International Journal of Data Science and Advanced Analytics (ISSN: 2563-4429)

Prediction of Customer Sentiment Based on Online Reviews Using Machine Learning Algorithms

Chinmayee Guru[1], Walaa Bajnaid[2]
[1]Liverpool John Moores University [2] King Abdulaziz University, Jeddah, Saudi Arabia
[1]Chinmayeeguru24@gmail.com [2]wnbajnaid@kau.edu.sa

Corresponding author email: wnbajnaid@kau.edu.sa
https://doi.org/10.69511/ijdsaa.v5i5.200
Received 15 July 2023; Received in revised form 26 November 2023; Accepted 29 November 2023
Available online 30 November 2023

***Abstract*—Customer opinions and feedback play a pivotal role in enhancing business operations and decision-making processes. Sentiment analysis is a crucial technique used to decipher customer opinions from their feedback and thus provide valuable insights for businesses. However, analysing and understanding reviews is an intricate process and prone to be misleading if not conducted meticulously. This study aims to extract and classify customer emotions from e-commerce reviews of women's clothing in terms of polarity of sentiment, enhancing sentiment analysis accuracy by means of machine learning (ML) classifiers. In addition, the study addresses the challenge of imbalanced data samples. Several supervised ML models, including Naïve Bayes, Random Forest, Support Vector Classifier (SVC) and Extreme Gradient Boosting (XGB), were employed for sentiment analysis. The study also attempts to deal with negations, and pre-processing steps were implemented to reduce the dimensionality and noise of the raw text. In SVC and XGB models, negation handling significantly improved the precision and recall values of minority classes. Moreover, a hybrid class balancing approach and the synthetic minority oversampling technique (SMOTE) were adopted. The findings indicate that the XGB classifier, combined with SMOTE sampling, produced the most accurate results, yielding better F1 and ROC AUC scores.**

***Keywords*— sentiment analysis - E- commerce reviews - Supervised machine learning - Sentiment classification**

1. Introduction

The evolution of internet services and e-commerce has brought substantial benefits to both businesses and customers. Online shopping, in particular, has empowered customers to make online payments and share their feedback through ratings and comments on products. Online reviews are considered as important as word of mouth. They provide a channel for customers to express their satisfaction and offer insights for necessary improvements. However, the e-commerce landscape presents many challenges for both consumers and merchants. Customers may encounter products that fail to meet expectations or fear that items will not fit properly. Merchants, on the other hand, must grapple with issues such as returns, exchanges, refunds, and potential damage to their reputation. However, online reviews have the potential to mitigate these challenges [1], as positive feedback often leads to sales increases, while negative feedback can drive improvements. Thus, understanding and analysing customer feedback is crucial for enhancing the customer journey and product quality. One way to achieve such understanding is by analysing their feedback through sentiment analysis, which delves into the emotional content of customer-written text.

Sentiment analysis leverages advanced tools and techniques to mine customer opinions and gauge their satisfaction. It involves the use of natural language processing (NLP) in conjunction with computer programming and machine learning (ML) models to perform sentiment analysis [2]. However, conducting sentiment analysis requires several phases of data preparation and cleaning before inputting data into ML classifiers. This process is rife with challenges. In this study, we aim to address some of these difficulties, including computationally expensive learning processes due to high dimensionality, the presence of negations in text and imbalanced class distribution in the sample dataset. Our primary goal is to generate a highly accurate model for extracting and classifying customer sentiment from e-commerce reviews. To achieve this, the study will focus on data cleaning without changing the original customer emotion, addressing class imbalances, implementing supervised ML algorithms and validating the chosen models.

2. Materials and Methods

2.1 Sentiment analysis in customer e- reviews

Sentiment analysis in the context of e-commerce is recognized as a reliable method for businesses to make informed judgements. It also allows merchants to gain insights into customer experiences and purchasing journeys [3]. Studies by Raheem et al [4] , have demonstrated that user sentiment analysis can assist businesses in avoiding negative reputation and provide quick and accurate answers to business-related questions. In addition, customer purchasing decisions have

been shown to be impacted by the sentiment factors that appear in other customer reviews.

