University of Hertfordshire UH

School of Physics,
Engineering and
Computer Science

MSc Data Science Project
7PAM2002-0509-2022
Department of Physics, Astronomy and Mathematics

Data Science Final Project Report

Project Title:

Determining Customer Sentiment in Fashion Product Reviews Using Lexicon-based Approach and Machine Learning Techniques.

Student Name and SRN:

Taiwo Olufemi Ayomide 21066565

Supervisor: Dr. William Alston

Date Submitted: 31/08/2023.

Word Count: 8488

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Taiwo Olufemi Ayomide

Student Name signature:

Student SRN number: 21066565

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

ACKNOWLEDGMENT

I want to express my sincere gratitude to everyone who contributed to successfully completing my final year thesis during my postgraduate studies. My most profound thanks are extended to Dr. Will Alston, my mentor, and my advisers for their essential advice, help, and knowledge. I owe a debt of gratitude to my friends and family for their unflagging support and patience during this process. An additional thank you to Yusuf Saka, whose insightful comments and mentoring helped shape this project's course. Without my loved ones' constant encouragement and support, this accomplishment would not have been possible. I appreciate your support of me and my endeavor.

ABSTRACT

Nowadays, E-commerce websites use online reviews to know what and how customers feel about their products. These reviews are essential to the growth and improvement of these companies. The fashion industry thrives on the feedback and perception of the user. Hence, the need to know the user's feelings and opinions after using their products. Sentiment analysis evaluates people's opinions from the reviews and helps the business make decisions based on its results. The analysis process involves natural language processing, text analysis, and opinion classification.

The sentiment analysis in this project using a fashion review dataset was done using the lexicon-based approach and machine learning algorithms. The lexicon-based approach classified the sentiments as negative, neutral, and positive. The classification was done using the polarity score of reviews, and the NLP techniques used are TextBlob and VADER sentiment.

The machine learning algorithms used are Logistic Regression, SVM, and AdaBoost. The logistic regression model and AdaBoost showed higher accuracy and precision than the SVM model.

Table of Contents

1 INTRODUCTION

1.1 BACKGROUND

Sentiment Analysis, also known as opinion mining, studies people's opinions about products or services expressed through spoken or written communication. Online reviews are crucial in the e-commerce industry, particularly fashion, where businesses use sentiment analysis to determine customer opinions and feelings. According to a survey by Wankhade et al. (2022), sentiment analysis is an effective tool for automating customer sentiment in online reviews. The information gathered from customer reviews can be processed using sentiment analysis to provide valuable insights for organisational growth (Jain, Pamula, & Srivastava, 2021).

Three categorization levels are involved in sentiment analysis: sentiment analysis at the document, phrase, and aspect levels. Every sentence in a document communicates the feeling. the sentence's objectivity or subjectivity must be determined in the first phase. The aspect-level sentiment analysis classifies the sentiment using the specific aspect of the entity. (Medhat, Hassan and Korashy, 2014a) (Sentiment Analysis and Opinion Mining - Bing Liu - Google Books, 2022)

Lexicon-based approaches like TextBlob and the Vader Sentiment NLP algorithms can perform sentiment analysis. These methods determine the polarity score of a sentence or document to identify the sentiment type contained. Moreover, traditional supervised machine-learning algorithms can also specify1 the sentiment linked to an entity. These algorithms classify reviews based on user recommendations of a product. Machine learning approaches use a training set to develop a sentiment classifier that classifies sentiment.

1.2 PROBLEM STATEMENT

Customer reviews are vital for fashion brands to grow and improve. Businesses make informed decisions based on user feedback to enhance their products. However, issues can arise during the review analysis process. There may be numerous reviews to sift through, as well as emotional biases on the part of humans, and other factors that can impede accurate sentiment prediction in studies. This is a problem for the e-commerce store because when they cannot determine the intention and meaning the user is trying to communicate, thus do not fulfill the need for the online review system.

Using machine learning alongside lexicon-based NLP techniques to help determine the sentiment in customer reviews will help create a robust hybrid system.

1.3 AIMS AND OBJECTIVES

This research aims to determine customer sentiment in fashion product reviews using a lexicon-based approach and machine learning techniques. This study aims to empower fashion businesses to make data-driven decisions, improve product offerings, and enhance customer satisfaction by developing an accurate and domain-specific sentiment analysis model.

The research objectives are as follows:

- To determine the sentiment types that appear most frequently in evaluations of clothing.
- To assist companies in finding opportunities for development and assisting them in making decisions based on customer feedback.

- To assist companies in identifying consumer groups that have favorable reviews and using information for focused marketing initiatives.
- To create fresh machine learning techniques for categorising reviews of fashion merchandise.
- To publish the study's findings in a journal that has undergone peer review.

1.4 JUSTIFICATION OF THE STUDY

This research utilises a combined methodology of lexicon-based analysis and advanced machine learning algorithms to analyse customer sentiment in fashion product reviews. These reviews significantly impact purchasing decisions and brand reputation in the fashion industry. By comprehending the emotions conveyed in these reviews, fashion firms can make informed decisions, improve customer satisfaction, and align their offerings with market expectations. The study aims to provide contextually aware and accurate sentiment analysis through developing a domain-specific lexicon and applying state-of-the-art machine-learning models. (Shivaprasad and Shetty, 2017). The findings of this research will contribute to the field of sentiment analysis and assist clothing retailers in making well-informed decisions.

1.5 STRUCTURE OF THE REPORT

Chapter one covers the project introduction, background, aims, objectives, research, and ethical, social, and legal considerations. Chapter two defines sentiment analysis, machine learning and its classifications, and related research. It also covers major terms and techniques used in this project. The research methodology and the approach to be used for the experiment are covered in Chapter three. Chapter four will provide an in-depth analysis of the model results and compares them with earlier work. Chapter five evaluates results, implications, and ethical considerations. Finally, Chapter six talks about recommendations and conclusion that must be considered.

1.6 ETHICAL, SOCIAL, AND LEGAL CONSIDERATIONS FOR THE PROJECT

This research does not require ethical approval because the dataset used is obtained from an online data repository, and doesn’t contain any personal, individual, or corporate information. In addition, there isn’t any social consideration for this project as it doesn’t contain any brand information in the dataset. Also, there isn’t any legal consideration for this project.

2 LITERATURE REVIEW

2.1 NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) that focuses on the interaction between computer and human language like speech and text. It is concerned with enabling computers with the ability to understand text and spoken words the way a human can – in a way that is both meaningful and valuable. (Chowdhary, 2020) In essence, NLP combines computational logistics with machine learning and deep learning models to enable computers to process human language and to understand its full meaning, complete with the writer's or speaker's intent and sentiment. (Christopher and Hinrich 1999). NLP has a wide range of uses in a variety of industries, including education, healthcare, fashion industry and finance. Deep learning models have been at the forefront of recent developments in natural language processing, achieving cutting-edge outcomes in a variety of NLP tasks, such as sentiment analysis and machine translation. For instance, a study by Zhang et al. (2018) found that deep learning models have significantly improved the accuracy of sentiment analysis, a task that involves identifying the sentiment or emotion expressed in each text. Similarly, another study by Vaswani et al. (2018) demonstrated the effectiveness of deep learning models in machine translation, which involves translating text from one language to another.

2.2 SENTIMENT ANALYSIS

Sentiment analysis is an NLP technique that extracts subjective information from a document related to an entity and helps businesses understand social sentiment. (Liu 2022) In simple terms, sentiment analysis involves identifying positive, negative, or neutral connotations in text to determine emotional tone. (Mohammed 2016) The analysis can provide valuable insights into how people feel about various subjects, including people, events, topics, and brands, allowing businesses, individuals, and organizations to make more informed decisions.

Sentiment analysis has been widely recognised as a valuable tool for businesses looking to improve their customer experience. It is used in various fields, such as politics, business, and market research. It helps understand customer opinions, identify trending topics, gauge public reactions to events, and assess public sentiment towards social issues.

A study by Harvard Business Review (2007) found that companies that actively manage customer experience have higher customer satisfaction rates and increased revenue.

Sentiment analysis can also be useful for fashion brands looking to improve their customer experience. By analysing customer reviews and social media posts, fashion brands can gain insights into the emotions and opinions expressed about their products and brand (Lang and Zhaobrands, 2020). This can help them identify areas of improvement, track trends, and make data-driven decisions to enhance the overall customer experience. With sentiment analysis, fashion brands can ensure they meet customer expectations and stay ahead of the competition. As NLP technology advances, sentiment analysis will play a significant role in understanding human emotions and opinions.

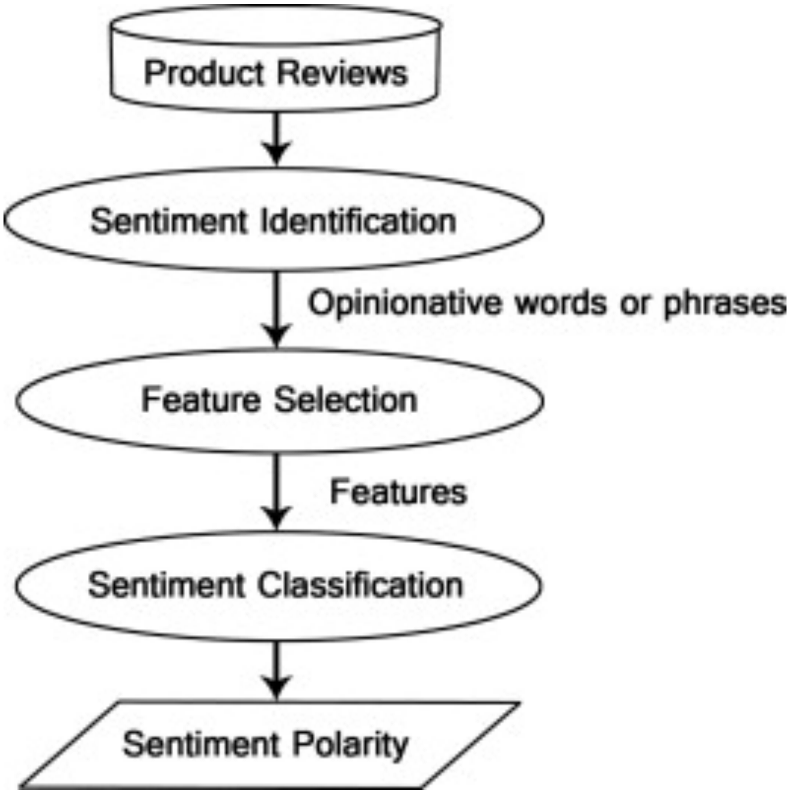


Fig 2.0 Steps of sentiment analysis in product reviews(Medhat, Hassan and Korashy, 2014b)

TYPES OF SENTIMENT ANALYSIS

Emotions can be represented in various ways, with positive, negative, or neutral being the most common. Different types of sentiment analysis can capture, meet consumer needs, and categorise emotions, with previous research focusing mostly on these categories.:

i. **Document-Level Sentiment Analysis:** This is the most basic form of sentiment analysis, where the overall sentiment of an entire document (such as a review, article, or tweet) is classified as positive, negative, or neutral. (Behdenna, Barigou, and Belalem, 2018).

ii. **Sentence-Level Sentiment Analysis:** In this type, sentiment analysis is performed in a document sentence-by-sentence basis. Each sentence is classified as positive, negative, or neutral, providing a more fine-grained analysis. (Bongirwar, 2015).

iii. **Entity-Level Sentiment Analysis:** Here, the focus is on specific entities within the text, such as people, products, or places. The sentiment towards each entity is analysed, allowing for a deeper understanding of opinions.

iv. **Aspect-Based Sentiment Analysis:** This analysis goes beyond entities and focuses on specific aspects or features. Its purpose is to determine the sentiment associated with each aspect. For instance, idea is analyzed separately for attributes such as design, performance, and customer services in a product review.

LEXICON-BASED SENTIMENT ANALYSIS

Lexicon-based sentiment analysis is a technique that uses predefined lexicons or dictionaries to classify sentiments based on associated words' sentiment scores. This approach is helpful in dealing with specialised text data with distinct sentiment implications, such as fashion product reviews (Kumaresh *et al.*, 2019). The lexicon used in this sentiment analysis contains a vast list of words, phrases, and expressions, each annotated with sentiment scores that capture the polarity of the associated word. (Taboada *et al.*, 2011).

During the analysis, the text is broken down into smaller units, such as individual words or sentences. The lexicon is then used to look up each word's sentiment score. The sentiment scores of all the words in each text are aggregated to determine the overall sentiment of the text. The final sentiment classification is often based on a threshold value, where scores above a certain threshold are classified as positive, scores below another threshold as unfavourable, and scores in between as neutral (Khan *et al.*, 2014).

2.3 MACHINE LEARNING

A branch of artificial intelligence known as machine learning focuses on creating computer programmes that automatically get more efficient at what they do as they gain more experience. The goal of machine learning is more application systems with more robust learning capabilities (Jordan and Mitchell, 2015).Machine learning involves building systems that learn independently and can make informed decisions based on what they have learned. A domain-independent enabling technology for a wide variety of computer applications is what machine learning research aims to create. Robotics, intelligent databases, computer-aided design, sentiment analysis, and knowledge-based consultant systems are just a few of the computer-related fields where an enormous advancement in machine learning could have a big influence.

