****

**UNIVERSITY OF HERTFORDSHIRE**

School of Computer Science

MSc Data Science and Analytics with Advance Research

7COM1075 -0501-2020 Data Science and Analytics Masters Project

Student ID: 19023236

**Final Project Report**

**Sentiment Analysis of Customers Reviews** **on Products sold on different E-Commerce Portals**

**MSc Final Project Declaration**

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science and Analytics at the University of Hertfordshire (UH).

It is my own work except where indicated in the report.

I did not use human participants in my MSc Project*.* I hereby give permission for the report to be made available on the university website provided the source is acknowledged.

**Acknowledgement**

First and foremost, I want to express my gratitude to my...... for the unwavering support, direction, and insightful input during the project's duration.

Finally, I must express my heartfelt thanks to my mum, my partner, children and to my spouse for everything they have done for me by maintaining consistent support and ongoing encouragement throughout my years of study and research. This feat would not have been possible without you all.

Thank you very much...

**Table of Content**

Declaration 2

Acknowledgement 3

Abstract 6

CHAPTER 1: INTRODUCTION 7

1.1: Background 7

1.2 : Problem Statement 8

1.3 : Research Question 9

1.4 : Research Aim 10

1.5 : Objectives 10

1.6 : Project Plan 11

1.7 : Major Limitation 13

1.8 : Chapter Overview 15 CHAPTER 2 : LITERATURE REVIEW 17

2.1 : Introduction 17

CHAPTER 3 : RESEARCH METHODOLOGY 24

3.1 : Data Collection and Structuring 24

3.2 : Text Cleaning and Analysis 27

3.3 : Output 28

3.3.1 : Construction of Interface Between the Two Application 28

CHAPTER 4 : RESULT AND DISCUSSION 30

4.1 : Application One 30

4.2 : Application Two 32

CHAPTER 5 : CONCLUSION 35

REFERENCES 36

APPENDICES 38

Table 1: Project Plan and timeline 12

Table 2 : Products to Review 30

Figure 1 : Sample Output of the System Developed 11

Figure 2 : Initial Interface 30

Figure 3 : System Search File 31

Figure 4 : Final Output Interface 31

Figure 5 : List of Review in an excel Sheet 32

Figure 6 : Initial website Page 33

Figure 7 : Website Logo with Graph of Months vs Average Review 33

**Abstract**

The internet has become a source of knowledge, and this has opened a wide range of options for consumers to obtain this information quickly and efficiently. E-commerce portals are locations where goods and services are provided or where data is transmitted using an electronic network. This project is focused on how to perform sentimental analysis of customer

reviews of products sold on various e-commerce portals by developing an application that will allow users to do this in a specific website, which is easier and more efficient than going through different sites to review the product they want to buy, which is time consuming. This is accomplished by gathering data, organising the data, and analysing it in a single website.

**CHAPTER 1 – INTRODUCTION**

**1.1 BACKGROUND**

E-commerce is the way to go, and the last decade has seen a dramatic surge in the number of e-commerce platforms. Everything, from food to clothing, is purchased on the internet. E-commerce portals have acquired most of the market from brick-and-mortar retailers for electronic boxed products such as mobile phones, laptops, and televisions, where there is minimal opportunity for scrutinising the product before purchasing (Bakos, 2001). Prospective clients have therefore learned to rely on the reviews left by other customers and evaluate the characteristics and costs of various electronic devices sold on various e-commerce sites such as Best Buy, Walmart, and Amazon. Social media apps such as Twitter are also venues where one may look for product reviews before purchasing them to make informed judgments (Kalia et. al., 2018).

Consequently, customer reviews play a significant part in enabling customers to understand what they buy and make informed purchases. For example, if someone wants to buy a phone, he or she will typically seek up reviews for that phone on a variety of e-commerce portals. Furthermore, various clients will have differing perspectives on the phone. It is impossible to keep track of all these factors. This will not only take a long time, but it will also be incredibly complicated.

Because of technological advancements, it is now possible to bring the answers to all these questions in one place and compare or understand the product much more readily than previously, making it easier to understand a certain product. This can be accomplished by developing a website that contains and analyses all recent reviews, including both good and negative reviews of the product that a client plans to buy.

The goal of this project is to create an application that can retrieve and analyse product reviews (mobile phones) from numerous retail websites and show the information in a single platform, enabling a buyer to make an informed purchase decision.

The initiative will attempt to answer problems such as:

• How do different customers give feedback on certain products (mobile phones) and how do one acquire client sentiments on the mobile phone product?

Keeping these questions and the overarching goal in mind, I defined specific goals for the task. The goals are as follows:

1. Creating an app that can provide the following information for the products searched for:

* Display polarity (positive, negative, or neutral), i.e., the product's sentiment score
* Display the product's most recent ten reviews or testimonials.
* A chart displaying its average sentiment score over the previous six months

2. Writing programmes in Jupyter notebook (python) to retrieve consumer reviews from four websites: BestBuy, Walmart, Amazon, and Twitter. These are all existing reviews, and no new reviews will be collected through separate questionnaires.

3. Using Python to clean and analyse data collected for text analytics and sentiment analysis to gain relevant insights from feedback data.

4. Finally, the user interfaces for the two applications must be created (rendered). Django will be used for this, and HTML codes will be used.

When it comes to designing applications and web development, Django and Flask are both excellent choices to consider (Ghimire, 2020). Flask and Django are both Python-based frameworks, although the difference between them is that Flask is designed for micro applications and rapid development, whilst Django is designed for simple and straightforward projects. As a result, I made the decision to use Django instead of Flask, keeping the restrictions and goal in mind.

