis for Hotel Reviews**

  The purpose of this system is to get an actual overview and summaries of textual comments about the hotel for a customer as well as hotel managers. This system is a part of BESAHOT system that handles data acquisition, analysis, and storage. It only deals with German reviews. In this system, the sentiment analysis is done using eXtensible Stylesheet Language Transformations (XSLT) script, statistic polarity classification, and Shallow text Processing with Unification and Typed feature structure (SProUT) (Walter Kasper, Mihaela Vela, 2011).

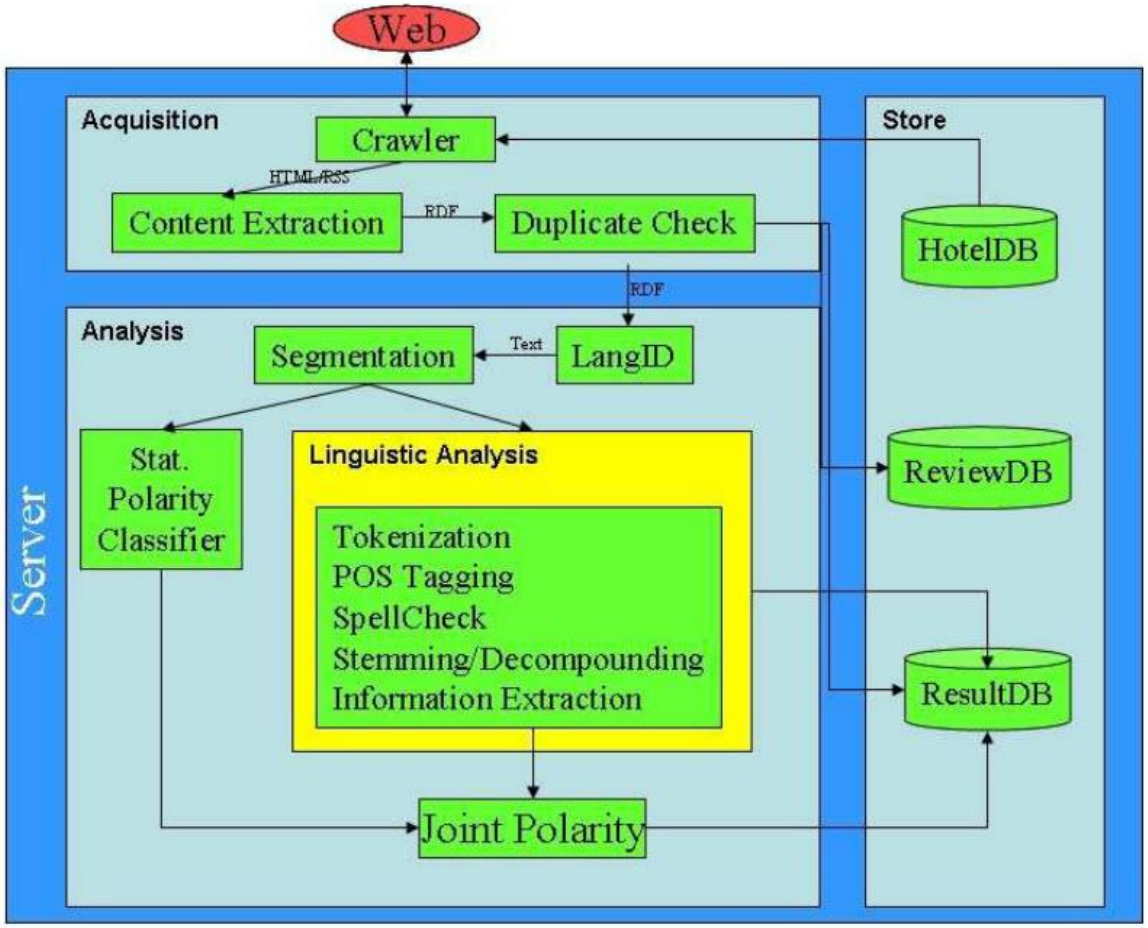



*Figure 4. Sentiment Analysis for Hotel Review.*

## 2.4 Comparisons of similar system with our system

| Feature | Sentiment Analysis of Hotel Reviews mined from Trip Advisor (My Project) | Aspect based Sentiment Oriented Summarization of Hotel Reviews | Multi-Language Sentiment Analysis for Hotel Reviews | Opinion Mining and Sentiment Analysis of Trip Advisor in for Hotel Reviews | Sentiment Analysis for Hotel Reviews |
|---|---|---|---|---|---|
| **English Language** | Yes | No | Yes | Yes | No |
| **List the bigrams in graph** | Yes | No | No | No | No |
| **List the trigrams in graph** | Yes | No | No | No | No |
| **Positive reviews listed** | Yes | Yes | No | No | No |
| **Negative reviews listed** | Yes | Yes | No | No | No |
| **Graph of reviews occurrence** | Yes | No | No | No | No |
| **Graph of positive reviews occurrence** | Yes | No | No | Yes | No |
| **Graph of negative reviews** | Yes | No | No | Yes | No |

| | | | | | |
|---|---|---|---|---|---|
| **occurrence** | | | | | |
| **List of frequent words in graph** | Yes | Yes | No | No | No |
| **Compare positive and negative reviews in graph** | Yes | No | No | Yes | No |
| **Compare rating in graph** | Yes | No | No | No | No |

*Table 1. Comparison of similar system with our system.*

Most of the sentiment analysis does not filter the bigrams and trigrams before classification of reviews. A simple bigram or trigram word can make a big impact on the result of the classification, so it is important to filter the bigrams and trigrams. The features of bigrams and trigrams are not included in the above similar projects.

This project displays a list of positive and negative reviews. It helps the user to go through the sentiment as per their requirement. It also displays graphs to know the positive and negative reviews, the number of reviews per in timeline, and list of frequent words makes it easy to go through the reviews and saves time.

# Chapter 3. Development

## 3.1 Considered Methodologies

The framework used to structure, plan, and control the process of developing an information system is a software development methodology. The process of developing and writing code can be organised in various ways. There are many methodologies such as Waterfall, Prototype, Spiral model, Rational unified process, Iterative and so on (Software, 2020). The considered methodologies are listed below:

- **Prototype Methodology**

The prototype methodology is a specialized software development methodology which initiates developer to make a simple resolution to meet the essence of the customer and make essential changes as time passed per request of the client before creating the authentic final solution (K, 2018).

**Justification to consider the methodology**

- It gives a clear idea of the workflow of the process of the software which reduces the risk of failure in a solution.
- It assists well in requirement gathering and analysis of the overall system.
- It requires excessive involvement of the client.

**Reasons to not select the methodology**

- Developers can get attached to prototypes due to their hard works and efforts on it due to which final system may not be like the desire of the client.
- Requirements of clients may not remain the same due to which sometimes final output may not be delivered to the client
- It has a chance of extension in management cost due to many changes occur during the development phase.

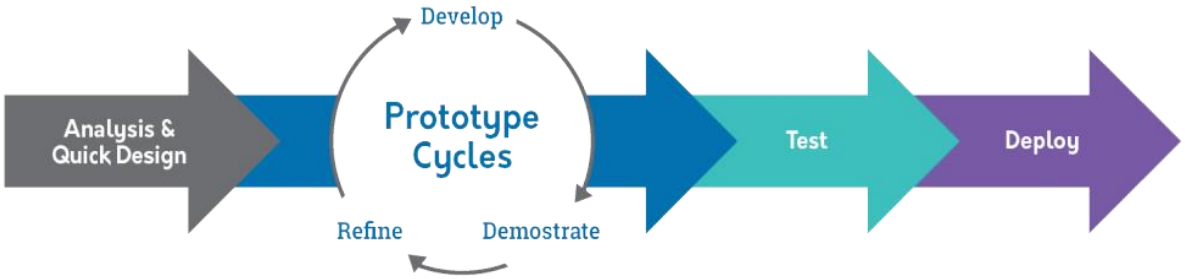



*Figure 5. Prototype Methodology.*

- **Rational Unified Process Methodology**

This methodology powers software using rational tools, it divides the development process into four different stages which include business modelling, scrutiny and design, enactment, testing, and disposition. It assists software developer by stating guidelines, templates, and specimens for all feature and stages of software development (K, 2018).

**Justification to consider the approach**

- It is focused on precise documentation, removes risks in the project linked with the client, and less interaction with the client.
- It takes the best part of Waterfall and changes them into the iterative process to allow change.
- It is architecture-centric / component-based.

**Reasons to not select the methodology**

- It requires the involvement of an expert software developer, complicated methodology, and integration is also very complicated to understand.
- It is process heavy and be slow for a certain type of projects.
- It relies heavily on stakeholder feedback.

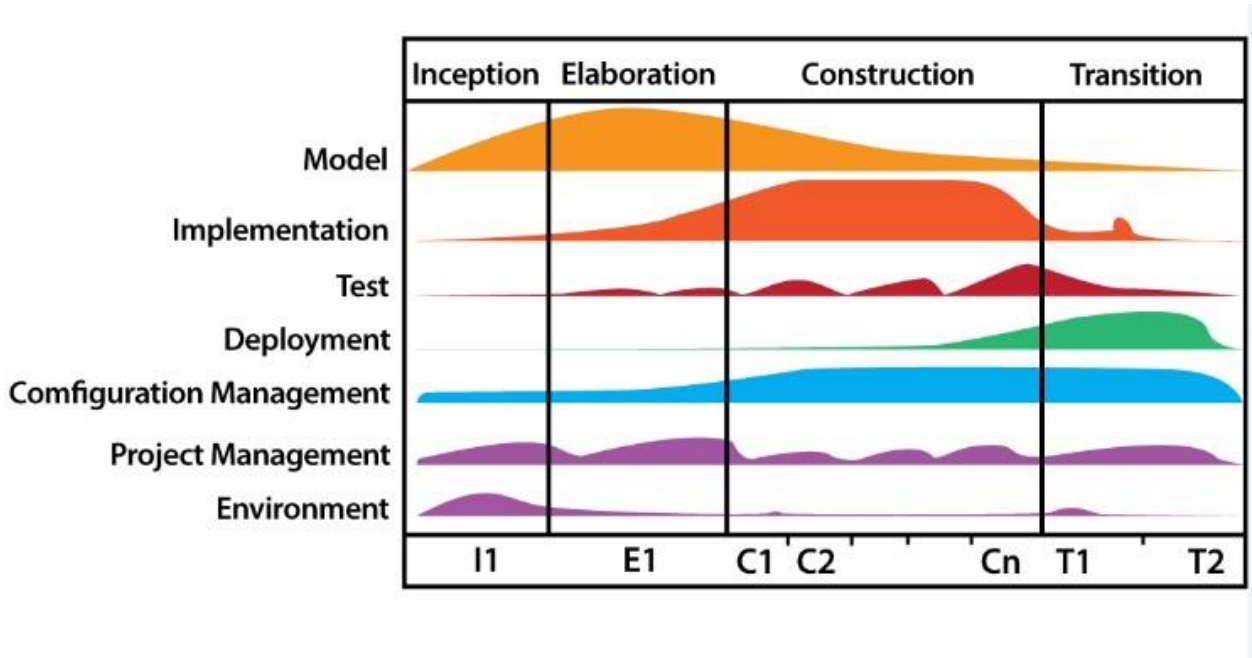



*Figure 6. Rational Unified Process methodology.*

- **Iterative methodology**

The iterative methodology does not require the full specification of the requirement to initialize the software (Ghahrai, 2018). It can be started with specifying and implementing a part of the software and reviewed to identify further requirement. It has the repetition of process multiple times to produce a new version of the software for each cycle of the model. The phases of the cycle include requirement phase, design phase, implementation and test phase and review phase.

**Justification to consider the approach**

- The key success of this methodology is rigorous validation of requirements.
- It verifies each version of the software against those requirements within each cycle of the model.
- It generates quick software during the software life cycle, flexible, less cost for changes, easy testing and debugging, easy to manage risk and each iteration is an easily managed milestone.

**Reasons to not select the methodology**

- The phases are rigid and do not overlap.
- There may be a problem in system architecture as the requirement gathering is not completed in an early stage.

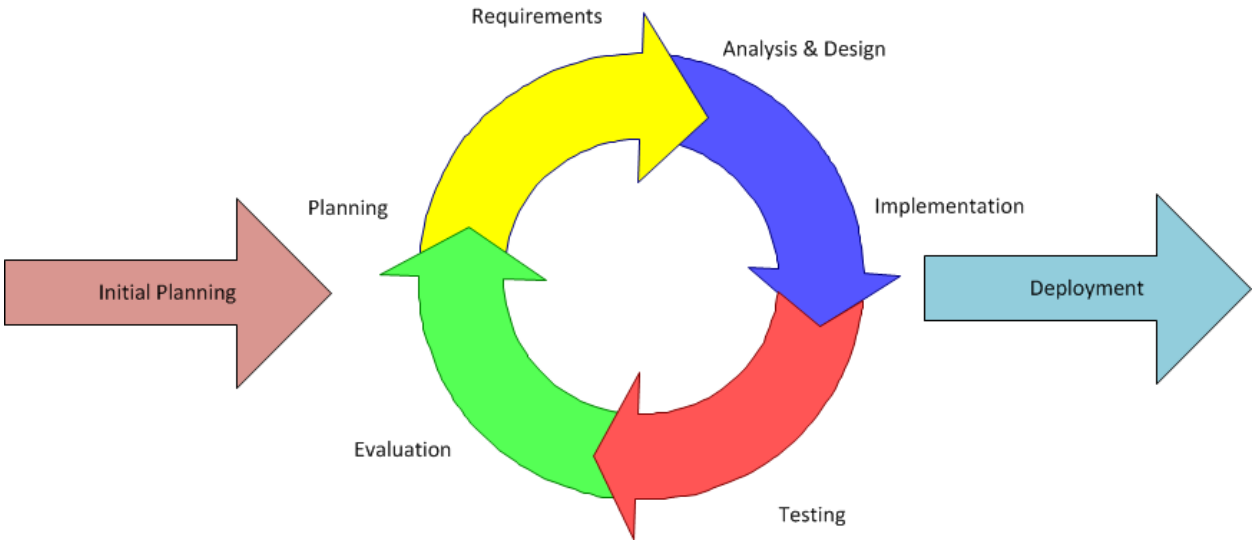



*Figure 7. Iterative methodology.*

- **Waterfall methodology**

The first SDLC Model to be widely used in Software Engineering emphasizes on a logical progression of steps to be taken throughout the Software Development Life Cycle (SDLC) (Tutoial Point, 2019). It illustrates the software development process in a linear sequential flow meaning the development process begins only after the completion of the previous phase. The phases of the cycle are requirement analysis, system design, implementation, testing, deployment, and maintenance.

**Justification to consider the approach**

- It is simple and easy to understand.
- The phases are processed and completed one at a time.
- The process and results are well-documented.

**Reasons to not select the methodology**

- It has a high amount of risks and uncertainty.
- It cannot accommodate changing requirements.
- It is not suitable for complex and object-oriented projects.

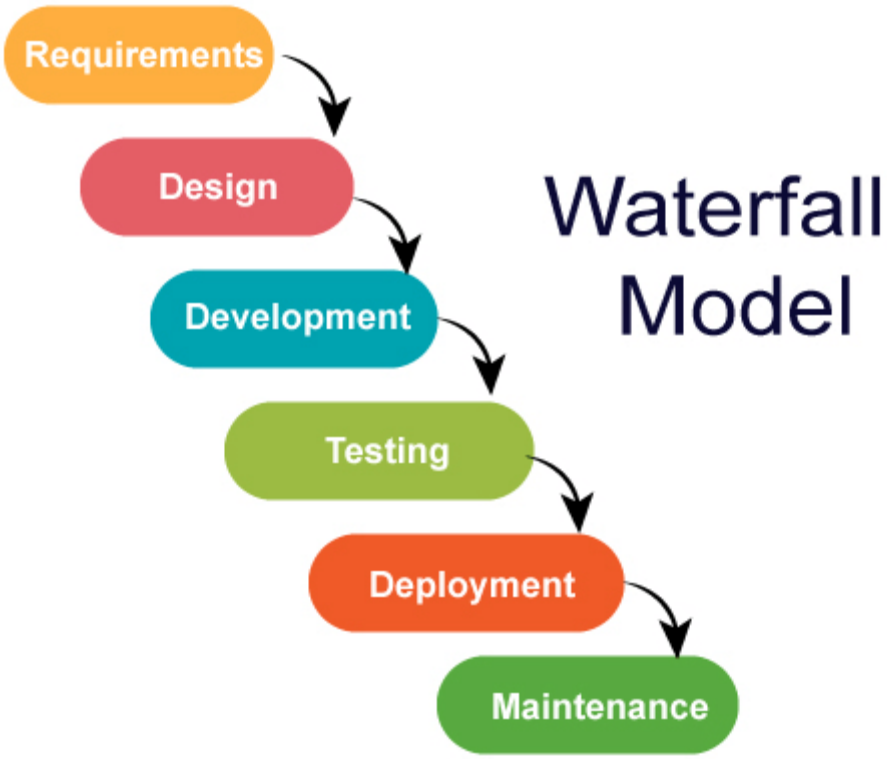



*Figure 8. Waterfall methodology*

## 3.2 Selected Methodology

The methodology chosen for this project is the Iterative methodology. It is an implementation of a software development life cycle (SDLC) that focuses on initial and simplified implementation, then further progresses to gains more complexity and border feature set until the completion of the system. The methodology is based on the concept of incremental development, which is often used liberally and interchangeably, it explains the incremental alterations made during the design and implementation of each new iteration (Powell-Morse, 2016).

The reasons for selecting the Iterative methodology are mentioned below:

- The development of the project is broken different phases where changes can be implemented easily so a model can be made more accurate.
- The sentiment model would be developed as a functional prototype.
- The model will be added more functionality with each iteration.
- The test for each change to the project can be done as each development phase requires testing of the project so potential defects can be easily spotted.
- The development phase acts as a milestone so progress can be easily measured.

## 3.3 Phases of Methodology

The different phases of Iterative methodology are listed below:

- Planning and Requirements

  In this stage, the initial requirements are listed out, related documents are collected, and a plan with a timeline is created for the first iterative cycle.

  The books related to data extraction, text filtration, and sentiment analysis are studied. Any similar projects are searched and studied. After gathering knowledge, a detailed plan is created with a milestone.

- Analysis and Design

  In this stage, the need for the project is finalised along with database models and technical requirements according to the plan. The working architecture, schematic, or algorithm are created as per the requirement of the project.

  The use case, high-level use case, expanded use case, sequence diagram and data flow diagram are created for the project along with the algorithm and flowchart.

- Implementation

  In this stage, the functionality and design required to meet the requirement are developed.

  The code for data extraction, text filtration, text labelling, training model, and classification is developed.

- Testing

  In this stage, the tests are conducted to locate any errors after the development of the code.

  The code is tested through unit testing and system testing.

- Evaluation and Review

  In this stage, the result is compared with the requirements of the project. If the requirements do not meet, the iterative process goes to step one. If the requirements meet, the next cycle is tackled.

The iteration goes for the second iteration for bigrams and third iterative for trigrams.

The reason to select an iterative methodology for this project is the repetition of the filtering for better performance as there are terms as bigrams and trigrams which plays a vital role in the sentiment score as the result of this project.

