SENTIMENT ANALYSIS FOR E-COMMERCE CLOTHING REVIEWS USING BIDIRECTIONAL RECURRENT NEURAL NETWORK

MUHAMMAD FARHAN FIRDAUS BIN HAIROL ZAMAN

Bachelor of Computer Science (Software Engineering) with Honors

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	MUHAMMAD FARHAN FIRDAUS BIN HAIROL ZAMAN
Date of Birth	:	
Title	:	Sentiment Analysis for E-commerce Clothing Reviews using Bidirectional Recurrent Neural Network
Academic Session	:	Semester II 2022/2023

I declare that this thesis is classified as:

☐ CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐ RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑ OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

_____	_____
(Student's Signature)	(Supervisor's Signature)
_____	Ts. Dr. Zuriani Mustaffa
New IC/Passport Number	Name of Supervisor
Date: 25/7/2023	Date: 26/07/2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name	Muhammad Farhan Firdaus Bin Hairol Zaman
Thesis Title	Sentiment Analysis for E-commerce Clothing Reviews using Bidirectional Recurrent Neural Network

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

_____
(Supervisor's Signature)

Date: 26/07/2023

Stamp: DR. ZURIANI BINTI MUSTAFFA
SENIOR LECTURER
FACULTY OF COMPUTING
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN PAHANG
TEL: 09-4244734

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in ……..

_____

(Supervisor's Signature)

Full Name : Ts. Dr. Zuriani Mustaffa

Position : Senior lecturer

Date : 26/07/2023

_____

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

_____

(Student's Signature)

Full Name : MUHAMMAD FARHAN FIRDAUS BIN HAIROL ZAMAN

ID Number : CB20153

Date : 22 JANUARY 2023

SENTIMENT ANALYSIS FOR E-COMMERCE CLOTHING REVIEWS USING BIDIRECTIONAL RECURRENT NEURAL NETWORK

MUHAMMAD FARHAN FIRDAUS BIN HAIROL ZAMAN

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy/Master of Science/Master of Engineering

Faculty of Computer

UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

ACKNOWLEDGEMENTS

In the accomplishment from this research, many people helped me during the hard work. This time, I want to thank all the people what directly and indirectly contributed for this research.

First thing first, I would like to praise Allah and thank Allah for giving me the chance and strength to complete this research. Then, to my supervisor Dr. Zuriani Binti Mustaffa who has been the most important person for me and non-stop guides and advises me to help me complete this research journey. Her recommendation and advice have served as a major contribution towards the completion of this project.

Next, I want to thank my both parents that always encouraging me to continue my degree in this Software Engineering field. Then, also having my backs when upside down completing this research. Next, the important person that supported morality throughout all the phases for this research journey is Nurul Awla Binti Mohamed as my best friend. Without her presence maybe this research cannot be completed.

Lastly, all of the people that indirectly helped me completing this research such as mentors and contributors in Internet that shares the beneficial information about the machine learning and other related to this research.

ABSTRAK

Kaedah pemasaran hari ini meletakkan keutamaan yang tinggi kepada memahami emosi pelanggan. Syarikat-syarikat akan mendapat wawasan tentang bagaimana pelanggan melihat barangan dan / atau perkhidmatan mereka, dan mereka akan mendapat idea tentang bagaimana untuk meningkatkan tawaran mereka. Kajian ini membuat usaha untuk memahami hubungan antara beberapa faktor dalam ulasan pelanggan di kedai dalam talian yang menjual pakaian wanita. Ia juga bertujuan untuk mengkategorikan setiap ulasan mengikut sama ada ia mengesyorkan produk yang dipertimbangkan dan sama ada beliau mengekspresikan sikap positif, negatif atau netral. Kami juga membangunkan rangkaian saraf berulang dua arah (RNN) dengan unit memori jangka pendek (LSTM) untuk klasifikasi perasaan. Hasil telah menunjukkan bahawa prediktor utama skor perasaan yang tinggi adalah cadangan, dan sebaliknya. Penilaian dalam ulasan produk, sebaliknya, adalah prediktor kabur skor perasaan. Selain itu, kami mendapati bahawa LSTM dua arah mencapai skor F1 0.93 untuk penilaian perasaan.

ABSTRACT

Today's marketing methods place a high priority on comprehending client emotions. Companies will gain insight into how customers view their goods and/or services, and they will get ideas on how to enhance their offerings. Traditional methods of sales and business are not as effective as the e-commerce approach. It is such a hassle for customers to walk into the retail store and purchase their product needs. It is a waste of time and energy for today which world is full of technology. This study makes an effort to comprehend the relationship between several factors in customer reviews on an online store selling women's clothes. It also aims to categorize each review according to whether it recommends the product under consideration and whether it expresses a positive, negative, or neutral attitude. Thus, this study proposed a bidirectional recurrent neural network (RNN) with a long-short-term memory unit (LSTM) for sentiment classification. Results have indicated that a major predictor of a high sentiment score is a recommendation, and vice versa. Ratings in product reviews, on the other hand, are hazy predictors of sentiment scores. Additionally, we discovered that the bidirectional LSTM achieved an F1-score of 0.93 for sentiment classification.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

BRNN	Bidirectional Recurrent Neural Network
CRM	Customer Relationship Management
GB	Gigabyte
GPU	Graphic Processing Unit
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
RAM	Random Access Memory
RFC	Random Forest Classifier
RNN	Recurrent Neural Network
SVM	Support Vector Machine
UMP	Universiti Malaysia Pahang

CHAPTER 1

INTRODUCTION

1.1 Research Background

Nowadays, e-commerce is no longer a rare word for people. Starting a business will be much more convenient if someone already knows to deploy an e-commerce website. Some urban people might think that shopping at the supermarket and shopping mall is a hassle thing to be done because there is a lot of necessary business that needs prior attention. As the statistic predicts, there will be around 2.14 billion online shoppers around the world that shows over a quarter of the global population are using the internet to access at least one e-commerce site and decided to make at least a purchase through that site. (*Online Shopping Statistics, Facts & Trends in 2022*, n.d.)

Despite the advantages brought by e-commerce such as cost reduction by the sellers and customers, it also comes with consequences. "*Customer is the king*" and "*Customer always right*" are popular words for anyone in any kind of business. Failing to follow the customer demands and needs will surely affect the profits of the business. Similar to the e-commerce site, the feedbacks and reviews from the customers are essential to improve the business itself, the website improvement, or even the security of the e-commerce. Customers will start to feel irritated with the e-commerce sites when their reviews and feedback do not count by the sellers.

Because of that, retailers and sellers need to be aware of the feedback or reviews are given by their customers. Good development of technology can help the seller to keep improving their e-commerce site and products. From the research in 2022, Customer Relationship Management (CRM) Impact Report shows an average customer churn rate of 32% globally. The same report also stated that 83% of marketing managers think these rates are because of the inability to produce solid communication with customers and

irrelevant messaging. (*9 Major Advantages of Ecommerce to Businesses in 2022 | Seller Blog*, n.d.)

Therefore, this study proposed the sentiment analysis prediction using Bidirectional Recurrent Neural Network (BRNN). The prediction model will be very helpful for retailers to analyze and identify customers' demands for further update improvement regarding the products as well as the system. The proposed model will help companies communicate well with customers and produce more relevant messages. By analyzing their emotions, retailers can get a better idea of their experience and provide the best service that produces the outcomes of decreasing customer churn. (*Top 5 Benefits of Sentiment Analysis for Businesses*, n.d.)

1.2 Problem Statement

In this modern era, huge companies are willing to make some investments to ensure their companies have consistent profits. Applied technology and computer science such as e-commerce sites, robot assistants, and augmented and virtual reality bring an enormous contribution to the business. People nowadays prefer online shopping compared to physical shopping as it reduces cost, saves time, and is convenient. Because of that, it becomes the responsibility of the companies to ensure their customers will always choose their e-commerce site to keep growing and gain profits. Reputation and reviews from the customers also will become a factor to attract new buyers and customers to the site.

In research conducted by Dixa in the year 2022, 93% of customers will read the online reviews before purchasing with 47% spreading the word of positive reviews while 95% from the rooftop are complaining about their negative experience. Besides, according to their data, 97% of buyers said customer reviews affect their decision to purchase any products, while 92% of consumers hesitate to purchase if there are no customer reviews displayed for the selected product. As the research stated, huge companies must concern more about reviews and comments from their customers.

Despite that, only the star rating display for certain e-commerce is insufficient. It is because the data collected from the rating itself is inaccurate and misleading. This could

happen when the customers want to give a 2-star rating only because of their reasons but, accidentally click the 5-star rating and then are unable to undo it. Other than that, judging by the rating only will cause misleading as sometimes the rating and review are not synchronized. For example, the customer gives a 2-star rating but all statements from the comments are positive, for example, it stated that the quality product is good, all the service from the shop is perfect and delivery is fast. That's why the contribution of technology and computer science which is machine learning is helpful.

Besides, some people might have a problem when giving a rating, recommendation, or review. For example, they wrote a bad review about a product they purchased, but they recommend new or other customers to purchase the products. Different scenarios also happened like people not recommending others to purchase the selected products but they give the highest rating and excellent quality review for the products. These problems will give mixed feelings for the new user to purchase or ignore and find another product.

1.3 Objective

The objectives of this research are:

i. To review the current issue regarding the customer review to the sentiment analysis.

ii. To design and develop a model of sentiment analysis using BRNN for Women's E-commerce sites.

iii. To evaluate the output of sentiment analysis from the customer reviews on the Women's E-commerce site.

1.4 Scope

The scope of this research:

i) The proposed model will be tested on the Women E-commerce site.

ii) The model can detect customer comments and reviews from Women's E-commerce only.

iii) English is the only language used when performing sentiment analysis.

iv) The model detects specific sentiment words and classifies to negative or positive output.

1.5 Thesis Organization

The thesis consists of 3 chapters for part 1. Chapter 1 has the starting and introduction for the research. The title proposed a technique and model that is suitable before starting to plan and execute the result. Then, followed by a problem statement for the research to solve the problem that occurred. The scope of this field of research also will be stated and explained in detail. Lastly, the thesis organization is the planning for the necessary context that needs to be included.

For chapter 2 needs to include a literature review from the related research or works done by other researchers with a detailed explanation. Then, need to follow the analysis and comparison of the related and past works. All the techniques, methods, models, advantages, and disadvantages need to be written. Last but not least chapter 2 is a summary of the related works and connects it with the proposed work.

Chapter 3 is about the proposed work. Firstly, the brief introduction to the contents of the chapter. Then, the methodology for the research such as processes, resources, and tools needs to be stated. Next, the project requirements include expected input, process, output, and limitations. Inside the research or the input for the research need to be ready with the datasets to be tested. After completing the details of the dataset, proof of the initial concept of the work needs to be done to provide evidence that the research will be functional. The next step is the validation and testing for the partially tested data and making the early comparison. Last but not least the potential use of the proposed solution and the Gantt chart for the progress of the research done.

Then, chapter 4 is about the result and discussion from the research. The discussions are about the partition for training dataset, validation and testing dataset. Then, the hyper parameters used for this BRNN-LSTM process such as cell size, droput rate and epochs. For the most important part is the result from the sentiment analysis and classification between three types of sentiment. All the results such as precision, accuracy, recall and f1 score are recorded.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter briefly stated previous research papers and models that used machine learning to identify sentiment analysis. Many techniques and algorithms are explored during the research ongoing so it will be helpful to give ideas for this research. There are many techniques implemented to execute the result from the chosen datasets. Basically, most of the research observing the reliability from the techniques to produce the good results from the selected datasets.

2.2 Related Works

2.2.1 Sentiment Analysis of Twitter Data Using Machine Learning Approaches Naïve Bayes

Research for this related work is about identifying the Twitter data for Sentiment Analysis using Machine Learning approaches (Elbagir & Yang, 2018). For the introduction, the research briefly stated sentiment analysis and machine learning. Sentiment analysis is a process in which the dataset consists of emotions, attitudes, or assessments that consider how a human thinks. For machine learning, there are two types of machine learning usually implemented for sentiment analysis which are supervised and unsupervised learning. For this research, supervised learning was implemented during the experiment.

For the preprocessing data, the datasets from Twitter were labeled using the unigram feature extraction technique. The framework used allows them to apply the raw sentences which look more appropriate to understand. It also deals with removing punctuation and repeated words to produce data that is more efficient. The dataset utilized

in this work, Twitter, is already labeled. A labeled dataset has a negative and positive polarity, making data analysis simple. For the feature extraction, using the unigram model extracts the adjective to make it show the data to positive and negative from a sentence.

For the decision of the techniques, they are using three supervised learning methods followed by semantic analysis. The techniques are Naïve Bayes, Support Vector Machine (SVM) and maximum entropy. The coding language they used is Python along with Natural Language Kit to train and classify all the techniques proposed. The data set size used is 19340 out of which 18340 remaining for 1000 data were used for testing. The data are computed in a formula and shown in a table with precision and accuracy. Table 1 to table 3 below shows the classification measurement from Naïve Bayes, maximum entropy and SVM techniques. Then, table 4 shows the comparison between all machine learning approaches accuracy comparison.

Table 2.1: Naive Bayes Classification Measurements

Performance Measures (%)	
Positive Recall	91.2
Negative Recall	85.4
Positive Precision	49.3
Negative Precision	39.3

Table 2.2: Maximum Entropy Classification Measurements

Performance Measures (%)	
Positive Recall	86.1
Negative Recall	80.0
Positive Precision	40.4
Negative Precision	33.6

Table 2.3: Support Vector Machine Classification Measurements

Performance Measures (%)	
Positive Recall	88.3
Negative Recall	83.5
Positive Precision	43.8
Negative Precision	35.7

Table 2.4: Accuracy comparison

Methods	Accuracy
Naive Bayes	88.2
Maximum Entropy	83.8
Support Vector machine	85.5
Semantic Analysis (WordNet)	89.9

From the experiment, the research concludes that from the machine learning approach, Naïve Bayes is the best technique to perform sentiment analysis based on text classifications. But the further accuracy obtained by the semantic analysis by using WordNet which is 89.9% leads to Naïve Bayes which is 88.2%.

2.2.2 Sentiment Analysis on Product Reviews Using Machine Learning Techniques Support Vector Machine (SVM) and Naïve Bayes

This research is about product reviews to identify sentiment analysis using machine learning techniques (Azhaguramyaa et al., 2022). The introduction of this research stated about sentiment analysis which is text analysis, natural language processing, and computational linguistics are all used to objectively detect, extract, and analyze subjective information from textual data. Then, the general meaning of the sentiment analysis itself is to determine a speaker's, writer's, or other subject's arrogance concerning a certain issue or contextual polarity to a given event, conversation, forum, interaction, or any documentation.

For the data sources, it stated that it requires numerous data on social media such as Blogs, and datasets such as movie, product and hotel reviews. Next, the review sites such as e-commerce websites such as Amazon, IMDB and CNET. Lastly, is by micro-blogging is known as a popular service for sending messages shortly such as Twitter, Tumblr, etc. The research also stated the related work that can be compared to their proposed work. From the previous research, they compared numerous methodologies for Sentiment Analysis and explored Machine Learning algorithms such as Support Vector Machine (SVM), Naïve Bayes (NB), and ME They also spoke about N-grams for

Sentiment Analysis. They also concluded from the previous experiment which SVM technique achieved 78% accuracy. A hybrid approach was also used when performing the experiments by combining a Lexicon-based approach and Machine Learning.

The datasets for this research are collected from Amazon and in JSON format. Each JSON file provides a count of the number of reviews. The dataset includes assessments of cameras, laptops, mobile phones, tablets, televisions, and video surveillance systems. For the preprocessing, the tokenization, stop word removal, stemming, punctuation mark removal, and so on have all been completed. It has been transformed into a bag of words. Then, the data for every sentence need to be analyzed and calculate the sentiment score.

The experiment result is presented in Figure 1 below in form of a Bar Chart. This shows the camera has the highest number of reviews which is 3000 and the lowest is televisions, below 2000 reviews.