**1.2 Problem Statement**

As previously said, it is a challenge for prospective customers to gather information on a given product from a variety of various websites. With technology advancement, it is now feasible to gather all the answers in one location and compare or comprehend the product much more readily than it was before possible, saving time and money. Creating a website that provides all the most current product reviews, whether they are favourable, neutral, or negative in nature, for the product the client wishes to purchase is one method of achieving this goal.

Truly speaking, such an application is not now accessible elsewhere, and if it can be developed utilising data science technologies, it will be of tremendous assistance to potential clients in their selection of the appropriate product among the various possibilities. Customers will benefit from being able to obtain big amounts of data in a short period of time thanks to this feature. It will also analyse the reviews and provide it with a sentiment score, which represents the consumers' perspective of the product. Making a tool that allows customers to evaluate a product in a short amount of time would be extremely beneficial to them in the long run and will enable them to make educated decisions before purchasing the items.

**1.3 Research Questions**

The tool that will be created as part of this project will attempt to answer queries such as:

* How should the necessary information, i.e., reviews, be collected?
* From whom are these reviews going to be sourced?
* How will the data be assessed, given that it is entirely composed of language rather than numbers?

• What is the best way to understand the sentiment of the review?

• How is the interface going to be designed?

• What will the user interface look like? It must be simple to read for the benefit of the customers.

**1.4 Research Aim**

The following are the research aims of this work:

Create an application in which users may submit an excel file containing the names of the items they are searching for, and the programme will then retrieve the reviews for all the products in that file (mobile phones). Then sending them to the customer in an excel file so that they may utilise and review them.

Secondly, to develop an application that can provide the following information about the items that have been searched for:

• Indicator of polarity (either good, negative, or neutral), for example, the sentiment score of the product.

• Display the most recent ten customer reviews or testimonials for the product.

• A bar chart displaying its average sentiment score over the previous six months.

**1.5 Objectives**

Following the direction of these questions and the overall goal in mind, I divided my objectives into four phases that would guide me through the task and ultimately provide answers to the research questions. They are as follows:

* Data collection: Python is used to get current and existing customer reviews by gathering data from these websites: BestBuy, Walmart, Amazon, and Twitter.
* Organizing the Data: The information gathered will be unstructured at first. With the help of Python, it will be possible to turn it into organised form by organising the elements in rows and columns.
* Text cleaning and analysis: The structured data will be acquired, cleaned, and analysed using the Django programming language. Text analytics and sentiment analysis are referred to as "cleaning" and "analysis" in this context.
* Output: Creating the user interface of the two applications on Django using html. To make searching and understanding results on my application simple, the user interface will look like this:

A picture containing graphical user interface

Description automatically generated

Fig 1 – Sample Output of the System Developed

**1.6 Project Plan**

The project work will be performed as per the following timeline. The details of the plan have been shown as per the following Table.

**Table 1 – Project Plan and Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Number | Task Name | Details of the Task | Duration | Completion Date |
| 1 | Fetching Data from eBay | Development of the Code for extracting product related review data from the eBay portal | 1 Week | 07/07/2021 |
| 2 | Fetching Data from Walmart | Development of the Code for extracting product related review data from the Walmart portal | 1 Week | 14/07/202i |
| 3 | Fetching Data from Amazon | Development of the Code for extracting product related review data from the Amazon portal | 1 week | 21/07/2021 |
| 4 | Fetching Data from BestBuy | Development of the Code for extracting product related review data from the BestBuy portal | 1 week | 28/07/2021 |
| 5 | Development of the Tool for Scrapping Data | Integrating all codes to build an application which will upload the excel file and give the results in a csv form | 2 weeks | 04/08/2021 |
| 6 | Creating Sentiment Analysis Algorithm | Creating an algorithm which can perform sentiment analysis on the scrapped data | 2 weeks | 18/08/2021 |
| 7 | Creating whole App based on Python and Django | Creating the whole app which will display all relevant data. Python and Django will be used for this. | 4 weeks | 02/09/2021 |
| 8 | Final Report preparation | Preparing the Final Report | 2 weeks | 25/11/2021 |

**1.7 Major Limitation**

The result, which is in the form of a list have been save to a CSV file after executing the function for the required product name to be able to save this text data in a more organised manner. The use of Twitter data extraction does have one significant drawback, however: there is a restriction on the number of days that may be extracted from the tweets, i.e., Twitter does not enable one to retrieve tweets for more than one week. To begin with, the code was designed in such a way that the for loop would keep adding one day to the date parameter and continuing extracting data until the desired number of tweets was reached (value of the count argument). Nonetheless, because to the one-week time constraint, I was required to clearly state the start and finish dates of the extraction process.

Following that, I began writing code for the other websites, beginning with Amazon. Beautiful Soup is the library that is being used for this procedure, and it is being used to extract data from the area of the web page where the data is being extracted. By studying the web page, you can determine where the customer reviews are located on the website. By selecting the "Inspect" option on the right and clicking anywhere on the page, the codes that are hidden behind the web page will be seen. The codes over here are frequently contained within tags to make them easier to read. Now, when you hover your mouse over the place where the customer reviews are located, the code associated with that region will be highlighted. By copying and pasting this code into the beautiful soup code, and then extracting the reviews into empty lists, I was able to get the reviews. I'm seeking for three items from this review: the date of the review, the name of the mobile device, and the content of the review. As a result, I will have three empty lists from which I will extract the three data points mentioned above. Following this extraction, I put the contents of the lists into CSV files to keep them organised. Unstructured information is therefore retrieved and stored in a structured rows and columns manner in Microsoft Excel (CSV format). Following the extraction of Amazon reviews, I used the identical techniques to the extraction of Walmart reviews, which is when I encountered an issue. Different web sites are constructed in a variety of ways, which created a dilemma. For each website, the position of customer reviews may or may not be the same as for the others. In those nested tags with customer reviews, the class of the code associated to those reviews will alter because of this. As a result, utilising the identical codes for Amazon and Walmart will not help one get reviews from Best Buy or Walmart. To resolve this, the class code for reviews was looked for on each website individually and the codes were adjusted as necessary as a result.