Machine learning models can be used to predict the sentiment contained in a document or sentence. Machine learning-based sentiment analysis is an advanced approach that uses the power of artificial intelligence (AI) and statistical modelling to classify sentiments in text data automatically. Unlike lexicon-based methods, machine learning techniques do not rely on predefined sentiment scores, and text dictionaries but instead learn from vast amounts of labelled data to identify patterns and relationships between words and sentiments. (Hasan *et al.*, 2018)

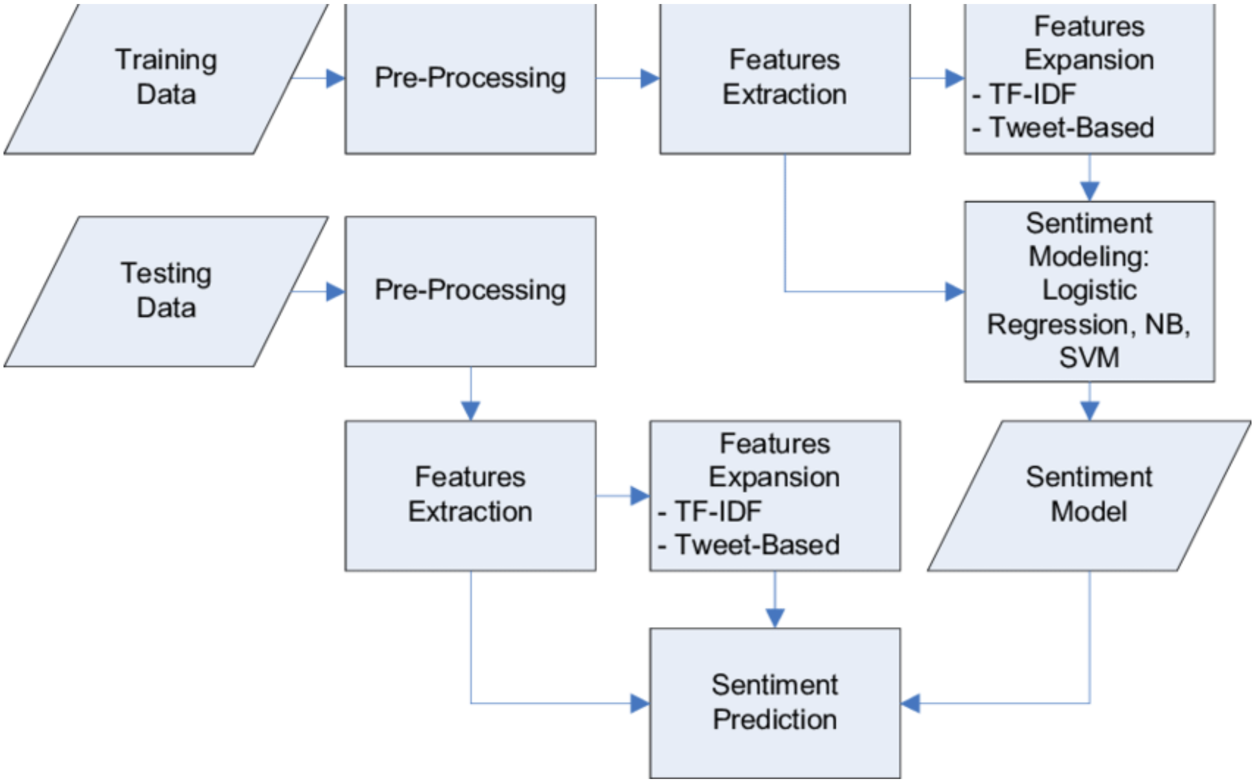


Fig 2.1 Sentiment-Analysis-Model-machine-learning-classifier-for-polarity-detection in sentences (Setiawan, Widyantoro and Surendro, 2018)

The process of machine learning-based sentiment analysis begins with a labeled dataset, where human annotators have assigned sentiments (positive, negative, or neutral) to various text samples. This labelled data serves as the training set for the machine learning model. The model then undergoes a training process where it iteratively adjusts its internal parameters to minimize errors and optimize sentiment predictions.

When the model has been trained, it can be applied to analyse new, unexplored text data. To prepare the text for analysis, it may involve tokenization, lemmatization, removing stop words, and other text-cleaning techniques. The model then applies its learned knowledge to classify the sentiment of the text accurately.

CLASSIFICATION OF MACHINE LEARNING

Machine learning algorithms are grouped into three categories: supervised, unsupervised, and reinforcement learning. A supervised machine learning algorithm learns from labelled training data to make predictions or decisions (Osisanwo *et al.*, 2017). Examples include support vector machines (SVM), Naïve Bayes, and Nearest Neighbour.

In contrast to supervised learning, unsupervised learning deals with unlabelled data. The algorithm's objective is to uncover the underlying structure or patterns in the data without any explicit guidance in the form of labelled examples. Instead, the algorithm seeks to identify similarities, clusters, or relationships within the data points (Alloghani *et al.*, 2020). Examples include clustering, K-means, and many more. Recommender systems and anomaly detection.

Finally, reinforcement learning describes machine learning algorithms that aim to train an agent to relate appropriately, considering its defined environment based on the correct rules that govern such an agent, the environment, and its behaviour in the background.

2.4 REVIEW OF RELATED WORKS

Nguyen *et al.* (2018) used a comparative study of text sentiment classification models using frequency-inverse document frequency vectorization in supervised machine learning and lexicon-based techniques. A combination of Lexicon-based and traditional supervised machine-learning algorithms will detect sentiment in customer feedback. Support Vector Machines (SVM), Gradient Boosting (GB), and Logistic Regression (LR) are the supervised machine learning algorithms. Likewise, three lexicon-based techniques are employed: the Pattern Lexicon, the SentiWordNet Lexicon, and the Valence Aware Dictionary and Sentiment Reasoner

(VADER). The study used data from the Amazon website, consisting of 43,620 product reviews from 1,000 products. The performance of the models was determined using a classification report, which gives the number of wrong and right predictions for each model. (Tasnim 2020). It also used other metrics like accuracy, precision, and F1 score. The findings demonstrate that the Logistics Regression, SVM, and Gradient Boosting models, respectively, have an F1-score of 94%, 92%, and 94%, and accuracy of 89%, 87%, and 90%; precision of 90%, 88%, and 91%; and recall of 98%, 98%, and 97%. Pattern, VADER, and SentiWordNet models had accuracy of 69%, 83%, and 80%, recall of 72%, 89%, and 88%, precision of 88%, 90%, and 90%, and F1-scores of 79%, 89%, and 88%, respectively. The supervised machine-learning models performed slightly better compared to the lexicon-based models.

The study by Yerpude *et al.* (2019) focused on using Natural Language Processing (NLP) and SentiWordNet to analyze the positive, negative, or neutral sentiment toward products. The research performed sentiment analysis on product reviews of electronic products and feature-based sentiment analysis on product reviews on mobile phones. It employs a lexicon-based methodology that tokenizes text to determine the sentiment of product reviews on an online marketplace.

In this study, sentiment analysis of customer reviews is carried out, and the results are presented as bar graphs and pie charts. The overall favourable and negative reviews of a product are depicted in the pie chart. It examined the polarity of each feature in each review of a particular product, and it generated a bar graph showing the total number of positive and negative comments of each item.

Rajeswari *et al. (*2020) used a hybrid approach combining lexicon-based and machine-learning techniques. It was claimed that the main flaw in existing systems was their exclusive concentration on classifying feedback as either positive or negative. A customer's opinion of a product or movie will be misinterpreted if the unbiased evaluation isn't taken into consideration, which will harm the industry or trend. The study uses machine learning algorithms like Support Vector Machine, Decision Tree, Logistic Regression, and Naive Bayes for sentiment analysis to resolve neutral opinions beyond the binary categorization of the customer's review. SentiWordNet is a lexicon-based technique. The authors concluded that the third-class neutral can be predicted using a hybrid strategy that uses the lexical approach to help solve the binary classification problem.

3 RESEARCH METHODOLOGY

The purpose of this chapter is to define and clarify the research methodology and current methods that can be used to address sentiment analysis in product reviews and how they can be utilised.

3.1 RESEARCH APPROACH

This project aims to use an inductive research approach as it involves qualitative data and making a generalisation based on observations. The study uses an existing review dataset and draws a conclusion from the analysis and training done on the dataset.

3.2 INDUCTIVE RESEARCH TECHNIQUE

Inductive research is a methodology used to find hidden patterns, develop fresh ideas, and form new theories or hypotheses. This form of research often begins with a question or a topic of interest, and then data is gathered using appropriate techniques. Patterns are looked for with the collected data, which can generate a hypothesis (Azungah, 2018; Hayes et al., 2010).

The inductive methodology includes the development of explanations and the lookout for patterns in observations. The researcher can change the study's focus once the research procedure has begun because, for the inductive research approach, no ideas or notions would be considered at the beginning of the investigation.

3.3 DATASET USED IN THE RESEARCH

The dataset used for this research is obtained from Kaggle, an online dataset repository and machine learning platform. The dataset has 23,486 entries and 11 columns. Below is a description of the dataset and an explanation of each column in the dataset:

i. Clothing ID: Integer Categorical variable that refers to the specific piece being reviewed.

ii. Age: Positive Integer variable of the reviewer's age.

iii. Title: String variable for the title of the review.

iv. Review Text: String variable for the review body.

v. Rating: Positive Ordinal Integer variable for the product score granted by the customer from 1 Worst to 5 Best.

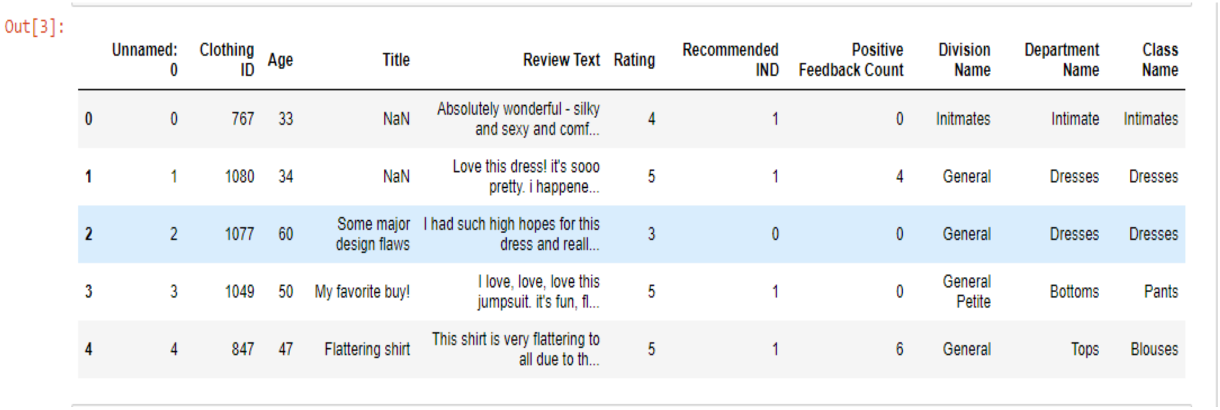
vi. Recommended IND: Binary variable stating where the customer recommends the product where 1 is recommended, and 0 is not recommended.

vii. Positive Feedback Count: Positive Integer documenting the number of other customers who found this review positive.

viii. Division Name: Categorical name of the product high-level division.

ix. Department Name: Categorical name of the product department name.

x. Class Name: Categorical name of the product class name.



Out[3]:

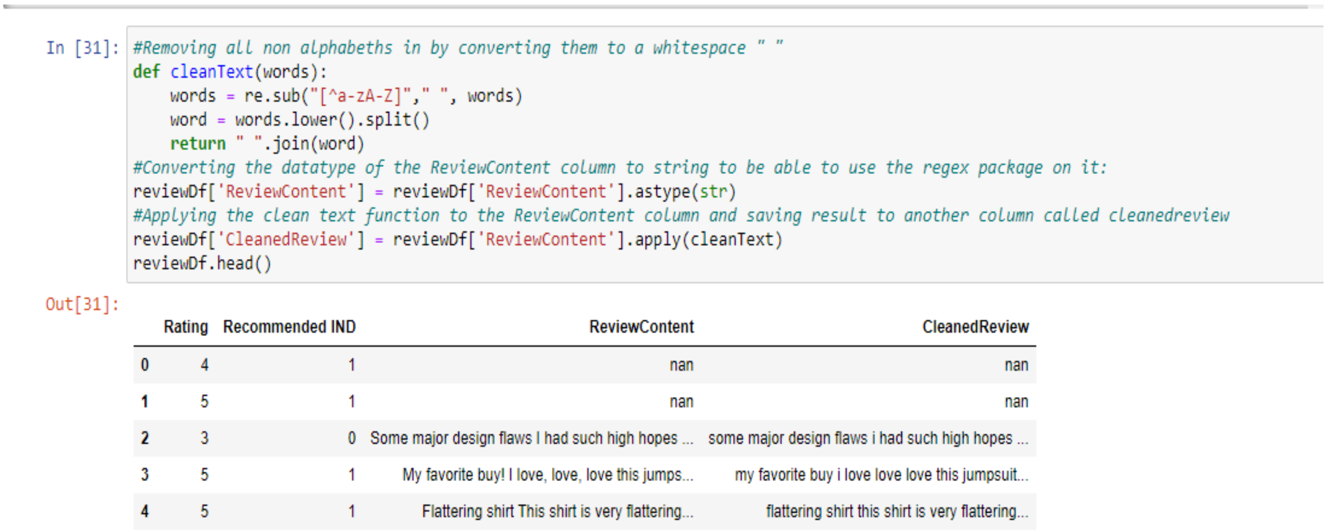
	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	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

Fig – 3.0 A data frame of the data created using pandas.

3.4 DATA CLEANING

Data cleaning entails identifying and correcting errors and irregularities in a dataset. (Rahm and Do 2000) It is a crucial phase of any data analysis or machine learning project. Raw data collected from online sources contains errors for many reasons, such as data entry and human errors (Chu *et al.*, 2016). These errors can lead to incorrect analysis and results if they are not corrected or removed. Data cleaning is a significant part of any good data analysis or machine learning project and should be noticed and addressed.

The data in this project needs to be cleaned and made suitable for the next stage. Entries with missing values or contents are removed from the dataset. Non-alphabets such as punctuation and numbers are also removed from the review column. The review column is also tokenized and lemmatized to break the words into their most basic form, which would aid a better understanding of the review content by the algorithms. This would speed up analysis, reduce distractions from the research goal, and make the algorithms more performant on the data.



In [31]:

```
#Removing all non alphabeths in by converting them to a whitespace " "
def cleanText(words):
    words = re.sub("[^a-zA-Z]"," ", words)
    word = words.lower().split()
    return " ".join(word)
#Converting the datatype of the ReviewContent column to string to be able to use the regex package on it:
reviewDf['ReviewContent'] = reviewDf['ReviewContent'].astype(str)
#Applying the clean text function to the ReviewContent column and saving result to another column called cleanedreview
reviewDf['CleanedReview'] = reviewDf['ReviewContent'].apply(cleanText)
reviewDf.head()
```

Out[31]:

	Rating	Recommended IND	ReviewContent	CleanedReview
0	4	1	nan	nan
1	5	1	nan	nan
2	3	0	Some major design flaws I had such high hopes ...	some major design flaws i had such high hopes ...
3	5	1	My favorite buy! I love, love, love this jumps...	my favorite buy i love love love this jumpsuit...
4	5	1	Flattering shirt This shirt is very flattering...	flattering shirt this shirt is very flattering...

3.5 DATA EXPLORATION AND ANALYSIS

We explore and study data sets using exploratory data analysis (EDA), which typically uses data visualisation techniques, and summarise their fundamental features. It makes finding patterns, identifying anomalies, testing hypotheses, or verifying assumptions simpler by determining how to modify data sources to achieve the answers needed (Morgenthaler, 2009; Komorowski *et al.*, 2016). We have done some EDA to show the relations and trends that exist between features of the dataset. Graphs and charts were generated during this process to explain the findings of the EDA.