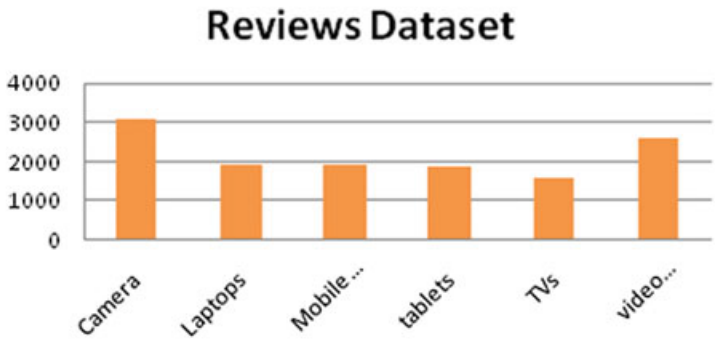


Figure 2.1: Dataset number of reviews

The Machine Learning techniques that are implemented to identify the sentiment analysis are Naïve Bayes and Support Vector Machine (SVM). The accuracy, precision and F-score are drawn by the Bar Chart in Figure 2.1. For a clearer data view, Table 5 shows all the details numbers for the technique used.

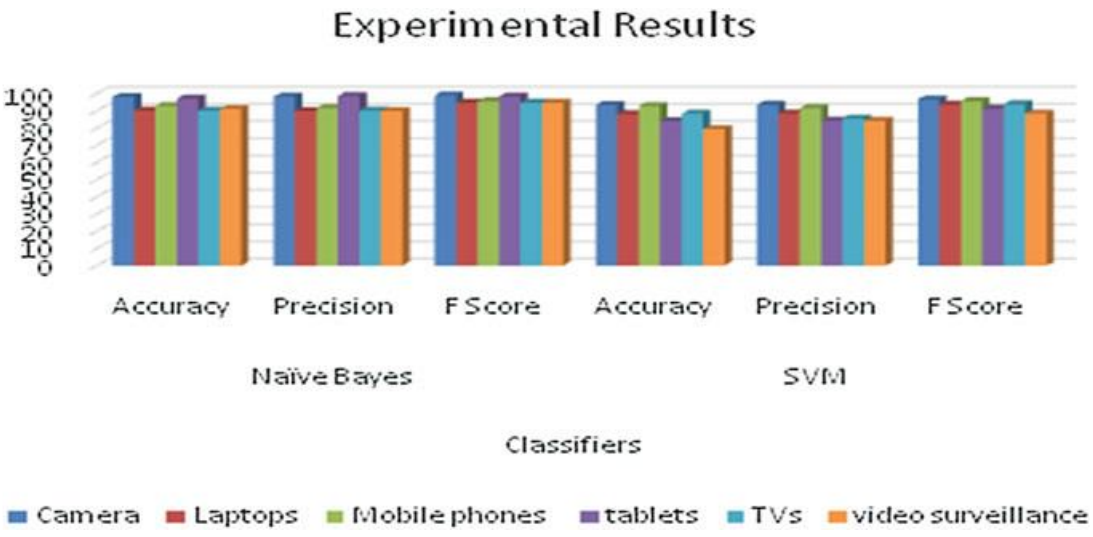


Figure 2.2: Experimental results for all the products

Table 2.5 Evaluation parameters for classifier of datasets

Dataset	Classifiers					
	Naïve Bayes			SVM		
	Accuracy	Precision	F score	Accuracy	Precision	F score
Camera	98.17	98.30	99.03	93.54	93.58	96.66
Laptops	90.22	90.01	94.74	88.16	88.52	93.71
Mobile phones	92.85	91.64	95.64	92.85	91.64	95.64
Tablets	97.17	98.73	98.31	84.12	84.31	91.37
TVs	90.16	90.17	94.72	88.49	85.56	93.89
Video surveillance	91.13	89.95	94.71	79.43	84.25	88.53

From the experiment for the sentiment analysis, the NB and SVM techniques have been used for better results. And they conclude that the Naïve Bayes classifier gains 98.17% accuracy for the Camera reviews while the Support Vector Machine which is SVM gains 93.14% accuracy for the same product which is the Camera. As the conclusion for better accuracy, Naïve Bayes need to be implemented to identify the sentiment analysis from all of the reviews, in term of accuracy, Naïve Bayes is leading by the percentage.

2.2.3 Sentiment Analysis of Movie Review Using Machine Learning Technique Random Forest Classifier

This research is about Sentiment Analysis using machine learning approaches. The dataset is a movie review based on the Lexicon model (Mitra, 2020). In the introduction of this research, they stated that at the Document level, the complete document is offered as a fundamental information unit to provide a scope of classification addicted to positive or negative class. Each sentence is classified initially into the subjective or the objective in Sentence level categorization, it is then classified as positive, negative, or neutral. The lexicon-based which is the second approach in the industry today contains a dictionary of positive and negative words that can be used to determine the sentiment polarity from the source of the dataset.

There are two methods mentioned in this research. Firstly, the training method. The training method is based on the 80:20 principle, the model trains itself to adapt to a given input to the associated output. 80% is applied here for the data to be fed into the application to train it. The input represents the text while the output represents the tags. For the prediction phase, the feature extractor's job is to convert unseen text inputs into feature vectors. Here, 20% from the principle 80:20 is applied. For the preparation before starting the experiments, they utilized 80% training data as input and then predicted using that obtained expertise in the previous stage, which is typically 20% training data as input.

The results and discussion are displayed in Table 6. They decided to separate between two approaches, which the first approach includes the Naïve Bayes algorithm, SKlearnBernouliieNB, and Sklearn SVC. While the second approach is, Decision Tree technique, Random Forest, and KNN.

Table 2.6: Accuracy for all the techniques

	Approach 1			Approach 2		
Algorithm	Naïve Bayes	SKlearnBernouliieNB	Sklearn SVC	Decision Tree	Random Forest	KNN
Accuracy (%)	70.44	70.15	75.37	52	80	71

The results of the sentiment analysis are formed by the bar chart in Figure 2.3 below. The bar chart shows that the Neutral outcomes from the experiment have the highest number which is 12000 reviews while the lowest data shows the Negative reviews with below than 2000 reviews in total.