Another problem was the occurrence of session time out problems. Web sites have a limited session period, and if the code takes an excessive amount of time to extract reviews from a web page, the code's session ends, and the code encounters a fatal error. To resolve this, I added an option (the time.sleep(1) parameter) that allows me to prolong the session time, i.e., have the page wait for a longer period. The same is seen in Figure 3.

As a result, I have evaluations for around 15 presently in-demand phone models from all four sites stored in separate CSV files, which I can access from any computer. When I was looking over the CSV files of reviews that had been generated, I saw that some of the reviews were being repeated. It's possible that this is because they were posted many times. Writing a function to delete duplicates from the output data was the solution to this problem. After deleting duplicates, each product has at least 300 reviews as output from each site, which is a significant amount of information. Each CSV file had three columns: the date of the tweet, the name of the product, and the review. The additional columns that resulted from the tweepy extraction were removed. To begin my study, I concatenate the data from all these CSV files into a single data frame, which will be used for Natural Language Processing (NLP).

I execute text analytics just on the twitter dataset, i.e., only on a part of the database developed, to verify that the fundamental logic behind the scripts is right, and since it is easier to deal with a smaller dataset first, I perform text analytics only on the twitter dataset. What I meant by data cleaning for textual data was the removal of sections of phrases that do not necessarily carry any meaning or information, as opposed to eliminating entire sentences. Numerical values and symbols such as punctuation marks (“;?!'), signs [@#$ percent &\*/>|] and integers do not transmit any information. The same may be said for regularly used words such as articles, prepositional phrases, pronouns, conjunctions, and so on, which may appear several times in a piece of writing yet have little informational value in and of themselves. They are referred to as "stop words" in the context of text analytics. As a result, these are the kind of terms that should be removed from any text analytics before they are performed. A module in Python called NLTK (Natural Language Toolkit) assists in accomplishing this by breaking down a given text into individual words (a process known as tokenisation) and deleting extraneous stop words and punctuation marks from the sentence. Also known as "lemmatisation," this procedure involves converting multiple versions of a word back to the original or root term. For instance, the words skip and skipping are derived from the root word skip. Lemmatisation would translate skipping to skip in this case, allowing the computer to accurately analyse the word and its frequency distribution. After that, there is sentiment analysis. In this section, I attempt to analyse the statement or review to comprehend the spirit behind it. Depending on the situation, it might be a positive review (for example: it's a good phone! ), a negative review (for example: it's a poor phone), or a neutral phrase (for e.g.: you can buy this phone at xyz site). Sentiment analysis allows you to categorise evaluations as either favourable, negative, or neutral based on their tone. The term "polarity" refers to the way in which a text can be measured as positive, negative, or neutral.

The Vader Lexicon package saved me from having to go through the time-consuming task of creating each one of these codes individually. Known as VADER (Valence Aware Dictionary and Sentiment Reasoner), it is a lexicon and rule-based sentiment analysis tool that is especially geared to the feelings expressed on social networking platforms. It has been designed in such a way that it automatically performs basic text cleaning tasks such as removing stop words and lemmatization, as well as sentiment analysis, which includes checking the polarity of reviews, rather than requiring the user to write codes for each of these tasks individually. Being sensitive to both the polarity (positive/negative) and intensity (strong) of emotion, VADER may be used to unlabelled text data without the need for further processing. It may be accessed as part of the NLTK package.

As a result, I was able to determine the polarity of customer reviews on Twitter by applying the Vader Lexicon model to my Twitter data. Now that the codes are functioning properly, I proceeded to utilise this code throughout the entire database in Python. However, it was at this point that I realised that Django would be a better choice if I wanted to do sentiment analysis on the entire dataset rather than just a sample of it. It took a long time to do sentiment analysis on such a large database, and it took a long time to analyse the results. I use Django to type the sentiment analysis code and to construct both the application's interfaces and the apps' interfaces themselves.

**1.8 Chapter Overview**

**Chapter 1: Introduction**

This section explains the fundamentals of the study. It offers information about the work's historical context, which explains why the work is essential. After that, it goes over the issue statement in detail. Following that, the research questions as well as the research's overall goal are outlined. Following that, the research goals are addressed, and finally the project plan is presented, which contains the name of the task, the specifics of the work that has been completed, and the anticipated completion date of the project. There is also a discussion of the project's limitations.

**Chapter 2: Review of the Literature**

This describes the topic's literature review, in which I discuss various similar works that have been published. The following works are connected to one another and will provide a perspective on the current work that has been done.

**Chapter 3: Research Methodology**

Specifically, this section explains the research methodology, as well as the instruments and strategies that were employed in the current work. It provides an overview of the work that has been completed.

**Chapter 4: Results and Discussion**

This section describes the findings that have been achieved as well as the final output that has been received.... You can look at screenshots of the software that has been built.

**CHAPTER TWO: LITERATURE REVIEW**

**2.1 Introduction**

I examined the backdrop of my research project, problem statement, research purpose and research goal in depth in the previous chapter, and I concluded with a discussion of the project plan. Specifically, I will address some of the relevant efforts that have been undertaken in this regard.