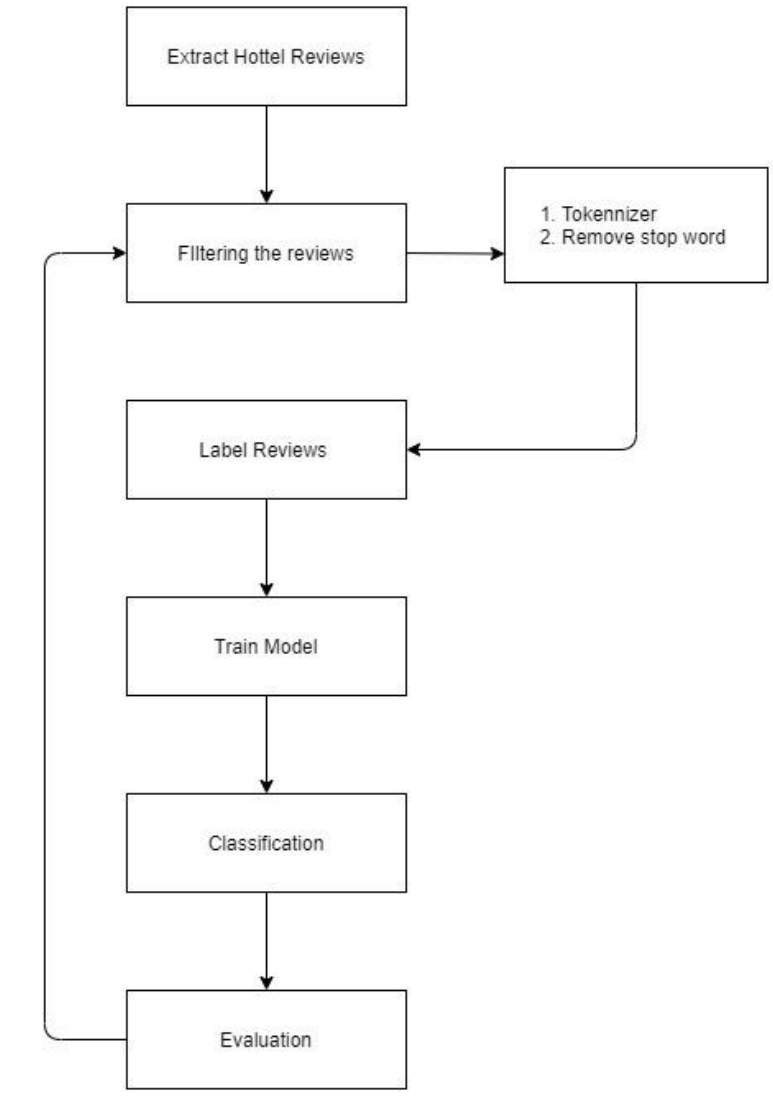



*Figure 9. Phases of methodology.*

## 3.4 Survey Results

### 3.4.1 Pre-Survey Results

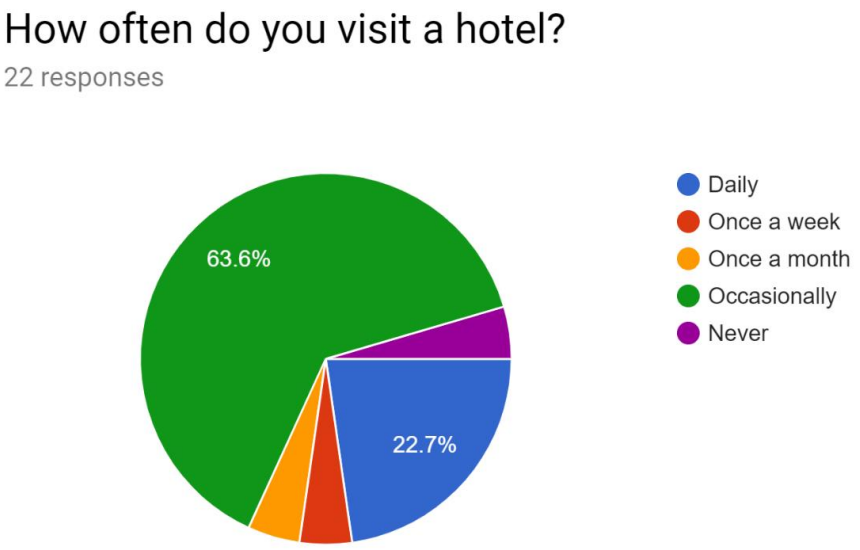



*Figure 10. Pre-survey result. fig(a)*

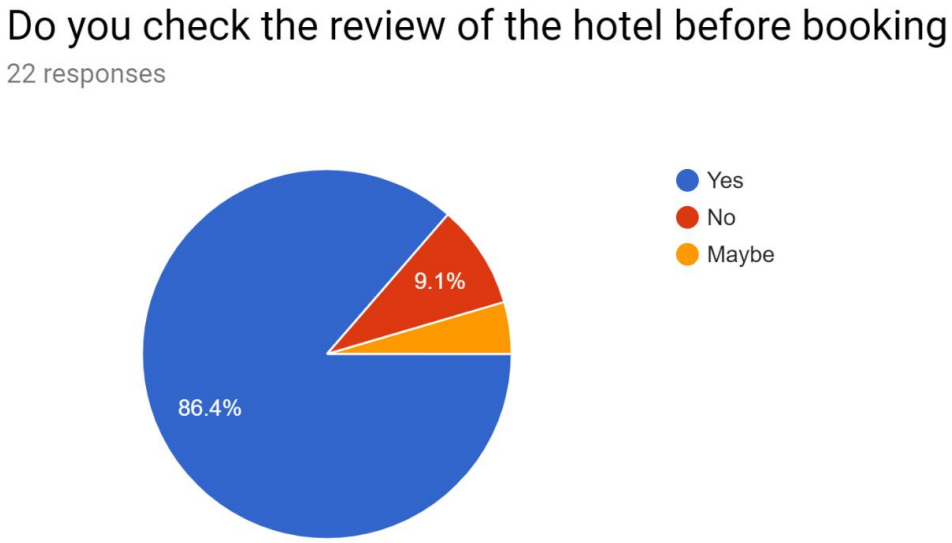



*Figure 11. Pre-survey result. fig(b)*

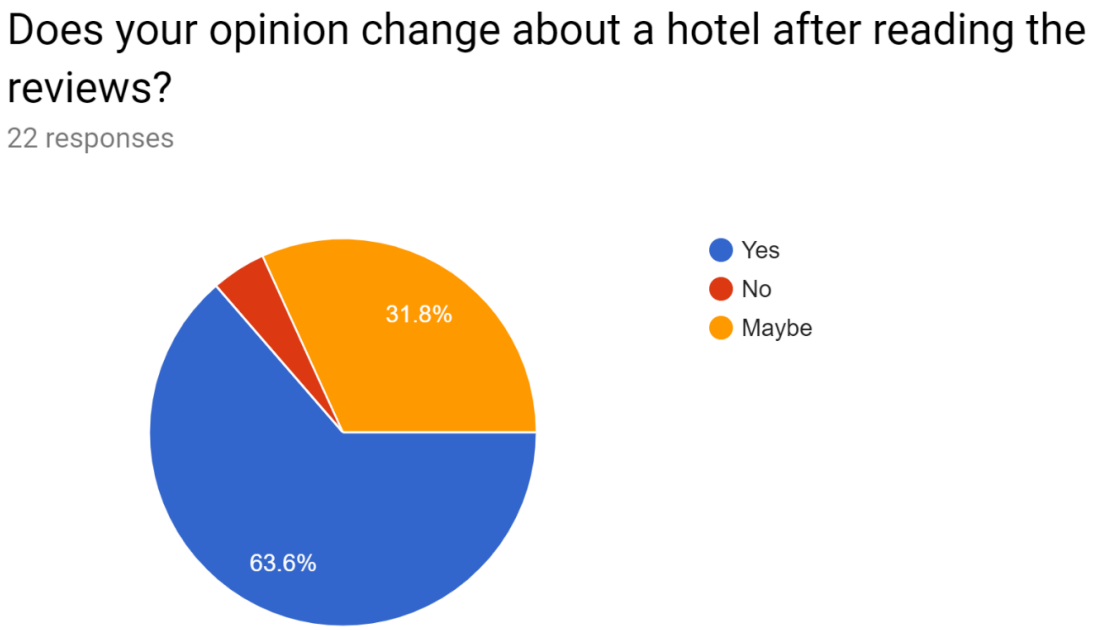



*Figure 12. Pre-survey result. fig(c)*

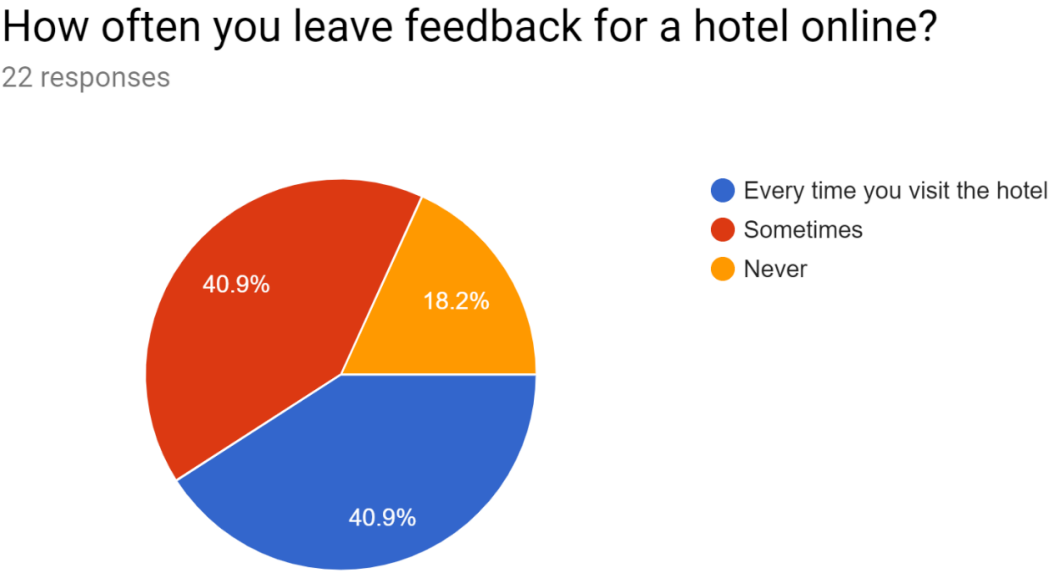



*Figure 13. Pre-survey result. fig(d)*

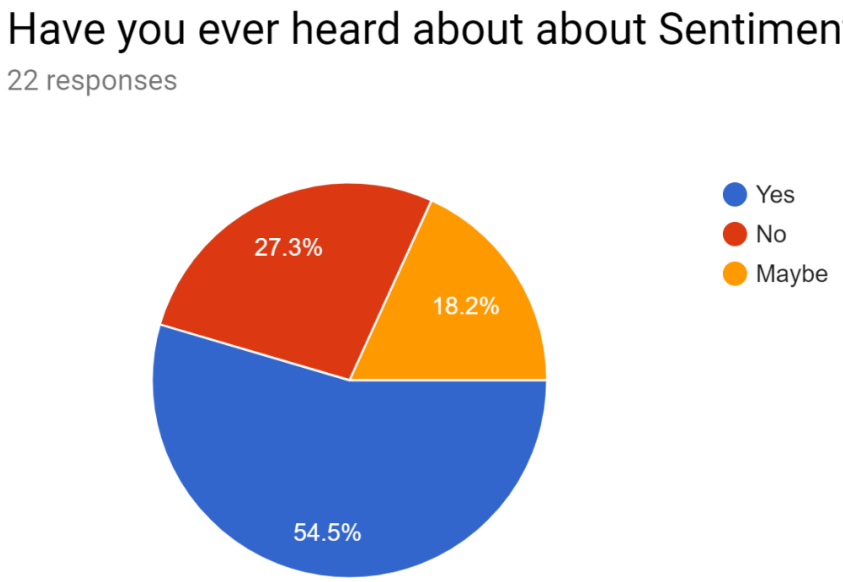



*Figure 14. Pre-survey result. fig(e)*

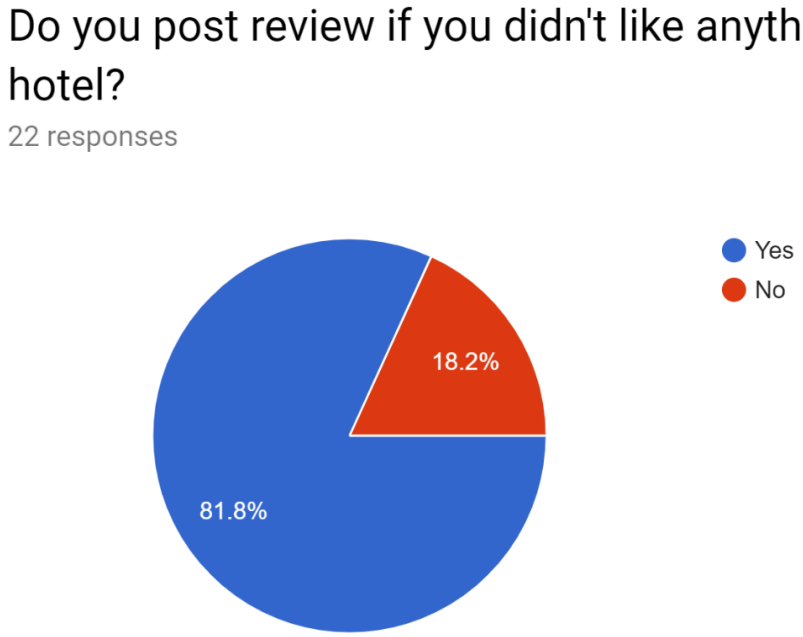



*Figure 15. Pre-survey result. fig(f)*

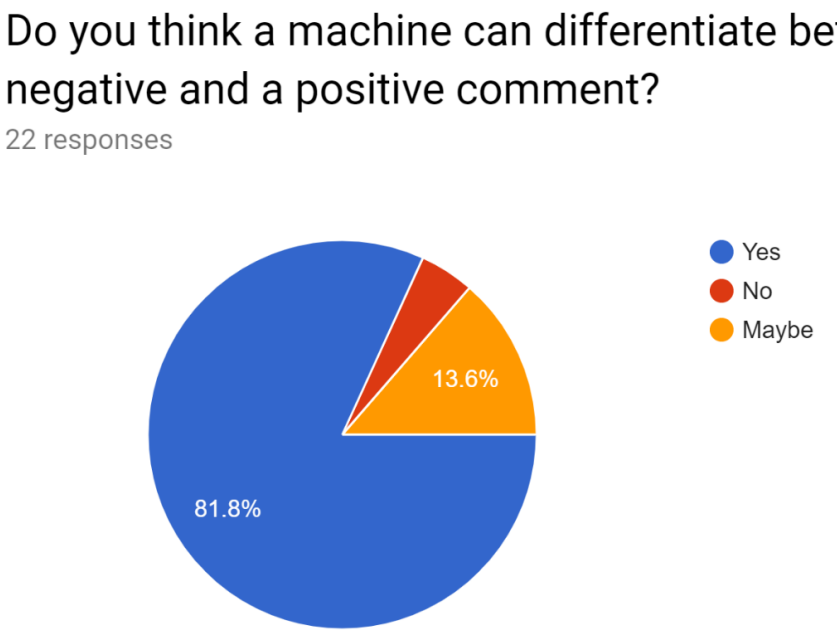



*Figure 16. Pre-survey result. fig(g)*

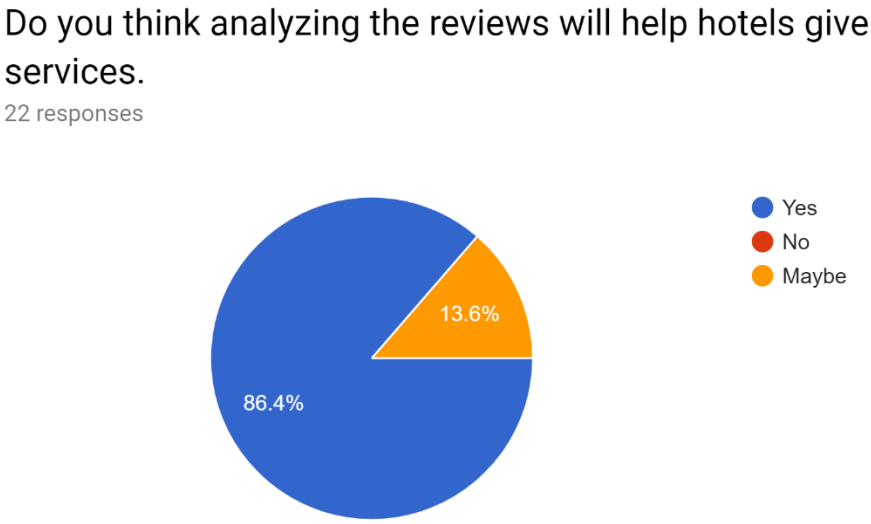



*Figure 17. Pre-survey result. fig(h)*

### 3.1.1 Post-Survey Results

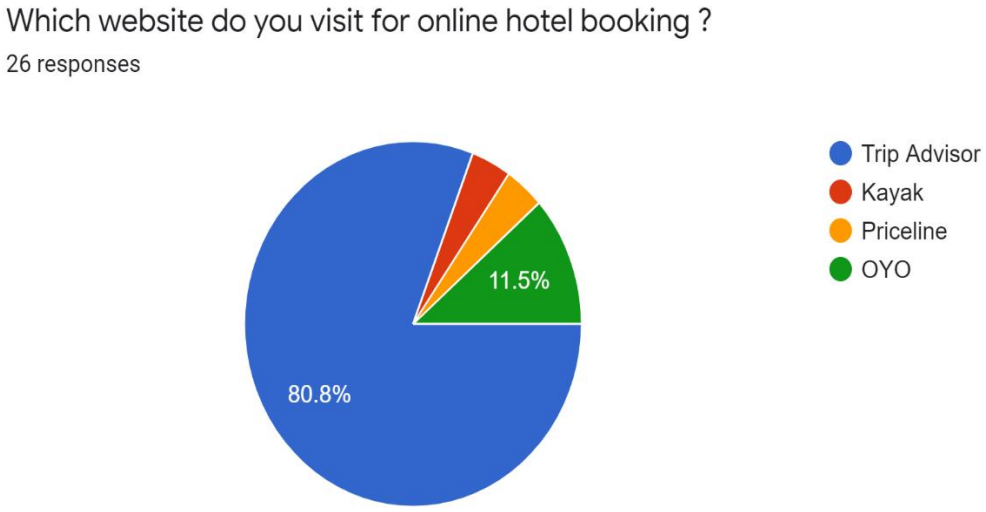



*Figure 18. Post-survey result. fig(a)*

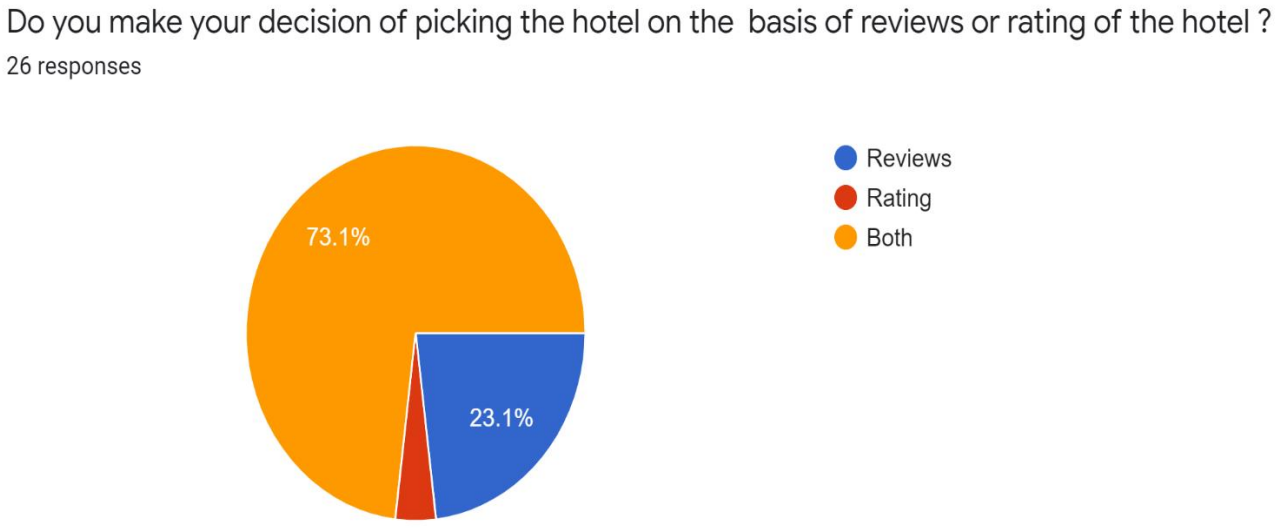



*Figure 19.Post-survey result. fig(b)*

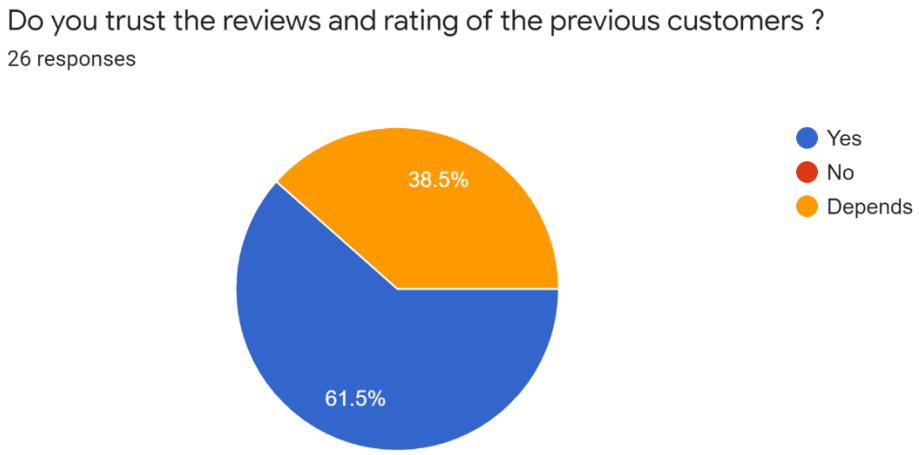



*Figure 20. Post-survey result. fig(c)*

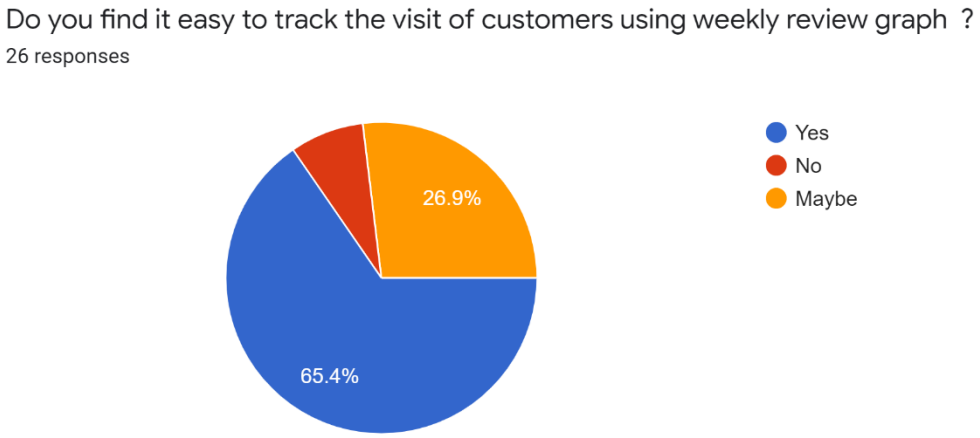



*Figure 21. Post-survey result. fig(d)*

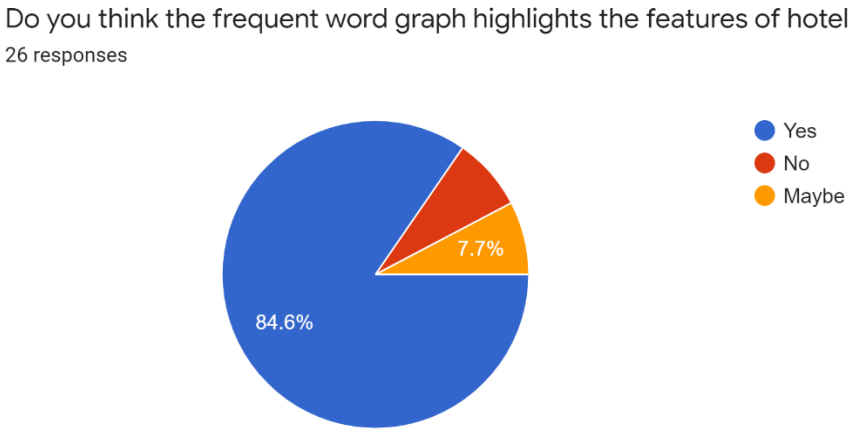



*Figure 22. Post-survey result. fig(e)*

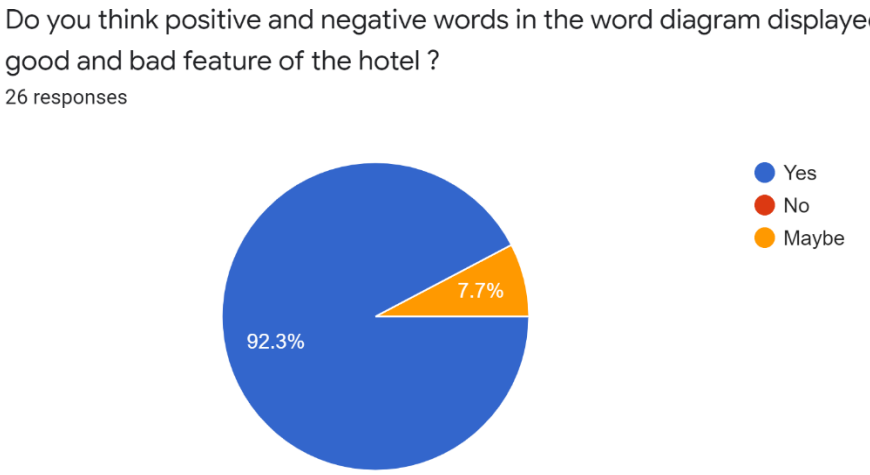



*Figure 23. Post-survey result. fig(f)*

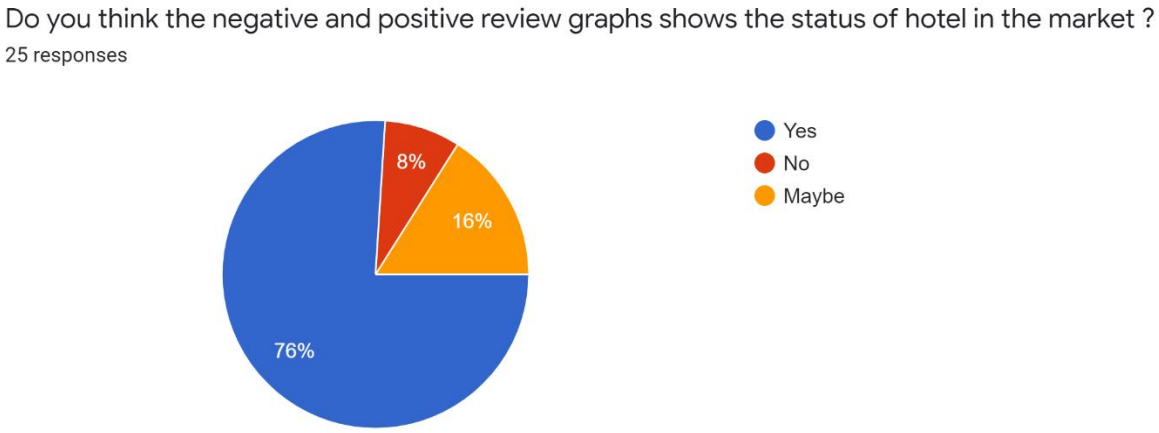



*Figure 24. Post-survey result. fig(g)*

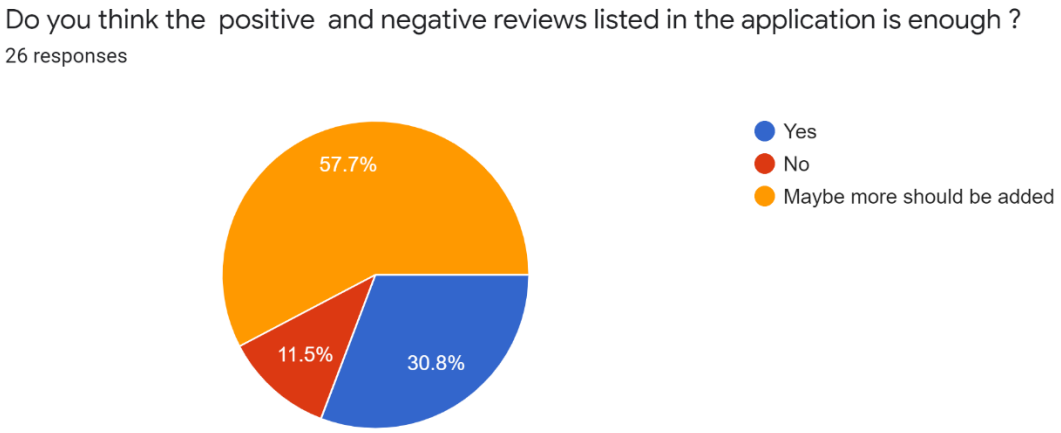



*Figure 25.Post-survey result. fig(h)*

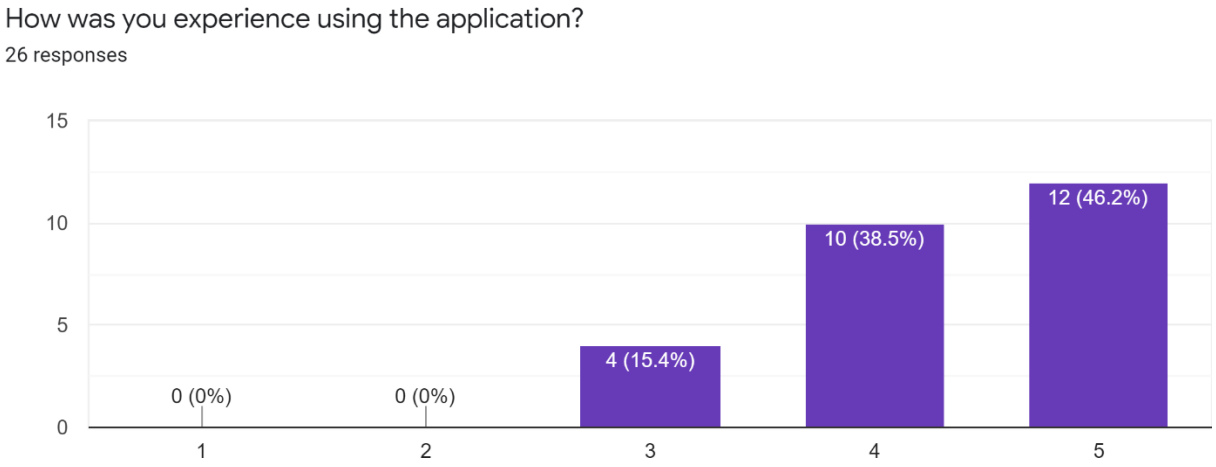



*Figure 26.Post-survey result. fig(i)*

## 3.5 Requirement Analysis

The topic of this project is a newly emerging topic of AI, sentiment analysis. The new concept, Sentiment analysis required a lot of research on the topic, its basics, its types, why it is used and its workflow. After studying the books and article related to Sentiment analysis, ML, and NLP, the concept of sentiment analysis was cleared. The requirements of the project such as data mining, data filtering, feature extraction, ML algorithm, and classification of reviews to present textual and visual resonation of reviews of the hotel which can be downloaded.

The data for sentiment analysis is mined from the trip advisor website. The mined data is filtered using the nltk for tokenizing the reviews, removing stop words and POS tagging. The data is labelled with the help of the rating of the reviews. Labelled reviews are used to train the model. The data is divided in the ratio 80:20, where 80% of the data is used to train the data and 20% of the data is classified by the model. The accuracy of the model is expected to be above 65%. The result of classification is presented to the user through a list of a positive and negative review and graphs.

The sentiment analysis of the reviews will present the list of positive and negative reviews, graphs to display the occurrence of reviews, graph to present the frequent words of the reviews, graph to present the bigram and trigram of the reviews, graph to the occurrence of positive and negative reviews, and the graph to compare the occurrence of positive and negative reviews.

## 3.6 Design

### 3.6.1 Use Case of the system

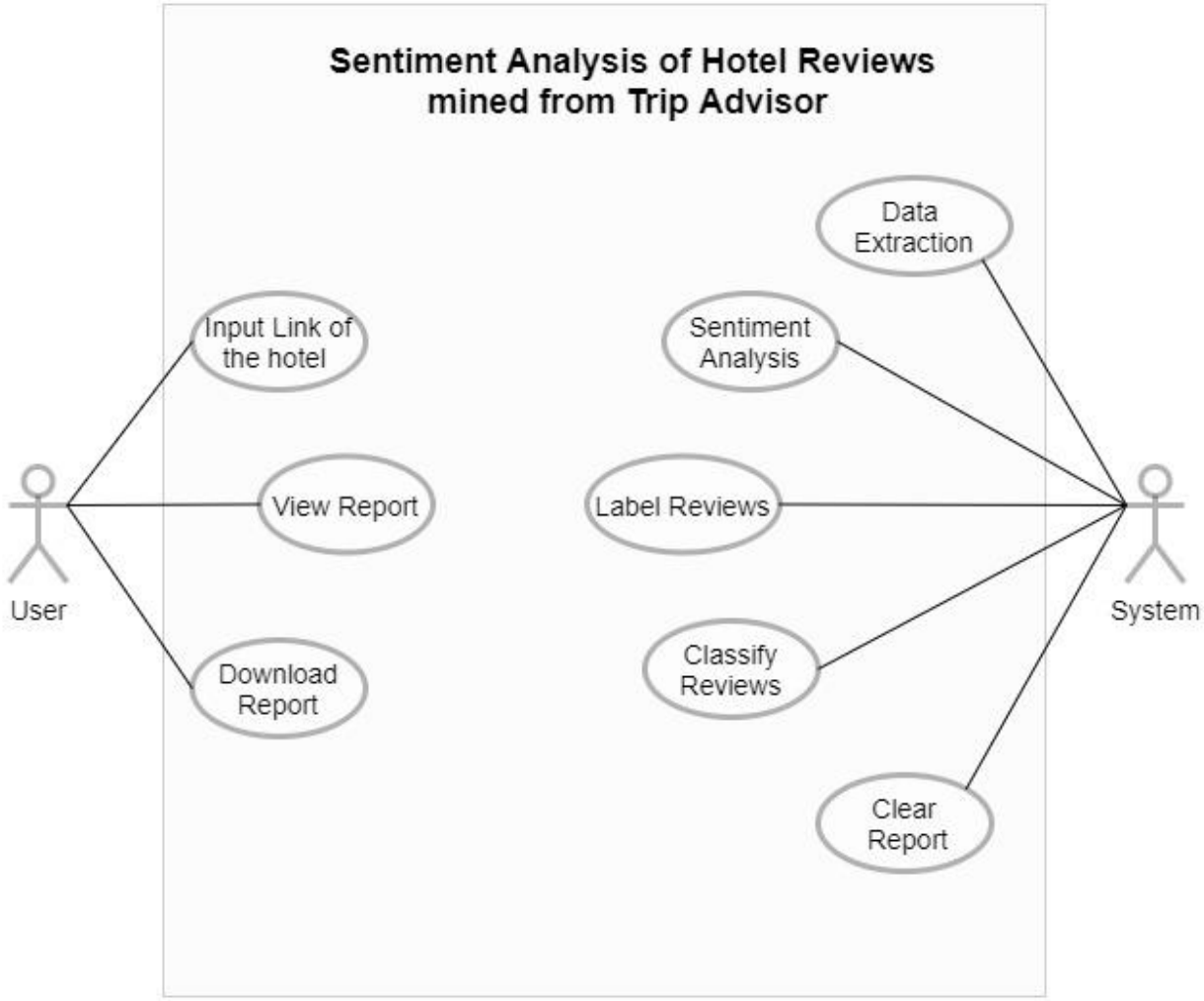



*Figure 27. Use of the system.*

### 3.6.2 High Level Use case

- **Input link**

Use Case : Input Link

Actors : User

Description : The user drops the link of the selected hotel from trip advisor website to view the sentiment analysis report in the text field of the main page.

- **Data Extraction**

Use Case : Data Extraction

Actors : System

Description : The system extracts the reviews, date of reviews and rating of the hotel from the link provided by the user.

- **Filter Reviews**

Use Case : Filter Reviews

Actors : System

Description : The system tokenizes and remove stop words the reviews.

- **Label Reviews**

Use Case : Label Reviews

Actors : System

Description : The system labels the reviews according to their rating.

- **Train Model**

Use Case : Train Model

Actors : System

Description : The system trains the model with the help of labelled data.

- **Classify Reviews**

Use Case : Classify Reviews

Actors : System

Description : The system classifies the reviews using the model.

- **Create Report**

Use Case : Create Report

Actors : System

Description : The system creates a report of classified report for the user.

- **View Report**

Use Case : View Report

Actors : User

Description : The report prepared by the system is viewed by the user.

- **Download Report**

| | | |
|---|---|---|
| Use Case | : | Download Report |
| Actors | : | User |
| Description | : | The user downloads the report if required. |

### 3.6.3 Expanded Use Case

- **Input link**

| | | |
|---|---|---|
| Use Case | : | Input Link |
| Actors | : | User |
| Description | : | The user drops the link of the selected hotel to view the sentiment analysis report in the text field of the main page. |

Typical Course of Events:

| User | System |
|---|---|
| 1. A user copies the link of a hotel from trip advisor. | |
| 2. The copied link is dropped in the text field of the websites. | |
| | 3. The system checks if the link provided by the user belongs to trip advisor or not. |
| | 4. If the link provided by the user exists, the data extraction process is proceeded and if the link does not exist the system asks the user to re-insert the link. |

- **Data Extraction**

| | | |
|---|---|---|
| Use Case | : | Data Extraction |
| Actors | : | System |
| Description | : | The system extracts the reviews of the hotel from the link provided by the user. |

Typical Course of Events:

| User | System |
|---|---|
| | 1. The reviews from the link provided by user is extracted. |
| | 2. The extracted data is stored in CSV file. |

- **Filter Reviews**

| | | |
|---|---|---|
| Use Case | : | Filter Reviews |
| Actors | : | System |
| Description | : | The system tokenizes and remove stop words the reviews. |

Typical Course of Events:

| User | System |
|---|---|
| | 1. The system tokenizes and remove stop words from the reviews |
| | 2. The filtered reviews are stored in a csv file. |

- **Label Reviews**

Use Case : Label Reviews

Actors : System

Description : The system labels the reviews according to their rating.

Typical Course of Events:

| User | System |
| --- | --- |
| | 1. The system labels the filtered reviews according to their rating. |
| | 2. The labelled data are stored in a csv file. |

- **Train Model**

Use Case : Train Model

Actors : System

Description : The system trains the model using the labelled data.

Typical Course of Events:

| User | System |
| --- | --- |
| | 1. The system trains the model using the labelled data. |

- **Classify Reviews**

Use Case : Classify Reviews

Actors : System

Description : The system classifies the reviews using the model.

Typical Course of Events:

| User | System |
| --- | --- |
| | 1. The system classifies the review with the help of the model. |
| | 2. The classified data is stored in CSV file. |

- **Create Report**

Use Case : Create Report

Actors : System

Description : The system creates a report of classified report for the user.

Typical Course of Events:

| User | System |
| --- | --- |
| | 1. The system creates the report using the classified data. |
| | 2. The report is presented to the user. |

- **View Report**

| | | |
|---|---|---|
| Use Case | : | View Report |
| Actors | : | User |
| Description | : | The report prepared after the sentiment analysis is presented to user which is viewed by the user. |

Typical Course of Events:

| User | System |
|---|---|
| 1. The report is viewed by the user. | |

- **Download Report**

| | | |
|---|---|---|
| Use Case | : | Download Report |
| Actors | : | User |
| Description | : | The user downloads the report if required. |

Typical Course of Events:

| User | System |
|---|---|
| | 1. The report by the user if required. |

## 3.7 Implementation

The system of sentiment analysis was developed after the implementation of the collected requirements. The front-end part of the code displayed the result of the sentiment analysis whereas the back-end part of the code extracted the data, filtered the reviews, labelled the reviews, trained the model, and classified the reviews. The implementation of code for the system is listed in the figures below.

- Main page of the system

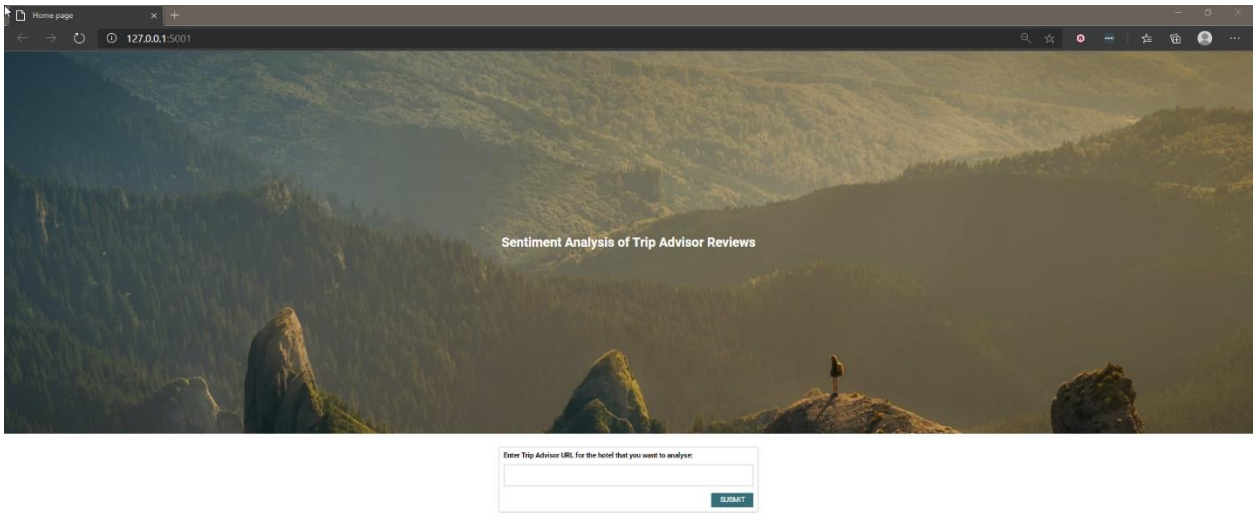



*Figure 28. Main page of the system.*

The above figure is the main page of the system, the main page consists of the text field and a button. The link of the hotel is dropped in the text field and the clicking of the button starts the data extraction process.

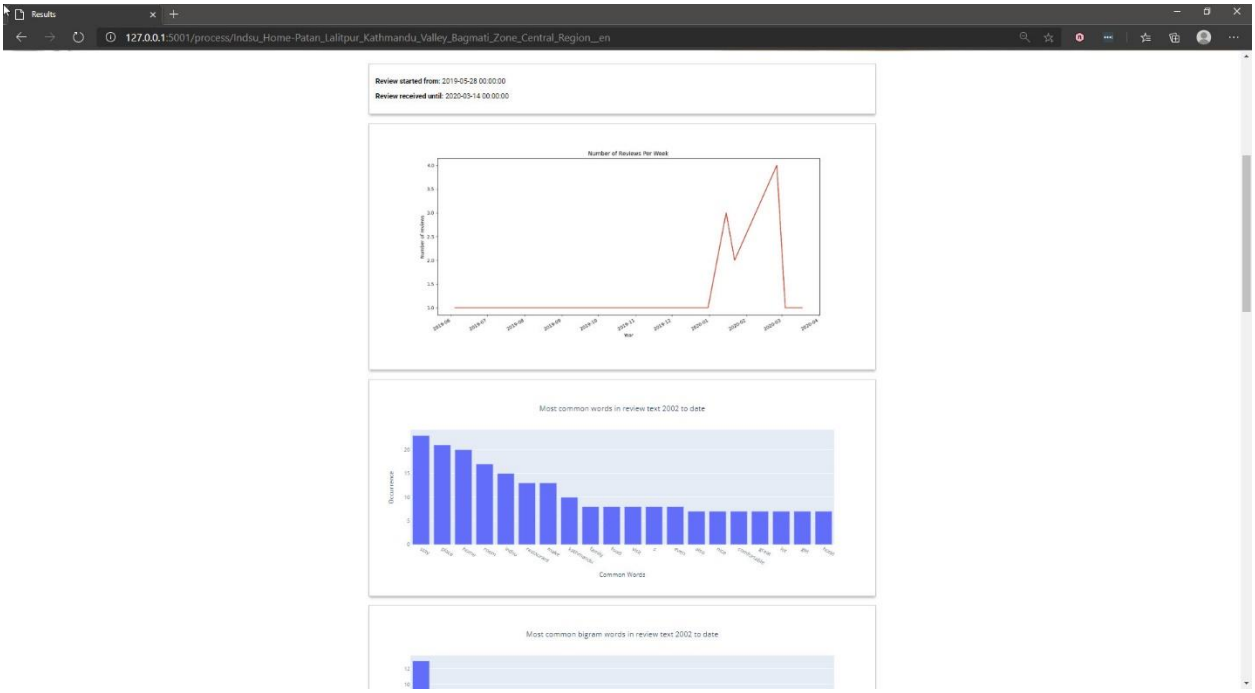



*Figure 29. Result page of the system.*

The above figure is the result page of the system. The result of the analysis is displayed on this page through the graph.

- Back end code

The back end of the system includes the data extraction process, filtration process, labelling process, model training and classifying process and displaying the result in the form of a graph. The back-end code for the above process is placed in different python files. The different python files relate to the help of the main file of the Flask framework, app.py file. The python files containing are presented in the figures below:

  - app.py file

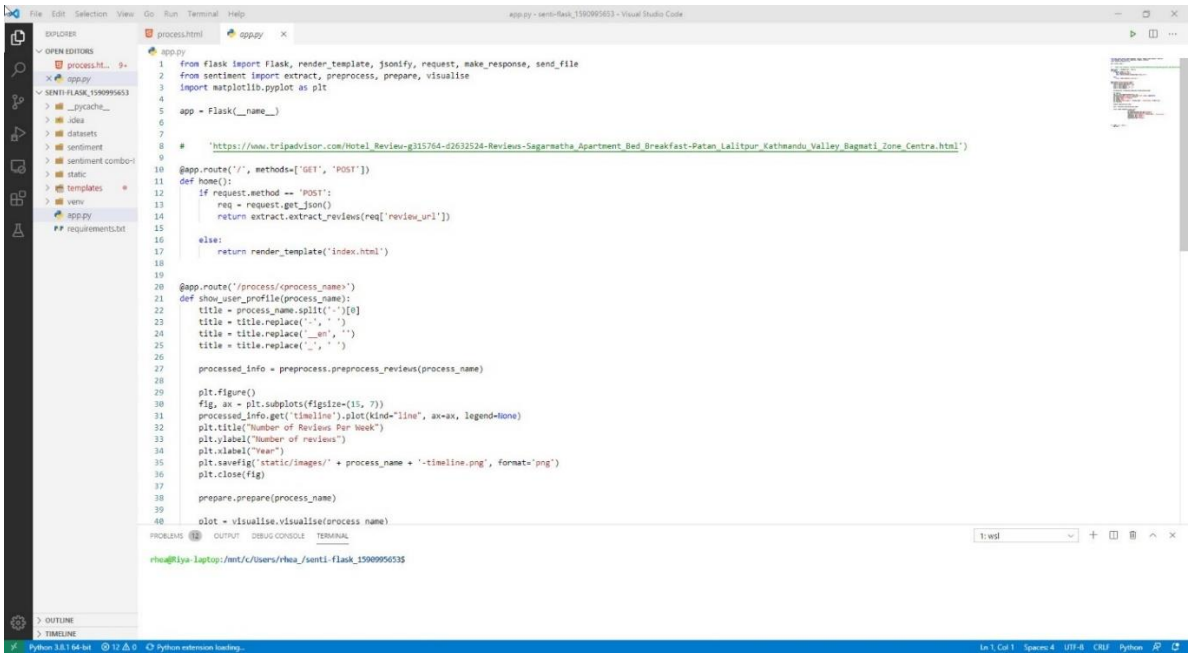
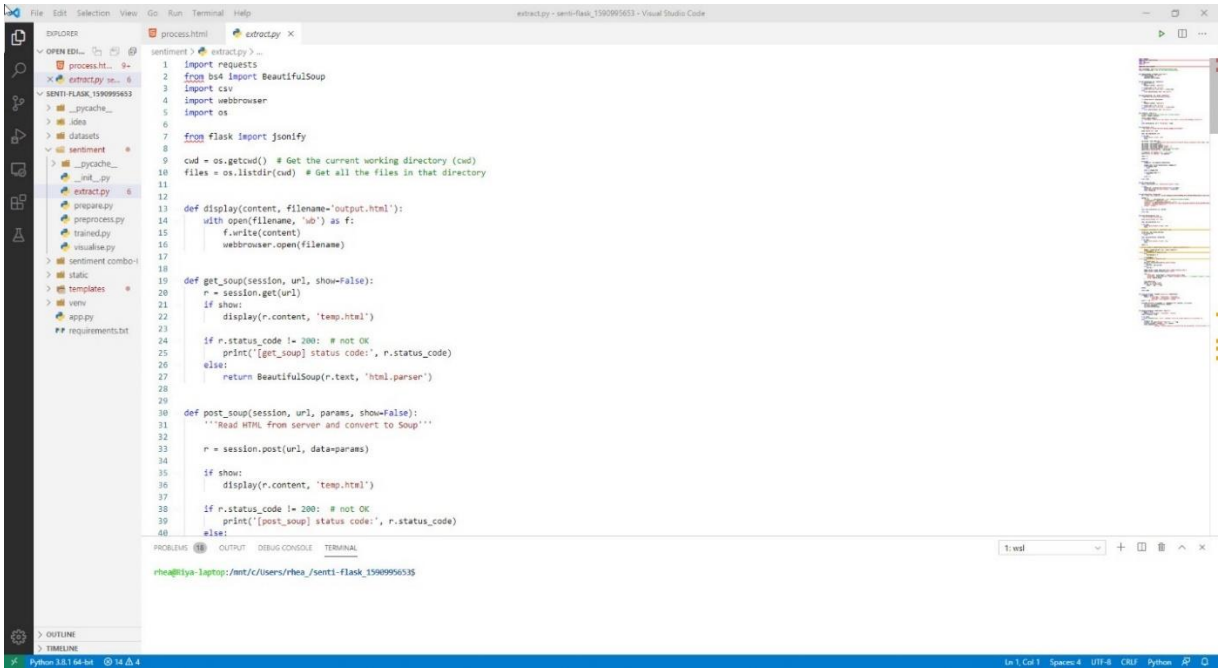



*Figure 30. Back-end code of the system. fig(a)*

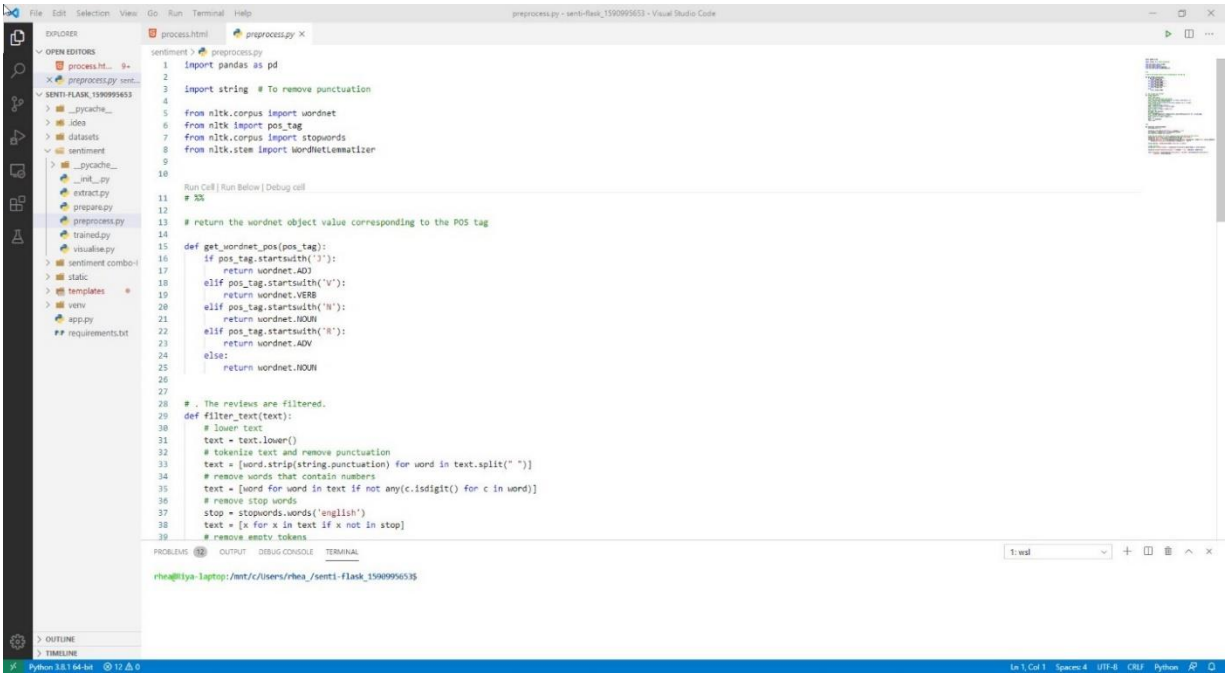
- extract.py



```
import requests
from bs4 import BeautifulSoup
import csv
import webbrowser
import os

from flask import jsonify

cwd = os.getcwd()  # Get the current working directory (cwd)
files = os.listdir(cwd)  # Get all the files in that directory


def display(content, filename='output.html'):
    with open(filename, 'wb') as f:
        f.write(content)
        webbrowser.open(filename)


def get_soup(session, url, show=False):
    r = session.get(url)
    if show:
        display(r.content, 'temp.html')

    if r.status_code != 200:  # not OK
        print('[get_soup] status code:', r.status_code)
    else:
        return BeautifulSoup(r.text, 'html.parser')


def post_soup(session, url, params, show=False):
    '''Read HTML from server and convert to Soup'''

    r = session.post(url, data=params)

    if show:
        display(r.content, 'temp.html')

    if r.status_code != 200:  # not OK
        print('[post_soup] status code:', r.status_code)
    else:
```

*Figure 31. Back-end code of the system. fig(b)*

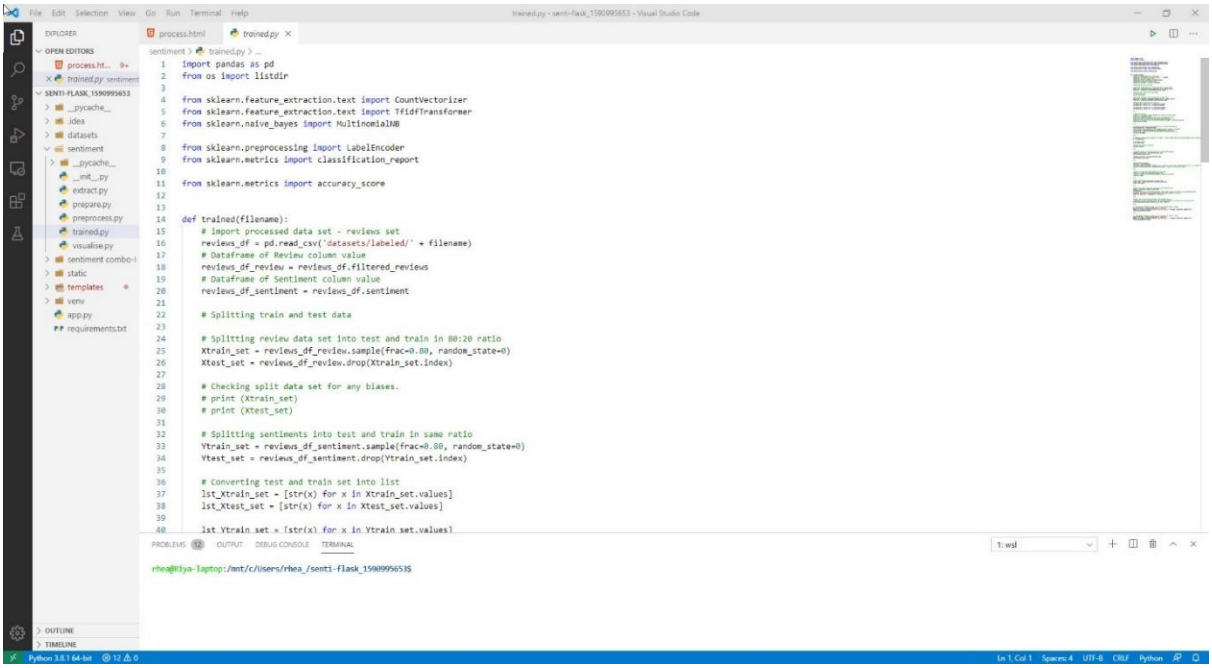
- preprocess.py



```
import pandas as pd

import string  # To remove punctuation

from nltk.corpus import wordnet
from nltk import pos_tag
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer


# %%

# return the wordnet object value corresponding to the POS tag

def get_wordnet_pos(pos_tag):
    if pos_tag.startswith('J'):
        return wordnet.ADJ
    elif pos_tag.startswith('V'):
        return wordnet.VERB
    elif pos_tag.startswith('N'):
        return wordnet.NOUN
    elif pos_tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN


# . The reviews are filtered.
def filter_text(text):
    # lower text
    text = text.lower()
    # tokenize text and remove punctuation
    text = [word.strip(string.punctuation) for word in text.split(" ")]
    # remove words that contain numbers
    text = [word for word in text if not any(c.isdigit() for c in word)]
    # remove stop words
    stop = stopwords.words('english')
    text = [x for x in text if x not in stop]
    # remove empty tokens
```

*Figure 32. Back-end code of the system. fig(c)*

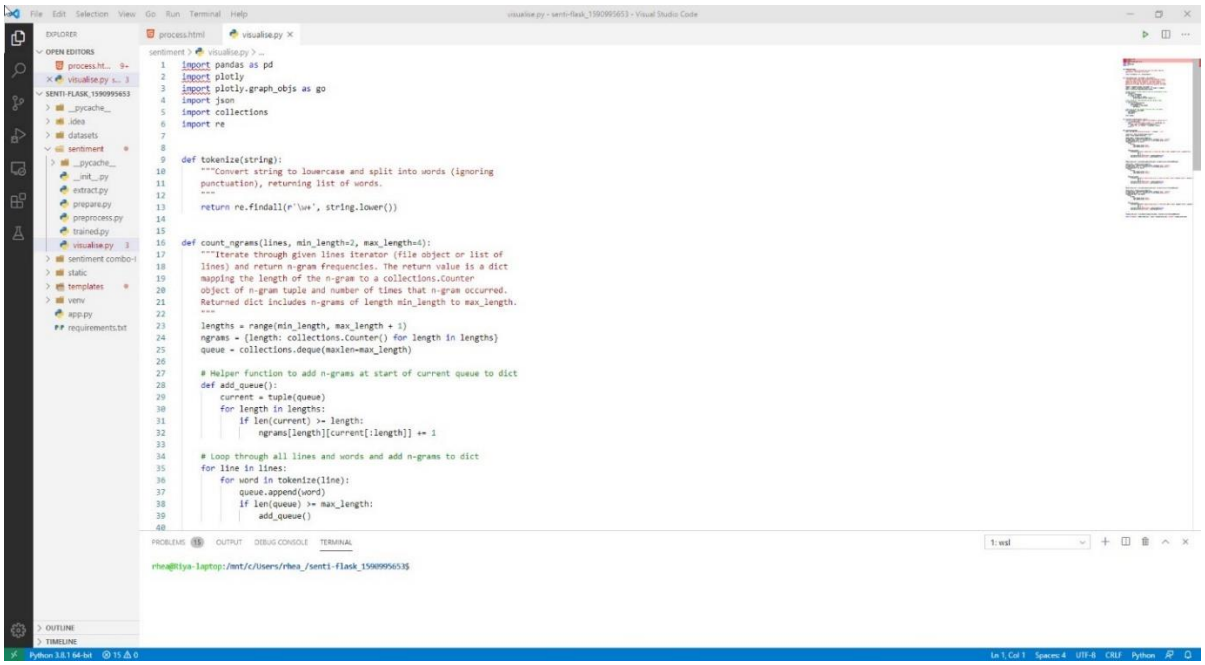
- train.py



```
import pandas as pd
from os import listdir

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB

from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report

from sklearn.metrics import accuracy_score


def trained(filename):
    # Import processed data set - reviews set
    reviews_df = pd.read_csv('datasets/labeled/' + filename)
    # DataFrame of Review column value
    reviews_df_review = reviews_df.filtered_reviews
    # DataFrame of Sentiment column value
    reviews_df_sentiment = reviews_df.sentiment

    # Splitting train and test data

    # Splitting review data set into test and train in 80:20 ratio
    Xtrain_set = reviews_df_review.sample(frac=0.80, random_state=0)
    Xtest_set = reviews_df_review.drop(Xtrain_set.index)

    # Checking split data set for any biases.
    # print (Xtrain_set)
    # print (Xtest_set)

    # Splitting sentiments into test and train in same ratio
    Ytrain_set = reviews_df_sentiment.sample(frac=0.80, random_state=0)
    Ytest_set = reviews_df_sentiment.drop(Ytrain_set.index)

    # Converting test and train set into list
    lst_Xtrain_set = [str(x) for x in Xtrain_set.values]
    lst_Xtest_set = [str(x) for x in Xtest_set.values]

    lst_Ytrain_set = [str(x) for x in Ytrain_set.values]
```