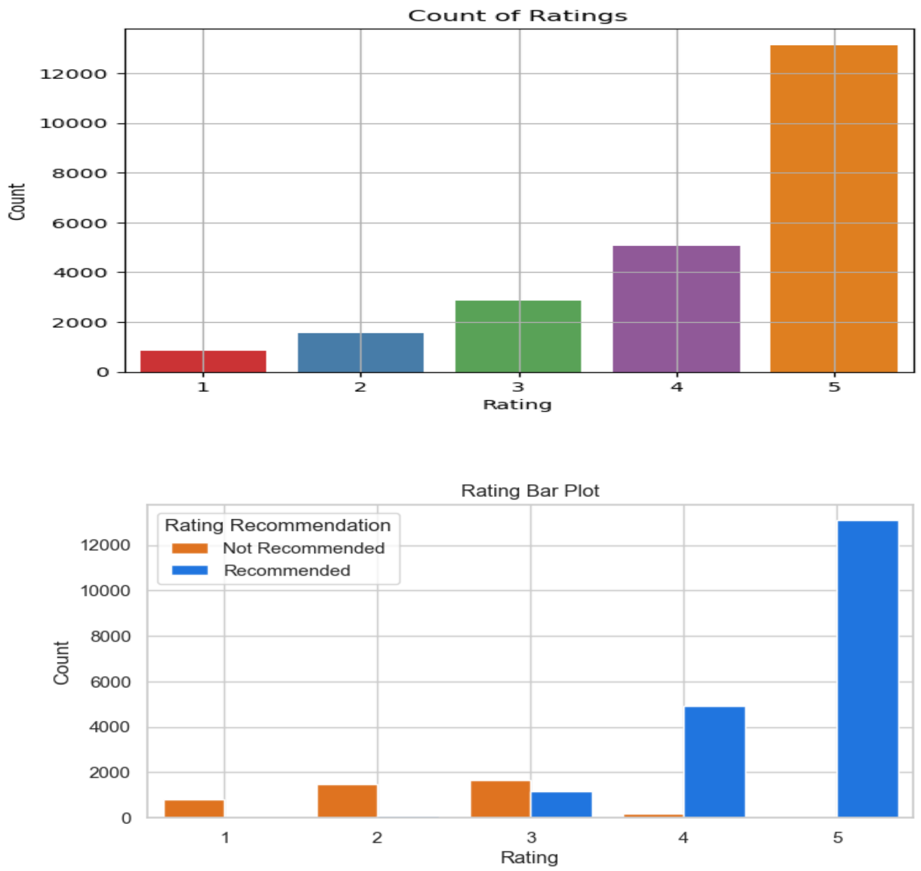


Fig – 3.1 CHARTS FROM EDA

The count rating chart (Fig 3.1) shows that products rated 1, and 2 (low) have low recommendations while those rated high (4,5) have high suggestions.

The rating bar plot is a plot of the rating given by the customer to each review. Each review is either recommended or not by the customer and the chart provides a visual depiction of how customers recommended the products based on reviews.

3.6 BUILDING THE MODEL

The programming language used in building the model is the Python programming language. The reason for adopting Python as the programming language is because it has libraries and frameworks to choose from, especially for sentiment analysis and machine learning.

The libraries that will be used in this project are TextBlob, Valence Aware Dictionary and Sentiment Reasoner (VADER), NumPy, Pandas, Seaborn, Matplotlib, Scikit-learn, and NLTK.

The data from the CSV file is read and preprocessed to clean it and make it suitable for analysis. NLTK techniques (TextBlob and VADER) are then applied to the cleaned data to determine their polarity score and classify them as positive, negative, or neutral.

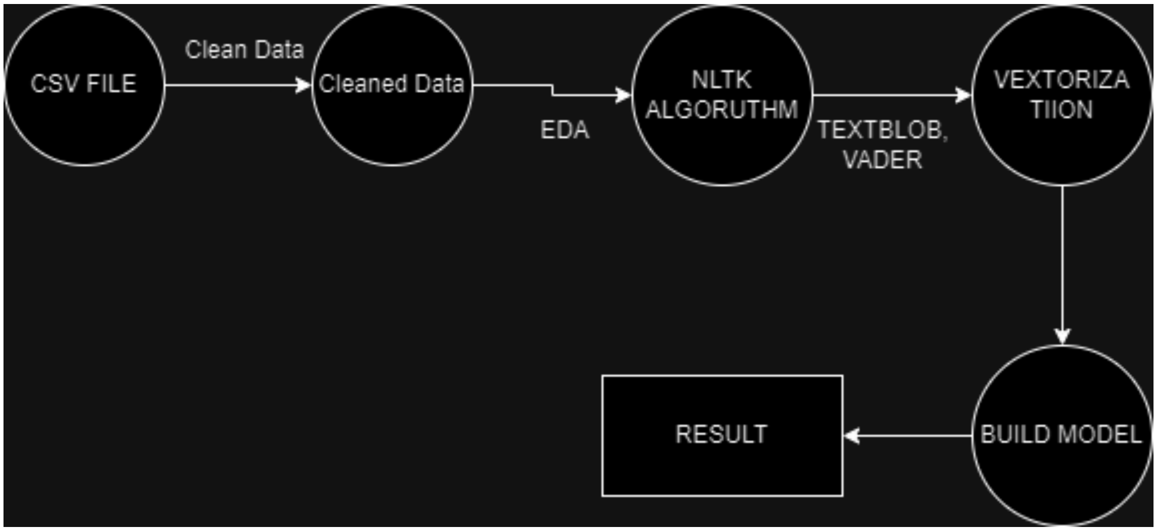


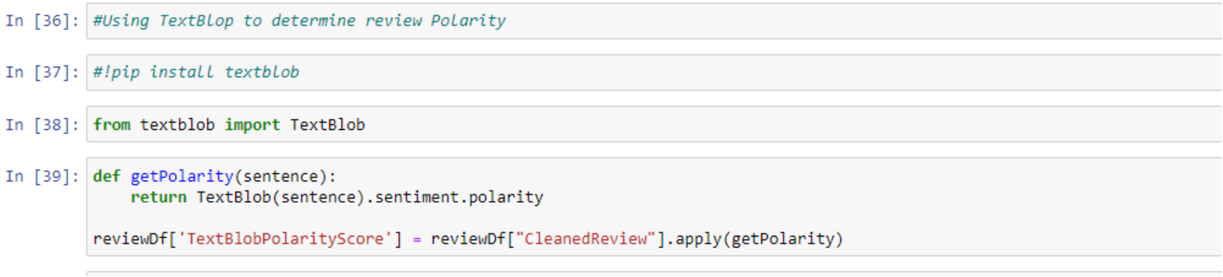
Fig – 3.2 System flow

3.7 NATURAL LANGUAGE TOOLKIT

Natural Language Toolkit (NLTK) is the most widely used Python library for writing programs that utilise human language data. NLTK offers a wide range of tools, algorithms, and resources for working with human language data which includes Tokenization, Part of Speech Tagging, Sentiment Analysis, Text Classification, Stemming and Lemmatization, and Syntax Parsing (Bird, 2006).

TEXTBLOB LIBRARY

TextBlob is an NLTK package, that offers a straightforward and intuitive API to perform NLP tasks such as Text Parsing, Sentiment Analysis, Text Classification, etc. It is simple and easy to use and when used to determine the polarity score of sentences or documents, gives an output between -1 and 1, where -1 denotes a negative sentiment or emotion, and 1 denotes a positive feeling or emotion.

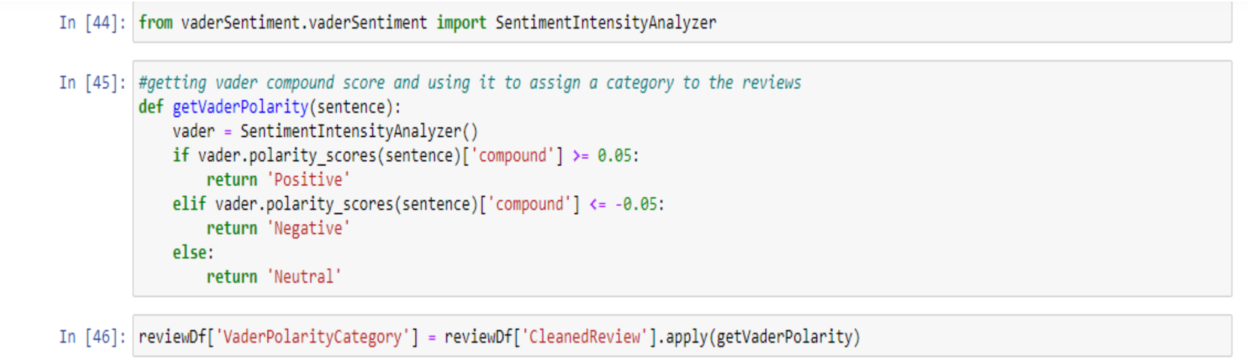


```
In [36]: #Using TextBlop to determine review Polarity

In [37]: #!pip install textblob

In [38]: from textblob import TextBlob

In [39]: def getPolarity(sentence):
             return TextBlob(sentence).sentiment.polarity

         reviewDf['TextBlobPolarityScore'] = reviewDf["CleanedReview"].apply(getPolarity)
```

Fig – 3.3 Using TextBlob to create polarity score for the reviews.

3.8 VALENCE AWARE DICTIONARY AND SENTIMENT REASONER (VADER) LIBRARY

VADER is a lexicon and rule-based sentiment analysis tool that analyses social media content and other short-form text. It focuses on capturing both the sentiment's polarity (negative, neutral, or positive) and the intensity of the view expressed in a text. Vader is designed to handle the ideas conveyed through emotions, emojis, and other special characters often used in online text. This is significant to our research as the reviews are gotten online and might contain emojis and special characters.



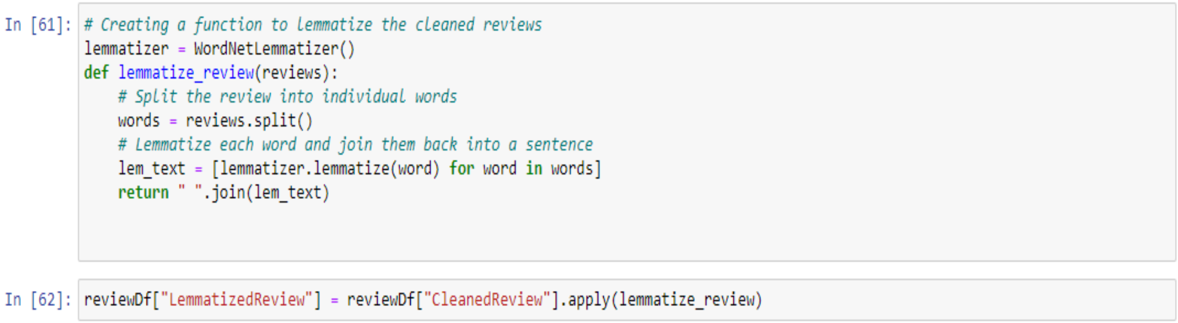
```
In [44]: from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

In [45]: #getting vader compound score and using it to assign a category to the reviews
         def getVaderPolarity(sentence):
             vader = SentimentIntensityAnalyzer()
             if vader.polarity_scores(sentence)['compound'] >= 0.05:
                 return 'Positive'
             elif vader.polarity_scores(sentence)['compound'] <= -0.05:
                 return 'Negative'
             else:
                 return 'Neutral'

In [46]: reviewDf['VaderPolarityCategory'] = reviewDf['CleanedReview'].apply(getVaderPolarity)
```

Fig – 3.4 Using VADER sentiment to create polarity score for the reviews.

3.9 LEMMATIZATION

Lemmatization is an NLP technique that reduces or transforms words into their base or root form. The objective is to transform inflected forms of an expression into a common base form so that they can be analyzed and processed as a single unit. This helps in sentiment analysis. Lemmatization is more advanced than stemming, as it takes the grammatical structure and the word class (part of speech) to produce the base form of the word (Balakrishnan & Lloyd-Yemoh, 2014).

When performing lemmatization, plural nouns are converted to their singular form, and inverted verbs are converted to their base form e.g., “walked” becomes “walk”, etc.



```
In [61]: # Creating a function to lemmatize the cleaned reviews
         lemmatizer = WordNetLemmatizer()
         def lemmatize_review(reviews):
             # Split the review into individual words
             words = reviews.split()
             # Lemmatize each word and join them back into a sentence
             lem_text = [lemmatizer.lemmatize(word) for word in words]
             return " ".join(lem_text)

In [62]: reviewDf["LemmatizedReview"] = reviewDf["CleanedReview"].apply(lemmatize_review)
```

Fig – 3.5 Lemmatizing each word in the reviews.

3.10 VECTORIZATION

Vectorization in machine learning is converting text-based or categorical data into numerical vectors that machine learning algorithms can understand. We cannot use textual data as input to machine learning algorithms, so we need to convert it to a numerical form that can be used as input to machine learning algorithms (Kozhevnikov & Pankratova, 2020). This is a fundamental step in preparing text and categorical data for the modelling stage, as most ML algorithms require numerical inputs alone.

There are different vectorization algorithms, but the ones used in this research are the count vectorizer and the TF-IDF vectorizer.

COUNT VECTORIZATION

Count Vectorization, or Bag of Words (BoW) representation, is a simple technique for vectorizing text data. It involves creating a matrix where each row represents a document, and each column represents a unique word from the entire text collection. The matrix contains the count of each word's occurrence in each document (Wendland et al., 2021).

TF-IDF VECTORIZATION

TF-IDF (Term Frequency-Inverse Document Frequency) Vectorization is an extension of Count Vectorization that considers the importance of words in a document relative to their occurrence in the entire text collection. It assigns higher weights to words frequent in a document but rare across the text collection, thus capturing the significance of words in a document (Wendland et al., 2021).

3.11 LOGISTIC REGRESSION

Logistic regression is a machine learning technique used to predict the likelihood of an outcome in a categorical dataset, usually binary, based on some input features (LaValley, 2008). In sentiment analysis, where we try to predict the sentiment related to some documents or entities, the outcome is usually a 0 or 1, where 0 represents negative sentiment, and 1 represents positive sentiment (Sharma & Tyagi, 2018). The input features used in the model are often derived from the text or document we want to perform sentiment analysis on through techniques like count vectorization (bag of words), TF-IDF (Term Frequency-Inverse Document Frequency), or word embeddings. These features quantify the importance of specific words or phrases in the text or document.

Once the model is trained, it can be used to classify new text samples into sentiment categories, and a threshold is chosen to determine the difference between negative and positive sentiments.

3.12 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) classifier is a machine learning technique that works by creating the best line that divides an n-dimensional space into classes. SVM aims to find a decision boundary that best separates the data points of different sentiment classes in a high-dimensional feature space. Regarding sentiment analysis, SVM creates a hyperplane that maximizes the margin between the data points of different sentiment classes (Zainuddin & Selamat, 2014). Once trained, the SVM model can then predict the sentiment of any new document introduced to it.