Regarding the characteristics of sentiment text, research has found that subjectivity in sentences carries more weight than objectivity in reflecting the writer's opinion [5]. Sentiment analysis can be applied at different levels, including the word level, sentence level and document level [6]. Document-level analysis, which captures the holistic emotion expressed, produces the best results [7]. Consequently, recent study have predominantly employed applied sentiment analysis at the document level to identify popular products that attract customers' attention [8]. Sentiment classification involves assigning emotions to text, which can be further analysed to determine the specific emotions that appear, such as excitement or anger [9]. However, in most cases, polarity of emotion (positive, negative or neutral) is used to label text [10].

2.2 **NLP and Sentiment analysis**

Natural language processing (NLP) is a method of analysing unstructured text. NLP techniques are used in automated fashion to pre-process customer reviews, making them suitable for ML models tasked with classifying emotions. These techniques transform the text into a format that ML algorithms can understand [8] . For this purpose, text data need to be cleaned and normalized [11], and noisy information must be removed as well [2]. Converting text into a structured format presents its own set of challenges [12]. In this study, we have employed various NLP methods to prepare text data for ML classifier models.

2.3 **Research Approach**

This study followed a five-step process in which the first step was to prepare and clean the data. A data exploratory analysis was applied to detect patterns and coloration among the predictors and to determine the distribution of the numerical data. Next, unstructured texts were transformed into a comprehensible vector format in which sentences were normalised and stopped, and negative and contraction words were handled. Based on the part of speech, the texts were broken down into tokens and then converted into a dictionary format using NPL techniques. This process was required to ensure that the word machine was readable. Because our data were unbalanced, class balancing techniques were applied after the dataset was divided into a training set and a test set. In the training data stage, the training set was applied to several supervised machine learning models. These trained models were used to analyse the data and obtain the results. Finally, a using the pd.read_csv() method. First, the null values in 845 review text columns were deleted, as well as in 14 columns labelled "Division name", "Department name", and "Class name". After the cleaning process, the total number of records was 22,641. An indepth understanding for data and variables was required to determine if any insights could be generated. Hence, exploratory data analysis (EDA) was applied to investigate dataset. On this stage researchers identify the relationship between variables, significant characteristics, correlations in data, trend, and distribution of data. Python's matplotlib and seaborn libraries used in this study to graphs and visualize trend and connections in data. In addition, the key insights were explored by conducting several analyses including univariate, bivariate, and multivariate confusion matrix was used to evaluate the models see figure 1.

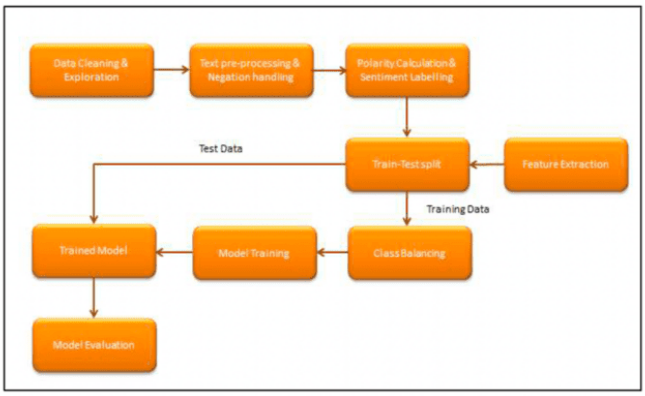

Figure 1Research Process Flowchart

*2.3.1 **Data collection***

The e-commerce data on women's clothing reviews was collect from Kaggle (web scraped dataset)[13] . The dataset contained 23,486 reviews and 10 variables. The company name in each review was anonymised and replaced by "tailer". The columns "Clothing ID" and "Title" were omitted because they were not relevant to the study's objectives. The following table 1 shows the ten variables that comprise the data dictionary.