*Figure 33. Back-end code of the system. fig(d)*

- visualize.py



```
import pandas as pd
import plotly
import plotly.graph_objs as go
import json
import collections
import re


def tokenize(string):
    """Convert string to lowercase and split into words (ignoring
    punctuation), returning list of words.
    """
    return re.findall(r'\w+', string.lower())


def count_ngrams(lines, min_length=2, max_length=4):
    """Iterate through given lines iterator (file object or list of
    lines) and return n-gram frequencies. The return value is a dict
    mapping the length of the n-gram to a collections.Counter
    object of n-gram tuple and number of times that n-gram occurred.
    Returned dict includes n-grams of length min_length to max_length.
    """
    lengths = range(min_length, max_length + 1)
    ngrams = {length: collections.Counter() for length in lengths}
    queue = collections.deque(maxlen=max_length)

    # Helper function to add n-grams at start of current queue to dict
    def add_queue():
        current = tuple(queue)
        for length in lengths:
            if len(current) >= length:
                ngrams[length][current[:length]] += 1

    # Loop through all lines and words and add n-grams to dict
    for line in lines:
        for word in tokenize(line):
            queue.append(word)
            if len(queue) >= max_length:
                add_queue()
```

*Figure 34. Back-end code of the system. fig(e)*

# Chapter 4. Testing and Analysis

## 4.1 Test Plan

### 4.1.1 Unit Testing, Test Plan

| S. No. | Unit Test Case |
|---|---|
| **1.** | Check if the raw extracted data is successfully saved in a csv file. |
| **2.** | Check if the csv file containing raw extracted data is opened successfully. |
| **3.** | Check if the graph “No. of reviews per week” is displayed successfully. |
| **4.** | Check if the reviews are filtered successfully. |
| **5.** | Check if the filtered reviews are successfully saved in a csv file. |
| **6.** | Check if the reviews are label as per their rating successfully. |
| **7.** | Check if the labelled reviews are successfully saved in a csv file. |
| **8.** | Check if the positive reviews are listed after classification. |
| **9.** | Check if the negative reviews after classification, display “No Negative Reviews” message if there are no negative reviews. |
| **10.** | Check if the frequent words are displayed in the diagram. |
| **11.** | Check if the occurrence of positive reviews is displayed in a graph. |
| **12.** | Check if there are no negative reviews, display “No Negative Reviews” message if there are no negative reviews |
| **13.** | Check if the graph to compare negative and positive reviews is displayed. |
| **14.** | Check if the graph to compare the rating is displayed. |
| **15.** | Check if the graph of bigrams is displayed. |
| **16.** | Check if the graph of trigrams is displayed. |

*Table 2 Unit Test Plan.*

### 4.1.2 System Testing, Test Plan

| S. No. | System Test Case |
|---|---|
| **1.** | Check if the reviews, reviews rating and date of reviews are successfully extracted from web scrapper. |
| **2.** | Check if the model is trained and gives high accuracy. |
| **3.** | Check if the graphs are displayed in the web page successfully. |
| **4.** | Check if the data of web page can be downloaded successfully. |

*Table 3. System Test Plan.*

## 4.2 Unit Testing

1. Check if the raw extracted data is successfully saved in csv file.

| Objective | The extracted reviews are saved in the csv under the name of their hotel. |
|---|---|
| Action | The extraction code is executed. |
| Excepted Result | The reviews will be displayed, and reviews will be successfully store in a csv file. |
| Actual Result | The reviews were displayed, and reviews was successfully store in a csv file. |
| Conclusion | Test was successful. |

*Table 4. Unit test 1: Check if the raw extracted data is successfully saved in csv file.*

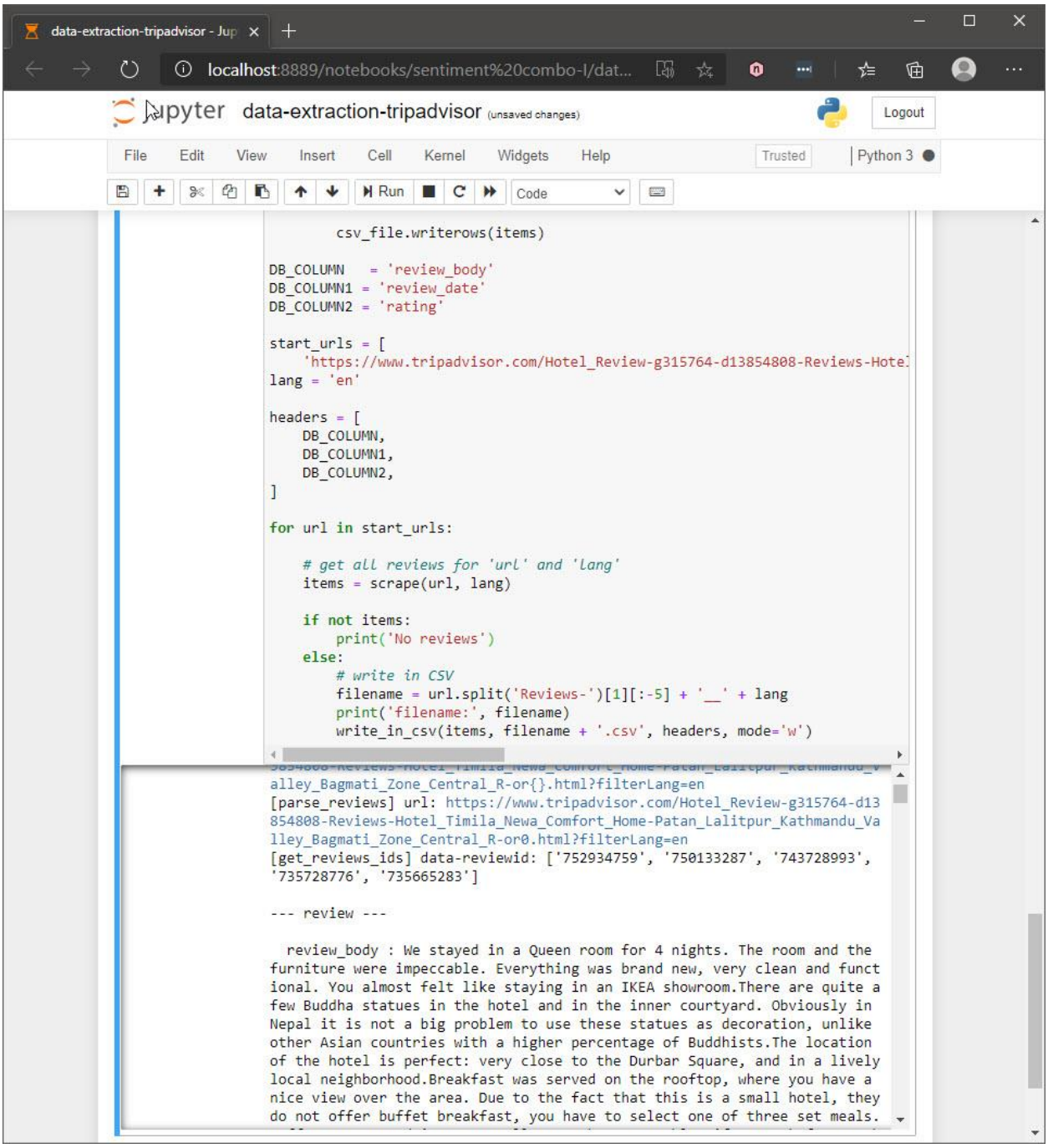



*Figure 35. Unit test 1: Check if the raw extracted data is successfully saved in csv file.*

2. Check if the csv file containing raw extracted data is opened successfully.

| Objective | The csv containing extracted reviews should open. |
|---|---|
| Action | The csv is read and displayed. |
| Excepted Result | The reviews will be displayed. |
| Actual Result | The reviews were displayed. |
| Conclusion | Test was successful. |

*Table 5. Unit test 2: Check if the csv file containing raw extracted data is opened successfully.*

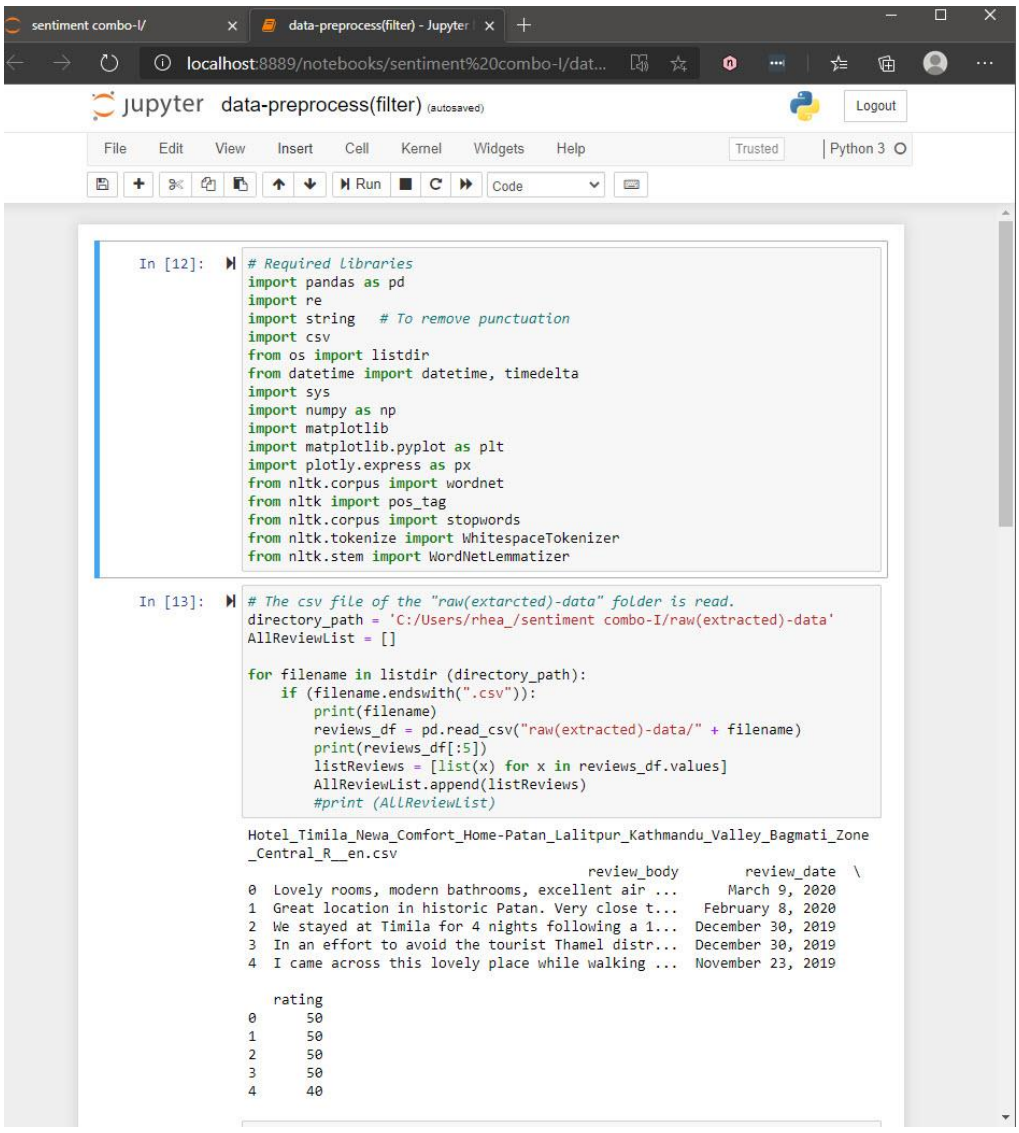



*Figure 36. Unit test 2: Check if the csv file containing raw extracted data is opened successfully.*

3. Check if the graph “No. of reviews per week” is displayed successfully.

| Objective | The graph should display the occurrence reviews according to their review date. |
|---|---|
| Action | The graph is plotted and shown. |
| Excepted Result | The graph will be displayed. |
| Actual Result | The graph was displayed. |
| Conclusion | Test was successful. |

*Table 6. Unit test 3: Check if the graph “No. of reviews per week” is displayed successfully.*

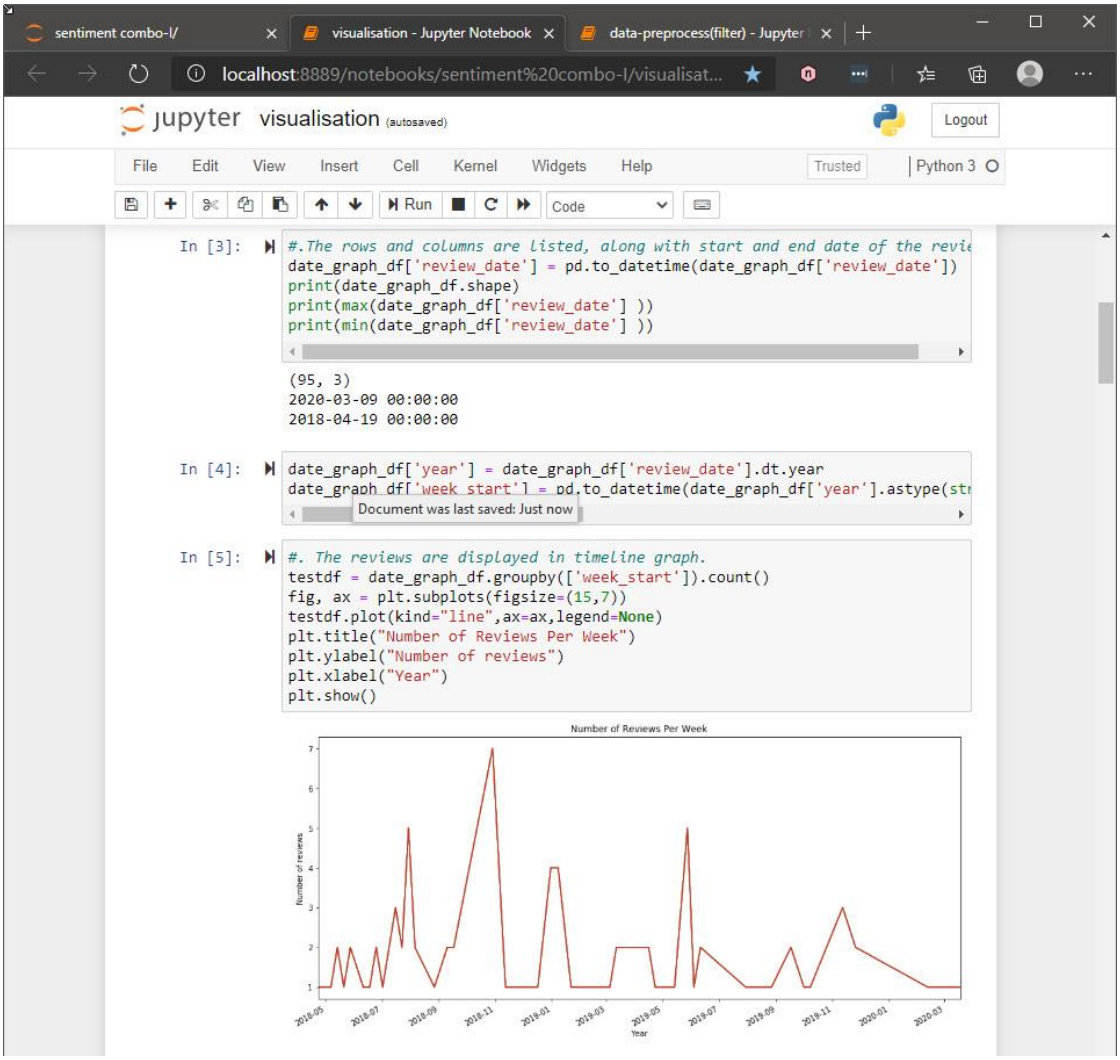
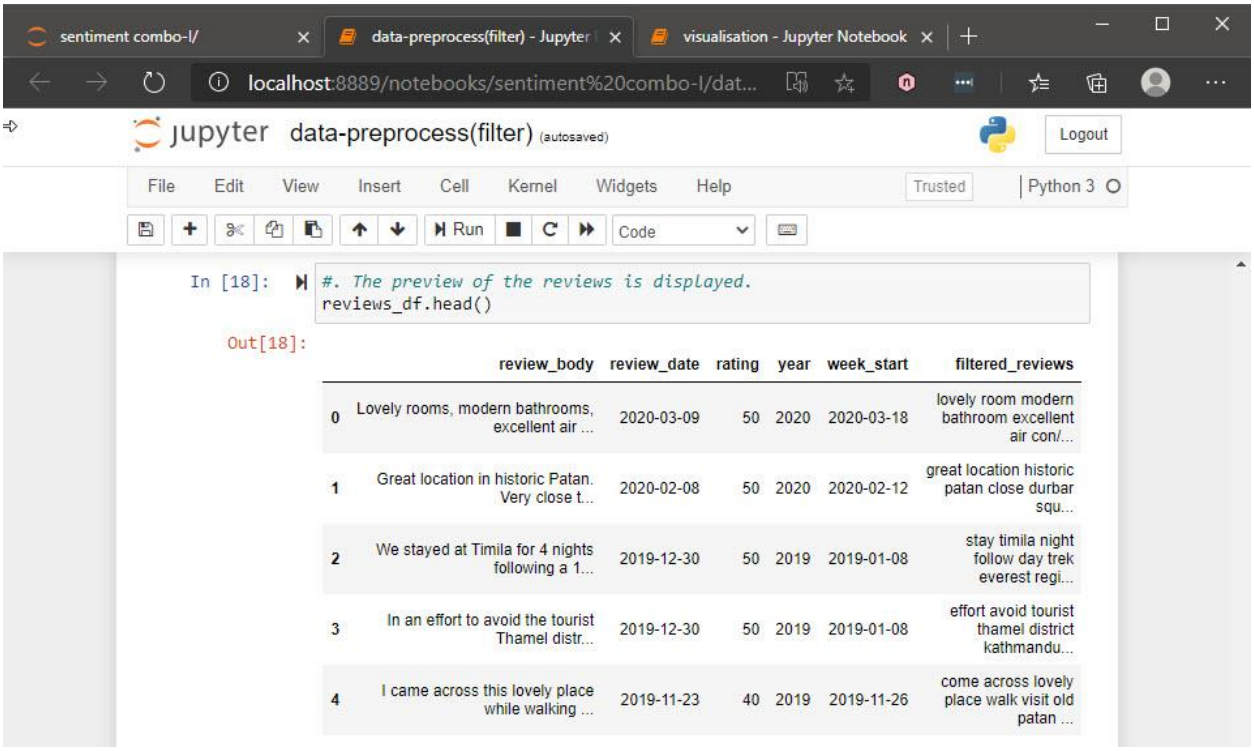



*Figure 37. Unit test 3: Check if the graph “No. of reviews per week” is displayed successfully.*

4. Check if the reviews are filtered successfully.

| Objective | Filter the extracted reviews. |
|---|---|
| Action | The reviews are filtered. |
| Excepted Result | The filtered data will be displayed. |
| Actual Result | The filtered data was displayed. |
| Conclusion | Test was successful. |

*Table 7. Unit test 4: Check if the reviews are filtered successfully.*




| | review_body | review_date | rating | year | week_start | filtered_reviews |
|---|---|---|---|---|---|---|
| 0 | Lovely rooms, modern bathrooms, excellent air ... | 2020-03-09 | 50 | 2020 | 2020-03-18 | lovely room modern bathroom excellent air con/... |
| 1 | Great location in historic Patan. Very close t... | 2020-02-08 | 50 | 2020 | 2020-02-12 | great location historic patan close durbar squ... |
| 2 | We stayed at Timila for 4 nights following a 1... | 2019-12-30 | 50 | 2019 | 2019-01-08 | stay timila night follow day trek everest regi... |
| 3 | In an effort to avoid the tourist Thamel distr... | 2019-12-30 | 50 | 2019 | 2019-01-08 | effort avoid tourist thamel district kathmandu... |
| 4 | I came across this lovely place while walking ... | 2019-11-23 | 40 | 2019 | 2019-11-26 | come across lovely place walk visit old patan ... |

*Figure 38. Unit test 4: Check if the reviews are filtered successfully.*

5. Check if the filtered reviews are successfully saved in a csv file.

| Objective | Filtered reviews are successfully saved in csv file. |
|---|---|
| Action | The filtered reviews should be saved and displayed. |
| Excepted Result | The filtered reviews will be saved and displayed |
| Actual Result | The filtered reviews were saved and displayed |
| Conclusion | Test was successful. |

*Table 8. Unit test 5: Check if the filtered reviews are successfully saved in a csv file.*

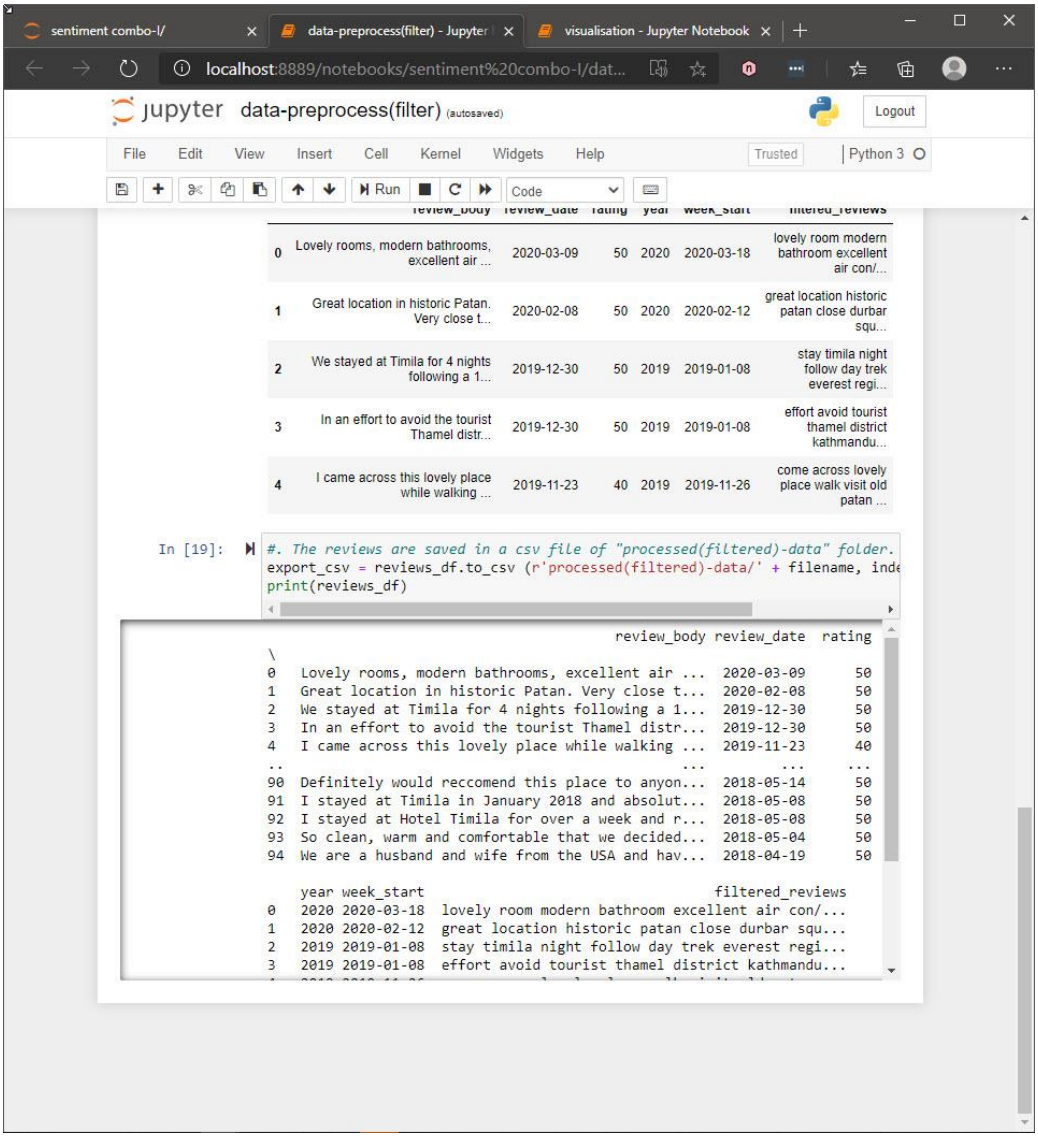



*Figure 39. Unit test 5: Check if the filtered reviews are successfully saved in a csv file.*

6. Check if the reviews are label as per their rating successfully.

| Objective | Label the reviews according to their rating. |
|---|---|
| Action | The reviews are labelled according to their ratings. |
| Excepted Result | The reviews will be labelled. |
| Actual Result | The reviews were labelled. |
| Conclusion | Test was successful. |

*Table 9. Unit test 6: Check if the reviews are label as per their rating successfully.*

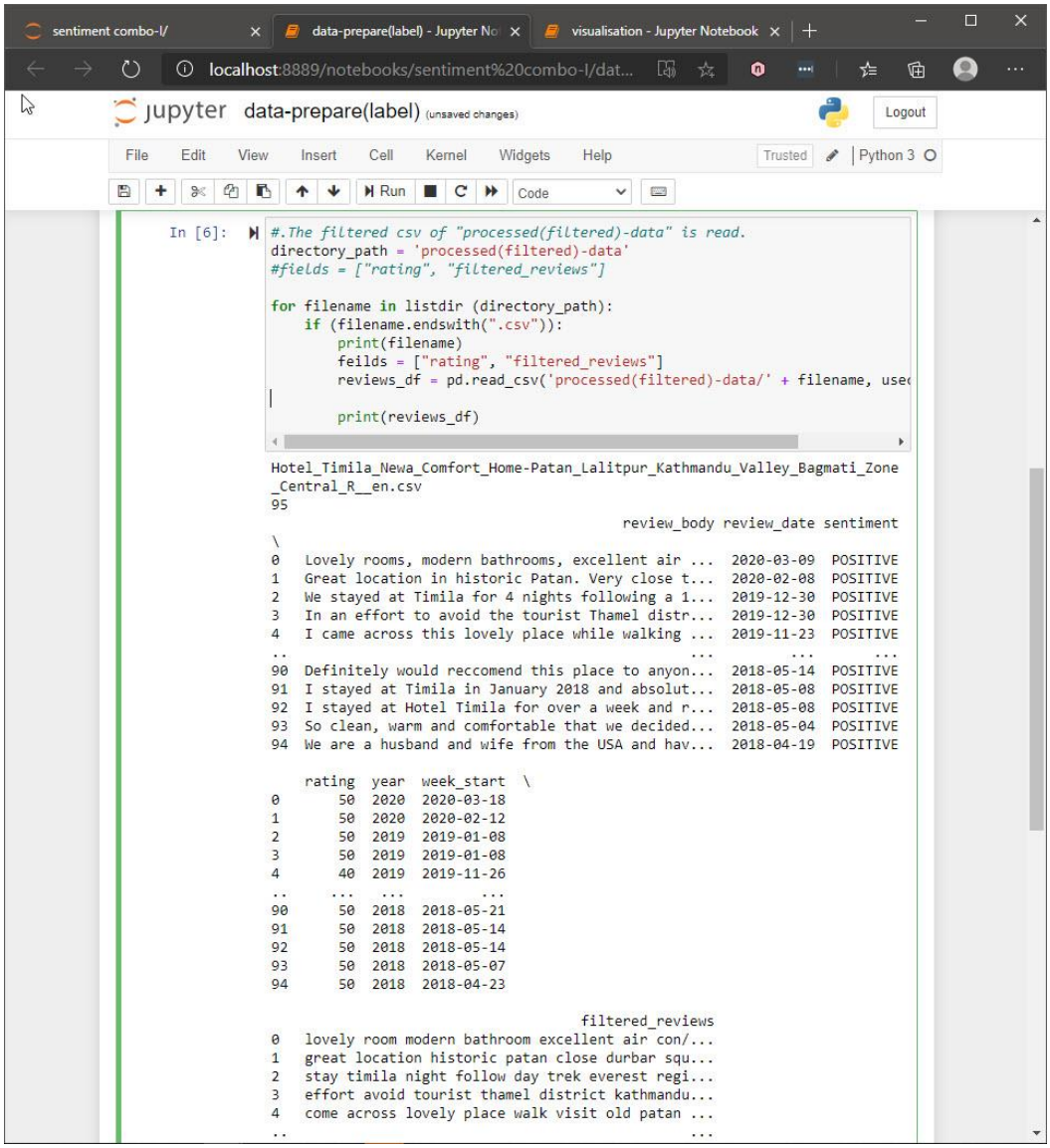



*Figure 40. Unit test 6: Check if the reviews are label as per their rating successfully.*

7. Check if the labelled reviews are successfully saved in a csv file.

| Objective | Labelled reviews should be saved in csv file. |
|---|---|
| Action | The labelled reviews are saved in a csv file. |
| Excepted Result | The labelled reviews will be saved in csv file and displayed. |
| Actual Result | The labelled reviews were saved in csv file and displayed. |
| Conclusion | Test was successful. |

*Table 10. Unit test 7: Check if the labelled reviews are successfully saved in a csv file.*

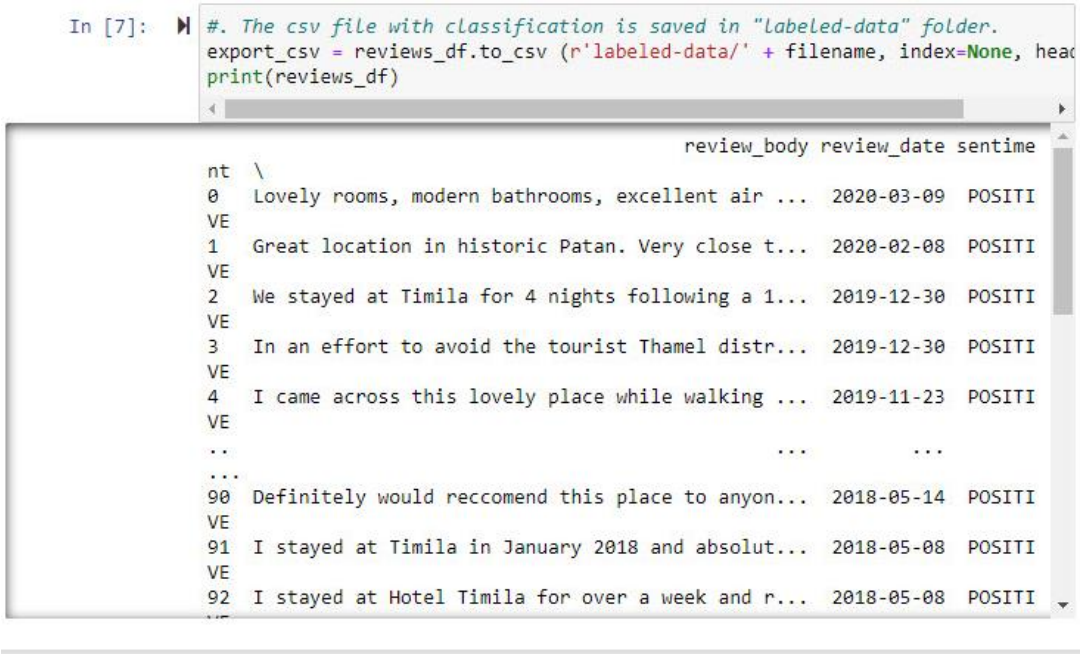



*Figure 41. Unit test 7: Check if the labelled reviews are successfully saved in a csv file.*

8. Check if the positive reviews are listed after classification.

| Objective | Positive reviews should be listed. |
|---|---|
| Action | The positive reviews are listed after classification. |
| Excepted Result | The positive reviews will be listed. |
| Actual Result | The positive reviews were listed. |
| Conclusion | Test was successful. |

*Table 11. Unit test 8: Check if the positive reviews are listed after classification.*

```
positive_review = sentiment_df.apply(lambda x: True if x['sentiment'] == "POSITIVE" else False, axis=1)
count_positive= len(positive_review[positive_review == True].index)
print(count_positive)

if count_positive >= 1:
    #. The positive reviews are saved in a csv file of the "result/" folder.
    positive_reviews = sentiment_df.sentiment == "POSITIVE"
    print(sentiment_df[positive_reviews])
else:
    print("No Positive Review")
```

```
95
                                    review_body sentiment  rating  \
0   Lovely rooms, modern bathrooms, excellent air ...  POSITIVE      50
1   Great location in historic Patan. Very close t...  POSITIVE      50
2   We stayed at Timila for 4 nights following a 1...  POSITIVE      50
3   In an effort to avoid the tourist Thamel distr...  POSITIVE      50
4   I came across this lovely place while walking ...  POSITIVE      40
..                                                ...       ...     ...
90  Definitely would reccomend this place to anyon...  POSITIVE      50
91  I stayed at Timila in January 2018 and absolut...  POSITIVE      50
92  I stayed at Hotel Timila for over a week and r...  POSITIVE      50
93  So clean, warm and comfortable that we decided...  POSITIVE      50
94  We are a husband and wife from the USA and hav...  POSITIVE      50

    week_start                                filtered_reviews
0   2020-03-18  lovely room modern bathroom excellent air con/...
1   2020-02-12  great location historic patan close durbar squ...
2   2019-01-08  stay timila night follow day trek everest regi...
3   2019-01-08  effort avoid tourist thamel district kathmandu...
4   2019-11-26  come across lovely place walk visit old patan ...
..         ...                                               ...
90  2018-05-21  definitely would reccomend place anyone plan s...
91  2018-05-14  stay timila january absolutely love stay room ...
92  2018-05-14  stayed hotel timila week return another night ...
93  2018-05-07  clean warm comfortable decide extend stay grat...
94  2018-04-23  husband wife usa travel extensively numerous c...

[95 rows x 5 columns]
```

*Figure 42. Unit test 8: Check if the positive reviews are listed after classification.*

9. Check if the negative reviews after classification, display “No Negative Reviews” message if there are no negative reviews.

| Objective | Display message “No Negative Reviews” if there are no negative reviews. |
|---|---|
| Action | The negative reviews are listed after classification. |
| Excepted Result | The negative reviews will be listed, if not found the message “No Negative Reviews” will be displayed. |
| Actual Result | The negative reviews were not found, so the message “No Negative Reviews” was displayed. |
| Conclusion | Test was successful. |

*Table 12. Unit test 9: Check if the negative reviews after classification, display “No Negative Reviews”.*

```
#.the graph for negative review per date
negative_review = sentiment_df.apply(lambda x: True if x['sentiment'] == "NEGATIVE" else False, axis=1)
count_negative = len(negative_review[negative_review == True].index)
print(count_negative)

if count_negative >= 1:
    #. The negative reviews are saved in a csv file of the "result/" folder.
    negative_reviews = sentiment_df.sentiment == "NEGATIVE"
    print(sentiment_df[negative_reviews])
else:
    print("No Negative Review")
```

```
0
No Negative Review
```

*Figure 43. Unit test 9: Check if the negative reviews after classification, display “No Negative Reviews”.*

10. Check if the frequent words are displayed in the diagram.

| Objective | Frequent words should be displayed in a graph. |
|---|---|
| Action | The frequent words are displayed in a graph. |
| Excepted Result | The frequent words will be displayed in the graph. |
| Actual Result | The frequent words were displayed in the graph. |
| Conclusion | Test was successful. |

*Table 13. Unit test 10: Check if the frequent words are displayed in the diagram.*

```
# print_most_frequent(ngrams,20)

common_words = ngrams[1].most_common(20)
df_common_words = pd.DataFrame(common_words, columns=['word', 'count'])
df_common_words.word  =  [' '.join(x) for x in df_common_words['word']]
fig = px.bar(df_common_words, x='word', y='count',
             labels={'word':'Common words','count': 'Occurence'},
             title={'text':"Most common words in review text 2002 to date",'x
        'x':0.5,})
fig.show()
```

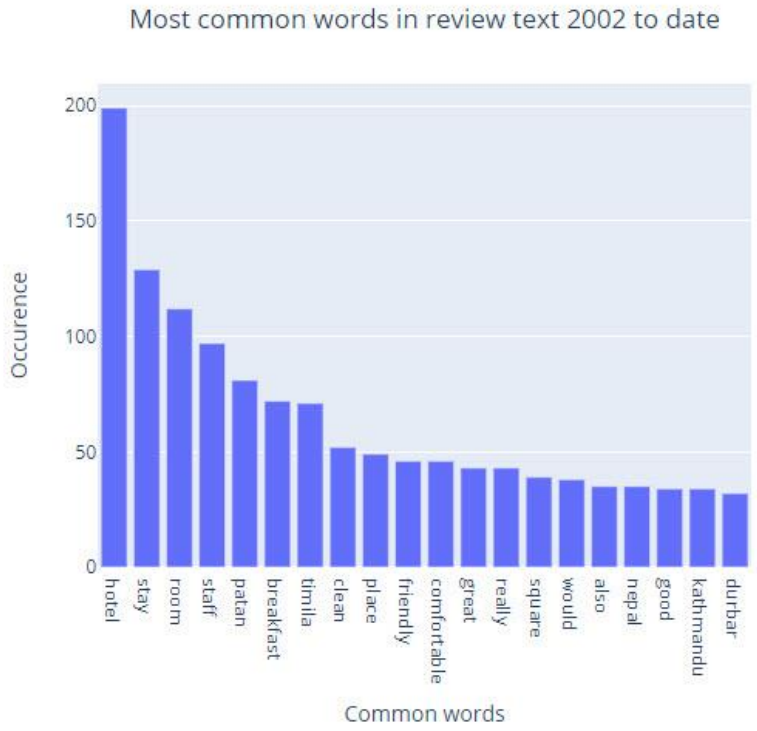



*Figure 44. Unit test 10: Check if the frequent words are displayed in the diagram.*

11. Check if the occurrence of positive reviews is displayed in a graph.

| Objective | Display the positive review occurrence in a graph. |
|---|---|
| Action | The graph is plotted to represent the occurrence of positive reviews. |
| Excepted Result | The graph will be displayed. |
| Actual Result | The graph was displayed. |
| Conclusion | Test was successful. |

*Table 14. Unit test 11: Check if the positive reviews are displayed in a graph.*

```
positive_review = sentiment_df.apply(lambda x: True if x['sentiment'] == "POSITIVE" else False, axis=1)
count_positive= len(positive_review[positive_review == True].index)
print(count_positive)

if count_positive >= 1:
    #. The reviews are displayed in timeline graph.
    testdf = sentiment_df[sentiment_df.sentiment == "POSITIVE"].groupby(['week_start']).count()
    fig, ax = plt.subplots(figsize=(15,7))
    testdf.plot(kind="line",ax=ax,legend=None)
    plt.title("Number of Reviews Per Week")
    plt.ylabel("Number of reviews")
    plt.xlabel("Year")
    plt.show()
else:
    print("No Positive Review")
```

```
95
```