3.13 ADABOOST

AdaBoost is an ensemble learning algorithm that helps to improve the accuracy of classification algorithms. It creates a highly accurate prediction rule by combining many relatively weak and inaccurate rules. (Schapire, 2013).

Regarding sentiment analysis, it helps to enhance the performance of sentiment by handling complex sentiment patterns present in textual data. By combining the predictions of weak learners, AdaBoost can achieve high accuracy and improved generalization to unseen data.

4 ANALYSIS OF THE RESULT

This section of the project report shows the result of the procedures that have been used in the research.

4.1 MODEL RESULTS

TOKENIZATION

The review column in the dataset was cleaned by removing the special characters and non-alphabetical characters from the review columns. This cleaned review is then tokenized to break the texts into a sequence of smaller units called tokens. This is an essential part of sentiment analysis, and it helps the NLP algorithm to perform better.

Below is a table that shows the reviews before tokenization and after tokenization.

ORDINARY CLEANED REVIEWS	TOKENIZED REVIEW
major design flaws high hopes wanted work initially ordered petite small usual size found outrageously small small fact could zip reordered petite medium ok overall half comfortable fit nicely bottom half tight layer several somewhat cheap net layers imo major design flaw net layer sewn directly zipper c	['major', 'design', 'flaws', 'high', 'hopes', 'really', 'wanted', 'work', 'initially', 'ordered', 'petite', 'small', 'usual', 'size', 'found', 'outrageously', 'small', 'small', 'fact', 'could', 'zip', 'reordered', 'petite', 'medium', 'ok', 'overall', 'half', 'comfortable', 'fit', 'nicely', 'bottom', 'half', 'tight', 'layer', 'several', 'somewhat', 'cheap', 'net', 'layers', 'imo', 'major', 'design', 'flaw', 'net', 'layer', 'sewn', 'directly', 'zipper', 'c', '.']

favorite buy love love love jumpsuit fun flirty fabulous every time get nothing great compliments	[favorite, 'buy', 'love', 'love', 'love', 'jumpsuit', 'fun', 'flirty', 'fabulous', 'every', 'time', 'get', 'nothing', 'great', 'compliments']
flattering flattering due to adjustable front tie length leggings sleeveless pairs well cardigan love	['flattering', 'flattering', 'due', 'adjustable', 'front', 'tie', 'perfect', 'length', 'leggings', 'sleeveless', 'pairs', 'well', 'cardigan', 'love']
not for the very petite i love Tracy Reese dresses but this one is not for the very petite i am just under feet tall and usually wear a P in this brand This dress was very pretty out of the package but a lot of dresses the skirt is long and very full so it overwhelmed my small frame not a stranger to alterations shortening and narrowing the skirt would take away from the embellishment of the garment i love	['not', 'for', 'the', 'very', 'petite', 'i', 'love', 'tracy', 'reese', 'dresses', 'but', 'this', 'one', 'is', 'not', 'for', 'the', 'very', 'petite', 'i', 'am', 'just', 'under', 'feet', 'tall', 'and', 'usually', 'wear', 'a', 'p', 'in', 'this', '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', 'overwhelmed', 'my', 'small', 'frame', 'not', 'a', 'stranger', 'to', 'alterations', 'shortening', 'and', 'narrowing', 'the', 'skirt', 'would', 'take', 'away', 'from', 'the', 'embellishment', 'of', 'the', 'garment', 'i', 'love']
cagrcoal shimmer fun i aded this in my basket at hte last mintue to see what it would loo	['cagrcoal', 'shimmer', 'fun', 'i', 'aded', 'this', 'in', 'my', 'basket', 'at', 'hte', 'last', 'mintue', 'to',

k 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 prefectly 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 skirts i waas trying on of which i kept all oops	'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', 'prefectly', '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', 'skirts', 'i', 'waas', 'trying', 'on', 'of', 'which', 'i', 'kept', 'all', 'oops']

Table 4.1. The reviews before cleaning and after cleaning

4.2 LEXICON BASED MODELS

This is an analysis of the lexicon-based models and how they performed on the sentiment analysis of the fashion product reviews.

TEXTBLOB MODEL

TextBlob is used for sentiment analysis, and it provides the polarity and the subjectivity of the sentence. The polarity score is between – 1 and 1 where -1 denotes negativity, and 1 represents absolute positivity. The textblob algorithm is supplied with the content of the review and the algorithm returns a value between -1 and 1. This value is then used to determine the sentiment associated with the review.

Below is a table of some reviews and the textblob polarity score.

REVIEW CONTENT	TEXTBLOB POLARITY	SENTIMENT SCORE
some major design flaws 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 was just ok overall the top half was comfortable and fit nicely but the bottom half had a very tight under layer and several somewhat cheap net over layers imo a major design flaw was the net over layer sewn directly into the zipper it c	0.075813	Neutral

my favorite buy i love love love this jumpsuit it s fun flirty and fabulous every time i wear it i get nothing but great compliments	0.500000	Positive
flattering shirt 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	0.393750	Positive
not for the very petite i love Tracy Reese dresses but this one is not for the very petite i am just under feet tall and usually wear a P in this brand This dress was very pretty out of the package but a lot of dresses the skirt is long and very full so it overwhelmed my small frame not a stranger to alterations shorteni	0.181111	Neutral

ng and narrowing the skirt would take away from the embellishment of the garment i love the color and the idea of the style but it just did not work on me i returned this dress		
cagrcoal shimmer fun 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 darker color only because i am so pale hte color is really gorgeous and turns out it matched everything i was trying on with it perfectly 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 a	0.100417	Neutral

nd the skirts i waas trying on of which i kept all oops		

Table 4.2. First 5 results of the TextBlob Sentiment analysis

4.3 VALENCE AWARE DICTIONARY AND SENTIMENT REASONER (VADER) MODEL

Vader is a more sophisticated rule-based sentiment analysis algorithm designed to handle the sentiment conveyed through emotions, emojis, and other special characters often used in online text and in the case of this research, online reviews. It also generates polarity values between -1 and 1. Vader is used on the review content to determine the polarity and sentiment of each review content.

REVIEW CONTENT	TEXTBLOB POLARITY	SENTIMENT SCORE
some major design flaws 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 was just ok overall	0.9398	Positive

the top half was comfortable and fit nicely but the bottom half had a very tight under layer and several somewhat cheap net over layers imo a major design flaw was the net over layer sewn directly into the zipper it c		
my favorite buy i love love love this jumpsuit it s fun flirty and fabulous every time i wear it i get nothing but great compliments	0.6276	Positive
flattering shirt 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	0.9344	Positive
not for the very petite i love tracy reese dresses but this o	0.9431	Positive

ne is not for the very petite i am just under feet tall and usually wear a p in this 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 overwhelmed my small frame not a stranger to alterations shortening and narrowing the skirt would take away from the embellishment of the garment i love the color and the idea of the style but it just did not work on me i returned this dress		
cagrcoal shimmer fun 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 gorgeo	0.7425	Positive

us and turns out it mathced everythiing i was trying on with it prefectly 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 skirts i waas trying on of which i kept all oops		

Table 4.3. First 5 results of the Vander Sentiment analysis

From table 4.3, the Vader algorithm predicted the first 5 reviews as positive reviews.

4.4 REMOVING STOPWORDS

Stop words are set of commonly used words in a language. In this project, we need to remove the stop words for English language and the ones related to the fashion industry in general.

REVIEWS	REVIEWS AFTER REMOVING STOPWORDS
some major design flaws 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	major design flaws high hopes really wanted work initially ordered petite small usual size found outrageously fact could zip reordered petite medium ok overall half comfortable fit nicely bottom half tight layer several so

could not zip it up i reordered it in petite medium which was just ok overall the top half was comfortable and fit nicely but the bottom half had a very tight under layer and several somewhat cheap net over layers imo a major design flaw was the net over layer sewn directly into the zipper it c	mewhat cheap net layers imo major design flaw net layer sewn directly zipper c
my favorite buy i love love love this jumpsuit it s fun flirty and fabulous every time i wear it i get nothing but great compliments	favorite buy love love love jumpsuit fun flirty fabulous every time get nothing great compliments
flattering shirt 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	flattering flattering due adjustable front tie perfect length leggings sleeveless pairs well cardigan love
not for the very petite i love tracy reese dresses but this one is not for the very petite i am just under feet tall and usually wear a p in this 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 overwhelmed my small frame not a stranger to alterations shorte	etite feet tall usually p brand pretty package lot long full overwhelmed small frame stranger alterations shortening narrowing would take away embellishment garment love idea style work returned

ning and narrowing the skirt would take away from the embellishment of the garment i love the color and the idea of the style but it just did not work on me i returned this dress	
cagrcoal shimmer fun 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 prefectly 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 skirts i waas trying on of which i kept all oops	cagrcoal shimmer fun aded basket hte last mintue see would look like person store pick went teh darkler pale hte really gorgeous turns mathced everythiing trying prefectly little baggy hte xs hte msallet size bummer petite decided jkeep though said matvehd everything ejans pants skirts waas trying kept oops

Table 4.4. Removing stop words from the reviews

4.5 LEMMATIZATION OF THE REVIEWS

Lemmatization is an NLP technique that is used to reduce or transform words into their base or root form. We used the WordNetLemmatizer library to perform lemmatization on the review contents.

REVIEWS BEFORE LEMMATIZATION	REVIEWS AFTER LEMMATIZATION
major design flaws high hopes really wanted work initially ordered petite small usual size found outrageously small small fact could zip reordered petite medium ok overall half comfortable fit nicely bottom half tight layer several somewhat cheap net layers imo major design flaw net layer sewn directly zipper c	major design flaw high hope really wanted work initially ordered petite small usual size found outrageously small small fact could zip reordered petite medium ok overall half comfortable fit nicely bottom half tight layer several somewhat cheap net layer imo major design flaw net layer sewn directly zipper c
favorite buy love love love jumpsuit fun flirty fabulous every time get nothing great compliments	favorite buy love love love jumpsuit fun flirty fabulous every time get nothing great compliment
flattering flattering due adjustable front tie perfect length leggings sleeveless pairs well cardigan love	flattering flattering due adjustable front tie perfect length legging sleeveless pair well cardigan love

etite feet tall usually p brand pretty package lot long full overwhelmed small frame stranger alterations shortening narrowing would take away embellishment garment love idea style work returned	petite love tracy reese dress one petite foot tall usually p brand pretty package lot long full overwhelmed small frame stranger alteration shortening narrowing would take away embellishment garment love idea style work returned
cagrcoal shimmer fun aded basket hte last mintue see would look like person store pick went teh darkler pale hte really gorgeous turns mathced everythiing trying prefectly little baggy hte xs hte msallet size bummer petite decided jkeep though said matvehd everything ejans pants skirts waas trying kept oops	cagrcoal shimmer fun aded basket hte last mintue see would look like person store pick went teh darkler pale hte really gorgeous turn mathced everythiing trying prefectly little baggy hte x hte msallet size bummer petite decided jkeep though said matvehd everything ejans pant skirt waas trying kept oops

Table 4.5. Lemmatizing the reviews

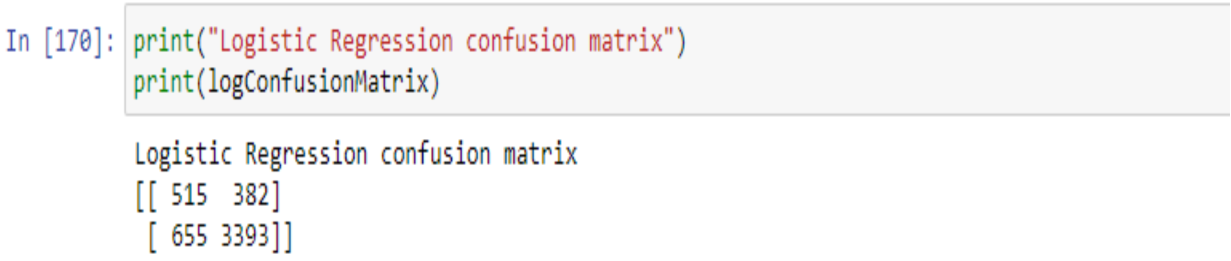
4.6 MACHINE LEARNING MODELS

Supervised machine learning models are employed in this project to help determine the sentiment associated with the customers' reviews. The machine learning algorithms adopted in this project are logistic regression, support vector machine, and AdaBoost.

The machine learning algorithms are trained with the review sparse matrix as the set of features, and the customer recommendation as the target variable.

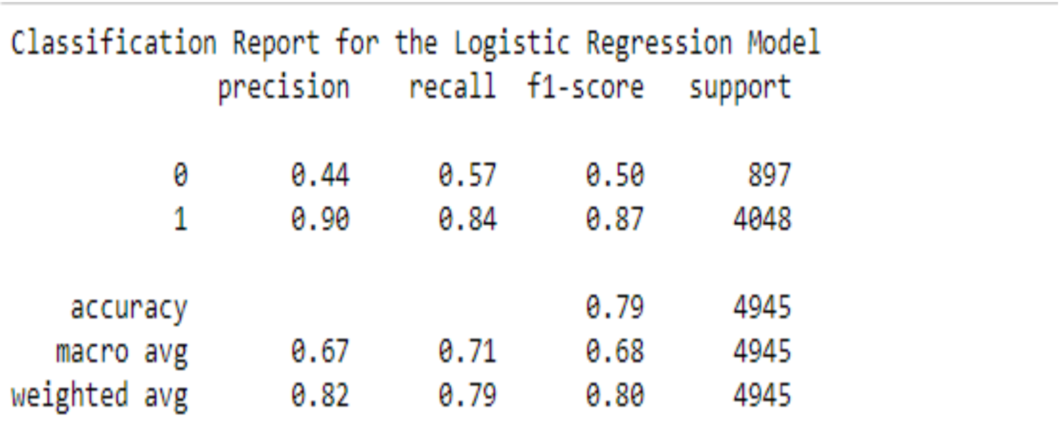
LOGISTIC REGRESSION MODEL

The already processed reviews which are now in the form of a sparse matrix are trained using the logistic regression algorithm. The logistic regression is trained using a class weight of balanced, and max iteration of 1000. The resulting model is then used to make predictions on the test dataset. The logistic regression prediction is then compared to the original customer recommendation to see how the model performs.



```
In [170]: print("Logistic Regression confusion matrix")
          print(logConfusionMatrix)

          Logistic Regression confusion matrix
          [[ 515  382]
           [ 655 3393]]
```

Fig. 4.0. Logistic Regression Confusion Matrix



```
Classification Report for the Logistic Regression Model
              precision    recall  f1-score   support

           0       0.44      0.57      0.50       897
           1       0.90      0.84      0.87      4048

    accuracy                           0.79      4945
   macro avg       0.67      0.71      0.68      4945
weighted avg       0.82      0.79      0.80      4945
```

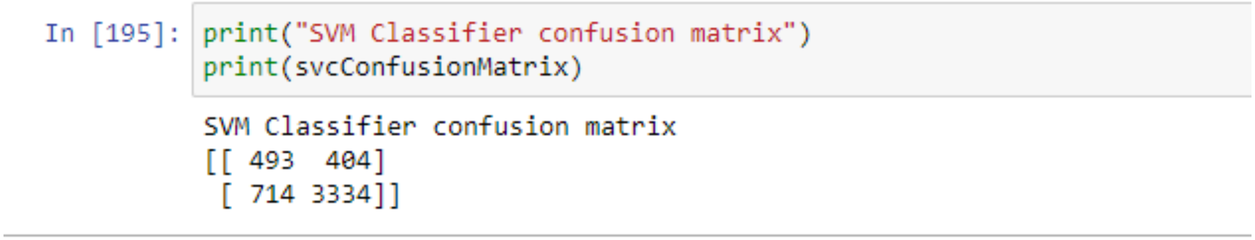
Fig. 4.1. Logistic Regression Classification Report

As seen in fig. 4.0. The logistic regression model has a true positive value of 3393 and a true negative value of 655. The false negative is 515, and the false positive is 382. This model seems to have performed better in classifying the positive reviews compared to classifying the negative reviews. In comparison to the other two models, the logistic regression model appears to have the most true positive results.