Table1: Dataset dictionary (Nicapotato, 2018)

Variable	Definition
Clothing ID	Integer Categorical variable that refers to the specific piece being reviewed
Age	Positive Integer variable of the reviewers age
Title	String variable for the title of the review
Review Text	String variable for the review body
Rating	Positive Ordinal Integer for the product score granted by customer -1 Worst, to 5 Best
Recommended IND	Binary variable stating where the customer recommends the product (1) or not(0)
Positive Feedback Count	Positive Integer documenting the number of customers who found this review positive
Division Name	Categorical name of the product high level division
Department Name	Categorical name of the product department name
Class Name	Categorical name of the product class name

2.4 **Analysis and design**

The researcher then defined a new column labelled "sentiment" to categorise the sentiments in the reviews based on the polarity scores "positive, negative, or neutral" using TextBlob, a Python (2 and 3) library for processing textual data. It provides a simple API for performing common natural language processing (NLP) tasks. A pre-trained sentiment classifier, this NLP library may be applied to perform complicated analyses, including lemmatisation, part-of-speech tagging, word inflection, noun phrase extraction, tokenisation, and the integration of WordNet.

2.4.1 ***Data Cleaning and Exploration***

Data mining and filtering are essential in prepare data for processing. Thus, to maintain data consistency, duplicated reviews, incorrect types of data, misspelled words, and missing values were removed from the dataset. The data were imported .

2.4.2. ***Exploratory Data Analyses***

An in-depth understanding of the data and variables was required to determine whether insights could be generated. Hence, an exploratory data analysis (EDA) was applied to investigate the dataset. In this stage, relationships between variables, significant characteristics, correlations in data, trends, and the distribution of data were identified. In this study, Python's Matplotlib and Seaborn libraries were used to develop graphs and visualise trends and connections in the data. In addition, key insights were explored by conducting several analyses, including univariate, bivariate, and multivariate analyses.

Univariate analyses were used to compare ratings and review counts. Five-star ratings appeared in more than half of the reviews, and four-star ratings appeared in most reviews. In contrast, one-star reviews were rare see figure 2. Moreover, the data distribution showed that most of the reviewers were aged between 30 and 40 years. Boxplots are used to show the locations and spreads of variables and reveal symmetry, skewness, and outliers in data. In this study, a boxplot was created to identify outliers in the age and positive feedback count see figure 3, showing that the range of values ranged between 18 and 99 and that the median value was 41. Regarding positive feedback, the minimum value was 0, the maximum was 122, and the median value was 1. Because the data points were continuous, and the range of data was normal and reasonable, no outliners were found, even though some data points had extremely high values.

Bivariate analyses were used to study the relationship between two predictors. A kernel density estimate (KDE) plot was used to visualise the rates of the variable "recommended" by a reviewer. According to hue, the maximum density was 2.5 in five rated reviews that indicated recommendations by the reviewers. However, among the non-recommending reviews, comments showing rates of 2 and 3 were the most common. These results indicated that products with higher ratings were likelier to be recommended by reviewers. In addition, the average positive feedback counts were examined in each rate category. As shown in Figure 4, the counts ranged between 2.4 and 3.5, and no large variations were found.

Multivariate analyses were applied to examine the relationships among several variables. First, the relationship between age and the rate of products was analysed. The results showed that all age groups were consistent and had the same pattern. Five-star ratings were found in most reviews, and one-star ratings were the lowest. Based on these results, the age factor was omitted from the sentiment analysis. Next, the relationships among division, department, and class were analysed. Similarly, the results showed that in each division, department, and class, most ratings were five-star and rarely one-star. Hence, these variables showed the same trend and might not have affected the predicted model. However, the ratings were strongly correlated with the recommended IND. Based on these results, the sentiment analysis was limited to the columns of the review texts.