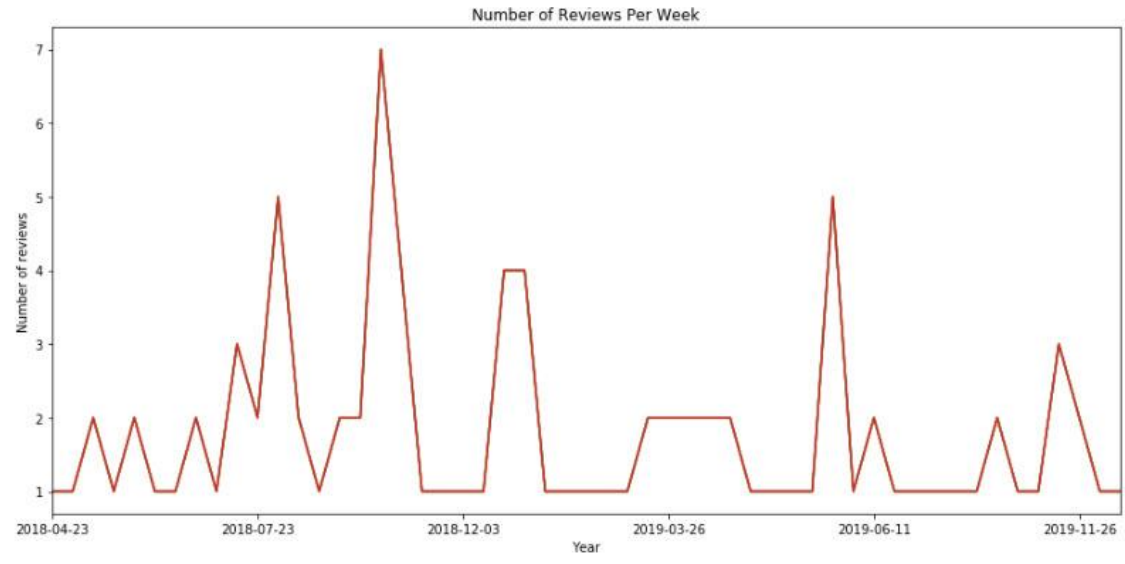



*Figure 45. Unit test 11: Check if the positive reviews are displayed in a graph.*

12. Check if there are no negative reviews, display “No Negative Reviews” message if there are no negative reviews.

| Objective | Display message “No Negative Reviews” if there is no negative reviews. |
|---|---|
| Action | The graph is plotted to represent the occurrence of negative reviews. |
| Excepted Result | The message “No Negative Reviews” will be displayed if there are no negative reviews. |
| Actual Result | The message “No Negative Reviews” was displayed when there are no negative reviews. |
| Conclusion | Test was successful. |

*Table 15. Unit test 12: Check if there are no negative reviews, display “No Negative Reviews”.*

```
#.the graph for negative review per date
negative_review = sentiment_df.apply(lambda x: True if x['sentiment'] == "NEG
count_negative = len(negative_review[negative_review == True].index)
print(count_negative)

if count_negative >= 1:
    #. The reviews are dsiplayed in timeline graph.
    testdf = sentiment_df[sentiment_df.sentiment == "NEGATIVE"].groupby(['wee
    fig, ax = plt.subplots(figsize=(15,7))
    testdf.plot(kind="line",ax=ax,legend=None)
    plt.title("Number of Reviews Per Week")
    plt.ylabel("Number of reviews")
    plt.xlabel("Year")
    plt.show()

else:
    print("No Negative Review")
```

```
0
No Negative Review
```

*Figure 46. Unit test 12: Check if there are no negative reviews, display “No Negative Reviews”.*

13. Check if the graph to compare occurrence of negative and positive reviews is displayed.

| Objective | Present bar graph to compare occurrence of positive and negative reviews. |
|---|---|
| Action | The graph is plotted to compare the occurrence of positive and negative reviews. |
| Excepted Result | The graph will be displayed. |
| Actual Result | The graph was displayed. |
| Conclusion | Test was successful. |

*Table 16. Unit test 13: Check if the graph to compare occurrence of negative and positive reviews is displayed.*

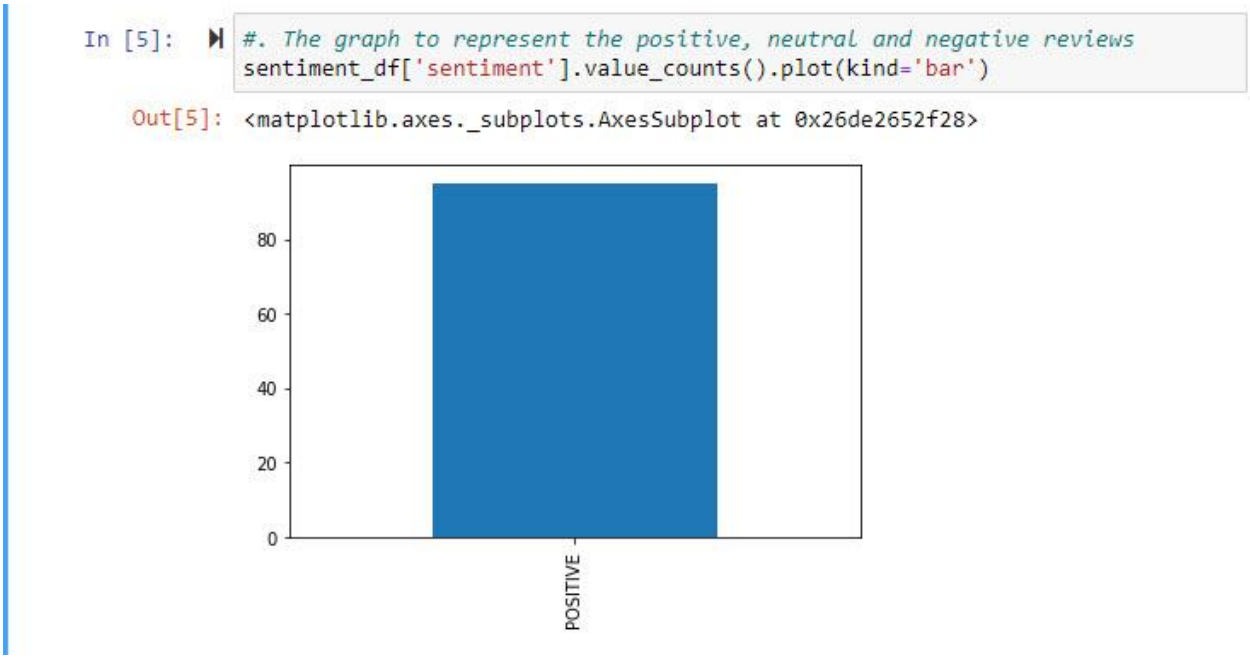



*Figure 47. Unit test 13: Check if the graph to compare occurrence of negative and positive reviews is displayed*

14. Check if the graph to compare the rating is displayed.

| Objective | Present bar graph to compare rating of reviews. |
|---|---|
| Action | The graph is plotted to compare rating of reviews. |
| Excepted Result | The graph will be displayed. |
| Actual Result | The graph was displayed. |
| Conclusion | Test was successful. |

*Table 17. Unit test 14: Check if the graph to compare the rating is displayed.*

```
#The rating summary graph
sentiment_df['rating'].value_counts().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f72cf13320>
```

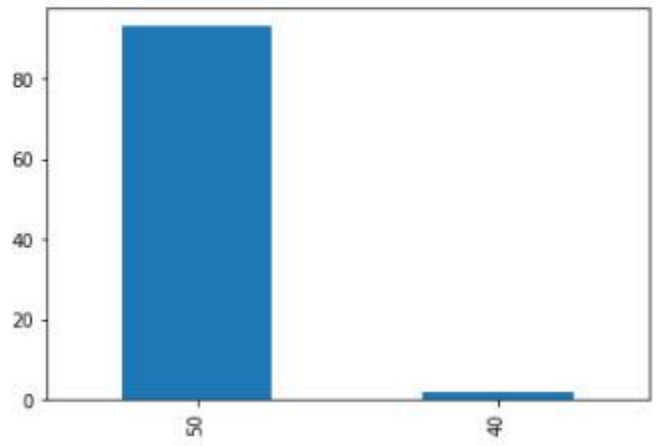



*Figure 48. Unit test 14: Check if the graph to compare the rating is displayed.*

15. Check if the graph of bigrams is displayed.

| Objective | Display the bigrams in a graph. |
|---|---|
| Action | The graph is plotted to display the bigrams. |
| Excepted Result | The graph will be displayed. |
| Actual Result | The graph was displayed. |
| Conclusion | Test was successful. |

*Table 18. Unit test 15: Check if the graph of bigrams is displayed.*

```
. The bigram words are displayed in bar graph.
ommon_words = ngrams[2].most_common(20)
f_common_words = pd.DataFrame(common_words, columns=['word', 'count'])
f_common_words.word  =  [' '.join(x) for x in df_common_words['word']]
ig = px.bar(df_common_words, x='word', y='count',
            labels={'word':'Common words','count': 'Occurence'},
            title={'text':"Most common biagram words in review text 2002 to date",'xanchor':'center','yanchor':'top','y':0.0,
       'x':0.5,})
ig.show()
```

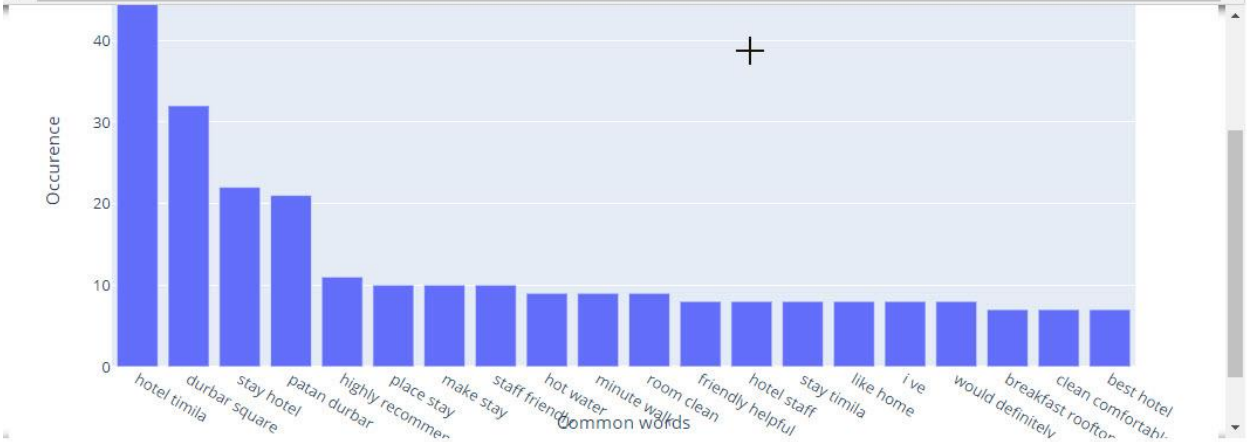



*Figure 49. Unit test 15: Check if the graph of bigrams is displayed.*

16. Check if the graph of trigrams is displayed.

| Objective | Display the trigrams in a graph. |
|---|---|
| Action | The graph is plotted to display the trigrams. |
| Excepted Result | The graph will be displayed. |
| Actual Result | The graph was displayed. |
| Conclusion | Test was successful. |

*Table 19. Unit test 16: Check if the graph of trigrams is displayed.*

```
#. The trigram words are displayed in bar graph.
common_words = ngrams[3].most_common(20)
df_common_words = pd.DataFrame(common_words, columns=['word', 'count'])
df_common_words.word  =  [' '.join(x) for x in df_common_words['word']]
fig = px.bar(df_common_words, x='word', y='count',
             labels={'word':'Common words','count': 'Occurence'},
             title={'text':"Most common trigram words in review text 2002 to date",'xanchor':'center','yanchor':'top','y':0.9
        'x':0.5,})
fig.show()
```


*Figure 50. Unit test 16: Check if the graph of trigrams is displayed.*

## 4.3 System Testing

1. Check if the reviews, reviews rating and date of reviews are successfully extracted from web scrapper.

| Objective | Extract data from the link provided by the user. |
|---|---|
| Action | The data is extracted from the given link and “Submit” button was clicked. |
| Excepted Result | The message “Success! Reviews is extracted from the provided URL. Click here to check analysed data” will be displayed. |
| Actual Result | The message “Success! Reviews is extracted from the provided URL. Click here to check analysed data” was displayed. |
| Conclusion | Test was successful. |

*Table 20. System test 1: Check if the reviews, reviews rating and date of reviews are successfully extracted from web scrapper.*

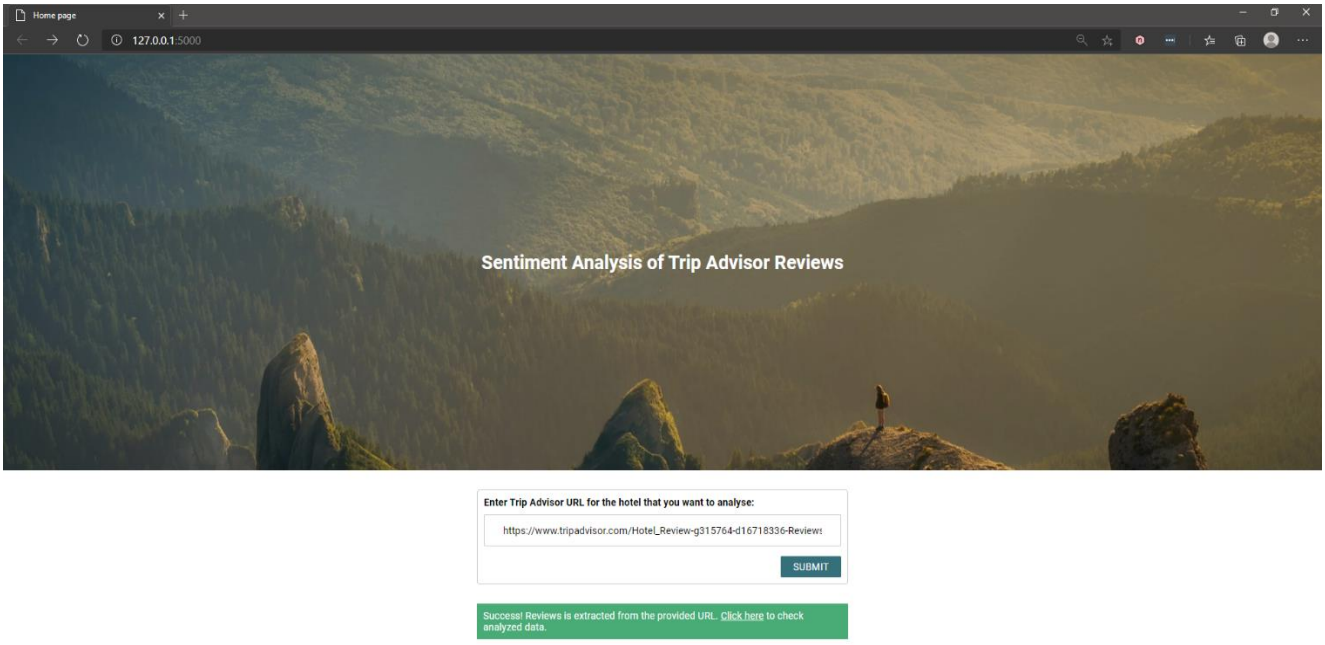



*Figure 51. System test 1: Check if the reviews, reviews rating and date of reviews are successfully extracted from web scrapper.*

2. Check if the model is trained and gives high accuracy.

| Objective | Accuracy of the model is checked. |
|---|---|
| Action | The accuracy of the model is checked. |
| Excepted Result | The accuracy will be more than 65%. |
| Actual Result | The accuracy was more than 65%. |
| Conclusion | Test was successful. |

*Table 21. System Test 2: Check if the model is trained and gives high accuracy.*

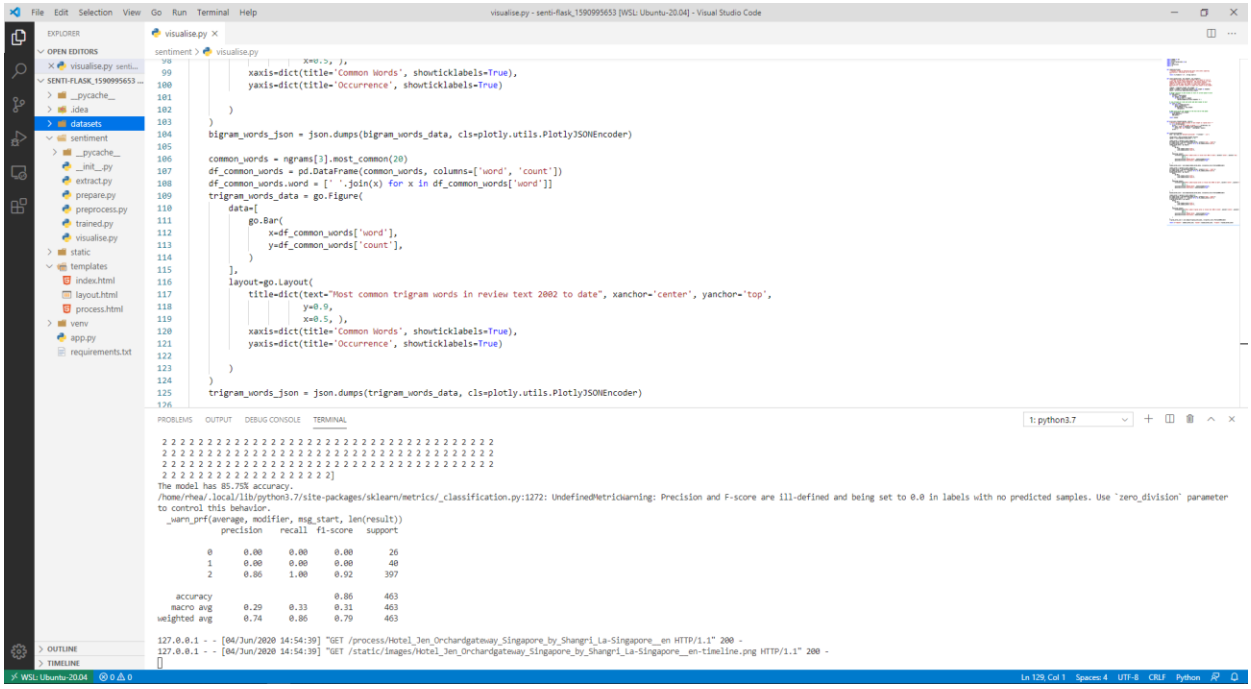



*Figure 52. System Test 2: Check if the model is trained and gives high accuracy.*

3. Check if the graphs are displayed in the web page successfully.

| Objective | Display the graphs. |
|---|---|
| Action | The result page is opened. |
| Excepted Result | The graphs will be displayed. |
| Actual Result | The graphs were displayed. |
| Conclusion | Test was successful. |

*Table 22. System Test 3: Check if the graphs are displayed in the web page successfully.*

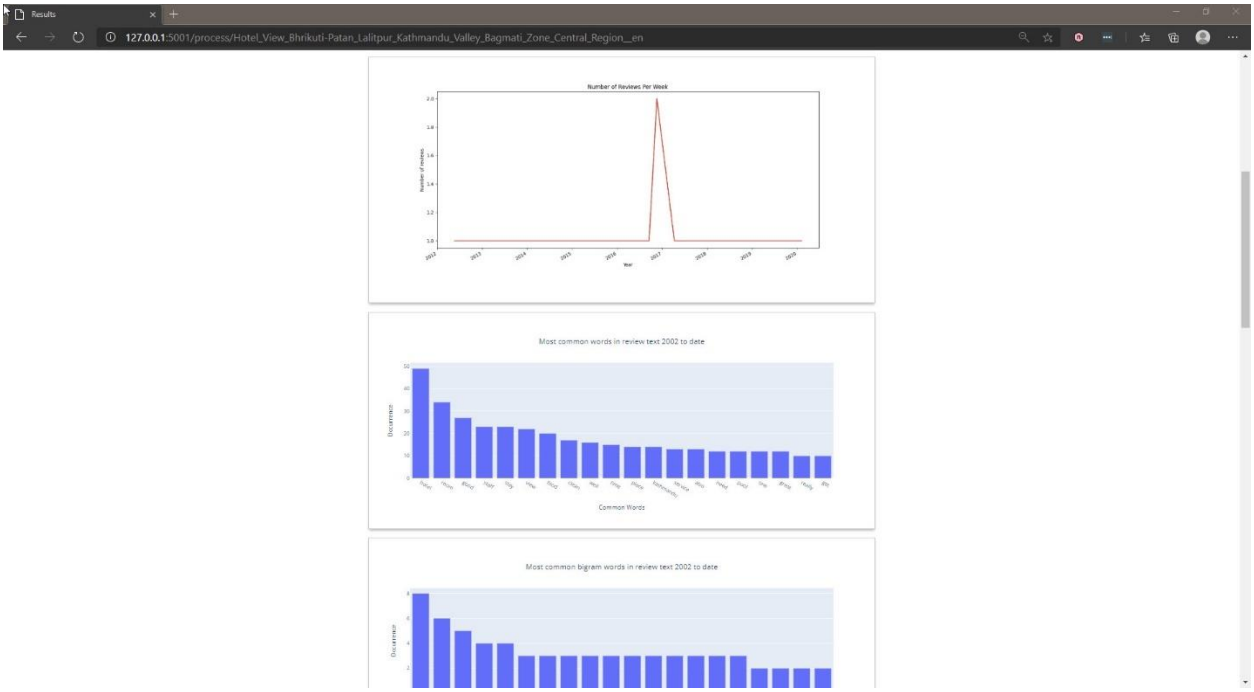



*Figure 53. System Test 3: Check if the graphs are displayed in the web page successfully.*

4. Check if the data of web page can be downloaded successfully.

| Objective | Download the analysed report of hotel reviews |
|---|---|
| Action | Click “Download Analysis Report” button to download. |
| Excepted Result | The report will be downloaded. |
| Actual Result | The report was downloaded. |
| Conclusion | Test was successful. |

*Table 23. System Test 4: Check if the data of web page can be downloaded successfully.*

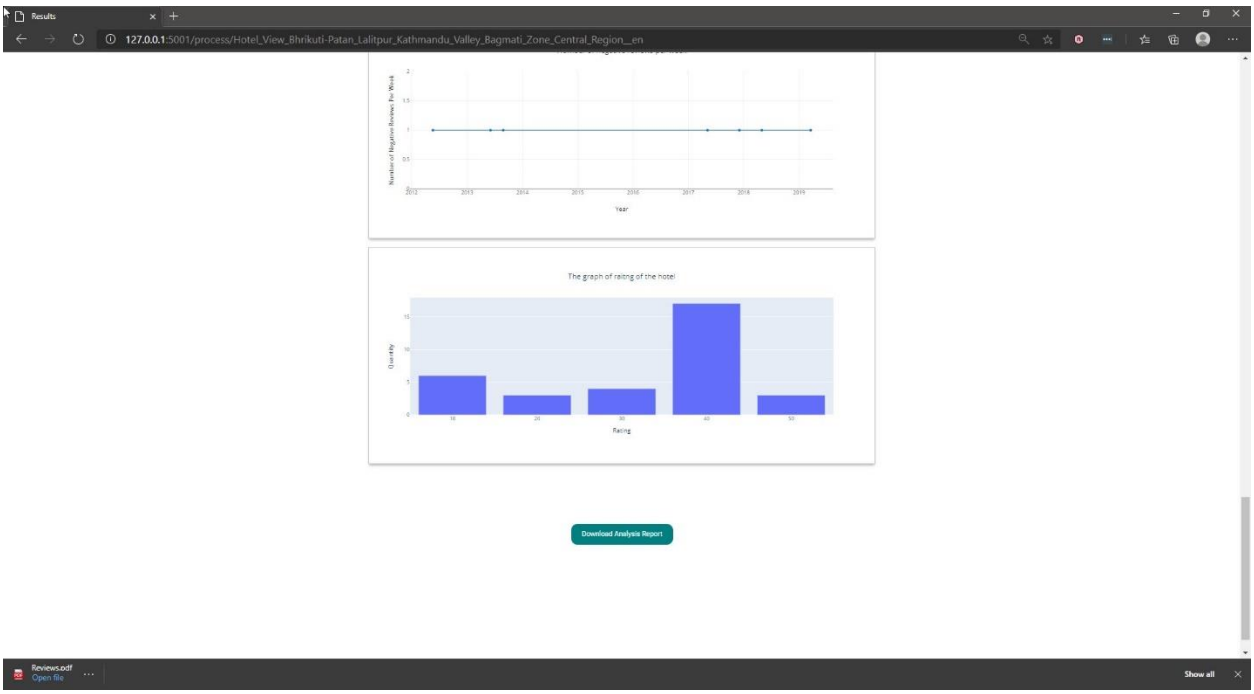



*Figure 54. System Test 4: Check if the data of web page can be downloaded successfully.*

## 4.4 Critical Analysis

This project aims to provide sentiment analysis of the reviews of the hotel. The system was developed to accomplish the aim of the project. The system can mine the reviews of the hotel from the trip advisor website. The system can label the data for training the model for classification. The accuracy of the model is above 65%. The positive and negative reviews are listed for a textual present for the user. The graphs are displayed for the users to represent the sentiment analysis done on the reviews of the hotel.

The data mining can be only done for hotels available in trip advisor website and the reviews available must be in the English language, any other language then English will not be mined. The reviews are filtered and labelled with the help of the rating of the reviews. The model is build using the Naïve Bayes algorithm. The accuracy of the sentiment depends on the quality of the review, the accuracy level increased with good quality reviews and decreased with low-quality reviews. The graphs were displayed to represent the sentiment analysis of the reviews to give a brief idea about the marketing value of the hotel. he sentiment report would be downloaded if required by the user.

The system as a whole was functional and fruitful, but it still required more functions to make the system more applicable for business. The system lacks flexibility, there are only 10 reviews listed. If the user desires to views more reviews, it is not possible. Likewise, in the graph representing the occurrence of reviews, it displays the occurrence of all reviews. It would be more efficient if the user is given the ability to decide if the user desire to see the occurrence of all the reviews, positive reviews, or negative reviews. In the same was for bigram and trigrams. The system meets the aim of the project but lacks felicity for business application.

# Chapter 5. Conclusion

## 5.1 Legal, Social and Ethical Issues

### 5.1.1 Legal Issue

With the increasing use of the Internet, the large amount of data is store in the cloud. There are multiple approaches to use the data, among them, Sentiment Analysis is one. The use of a large volume of data leads to a safety issue. The data mined of the unstructured data is a challenge to data privacy. In the last decade, the computing environment has faced following data privacy challenges such as Organizations sharing private and sensitive for commercial purpose and Appling data mining algorithm in var large datasets (Sharvari tamane, Vijendra Kumar Solanki, Nilanjan Dey, 2017). The legal issues regarding this project are listed below:

- This project uses the data which is mined from the trip advisor. The data mined from trip advisor are data which can be accessed by everyone, so the data used in the project is not illegal.
- The sentiment analysis can be done with the help of corpus which is trained using a large dataset. The dataset used to train corpus may be illegal. The model of this project is trained with the help of the rating. So, the project does not use any corpus.
- The data mined from trip advisor are reviews, rating and review date, there is no information about the customers. So, there is harm to the privacy of customers.

### 5.1.2 Social Issues

Sentiment analysis is mainly used for business purposes, to the reviews of customer towards their product and the market value of their product. The increasing use of e-commercial business, the reviews of the product plays an important role in the selection of the brand. This project gives a summary of the reviews using sentiment analysis. The social issues faced by the project are listed below:

- The reviews make an important role in decision making, so the negative reviews of the hotel will degrade the market value of the hotel.
- The positive review of the hotel is an advertisement itself as it is attracting customers through its positive reviews.
- The new market advertising is reviews of the customers. It raises the question “Are the online reviews a valid source of the information about these hotels?”. The project uses reviews and gives a summary of the reviews inform of graphs. The sentiment analysis itself only gives 65-70% accuracy so it can be referred but not blindly trusted.

### 5.1.3 Ethical Issues

Sentiment analysis deals with the large data which could be provided or hacked. In ethical considerations, the data mined should be publicly available. The data behind a firewall or not publicly available is off-limits. The data of websites with log-in functionality is also off-limits.

In this project, the data is minded from trip advisor. The website has hotel reviews publicly available for users without any log-in functionality. The system has no ethical issues.

## 5.2 Advantages

The sentiment analysis topic is a developing field of AI. Even though it is a developing field, it has gained quite a fame is a short period. Sentiment analysis is mostly used in business. With an increment of e-commerce, their users, and reviews of their product are increasing in the large amount. Sentiment analysis is used to summarize the reviews for the views. This project uses sentiment analysis to summarize the reviews of a hotel of trip advisor. The advantages of this project are listed below:

- The status of reviews helps to track the perception of the hotel by the customers.
- The detail of customer attitude can be specified by analyzing the reviews.
- The time of popularity can be located through a graph of reviews.
- The frequency mentioned feature of the hotel is analyzed for the hotel owners.
- The status of hotel popularity can be tracked.
- The analysis helps the hotel to improve the service mentioned in negative reviews.
- The work of market research is done by this system.
- The recommendation for the hotel, mentioned by the customer can be listed.
- The system highlights customer prioritization.
- It provides an opportunity to develop a new level of customer engagement and reputation of the hotel.
- It acts as a gateway to understanding customer needs which increases the quality of customer service.

## 5.3 Limitations

The sentiment analysis is a new type of “digital hybrid” which is an overlap of computational linguistics, NLP, and text analysis, with an increasing generous dose of AI, blended into the mix. It has the power to shred light in issues which is difficult to measure or simply ignored until now (Mccollum, 2016). The new concept is not perfect yet. This concept was implemented in this project which has a lot of room for improvement. The limitations of this project are listed below:

- It has automation issue as the algorithm would not be able to perceive the “tone” in the writer’s voice. It faces difficulties to identify and parse humour and sarcasm in text for automated sentiment analysis platform.
- This system is only applicable to “English” language only.
- The accuracy even in the best of circumstances is only up to 65 to 70%.
- The data extracted could be bias.
- There are only 10 positive and negative reviews are displayed, the number of reviews displayed should more flexible as per the requirement of the user.
- The ML Algorithm used in this system is Naïve Bayes, but the Support Vector Machine gives more accuracy according to the studies.
- The system lacks the sentiment of word-wise.
- The system can only extract data from the trip advisor website.

## 5.4 Future Work

This project was completed in a given time, but it has a certain limitation. I would like to work on the limitation in future to make it more user- friendly and give better results. The future work for this project is mentioned below:

- I would learn another ML Algorithm so I can implement another algorithm for better accuracy of the model.
- I would make the data extraction more flexible for other websites such as Kayak, Priceline, and such.
- I would make the list of reviews more flexible so the users can select the number of reviews as per their requirement.
- I would like to improve the quality of sentiment analysis by applying sentiment for word than for a sentence.

## Bibliography


- https://www.sciencedirect.com/topics/engineering/sentiment-analysis
- https://www.researchgate.net/figure/Flowchart-of-the-Proposed-Twitter-Sentiment-Analysis-System_fig1_273635463
- https://www.slideshare.net/anilsth91/tweet-sentiment-analysis-78718965
- https://theappsolutions.com/blog/development/sentiment-analysis/
- https://www.sciencedirect.com/science/article/pii/S2090447914000550
- https://www.sciencedirect.com/science/article/pii/S0020025516303917
- https://stackabuse.com/python-for-nlp-sentiment-analysis-with-scikit-learn/
- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.where.html
- https://swcarpentry.github.io/python-novice-gapminder/09-plotting/
- https://plotly.com/python/plot-data-from-csv/
- https://pdfs.semanticscholar.org/d0e7/b0b3dcea6aed5c746f9fab73d8b183dbb79e.pdf
- https://books.google.com.np/books?id=nrQ0CgAAQBAJ&pg=PA476&lpg=PA476&dq=legal+issues+related+to+sentiment+analysis&source=bl&ots=iEN4egmNxr&sig=ACfU3U1Yg50w5VRiHbhKeAH7BSOB7YLmWg&hl=en&sa=X&ved=2ahUKEwjp3IqqpdvpAhUR73MBHanXDRMQ6AEwEHoECAoQAQ#v=onepage&q=legal%20issues%20related%20to%20sentiment%20analysis&f=false
- https://books.google.com.np/books?id=v3HgDwAAQBAJ&pg=PA347&lpg=PA347&dq=legal+issues+related+to+sentiment+analysis&source=bl&ots=bkvY9AFNus&sig=ACfU3U3PRiLaGprz9lKQQaFwsnJVAsnFIA&hl=en&sa=X&ved=2ahUKEwjp3IqqpdvpAhUR73MBHanXDRMQ6AEwD3oECAkQAQ#v=onepage&q=legal%20issues%20related%20to%20sentiment%20analysis&f=false
- https://pdfs.semanticscholar.org/4f4b/4e5b1f8b5bbef3abfe83f352fc6b1b7fcce3.pdf
- https://www.onespace.com/blog/2017/05/what-is-sentiment-analysis-uses-challenges/
- https://www.dhi.ac.uk/san/waysofbeing/data/data-crone-kennedy-2012.pdf

- https://www.ijert.org/research/sentiment-analysis-and-its-challenges-IJERTV4IS030925.pdf
- https://www.guru99.com/what-is-big-data.html
- https://towardsdatascience.com/mach ine-learning-vs-big-data-lets-find-the-relationship-between-them-e55c9c861311
- https://theappsolutions.com/blog/development/machine-learning-and-big-data/
- https://ieeexplore.ieee.org/abstract/document/7219856
- https://patents.google.com/patent/US8838633B2/en
- https://monkeylearn.com/sentiment-analysis/
- https://becominghuman.ai/connection-between-data-science-ml-and-ai-d1c18d89b0bd
- https://books.google.com.np/books?hl=en&lr=&id=lSkIewOw2WoC&oi=fnd&pg=PR5&dq=machine+learning+and+artificial+intelligence+relation&ots=T1EMP-egn0&sig=_g0XKi7AqOOowJy_51FsLtchi8&redir_esc=y#v=onepage&q=machine%20learning%20and%20artificial%20intelligence%20relation&f=false
- https://www.ibm.com/developerworks/library/ba-sentiment-analysis-big-data/index.html
- https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-1-2697bb111ed4
- https://www.researchgate.net/publication/312176414_Sentiment_Analysis_in_Python_using_NLTK
- https://tweepy.readthedocs.io/en/latest/
- https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/
- https://stackabuse.com/python-for-nlp-introduction-to-the-textblob-library/
- https://www.nltk.org/book/ch02.html
- https://www.nltk.org/api/nltk.corpus.html
- https://www.nltk.org/book/
- https://pythonspot.com/category/nltk/
- https://realpython.com/python-matplotlib-guide/

- https://www.geeksforgeeks.org/python-introduction-matplotlib/
- http://cs231n.github.io/python-numpy-tutorial/
- https://pandas.pydata.org/pandas-docs/version/0.15/tutorials.html
- https://pandas.pydata.org/
- https://python-graph-gallery.com/seaborn/
- https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Methodology
- http://www.professionalqa.com/iterative-model
- https://searchsoftwarequality.techtarget.com/definition/iterative-development
- https://airbrake.io/blog/sdlc/waterfall-model
- https://www.analyticsinsight.net/benefits-of-sentiment-analysis-for-businesses/
- https://www.growthaccelerationpartners.com/tech/sentiment-analysis/#:~:text=It%20can%20be%20used%20to,advocates%20or%20social%20media%20influencers.
- https://books.google.com.np/books?id=mDxEDwAAQBAJ&pg=PA366&lpg=PA366&dq=legal+issues+related+to+sentiment+analysis+on+reviews+mined+from+tripadvisor&source=bl&ots=iqb--VZyVA&sig=ACfU3U3acMSBadPYR5rdAS4uc550jouhxA&hl=en&sa=X&ved=2ahUKEwje5cr09eXpAhVM8XMBHSZBAD8Q6AEwAnoECAkQAQ#v=onepage&q=legal%20issues%20related%20to%20sentiment%20analysis%20on%20reviews%20mined%20from%20tripadvisor&f=false
- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3357751
- https://www.manipalprolearn.com/blog/sentiment-analysis-algorithms-and-their-applications-data-science-and-ai

## Chapter 6. References

Advisor, T. (2019) *TripAdvisor Network Effect and the Benefits of Total Engagement* [Online]. Available from: https://www.tripadvisor.com/TripAdvisorInsights/w828 [Accessed 5 October 2019].

Brid, R.S. (2018) *Introduction to Decision Trees* [Online]. Available from: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb [Accessed 21 May 2020].

Divyashree N, Santhosh Kumar K L, Jharna Majumdar. (2017) Opinion Mining and Sentimental Analysis of TripAdvisor.in for Hotel Reviews. *International Research Journal of Engineering and Technology (IRJET)* , 4(11), pp.1462 - 1467.

EDP Sciences. (2016) *Multi-Language Sentiment Analysis for Hotel Reviews*. EDP Sciences.

Gandhi, R. (2018) *Naive Bayes Classifier* [Online]. Available from: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c [Accessed 21 May 2020].

Gandhi, R. (2018) *Support Vector Machine — Introduction to Machine Learning Algorithms* [Online]. Available from: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 [Accessed 21 May 2020].

Ghahrai, A. (2018) *Iterative Model* [Online]. Available from: https://www.testingexcellence.com/iterative-model/.

GitHub. (2020) *Chardet* [Online]. Available from: https://chardet.github.io/ [Accessed 31 May 2020].

GNU Guile. (2019) *python-certifi 2019.3.9* [Online]. Available from: https://guix.gnu.org/packages/python-certifi-2019.3.9/ [Accessed 31 May 2020].

HackerEarth. (2017) *An Introduction to the Naive Bayes Algorithm (with codes in Python and R)* [Online]. Available from: https://medium.com/@hackerearth/an-introduction-to-

the-naive-bayes-algorithm-with-codes-in-python-and-r-7c85cdb03490 [Accessed 21 May 2020].

K, J. (2018) *12 BEST SOFTWARE DEVELOPMENT METHODOLOGIES WITH PROS AND CONS* [Online]. Available from: 12 BEST SOFTWARE DEVELOPMENT METHODOLOGIES WITH PROS AND CONS.

matplotlib. (2020) *matplotlib* [Online]. Available from: https://matplotlib.org/.

McCallum, A.K. (2018) *MALLET: A Machine Learning for Language Toolkit* [Online]. UMASS AMHERST Available at: http://mallet.cs.umass.edu/index.php [Accessed 1 May 2020].

Mccollum, O. (2016) *More Than Just a Gut Feeling: Using Sentiment Analysis in Risk Management* [Online]. Available from: file:///C:/Users/rhea_/OneDrive/Documents/College%20Assignment%20and%20Lectures/Final%20Year%20Project/Proposal_0.1.pdf [Accessed 12 October 2019].

Murphy, R. (2020) *Local Consumer Review Survey* [Online]. Available from: https://www.brightlocal.com/research/local-consumer-review-survey/ [Accessed 15 April 2020].

Nadeem Akhtara, N.Z.A.K.T.A. (2017) *Aspect based Sentiment Oriented Summarization of Hotel Reviews*. Cochin, India : Elsevier B. V. 7th International Conference on Advances in Computing & Communications.

Netbeans. (2020) *Introduction to the Google Web Toolkit Framework* [Online]. Netbeans Available at: https://netbeans.org/kb/74/web/quickstart-webapps-gwt.html [Accessed 21 May 2020].

OpeNER. (2020) *What is OpeNER?* [Online]. OpeNER Available at: https://www.opener-project.eu/getting-started/ [Accessed 21 May 2020].

Podium. (2020) *Online Reviews: The Ultimate Guide for Business Owners* [Online]. Available from: https://www.podium.com/article/online-reviews-guide/ [Accessed 15 January 2020].

Powell-Morse, A. (2016) *Iterative Model: What Is It And When Should You Use It?* [Online]. Available from: https://airbrake.io/blog/sdlc/iterative-model [Accessed 05 September 2018].

Prof. Dr. -Ing, S.M. (2020) *BESAHOT Explanative Bewertungsanalyse für die Saarländische Hotellerie* [Online]. Deutsches Forschungszentrum für Künstliche Intelligenz Available at: https://www.dfki.de/en/web/research/projects-and-publications/projects/project/besahot/ [Accessed 21 May 2020].

Sharvari tamane, Vijendra Kumar Solanki, Nilanjan Dey. (2017) *Privacy and Security Policies in Big Data*. Hershey Pa: IGI Globa.