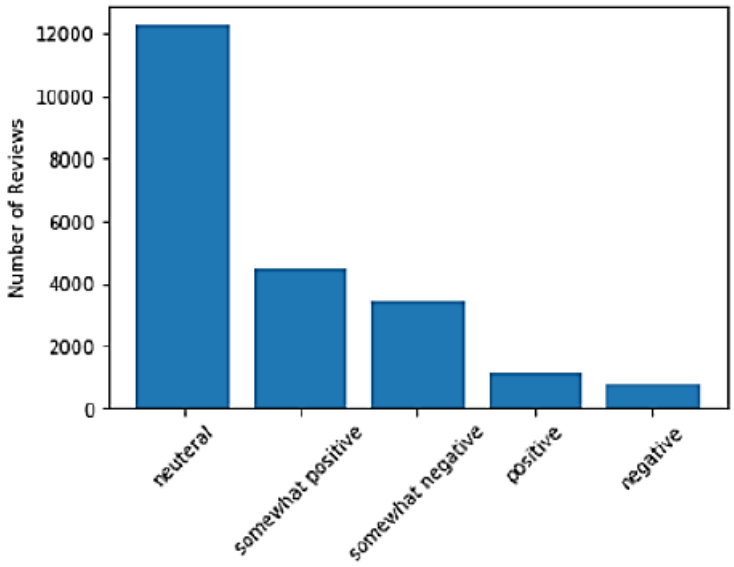


Figure 2.3: Bar Chart sentiment expressed in Reviews

Lastly, the highest accuracy from all of the techniques is proven from the calculation as shown in Figure 2.4 below. It also displays the precision, F1-score and support for the technique Random Forest.

```
pipeRFC.fit(x_train,y_train)
pred = pipeRFC.predict(x_test)
print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.88	0.97	0.92	19547
1	0.77	0.77	0.77	20328
2	0.67	0.61	0.64	19854
3	0.75	0.66	0.70	16642
4	0.89	0.97	0.92	18303
accuracy			0.80	94674
macro avg	0.79	0.80	0.79	94674
weighted avg	0.79	0.80	0.79	94674

Figure 2.4: RFC technique calculation for accuracy

2.3 Analysis of Related Works

From all three related works, objectives, techniques, frameworks, and tools are exposed. Thus, the analysis from the previous research needs to be done to be implemented in future works and brainstorming the improvement skills for the existing related works.

2.3.1 Comparison from Related Works

Table 2.7: Comparison between Related Works

Technique	**Naïve Bayes**	**SVM and Naïve Bayes**	**Random Forest Classsifier**	**Bidirectional Recurrent Neural Network BRNN**
Analysis	To do the sentiment analysis using techniques inside supervised learning and semantic analysis.	To study the text analysis, natural language processing and identify subjective information from textual data.	To produce the sentiment analysis using the training method and then the prediction method.	To analyze customer reviews on Women E-commerce by employing statistical analysis and sentiment classification
Data	Twitter datasets	Amazon product review datasets consist of camera, laptop, mobile phones and TV.	Movie review	Women E-commerce clothing site
Tools/ programming language	1. Python language 2. Natural Language Tool kit 3. WordNet	1. JSON file	Not stated	Python language
Advantage	- All the formulas are stated clearly before starting the calculation - Clear comparison between used techniques related to	- Prepare the proposed methodology before starting the experiments.	- Exploring machine learning techniques along with Lexicon based.	- Duplicates the RNN processing chain that input processed forward and backward, allows BRNN to look for the future context as well.

	sentiment analysis.			
Disadvantage	- Complex and complicated training and classification.	- Not stated the specific formula to calculate the accuracy. - Positive and Negative outcomes do not show in the result.	- Work consider completed after using Logical Regression and Random Forest - Not state the details to calculate the accuracy of the techniques.	- The entire sequence must be available before making the predictions
Highest accuracy	For machine learning, Naïve Bayes 88.2%	Naïve Bayes, 98% for Camera reviews	Random forest, 80%	Expected on recommendation classification, 85% and sentiment classification, 90%

2.3.2 Comparison for advantages

All methods and algorithms proposed must have benefits for the users and the experiment to be done or the content made by the researchers. In the first related work which is using the technique Naïve Bayes for the Twitter datasets, it is available all the formulas are stated clearly before starting the calculation. Then, it also has a clear comparison between the used techniques related to sentiment analysis. Move the second technique benefits using two techniques SVM and Naïve Bayes to do machine learning for Amazon product review datasets. Based on the observation, the thesis benefits are only available with one benefit which is preparing the proposed methodology before starting the experiments. Other than that, the third related work is using the technique of Random Forest Classifier (RFC) to study the data from a movie review. The available advantage of this technique is exploring machine learning techniques along with Lexicon based. Lastly, for the proposed technique which is BRNN using LSTM benefit is it can duplicate the Recurrent Neural Network (RNN) processing chain in that input is processed forward and backward, allowing BRNN to look for the future context as well.

2.3.3 Comparison for disadvantages

All methods and techniques that come with benefits also have side effects or consequences to the experiments or the documents done by the researchers. The first related work which is Twitter datasets using the Naïve Bayes technique has the side effect from their process itself which is complex and complicated training and classification. Next, for the Amazon products review using two techniques, SVM and Naïve Bayes in terms of the process of technique positive and negative outcomes do not show in the result. For the document, bad observation is not stated in the specific formula to calculate the accuracy. Then, for the movie review by RFC consequences are the work from the process of technique work consider completed after using Logical Regression and Random Forest only. Then, the aspect of the thesis it not stated the details to calculate the accuracy of the techniques. Lastly, the disadvantage of the proposed technique BRNN with LSTM is the entire sequence must be available before making the predictions in terms of the process from this technique.

2.4 Summary

In conclusion, related works techniques, tools and pre-processing for the machine learning process indirectly blast out the ideas for the future project or research. Machine learning implementation techniques such as Naïve Bayes, KNN, SVM and Maximum Entropy accuracy measurements will be reconsidered before using the technique for the sentiment analysis. Formula, model, and graph while experimenting will make the work more interactive and professional.

Improvement elements will be considered before deciding on the newly proposed solution for the research work. The elements such as the accuracy and the quality of the experimental results, the time taken for the dataset pre-processing and the selection of the cleaned datasets need to be considered. The elements stated will effecting the process during the execution of the results and validation process.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will focus on the methodology in detail and the processes that will be explained thoroughly for the research. The proposed model, framework, design, and data selection will be included. Phases terminologies such as Gantt Chart and other appendixes will be used for this research.

3.1.1 Recurrent Neural Network (RNN)

The definition of Recurrent Neural Network is a type of Neural Network machine learning technique where the output is transformed to the input as before the current state. For the traditional neural network, input and output are independent of each other but, in cases to predict the next word of the sentence, the previous word is necessary to be remembered (Introduction to Recurrent Neural Network - GeeksforGeeks, n.d.). RNN has the memory to remember all information about the data that has been calculated.

It employs the same settings for each input since it performs the same work on all inputs or hidden layers to generate the result. Unlike other neural networks, this minimizes parameter complexity. The working process for the RNN can be used from this example, there is a deeper network that pairs with one input layer, one output layer and three hidden layers. Each one of the hidden layers has its own set of weights and biases. Then, each of these layers is independent of the others. Then, RNN will do the job firstly to convert the independent activation to the dependent one by giving similar biases and weights for all layers but reducing the complexity by increasing their parameters and remembering their previous output by giving the input to the next hidden

layer. Then, all three layers can be joined together because their bias and weight are the same in a single recurrent layer.

3.1.2 Bidirectional Recurrent Neural Network

Bidirectional Recurrent Neural Network is an extended version of RNN that is used in a combination of two RNNs. One RNN technique moves forward while the other one is moving backward. BRNN differs from RNN because it has an additional layer to cope with the backward training process. The given time is represented by *t* as the formula below:

3.1.3 Research Framework

The research framework from this research-based consist of four main phases which started with a literature study from previous and related works, planning or design, implementation, and testing. The research framework will act as a foundation for good research as a good building needs a strong foundation. (Conceptual and Theoretical Frameworks for Thesssis Studies: What You Must Know, n.d.)

The first step for the research framework is the **literature review**. This literature review's purpose is for gaining as many ideas as possible from previous research and works related to sentiment analysis with machine learning. The second phase is the **planning and design** phase, which is deciding the correct model, planning the input, and process, and predicting the output before the implementation process. Then, during the **implementation** process, the datasets will be run and evaluated. All the expected output with the exact output will be recorded. Lastly is the **testing and validation** process to extract all the constraints and limitations from this research for enhancement for future research.

3.1.4 Project tools

Table 3.1: Project tools for research

Software/Website	Functionality
Keras with Google TensorFlow	Train the data to follow the bidirectional recurrent neural network with long-short term memory (LSTM)
Jupyter Notebok	Compile and run all the codes related from this research project
NumPy and pandas	Handle libraries in python for preprocessing data
matplotlib	Data visualization, graphs, and results
Microsoft Excel	Read and used to view the dataset

3.2 Research Requirement

3.2.1 Dataset Pre-processing

i. Text Cleaning

The definition for text cleaning is the process of preparing raw text for Natural Language Processing so that machines can easily understand human language (Text Cleaning for NLP: A Tutorial, n.d.). The review texts from the dataset were cleaned by removing the delimiters in words such as /n and /r.

ii. Sentiment Analysis

The goal of the research is to do a sentiment analysis. But is a really difficult process to manually tag the review texts. The effective solution is by using the sentiment analyzer tools in the python library which is 'NLTK'. It will automate the process for the sentiment analysis which will consider the rating that is equal and more than 3 as positive feedback for the review. Otherwise, it is considered negative feedback.

iii. Word Embeddings

The definition for word embedding is a language modeling technique that has the purpose to map words to vectors from real numbers. The word embedding process will

be run in the 'NLTK' python library during the sentiment analysis process. The word embedding process will be using the GloVe file.

3.2.2 Input

i. Recommendation classification

The recommended classification consists of 1 which means recommendable and 0 which is not recommendable to buy in the dataset that represents the customer recommendation to another new customer to buy that product or not.

ii. Review text sentiment classification

Review text is the review from the customers that already purchased the products and give any feedback regarding the quality, design, features and materials of the products.

3.2.3 Expected output

The expected output for this research is the test accuracy and precision between two types of inputs which is **recommendation** and **review text**.

Table 3.2: Statistical Analysis for Recommendation classification using Bidirectional LSTM

Class	**Accuracy**	**Precision**
(0) Not recommended	≈ 0.70	≈ 0.70
(1) Recommended	≈ 0.90	≈ 0.90
Average	≈ 0.85	≈ 0.85

Data from the column Recommendation IND which is Not recommended has a value of 0 and Recommended value of 1. The expected accuracy and precision predicted amount is as in the table above.

Table 3.3: Statistical analysis for Review text sentiment classification using Bidirectional LSTM

Class	**Accuracy**	**Precision**
(0) Negative	≈ 0.75	≈ 0.70
(1) Neutral	≈ 0.40	≈ 0.50
(2) Positive	≈ 0.90	≈ 0.80
Average	≈ 0.90	≈ 0.90

Data from the Review Text column will be classified into three classes which is Negative, Neutral and Positive. Expected output from the data sentiment as the table above.

3.2.4 Constraints and Limitations

1. Technology constraints

Training, validation and testing dataset processes were run from the average specification of hardware such as 8 GB RAM and using only Intel IRIS for GPU. Running code processing may take a very long time to be completed. Then, it also may cause the performance from the code devices.

2. Knowledge constraints

BRNN and LSTM are rarely used for algorithms for machine learning. It is difficult to find previous research that used these types of algorithms for reference purposes. Raw information and data need to be found from the Internet and the information is not specific. It also includes the knowledge and experience encountering library and Jupyter Notebook as the tool to run the Python code.

3.3 Propose Design

3.3.1 Model

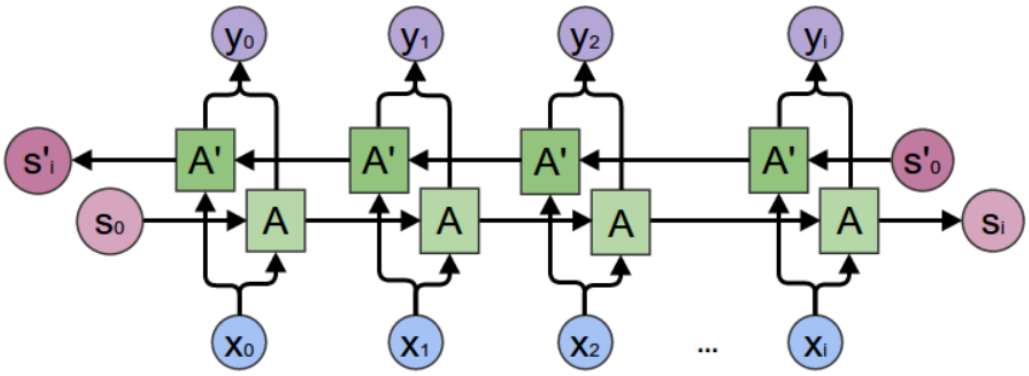


Figure 3.1: Bidirectional RNN map

Figure 3.1 shows the Bidirectional Recurrent Neural Network mapping that will be used to execute and run the Women's E-commerce dataset. The map input sequences *x* to target the sequences *y*, with loss of L(*t*), for each time, *t*. RNN *s* propagates forward in time to the right while RNN cells *s'*, propagate backward in time which is to the left. Before applying the activation function to get *y*, for each time *t*, *the* output unit represents by *o(t).* It can benefit from a relevant summary of the past which indicates *s(t)* input and from a relevant summary of the future which is *s'(t)* input.

However, the proposed solution recurrent neural network has its consequences which are their vanilla RNN can suffer from vanishing gradients even though the most appropriate algorithm to do the sentiment classification task is using RNN. Hence, the revamped technique needs to use which is adding the long-short-term memory (LSTM) units designed to solve the problem stated. Bidirectional with LSTM can be seen in Figure 4 that the model has the capability to learn from the past and future of the text sequence.

Below is the model or the formula used as the LSTM gate equation that is implemented inside Google TensorFlow.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

f represents the forget gate that will forget targeted non-essential information for the model. Input gate which is *i* will accept new data input at the given time step s_t. $\tilde{C}$ bring the meaning candidate cell state value of each LSTM cell. C is the value that needs to be passed to the next RNN LSTM cell. *o* same as the above table which brings the meaning of the output gate will have the task to decide what the cell state will output. Lastly, *h* is the cell state output from the decided output and a cell state value.

This machine learning model will be applied to two text classification problem on the datasets which is recommendation classification and sentiment classification. Recommendation classification's purpose is to determine whether the review text has a relationship with the text recommendations from the reviewed product while sentiment classification is to determine the perception of customers toward the purchased products.

3.3.2 Flowchart

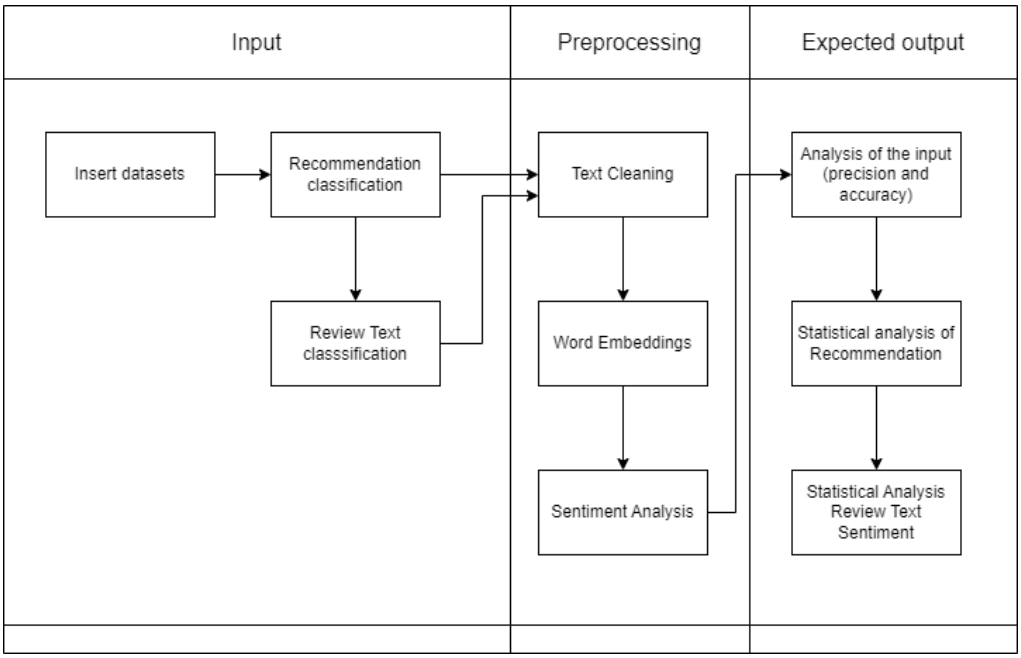
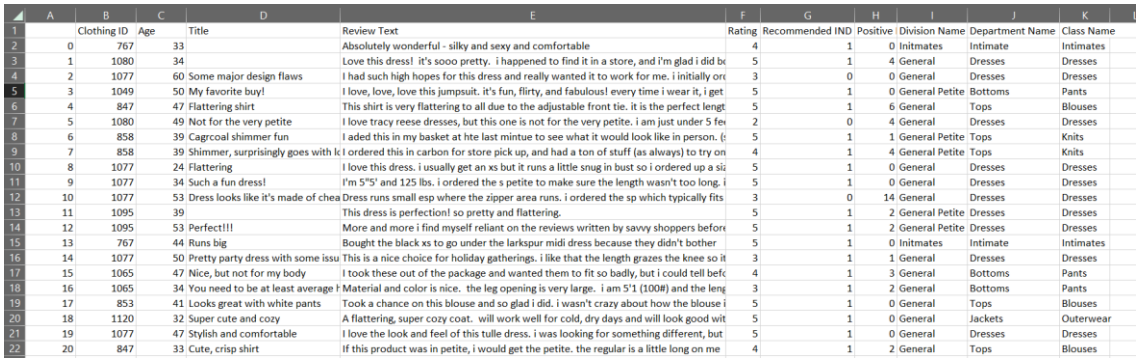


Figure 3.2: Flowchart for BRNN technique sentiment analysis

From the figure 3.2 flowchart above is explained briefly about all the crucial steps to perform the execution. Starting from giving the input and classifying from 2 different columns inside the dataset then, the preprocessing steps that need to follow to get more accurate data from datasets. Lastly, for the expected output the analysis from the inputs.

3.4 Datasets

The dataset used for this Sentiment Analysis research is women's Clothing E-commerce reviews which were downloaded from the website Kaggle.com. The dataset contains the reviews written by their real customers and the references for the company replaced with the word “retailer”. Figure 7 below shows the first 20 rows from the total of 23,485 rows. The table below shows the Frequency Distribution from the unique count of the dataset features.



	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive	Division Name	Department Name	Class Name
0	767	33		Absolutely wonderful - silky and sexy and comfortable	4	1	0	Initmates	Intimate	Intimates
1	1080	34		Love this dress! it's sooo pretty. i happened to find it in a store, and i'm glad i did bc	5	1	4	General	Dresses	Dresses
2	1077	60	Some major design flaws	I had such high hopes for this dress and really wanted it to work for me. i initially orc	3	0	0	General	Dresses	Dresses
3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, flirty, and fabulous! every time i wear it, i get	5	1	0	General Petite	Bottoms	Pants
4	847	47	Flattering shirt	This shirt is very flattering to all due to the adjustable front tie. it is the perfect lengt	5	1	6	General	Tops	Blouses
5	1080	49	Not for the very petite	I love tracy reese dresses, but this one is not for the very petite. i am just under 5 fe	2	0	4	General	Dresses	Dresses
6	858	39	Cagrcoal shimmer fun	I aded this in my basket at hte last mintue to see what it would look like in person. (s	5	1	1	General Petite	Tops	Knits
7	858	39	Shimmer, surprisingly goes with k	I ordered this in carbon for store pick up, and had a ton of stuff (as always) to try on	4	1	4	General Petite	Tops	Knits
8	1077	24	Flattering	I love this dress. i usually get an xs but it runs a little snug in bust so i ordered up a siz	5	1	0	General	Dresses	Dresses
9	1077	34	Such a fun dress!	I'm 5"5' and 125 lbs. i ordered the s petite to make sure the length wasn't too long. i	5	1	0	General	Dresses	Dresses
10	1077	53	Dress looks like it's made of chea	Dress runs small esp where the zipper area runs. i ordered the sp which typically fits	3	0	14	General	Dresses	Dresses
11	1095	39		This dress is perfection! so pretty and flattering.	5	1	2	General Petite	Dresses	Dresses
12	1095	53	Perfect!!!	More and more i find myself reliant on the reviews written by savvy shoppers before	5	1	2	General Petite	Dresses	Dresses
13	767	44	Runs big	Bought the black xs to go under the larkspur midi dress because they didn't bother	5	1	0	Initmates	Intimate	Intimates
14	1077	50	Pretty party dress with some issu	This is a nice choice for holiday gatherings. i like that the length grazes the knee so it	3	1	1	General	Dresses	Dresses
15	1065	47	Nice, but not for my body	I took these out of the package and wanted them to fit so badly, but i could tell befc	4	1	3	General	Bottoms	Pants
16	1065	34	You need to be at least average h	Material and color is nice. the leg opening is very large. i am 5'1 (100#) and the leng	3	1	2	General	Bottoms	Pants
17	853	41	Looks great with white pants	Took a chance on this blouse and so glad i did. i wasn't crazy about how the blouse i	5	1	0	General	Tops	Blouses
18	1120	32	Super cute and cozy	A flattering, super cozy coat. will work well for cold, dry days and will look good wit	5	1	0	General	Jackets	Outerwear
19	1077	47	Stylish and comfortable	I love the look and feel of this tulle dress. i was looking for something different, but	5	1	0	General	Dresses	Dresses
20	847	33	Cute, crisp shirt	If this product was in petite, i would get the petite. the regular is a little long on me	4	1	2	General	Tops	Blouses

Figure 3.3: First 20 rows of the Women E-commerce datasets

From the figure 3.3, the first labeled column is **Clothing ID** which represents the unique code for the specific name of the products. Then, the **Age** column is from the buyer/customer age from the e-commerce store. Followed by a **Title** for the next column and **Review Text** for the customer to show their expression from the products they have chosen. Next is the **Rating** column where customers can give a rating from 1 to 5. **Recommended by IND** with **Positive Feedback Count** for the other columns. For the last three columns are **Division Name**, **Department Name** and **Class Name**.

Table 3.4: Frequency Distribution Dataset Features

Feature	Unique Count
Clothing ID	1172
Age	77
Title	13984
Review Text	22621
Age	77
Title	13984
Review Text	22621
Rating	5
Recommended IND	2
Positive Feedback Count	82
Division Name	3
Department Name	6
Class Name	20

The data from the dataset that will be used for the sentiment analysis is from column Recommendation IND to observe whether it has a relationship with the Sentiment Analysis data that will be executed from the Review Text column.

3.5 Proof of initial concept

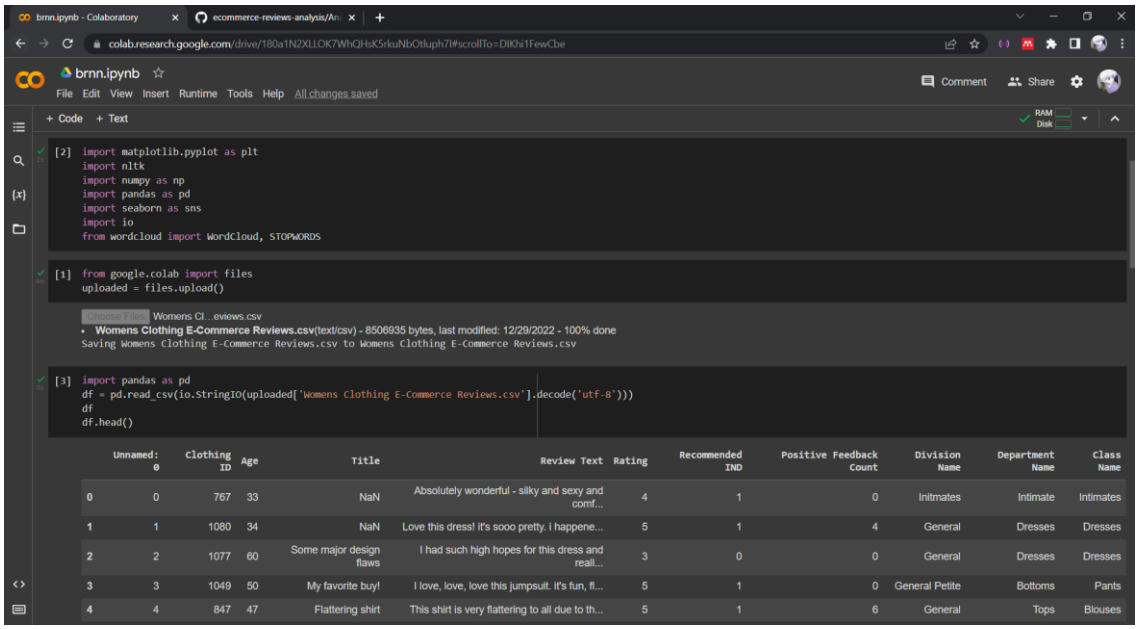


Figure 3.4: Starting of the code

Figure 3.4 shows the coding to do the starting of the initial work. The code starting such as import all the library that will be using during the output for the experiments. Last of the code also have the command to display head of the datasets which is starting 5 data from the datasets.

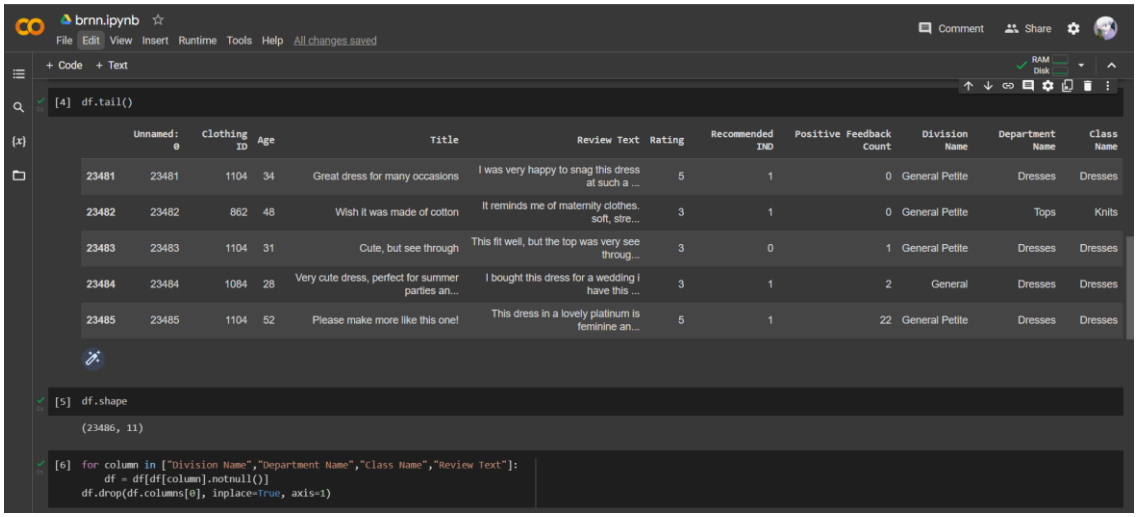
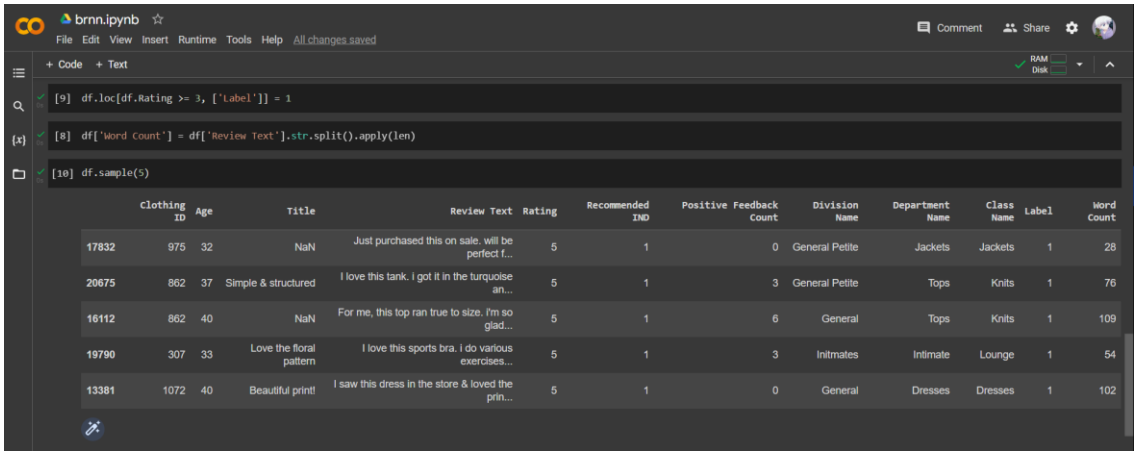


Figure 3.5: Tail code

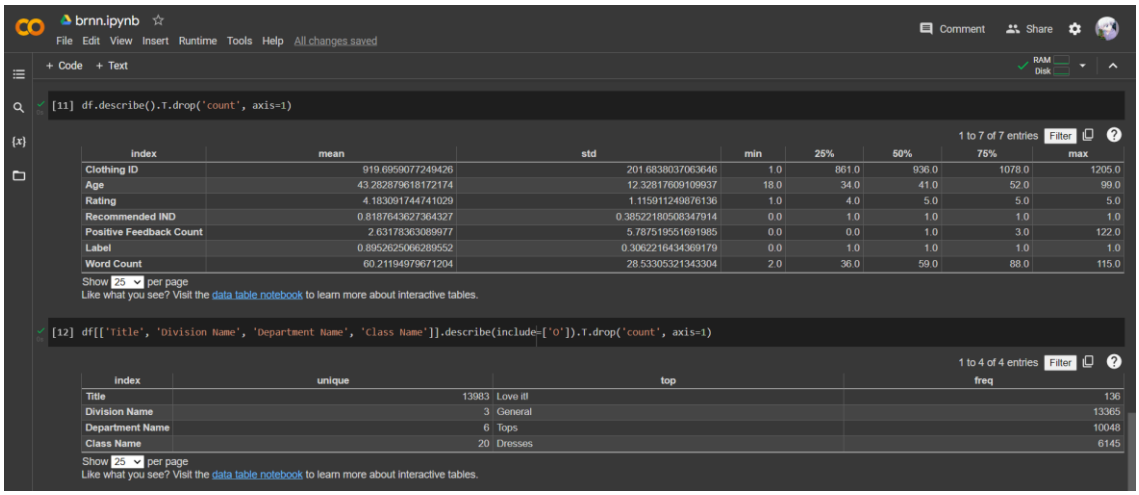
Figure 3.5 shows the tail code to display last 5 columns inside the dataset. For the below code is the shape code to display total of the column and row inside the dataset.



	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name	Label	Word Count
17832	975	32	NaN	Just purchased this on sale. will be perfect f...	5	1	0	General Petite	Jackets	Jackets	1	28
20675	862	37	Simple & structured	I love this tank. i got it in the turquoise an...	5	1	3	General Petite	Tops	Knits	1	76
16112	862	40	NaN	For me, this top ran true to size. i'm so glad...	5	1	6	General	Tops	Knits	1	109
19790	307	33	Love the floral pattern	I love this sports bra. i do various exercises...	5	1	3	Initmates	Intimate	Lounge	1	54
13381	1072	40	Beautiful print!	I saw this dress in the store & loved the prin...	5	1	0	General	Dresses	Dresses	1	102

Figure 3.6: Coding to display the exact Rating and Label

Figure 3.6 shows the coding to display the sample from the datasets that meet with any rating that equal or higher than 3 and have the label of 1.



index	mean	std	min	25%	50%	75%	max
Clothing ID	919.6959077249426	201.6838037063646	1.0	861.0	936.0	1078.0	1205.0
Age	43.282879618172174	12.328176091099937	18.0	34.0	41.0	52.0	99.0
Rating	4.183091744741029	1.115911249876136	1.0	4.0	5.0	5.0	5.0
Recommended IND	0.8187643627364327	0.38522180508347914	0.0	1.0	1.0	1.0	1.0
Positive Feedback Count	2.63178363089977	5.787519551691985	0.0	0.0	1.0	3.0	122.0
Label	0.8952625066289552	0.3062216434369179	0.0	1.0	1.0	1.0	1.0
Word Count	60.21194979671204	28.53305321343304	2.0	36.0	59.0	88.0	115.0

index	unique	top	freq
Title	13983	Love it!	136
Division Name	3	General	13365
Department Name	6	Tops	10048
Class Name	20	Dresses	6145

Figure 3.7: Coding to count the data from the title of column

Figure 3.7 shows the coding to display the analysis number to display the analysis from the columns inside the datasets. Foe the below code is to count and display the unique number from column Title, Division Name, Department Name and Class Name.

3.6 Testing and validation plan

For the testing and validation plan, the datasets will be going through the data preprocessing phase such as text cleaning and word embeddings. Then, the data will be split through the training, validation and testing dataset. The partition used is 60/20/20. 60% represents training dataset partition. While 20% represents the validation and testing dataset process. The process for training, validation and testing datasets will be run in Jupyter Notebook using Python code in training phase from the dataset. Then, the next validation process is through the hyper parameter tuning. Hyper parameters used in BRNN-LSTM are batch size, cell size, dropout rate, epochs and learning rate.

3.7 Potential Use of Proposed Solution

From the expected output from this research which is the relationship between recommendation and the review text from customer, it can be very helpful in many ways. Firstly, for the retailer or owner of the business, they can easily predict the highest demand of the product and also the lowest from the analysis and expected output from this research work then take the significant action. Retailer or owner can request to the supplier to produce more product based on the highest demand product category or the similar type from that specific product.

Then, retailer can also improve for the positive-future sides from the received review comments from the customer. Even the raw data for the text review, retailer can simply read or do the analysis from what the majority of customers desire for the improvement for the business. Such as, the quality from the material of the products, the colour variation from the products and even the availability for certain products.

The application in sentiment analysis using machine learning for this matter will be the solution for retailer to discuss and plan for the future improvement for the sake of their business. From the recommendation relationship with the sentiment from text review, retailer or owner can classify which products that have many 0 point for recommendation and planning for the wise action to transform the product to be better.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the content will discuss the results and discussions throughout the research journey, including the proposed technique, algorithms and steps employed to achieve optimal output.

4.2 Early development

From this early evaluation or development process, there are a few libraries need to be import to produce the good output. The libraries imported are ‘matplotlib’, ‘numpy’, ‘pandas’, ‘seaborn’ and ‘wordcloud’. For this early evaluation process, all the codes are more to display simple things such as testing to display the first 5 rows from the dataset to ensure the data is correctly read from the dataset.