A robust framework, developed by Tucker et al. (2011), is presented to enrich the new product design process by dynamically capturing customer preference trends. According to the authors, the framework autonomously captures customer preference trends from publicly available product review data, which is abundant but grossly underutilised at the time of publication. The method described by the authors is effective in overcoming a significant challenge that has plagued the product design community for some time. Inability to guide the new product design process due to a lack of large-scale, realistic customer data and its meaningful interpretation. As stated by the authors, the test is derived from the conventional, frequent practise of obtaining customer assessments through interviews that are often expensive and time-consuming, while the amount of readily available information is typically on a small scale. A total of three processes are included in the system: recovering client audit texts, mining item highlight texts, and predicting future patterns of item inclination. A system like this is proposed by Tucker et al. (2011).

According to Jones et al. (2015), online views, often referred to as client audits, provide a wealth of information based on the client's point of view and are extremely valuable. According to the writers, by employing strategies to eliminate the client requirements, item designers will be more likely to appreciate the demands and requirements of their clients and to use this information to achieve company objectives. The paper Jones et al. (2015) describes the process of item include extraction and illustrates how this information may be used to construct a Product Feature Information Hierarchy based on the data. A technique for removing client audit inclination sentences from customer survey information is also presented, as is the feasibility of "planning" the client audit inclination sentences sentence to the 'Item Feature Information Hierarchy' by utilising regulated learning strategies to extract client audit inclination sentences. This investigation makes use of the My Starbucks Idea website as an online client audit webpage, eliminating and physically inspecting more than 5,100 customer survey phrases that have been stored on this website throughout the course of the investigation. A grasp of how an ordered need assessment might be recognised using unstructured customer survey information in the support item sector is gained by examining the detailed conclusions. The findings of this investigation suggest that customer requirements may be extracted from unstructured text and coordinated fundamentally by applying a Product Feature Information Hierarchy in conjunction with controlled learning classifiers, as seen in the following example.

According to Jeyapriya et al. (2015), social media is becoming increasingly popular on the internet. This media expertise assists individuals, businesses, and organisations in analysing information to make vital decisions. According to the authors, opinion mining, also known as sentiment analysis, is the process of developing a system to gather and analyse opinions about a product expressed in reviews, tweets, comments, and blog posts on the internet. To be used in critical applications such as opinion mining and summarization, sentiment classification is done automatically. According to the authors, to make worthwhile judgments in marketing analysis, it is necessary to efficiently implement sentiment categorization. They claim that reviews are composed of sentiment, which is represented in various ways across different domains, and that it is time-consuming to annotate data for each new area. According to the authors, the examination of online customer evaluations in which corporations are unable to determine exactly what customers liked and did not like in document-level and sentence-level opinion mining is a case in point. As a result, current research in opinion mining is focused on phrase-level opinion mining. It does a finer-grained analysis and directly examines the opinions expressed in internet testimonials. The suggested technique for analysing customer feedback is based on phrase-level analysis. Aspect-based opinion mining is another term for phrase-level opinion mining, which is equally well-known. It is used to extract the most essential characteristics of an item from item evaluations and to forecast the orientation of each aspect based on the most important aspects extracted. The proposed system executes aspect extraction utilising regular itemset mining in customer product reviews and mining opinions, whether they are favourable or negative, to improve the quality of the information. Using supervised learning algorithms, it is possible to identify the sentiment orientation of each aspect in customer evaluations.

According to Wang et al. (2017), internet reviews are acknowledged as an important source of item information when customers make purchasing decisions. Items survey information on the Internet, according to the author, is overly numerous and contains a large amount of needless data in this time of data over-burden. According to Wang et al. (2017), this makes it difficult for clients to find and participate in useful surveys. To resolve this issue, several online business websites provide catchphrases for item audits; however, these are established in advance and may distort customers' perceptions of the products. A bidirectional long short memory (LSTM) intermittent neural organisation is used in this study, according to the authors, to provide a programmed catchphrase extraction approach that is dependent on LSTM intermittent neural structure (RNN). The results of the studies conducted on item audits obtained through information slithering jd.com reveal that the suggested technique has an incredibly high precision in terms of catchphrase extraction when compared to other approaches. The reduction of human explanation efforts in web-based company can be facilitated by using this method.

According to Liu et al. (2020), the rapid development of information and communication technologies (ICT) has enabled customers to publish a vast volume of their worries and expectations online, which is generally recognised as a great resource for product designers. However, according to the author, only a tiny percentage of small and medium-sized firms (SMEs) have the capacity to exploit customer online data for design innovation, even though SMEs account for a considerable portion of national economies' development. This study uses manufacturing SMEs in the South Wales and Greater Manchester industrial areas of the United Kingdom as a concrete example, and investigates their potential motivations for using, as well as their knowledge of, big data-based customer analytics. The goal is to uncover the underlying reasons for the barriers that prevent them from making effective use of big data analytics in their businesses. According to the authors, an exploratory survey was carried out to determine the types of customer data they have, the storage methods they use, the volume of customer data they have, and so on. Following that, a carefully designed exploratory study was conducted to gain a better understanding of how SMEs perceive the relationships between customer data and product design, what they expect from big customer data analytics, and what they believe are the most significant challenges to SMEs in maximising the value of big customer data. Additionally, the authors have established a demonstration platform to show small and medium-sized enterprises (SMEs) how to automate the process of analysing customer online evaluations, as well as how to increase their capacity for consumer insights collection and strategic decision making. Finally, the findings from two focus groups reveal the many managerial and technological factors that SMEs must consider when contemplating deploying big data and consumer analytics systems. According to the authors, this study makes it easier to convince SMEs to embrace large amounts of customer data and implies that a cloud-based strategy may be the most effective method of providing access to big data analytics tools to small and medium-sized enterprises.