Software, A. (2020) *An Introduction To Software Development Methodologies* [Online]. Available from: https://www.alliancesoftware.com.au/introduction-software-development-methodologies/.

Tripadvisor. (2017) *TripAdvisor Network Effect and the Benefits of Total Engagement* [Online]. Available from: https://www.tripadvisor.com/TripAdvisorInsights/w828 [Accessed 5 December 2019].

Tutoial Point. (2019) *SDLC - Waterfall Model* [Online]. Available from: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm [Accessed 19 October 2019].

Walter Kasper, Mihaela Vela. (2011) Sentiment Analysis for Hotel Reviews. *Proceedings of the Computational*, pp.45 - 52.

# Chapter 7. Appendix

## 7.1 Appendix A: Pre-survey

### 7.1.1 Pre-Survey Form

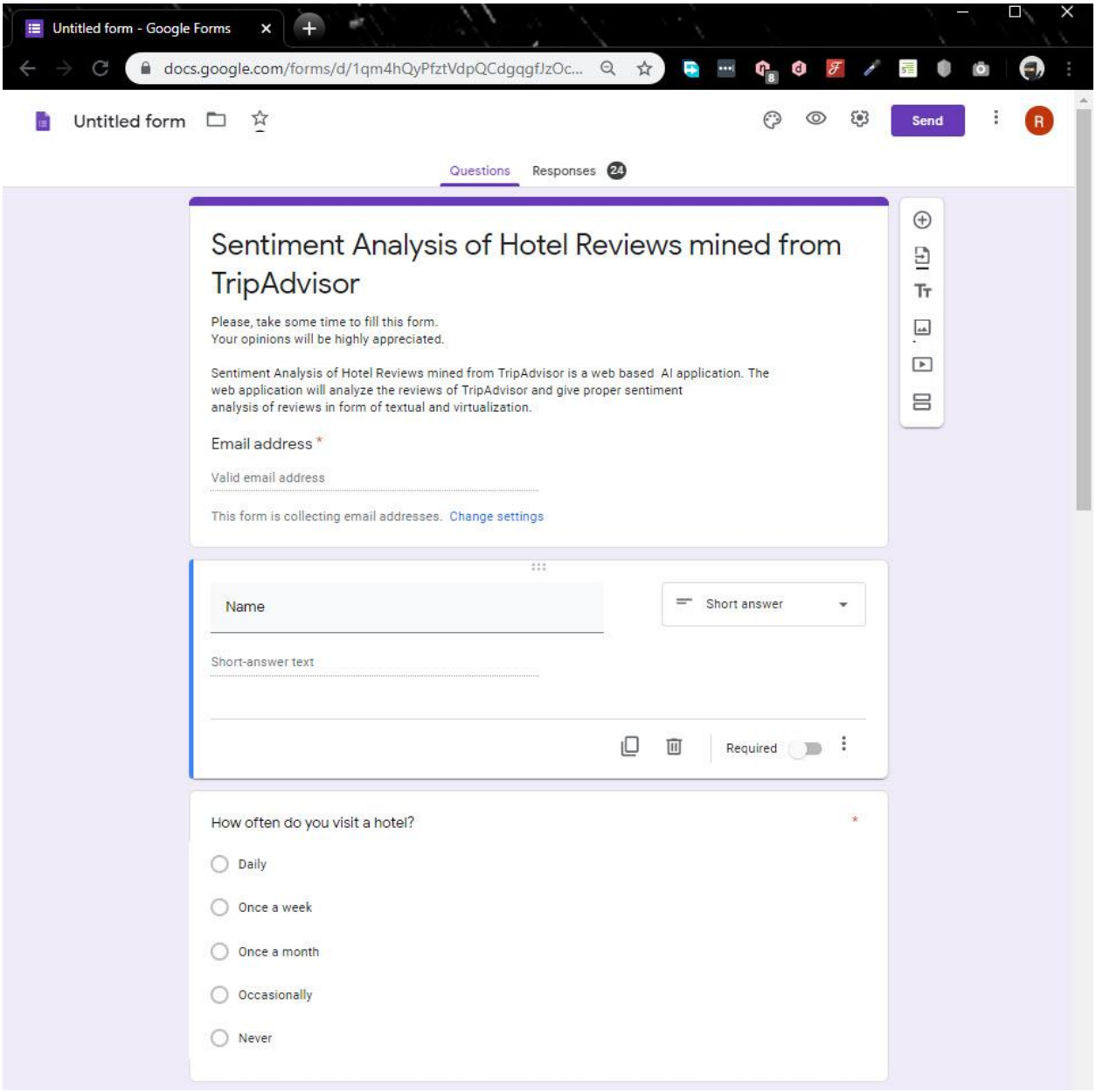



*Figure 55. Pre-survey form. fig(a)*

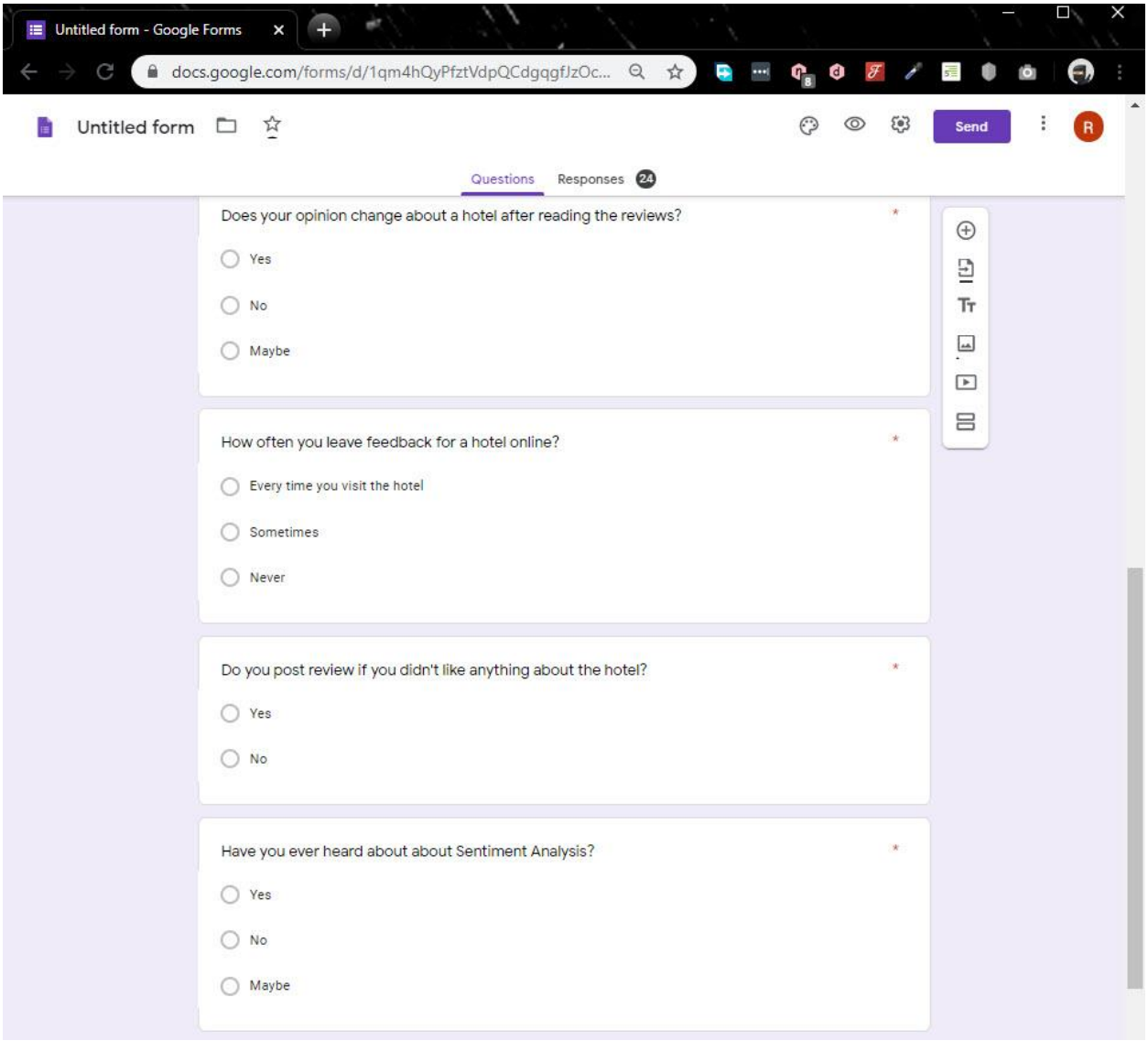



*Figure 56. Pre-survey form. fig(b)*

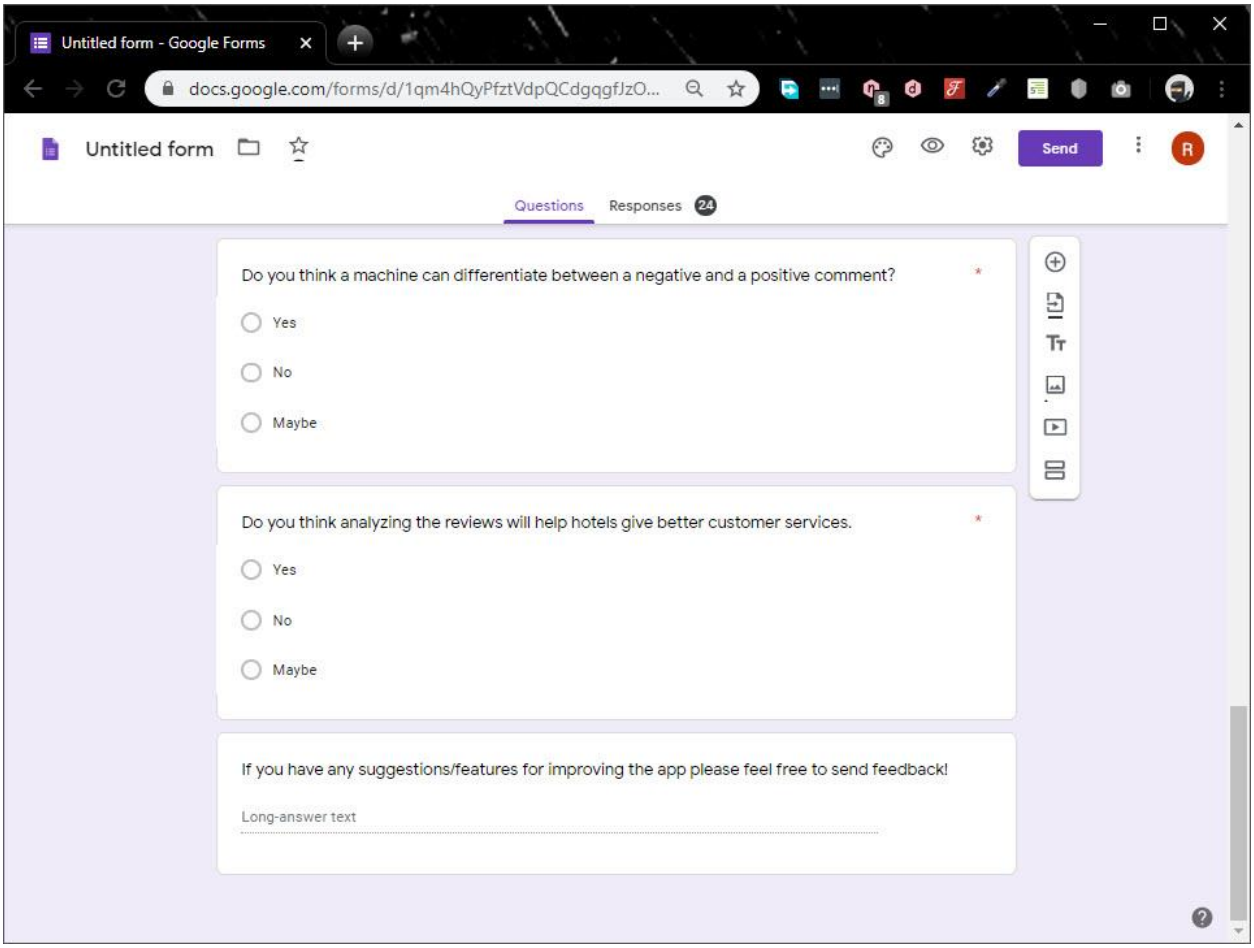



*Figure 57. Pre-survey form. fig(c)*

### 7.1.2 Sample of Filled Pre-Survey Forms

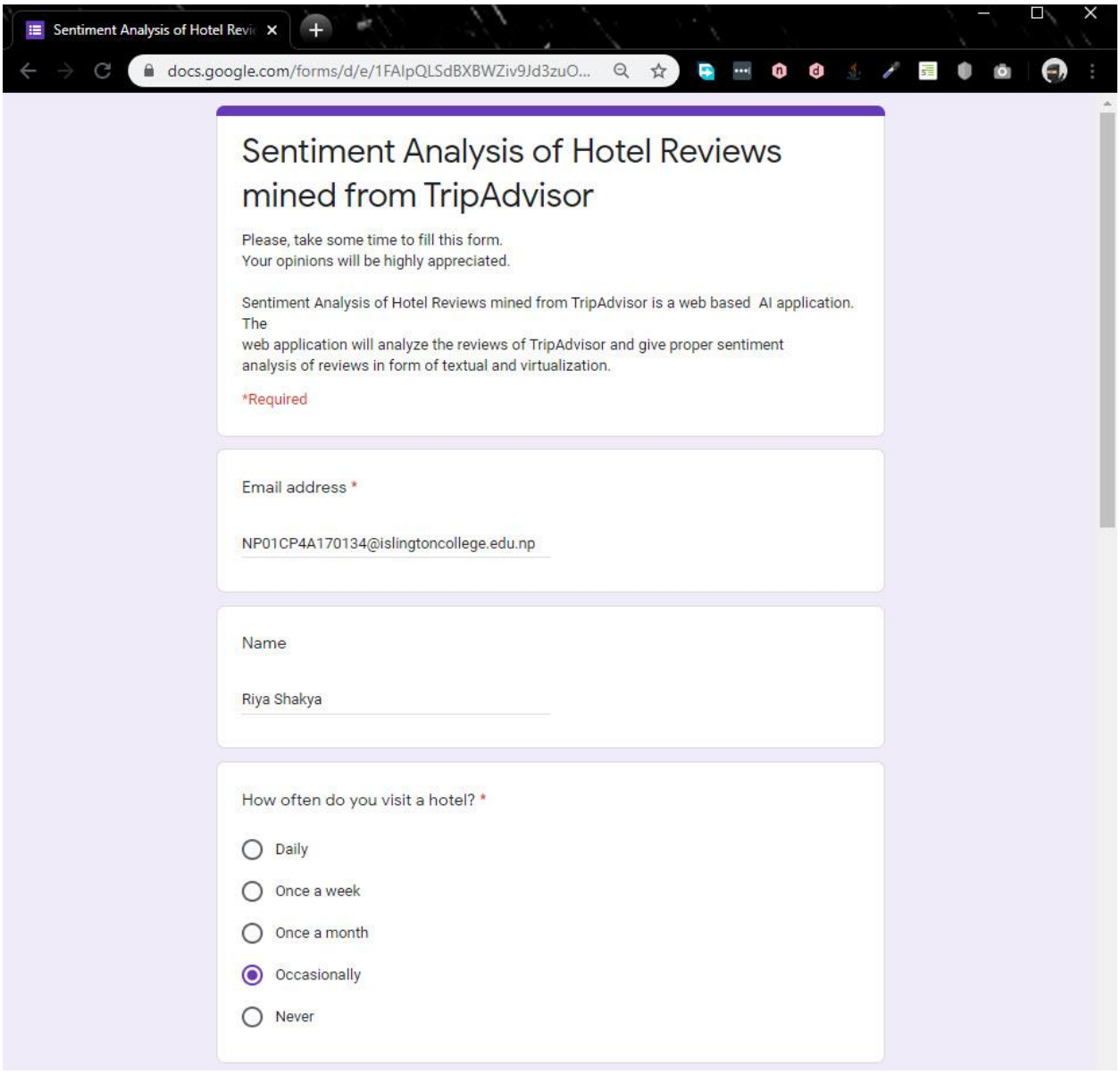



*Figure 58. Sample of filled pre-survey. Fig(a)*

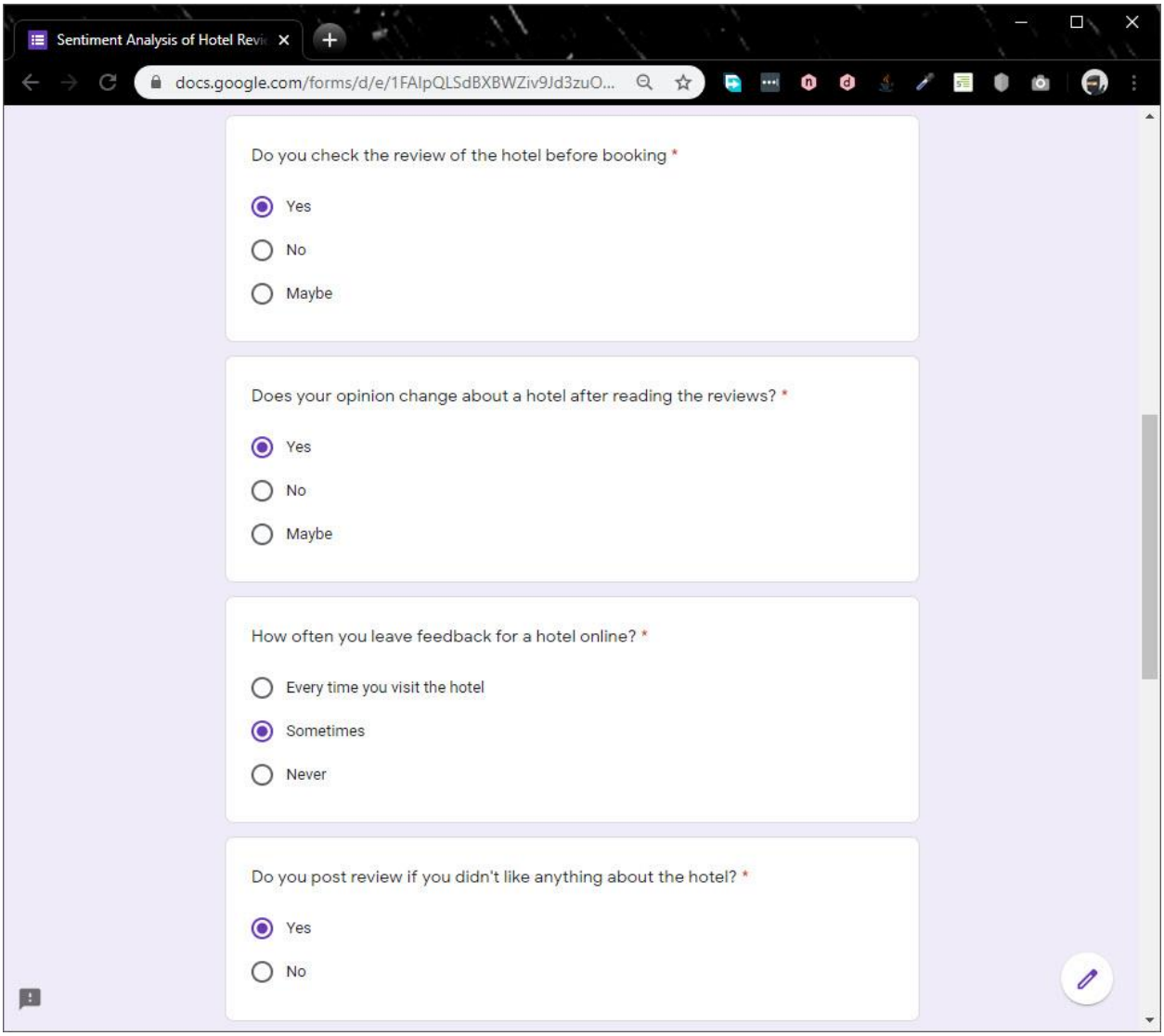



*Figure 59. Sample of filled pre-survey. Fig(b)*

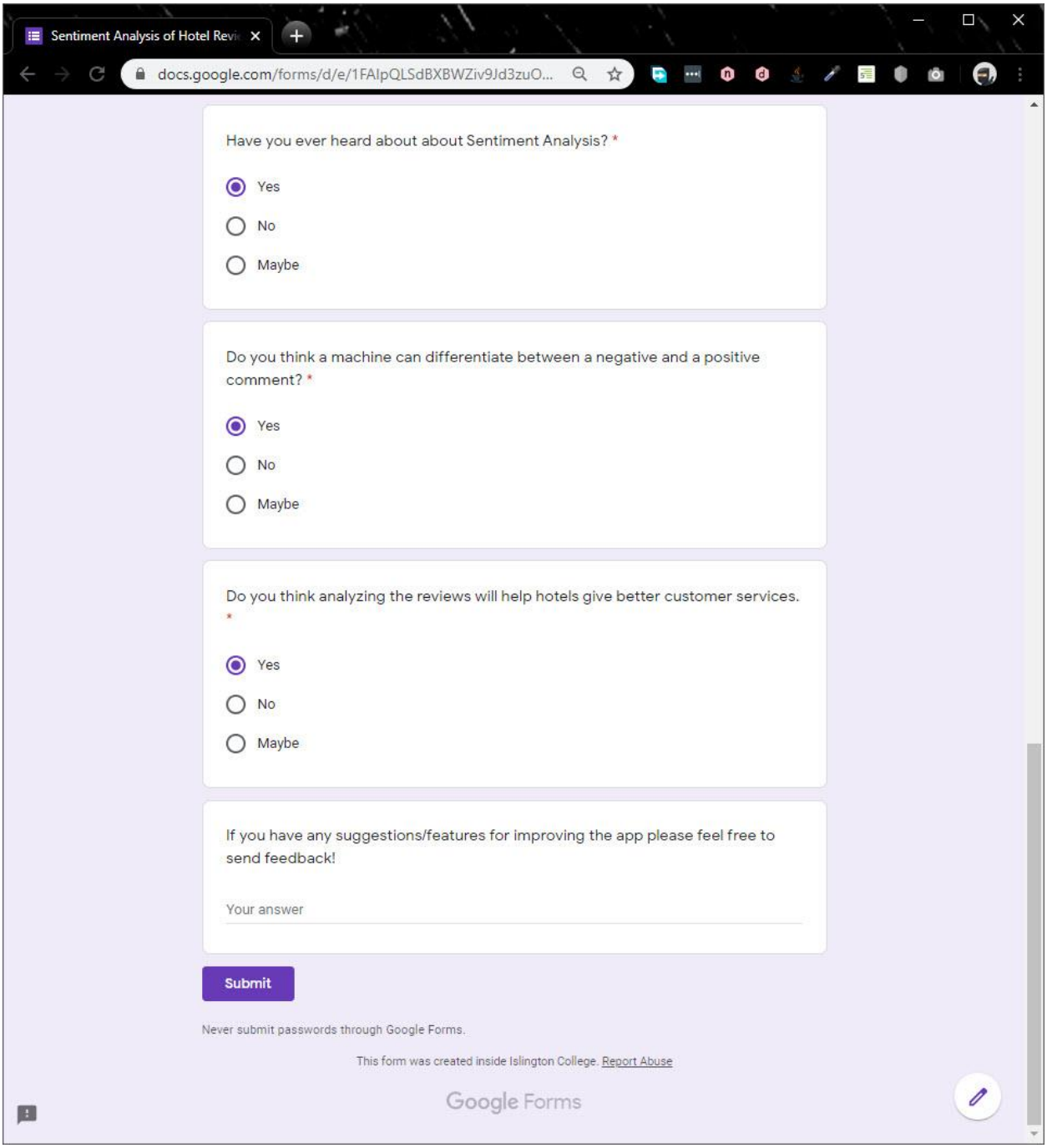



*Figure 60. Sample of filled pre-survey. Fig(c)*

### 7.1.3 Pre-Survey Result

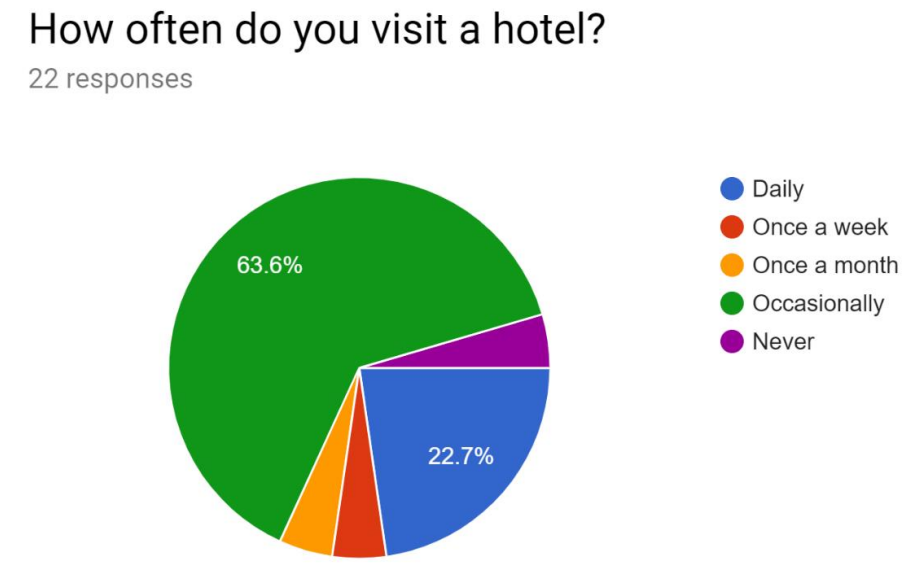



*Figure 61. Pre-survey result. fig(a)*

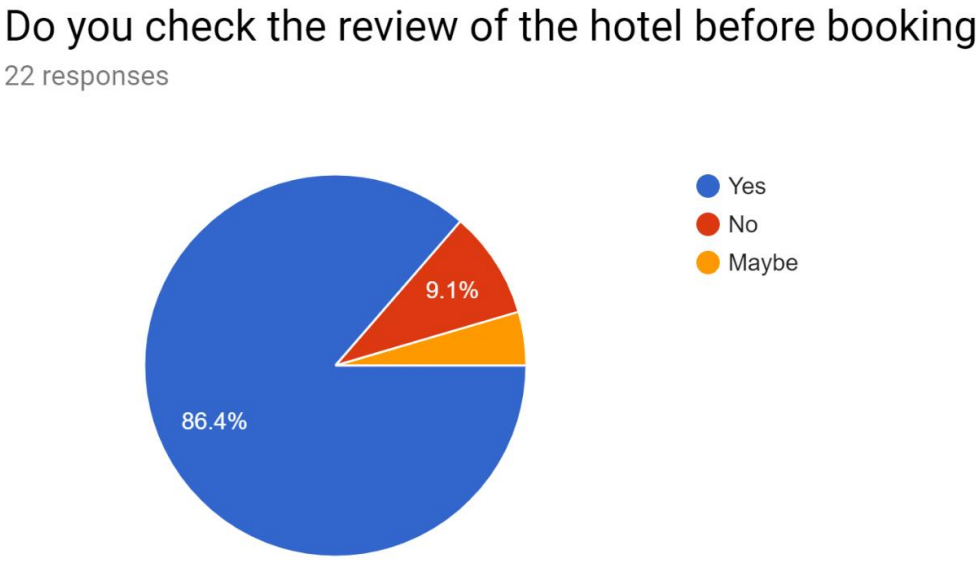



*Figure 62. Pre-survey result. fig(b)*

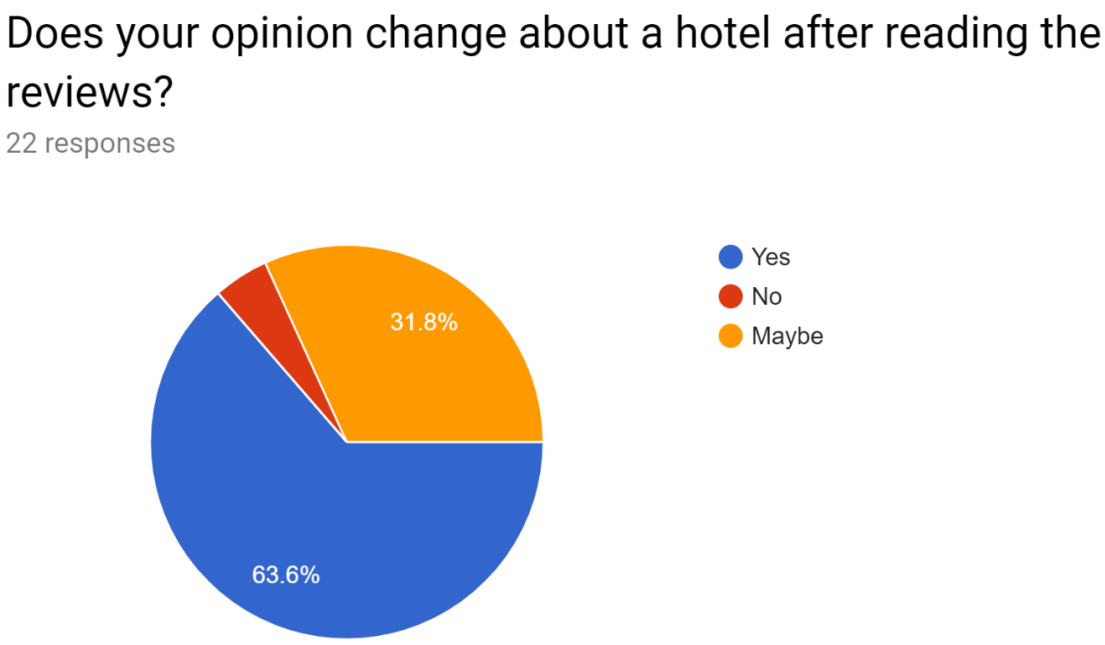



*Figure 63. Pre-survey result. fig(c)*

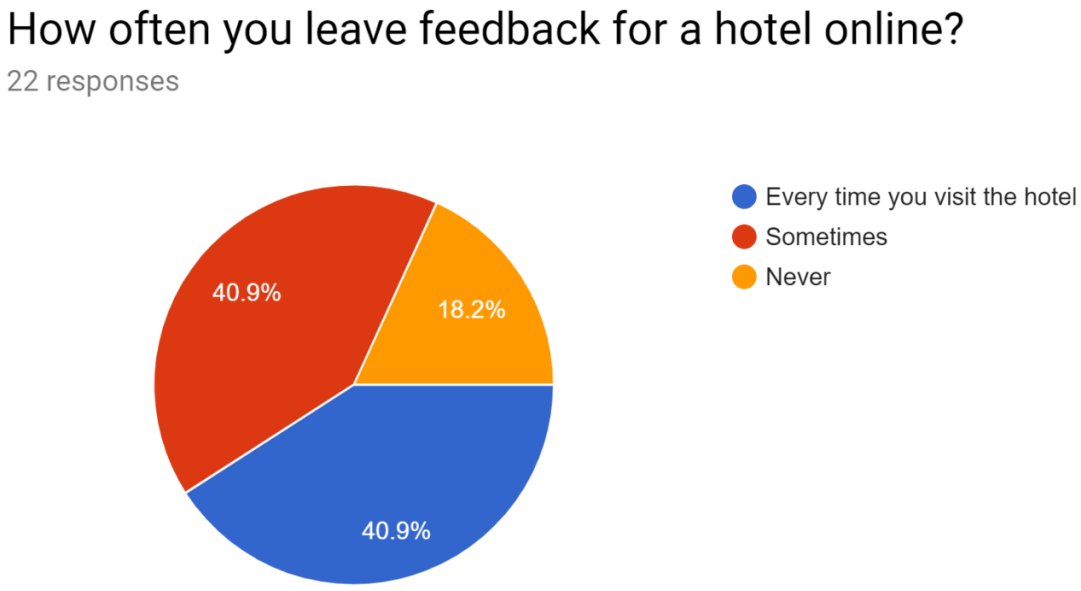



*Figure 64. Pre-survey result. fig(d)*

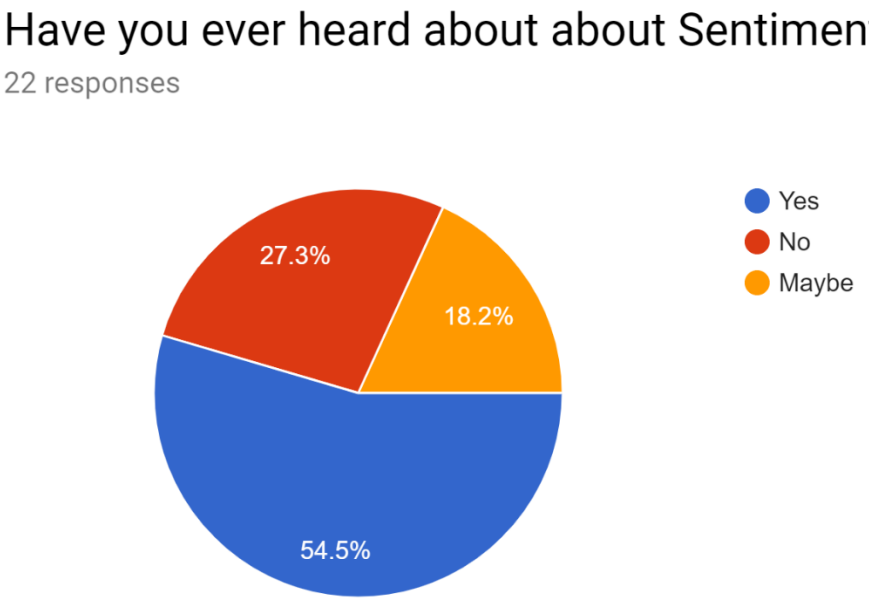



*Figure 65. Pre-survey result. fig(e)*

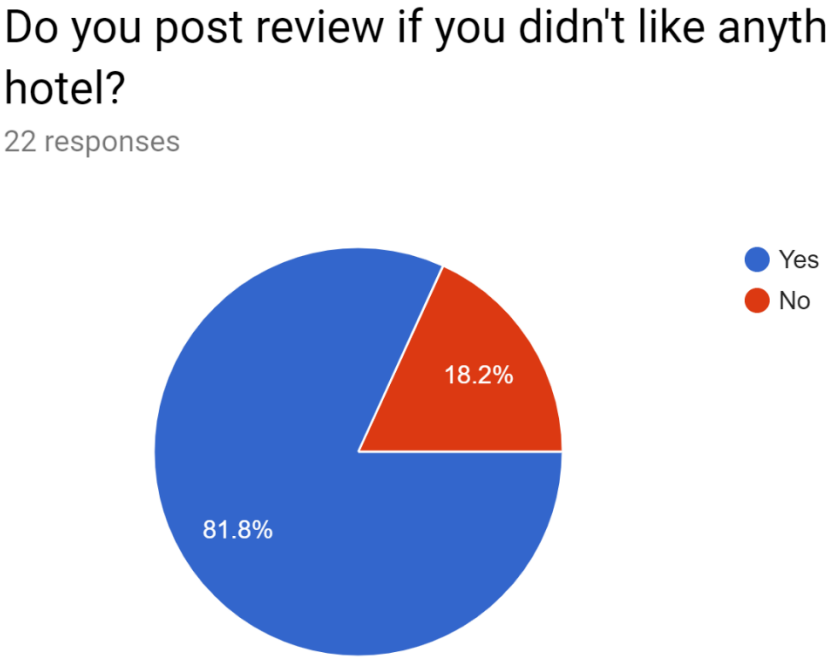



*Figure 66. Pre-survey result. fig(f)*

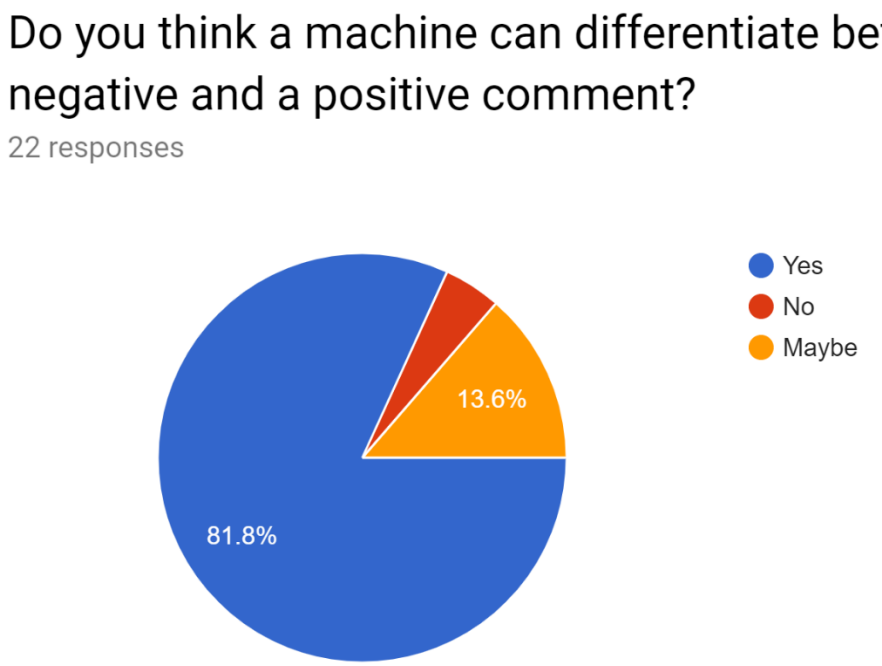