```
In [3]: import matplotlib.pyplot as plt
        import nltk
        import numpy as np
        import pandas as pd
        import seaborn as sns
        import io
        from wordcloud import WordCloud, STOPWORDS

In [4]: pip install wordcloud

        Requirement already satisfied: wordcloud in c:\users\user\anaconda3\lib\site-packages (1.9.1.1)
        Requirement already satisfied: matplotlib in c:\users\user\anaconda3\lib\site-packages (from wordcloud) (3.4.3)
        Requirement already satisfied: pillow in c:\users\user\anaconda3\lib\site-packages (from wordcloud) (8.4.0)
        Requirement already satisfied: numpy>=1.6.1 in c:\users\user\anaconda3\lib\site-packages (from wordcloud) (1.22.4)
        Requirement already satisfied: python-dateutil>=2.7 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordcloud)
        (2.8.2)
        Note: you may need to restart the kernel to use updated packages.Requirement already satisfied: pyparsing>=2.2.1 in c:\users\us
        er\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.4)
        Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.
        3.1)
        Requirement already satisfied: cycler>=0.10 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.10.0)
        Requirement already satisfied: six in c:\users\user\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib->wordcloud) (1.1
        6.0)
```

Figure 4.1: Early import code

Figure 4.1 displays the initial codes for executing the sentiment analysis work. the first code imports the necessary libraries to begin coding. Following that, the subsequent code installs the ‘wordclouds’ library before importing it.

```
In [5]: df = pd.read_csv('Womens Clothing E-Commerce Reviews.csv')
        df.head()
```

Out[5]:

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	Tops	Blouses

```
In [6]: df.tail()
```

Out[6]:

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
23481	23481	1104	34	Great dress for many occasions	I was very happy to snag this dress at such a ...	5	1	0	General Petite	Dresses	Dresses
23482	23482	862	48	Wish it was made of cotton	It reminds me of maternity clothes. soft, stre...	3	1	0	General Petite	Tops	Knits
23483	23483	1104	31	Cute, but see through	This fit well, but the top was very see throug...	3	0	1	General Petite	Dresses	Dresses
23484	23484	1084	28	Very cute dress, perfect for summer parties an...	I bought this dress for a wedding i have this ...	3	1	2	General	Dresses	Dresses
23485	23485	1104	52	Please make more like	This dress in a lovely platinum is feminine	5	1	22	General	Dresses	Dresses

Figure 4.2: Head and tail code

Figure 4.2 shows the head and tail code to display the first and last 5 rows of data from the datasets. The code is important to ensure the dataset is usable.

```
In [9]: df['Label'] = 0
In [10]: df.loc[df.Rating >= 3, ['Label']] = 1
In [11]: df['Word Count'] = df['Review Text'].str.split().apply(len)
In [12]: df.sample(5)
```

Out[12]:

	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name	Label	Word Count
19771	818	42	Lovely flowing top with detail	Contrary to the first review, i find this top ...	5	1	2	General	Tops	Blouses	1	76
1300	850	65	Wanted to love	I bought two, one in white and one in blue pri...	2	0	1	General	Tops	Blouses	0	103
16273	839	62	Great top	I love the ease of wearing and styling this to...	5	1	0	General Petite	Tops	Blouses	1	21
20926	1009	49	NaN	It's difficult to tell, but the skirt is 3 col...	4	1	2	General	Bottoms	Skirts	1	76
1511	1080	39	Red "evanthe" dress	Agreed with the previous reviewer, this is the...	4	1	1	General	Dresses	Dresses	1	92

Figure 4.3: Drop column and sample code

Figure 4.3 at the first box shows to create the Label column is ‘0’ for all the dataset, then adding the new rule if the Rating is equal or more than 3, the Label is

displayed as ‘1’. The second box is the code to execute the sample after running the early code.

4.3 Analysis and Visualization

This subtopic is the documentation for analysis and visualization from all the columns stated in the dataset. The analysis was made to recognize the relationship between one column to another column. The purpose for this analysis is to view the relationship between the data to make the process for sentiment classification easier.

```
In [59]: row_plots = ['Division Name', 'Department Name']
         f, axes = plt.subplots(1, len(row_plots), figsize=(14, 4), sharex=False)

         for i, x in enumerate(row_plots):
             sns.countplot(y=x, data=df, order=df[x].value_counts().index, ax=axes[i])
             axes[i].set_title('Count of Categories in {}'.format(x))
             axes[i].set_xlabel('')
             axes[i].set_xlabel('Frequency Count')
         axes[0].set_ylabel('Category')
         axes[1].set_ylabel('')
         plt.savefig('divname-and-deptname-freqdist.png', format='png', dpi=600)
         plt.show()
```

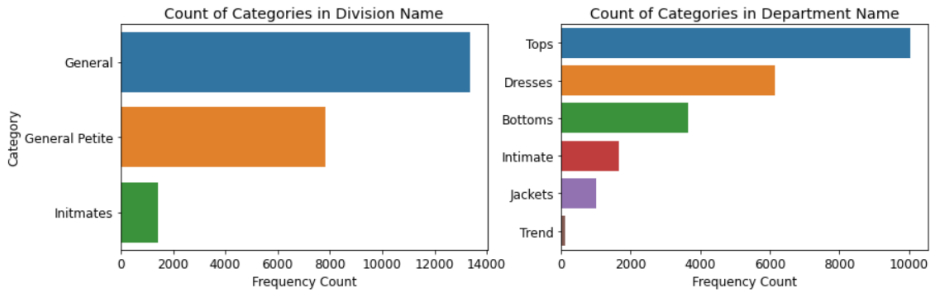


Figure 4.4: Distribution for Division and Department Name

Figure 4.4 shows the code and output visualization to display the Division Name and Department Name inside the dataset. The code and output to show the count categories for both division and department name.

```
In [18]: # Class Name
plt.subplots(figsize=(12, 8))
sns.countplot(y='Class Name', data=df,order=df['Class Name'].value_counts().index)
plt.title('Frequency Distribution of Class Name')
plt.xlabel('Frequency')
plt.tight_layout()
plt.savefig('freqdist-classname.png', format='png', dpi=300)
plt.show()
```

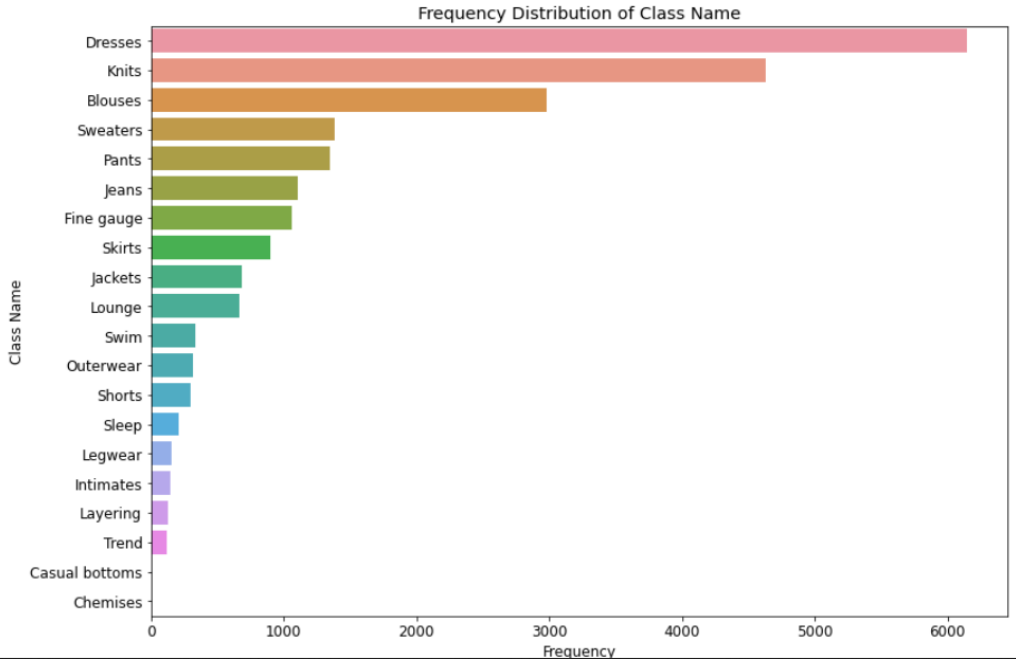


Figure 4.5: Frequency Distribution code and graph of Class Name

Figure 4.5 shows the code to display the visualization in form of bar graph for Class Name inside the dataset. The class name was utilized by the store to categorize each clothing item before commencing sales on the website.