Also, from the classification report, the logistic regression model has a 79% accuracy, 82% precision, a recall of 79%, and an f1-score of 80%.

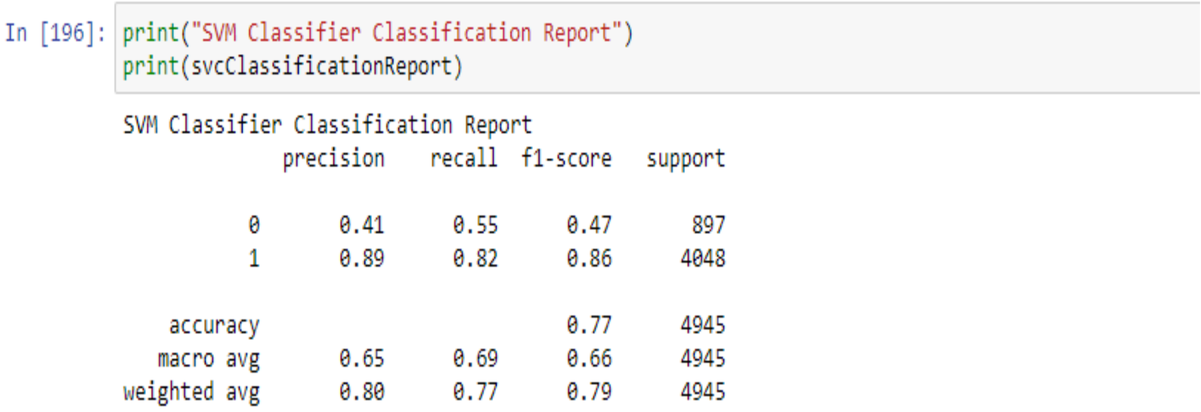
SUPPORT VECTOR MACHINE MODEL

The SVM model is trained using a linear kernel and a balanced class weigh, and a C value of 1.0. The result from using the support vector machine classifier model to predict the sentiment analysis of the is shown below in fig. 4.2 below.



```
In [195]: print("SVM Classifier confusion matrix")
          print(svcConfusionMatrix)

          SVM Classifier confusion matrix
          [[ 493  404]
           [ 714 3334]]
```

Fig. 4.2. SVM Classifier Confusion Matrix



```
In [196]: print("SVM Classifier Classification Report")
          print(svcClassificationReport)

          SVM Classifier Classification Report
                        precision    recall  f1-score   support

                     0       0.41      0.55      0.47       897
                     1       0.89      0.82      0.86      4048

              accuracy                           0.77      4945
             macro avg       0.65      0.69      0.66      4945
          weighted avg       0.80      0.77      0.79      4945
```

Fig. 4.3. SVM Classifier Classification Report

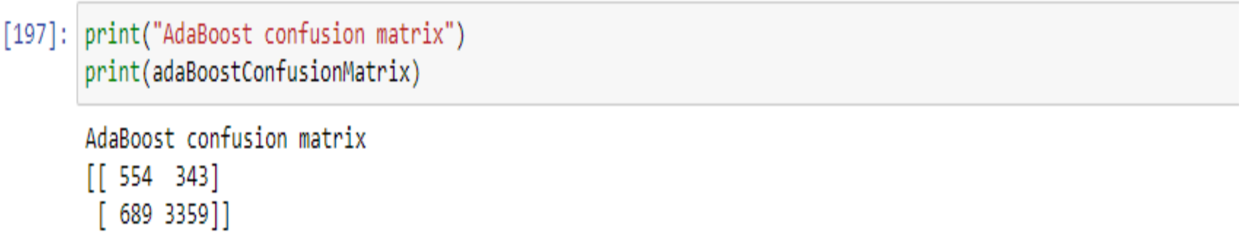
The SVM classifier handles classifying negative sentiments better compared to the logistic regression model as it has a true negative value of 714. It appears to be the model with the highest number of true negatives among the 3 models.

The SVM model has a 77% accuracy, 80% precision, 77% recall, and a f1-score of 79%, according to the classification report.

ADABOOST MODEL

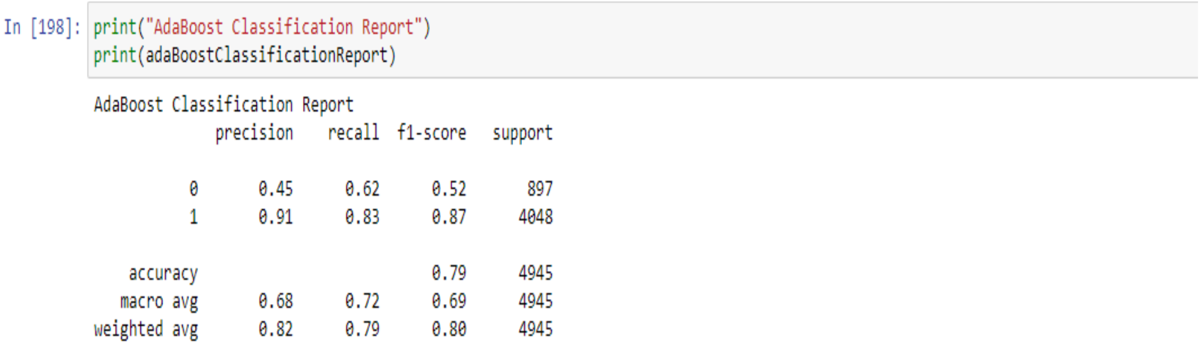
The AdaBoost Model is a boosting algorithm and the base algorithm used in this case is the decision tree classifier algorithm. The reason for using decision tree as the base estimator is because it is versatile and creates a robust ensemble model.

This algorithm creates a balance between the negative and positive predictions.



```
[197]: print("AdaBoost confusion matrix")
       print(adaBoostConfusionMatrix)

AdaBoost confusion matrix
[[ 554  343]
 [ 689 3359]]
```

Fig. 4.4. AdaBoost Confusion Matrix



```
In [198]: print("AdaBoost Classification Report")
          print(adaBoostClassificationReport)

AdaBoost Classification Report
              precision    recall  f1-score   support

           0       0.45      0.62      0.52       897
           1       0.91      0.83      0.87      4048

    accuracy                           0.79      4945
   macro avg       0.68      0.72      0.69      4945
weighted avg       0.82      0.79      0.80      4945
```

Fig. 4.5. AdaBoost Classification Report

The accuracy score for the model is 79%, the precision is 82%, the recall is 79%, and the f1-score is 80%.