According to Wang et al. (2018), a better knowledge of customer requirements (CNs) and an effective translation of CNs into design parameters are essential for successful product design (DPs). Considering the recent trend toward diversification of CNs, the quick introduction of new goods, and the shortening of lead times, there is a rising need to accelerate the mapping of CNs to distribution points (DPs). According to the authors, this study presents a deep learning-based technique to increase the efficacy and efficiency of mapping CNs to DPs by exploiting product review data acquired from e-commerce websites. This approach, they claim, is based on data extracted from e-commerce websites. According to the authors, the findings demonstrate that the suggested strategy is capable of meeting consumer demands in a highly efficient manner.

Researchers Lee et al. (2017) set out to investigate how emotional expressions inherent in online hotel evaluations impact consumers' impressions of the helpfulness of the establishment. As stated by the authors, in this study, empirical data is analysed using a text-mining method in the context of hotels to investigate how review valence influences the perceived helpfulness of online hotel reviews and to examine the role of negative emotional expressions embedded in online consumer reviews with respect to perceived helpfulness. In accordance with the authors, the study collected 520,668 online reviews from 488 hotels in New York City (NYC) that were listed on the website Tripadvisor.com. A text mining algorithm and negative binomial regression were used to analyse the reviews, with 69,202 reviews (13.29 percent) receiving helpfulness votes being included in the analysis. This study reveals that when potential consumers read online hotel reviews for their future stay, they are more likely to see critical evaluations as being more useful than good reviews. In contrast, when intensely negative feelings were expressed, the degree of helpfulness regarding unfavourable ratings was reduced. When it comes to online customer evaluations, emotional expressions are prevalent. However, there has been surprisingly little research into the repercussions of emotional expressions on consumers' information processing and decision-making. Because of the nature of service, and because of the inseparability of production and consumption, which frequently makes it difficult to provide faultless service, customers tend to rely more on customer feedback to reduce any potential problems they may find later in the process. As a result, the authors claim that the study fills a gap by revealing that negative evaluations and emotional expressions play a more significant part in customers' information processing and decision-making.

According to Wang et al. (2009), with the popularisation and growth of the Internet as a new business approach, e-business Web sites are drawing the attention of the public and igniting heated competition among their competitors. The writers attempt to provide a solution to the topic of how to make Web sites indestructible while also assisting potential clients in making judgments. According to the authors, the only way forward is to conduct a thorough and accurate assessment of the sites. According to the authors, several conventional ways have been successful in accomplishing this goal, the majority of which are qualitative and subjective. In this study, I suggest a novel strategy that is based on the mining of public opinion. I can extract quantitative information from the customer reviews that have been uploaded so that I may evaluate various Web site indexes. The authors have employed a method known as MRA to increase the accuracy of the mining results they have obtained (mutual reinforcement approach). In contrast to many opinion mining methods that are focused on mining explicit factor-opinion pairs, the MRA can effectively mine implicit factor-opinion pairs based on a previously constructed association set. Most notable about my evaluation technique is that it is customer-oriented, making it more reliable and trustworthy than a traditional approach in many ways. They claim that an experimental investigation and evaluation have revealed that the findings obtained via the suggested technique are consistent with those obtained using the old approach but are more dependable.

The rising number of customers conducting research and making purchases for hospitality and travel services online, according to Han et al. (2021), has created new study possibilities for hospitality academics. Hospitality experts are becoming increasingly interested in understanding the internet travel economy, according to the authors of the paper in question. Even though many academics have sought to better understand the internet travel industry using analytical models, experiments, or survey gathering, these studies frequently fall short of capturing the entire complexity of the market in their findings. According to the authors, academics frequently rely on survey data or experiments because of the simplicity with which they may be collected or, in other cases, because of the difficulties in compiling online data. The authors of this study expect that it will provide hospitality researchers with the tools and approaches they need to supplement their conventional data sources with easily available information that customers use to make travel decisions in the future. The writers of this page have offered a guideline, as well as Python code, on how to obtain publicly available internet hotel data in the most efficient manner. The authors have concentrated their efforts on the acquisition of internet data from a variety of sources, including online travel agencies, review sites, and hotel brand websites. Some intriguing options for how these data sources may be used have been presented by the authors, who have also discussed some of the constraints that must be taken into consideration when analysing internet data.

Aiming to determine if online reviews (such as valence and volume), online advertising techniques (such as free delivery and discounts), as well as feelings from user reviews, might aid in the prediction of product sales, according to Chong et al. (2016). The authors attempted to develop a big data architecture and deployed Node.js agents for collecting Amazon.com pages using asynchronous input/output operations to collect data. According to them, the completed web crawling and scraping data sets were reprocessed for sentimental and neural network analysis after they had been completed. In this investigation, the neural network was used to determine which factors in the study were significant predictors of product sales, and which ones were not. Online reviews, online promotional methods, and online feelings are all predictive of product sales, according to the findings of the article. However, certain factors are more relevant predictors of product sales than others, according to the study. They discovered that the interactions between these variables become more significant than the effects of each variable on their respective variables. Examples include the fact that interactions between online volume and sentiments and discounts are more important than the separate predictors of discounts, emotions, and online volume on their own. These researchers developed a big data architecture, which they combined with emotional and neural network analysis, to make it easier to conduct future business research for forecasting product sales in an online context. Predictive analytic approaches (e.g., neural networks) were used to assess the variables, and the authors state that this technique is valuable for future data analysis in a large data environment, where prediction might have more practical consequences than statistical significance tests. Further research included in the report investigated how online reviews, feelings, and promotional techniques interacted with one another. According to the authors, they have primarily been investigated independently in prior research, however here, the authors have seen an interplay between them.

Mesarosh et. al. (2017) shares preliminary findings from their work on developing a framework for analysing internet ratings. Among the four key components of this framework, the authors have developed data retrieval, data processing, data analytics, and data visualisation, among other things. This module is responsible for scraping or streaming data from internet rating sites, according to the authors of the paper. According to the authors, the data processing module is responsible for the cleaning, filtering, and parsing of unstructured information. After the data has been prepared for computation, the data analytics module computes the needed analytics, and lastly, the data visualisation module shows the meaningful results to the consumers, according to the authors. A testbed based on popular internet rating data has been deployed, and the outcomes and findings of this study are reported in this publication, according to the authors.