```
In [61]: cat_dtypes = ['Rating', 'Recommended IND', 'Label']
increment = 0
f, axes = plt.subplots(1, len(cat_dtypes), figsize=(16, 6), sharex=False)

for i in range(len(cat_dtypes)):
    sns.countplot(x=cat_dtypes[increment], data=df, order=df[cat_dtypes[increment]].value_counts().index, ax=axes[i])
    axes[i].set_title('Frequency Distribution for\n{}'.format(cat_dtypes[increment]))
    axes[i].set_ylabel('Occurrence')
    axes[i].set_xlabel('{}'.format(cat_dtypes[increment]))
    increment += 1
axes[1].set_ylabel('')
axes[2].set_ylabel('')
plt.savefig('freqdist-rating-recommended-label.png', format='png', dpi=300)
plt.show()
```

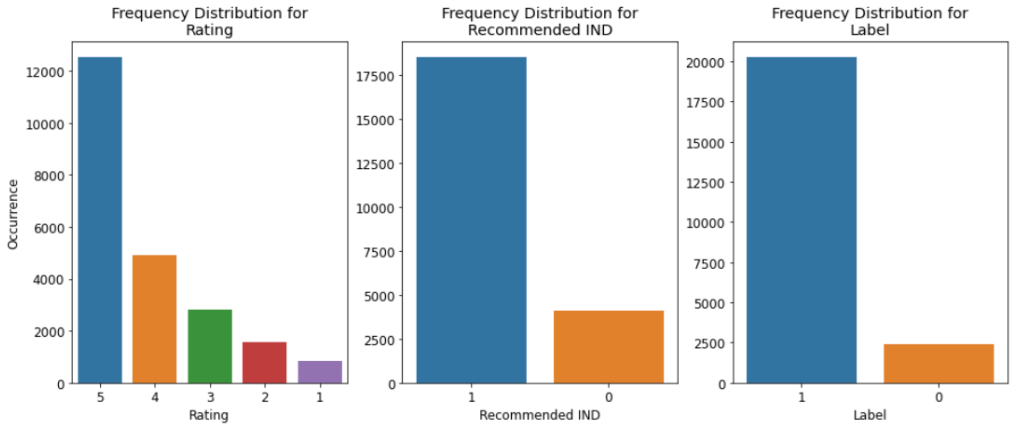


Figure 4.6: Frequency Distribution of Rating, Recommended IND and Label

Figure 4.6 displays the bar graph that was generated from the coding at figure 4.7. The chart shows the frequency distribution for Rating, Recommended IND and the Label. Then, it shows the occurrence for each Rating, Recommended IND and Label numbers.

```
In [64]: def percentstandardize_barplot(x, y, hue, data, ax=None, order=None):
    """
    Standardize by percentage the data using pandas functions, then plot using Seaborn.
    Function arguments are and extention of Seaborns'.
    """
    sns.barplot(x=x, y=y, hue=hue, ax=ax, order=order,
    data=(data[[x, hue]]
          .reset_index(drop=True)
          .groupby([x])[hue]
          .value_counts(normalize=True)
          .rename('Percentage').mul(100)
          .reset_index()
          .sort_values(hue)))
    plt.title('Percentage Frequency of {} by {}'.format(hue, x))
    plt.ylabel('Percentage %')
```

Figure 4.7: Declaration and Definition for Visualization

```
In [65]: huevar = 'Recommended IND'
f, axes = plt.subplots(1, 2, figsize=(16, 7))
percentstandardize_barplot(x='Department Name', y='Percentage', hue=huevar,data=df, ax=axes[0])
axes[0].set_title('Percentage Frequency of {}\nby Department Name'.format(huevar))
axes[0].set_ylabel('Percentage %')
percentstandardize_barplot(x='Division Name', y='Percentage', hue=huevar, data=df, ax=axes[1])
axes[1].set_title('Percentage Frequency of {}\nby Division Name'.format(huevar))
axes[1].set_ylabel('')
plt.savefig('recommended-deptname-divname.png', format='png', dpi=300)
plt.show()
```

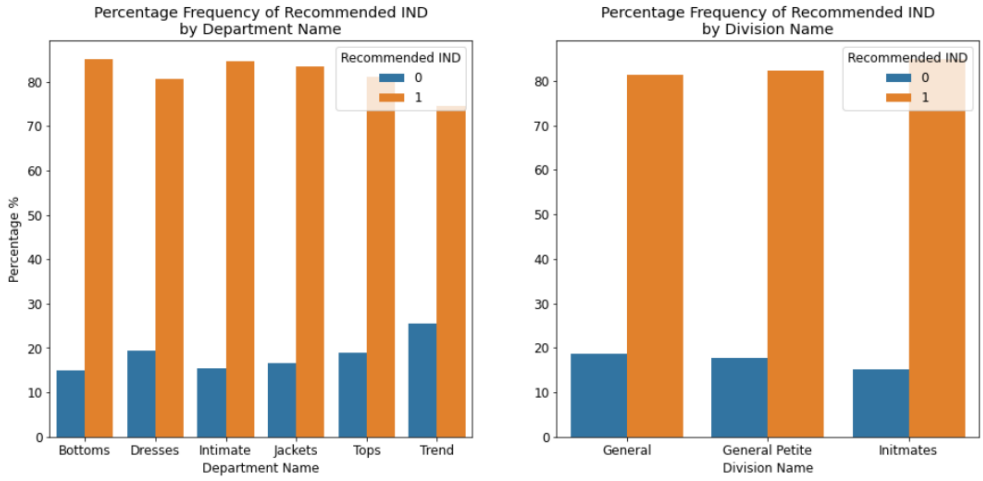


Figure 4.8: Comparison Recommended IND by Department and Division Name

Figure 4.8 shows the comparison in percentage for Recommended IND with Department Name and Division Name. For Recommended IND by Department Name, it shows that '1' has the big percentage compared to '0'. Similar to Division Name, the graph also shows that '1' also has the big percentage while '0' has the low percentage.

```
In [67]: xvar = ['Department Name', 'Division Name']
huevar = 'Rating'
f, axes = plt.subplots(1, 2, figsize=(16, 7))
percentstandardize_barplot(x=xvar[0], y='Percentage', hue=huevar, data=df, ax=axes[0])
axes[0].set_title('Percentage Frequency of {}\nby {}'.format(huevar, xvar[0]))
axes[0].set_ylabel('Percentage %')
percentstandardize_barplot(x=xvar[1], y='Percentage', hue=huevar, data=df, ax=axes[1])
axes[1].set_title("Percentage Frequency of {}\nby {}".format(huevar, xvar[1]))
plt.savefig('rating-deptname-divname.png', format='png', dpi=300)
plt.show()
```

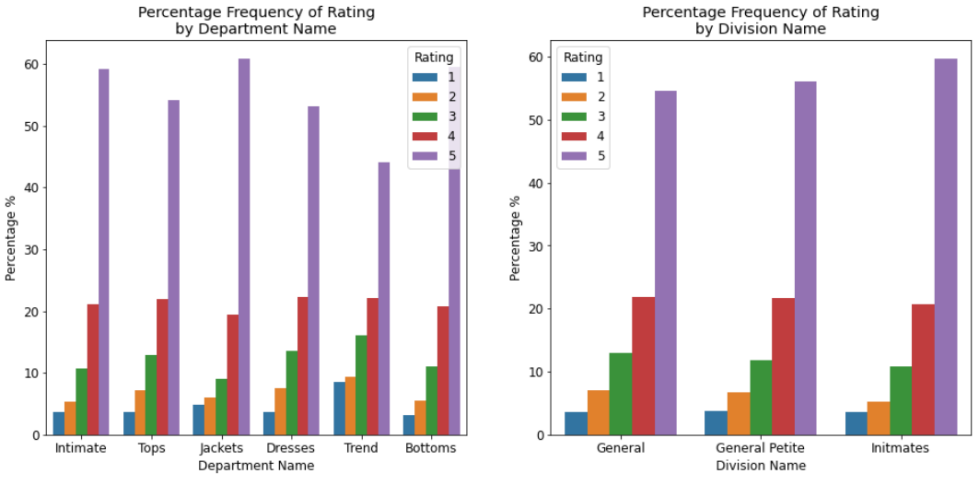
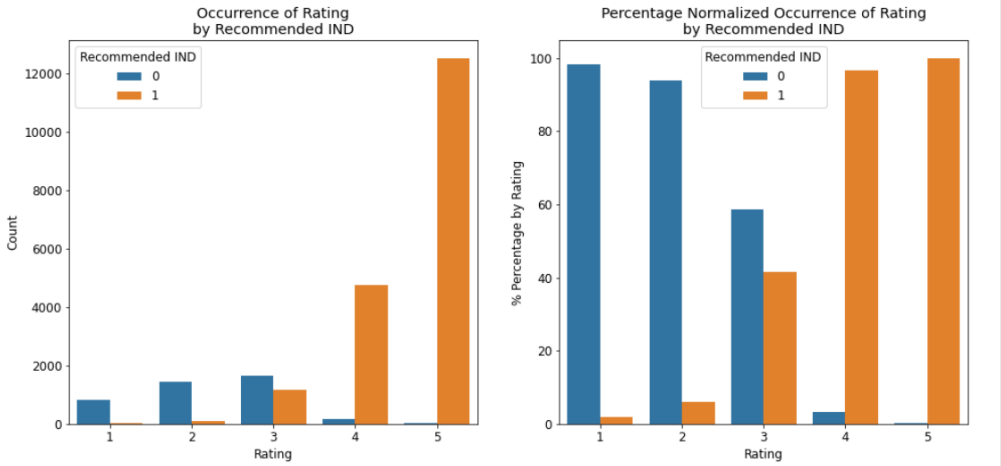


Figure 4.9: Rating by Department and Division Name

Figure 4.9 shows that Rating inside both Department and Division Name. This analysis purpose is to observe which one of the five rating has the highest amount for Division and Department Name.

```
In [68]: huevar = 'Rating'
         f, axes = plt.subplots(1, 2, figsize=(16, 7))
         sns.countplot(x='Rating', hue='Recommended IND', data=df, ax=axes[0])
         axes[0].set_title('Occurrence of {}\nby {}'.format(huevar, 'Recommended IND'))
         axes[0].set_ylabel('Count')
         percentstandardize_barplot(x='Rating', y='Percentage', hue='Recommended IND', data=df, ax=axes[1])
         axes[1].set_title('Percentage Normalized Occurrence of {}\nby {}'.format(huevar, 'Recommended IND'))
         axes[1].set_ylabel('% Percentage by Rating')
         plt.savefig('rating-recommended.png', format='png', dpi=300)
         plt.show()
```



Therefore, the analysis works done is to extract which of the features are required to do the sentiment work. The features from the dataset that need to be used are Recommended IND, Rating, Review Text and Department Name. All these features need to be included in the testing, validating and training phases.

4.4 Text Cleaning

Text cleaning is one of the datasets pre-processing phase. Another pre-processing are sentiment analysis and word embeddings. The purpose for this step is essential for any machine learning evaluation that used textual data. Text cleaning process will remove irrelevant information from the Review Text in the dataset then transform it to the consistent representation data that will be more easily to execute later.

```
In [17]: import nltk
         nltk.download('stopwords')

         [nltk_data] Downloading package stopwords to
         [nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
         [nltk_data]   Package stopwords is already up-to-date!

Out[17]: True

In [18]: from nltk.corpus import stopwords
         from nltk.stem.porter import PorterStemmer
         from nltk.tokenize import RegexpTokenizer

         ps = PorterStemmer()

         tokenizer = RegexpTokenizer(r'\w+')
         stop_words = set(stopwords.words('english'))

         def preprocessing(data):
             txt = data.str.lower().str.cat(sep=' ') #1
             words = tokenizer.tokenize(txt) #2
             words = [w for w in words if not w in stop_words] #3
             #words = [ps.stem(w) for w in words] #4
             return words
```

Figure 4.10: Importing nltk code and Text Cleaning

Figure 4.10 at the top shows the code to import ‘nltk’ and downloading library for ‘stopwords’. Then, the below code is to do the text cleaning for the dataset. The library included for this code is ‘stopwords’, ‘PorterStemmer’ and ‘RegexpTokenizer’.

4.5 Sentiment Analysis

```
In [28]: import nltk
         nltk.download('vader_lexicon')

         [nltk_data] Downloading package vader_lexicon to
         [nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
         [nltk_data]   Package vader_lexicon is already up-to-date!

Out[28]: True

In [29]: from nltk.sentiment.vader import SentimentIntensityAnalyzer

         # Pre-Processing
         SIA = SentimentIntensityAnalyzer()
         df['Review Text']= df['Review Text'].astype(str)

         # Applying Model, Variable Creation
         df['Polarity Score'] = df['Review Text'].apply(lambda x: SIA.polarity_scores(x)['compound'])
         df['Neutral Score'] = df['Review Text'].apply(lambda x: SIA.polarity_scores(x)['neu'])
         df['Negative Score'] = df['Review Text'].apply(lambda x: SIA.polarity_scores(x)['neg'])
         df['Positive Score'] = df['Review Text'].apply(lambda x: SIA.polarity_scores(x)['pos'])

         # Converting 0 to 1 Decimal Score to a Categorical Variable
         df['Sentiment'] = ''
         df.loc[df['Polarity Score'] > 0, 'Sentiment'] = 'Positive'
         df.loc[df['Polarity Score'] == 0, 'Sentiment'] = 'Neutral'
         df.loc[df['Polarity Score'] < 0, 'Sentiment'] = 'Negative'
```

Figure 4.11: Code Pre-processing for Sentiment Analysis

Figure 4.11 shows at the first box is the code for importing the ‘nltk’ and downloading ‘vader_lexicon’ library. Then, in line 29 the first box shows the code to do pre-processing for the Review Text column. The subsequent box applies the model for three sentiments: positive, negative and neutral. It also displays the polarity score from the Review Text column. The last box showing the code to converting the number 0 and 1 to become as the statement and represent the sentiment.