4.7 GRAPHICAL REPRESENTATION OF THE MACHINE LEARNING MODEL RESULTS

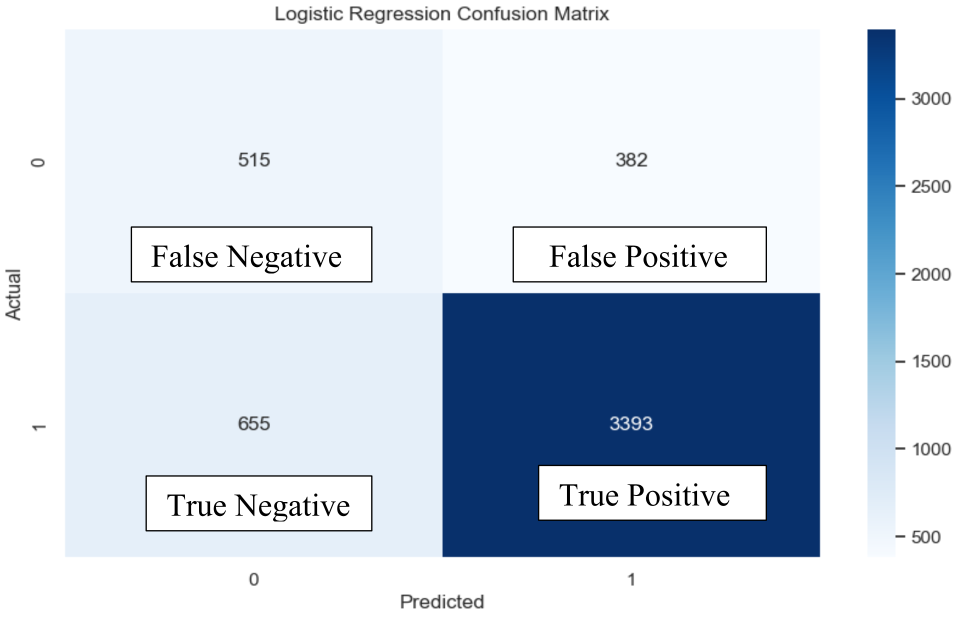


Fig. 4.6. CONFUSION MATRIX OF THE LOGISTIC REGRESSION MODEL

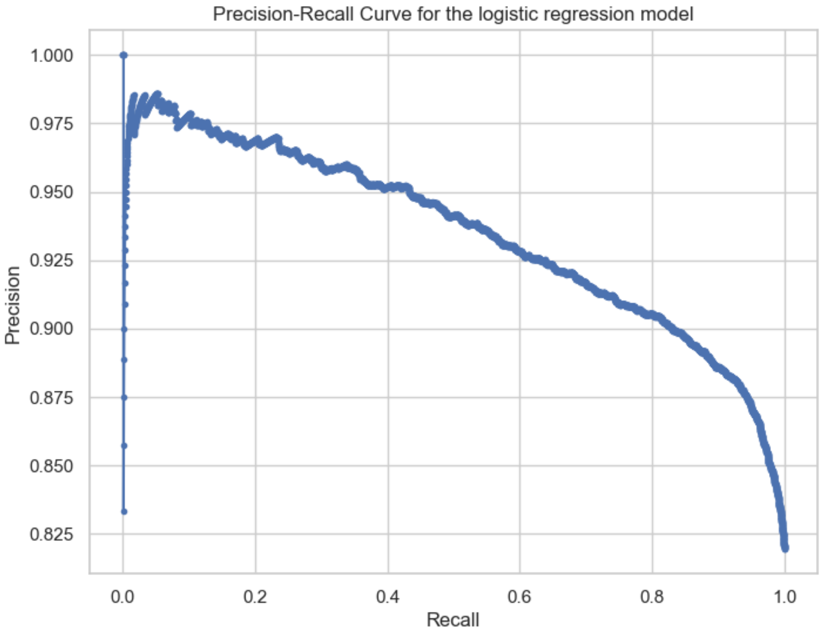


Fig. 4.7. PRECISION-RECALL CURVE OF THE LOGISTIC REGRESSION MODEL

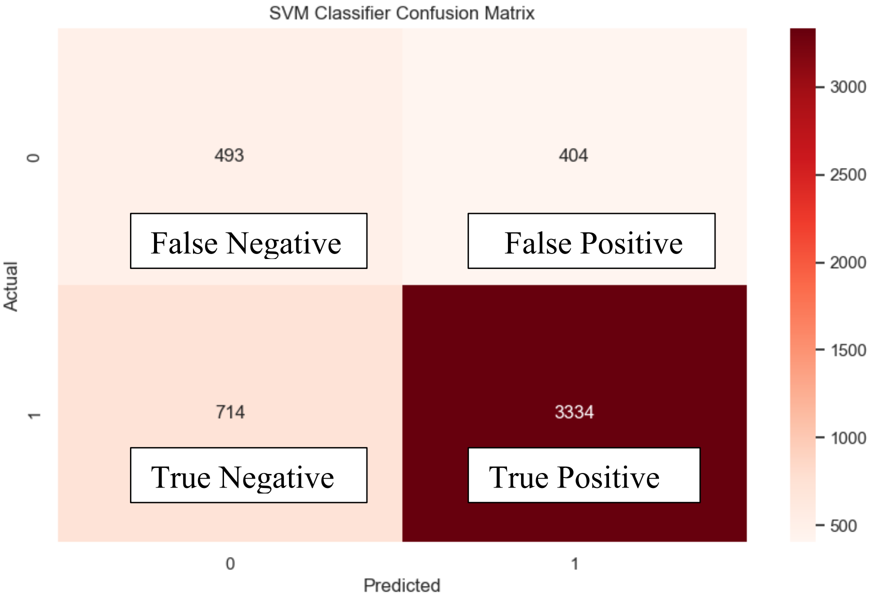


Fig. 4.8. CONFUSION MATRIX OF THE SVM CLASSIFIER MODEL

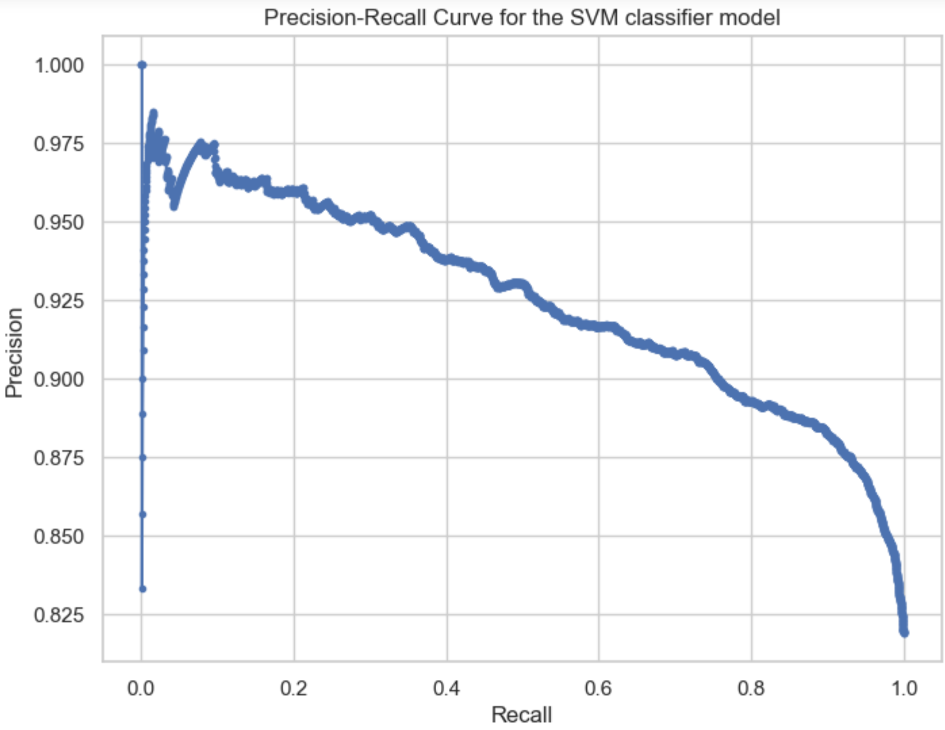


Fig. 4.9. PRECISION RECALL CURVE OF THE SVM CLASSIFIER MODEL

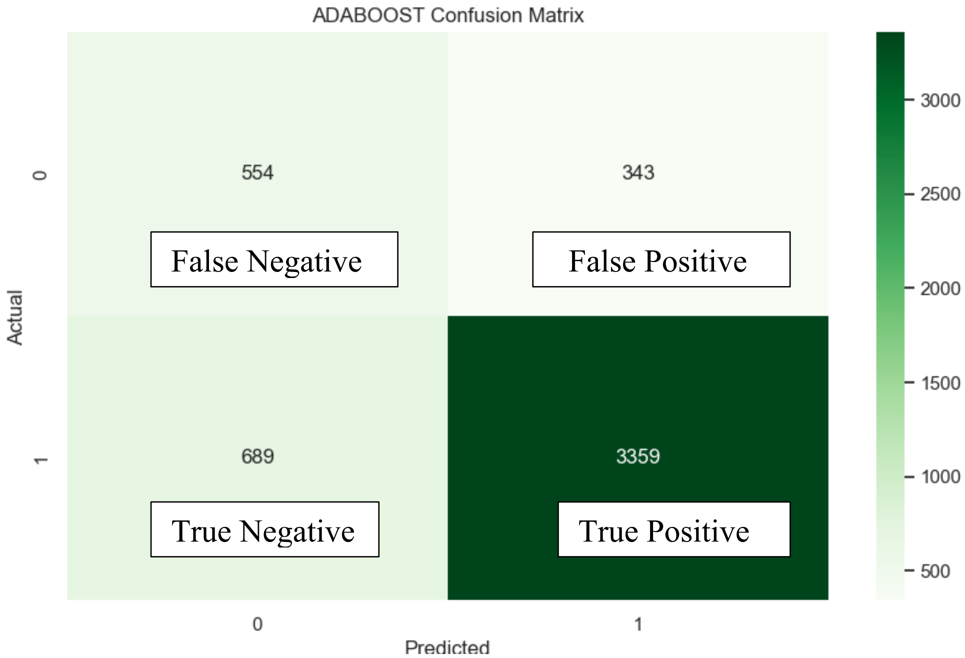


Fig. 4.10. CONFUSION MATRIX OF THE ADABOOST MODEL

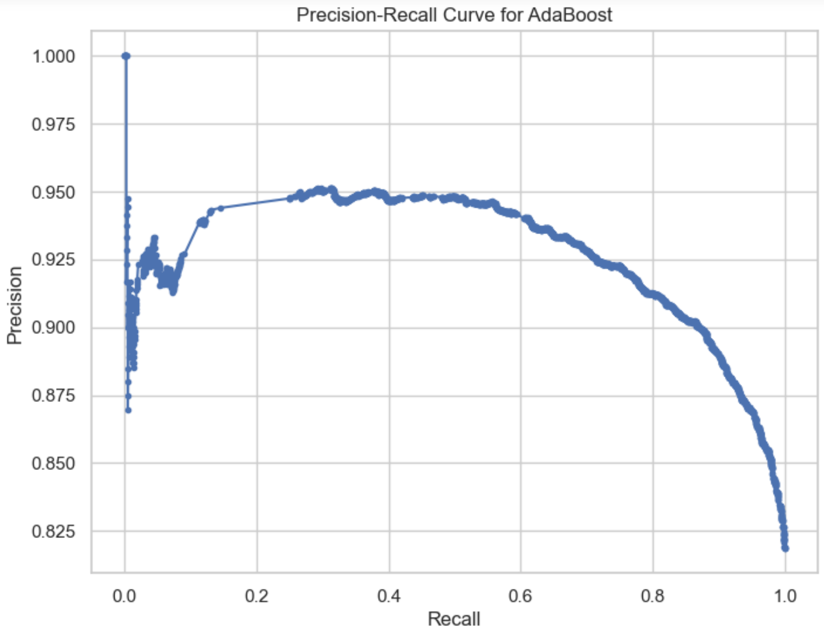


Fig. 4.11. PRECISION RECALL CURVE OF THE ADABOOST MODEL

4.8 COMPARISON OF MODEL RESULT

	LOGISTIC REGRESSION	SUPPORT VECTOR MACHINE	ADABOOST
ACCURACY (%)	79	77	79
PRECISION (%)	82	80	82
RECALL (%)	79	77	79
F1 SCORE (%)	80	79	80
TRUE POSITIVE	3393	3334	3359
TRUE NEGATIVE	655	714	689
FALSE POSITIVE	382	404	343
FALSE NEGATIVE	515	493	554

Table 4.6. Comparative analysis of the machine learning models.

From Table 4.6 we can see the metrics of the various supervised machine-learning models.

- **Accuracy:** Although accuracy is a crucial indicator in sentiment analysis, it may not always present a whole picture, particularly in cases where classes are unbalanced. The maximum accuracy is attained by Adaboost and Logistic Regression, both at 79%. This indicates that they accurately categorise roughly 79% of evaluations, both positive and negative.
- **Precision:** Adaboost and Logistic Regression have a precision of 82%, which is the highest. This suggests that these models are accurate roughly 82% of the time when they predict a review to be positive. This accuracy is significant in situations when accurately identifying good evaluations in reviews.
- **Recall:** The recall rate for all models is 79%. Accordingly, they can recognise about 79% of the genuine favourable evaluations. When it's critical not to miss positive reviews, a high recall is important.
- **F1 Score:** The two models with the highest F1 scores, Adaboost and Logistic Regression, both have 80%. This suggests a fair trade-off between recall and precision. While maintaining a healthy balance between false positives and false negatives, these models successfully categorise both positive and negative reviews with good accuracy.
- **True Positive:** Adaboost has the highest true positive count, which indicates that it forecasts most positive reviews with the most accuracy.
- **True Negative:** Support Vector Machine has the greatest true negative count, which suggests that it is more accurate in predicting bad reviews.

- **False Positive:** Logistic Regression frequently predicts a review as positive when it is negative because it has the highest false positive rate.
- **False Negative:** Adaboost has the largest number of false negatives, which indicates that occasionally it forecasts a review as unfavourable when it is positive.

The image below – Fig 4.12 shows the comparisons between the 3 machine learning models. The recall and precision metrics is used for the comparisons.

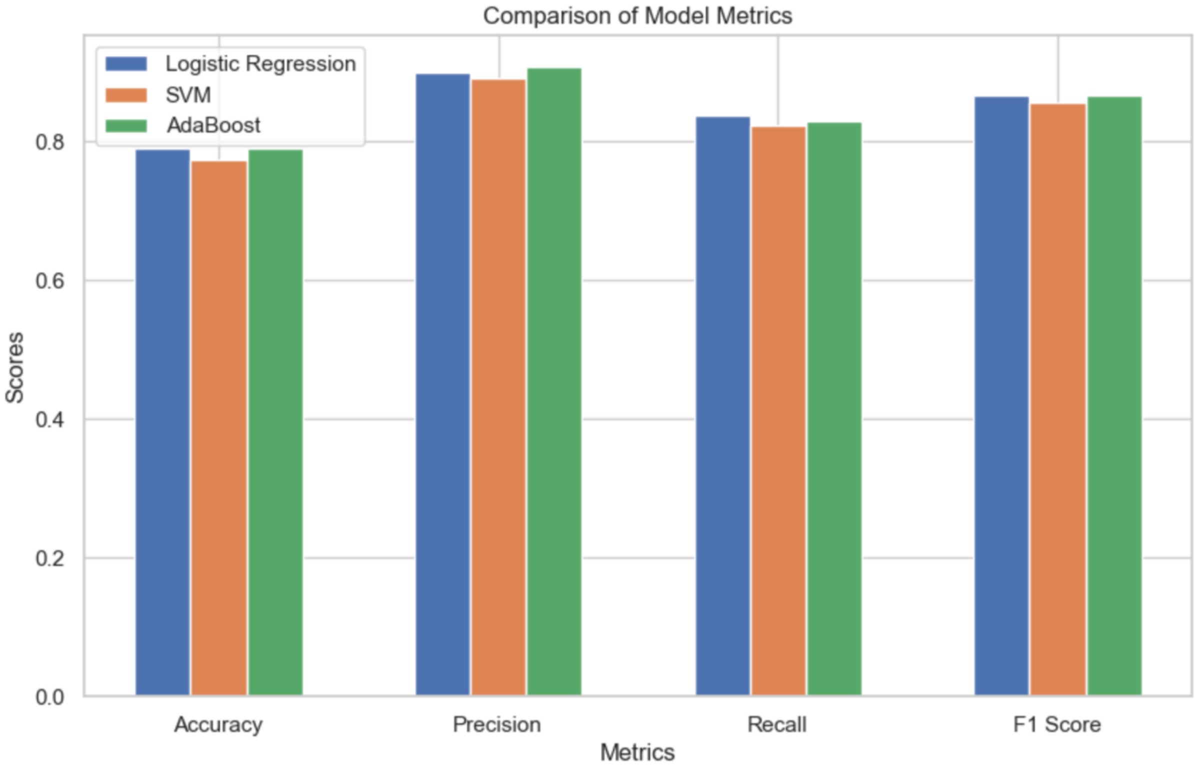


Fig. 4.12. COMPARISON OF THE THREE MODEL

4.9 COMPARISON OF LEXICON-BASED TECHNIQUES RESULTS

This section is to analyze the results of the lexicon-based sentiment analysis techniques used in the research. The two lexicon-based techniques used are Textblob and Vader sentiment.

	TEXTBLOB	VADER SENTIMENT
NEGATIVE (%)	4.7	2.3
NEUTRAL (%)	13.3	1.1
POSITIVE (%)	82.0	96.6

Table 4.7. Comparative analysis of the lexicon-based models

The lexicon-based approach is not as robust as the traditional machine learning models as it has to depend on creating a threshold for positive, negative, and neutral sentiment based on the polarity score, and that the result for this is different for the textblob and vader sentiment models.

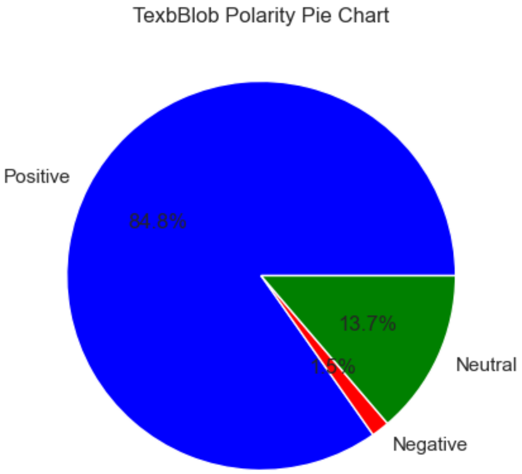


Fig. 4.12. TEXTBLOB Polarity Pie Chart

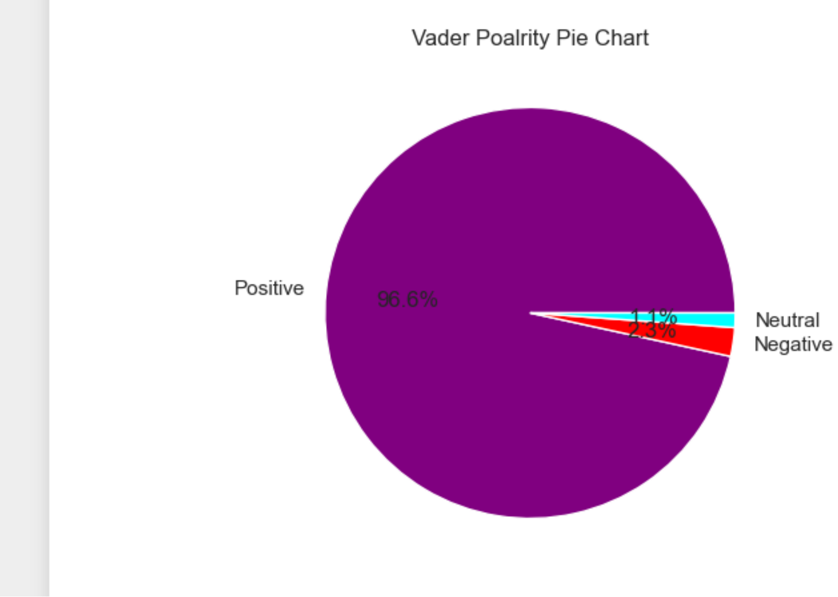


Fig. 4.13. VADER SENTIMENT Polarity Pie Chart

Interesting conclusions are drawn from (Fig 4.12, Fig 4.13) the comparison of VADER with TextBlob for sentiment analysis. When it comes to identifying negative emotions, VADER outperforms TextBlob, classifying 2.3% of the text as negative versus 4.7% for TextBlob. This shows that VADER is better able than TextBlob in detecting positive attitudes. In contrast, although VADER only classifies 1.1% of the text as neutral, TextBlob classifies 13.3% of the material as neutral, demonstrating TextBlob's stronger propensity to do so. When it comes to positive feelings, VADER identifies 96.6% of the text as such, while TextBlob assigns 82.0%. This demonstrates how variable VADER's classification of positive sentiment is.

5 DISCUSSION AND EVALUATION OF RESULTS

In this section, we discuss the key findings as well as the interpretation and analysis of sentiment analysis results, we have obtained using the lexicon-based tools and the machine learning algorithms.

5.1 KEY FINDINGS

The sentiment analysis done using the lexicon-based algorithms TextBlob and VADER, using the fashion review dataset, revealed some fascinating insights. The TextBlob and VADER techniques showed a very high percentage of 82% and 92.2% positive sentiments, respectively, while they showed minimal, neutral results of 13.3% and 1.1% and negative results of 4.7% and 2.3%.