**Chapter 3 – Research Methodology**

To make the concept of this project more understandable, it has been separated into two sections —

• Data collection and analysis

• Designing and developing the two apps

This will make it simpler to set goals, measure progress, and adhere to timetables in the future.   Before I could begin collecting data, I needed to identify which phone models would be used for the extraction and which would be ignored.   I have decided to proceed with extracting evaluations for phones that are listed on the top two pages of each website, as they will reflect the most popular and in-demand phones on the market. The current top 20 cell phone models have been researched on the internet, and the data about those phones has been retrieved. There were around 15 different phone models in all. Other colour or storage versions of these phone models do not need to be described precisely because the programmes for data extraction take care of that task in this instance. Some of these brands include the iPhone, Samsung, and Google, to name a few. The establishment of timeframes for each target was also required. It required a long time to develop the code for data collection using web extraction because the information needed to be gathered from four separate websites. Several weeks were allocated for this job alone, as specified in the project timetable (plan). The faults that happened throughout the extraction were readily corrected, and it only took the first few days of the week to complete the process. Three to four days were given for doing sentiment analysis in Python, followed by the implementation of Django code to complete the project. This is done to determine whether the core logic that underlies the codes operated smoothly and appropriately. Then one week was set out for getting to know Django, creating the code there, analysing the output, and dealing with any bugs that came up. The creation of the user interface was given 5 days of time on the schedule. These timescales are significantly different from those anticipated when the project was first initiated since, as the work progressed, several procedures were discovered to be far less time demanding than previously believed.

**3.1: Data Collection and Structuring**

This step was completed by researching techniques for getting evaluations from the four websites listed previously; this was done to obtain the information I desired. In this instance, the information being referenced to is the client feedback. So, basically, I needed to develop some Python code that would allow me to go to the customer review portion of the website and collect the reviews I want. This is sometimes referred to as "web scraping," which is the process of extracting information from a website in bulk.

Because of specific security constraints placed on the programme, it is not possible to just develop in Python for Twitter at this time, for example. Since web scraping from Twitter is a well-known technique, Twitter has created something known as a "developers account," which allows developers to construct their own projects using data extracted from the app. As a result, this is the technique that has been utilised to extract the reviews from the Twitter platform. To gain access to Twitter, a developer account was first formed at <https://developer.twitter.com/en/apply-for-access> which took no more than 15 minutes. Once the account is created, Twitter requests that the account users create a fake app for the user to be able to generate their own API key and security code. After that, I moved on to my jupyter notebook, where I used the "tweepy" library to connect to and retrieve reviews from the Twitter platform, among other things. After writing some typical code, I connect to the app by utilising the API key and secret codes that were produced earlier in the process. Now, I construct a function named "get data," which will accept four arguments, as listed in the following section. This for-loop performs most of the extraction work by utilising the values from these four inputs. Tweepy will extract the following information from each tweet containing the product name mentioned: tweet.id (tweet number), tweet.created at (date of the tweet), tweet.user.id str (user id of the person tweeting), tweet.full text (text of the tweet), and tweet.user id str (user id of the person tweeting) (the actual review). The four points of contention are as follows:

* Query – This is the name of the product that I'm trying to find.
* Alt – This is an empty list that will be populated with new values each time the for-loop is executed for a single tweet.
* Msg - This is an empty loop in which the value of "alt" is kept for each time the loop is executed.
* Count - Number of tweets I'm looking for.

To make use of the text data in a structured way after executing the function for the required product name, I save the result, which is in the form of a list, to an Excel spreadsheet.

For websites such as Amazon, Best Buy, and Walmart, web scraping on Python often follows the following steps:

1.By locating the URL of the website that will be scrapped.

2.By reviewing, i.e., read the page in detail

3. By locating the section of the site that has the data that one seeks. (For web scraping to be successful, code must be built that sends a request to the URL specified. Consequently, the site will allow one to view that page and locate the data.)

4. Write the program's code.

5. Execute the code and get the data

6. Ensure that the data is stored in the appropriate format.

Beautiful Soup is the name of the library that was utilised in this approach. Once a small code is written to connect to the web browser and access the URL, beautiful soup is used to extract data from the section of the page where the data is found. Reading the page i.e., inspecting the page, allows me to determine the placement of the data on the page. The codes behind a web page may be seen by using the control+shift+i keys on the keyboard or by right clicking anywhere on the page and selecting "Inspect" from the drop-down menu. The codes over here are frequently contained within tags to make them easier to read. As a result, I examine the page to determine beneath which tag the data I need to scrape is nested. This is simple to locate. Simply hover your cursor over the place where the customer reviews are located, and the code associated with that region will be highlighted. I then copy this code and put it into my beautiful soup code, where the reviews are extracted and placed into empty lists. I'm seeking for three items from this review: the date of the review, the name of the mobile device, and the content of the review. As a result, I will have three empty lists from which I will extract the three data points mentioned above. Following this extraction, I put the contents of the lists into CSV files to keep them organised. Unstructured information is therefore retrieved and stored in a structured rows and columns manner in Microsoft Excel (CSV format).

After employing both codes (tweepy and beautiful soup codes), I now have data for about fifteen currently in-demand phone models from all four sites, which I have stored in separate CSV files because of my efforts. When I was looking over the CSV files of reviews that had been generated, I saw that some of the reviews were being repeated. It's possible that this is because they were posted many times. Writing a function to delete duplicates from the output data was the solution to this problem. After deleting duplicates, each product has at least 300 reviews as output from each site, which is a significant amount of information. Each CSV file had three columns: the date of the tweet, the name of the product, and the review. The additional columns that resulted from the tweepy extraction were removed.