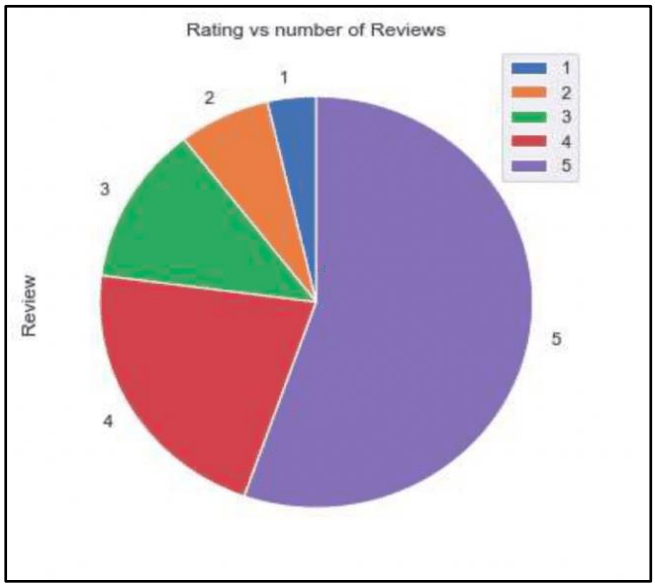

Figure 2 Rating vs Review count

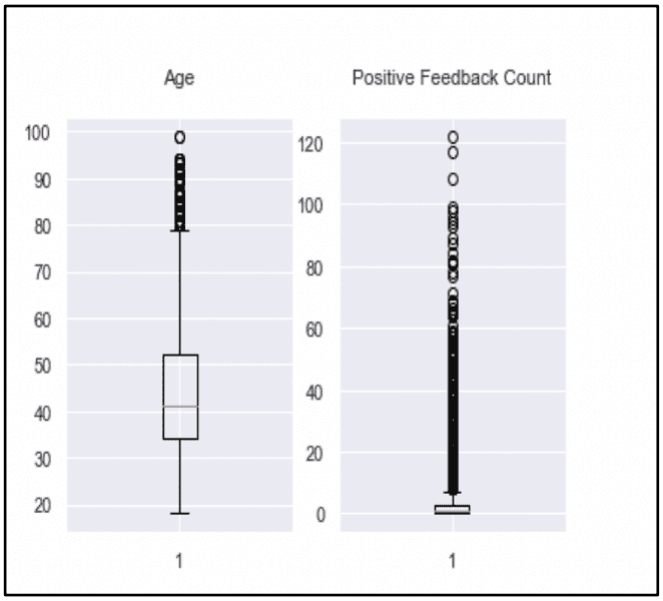

Figure 3 Outlier analysis for Age and Positive Feedback Count

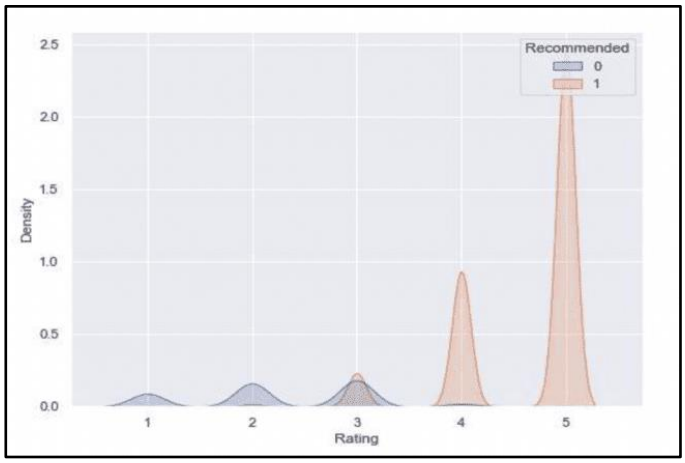

Figure 4 KDE plot for Rating

2.5 **Text Preprocessing**

To process data on machine learning (ML), the reviews on the website were converted to a readable format using NLP techniques to improve ML model performance [14]. The text was passed through several steps. The first step was normalisation, in which all words were changed to lower-case so that words in uppercase would not be differently by NLP. White spaces and numbers were deleted from the texts, followed by handling contractions in customers' comments that contained shortened words or abbreviations. The texts were then split into tokens of words, phrases, and paragraphs, which is known as tokenisation [15]. In this study, the texts were robustly tokenised using the algorithm in the Natural Language Toolkit (NLTK), which is a platform used to build Python programmes that work with human language data for application in statistical NLP. Additionally, punctuation was removed to ensure that the word tokens were weighted equally. Stopwords were then removed from the text, which included pronouns, articles, and prepositions. These words have almost no expressive implications[16]. Therefore, removing stopwords improved the time and speed required to perform the analysis by decreasing the size of the dataset. The NLTK list of stopwords was used in this study. The next step involved the tagging part of speech (POS). The purpose of this step was to help identify distinct components of a sentence based on the functional role of a token within that sentence [17] Finally, in the lemmatisation step, a morphological analysis was performed to break down a word into the vocabulary form known as a "lemma" [18]. In this study, POS-tagged tokens were used to generate lemmas.

2.5 **Negations**

Negations such as "no" or "neither" affect the mining of phrases and expressed emotions. However, in the stopword phase, negations were removed, which led to changes in the meaning of phrases and sentiments. For example, after stopwords were removed, the sentiment "I don't love this item" might be changed to "I love this item". Therefore, negations were managed before the stopwords were removed. Negations may be addressed in several ways, such as by adding prefixes or suffixes and using antonyms instead of tagged words[19]. In this study, Mukherjee et al.'s [20]method was applied to transform negative words into words prefixed by "not".