The supervised machine learning algorithms also create valuable insights into sentiment analysis. The classification is binary; hence, there are only negative and positive sentiments. Logistic regression has a 79% accuracy, 82% precision, 79% recall, and an 80% f1 score. The SVM Classifier has a 77% accuracy, 80% precision, 77% memory, and a 79% f1-score. AdaBoost has a 79% accuracy, 82% precision, 79% memory, and an 80% F1 score.

5.2 IMPLICATIONS AND APPLICATIONS

Through the sentiment analysis project, the fashion product company can gain valuable insights into customer feedback, satisfaction, and preferences. This information can be used to improve products, develop effective marketing strategies, and better meet the needs of customers. Additionally, identifying positive sentiment towards certain features can help identify emerging trends and inform strategic product development. The sentiment analysis can also be used for brand reputation management by monitoring real-time sentiments and swiftly responding to negative feedback. This project can serve as a foundation for future research on the use of machine learning models for sentiment analysis.

5.3 ETHICAL CONSIDERATIONS

This project has followed all ethical rules and standards. Ethics approval was not necessary because the dataset used does not contain any personal information or identities of the users involved. Additionally, the fashion store whose reviews were used is not identified in the dataset. The dataset was obtained from a public data repository. The study focused on the ethical, social, and legal issues highlighted by the university, and no violations were committed.

6 CONCLUSION AND RECOMMENDATIONS

6.1 CONCLUSION

After examining the reviews using both a lexicon-based approach and supervised machine learning models, we found that the TextBlob and Vader tools mostly categorized the reviews as positive. However, there is still an imbalance present. The TextBlob method classified a significant number of reviews as neutral, but this isn't an accurate representation as the neutral category is based on the tradeoff between negative and positive reviews.

The machine learning algorithms, on the other hand categorized the reviews as either negative or positive and performed well in organizing them. We have concluded that the majority of the reviews are positive, with fewer negative reviews. This is expected due to the imbalanced dataset.

Overall, we strive for fairness and accuracy in our analysis, regardless of the tool or method used.

6.2 RECOMMENDATIONS

In this project, we investigated sentiment analysis for a specific type of reviews. However, we suggest utilizing a wider range of data to evaluate the effectiveness of the models. Additionally, businesses that operate in international markets could expand the project to incorporate multi-lingual sentiment analysis and cross-domain assessment to determine the system's ability to adapt to different contexts.

REFERENCES

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A. and Aljaaf, A.J., 2020. A systematic review on supervised and unsupervised machine learning algorithms for data science. *Supervised and unsupervised learning for data science*, pp.3-21.

Azungah, T., 2018. 'Qualitative research: deductive and inductive approaches to data analysis', *Qualitative Research Journal*, 18(4), pp. 383–400. Available at: https://doi.org/10.1108/QRJ-D-18-00035/FULL/XML.

Balakrishnan, V. and Lloyd-Yemoh, E., 2014. 'Stemming and lemmatization: A comparison of retrieval performances.

Behdenna, S., Barigou, F. and Belalem, G., 2018. Document level sentiment analysis: a survey. *EAI endorsed transactions on context-aware systems and applications*, *4*(13), pp.e2-e2.

Bird, S., 2006. 'NLTK: The Natural Language Toolkit', pp. 69–72.

Bongirwar, V.K., 2015. A survey on sentence level sentiment analysis. *International Journal of Computer Science Trends and Technology (IJCST)*, *3*(3), pp.110-113.

Chowdhary, K.R., 2020. 'Natural Language Processing', *Fundamentals of Artificial Intelligence*, pp. 603–649. Available at: https://doi.org/10.1007/978-81-322-3972-7_19.

Christopher, D.M. and Hinrich, S., 1999. Foundations of statistical natural language processing.

Chu, X., Ilyas, I.F., Krishnan, S. and Wang, J., 2016, June. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data* (pp. 2201-2206). Available at: https://doi.org/10.1145/2882903.2912574.

Hasan, A., Moin, S., Karim, A. and Shamshirband, S., 2018. Machine learning-based sentiment analysis for twitter accounts. *Mathematical and computational applications*, *23*(1), p.11.https://doi.org/10.3390/MCA23010011.

Hayes, B.K., Heit, E. and Swendsen, H., 2010 ‘Inductive reasoning’, *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(2), pp. 278–292. Available at: https://doi.org/10.1002/WCS.44.

Jain, P.K., Pamula, R. and Srivastava, G., 2021. ‘A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews’, *Computer Science Review*, 41, p. 100413. Available at: https://doi.org/10.1016/J.COSREV.2021.100413.

Jordan, M.I. and Mitchell, T.M.,2015. ‘Machine learning: Trends, perspectives, and prospects’, *Science*, 349(6245), pp. 255–260. Available at: https://doi.org/10.1126/SCIENCE.AAA8415.

Khan, A., Kundi, F.M., Ahmad, S. and Asghar, M.Z., 2014. Lexicon-based sentiment analysis in the social web. *Journal of Basic and Applied Scientific Research*, *4*(6), pp.238-48.Available at: https://www.researchgate.net/publication/283318830 (Accessed: 24 August 2023)..

Komorowski, M., Data, M.C., Marshall, D.C., Salciccioli, J.D. and Crutain, Y., 2016. Exploratory data analysis. *Secondary analysis of electronic health records*, pp.185-203. Available at: https://doi.org/10.1007/978-3-319-43742-2_15.

Kozhevnikov, V.A. and Pankratova, E.S., 2020. ‘RESEARCH OF THE TEXT DATA VECTORIZATION AND CLASSIFICATION ALGORITHMS OF MACHINE LEARNING’. Available at: https://doi.org/10.15863/TAS.

Kumaresh, N. Bonta, V., and Janardhan, N., 2019. A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, *8*(S2), pp.1-6. Available at: https://doi.org/10.51983/ajcst-2019.8.S2.2037.

Lang, C., Li, M. and Zhao, L., 2020. Understanding consumers’ online fashion renting experiences: A text-mining approach. *Sustainable Production and Consumption*, *21*, pp.132-144.

LaValley, M.P., 2008. ‘Logistic Regression’, *Circulation*, 117(18), pp. 2395–2399. Available at: https://doi.org/10.1161/CIRCULATIONAHA.106.682658.

Liu, B., 2022. *Sentiment analysis and opinion mining*. Springer Nature.

Medhat, W., Hassan, A. and Korashy, H. (2014a) 'Sentiment analysis algorithms and applications: A survey', *Ain Shams Engineering Journal*, 5(4), pp. 1093–1113. Available at: https://doi.org/10.1016/J.ASEJ.2014.04.011.

Medhat, W., Hassan, A. and Korashy, H. (2014b) 'Sentiment analysis algorithms and applications: A survey', *Ain Shams Engineering Journal*, 5(4), pp. 1093–1113. Available at: https://doi.org/10.1016/J.ASEJ.2014.04.011.

Meyer, C. and Schwager, A., 2007. Understanding customer experience. *Harvard business review*, *85*(2), p.116.

Mohammad, S.M., 2016. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In *Emotion measurement* (pp. 201-237). Woodhead Publishing.

Morgenthaler, S., 2009. 'Exploratory data analysis', *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1), pp. 33–44. Available at: https://doi.org/10.1002/WICS.2.

Nguyen, H., Veluchamy, A., Diop, M. and Iqbal, R., 2018. Comparative study of sentiment analysis with product reviews using machine learning and lexicon-based approaches. *SMU Data Science Review*, *1*(4), p.7. Available at: https://scholar.smu.edu/datasciencereviewAvailableat:https://scholar.smu.edu/datasciencereview/vol1/iss4/7http://digitalrepository.smu.edu. (Accessed: 14 August 2023).

Osisanwo, F.Y., Akinsola, J.E.T., Awodele, O., Hinmikaiye, J.O., Olakanmi, O. and Akinjobi, J., 2017. Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*(3), pp.128-138. Available at: https://doi.org/10.14445/22312803/IJCTT-V48P126.

Rahm, E., and Do, H.H., 2000. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, *23*(4), pp.3-13.

Rajeswari, A.M., Mahalakshmi, M., Nithyashree, R. and Nalini, G., 2020, July. Sentiment analysis for predicting customer reviews using a hybrid approach. In 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA) (pp. 200-205). IEEE. Available at: https://doi.org/10.1109/ACCTHPA49271.2020.9213236.

Schapire, R.E., 2013. ‘Explaining adaboost’, *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pp. 37–52. Available at: https://doi.org/10.1007/978-3-642-41136-6_5/COVER.

Sentiment Analysis and Opinion Mining - Bing Liu - Google Books (no date). Available at: https://books.google.com.ng/books?hl=en&lr=&id=xYhyEAAAQBAJ&oi=fnd&pg=PP1&ots=rk_wJEO6Et&sig=birSsdCl4Nf3rqliMWrcFyFHXD0&redir_esc=y#v=onepage&q&f=false (Accessed: 17 August 2023).

Setiawan, E.B., Widyantoro, D.H. and Surendro, K., 2018. ‘Feature Expansion for Sentiment Analysis in Twitter’, *Proceeding of the Electrical Engineering Computer Science and Informatics*, 5(5). Available at: https://doi.org/10.11591/EECSI.V5I5.1660.

Sharma, N. and Tyagi, A., 2018. ‘Sentiment Analysis using Logistic Regression and Effective Word Score Heuristic Article in’, *International Journal of Engineering & Technology*, 7(2), pp. 20–23. Available at: https://doi.org/10.14419/ijet.v7i2.24.11991.

Shivaprasad, T.K. and Shetty, J.,2017. ‘Sentiment analysis of product reviews: A review’, *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2017*, pp. 298–303. Available at: https://doi.org/10.1109/ICICCT.2017.7975207.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M., 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, *37*(2), pp.267-307.Available at: https://doi.org/10.1162/COLI_A_00049.

Tasnim, Nazia. "Machine Learning in the Classification of Computer Code." 2020, https://core.ac.uk/download/346635154.pdf.

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A.N., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., Parmar, N. and Sepassi, R., 2018. Tensor2tensor for neural machine translation. arXiv preprint arXiv:1803.07416.

Wankhade, M., Rao, A.C.S. and Kulkarni, C., 2022. A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review, 55(7), pp.5731-5780.

Wendland, A., Zenere, M. and Niemann, J., 2021. 'Introduction to Text Classification: Impact of Stemming and Comparing TF-IDF and Count Vectorization as Feature Extraction Technique', *Communications in Computer and Information Science*, 1442, pp. 289–300. Available at: https://doi.org/10.1007/978-3-030-85521-5_19/COVER.

Yerpude, A., Phirke, A., Agrawal, A. and Deshmukh, A., 2019. Sentiment analysis on product features based on lexicon approach using natural language processing. *International Journal on Natural Language Computing (IJNLC)*, *8*(3), pp.1-15.Available at: https://doi.org/10.5121/ijnlc.2019.8301.

Zainuddin, N. and Selamat, A., 2014 'Sentiment analysis using Support Vector Machine', *I4CT 2014 - 1st International Conference on Computer, Communications, and Control Technology, Proceedings*, pp. 333–337. Available at: https://doi.org/10.1109/I4CT.2014.6914200.

Zhang, L., Wang, S. and Liu, B., 2018. Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), p.e1253.

APPENDIX

Below is the code for the Project.

```
# IMPORTING MODULES TO BE USED
import pandas as pd
import numpy as np
import scipy
import re
import string

import seaborn as sns
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import sent_tokenize, word_tokenize
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer

from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.decomposition import PCA

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.svm import SVC


# Read the CSV file containing Women's Clothing E-Commerce Reviews
# and store the data in the 'df' DataFrame using Pandas
df = pd.read_csv("./Womens Clothing E-Commerce Reviews.csv")

# Showing the first 5 data in the dataframe
df.head()

#printing dataframe columns
df.columns
```

```
#Performing statistical summary of the data
df.describe()
df.describe().T
# Getting info about the dataset
df.info()

df.shape
```

The above infers that the original dataset has 23,486 rows and 11 columns

```
# Grouping datasets  by Ratings and getting the number of recommendation each rating has:
df.groupby(['Rating'])['Recommended IND'].count()

# Grouping datasets  by Ratings and then by Recommendations and getting the number of positive and negative recommendation each rating has:
df.groupby(['Rating', 'Recommended IND'])['Recommended IND'].count()
```

From the analysis above, it can be seen that products rated 1, 2 (low) have low recommendations while those rated high (4,5) have high recommendations¶

Creating another dataFrame which includes just the needed columns.

```
reviewDf = df[["Title", "Review Text", "Rating", "Recommended IND"]]
reviewDf.head()
reviewDf.tail()
reviewDf.shape
# Checking for the sum of nall columns in each row of the new dataframe
reviewDf["Review Text"].isna().sum()
reviewDf["Title"].isna().sum()
reviewDf["Rating"].isna().sum()
reviewDf["Recommended IND"].isna().sum()
```

Combining the Title and Review Text field into one:

```
reviewDf["ReviewContent"] = reviewDf["Title"] + " " + reviewDf["Review Text"]
reviewDf.head()
# Dropping the "Review Text" and the "Title" column as it is no longer needed
reviewDf = reviewDf.drop(["Review Text", "Title"],axis=1)
reviewDf.head()
reviewDf.isna().sum()
```

Seeing that there are 3811 empty columns in the ReviewContent column, we need to drop the columns.

```
reviewDf.dropna()
reviewDf.head()

#barchart showing product rating
sns.countplot(x="Rating", data=reviewDf, palette="Set1")
plt.xlabel("Rating")
plt.ylabel("Count")
plt.title("Count of Ratings")
plt.grid(True)
plt.show()

#Renaming the field
reviewDf.rename(columns={"Recommended IND": "Recommended"})

#bar plot showing customer recommendations.
sns.countplot(x="Recommended IND", data=reviewDf)
plt.legend(labels=['Low', 'High'], title="Recommended IND")
plt.grid(True)
plt.show()
reviewDf.drop("Rating", axis=1)

# Create the bar plot using seaborn showing product rating and customer recommendation.
sns.set(style='whitegrid')
plt.figure(figsize=(8, 4))
colors = ['#FF6F00', '#0072ff']
sns.countplot(x='Rating', data=reviewDf, hue='Recommended IND', palette=colors)
# Set plot title and labels
plt.title('Rating Bar Plot')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.legend(title='Rating Recommendation', labels=['Not Recommended', 'Recommended'])
plt.grid(True)
# Display the plot
plt.show()
```

Data cleaning

```
#Removing all non alphabeths in by converting them to a whitespace " "
def cleanText(words):
   words = re.sub("[^a-zA-Z]"," ", words)
   word = words.lower().split()
   return " ".join(word)
#Converting the datatype of the ReviewContent column to string to be able to use the regex
package on it:
reviewDf['ReviewContent'] = reviewDf['ReviewContent'].astype(str)
#Applying the clean text function to the ReviewContent column and saving result to another
column called cleanedreview
reviewDf['CleanedReview'] = reviewDf['ReviewContent'].apply(cleanText)
reviewDf.head()
```

```
#dropping nan values
reviewDf = reviewDf.drop(reviewDf[reviewDf['CleanedReview'] == 'nan'].index)
reviewDf.head()

#display content of the CleanedReview
with pd.option_context('display.max_colwidth', None):
  display(reviewDf['CleanedReview'])

#Tokenization of ReviewContent
from nltk.tokenize import TweetTokenizer
tokenizer = TweetTokenizer()
reviewDf['CleanedReview'] = reviewDf['CleanedReview'].apply(tokenizer.tokenize)
reviewDf['CleanedReview']= [' '.join(map(str, token)) for token in reviewDf['CleanedReview']]
reviewDf.head()
```

7 Using TextBlob to determine review Polarity.

```
!pip install textblob
conda install -c conda-forge textblob
from textblob import TextBlob
def getPolarity(sentence):
    return TextBlob(sentence).sentiment.polarity
reviewDf['TextBlobPolarityScore'] = reviewDf["CleanedReview"].apply(getPolarity)
reviewDf.head()
reviewDf.tail()
reviewDf['CleanedReview'].head()

#display content of the CleanedReview
with pd.option_context('display.max_colwidth', None):
display(reviewDf['CleanedReview'])

# Printing the first 10 reviews with negative polarity
reviewDf[reviewDf["TextBlobPolarityScore"] < 0].head(10)

#getting compound TextBlob compound score
def convertTextBlobPolarityToCategory(score):
    if score >= 0.1:
        return 'Positive'
    elif score <= -0.1:
        return 'Negative'
    else:
        return 'Neutral'

reviewDf['TextBlobPolarityCategory'] =
reviewDf['TextBlobPolarityScore'].apply(convertTextBlobPolarityToCategory)
reviewDf.head()
```

Polarity score greater than 0 to 1 indicates positive polarity while those lesser than 0 to -1 indicate negative polarity.