```
In [22]: huevar = 'Recommended IND'
xvar = 'Sentiment'
f, axes = plt.subplots(1, 2, figsize=(16, 9))
sns.countplot(x=xvar, hue=huevar,data=df, ax=axes[0], order=['Negative', 'Neutral', 'Positive'])
axes[0].set_title('Occurence of {}\nby {}'.format(xvar, huevar))
axes[0].set_ylabel('Count')
percentstandardize_barplot(x=xvar, y='Percentage', hue=huevar, data=df, ax=axes[1])
axes[1].set_title('Percentage Normalized Occurence of {}\nby {}'.format(xvar, huevar))
axes[1].set_ylabel('% Percentage by {}'.format(huevar))
plt.savefig('norm-sentimentdist.png', format='png', dpi=300)
plt.show()
```

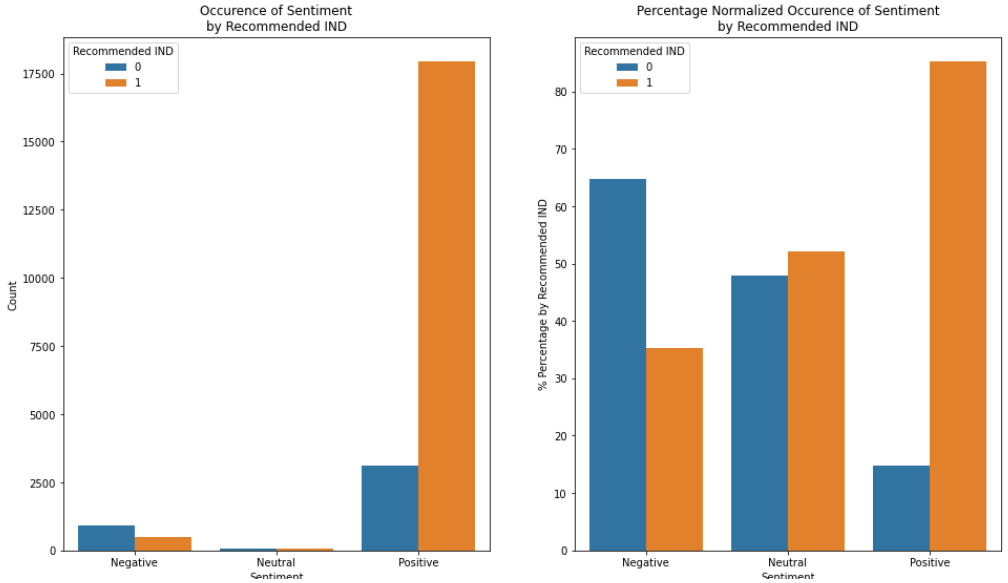


Figure 4.12: Bar chart for Sentiment Distribution

Figure 4.12 shows the count for the three sentiment which are negative, neutral and positive sentiment. For the left graph, it shows that positive sentiment has the highest count while neutral has the lowest both for '0' and '1' Recommended IND. While the right graph shows for Recommended IND '1' positive sentiment has the highest while the highest for '0' Recommended IND is negative sentiment.

```
In [23]: f, axes = plt.subplots(2, 2, figsize=[10, 10])
sns.countplot(x='Sentiment', data=df, ax=axes[0, 0], order=['Negative', 'Neutral', 'Positive'])
axes[0,0].set_xlabel('Sentiment')
axes[0,0].set_ylabel('Count')
axes[0,0].set_title('Overall Sentiment Occurrence')

sns.countplot(x='Rating', data=df, ax=axes[0, 1])
axes[0,1].set_xlabel('Rating')
axes[0,1].set_ylabel('')
axes[0,1].set_title('Overall Rating Occurrence')

percentstandardize_barplot(x='Rating', y='Percentage', hue='Sentiment', data=df, ax=axes[1, 0])
axes[1,0].set_xlabel('Rating')
axes[1,0].set_ylabel('Percentage %')
axes[1,0].set_title('Standardized Percentage Rating Frequency\nby Sentiment')

percentstandardize_barplot(x='Sentiment', y='Percentage',hue='Rating', data=df, ax=axes[1, 1])
axes[1,1].set_ylabel('Occurrence Frequency')
axes[1,1].set_title('Standardized Percentage Sentiment Frequency\nby Rating')
axes[1,1].set_xlabel('Sentiment')
axes[1,1].set_ylabel('')

f.suptitle('Distribution of Sentiment Score and Rating for Customer Reviews', fontsize=14)
f.tight_layout()
f.subplots_adjust(top=0.92)
plt.savefig('sentimentscoredist-rating.png', format='png', dpi=300)
plt.show()
```

Figure 4.13: Code for Relationship between Column to Sentiment

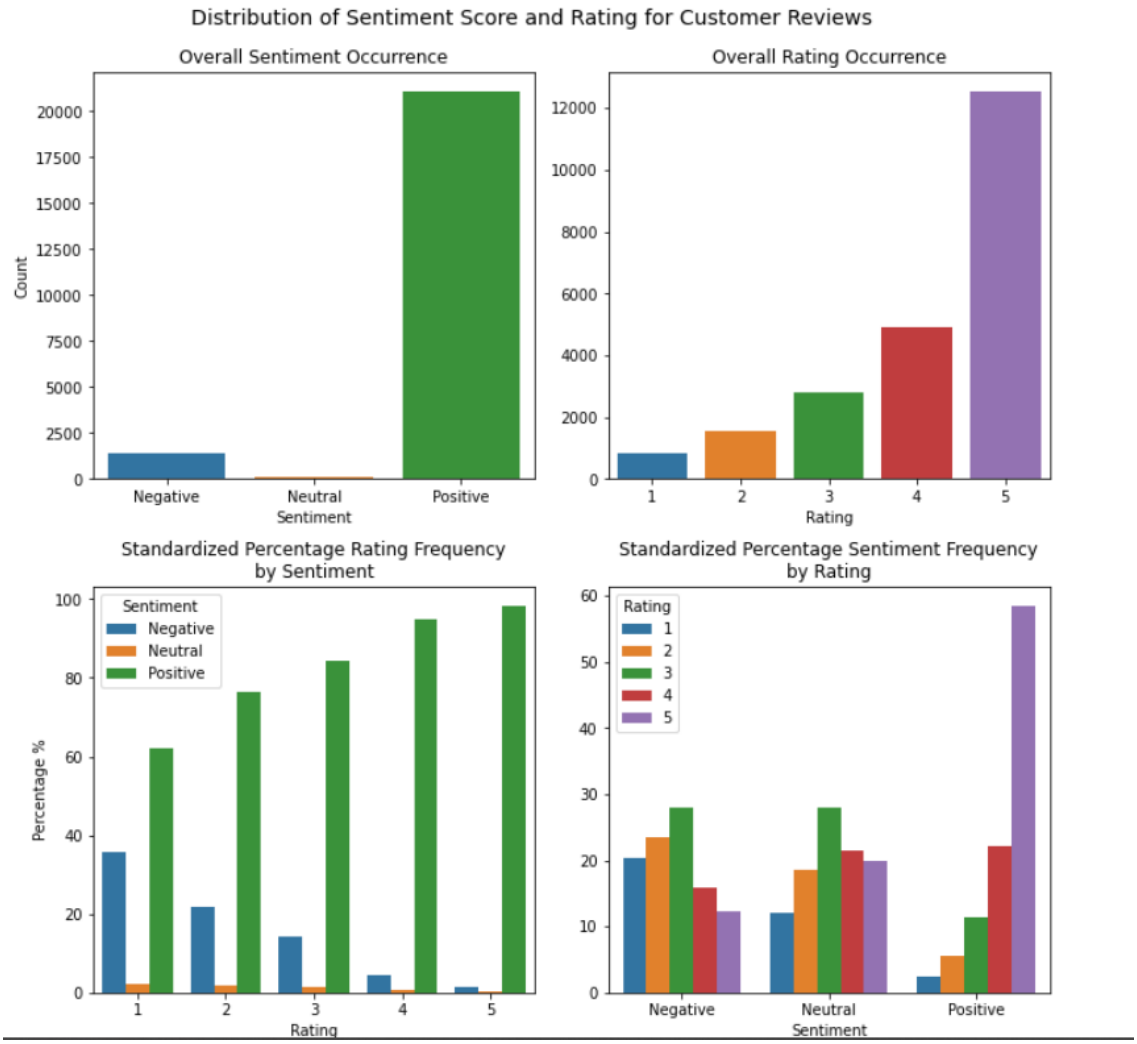


Figure 4.14: Visualization for all the Sentiment with Columns

Both figures 4.13 and 4.14 show the code and the graphs for the sentiment analysis. The first graph shows overall sentiment occurrence with the count for each sentiment. The second graph shows the overall rating occurrence. The third graph shows the percentage rating frequency by all three sentiments. The last graph shows the percentage sentiment frequency by the Rating.

In [24]:

```
xvar = 'Sentiment'
huevar = 'Department Name'
rowvar = 'Recommended IND'

# Plot
f, axes = plt.subplots(2, 2, figsize=(12, 12), sharex=False, sharey=False)
for i,x in enumerate(set(df[rowvar][df[rowvar].notnull()])):
    percentstandardize_barplot(x=xvar, y='Percentage', hue=huevar, data=df[df[rowvar] == x],
                               ax=axes[i,0], order=['Negative','Neutral','Positive'])
    percentstandardize_barplot(x=xvar, y='Percentage', hue='Rating', data=df[df[rowvar] == x],
                               ax=axes[i,1], order=['Negative','Neutral','Positive'])

# Plot Aesthetics
axes[1,0].legend_.remove()
axes[1,1].legend_.remove()
axes[0,1].set_ylabel('')
axes[1,1].set_ylabel('')
axes[0,0].set_xlabel('')
axes[0,1].set_xlabel('')
axes[0,0].set_ylabel('Recommended = FALSE\nPercentage %')
axes[1,0].set_ylabel('Recommended = TRUE\nPercentage %')
axes[1,1].set_title('')

# Common title and ylabel
f.text(0.0, 0.5, 'Subplot Rows\nSliced by Recommended', va='center', rotation='vertical', fontsize=12)
f.suptitle('Review Sentiment by Department Name and Rating\nSubplot Rows Slice Data by Recommended', fontsize=16)
f.tight_layout()
f.subplots_adjust(top=0.93)
plt.savefig('sentiment-deptname-rating-recommended.png', format='png', dpi=300)
plt.show()
```

Figure 4.15: Code for Review Sentiment by Department Name and Rating

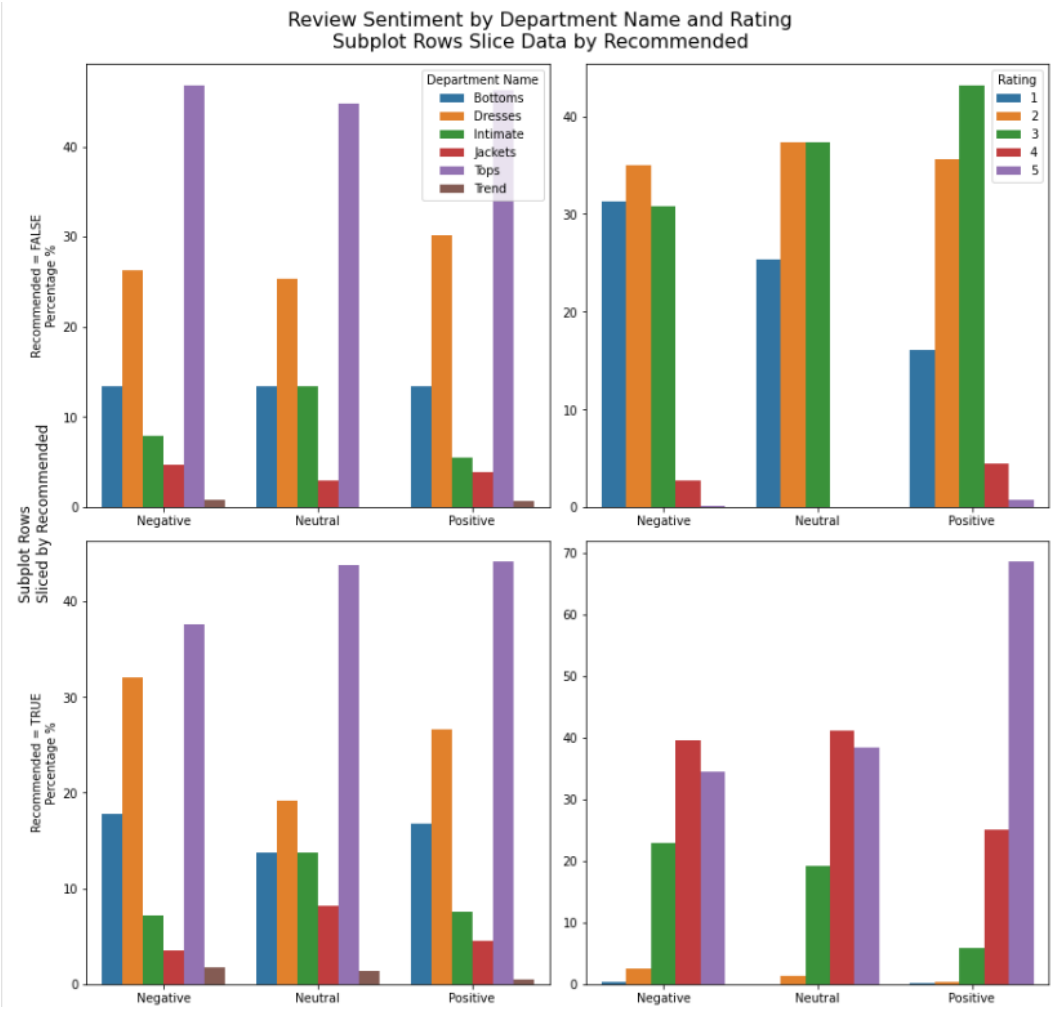


Figure 4.16: All visualization from code above

Figure 4.15 and 4.16 show the code and graphs for the review for sentiment analysis. The 2 left bar graph shows the relationship between department name and sentiment classifications with Recommended IND percentage. Then, the 2 right graphs shows the relationship between sentiment classification with the Rating in percentage. For the first 2 left graph is about the Department name related with Recommended IND true or false. It shows that for both graph, 'Tops' is a product that have highest review for all of three sentiments. For the right graph it displays that Rating is related to Recommended IND true or false. If the Recommended IND is False, it shows that the highest rating is '3' and the lowest is '5' while if the Recommended IND is True, the highest rating is '5' while the lowest is '1'.

4.6 Supervised learning

From this phase, datasets are going through the training, validation and testing phase. Then, the last dataset pre-processing also from this step which is word embeddings. There are many libraries imported in this steps such as ‘to_categorical’, ‘pad_sequences’ and ‘Tokenizer’. This step is the last step before the sentiment classification and displaying the results and output. From this step, all the data already cleaned and ready to be inserted for sentiment classification.

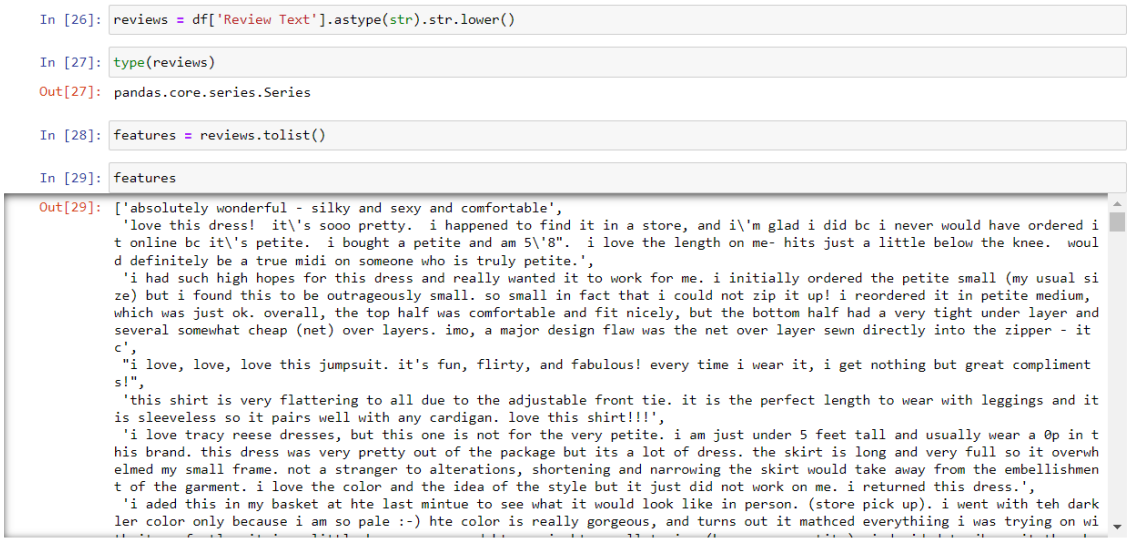


Figure 4.17: Preparation code for supervised learning

Figure 4.17 displays the preparation code for the initial phase of supervised learning. Following the code, the output is generated by the features of the code, which processes the review text contents within the dataset.