2.6 **Feature Extraction**

Term frequency-inverse document frequency (TF-IDF) was used to convert words into vectors. TF-IDF is a technique that has proven to be efficient in text classification [21] and has good performance in managing complex, small, and unbalanced data [22]. Arya et al., [6] used TF-IDF to analyse sentiment in Amazon reviews, showing that this technique produced good results in accuracy, F-measures, recall, and precision. In the TF-IDF technique, TF refers to the frequency of terms in a dataset. IDF refers to the ratio of a word to the total number of documents. In this study, a TfidfVectorizer was used to transfer the texts into vectors. The results showed that, based on the assumption that the threshold was 10, there were approximately 3,000 words with frequencies greater than the threshold.

2.7 Class Balancing

Unbalanced data are remedied by adjusting the distribution of data to ensure that they are balanced. In the dataset, 92% of the records were positive, and less than 10% were negative. A few reviews were natural. However, to balance the data, the frequency of the majority class was decreased, and the minority class, or a combination of both classes, was increased [23]. In this study, the dataset was divided into a training set and a test set at a ratio of 70:30. Data balancing was applied only to the training set to ensure reliable performance. In the training set, data up-sampling occurred in the minority classes, which was less than 10% in the training set. The most common problem associated with up-sampling is overfitting[24]. In this study, this problem was avoided by including many positive reviews in the training set. Moreover, the positive reviews were undersampled to decrease gaps in the data distribution. Finally, a combination of the synthetic minority oversampling technique (SMOTE) and random undersampling was used to improve the dataset balance. After the data balancing process, the number of positive reviews decreased to 10,000, and the number of negative reviews increased to 3,500. The number of neutral reviews was increased to 5,000.

2.8 **Sentiment Labelling and Polarity Calculation**

A column was added to classify sentiment in the reviews into positive, negative, or neutral according to text polarity, which was calculated using the TextBlob library as follows:

- If polarity > 0, then Sentiment = positive.
- If polarity = 0, then Sentiment = Neutral.
- If polarity < 0, then Sentiment = Negative.