Using Vader to determine review Polarity.

```
!pip install vaderSentiment
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
def getPolarity(sentence):
   analyzer = SentimentIntensityAnalyzer()
   return analyzer.polarity_scores(sentence)['compound']

reviewDf['VaderPolarityScore'] = reviewDf["CleanedReview"].apply(getPolarity)
reviewDf.head()

#getting vader compound score and using it to assign a category to the reviews

def getVaderPolarity(sentence):
   vader = SentimentIntensityAnalyzer()
   if vader.polarity_scores(sentence)['compound'] >= 0.1:
      return 'Positive'
   elif vader.polarity_scores(sentence)['compound'] <= -0.1:
      return 'Negative'
   else:
      return 'Neutral'
reviewDf['VaderPolarityCategory'] = reviewDf['CleanedReview'].apply(getVaderPolarity)
reviewDf.head()
```

where the sentiments are different (i.e where Textblob sentiment and Vader sentiment are not same return 0 and if they are same return 1)

```
#Visual representation of positive, negtive, neutral for Textblob sentiment and Vader sentiment.

negative = (len(reviewDf.loc[reviewDf.TextBlobPolarityScore <-
0.1,['CleanedReview']].values)/len(reviewDf))*100
positive = (len(reviewDf.loc[reviewDf.TextBlobPolarityScore
>0.1,['CleanedReview']].values)/len(reviewDf))*100
neutral  = len(reviewDf.loc[reviewDf.TextBlobPolarityScore >-0.1 ,['CleanedReview']].values) -
len(reviewDf.loc[reviewDf.TextBlobPolarityScore >0.1 ,['CleanedReview']].values)
neutral = neutral/len(reviewDf)*100
plt.figure(figsize =(10, 5))
plt.title("TexbBlob Polarity Pie Chart")
plt.pie([positive,negative,neutral], labels = ['Positive','Negative','Neutral'] , colors = ["blue"
,"red" ,"green"], autopct='%1.1f%%')
#Visual representation of positive, negtive, neutral for Vader sentiment.
```

```
negative = (len(reviewDf.loc[reviewDf.VaderPolarityCategory ==
"Negative",['CleanedReview']].values)/len(reviewDf))*100
positive = (len(reviewDf.loc[reviewDf.VaderPolarityCategory ==
"Positive",['CleanedReview']].values)/len(reviewDf))*100
neutral  = len(reviewDf.loc[reviewDf.VaderPolarityCategory == "Neutral"
,['CleanedReview']].values)
neutral = neutral/len(reviewDf)*100
plt.figure(figsize =(10, 5))
plt.title("Vader Poalrity Pie Chart")
plt.pie([positive,negative,neutral], labels = ['Positive','Negative','Neutral'] , colors = ["purple"
,"red" ,"cyan"], autopct='%1.1f%%')
#Using English Stop words to clean the Data.
englishStopwords = stopwords.words("english")
englishStopwords[:20]
englishStopwords[::10]
clothingStopwords =['top', 'blowse', 'sweater','shirt',
            'skirt', 'pant','material', 'dress',  'white', 'black',
            'jeans', 'fabric', 'color','order', 'wear', 'suit', 'jacket', 'boxers', '']
#Creating a function to remove all of the stopwords both ENglish and clothing stopwords from
the reviews
def removeStopwords(review):
   text = []
   words = review.split()
   for word in words:
      if word.lower() not in englishStopwords and word.lower() not in clothingStopwords:
         text.append(word.lower())
   return " ".join(text)
reviewDf["CleanedReview"] = reviewDf["CleanedReview"].apply(removeStopwords)
reviewDf["CleanedReview"].head(10)


#display content of the CleanedReview after removingstopwords
with pd.option_context('display.max_colwidth', None):
  display(reviewDf['CleanedReview'])
reviewDf.head()
```

Observation up to this point: Stop words have been removed and th next step is to break words into their most basic and simplest form using Lemmatization.

```
# Creating a function to lemmatize the cleaned reviews

lemmatizer = WordNetLemmatizer()

def lemmatize_review(reviews):

   # Split the review into individual words

   words = reviews.split()

   # Lemmatize each word and join them back into a sentence
```

```
    lem_text = [lemmatizer.lemmatize(word) for word in words]
    return " ".join(lem_text)


reviewDf["LemmatizedReview"] = reviewDf["CleanedReview"].apply(lemmatize_review)

#display content of the CleanedReview after Lemmatization
with pd.option_context('display.max_colwidth', None):
display(reviewDf['LemmatizedReview'])
reviewDf.head()

from wordcloud import WordCloud
words = ' '.join(reviewDf["LemmatizedReview"])
#wordcloud = WordCloud().generate(words)

wordcloud = WordCloud(background_color="white",max_words=len(words),\
                  max_font_size=40, relative_scaling=.5, colormap='gist_heat').generate(words)
plt.figure(figsize=(13,13))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Performing Feature Extraction using CountVectorization

```
reviews = reviewDf['LemmatizedReview']
# Create a Vectorizer Object
vectorizer = CountVectorizer()
vectorized = vectorizer.fit(reviews)
review_counts = vectorizer.transform(reviews)
review_counts

[i for i in vectorized.vocabulary_.items()][5:100:5]
```

8 Using TF-IDF (Term Frequency-Inverse Document Frequency) to assign weight to each word.

```
tfidf_transformer = TfidfTransformer().fit(review_counts)

reviews_tfidf = tfidf_transformer.transform(review_counts)
reviews_tfidf

reviews_tfidf = reviews_tfidf.toarray()

# Converting the text to sparse matrix generated to a pandas dataframe
reviews_tfidf = pd.DataFrame(reviews_tfidf)
reviews_tfidf.head(10)
reviews_tfidf.shape
```

Creating a new dataframe by merging the reviewDf with the new dataframe of the sparse matric¶

```
finalDf = pd.merge(reviewDf, reviews_tfidf,right_index=True, left_index=True)
```

```
finalDf.head()
```

```
finalDf = finalDf.drop("Rating", axis=1)
```

Dropping all the reviews columns from the dataframe

```
finalDf = finalDf.drop(['ReviewContent', 'CleanedReview', 'LemmatizedReview', 'VaderPolarityCategory', 'TextBlobPolarityCategory'], axis=1)
```

```
finalDf.head()
```

```
###

X = finalDf.drop("Recommended IND", axis=1)
y = finalDf["Recommended IND"]

X.shape
y.shape

###
from sklearn.model_selection import train_test_split
#Train-Test Split
seed=101
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3, stratify=y, random_state=seed)
# Data Scaling using Min-Max Scaling
scaler = MinMaxScaler()
X_train_s = scaler.fit_transform(X_train)
X_test_s = scaler.transform(X_test)
```

8.1 Dimensionality Reduction using PCA

```
pca_transformer = PCA(n_components=2).fit(X_train_s)
X_train_s_pca = pca_transformer.transform(X_train_s)
X_test_s_pca = pca_transformer.transform(X_test_s)
X_train_s_pca[:1]
```

```
X_train_s_pca.shape

X_train_s = scipy.sparse.csr_matrix(X_train_s)
X_test_s = scipy.sparse.csr_matrix(X_test_s)

X_train = scipy.sparse.csr_matrix(X_train.values)
X_test = scipy.sparse.csr_matrix(X_test.values)
X_test

X_train.shape
```

9 Creating the Machine Learning Models

Logistic Regression Model

```
#Logistic Regression
logRegModel = LogisticRegression(class_weight='balanced',
                        random_state=seed,max_iter=1000)
logRegModel.fit(X_train, y_train)
logRegPred = logRegModel.predict(X_test)
logRegPred
##
from sklearn.metrics import accuracy_score, precision_score, confusion_matrix
logAccuracy = accuracy_score(y_test, logRegPred)
logPrecision = precision_score(y_test, logRegPred)
print(logRegModel.classes_)

logConfusionMatrix = confusion_matrix(y_test, logRegPred, labels=logRegModel.classes_)
logRegMetricsDf = pd.DataFrame({'Metric': ['Accuracy', 'Precision'],
                       'Score': [logAccuracy, logPrecision]})
from sklearn.metrics import classification_report
logRegClassificationReport = classification_report(y_true=y_test, y_pred=logRegPred)
print("Classification Report for the Logistic Regression Model")
print(logRegClassificationReport)
logRegMetricsDf.head()

print("Logistic Regression confusion matrix")
print(logConfusionMatrix)
```

SVM Classifier Model

```
svc_model = SVC(C=1.0,
          kernel='linear',
          class_weight='balanced',
          probability=True,
            random_state=111)

svc_model.fit(X_train, y_train)

svc_prediction = svc_model.predict(X_test)

svcConfusionMatrix = confusion_matrix(y_test, svc_prediction)

print("SVM Classifier confusion matrix")
print(svcConfusionMatrix)
svcClassificationReport = classification_report(y_true=y_test, y_pred=svc_prediction)

print("SVM Classifier Classification Report")
print(svcClassificationReport)
```

AdaBoost Model

```
dt = DecisionTreeClassifier(max_depth=5, class_weight='balanced', random_state=555)
adaBoost_model = AdaBoostClassifier(base_estimator=dt, learning_rate=0.001, n_estimators=1
000, random_state=222)
adaBoost_model.fit(X_train ,y_train)

adaBoostPrediction = adaBoost_model.predict(X_test)
adaBoostConfusionMatrix = confusion_matrix(y_test, adaBoostPrediction)
print("AdaBoost confusion matrix")
print(adaBoostConfusionMatrix)
adaBoostClassificationReport = classification_report(y_true=y_test, y_pred=adaBoostPrediction
)
print("AdaBoost Classification Report")
print(adaBoostClassificationReport)
```

9.1 Creating visualizations from the model results

```
# Creating a confusion matrix heatmap for the logistic regression model
plt.figure(figsize=(10, 6))
sns.heatmap(logConfusionMatrix, annot=True, fmt='d', cmap='Blues',
          xticklabels=logRegModel.classes_, yticklabels=logRegModel.classes_)
plt.title("Logistic Regression Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# Creating a confusion matrix heatmap for the support vector machine model
plt.figure(figsize=(10, 6))
```

```
sns.heatmap(svcConfusionMatrix, annot=True, fmt='d', cmap='Reds',
            xticklabels=logRegModel.classes_, yticklabels=logRegModel.classes_)
plt.title("SVM Classifier Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# Creating a confusion matrix heatmap for the Adaboost model
plt.figure(figsize=(10, 6))
sns.heatmap(adaBoostConfusionMatrix, annot=True, fmt='d', cmap='Greens',
            xticklabels=logRegModel.classes_, yticklabels=logRegModel.classes_)
plt.title("ADABOOST Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# Creating a Precison-Recall curve for the Logistic regression model
from sklearn.metrics import precision_recall_curve
precision, recall, _ = precision_recall_curve(y_test, logRegModel.predict_proba(X_test)[:, 1])

plt.figure(figsize=(8, 6))
plt.plot(recall, precision, marker='.')
plt.title("Precision-Recall Curve for the logistic regression model")
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.show()

# Creating a Precison-Recall curve for the SVM model
from sklearn.metrics import precision_recall_curve
precision_svm, recall_svm, _ = precision_recall_curve(y_test, svc_model.decision_function(X_t
est))

plt.figure(figsize=(8, 6))
plt.plot(recall_svm, precision_svm, marker='.')
plt.title("Precision-Recall Curve for the SVM classifier model")
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.show()

# Creating a Precison-Recall curve for the Adaboost model
precision_adaboost, recall_adaboost, _ = precision_recall_curve(y_test, adaBoost_model.predict
_proba(X_test)[:, 1])
```

```
plt.figure(figsize=(8, 6))
plt.plot(recall_adaboost, precision_adaboost, marker='.')
plt.title("Precision-Recall Curve for AdaBoost")
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.show()

#Bar plot to compare Model metrics
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

metrics = ['Accuracy', 'Precision', 'Recall', 'F1 Score']
scores_logreg = [logAccuracy, logPrecision, recall_score(y_test, logRegPred), f1_score(y_test, logRegPred)]  # Add F1 score
scores_svc = [accuracy_score(y_test, svc_prediction), precision_score(y_test, svc_prediction), recall_score(y_test, svc_prediction), f1_score(y_test, svc_prediction)]  # Add F1 score
scores_adaboost = [accuracy_score(y_test, adaBoostPrediction), precision_score(y_test, adaBoostPrediction), recall_score(y_test, adaBoostPrediction), f1_score(y_test, adaBoostPrediction)]  # Add F1 score

x = range(len(metrics))

plt.figure(figsize=(10, 6))
plt.bar(x, scores_logreg, width=0.2, label='Logistic Regression')
plt.bar([i + 0.2 for i in x], scores_svc, width=0.2, label='SVM')
plt.bar([i + 0.4 for i in x], scores_adaboost, width=0.2, label='AdaBoost')
plt.xticks([i + 0.2 for i in x], metrics)
plt.xlabel('Metrics')
plt.ylabel('Scores')
plt.title('Comparison of Model Metrics')
plt.legend()
plt.show()
```