```
In [30]: import re
         from string import punctuation

In [31]: for index in range(len(features)):
             all_text = ''.join([character for character in features[index] if character not in punctuation])
             features[index] = re.split(r'\n|\r', all_text)
             features[index] = ' '.join([word for word in features[index]])

In [32]: features

Out[32]: ['absolutely wonderful  silky and sexy and comfortable',
          'love this dress  its sooo pretty  i happened to find it in a store and im glad i did bc i never would have ordered it onlin
         e bc its petite  i bought a petite and am 58  i love the length on me hits just a little below the knee  would definitely be
         a true midi on someone who is truly petite',
          'i had such high hopes for this dress and really wanted it to work for me i initially ordered the petite small my usual size
         but i found this to be outrageously small so small in fact that i could not zip it up i reordered it in petite medium which w
         as just ok overall the top half was comfortable and fit nicely but the bottom half had a very tight under layer and several s
         omewhat cheap net over layers imo a major design flaw was the net over layer sewn directly into the zipper  it c',
          'i love love love this jumpsuit its fun flirty and fabulous every time i wear it i get nothing but great compliments',
          'this shirt is very flattering to all due to the adjustable front tie it is the perfect length to wear with leggings and it
         is sleeveless so it pairs well with any cardigan love this shirt',
          'i love tracy reese dresses but this one is not for the very petite i am just under 5 feet tall and usually wear a 0p in thi
         s brand this dress was very pretty out of the package but its a lot of dress the skirt is long and very full so it overwhelme
         d my small frame not a stranger to alterations shortening and narrowing the skirt would take away from the embellishment of t
         he garment i love the color and the idea of the style but it just did not work on me i returned this dress',
          'i aded this in my basket at hte last mintue to see what it would look like in person store pick up i went with teh darkler
         color only because i am so pale  hte color is really gorgeous and turns out it mathced everythiing i was trying on with it pr
         efectly it is a little baggy on me and hte xs is hte msallet size bummer no petite i decided to jkeep it though because as i
         said it matvehd everything my ejans pants and the 3 skirts i waas trying on of which i kept all  oops',
```

Figure 4.18: Removing the punctuation

Figure 4.18 shows the coding to remove the punctuation for the existing review text to make the words easier to analyze it. All the punctuations inside the text has been removed.

```
In [33]: labels = np.array(df['Recommended IND'], dtype=int)

In [34]: labels.shape

Out[34]: (22628,)

In [35]: labels[labels == 1].shape[0]

Out[35]: 18527

In [36]: labels[labels == 0].shape[0]

Out[36]: 4101
```

Figure 4.19: Shape code, Labels code

Figure 4.19 shows the code for shape. Then, it shows the coding for label for the Label = 0 and Label =1.

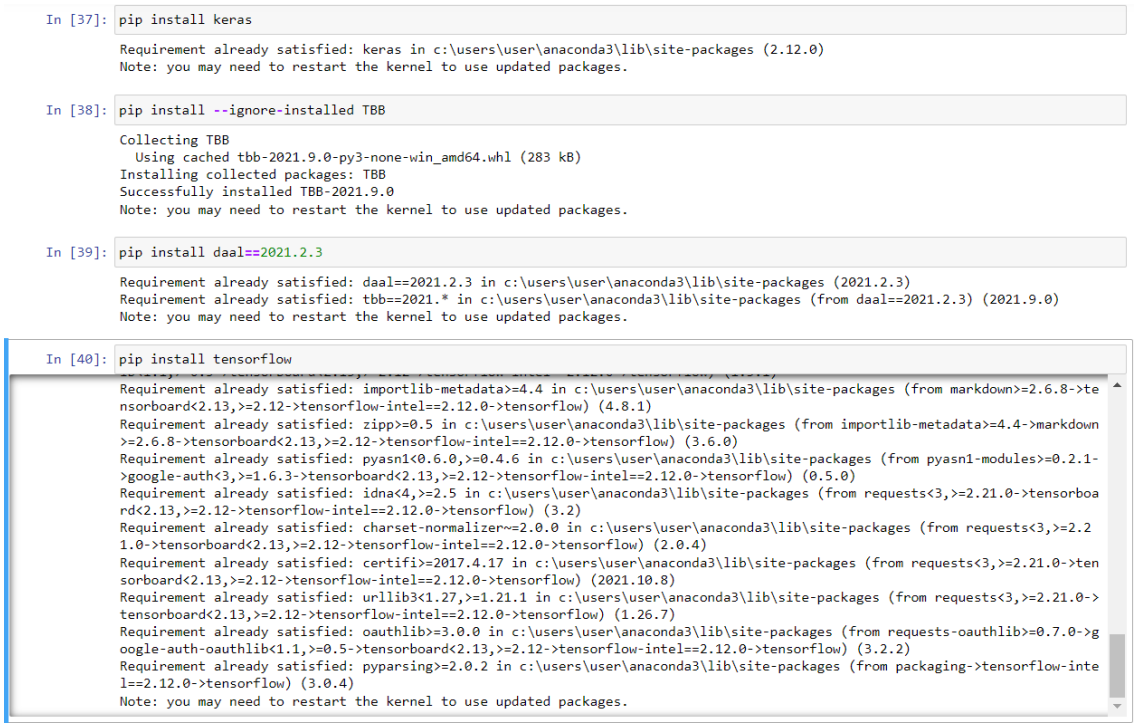


Figure 4.20: Installation requirement

Figure 4.20 shows the installation requirement for the supervised learning works. Need to install Keras, daal and TensorFlow library to make the code after this can be work.

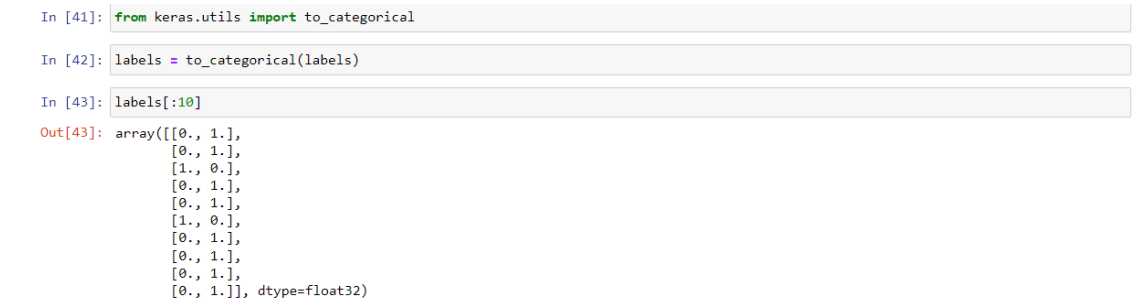


Figure 4.21: Code for Import Keras library

Figure 4.21 shows the coding to import the categorical from keras utilities. Then, displays the array for the label.

```
In [45]: from tensorflow.keras.preprocessing.sequence import pad_sequences
         from tensorflow.keras.preprocessing.text import Tokenizer

In [46]: t = Tokenizer()
         t.fit_on_texts(features)
         vocabulary_size = len(t.word_index) + 1

In [47]: print('Vocabulary size : {}'.format(vocabulary_size))

         Vocabulary size : 19370

In [48]: encoded_features = t.texts_to_sequences(features)

         max_length = 300

         padded_features = pad_sequences(encoded_features, maxlen=max_length, padding='post')
```

Figure 4.22: Tokenizer code and print Vocabulary size

Figure 4.22 at the top of coding shows the process to import the pad sequences and 'Tokenizer' to 'Tensorflow' preprocessing text. Then, it shows the code to print the vocabulary size. Lastly, code for the features for text to be displayed and analyzed.

```
In [57]: embeddings_index = dict()
         with open('glove.840B.300d.txt', encoding='utf-8') as file:
             data = file.readlines()

         # store <key, value> pair of FastText vectors
         for line in data[1:]:
             word, vec = line.split(' ', 1)
             embeddings_index[word] = np.array([float(index) for index in vec.split()], dtype='float32')
         print('Loaded {} word vectors.'.format(len(embeddings_index)))


         embedding_matrix = np.zeros((vocabulary_size, max_length))
         for word, i in t.word_index.items():
             embedding_vector = embeddings_index.get(word)
             if embedding_vector is not None:
                 embedding_matrix[i] = embedding_vector
                 embedding_matrix[i] = embedding_vector

         Loaded 2196016 word vectors.
```

Figure 4.23: Code to load the word vectors

Figure 4.23 shows the code to load the word vectors and the purpose to load the word vectors is to do the word embeddings and embedding matrix. Word embeddings usually used to initialize the layer for neural networks for Natural Language Processing (NLP) tasks such as text classification and sentiment analysis. While the embedding matrix function is to initialize the embedding layers for NLP model.

```
In [58]: words = []
         for word, i in t.word_index.items():
             embedding_vector = embeddings_index.get(word)
             if embedding_vector is not None:
                 words.append(word)

In [59]: print('{} words covered.'.format(len(words)))

         13911 words covered.

In [60]: percentage = (len(words) / vocabulary_size) * 100.00
         print('{}% of {} words were covered'.format(percentage, vocabulary_size))

         71.81724315952503% of 19370 words were covered
```

Figure 4.24: Code to display words after embedding vectors

Figure 4.24 shows the code to display the word vectors and the below code is to display the vocabulary size in percentage format. This code important to recognize and determine the length of the words covered from the datasets.

```
In [61]: def train_test_split(features, labels, **kwargs):

             # concatenate the features and labels array
             dataset = np.c_[features, labels]

             # shuffle the dataset
             np.random.shuffle(dataset)

             # split the dataset into features, labels
             features, labels = dataset[:, :-1], dataset[:, -1]

             # get the split size for training dataset
             split_index = int(kwargs['train_size'] * len(features))

             # split the dataset into training/validation dataset
             train_features, validation_features = features[:split_index], features[split_index:]
             train_labels, validation_labels = labels[:split_index], labels[split_index:]

             # get the split size for validation dataset
             split_index = int(kwargs['validation_size'] * len(validation_features))

             # split the validation dataset into validation/testing dataset
             validation_features, test_features = validation_features[:split_index], validation_features[split_index:]
             validation_labels, test_labels = validation_labels[:split_index], validation_labels[split_index:]

             # return the partitioned dataset
             return [train_features, train_labels], [validation_features, validation_labels], [test_features, test_labels]
```

Figure 4.25: Training code

Figure 4.25 illustrates the code for training the data from the datasets. The first two lines of code depict the steps to concatenate and shuffle the datasets. The first red square of code signifies the process of splitting the data into training and validation sets. The second square of code represents the code used to further split the validation set into testing sets.

```
In [63]: print('Dataset size : {}'.format(padded_features.shape[0]))
         print('Train dataset size : {}'.format(train_dataset[0].shape[0]))
         print('Validation dataset size : {}'.format(validation_dataset[0].shape[0]))
         print('Test dataset size : {}'.format(test_dataset[0].shape[0]))

         Dataset size : 22628
         Train dataset size : 13576
         Validation dataset size : 4526
         Test dataset size : 4526
```

Figure 4.26: Display total from the past codes

Figure 4.26 shows the code to display all the numbers and total for dataset size, training dataset size, validation datasets size and testing dataset size. For the training dataset size is 60% from the total of dataset size. While validation and training dataset both represents 20% from dataset size.

4.7 Sentiment Classification

In the final step of the Result and Discussion chapter, the execution of the code resulted in the construction of the confusion matrices. This evaluation involved the utilization of several imported elements including ‘callbacks’, ‘Bidirectional’, ‘Dense’, ‘Dropout’, ‘Embedding’, ‘LSTM’, ‘Sequential’ and ‘StratifiedKFold’

```
In [64]: from keras import callbacks
         from keras.layers import Bidirectional
         from keras.layers import Dense
         from keras.layers import Dropout
         from keras.layers import Embedding
         from keras.layers import LSTM
         from keras.models import Sequential
         from sklearn.model_selection import StratifiedKFold
```

Figure 4.27: Libraries imported to execute result

Figure 4.27 shows the essential libraries that need to be imported to execute the results. The most essential libraries are ‘LSTM’ amd ‘Bidirectional’ to train the model followed by the chosen algorithm.

```
In [66]: labels = np.array(df['Sentiment'])

In [67]: labels

Out[67]: array(['Positive', 'Positive', 'Positive', ..., 'Positive', 'Positive',
                'Positive'], dtype=object)

In [68]: labels = np.array([2 if label == 'Positive' else (1 if label == 'Neutral' else 0) for label in labels],int)

In [69]: labels

Out[69]: array([2, 2, 2, ..., 2, 2, 2])
```

Figure 4.28: Labels detection code

Figure 4.28 shows the code to display the labels before sentiment classification steps. The first two lines of code show the steps to display all three sentiments in array format. Then, the next step is to declare all three sentiments to the integers format to make it easier to run.

```
In [70]: positive_class = int(labels[labels == 2].shape[0])
         neutral_class = int(labels[labels == 1].shape[0])
         negative_class = int(labels[labels == 0].shape[0])

         df = pd.DataFrame.from_dict({'positive': [positive_class], 'negative': [negative_class], 'neutral': [neutral_class]})

         plt.figure(figsize=(8, 8))
         sns.set(font_scale=2)
         sns.set_style('whitegrid')
         ax = sns.barplot(data=df)
         ax = ax.set_xlabel('Frequency Distribution of Sentiment Classes')
```

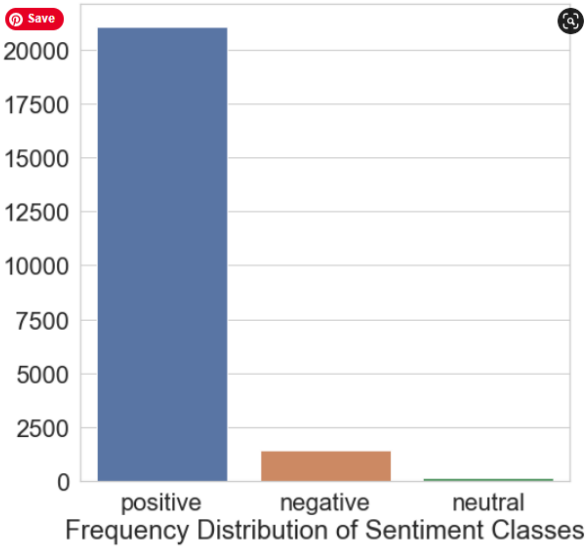


Figure 4.29: Code to build the bar graph from sentiment classification

In Figure 4.29, the accompanying bar graph illustrates the sentiment classification numbers depicted in the provided code. The graph indicates that positive sentiment has the highest count, while neutral has the lowest count among the three categories.

```
In [71]: labels = to_categorical(labels)

In [72]: train_dataset, validation_dataset, test_dataset = train_test_split(features=padded_features, labels=labels,
                                                                            train_size=0.60, validation_size=0.50)
```

Figure 4.30: Code to assign training and validation size of the dataset

Figure 4.30 shows the code to display the table in categorical format. Then, the second line of code is to assign the size number for validation and the train size of the dataset. The train size is set to 60% and the validation size is set to 50%.