2.9 **Word Count Analysis**

To build the model and ease the calculation, the word counts were determined. In this dataset, positive reviews had higher word counts compared with negative and neutral reviews. While the word counts in the positive and negative reviews were similar, those in the neutral reviews differed see figure 4.

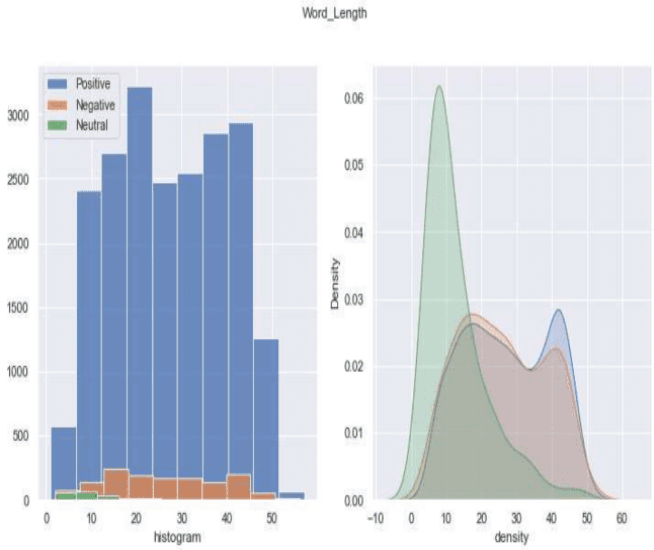

Figure 4Word length distribution plot

2.9 Word Frequency Analysis

The most frequently used words in the sentiments category were determined. Regarding positive sentiments, the words dress, fit, love, length, wear, comfortable, glad, and flatter were the most frequently used. In the negative reviews, the words dress, small, material, little, fit, look, and bust were the most frequently used. Finally, in the neutral reviews, the most frequently used words were dress, look, fabric, size, colour, wear, and return.

2.10 **Model Development and Evaluation**

A supervised ML algorithm was used to classify the data and discover patterns in the dataset. The following well-known effective supervised ML models were used: random forest (RF), naïve Bayes (NB), support vector machine (SVC; LinearSVC), and XGBoost.

RF is an ML algorithm used to create a predictive model using both regression and classification. Thus, it is a meta-estimator [2]. RF has the advantages of accuracy, overfitting reduction, good performance with unbalanced data, effectivity, and reliability. It is also able to automate missing values. However, RF has the limitations of reduced interpretability, high training time consumption, and high computational cost.

NB is an ML classifier that uses a conditional probability model. NB was built on the assumption that predictors do not interact and are unrelated to each other. However, each predictor contributes equally to the outcome. NB is represented as follows:

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$$

NB has a high classification accuracy of 88.17%. It performs well in sentiment analysis and does not require huge amounts of time to train data. Nevertheless, it has some disadvantages, such as independent factors, which are unusual in real time. Moreover, if variables are not observed in the training set, zero probability will be assigned to those that have values in the test data.

SVM is an ML that uses the Scikit-learn library to detect problems in regression and classification. SVM is based on the assumption of a linear hyperplane (Awasthi,2020). The results of SVM are highly accurate (Dey et al., 2020). SVM works only in binary classification. Thus, multiple classifications require division into several binary classifications.

XGBoost is an algorithm that is used to optimise and support fine-tuning parameters. This algorithm is highly accurate and flexible, and it can account for missing values. However, its results are difficult to interpret, which causes overfitting issues and lengthy training times.

To evaluate the prformance of the effectiveness of a model, a confusion matrix was used to compare the predicted values with the actual outcomes . The Confusion Matrix shows a binary classification that is 2X2, were the correct predictions were presented by the green diagonal and the incorrect ones were shown by a red diagonal. A model can be evaluated for its ability to fit the training data and its accuracy in making predictions by identifying errors in the prediction as following: True Positive (TP),False Positive (FP),True Negative (TN) and False Negative (FN) [25]. After the building of a model its performance is assessed by the following scalar values which are generated from the confusion matrix as following; Accuracy, Precision, Sensitivity or Recall and Specificity see table2.