*Figure 67. Pre-survey result. fig(g)*

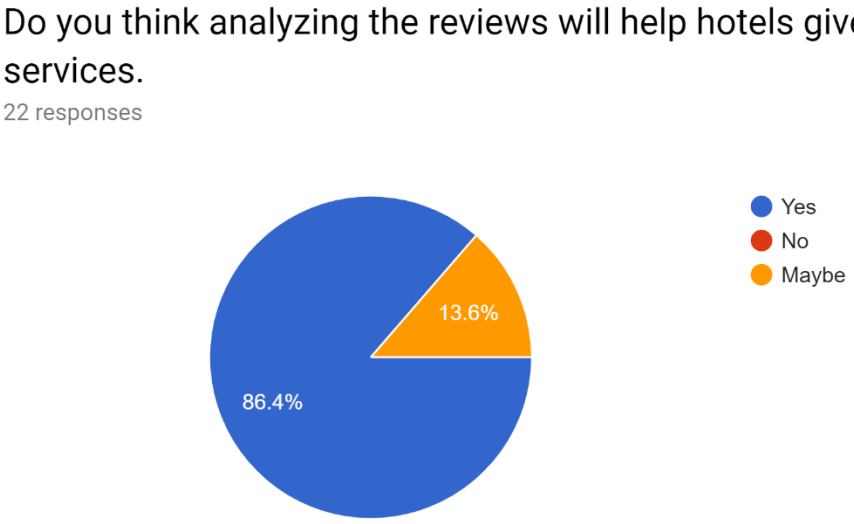



*Figure 68. Pre-survey result. fig(h)*

## 7.2 Appendix B: Post-Survey

### 7.2.1 Post-Survey Form

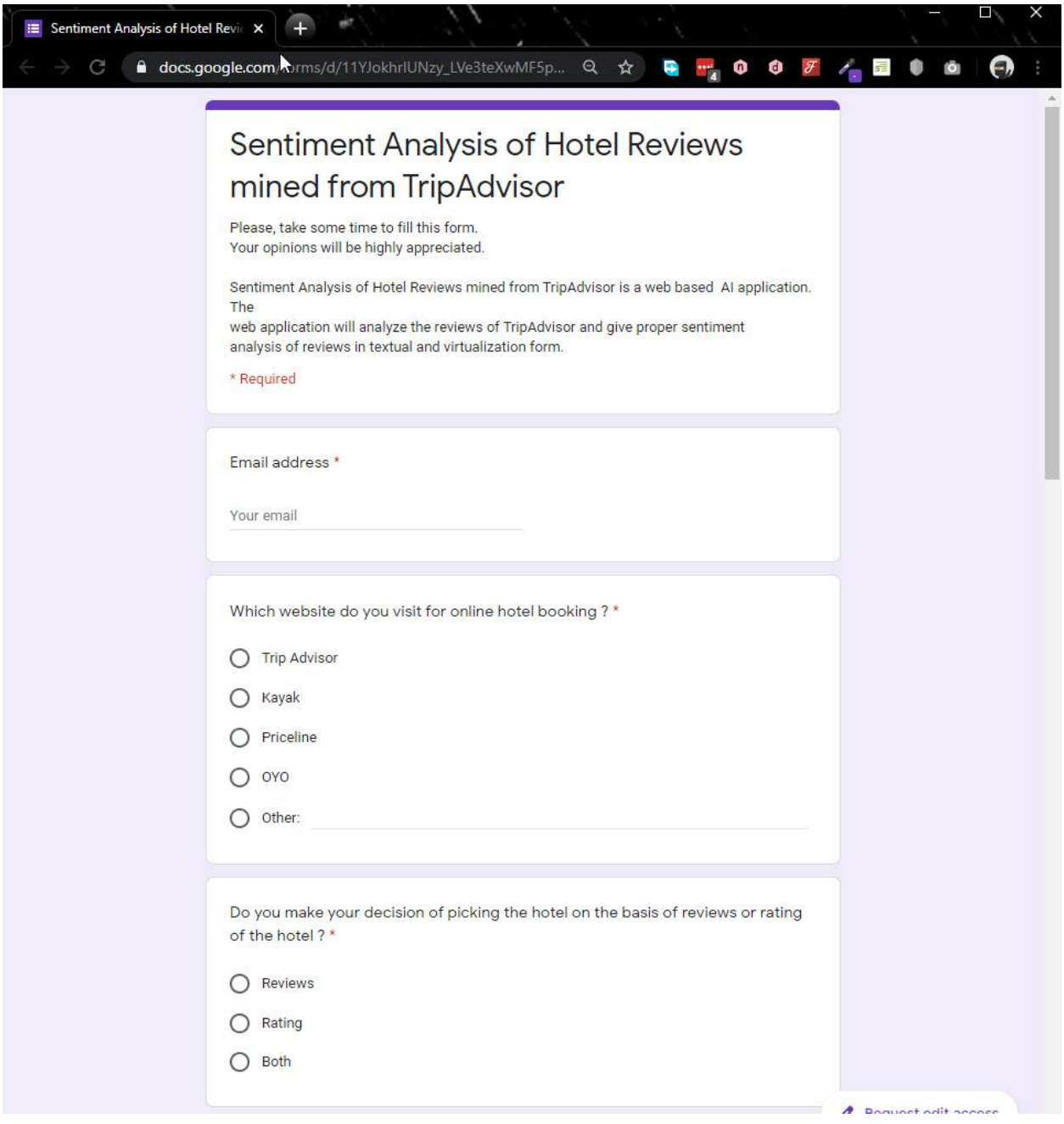



*Figure 69.Post-survey form. fig(a)*

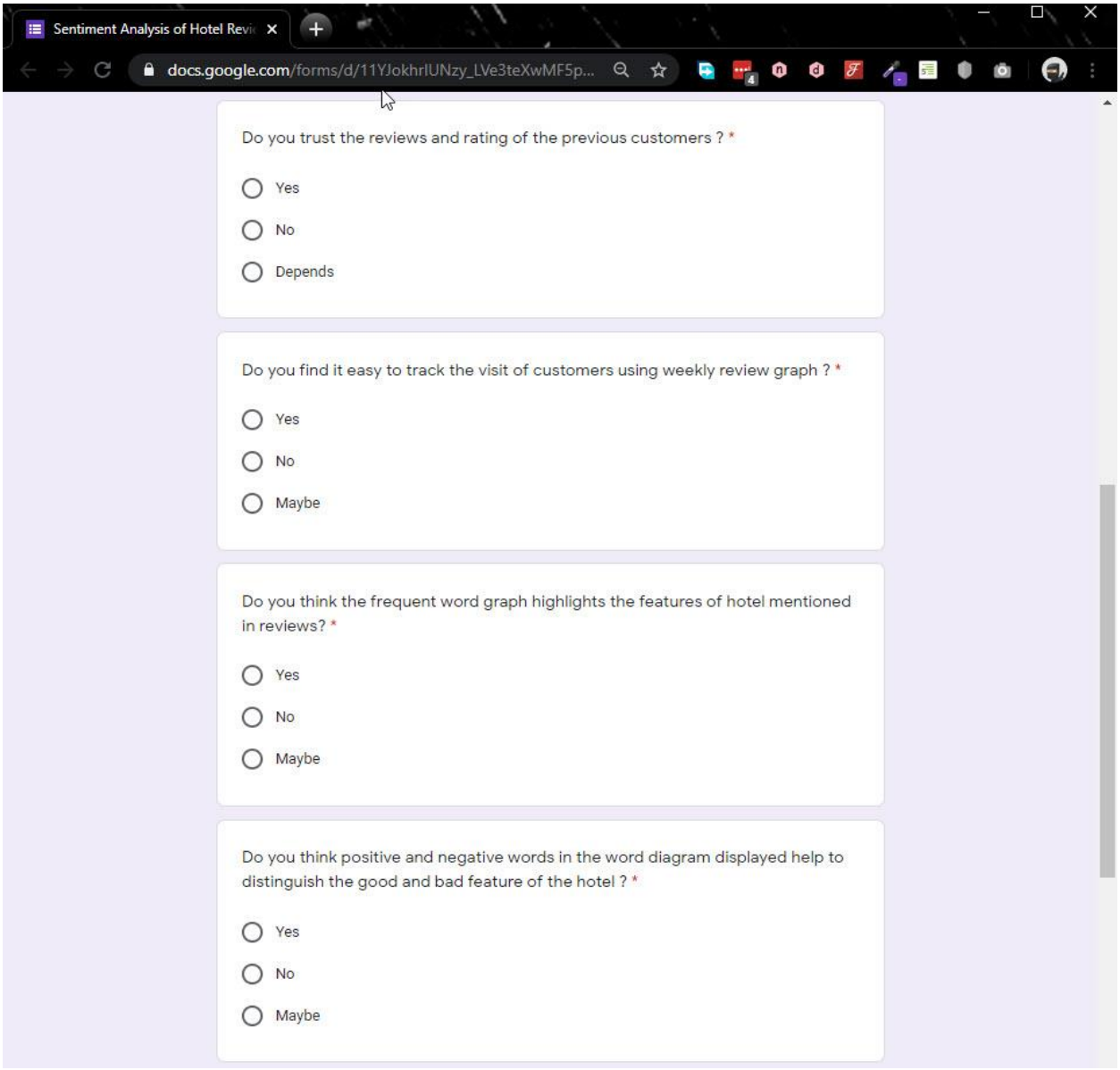



*Figure 70. Post-survey form. fig(b)*

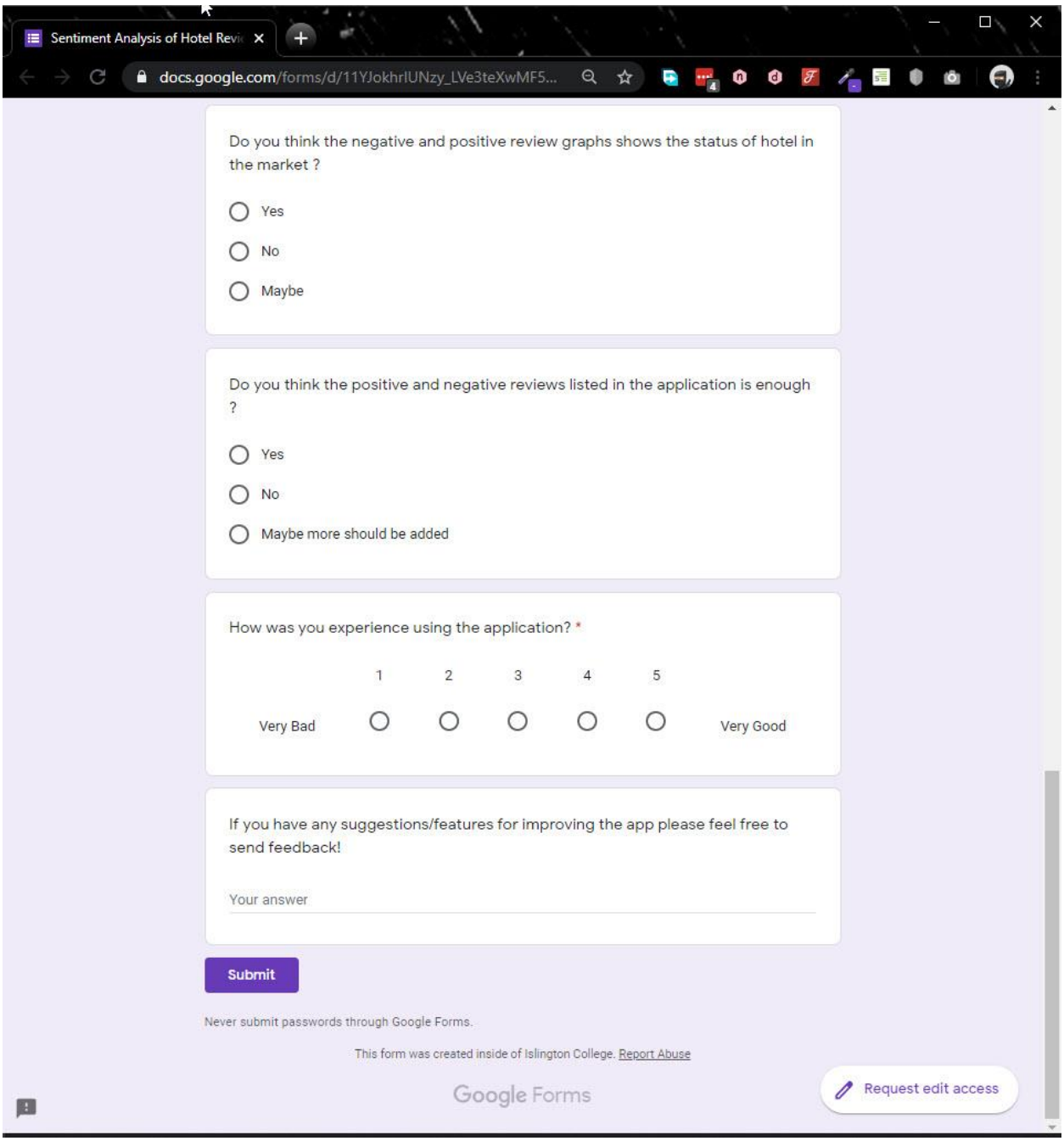



*Figure 71. Post-survey form. fig(c)*

### 7.2.2 Sample of Filled Post-Survey Forms

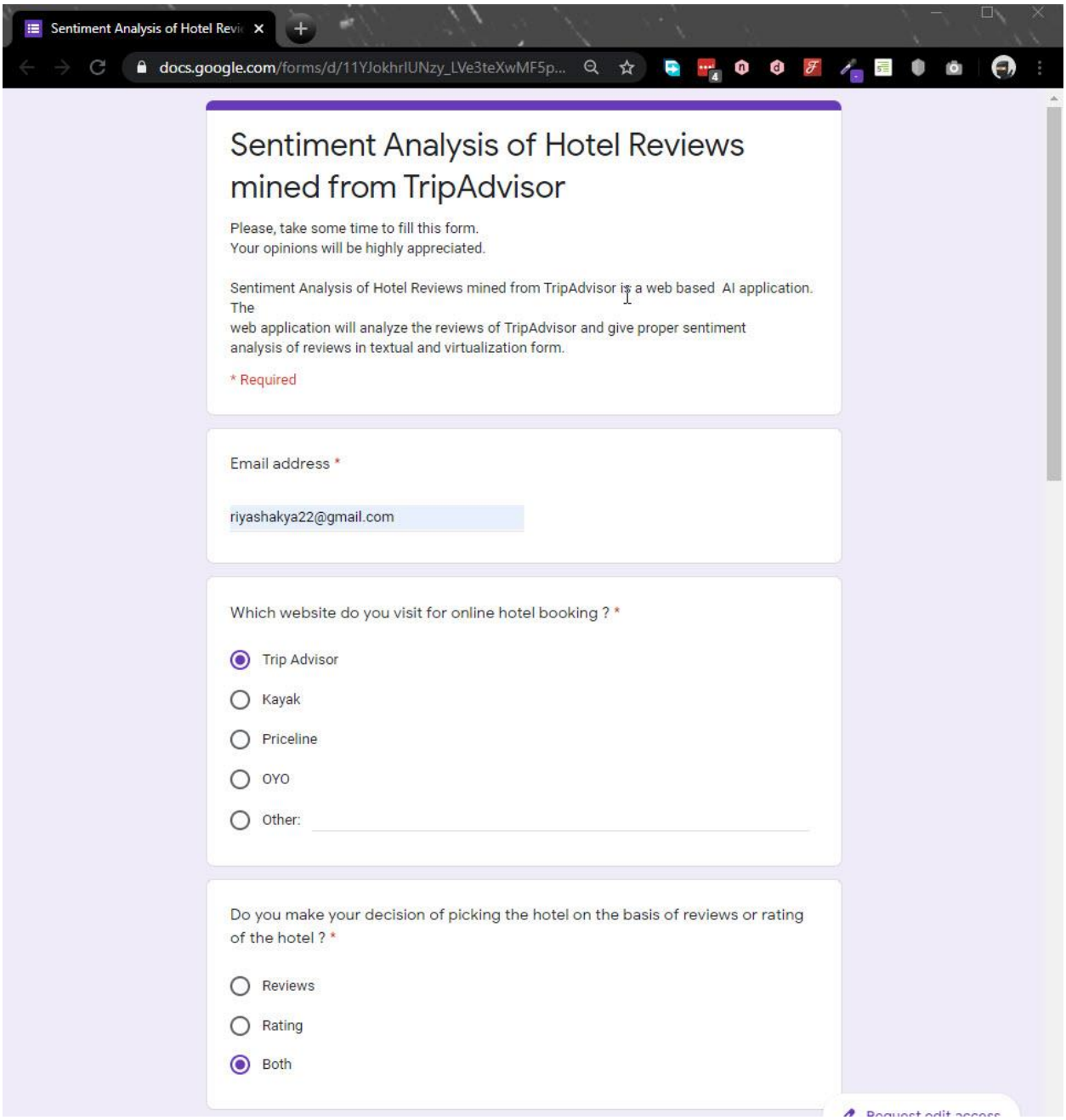



*Figure 72. Sample of filled post-survey. Fig(a)*

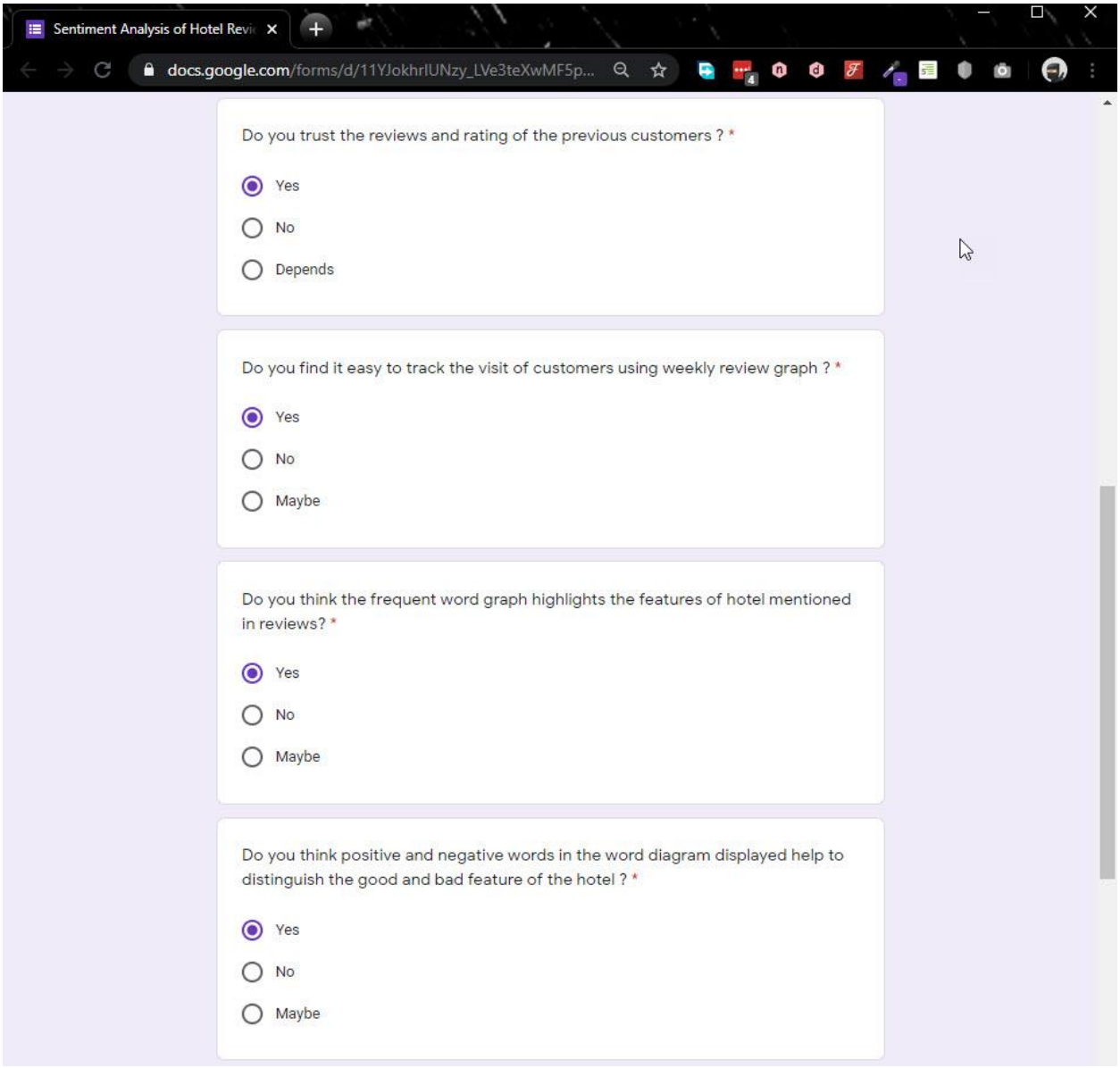



*Figure 73. Sample of filled post-survey. Fig(b)*

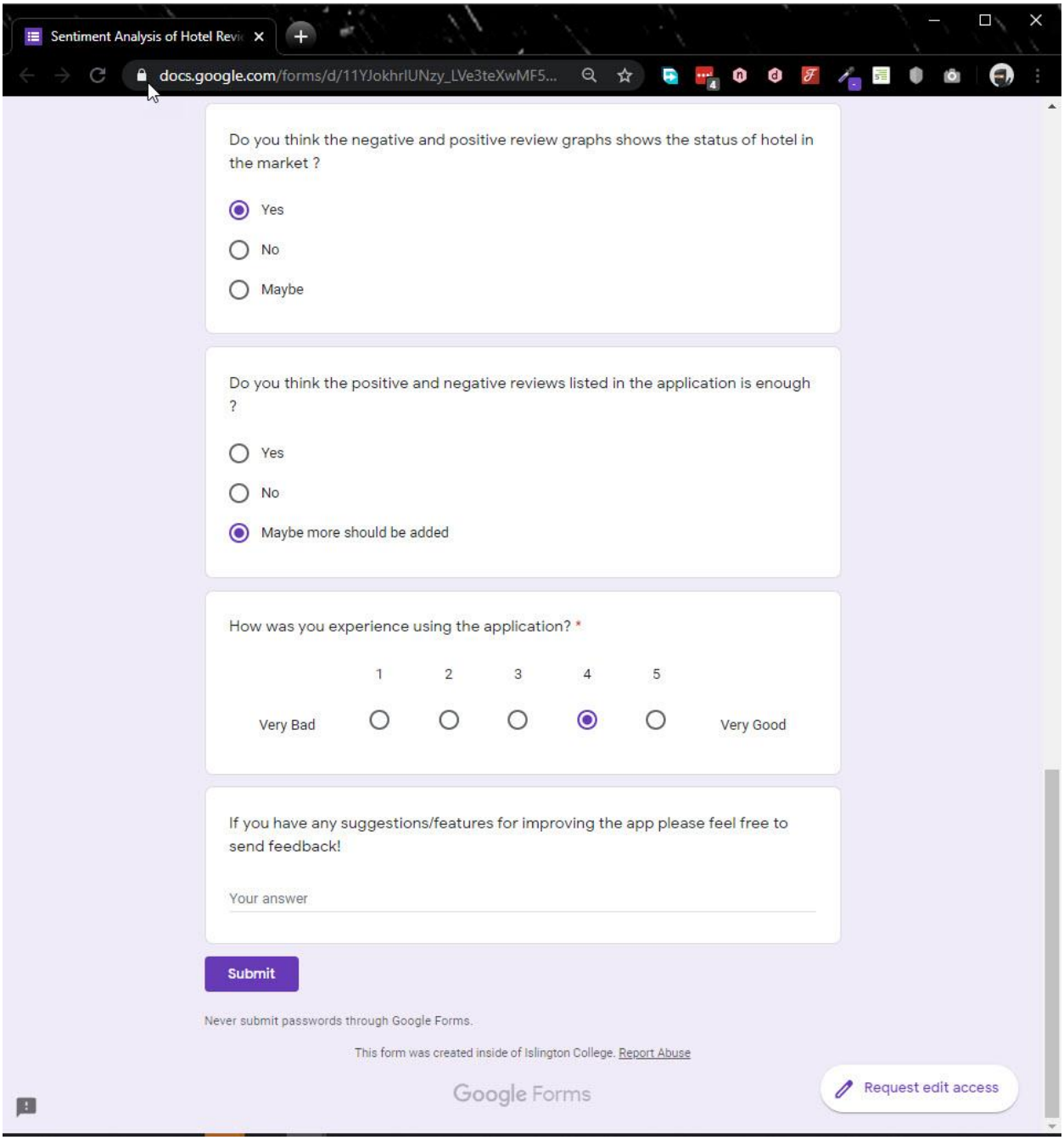



*Figure 74. Sample of filled post-survey. Fig(c)*

### 7.2.3 Post-Survey Result


*Figure 75. Post-survey result. fig(a)*

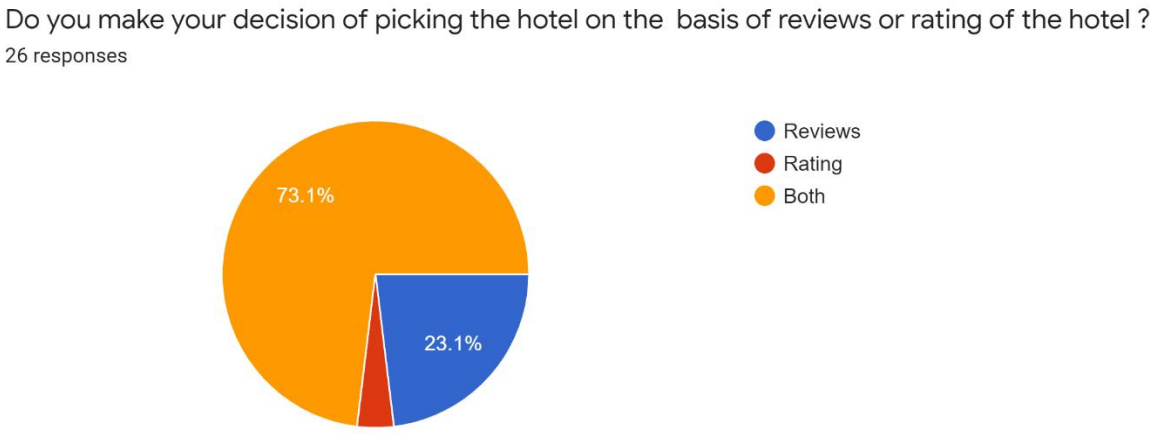



*Figure 76. Post-survey result. fig(b)*

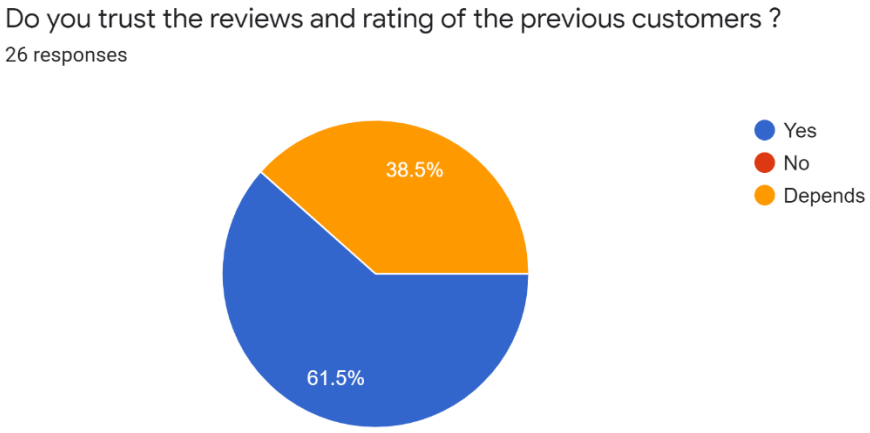



*Figure 77. Post-survey result. fig(c)*

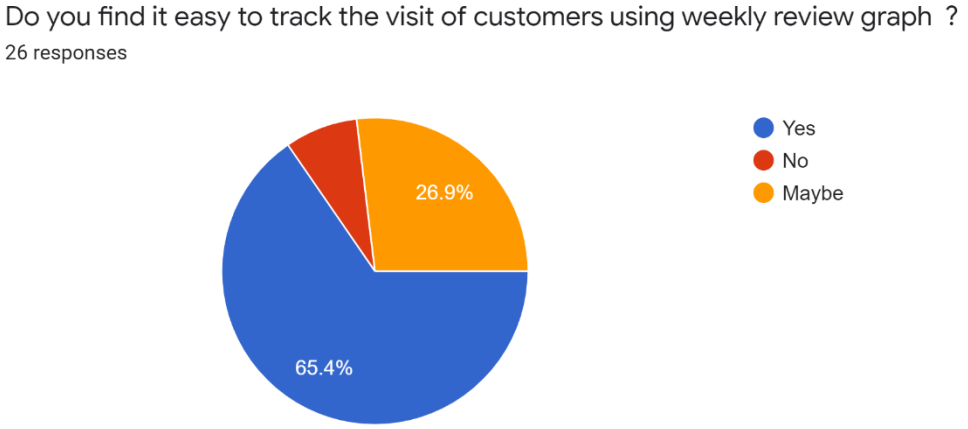



*Figure 78. Post-survey result. fig(d)*

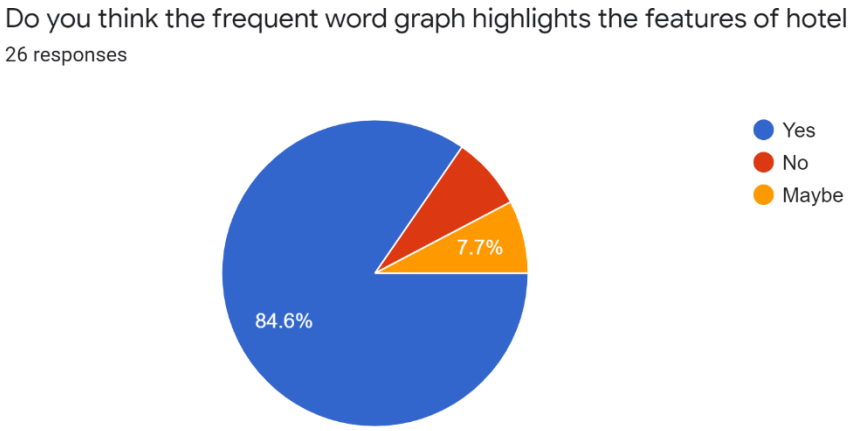



*Figure 79. Post-survey result. fig(e)*

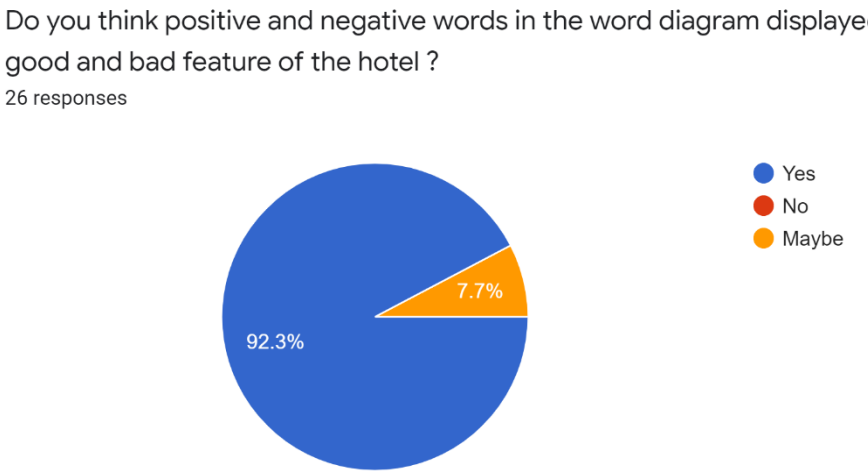



*Figure 80. Post-survey result. fig(f)*

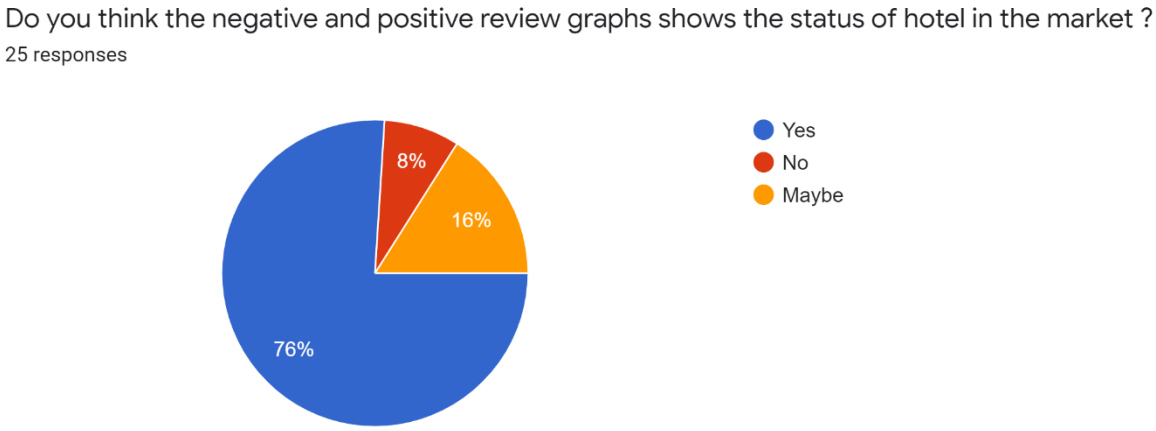



*Figure 81. Post-survey result. fig(g)*

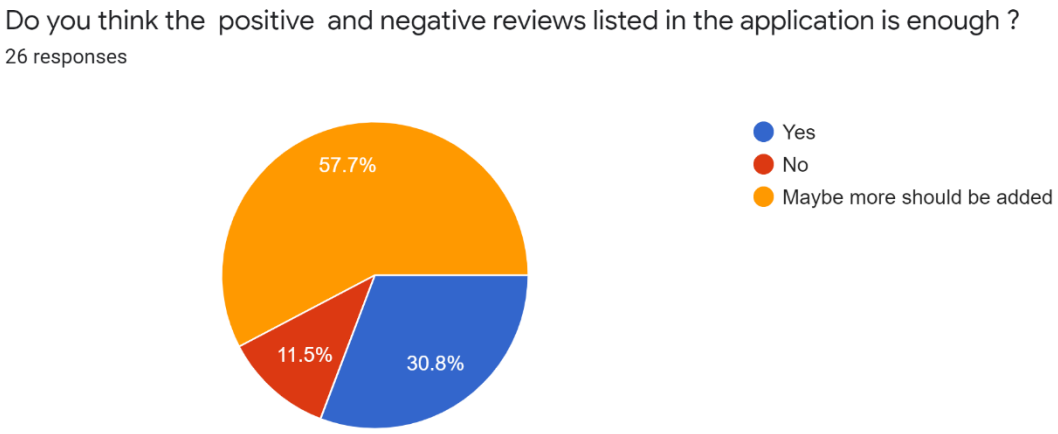



*Figure 82. Post-survey result. fig(h)*

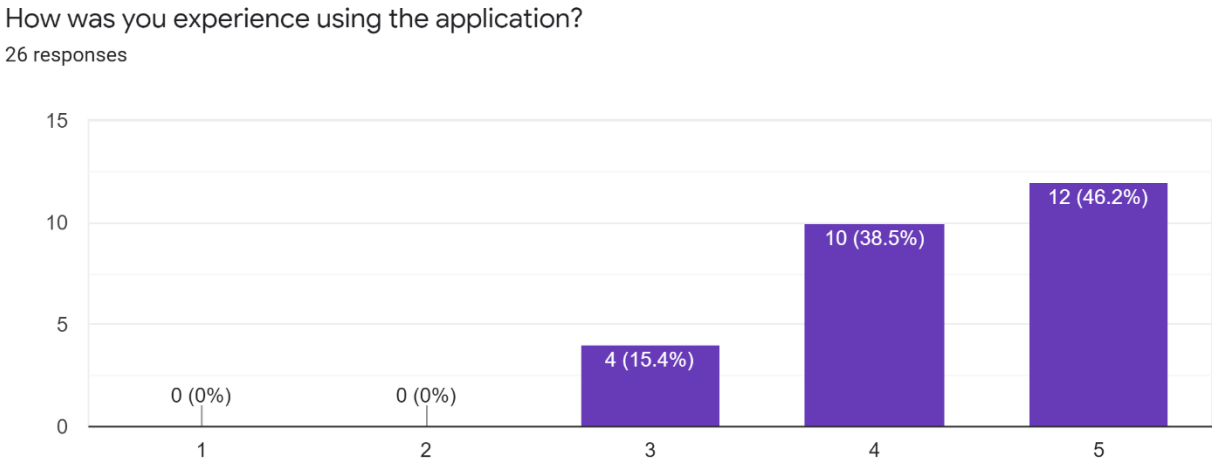



*Figure 83. Post-survey result. fig(i)*

## 7.3 Appendix C: Sample Codes

### 7.3.1 Sample Code of the front-end

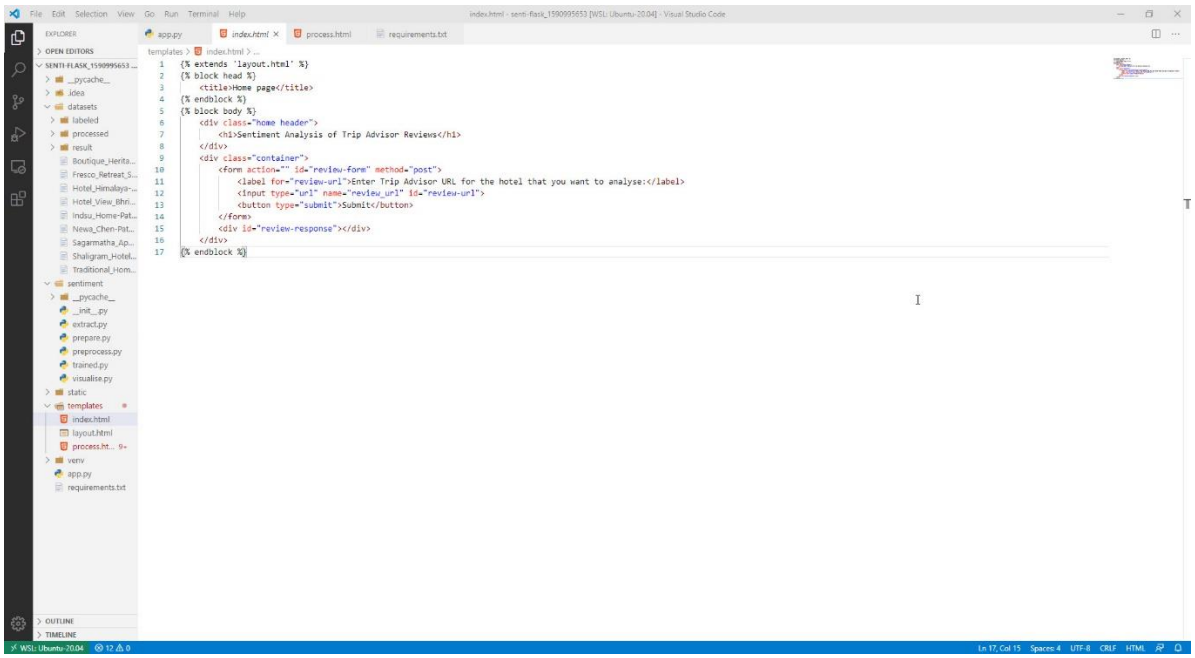



*Figure 84. The front-end code of the system. fig(a)*

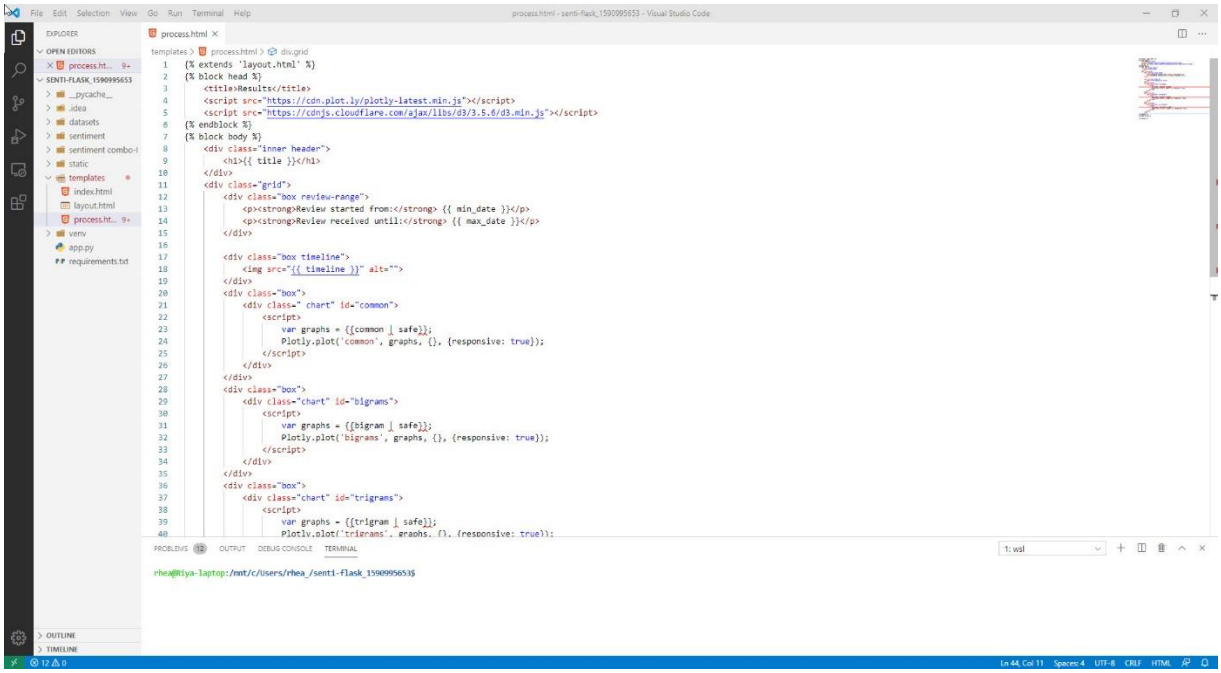



*Figure 85. The front-end code of the system. fig(b)*

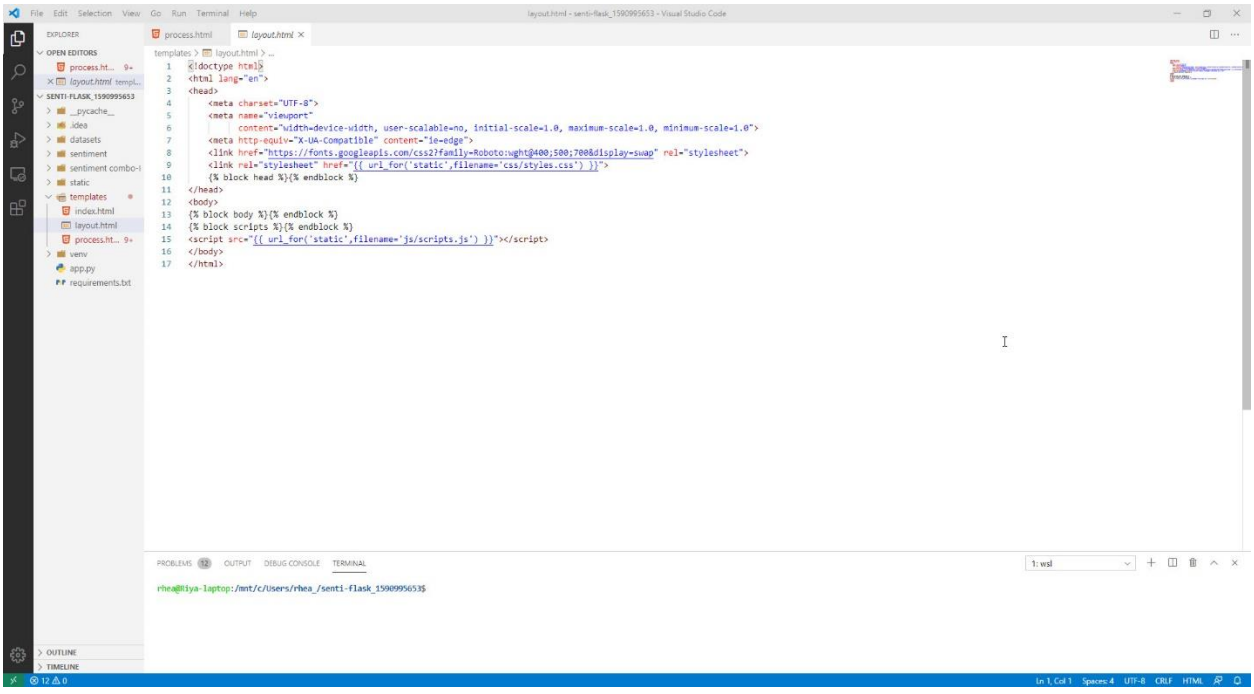
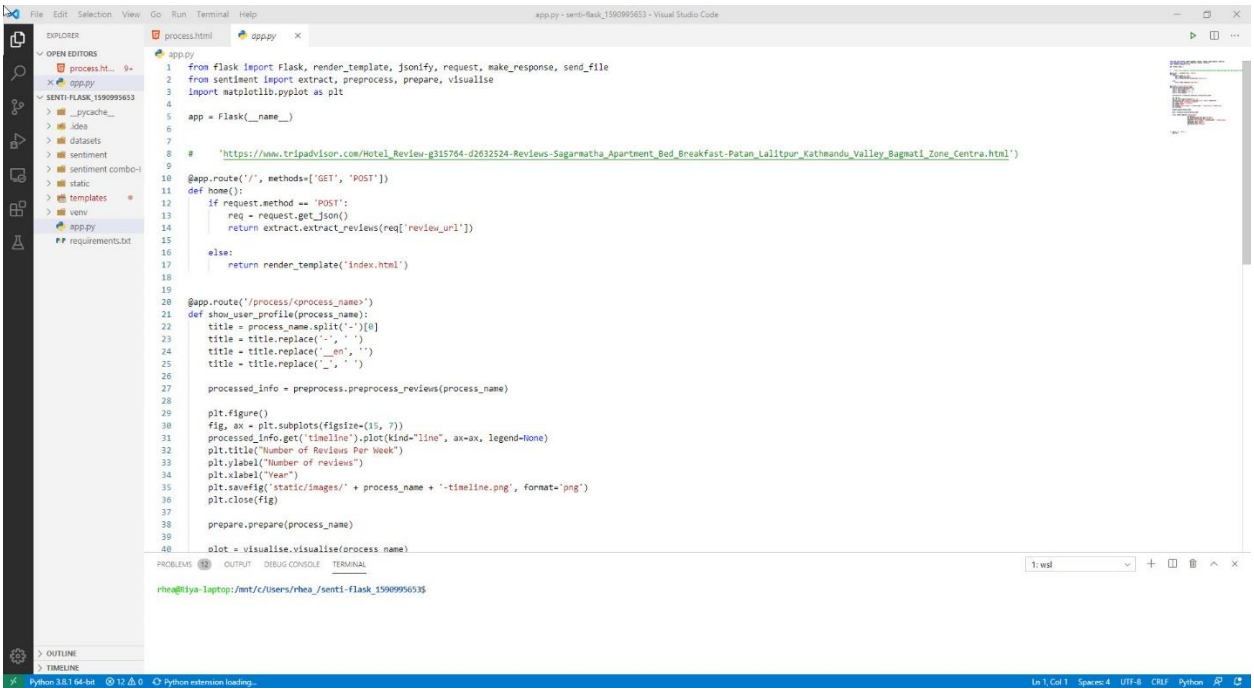