```
In [73]: model = Sequential()
e = Embedding(vocabulary_size, max_length,
              weights=[embedding_matrix], input_length=max_length, trainable=False)
model.add(e)
model.add(Bidirectional(LSTM(256), merge_mode='sum'))
model.add(Dropout(0.50))
model.add(Dense(3, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(train_dataset[0], train_dataset[1], epochs=32, batch_size=256, verbose=1,
          validation_data=(validation_dataset[0], validation_dataset[1]))

score = model.evaluate(test_dataset[0], test_dataset[1], verbose=1)

print('loss : {}, acc : {}'.format(score[0], score[1]))

Epoch 1/32
54/54 [==============================] - 1515s 27s/step - loss: 0.3057 - accuracy: 0.9151 - val_loss: 0.2714 - val_accuracy: 0.
9280
Epoch 2/32
54/54 [==============================] - 933s 17s/step - loss: 0.2532 - accuracy: 0.9304 - val_loss: 0.2703 - val_accuracy: 0.9
286
Epoch 3/32
54/54 [==============================] - 1051s 19s/step - loss: 0.2735 - accuracy: 0.9301 - val_loss: 0.2623 - val_accuracy: 0.
9280
Epoch 4/32
54/54 [==============================] - 1153s 21s/step - loss: 0.2343 - accuracy: 0.9317 - val_loss: 0.2574 - val_accuracy: 0.
9291
Epoch 5/32
54/54 [==============================] - 1148s 21s/step - loss: 0.2112 - accuracy: 0.9318 - val_loss: 0.2113 - val_accuracy: 0.
9295
Epoch 6/32
54/54 [==============================] - 1133s 21s/step - loss: 0.1990 - accuracy: 0.9321 - val_loss: 0.2069 - val_accuracy: 0.
9308
Epoch 7/32
54/54 [==============================] - 1140s 21s/step - loss: 0.1922 - accuracy: 0.9349 - val_loss: 0.2410 - val_accuracy: 0.
9302
Epoch 8/32
54/54 [==============================] - 1139s 21s/step - loss: 0.1797 - accuracy: 0.9344 - val_loss: 0.2005 - val_accuracy: 0.
9315
Epoch 9/32
```

Figure 4.31: Construction, compilation, training and evaluation code

Figure 4.31 shows 4 steps of code to train and validate the datasets. ‘model.add’ shows the construction code such as add the Bidirectional and LSTM layer. The, the code that starts with ‘model.compile’ shows the compilation process which is to to configure the learning process of the model. Next, the training code starts with ‘model.fit’ code then followed by the code after that. It shows the code function is used to train the model on the training dataset. Lastly, model evaluation code starts with ‘model.evaluate’ and followed by the next code after that to evaluate the model's performance on the test dataset.

```
In [75]: print(report)
```

	precision	recall	f1-score	support
(0) Negative class	0.45	0.42	0.44	284
(1) Neutral class	0.38	0.44	0.41	25
(2) Positive class	0.96	0.96	0.96	4217
accuracy			0.93	4526
macro avg	0.60	0.61	0.60	4526
weighted avg	0.93	0.93	0.93	4526

Figure 4.32: Results for sentiment classification

Figure 4.32 shows the result from sentiment classification. The results shows that the findings are biased classification towards the class with higher frequency distribution. Supported by confusion matrix in figure 4.33. The study from this report demonstrates that model had relatively weaker predictive performance for the negative and neutral sentiments. The results supporting the claims that BRNN-LSTM captures better context from the review texts that leads to good predictive performance. Thus, for the fair comparison, BRNN cannot satisfied that elements and the suggested model is uni-directional RNN-LSTM.

```
In [76]: conf_matrix = confusion_matrix(np.argmax(test_dataset[1], axis=1), test_predictions)
         print(conf_matrix)

         [[ 120    5  159]
          [   8   11    6]
          [ 136   13 4068]]

In [77]: plt.figure(figsize=(8, 8))
         plt.savefig('conf_matrix_sentiment.png', format='png', dpi=300)
         sns.heatmap(conf_matrix, annot=True, annot_kws={'size': 16}, cmap='coolwarm', fmt='.2f')

Out[77]: <AxesSubplot:>
```

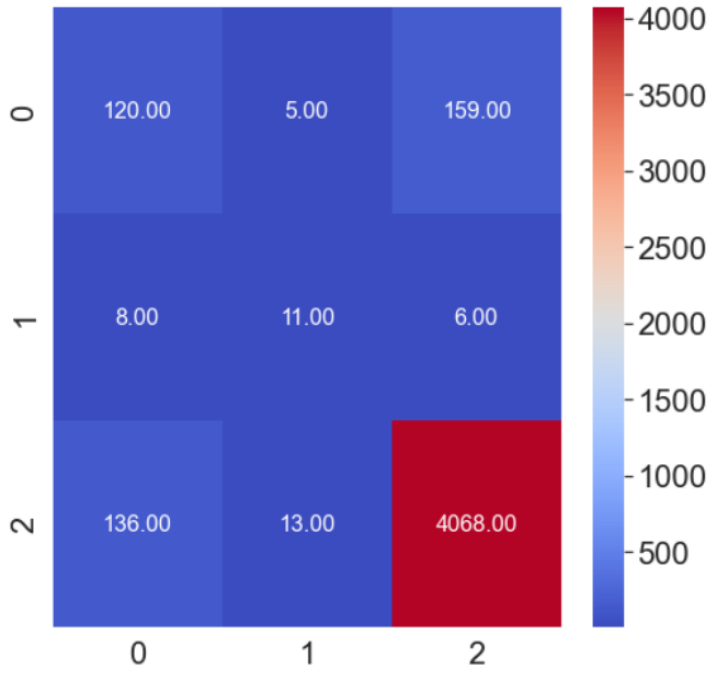


Figure 4.33: Confusion matrix for Sentiment Classification

CHAPTER 5

CONCLUSION

5.1 Introduction

As the conclusion from the full evaluation process and documentation before starting the code, mostly all of the research objectives can be achieved. Based on the introduction for the research, the problem statement can be solved after executing the results. Then, the comparison between the three previous works is really helpful to boost the idea for this research. All the execution, training, testing and validation phases are well done.

From the methodology proposed, many elements such as framework, and requirements are successful to be followed. Only the execution results for the Recommendation class failed to be executed for limitation and constraint reasons. The datasets used were good and easy to make progress such as text cleaning, analysis and sentiment classification.

For future enhancement, more than one models need to be trained to get an accurate and fair comparison between the three types of sentiment classification. BRNN-LSTM is a good approach to capture the context from the text reviews that leads to a good predictive performance but it only has the better prediction for the class that has higher frequency distribution.

5.2 Research Constraint

From the executed results and proposed work, there is one of the results that is incapable of running it. The results for the Recommendation IND class have ‘1’ and ‘0’ values. From the observation, the code cannot be run because of time and knowledge limitations. All the knowledge that I know is already implemented to make the code work

but it still displays several errors. Thus, the technology constraint is also one of the factors that the code cannot be run. For the sentiment classification output, the laptop takes about 10 hours to completely run the code with only 32 epochs.

5.3 Future works

In future works focused on sentiment analysis research, utilizing BRNN-LSTM as the technique holds significant potential. Building upon the strengths of bidirectional recurrent neural networks (BRNNs) and long short-term memory (LSTM) units, this approach offers a robust framework for capturing contextual information in textual data. To advance this field, future studies could explore various avenues, such as incorporating attention mechanisms to enhance the model's ability to identify important features, experimenting with different architectural variations of BRNN-LSTM, integrating external knowledge sources or pre-trained word embeddings, and investigating transfer learning techniques to adapt the model to different domains and languages. Additionally, exploring ensemble methods and combining BRNN-LSTM with other deep learning architectures, such as convolutional neural networks (CNNs), could further enhance sentiment analysis performance and generalize its applications to real-world scenarios.

REFERENCES

9 Major Advantages of Ecommerce to Businesses in 2022 | Seller Blog. (n.d.). Retrieved December 5, 2022, from https://sell.amazon.in/seller-blog/advantages-of-ecommerce

Online Shopping Statistics, Facts & Trends in 2022. (n.d.). Retrieved December 5, 2022, from https://www.cloudwards.net/online-shopping-statistics/

Top 5 Benefits of Sentiment Analysis for Businesses. (n.d.). Retrieved December 5, 2022, from https://research.aimultiple.com/sentiment-analysis-benefits/

Azhaguramyaa, V. R., Janet, J., Madhavan, G. R., Balakrishnan, S., & Arunkumar, K. (2022). Sentiment Analysis on Book Reviews Using Machine Learning Techniques. In *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022*. Springer Singapore. https://doi.org/10.1109/ICACCS54159.2022.9785311

Conceptual and Theoretical Frameworks for Thesis Studies: What You Must Know. (n.d.). Retrieved January 18, 2023, from https://www.enago.com/thesis-editing/blog/conceptual-and-theoretical-frameworks-for-thesis-studies-what-you-must-know

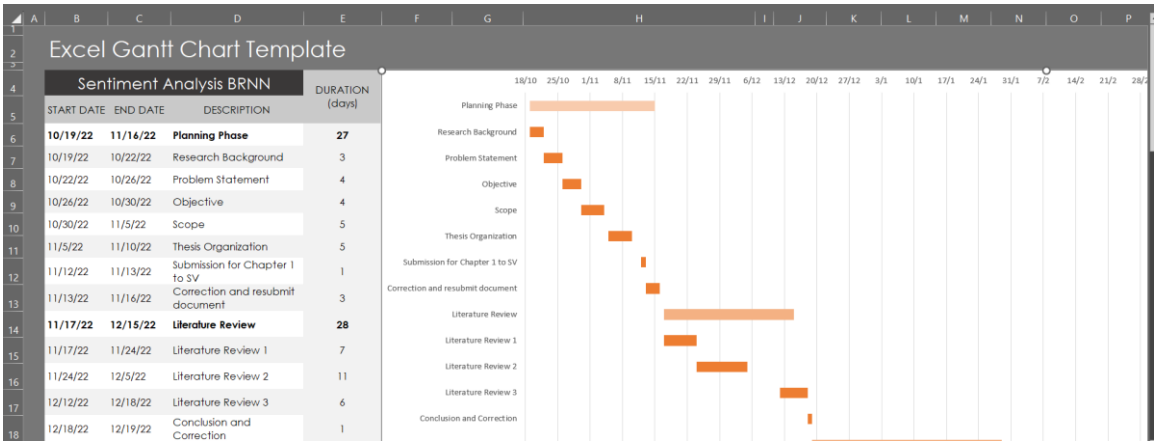
Elbagir, S., & Yang, J. (2018). Sentiment analysis of twitter data using machine learning techniques and scikit-learn. *ACM International Conference Proceeding Series*, 0–5. https://doi.org/10.1145/3302425.3302492

Introduction to Recurrent Neural Network - GeeksforGeeks. (n.d.). Retrieved December 29, 2022, from https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

Mitra, A. (2020). Sentiment Analysis Using Machine Learning Approaches (Lexicon based on movie review dataset). *Journal of Ubiquitous Computing and Communication Technologies*, *2*(3), 145–152. https://doi.org/10.36548/jucct.2020.3.004

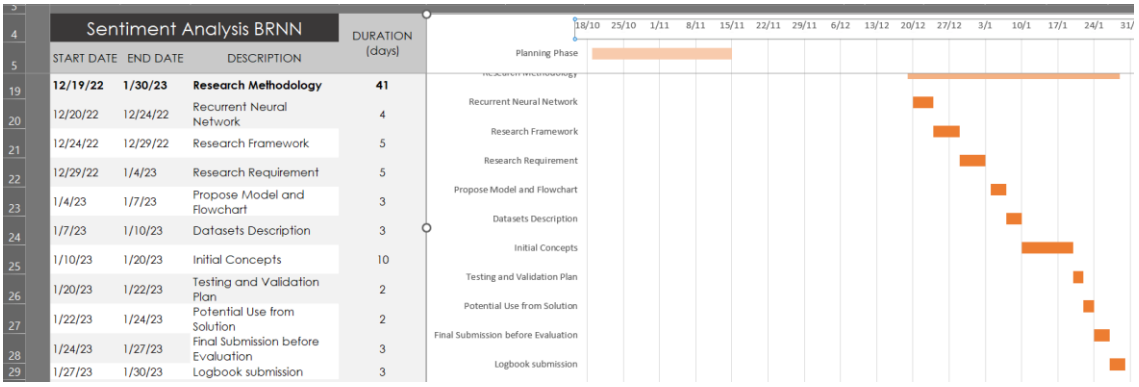
Text Cleaning for NLP: A Tutorial. (n.d.). Retrieved December 29, 2022, from https://monkeylearn.com/blog/text-cleaning/

APPENDIX A
GANTT CHART

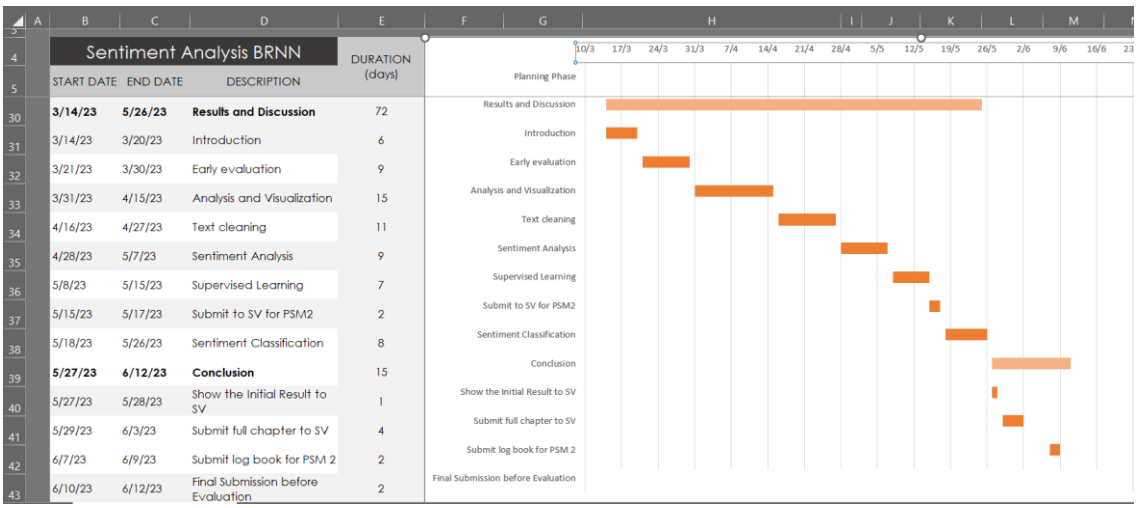


Excel Gantt Chart Template

Sentiment Analysis BRNN			DURATION (days)
START DATE	END DATE	DESCRIPTION	
10/19/22	**11/16/22**	**Planning Phase**	**27**
10/19/22	10/22/22	Research Background	3
10/22/22	10/26/22	Problem Statement	4
10/26/22	10/30/22	Objective	4
10/30/22	11/5/22	Scope	5
11/5/22	11/10/22	Thesis Organization	5
11/12/22	11/13/22	Submission for Chapter 1 to SV	1
11/13/22	11/16/22	Correction and resubmit document	3
11/17/22	**12/15/22**	**Literature Review**	**28**
11/17/22	11/24/22	Literature Review 1	7
11/24/22	12/5/22	Literature Review 2	11
12/12/22	12/18/22	Literature Review 3	6
12/18/22	12/19/22	Conclusion and Correction	1



Sentiment Analysis BRNN			DURATION (days)
START DATE	END DATE	DESCRIPTION	
12/19/22	**1/30/23**	**Research Methodology**	**41**
12/20/22	12/24/22	Recurrent Neural Network	4
12/24/22	12/29/22	Research Framework	5
12/29/22	1/4/23	Research Requirement	5
1/4/23	1/7/23	Propose Model and Flowchart	3
1/7/23	1/10/23	Datasets Description	3
1/10/23	1/20/23	Initial Concepts	10
1/20/23	1/22/23	Testing and Validation Plan	2
1/22/23	1/24/23	Potential Use from Solution	2
1/24/23	1/27/23	Final Submission before Evaluation	3
1/27/23	1/30/23	Logbook submission	3



Sentiment Analysis BRNN			DURATION (days)
START DATE	END DATE	DESCRIPTION	
3/14/23	**5/26/23**	**Results and Discussion**	**72**
3/14/23	3/20/23	Introduction	6
3/21/23	3/30/23	Early evaluation	9
3/31/23	4/15/23	Analysis and Visualization	15
4/16/23	4/27/23	Text cleaning	11
4/28/23	5/7/23	Sentiment Analysis	9
5/8/23	5/15/23	Supervised Learning	7
5/15/23	5/17/23	Submit to SV for PSM2	2
5/18/23	5/26/23	Sentiment Classification	8
5/27/23	**6/12/23**	**Conclusion**	**15**
5/27/23	5/28/23	Show the Initial Result to SV	1
5/29/23	6/3/23	Submit full chapter to SV	4
6/7/23	6/9/23	Submit log book for PSM 2	2
6/10/23	6/12/23	Final Submission before Evaluation	2