Following that, I use a simple piece of code to concatenate the data from all these CSV files into a data frame, from which I can begin my analysis or Natural Language Processing process (NLP).

**3.2: Text Cleaning and Analysis**

First, I code in Jupyter Notebook as a trial round before implementing the codes in Django, to ensure that the fundamental logic behind the codes is valid. To begin, I just utilise a portion of my database because it is convenient to test my scripts on smaller datasets first. Once the programmes in the Jupyter notebook are running properly, I execute sentiment analysis on the entire database in Django.

Textual data cleaning refers to the process of deleting sections of sentences that do not necessary communicate a message or provide any information. Numerical values and symbols such as punctuation marks (“;?!'), signs [@#$ percent &\*/>|] and integers do not transmit any information. The same may be said for regularly used words such as articles, prepositional phrases, pronouns, conjunctions, and so on, which may appear several times in a piece of writing yet have little informational value in and of themselves. They are referred to as "stop words" in the context of text analytics. As a result, these are the kind of terms that should be removed from any text analytics before they are performed. A programme in Python called NLTK (Natural Language Toolkit) assists me in accomplishing this by breaking down the provided text into distinct words (a process known as tokenisation) and deleting any extraneous stop words and punctuation marks. Also known as "lemmatisation," this procedure involves converting multiple versions of a word back to the original or root term. For instance, the words skip and skipping are derived from the root word skip. Lemmatisation would translate skipping to skip in this case, allowing the computer to accurately analyse the word and its frequency distribution. Then there's the sentiment analysis. In this section, I attempt to analyse the statement or review to determine the sentiment behind it. In some cases, it may be a positive review (for example, it's a great phone!). In others, it could be a negative review (for example, it is an awful phone). In some cases, it could be a neutral phrase (for e.g., you can buy this phone at xyz site). Sentiment analysis enables me to categorise evaluations as either favourable, negative, or neutral based on their overall tone. The term "polarity" refers to the way in which a text can be measured as positive, negative, or neutral.

Instead of going through the time-consuming process of developing these codes one by one, I turned to a programme called Vader Lexicon for assistance instead. Known as VADER (Valence Aware Dictionary and Sentiment Reasoner), it is a lexicon and rule-based sentiment analysis tool that is especially geared to the feelings expressed on social networking platforms. It has been designed in such a way that it automatically performs basic text cleaning tasks such as removing stop words and lemmatisation, as well as sentiment analysis, which includes checking the polarity of reviews, rather than requiring the user to write codes for each of these tasks individually. Being sensitive to both the polarity (positive/negative) and intensity (strong) of emotion, VADER may be used to unlabelled text data without the need for further processing. It may be accessed as part of the NLTK package.

As a result, I was able to determine the polarity of customer reviews by applying the Vader Lexicon model to my database. As soon as the programmes are running well and there are no mistakes, I move on to Django to enter in the sentiment analysis code and construct the interfaces for the two apps. "Django" is a free and open-source web framework designed by experienced developers that is based on the Python programming language. It helps one to concentrate on web development and writing their app without having to worry about anything else. This is followed by the fourth and last aim of this stage.

**3.3: Output**

**3.3.1: Construction of interfaces between the two apps**

My data base has been built in Python, and sentiment analysis has been performed using Python at the back end up to this point. I'm going to use HTML codes to create the interface for the front end of the application, and I'll be utilising the Django framework to do this. There is a Python framework called Flask that can be used to accomplish all of this, however Flask is designed for micro apps and quick development, whereas Django is designed for straightforward and basic projects. Consequently, keeping my future scope and purpose in mind, I chose to proceed using Django rather than using Flask. Django's codes may be thought of as five blocks of code, which is how they are organised.

When the post request is received from the interface, i.e., the name of an item, the code defined in the first block is responsible for retrieving the item's name and putting it in a variable. Using this variable, I attempt to search my database for reviews that are like the one that was entered by the user.

My database has a field for dates since the final application will display the most current reviews, as well as a bar chart showing the previous six months' worth of reviews. As a result, the scripts written in the second block are concerned with gathering feedback from customers during the previous six months.

Data cleaning and sentiment analysis on the reviews of the product specified by the user are performed in the third block, which results in the production of polarity scores. The Vader lexicon codes are contained inside this block.

Following that, HTML codes are generated to display the interface. I store these scripts in a file named template, where I style my page with Bootstrap, which I include in the code. Codes are being created for the interfaces of both apps, with the first app including a "Choose File" option and a "Submit" button, and the second app including a "Choose File" option and a "Submit" button. The second app is expected to provide an input box for the user to fill in a product name, the three polarity scores, the most recent ten reviews of the product, and a chart below that displays the average sentiment score of the product over the last six months, according to the manufacturer.

Finally, the last block contains a message for items that do not exist in my database; for example, if a user searches for a product for which I did not fetch data, the app will return a message saying "Sorry, I do not have data for this product; please try another product."

**Chapter 4 – Results and Discussion**

My apps were successfully developed after weeks of coding, going through the process of web scraping (cleaning), analysis (dealing with mistakes), and learning Django. One thing to keep in mind is that both programmes will only be available on the local server.

**4.1: Application one**

An excel file containing the names of products will be uploaded in this section by the user. Once the file has been uploaded, the programme will display a download button. Once you click on it, the app will download all the reviews and the dates they were written for each of the products listed in the uploaded excel file. The following procedures and application are demonstrated. The sheet to be uploaded has two product names as shown in table two below.