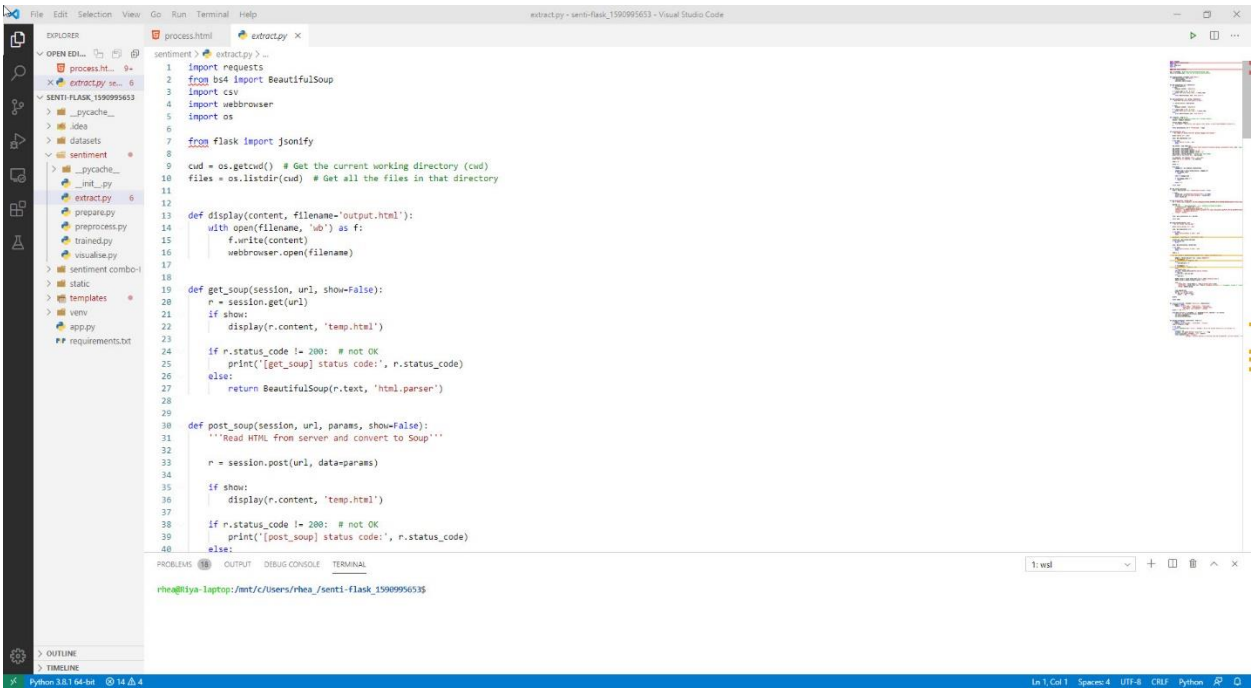

*Figure 86. The front-end code of the system. fig(c)*

### 7.3.2 Sample Code for the back end



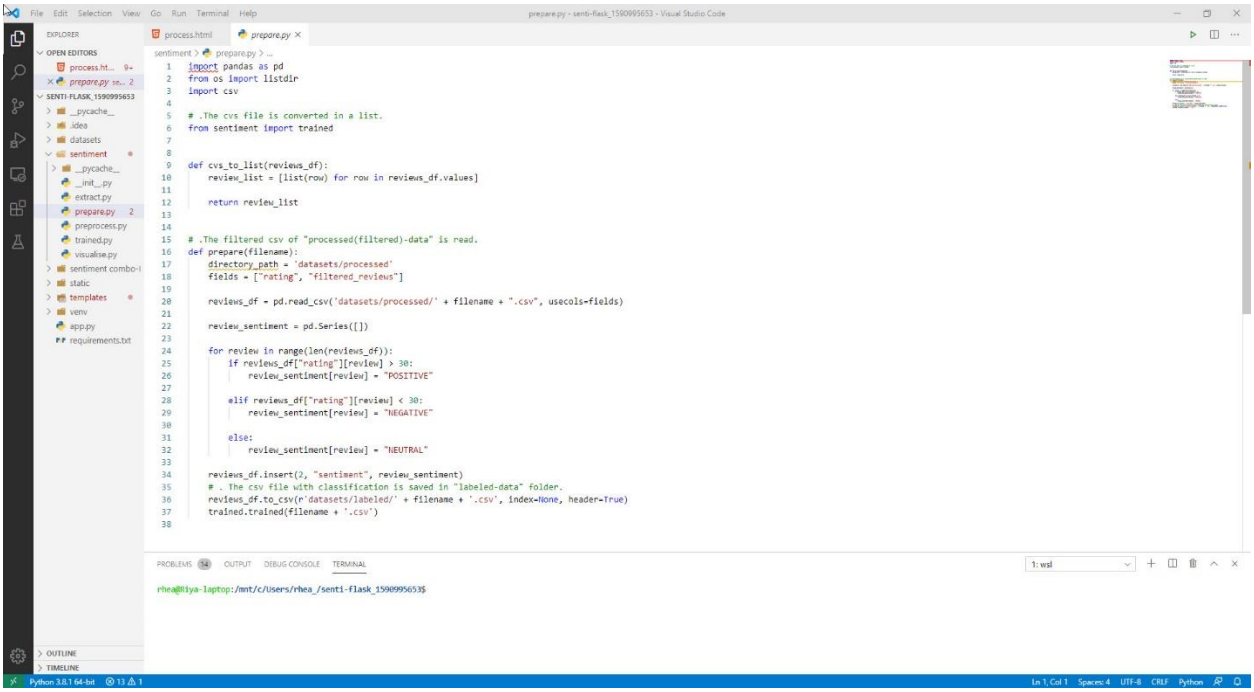
```
from flask import Flask, render_template, jsonify, request, make_response, send_file
from sentiment import extract, preprocess, prepare, visualise
import matplotlib.pyplot as plt

app = Flask(__name__)


#     'https://www.tripadvisor.com/Hotel_Review-g315764-d2632524-Reviews-Sagarmatha_Apartment_Bed_Breakfast-Patan_Lalitpur_Kathmandu_Valley_Bagmati_Zone_Centra.html')

@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':
        req = request.get_json()
        return extract.extract_reviews(req['review_url'])

    else:
        return render_template('index.html')


@app.route('/process/<process_name>')
def show_user_profile(process_name):
    title = process_name.split('-')[0]
    title = title.replace('-', ' ')
    title = title.replace('__en', '')
    title = title.replace('_', ' ')

    processed_info = preprocess.preprocess_reviews(process_name)

    plt.figure()
    fig, ax = plt.subplots(figsize=(15, 7))
    processed_info.get('timeline').plot(kind="line", ax=ax, legend=None)
    plt.title("Number of Reviews Per Week")
    plt.ylabel("Number of reviews")
    plt.xlabel("Year")
    plt.savefig('static/images/' + process_name + '-timeline.png', format='png')
    plt.close(fig)

    prepare.prepare(process_name)

    plot = visualise.visualise(process_name)
```

*Figure 87. The back-end code of the system. fig(a)*



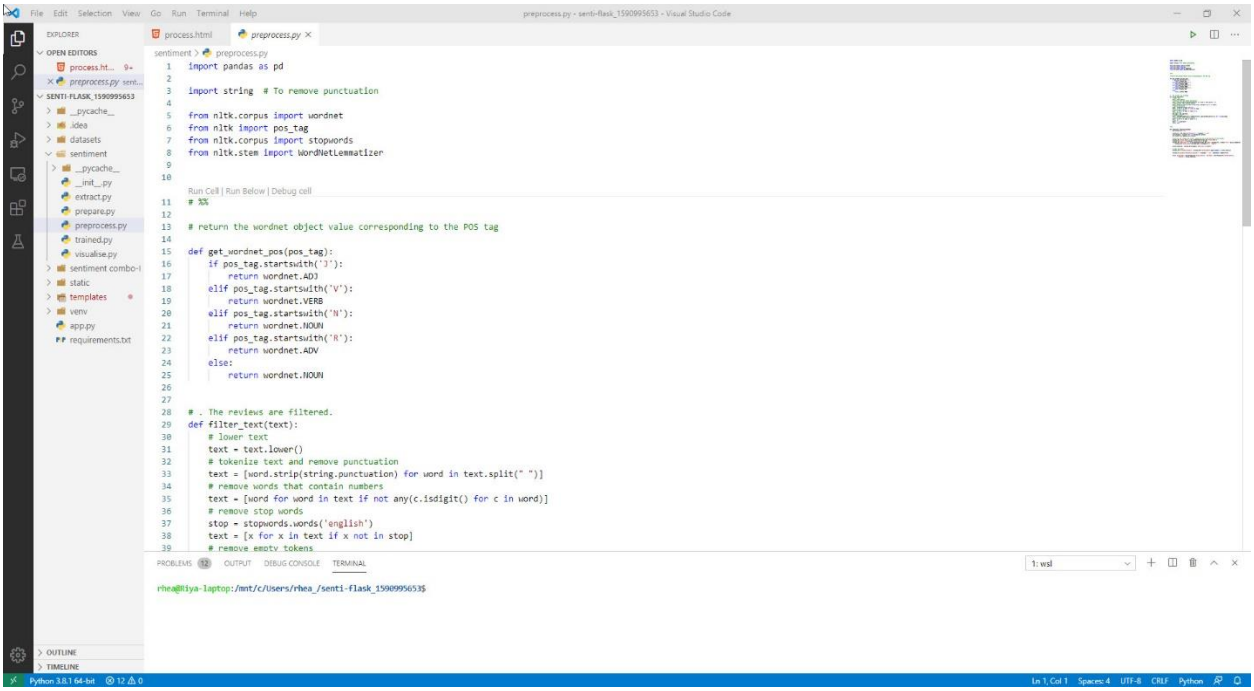
```
import requests
from bs4 import BeautifulSoup
import csv
import webbrowser
import os

from flask import jsonify

cwd = os.getcwd()  # Get the current working directory (cwd)
files = os.listdir(cwd)  # Get all the files in that directory


def display(content, filename='output.html'):
    with open(filename, 'wb') as f:
        f.write(content)
        webbrowser.open(filename)


def get_soup(session, url, show=False):
    r = session.get(url)
    if show:
        display(r.content, 'temp.html')

    if r.status_code != 200:  # not OK
        print('[get_soup] status code:', r.status_code)
    else:
        return BeautifulSoup(r.text, 'html.parser')


def post_soup(session, url, params, show=False):
    '''Read HTML from server and convert to Soup'''

    r = session.post(url, data=params)

    if show:
        display(r.content, 'temp.html')

    if r.status_code != 200:  # not OK
        print('[post_soup] status code:', r.status_code)
    else:
```

*Figure 88.The back -end code of the system. fig(b)*



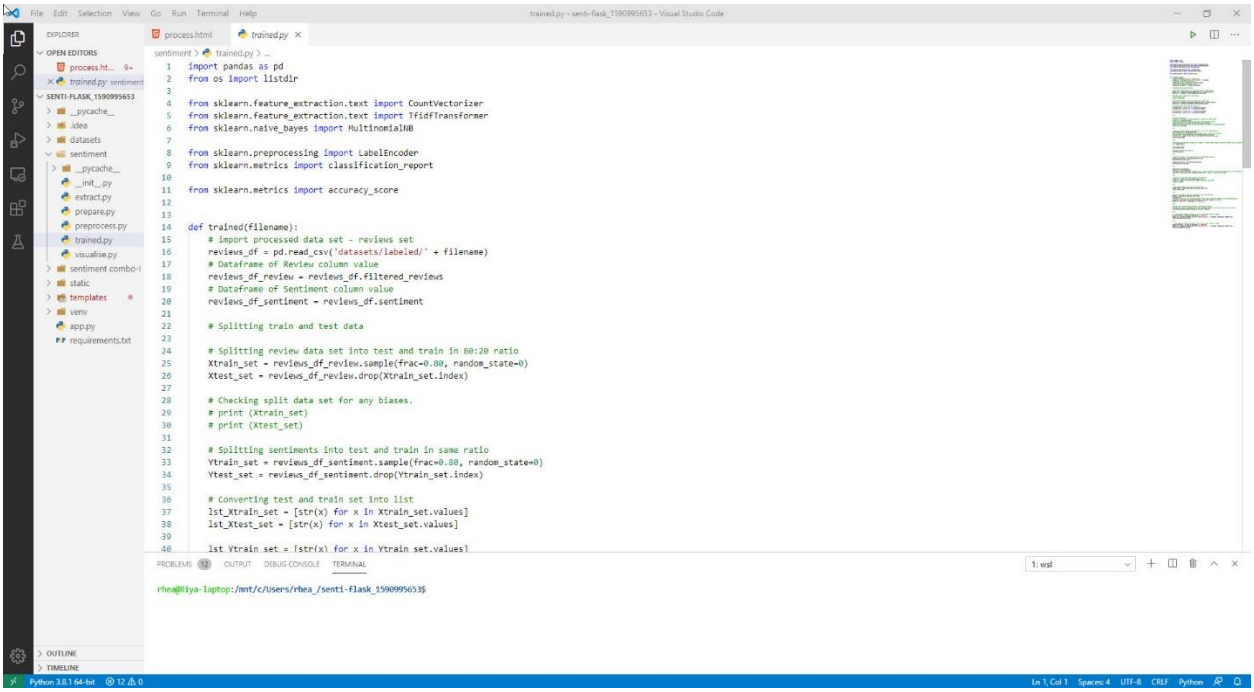
```
import pandas as pd
from os import listdir
import csv

# .The cvs file is converted in a list.
from sentiment import trained


def cvs_to_list(reviews_df):
    review_list = [list(row) for row in reviews_df.values]

    return review_list


# .The filtered csv of "processed(filtered)-data" is read.
def prepare(filename):
    directory_path = 'datasets/processed'
    fields = ["rating", "filtered_reviews"]

    reviews_df = pd.read_csv('datasets/processed/' + filename + ".csv", usecols=fields)

    review_sentiment = pd.Series([])

    for review in range(len(reviews_df)):
        if reviews_df["rating"][review] > 30:
            review_sentiment[review] = "POSITIVE"

        elif reviews_df["rating"][review] < 30:
            review_sentiment[review] = "NEGATIVE"

        else:
            review_sentiment[review] = "NEUTRAL"

    reviews_df.insert(2, "sentiment", review_sentiment)
    # . The csv file with classification is saved in "labeled-data" folder.
    reviews_df.to_csv(r'datasets/labeled/' + filename + '.csv', index=None, header=True)
    trained.trained(filename + '.csv')
```

*Figure 89. The back-end code of the system. fig(c)*



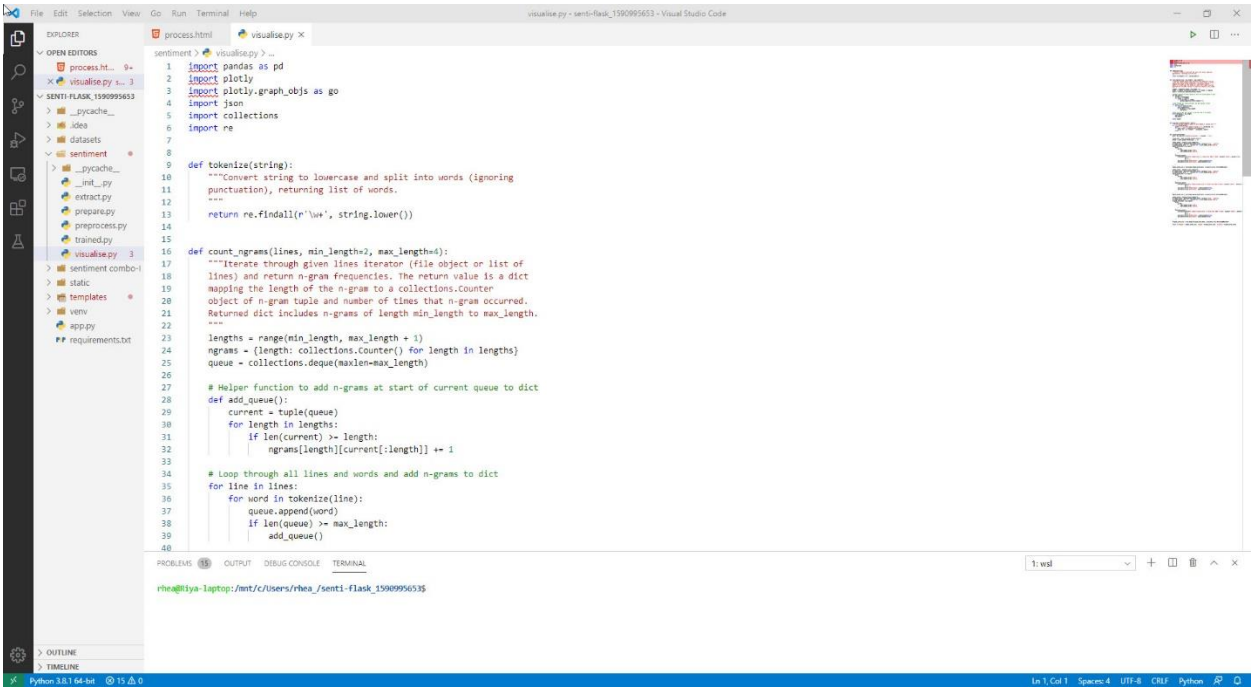
```
import pandas as pd

import string  # To remove punctuation

from nltk.corpus import wordnet
from nltk import pos_tag
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer


# %%

# return the wordnet object value corresponding to the POS tag

def get_wordnet_pos(pos_tag):
    if pos_tag.startswith('J'):
        return wordnet.ADJ
    elif pos_tag.startswith('V'):
        return wordnet.VERB
    elif pos_tag.startswith('N'):
        return wordnet.NOUN
    elif pos_tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN


# . The reviews are filtered.
def filter_text(text):
    # lower text
    text = text.lower()
    # tokenize text and remove punctuation
    text = [word.strip(string.punctuation) for word in text.split(" ")]
    # remove words that contain numbers
    text = [word for word in text if not any(c.isdigit() for c in word)]
    # remove stop words
    stop = stopwords.words('english')
    text = [x for x in text if x not in stop]
    # remove empty tokens
```

*Figure 90. The back-end code of the system. fig(d)*



```
import pandas as pd
from os import listdir

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB

from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report

from sklearn.metrics import accuracy_score


def trained(filename):
    # import processed data set - reviews set
    reviews_df = pd.read_csv('datasets/labeled/' + filename)
    # Dataframe of Review column value
    reviews_df_review = reviews_df.filtered_reviews
    # Dataframe of Sentiment column value
    reviews_df_sentiment = reviews_df.sentiment

    # Splitting train and test data

    # Splitting review data set into test and train in 80:20 ratio
    Xtrain_set = reviews_df_review.sample(frac=0.80, random_state=0)
    Xtest_set = reviews_df_review.drop(Xtrain_set.index)

    # Checking split data set for any biases.
    # print (Xtrain_set)
    # print (Xtest_set)

    # Splitting sentiments into test and train in same ratio
    Ytrain_set = reviews_df_sentiment.sample(frac=0.80, random_state=0)
    Ytest_set = reviews_df_sentiment.drop(Ytrain_set.index)

    # Converting test and train set into list
    lst_Xtrain_set = [str(x) for x in Xtrain_set.values]
    lst_Xtest_set = [str(x) for x in Xtest_set.values]

    lst_Ytrain_set = [str(x) for x in Ytrain_set.values]
```

*Figure 91. The back-end code of the system. fig(e)*



```
import pandas as pd
import plotly
import plotly.graph_objs as go
import json
import collections
import re


def tokenize(string):
    """Convert string to lowercase and split into words (ignoring
    punctuation), returning list of words.
    """
    return re.findall(r'\w+', string.lower())


def count_ngrams(lines, min_length=2, max_length=4):
    """Iterate through given lines iterator (file object or list of
    lines) and return n-gram frequencies. The return value is a dict
    mapping the length of the n-gram to a collections.Counter
    object of n-gram tuple and number of times that n-gram occurred.
    Returned dict includes n-grams of length min_length to max_length.
    """
    lengths = range(min_length, max_length + 1)
    ngrams = {length: collections.Counter() for length in lengths}
    queue = collections.deque(maxlen=max_length)

    # Helper function to add n-grams at start of current queue to dict
    def add_queue():
        current = tuple(queue)
        for length in lengths:
            if len(current) >= length:
                ngrams[length][current[:length]] += 1

    # Loop through all lines and words and add n-grams to dict
    for line in lines:
        for word in tokenize(line):
            queue.append(word)
            if len(queue) >= max_length:
                add_queue()
```

*Figure 92. The back-end code of the system. fig(f)*

## 7.4 Appendix D: Designs

### 7.4.1 Gantt Chart

| | | | |
|---|---|---|---|
| **Phase 1: Iteration 1** | | | |
| **Analyze** | | | |
| Analyze of Data Extraction | 6 days | Tue 11/5/19 | Tue 11/12/19 |
| Analyze of NLP | 5 days | Thu 11/7/19 | Wed 11/13/19 |
| Analyze of Sentiment Analysis | 9 days | Sun 11/10/19 | Wed 11/20/19 |
| **Requirements** | | | |
| Data Acquistition | 15 days | Thu 11/21/19 | Wed 12/11/19 |
| Text preprocessing | 7 days | Sat 12/7/19 | Mon 12/16/19 |
| **Analysis** | | | |
| Feature selection | 7 days | Tue 12/17/19 | Wed 12/25/19 |
| Feature extraction | 11 days | Sun 12/22/19 | Fri 1/3/20 |
| **Design** | | | |
| Sentiment Classification | 20 days | Sat 1/4/20 | Thu 1/30/20 |
| **Testing** | | | |
| Create model | 18 days | Wed 1/22/20 | Fri 2/14/20 |
| **Implementation** | | | |
| Polarity detection | 18 days | Fri 2/7/20 | Tue 3/3/20 |
| **Review** | | | |
| Validation and evaluation | 7 days | Sat 2/29/20 | Mon 3/9/20 |
| **Phase 2: Iteration 2** | | | |
| **Design** | | | |
| Classification for bigrams | 5 days | Tue 3/10/20 | Mon 3/16/20 |
| **Testing** | | | |
| Create new model with new data for bigrams | 4 days | Fri 3/13/20 | Wed 3/18/20 |
| **Implementation** | | | |
| Polarity detection for bigrams | 3 days | Mon 3/16/20 | Wed 3/18/20 |
| **Review** | | | |
| Validation and evaluation | 4 days | Wed 3/18/20 | Mon 3/23/20 |
| **Phase 3: Iteration 3** | | | |
| **Design** | | | |
| Classification for trigrams | 3 days | Mon 3/23/20 | Wed 3/25/20 |
| **Testing** | | | |
| Create model with new data for trigrams | 2 days | Thu 3/26/20 | Fri 3/27/20 |
| **Implementation** | | | |
| Polarity detection for trigrams | 2 days | Mon 3/30/20 | Tue 3/31/20 |
| **Review** | | | |
| Integration and Test Support | 2 days | Wed 4/1/20 | Thu 4/2/20 |
| **Deployment** | | | |
| Plan Deployment | 3 days | Fri 4/3/20 | Tue 4/7/20 |
| Develop Support Material | 2 days | Wed 4/8/20 | Thu 4/9/20 |
| Produce Deployment Unit | 4 days | Wed 4/8/20 | Mon 4/13/20 |
| **Maintenance** | | | |
| Manage Acceptance test ( At Development Site ) | 3 days | Sat 4/11/20 | Tue 4/14/20 |
| Manage Acceptance test ( At Installation Site ) | 3 days | Sun 4/12/20 | Tue 4/14/20 |
| **Documentation** | | | |
| Interim Report | 43 days | Sun 11/10/19 | Tue 1/7/20 |
| Final Report | 72 days | Fri 1/10/20 | Mon 4/20/20 |

*Figure 93. Gantt Chart of the project.*

### 7.4.2 Work breakdown Structure

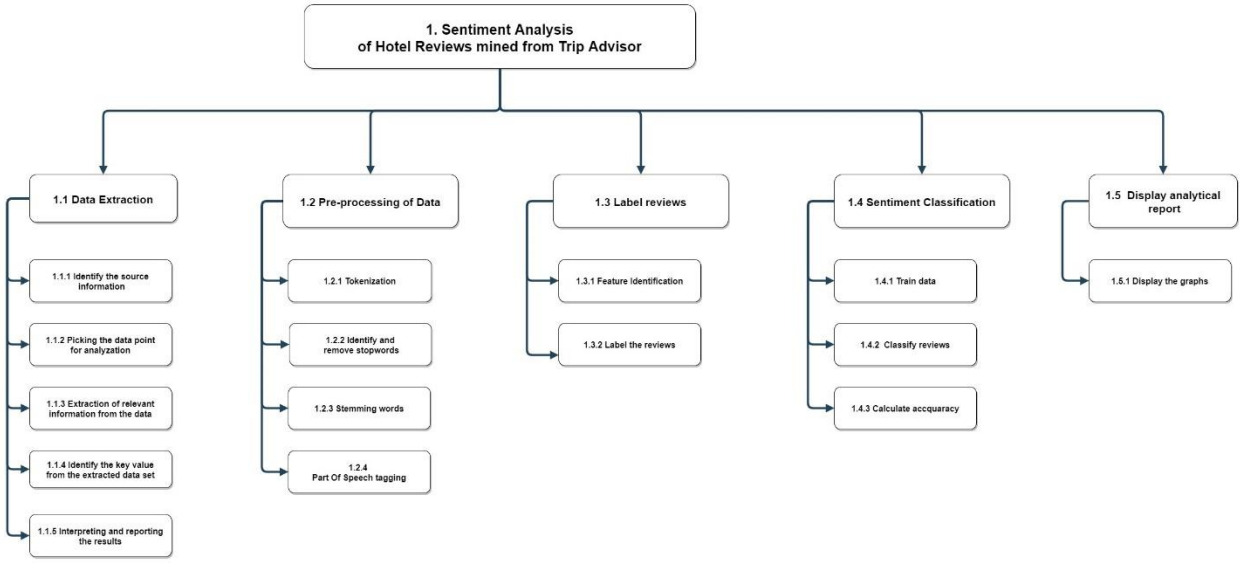



*Figure 94. Work break down structure of the system.*

### 7.4.3 Algorithms and Flowchart

- **Algorithm**

Step 1: Select the select the hotel from sentiment analysis and copy the link.

Step 2: Drop the link of the hotel in the “Sentiment Analysis of trip Advisor Reviews" website.

Step 3: Click on “Submit” button to start the data mining process.

Step 4: The reviews, review date and rating of the hotel are extracted.

Step 5: After the extraction of data, display message for continuation.

Step 6: Click on “Click here” to start the sentiment analysis process.

Step 7: The data is filtered.

Step 8: The filtered data is labelled.

Step 9: The model is trained.

Step 10: The model classifies the reviews.

Step 11: Display the textual and visual report of sentiment analysis.

Step 12: Click on “Download” to download the report.