Table2 : Products to Review

Table

Description automatically generated

Step I.) Go to the application interface, click on “Choose File”

Graphical user interface, text, application

Description automatically generatedFig 2: Initial Interface

Step ii.) Select a file

A screenshot of a computer

Description automatically generated

Fig 3: System search File

Step iii.) File has been uploaded, click on “Submit”, a “Download” option appears

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedfig 4: Final Output Interface

Step iv.) Click download to find the following output file. It contains all reviews about iPhone 11 and Samsung A52 in a excel sheet.

Graphical user interface, text

Description automatically generated

Figure 5: List of review in an Excel Sheet

**4.2: Application two**

Initially, the interface looks like this:

Graphical user interface, application

Description automatically generated

I.) If the product mentioned by the user does not exist in our database:

Graphical user interface, text, application

Description automatically generated

Figure 6:Initial Website

ii.) When the product exists in the database, the following analysis is shown:

Graphical user interface

Description automatically generated with low confidence

Fig 7: Website Logo with Graph of Months vs Average Review

In this graphic, for example, the term "iPhone" is being searched for. Now, it has a low positive sentiment score of 0.242, which means that only 24.2% of consumers had pleasant things to say about the product. However, at the same time, negative sentiment is at an all-time low. Only 7.1% of users have problems with their iPhones and lodge complaints with Apple. In general, most buyers (68.7%) have an indifferent attitude about iPhones. This might be because there are multiple new competitors in the mobile sector that provide capabilities comparable

to, and often even better than, iPhones at a significantly lower cost, therefore attracting the attention of most customers. The fact that this is a recent issue can be seen in the figure below, which shows that until 4 months ago, the average sentiment rating for iPhones was a perfect 100%, indicating that iPhones were universally adored! However, something happened in the 5th month, perhaps the release of a new, better, and comparatively cheaper phone, or the release of a new phone by Apple's long-time competitor Samsung, which caused the average sentiment score for iPhones to fall from 100% to 99.2%. It has recovered, though, and currently has a 99.8% positive sentiment score, which is the most in the previous month.

**Chapter 5 – Conclusion**

The time constraint was the most significant drawback of this undertaking. If the necessary sources and time are available, this application has the potential to be real-time and dynamic. Making the programme real-time would need extracting data every day to ensure that the most recent evaluations are presented even if the application has been in use for several days or months. Making it dynamic would imply having more options, which would imply more data extraction daily. This would make the database enormous and necessitate the use of a database management system (DBMS), which is outside the scope of this project and not cost-effective at all; or it would necessitate the development of code that deletes the reviews extracted after they have been analysed and displayed to the user.

Furthermore, getting data from many different websites is difficult because the page layout and options for each website are different, and so a single for loop will not be sufficient to complete the task. Each website will require its own version of the code, which will need to be customised.

**References**

Bakos, Yannis. 2001. "The Emerging Landscape for Retail E-Commerce." *Journal of Economic Perspectives*, 15 (1): 69-80.

Chong, A. Y. L., Li, B., Ngai, E. W., Ch'ng, E., & Lee, F. (2016). Predicting online product sales via online reviews, sentiments, and promotion strategies: A big data architecture and neural network approach. *International Journal of Operations & Production Management*.

Han, S., & Anderson, C. K. (2021). Web scraping for hospitality research: Overview, opportunities, and implications. *Cornell Hospitality Quarterly*, *62*(1), 89-104.

Jeyapriya, A., & Selvi, C. K. (2015, February). Extracting aspects and mining opinions in product reviews using supervised learning algorithm. In *2015 2nd international conference on electronics and communication systems (ICECS)* (pp. 548-552). IEEE.

Jones, I. A., & Kim, K. Y. (2015). Systematic service product requirement analysis with online customer review data. *Journal of Integrated Design and Process Science*, *19*(2), 25-48.

Lee, M., Jeong, M., & Lee, J. (2017). Roles of negative emotions in customers’ perceived helpfulness of hotel reviews on a user-generated review website: A text mining approach. *International Journal of Contemporary Hospitality Management*.

Liu, Y., Soroka, A., Han, L., Jian, J., & Tang, M. (2020). Cloud-based big data analytics for customer insight-driven design innovation in SMEs. *International Journal of Information Management*, *51*, 102034.

Mesarosh, K., & Sun, F. S. (2017, August). A Framework for Analyzing Online Customer Ratings. In *2017 25th International Conference on Systems Engineering (ICSEng)* (pp. 401-404). IEEE.

Tucker, C., & Kim, H. (2011). Predicting emerging product design trend by mining publicly available customer review data. In *DS 68-6: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 6: Design Information and Knowledge, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011*.

Wang, W., & Zhou, Y. (2009, June). E-business websites evaluation based on opinion mining. In *2009 International Conference on Electronic Commerce and Business Intelligence* (pp. 87-90). IEEE.

Wang, Y., & Zhang, J. (2017, December). Keyword extraction from online product reviews based on bi-directional LSTM recurrent neural network. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 2241-2245). IEEE.

Wang, Y., Mo, D. Y., & Tseng, M. M. (2018). Mapping customer needs to design parameters in the front end of product design by applying deep learning. *CIRP Annals*, *67*(1), 145-148.

**APPENDICES**

Some sites that have been used for development of the project are the followings -

<https://www.researchgate.net/publication/326706634_Sentiment_Analysis_of_Review_Data_of_a_Product_Using_Python>

<https://www.hrpub.org/download/201307/aeb.2013.010101.pdf>

<https://www.researchgate.net/publication/344869545_Sentiment_Analysis_on_Amazon_reviews>

<https://www.sciencedirect.com/science/article/pii/S1877050917329253>

**Code**

Figure1Text

Description automatically generated

Figure 2

Text

Description automatically generated

Figure 3

Graphical user interface, text, application, email

Description automatically generated