- **Flowchart**

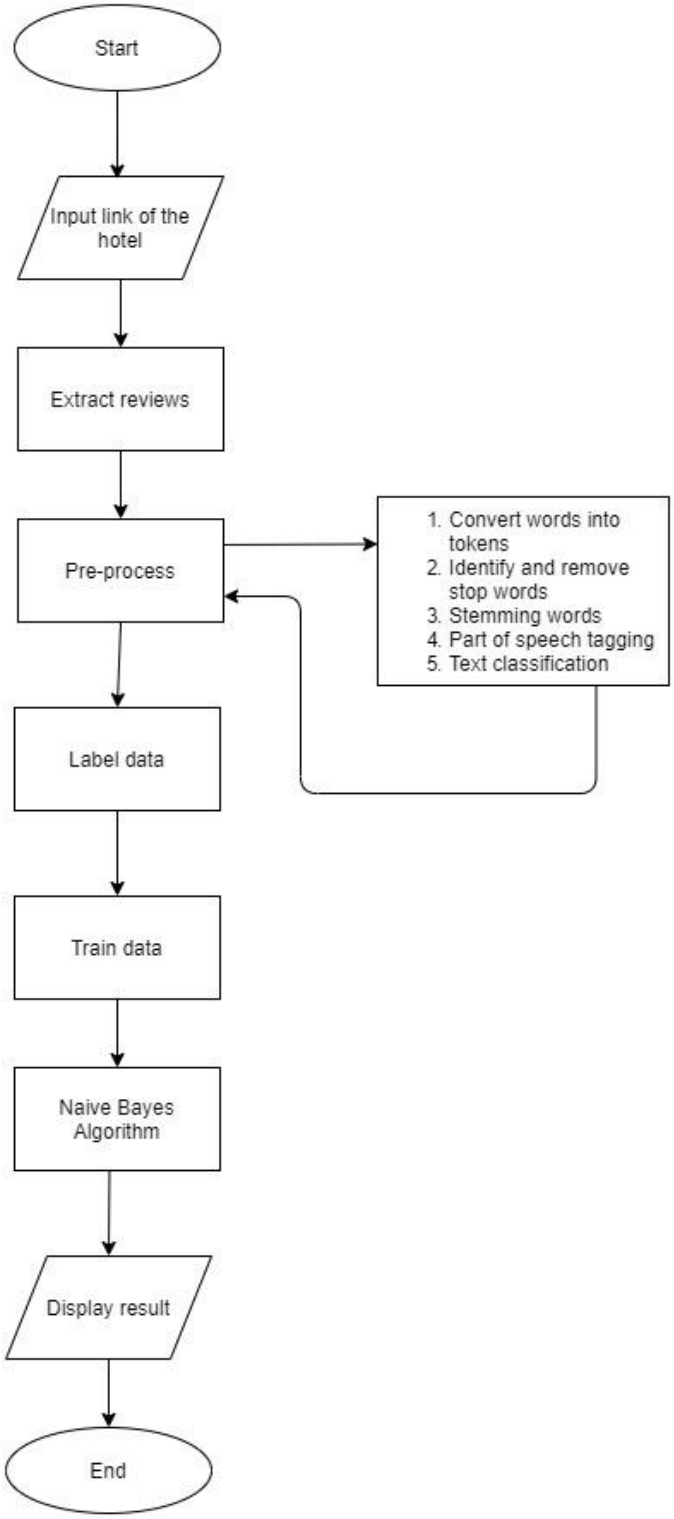



*Figure 95. Flowchart of the system.*

### 7.4.4 Data Flow Diagrams

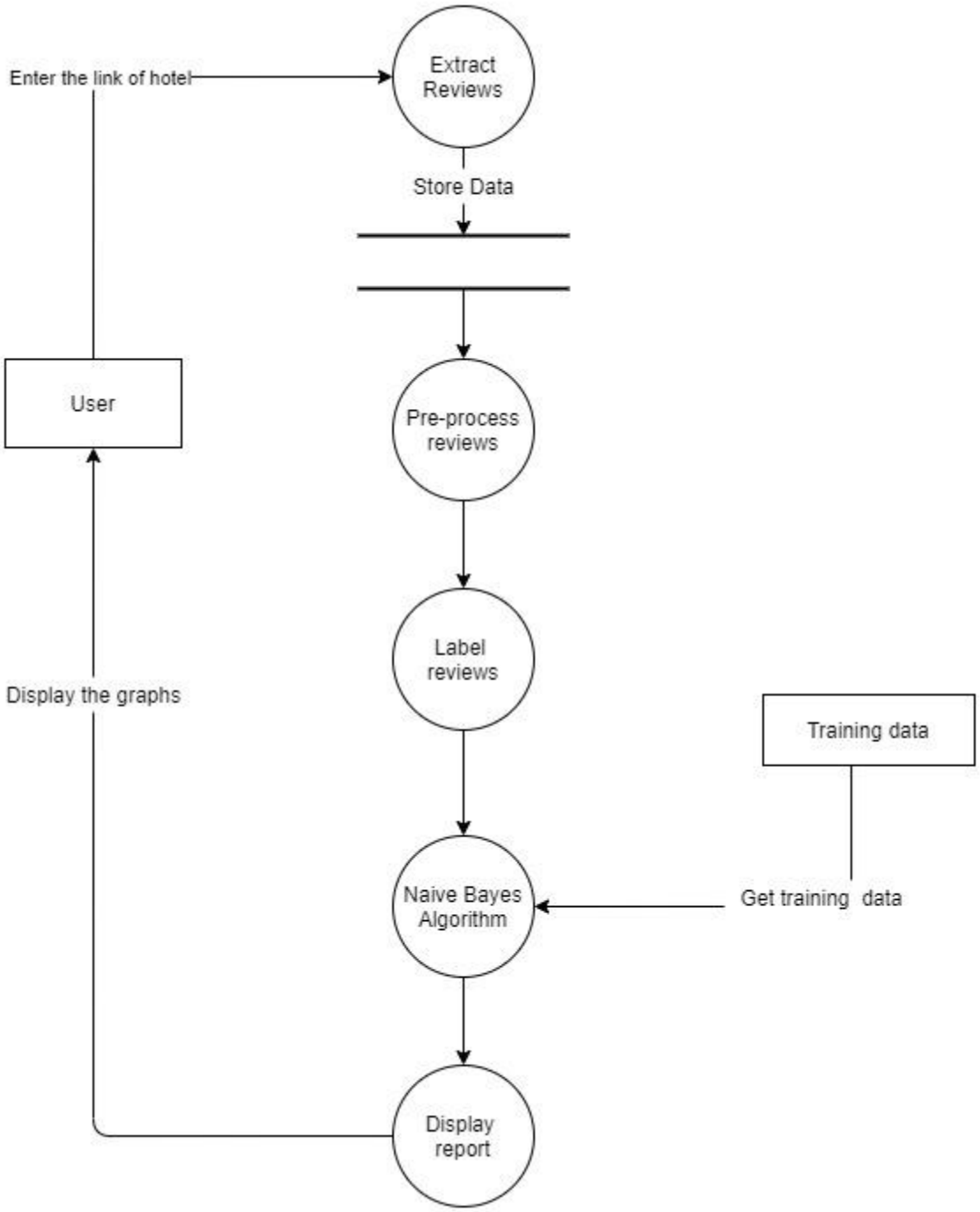



*Figure 96. DFD of the system.*

### 7.4.5 Individual Use Case

- **Use case of Input link:**

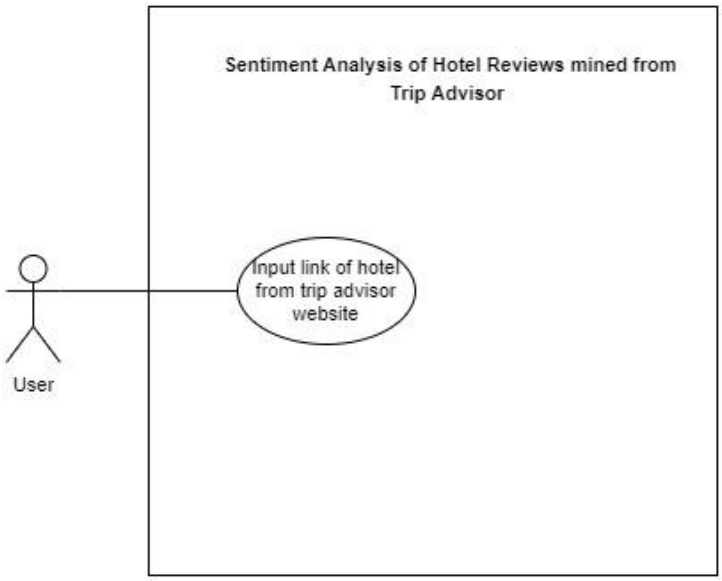



*Figure 97. Individual Use case: Input Link.*

- **Use case of data extraction:**

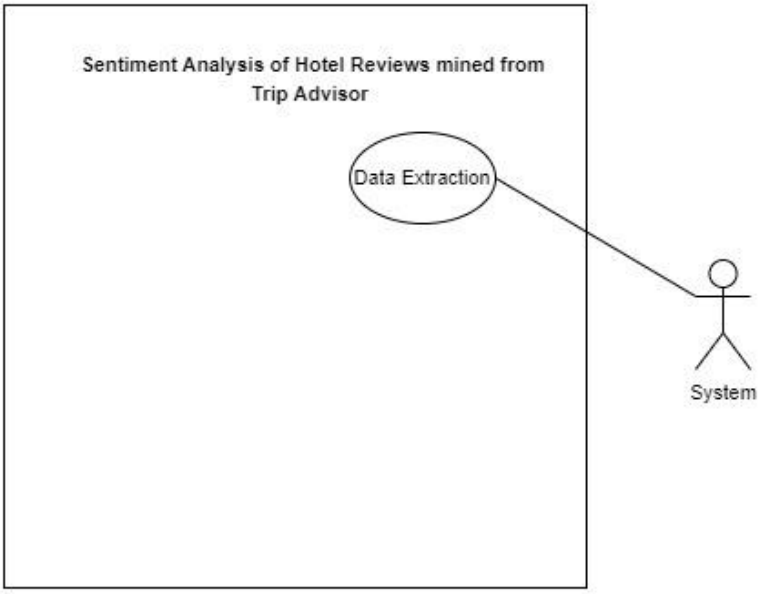



*Figure 98. Individual Use case: Data Extraction.*

- **Use case of filter review:**

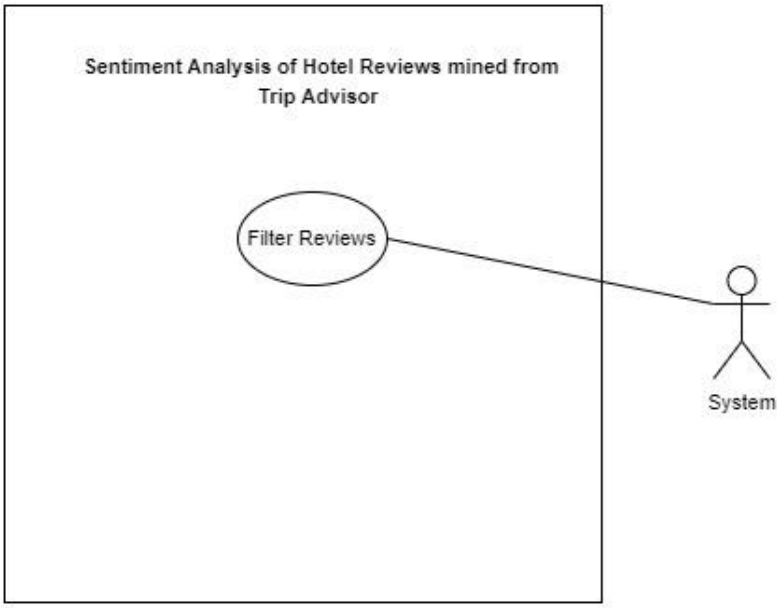



*Figure 99. Individual Use case: Filter Reviews.*

- **Use case of label review:**

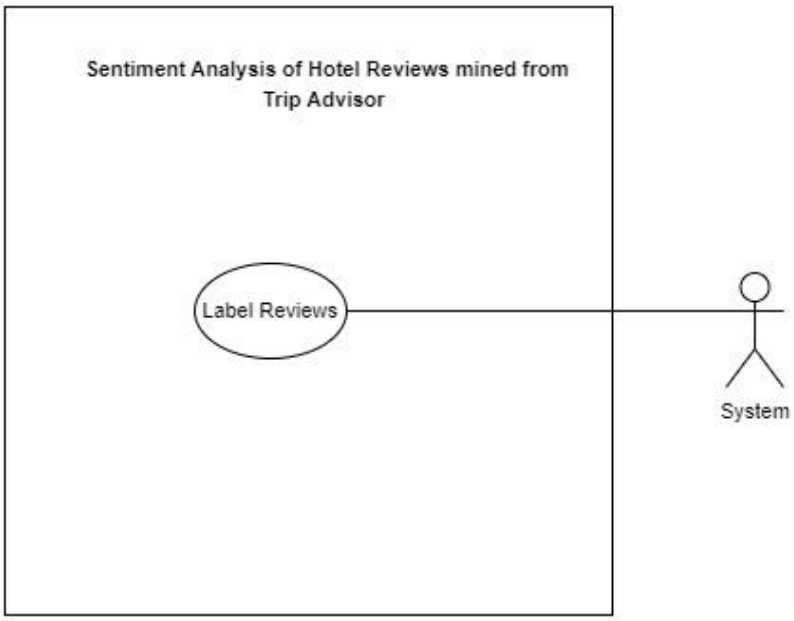



*Figure 100.Individual Use case: Label Review.*

- **Use case of classify review:**

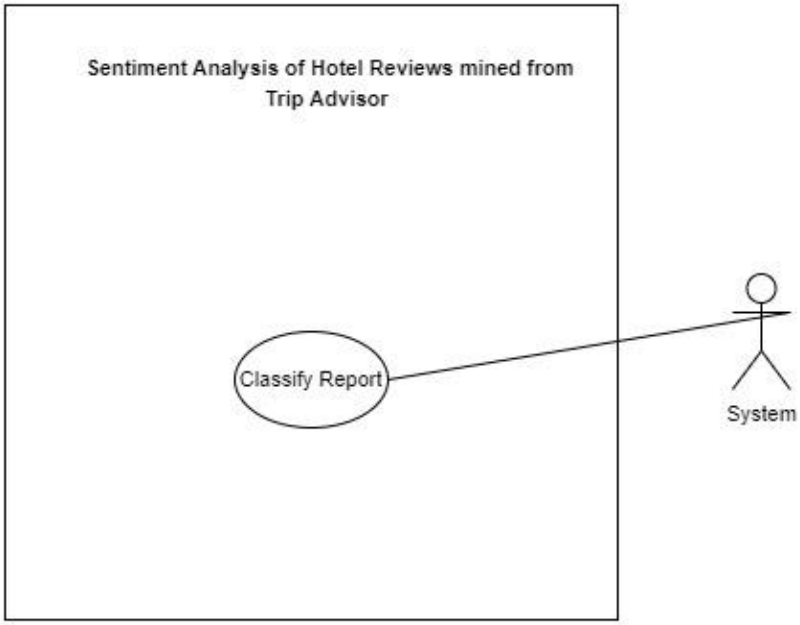



*Figure 101. Individual Use case: Classify Report.*

- **Use case of create report:**

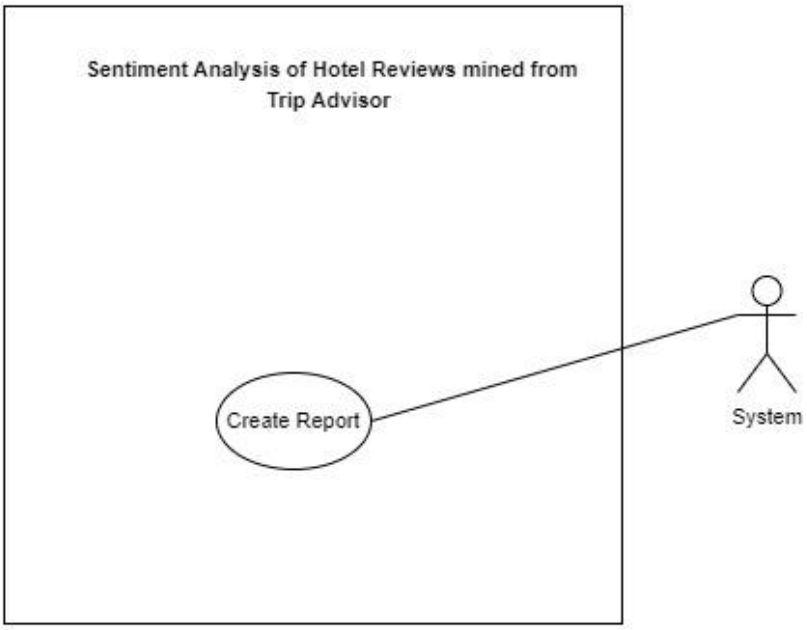



*Figure 102. Individual Use case: Create Report*

- **Use case of view report:**

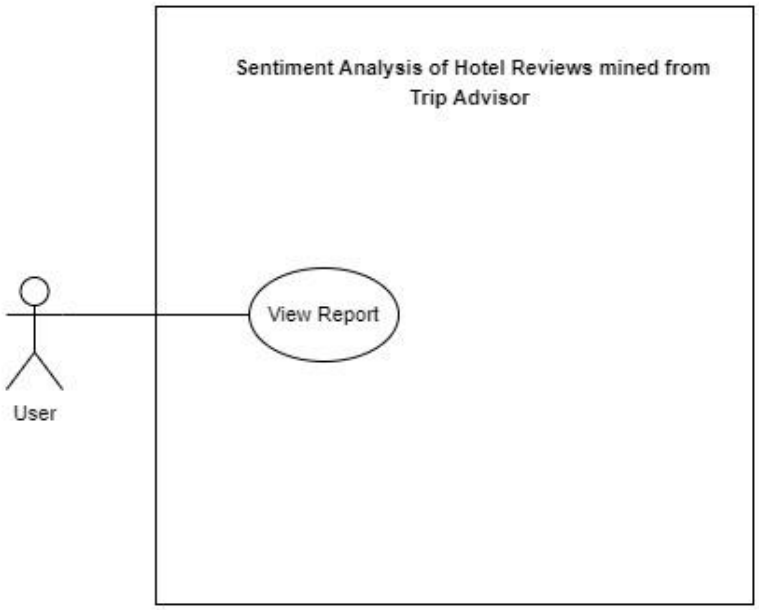



*Figure 103. Individual Use case: View Report.*

- **Use case of download report:**

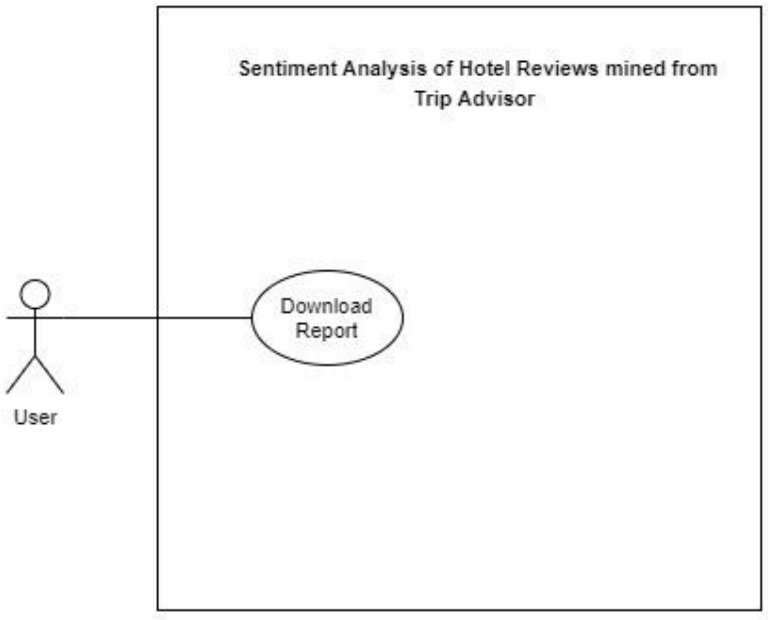



*Figure 104. Individual Use case: Download Report.*

### 7.4.6 Sequence Diagram

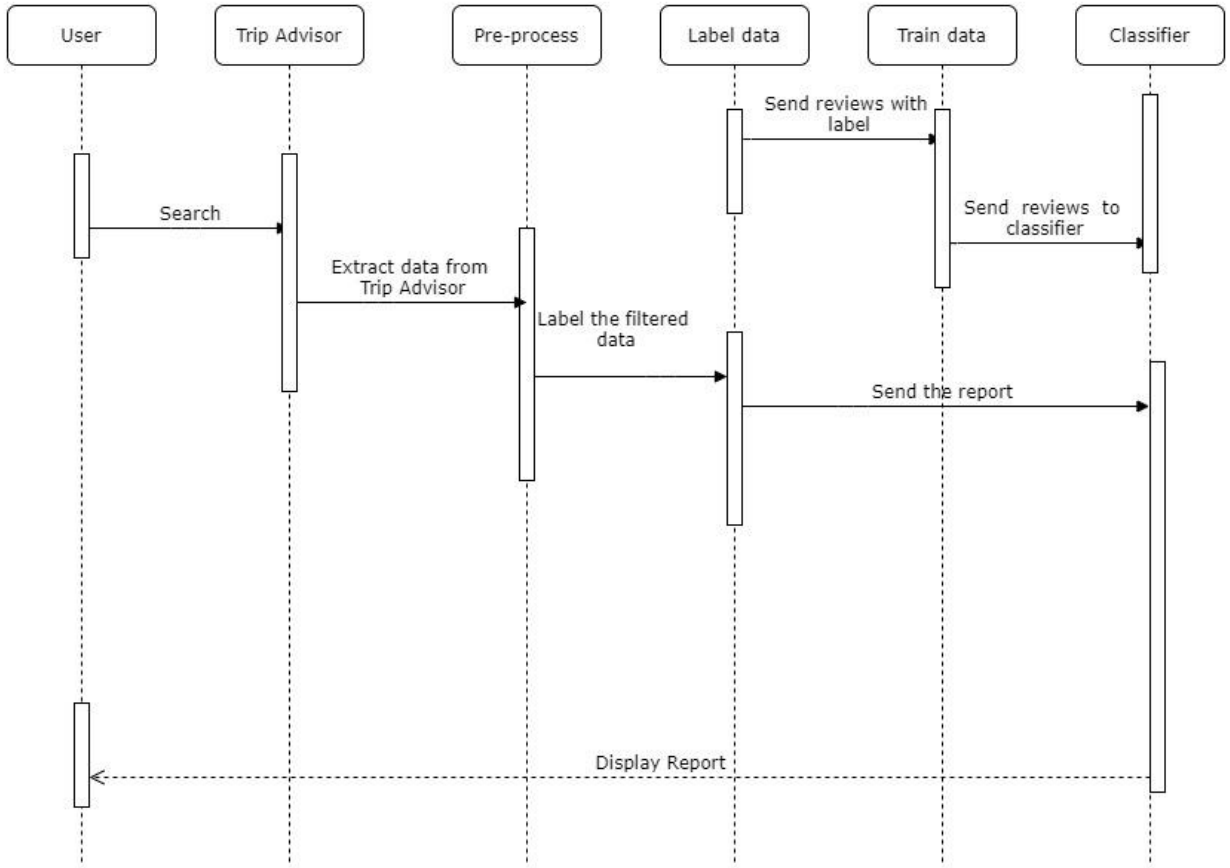



*Figure 105. Sequence diagram of the system.*

### 7.4.7 Activity Diagram

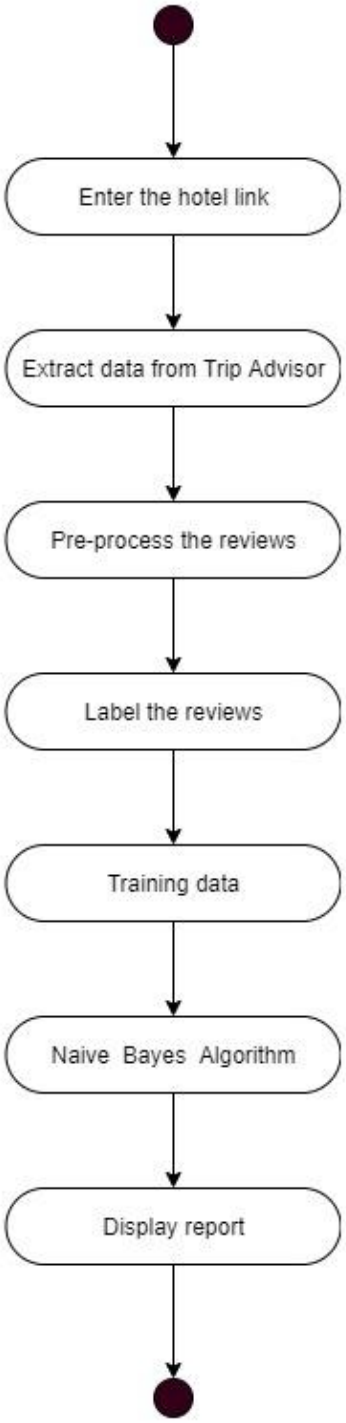



*Figure 106 Activity diagram of the system.*

### 7.4.8 Wireframe

- **Main page**

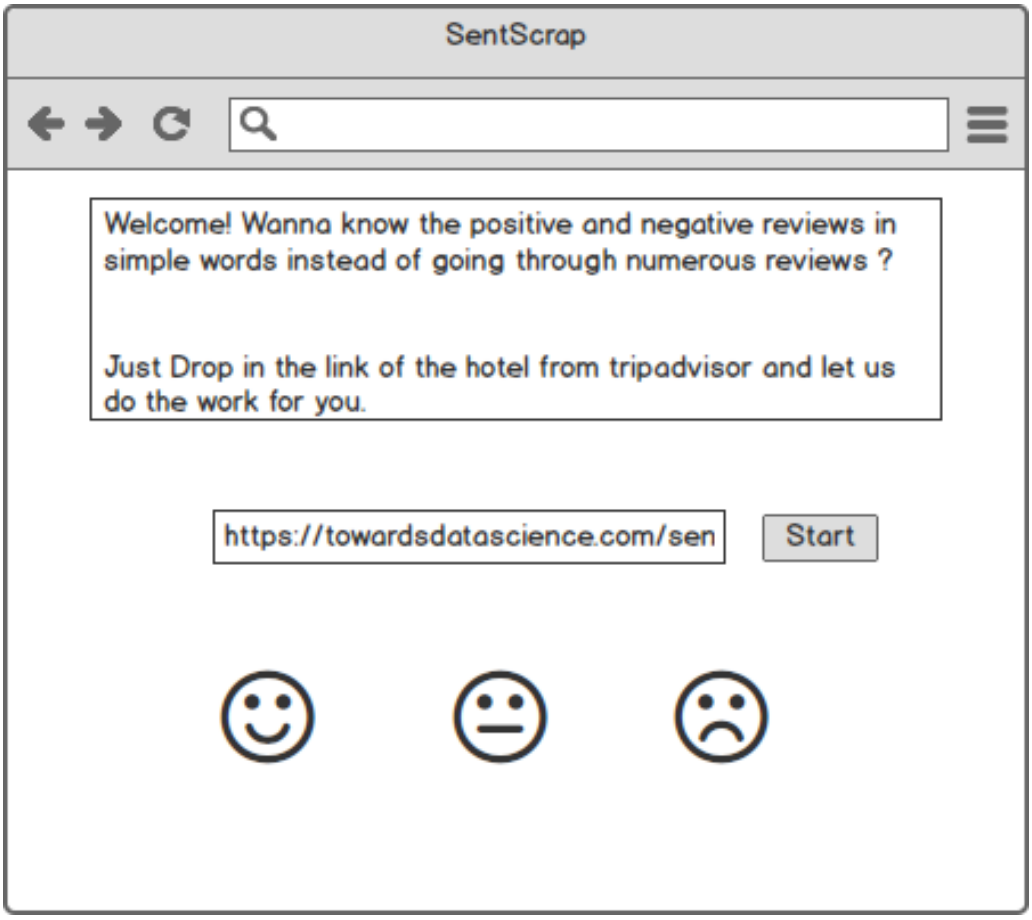



*Figure 107. Wireframe: Main page*

- **Result age**


*Figure 108. Wireframe: Result page fig (a).*

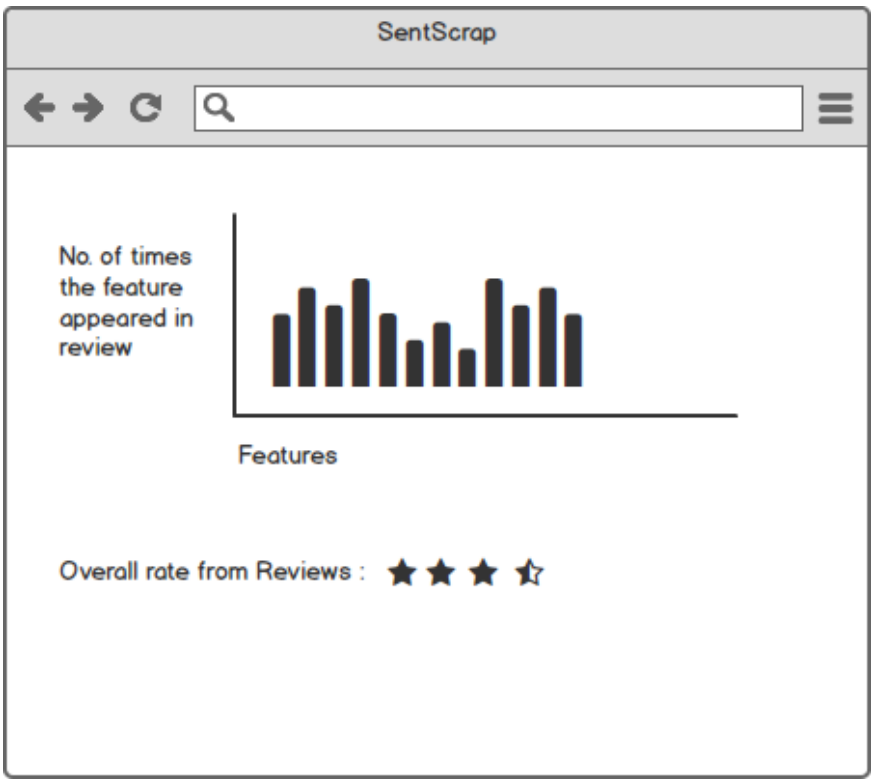



*Figure 109. Wireframe: Result page fig (b).*

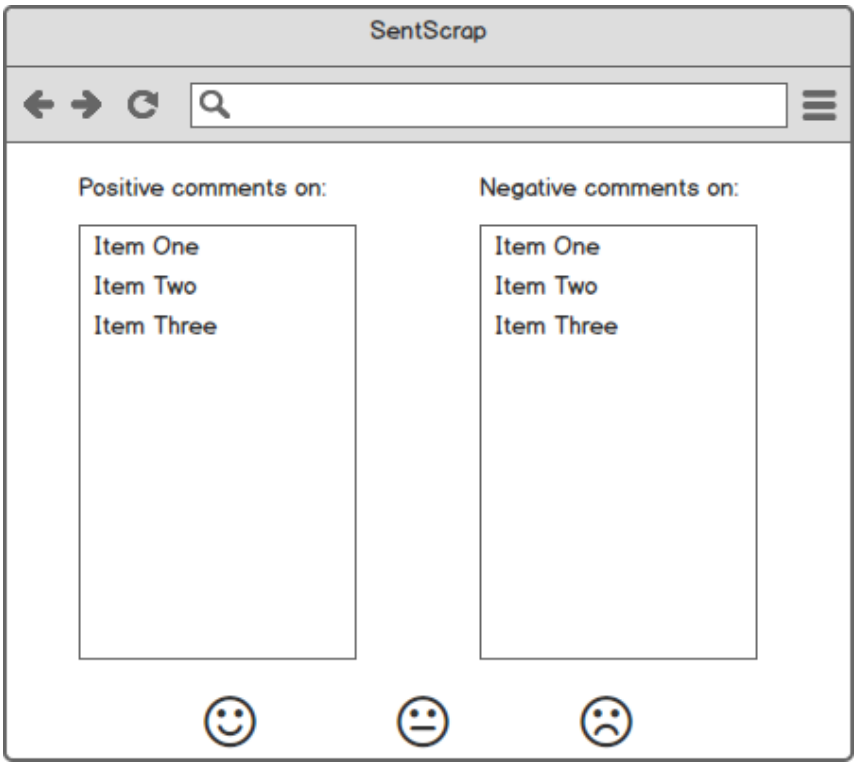



*Figure 110. Wireframe: Result page fig (c).*

## 7.5. Appendix E: Screenshots of the system

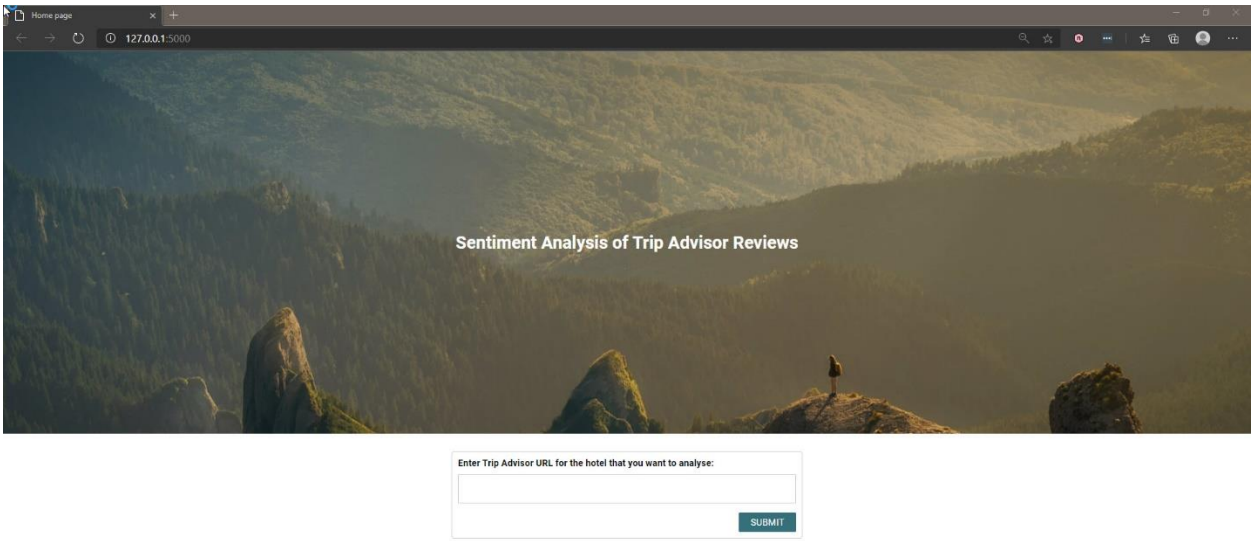



*Figure 111. The system UI. fig(a)*

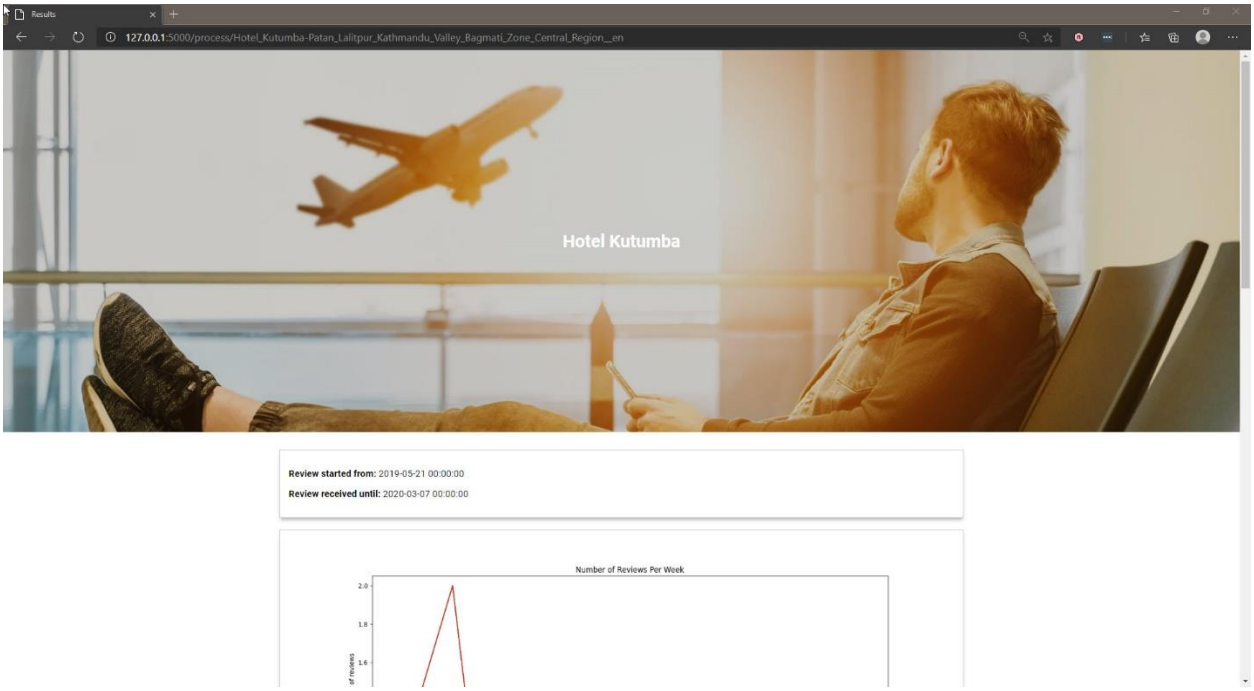



*Figure 112. The system UI. fig(b)*

## 7.6 Appendix F: User Manual

### Step 1: Drop the link of hotel from trip advisor website

1. Text field: Drop the link of the hotel.
2. “Submit button”: To start the process of data extraction.

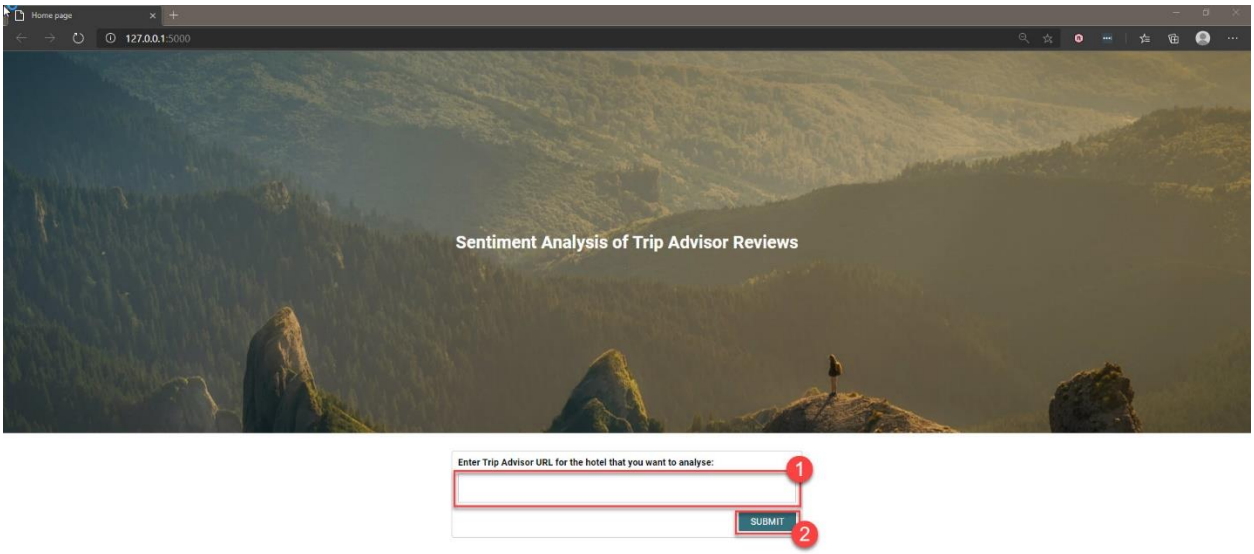



*Figure 113. User manual: Drop the link of hotel from trip advisor website.*

After the data extraction completed, a message is displayed to start the sentiment analysis process.

### Step 2: Start sentiment analysis.

1. Message: Click on “**Click Here**” to start the sentiment analysis.

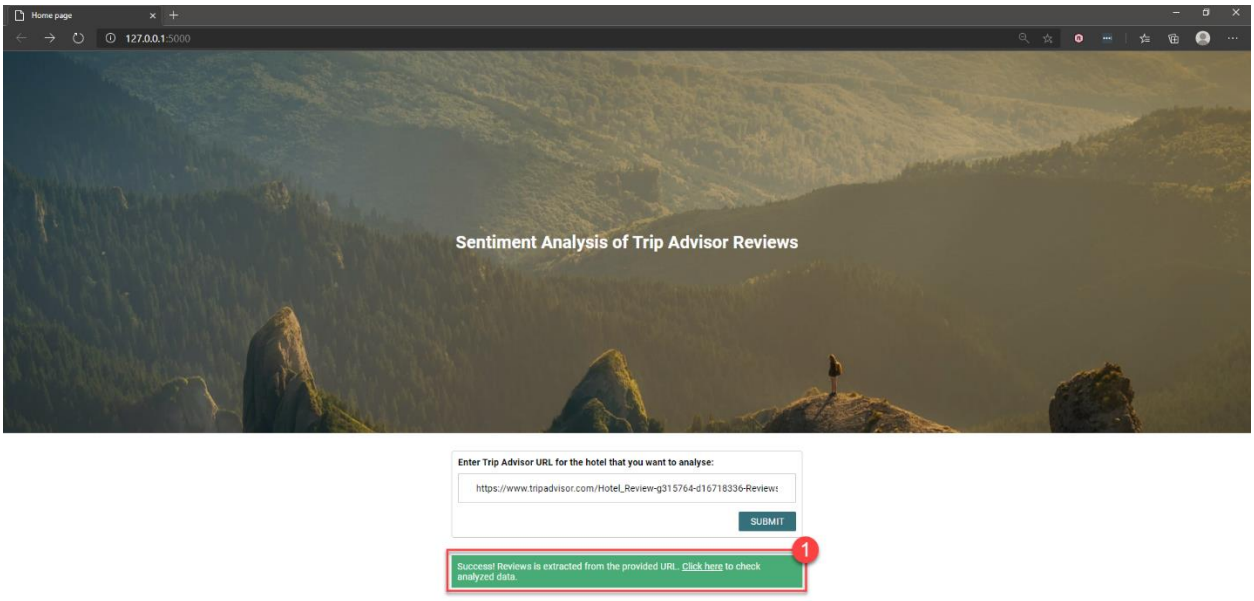



*Figure 114 User manual: Start sentiment analysis.*

## Step 3: View result of sentiment analysis.

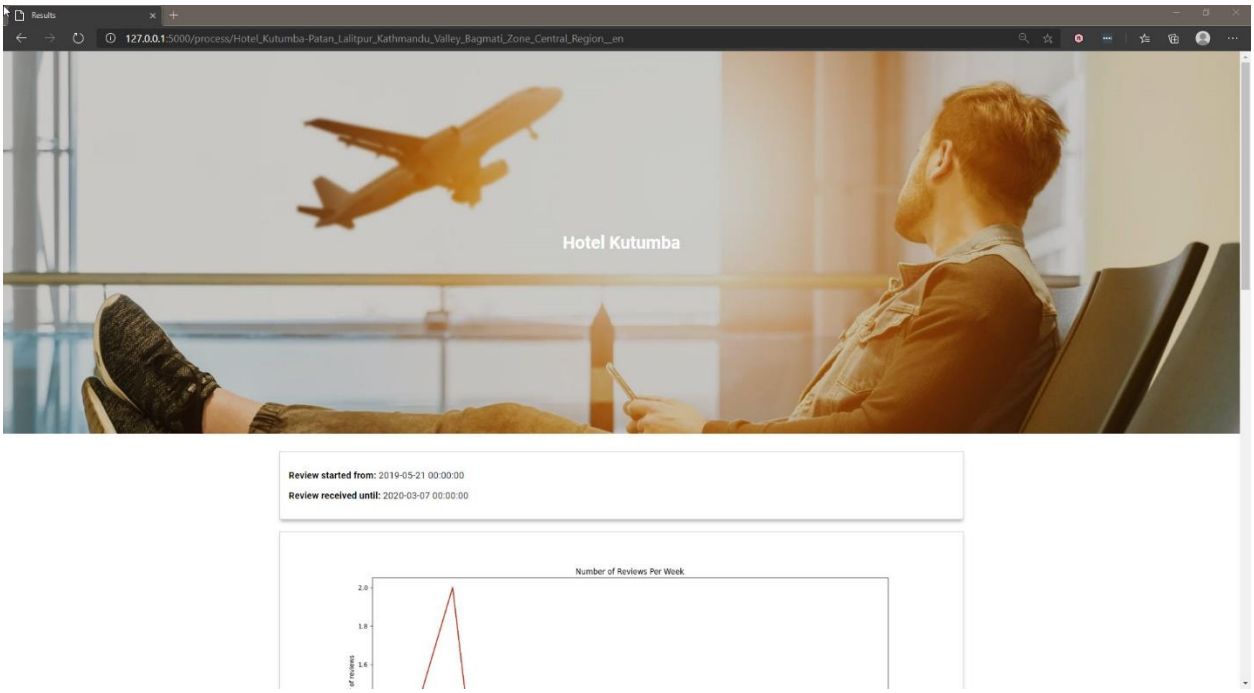



*Figure 115. User Manual: View result of sentiment analysis.*

**Step 4: Download the Sentiment report.**

1. “Download Analysis Report” button: To download the report of sentiment analysis.

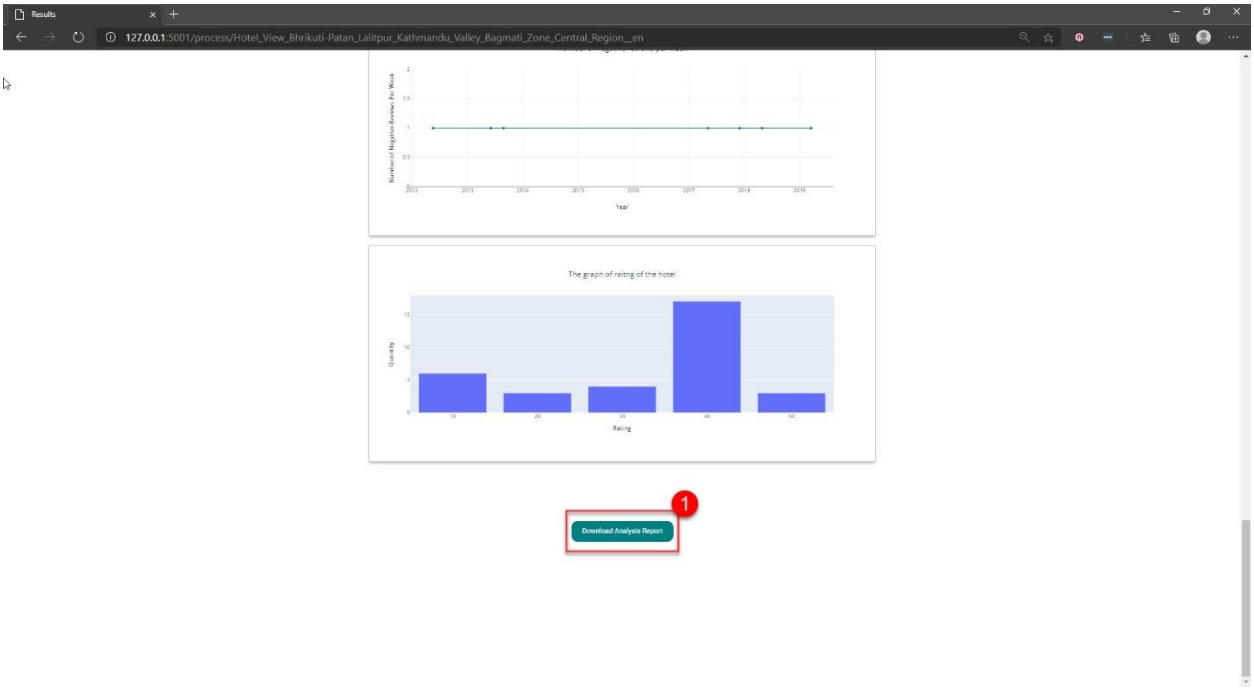



*Figure 116. User Manual: Download the sentiment report.*