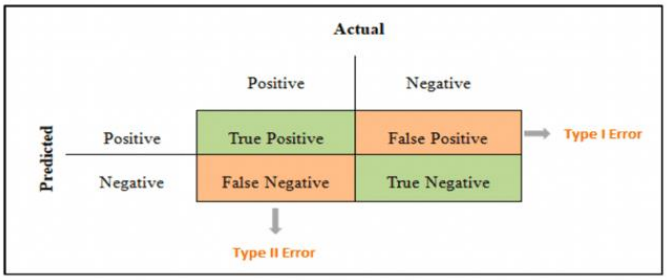

Figure 5 Confusion Matrix

Table 2 Confusion matrix

Accuracy	Precision	Sensitivity or Recall	Specificity
$\frac{TP+TN}{(TP+TN+FP+FN)}$	$\frac{TP}{(TP+FP)}$	$\frac{TP}{(TP+FN)}$	$\frac{TN}{(TN+FP)}$

3. **Results and Discussion**

The current study aim to predict customer sentiments from their online reviews, Hence we proposed model of naive bayes, random forest, support vector classifier, and XGBoost to predict the sentiments. Four prediction rounds were performed on the training set using various sampling approaches. At the first round sample did not adjusted with any method, However, on the second iteration under- sampling strategies were used , following by over-sampling and the last round the hybrid strategies of balancing sampling were used the following are the results of each classifer in each round. Then the accuracy, F1- score and ROC AUC were calculated.

This study aimed to predict customers' sentiments based on their online reviews. The following models were used to predict sentiments: NB, RF, SVC, and XGBoost. Four prediction rounds were performed on the training set using various sampling approaches. In the first round, sampling was not adjusted by any method. However, in the second iteration, undersampling strategies were used, followed by oversampling. In the last round, hybrid balancing strategies were applied. Accuracy, F1 scores, and ROC AUC were calculated.

3.1 **Naïve Bayes**

The multinomial NB method was applied to analyse the training model. Without class balancing, the method accurately predicted 6,293 positive reviews, 2 negative reviews, and 0 neutral reviews. However, 70 positive records were mistakenly predicted as neutral, and 428 positive records were labelled as negative. In the undersampling round, the confusion matrix represented the correct results for 6,292 positive reviews and 3 negative reviews. In the third iteration, the confusion matrix SMOTE showed that 6,210 reviews were positive, 122 were negative, and 7 were neutral. Lastly, we applied a hybrid of the confusion matrix, undersampling, and SMOTE, which produced accurate results showing that 6,066 were positive, 189 were negative, and 12 were neutral. However, the accuracy of the NB classifier was similar in all rounds (see Table 2). SMOTF showed higher ROC AUC scores, and both the SMOTF and hybrid approaches produced F1 scores of 0.92 (see Table 3).

Table 3 Naive Bayes Confusion Matrix

Naïve Bayes		Predicted Label		
		Positive	Neutral	Negative
without Class Balancing	True label	Positive	6293	0	0
		Neutral	70	0	0
		Negative	428	0	2
with Under-Sampling	True label	Positive	6292	0	1
		Neutral	70	0	0
		Negative	427	0	3
with SMOTE	True label	Positive	6210	7	76
		Neutral	59	7	4
		Negative	303	5	122
with Under-Sampling and SMOTE	True label	Positive	6066	17	210
		Neutral	50	12	8
		Negative	236	5	189

Table 4 Naive Bayes Performance Measures

Naïve Bayes	No Balancing	Under-sampling	SMOTE	Hybrid
ACCURACY	0.93	0.93	0.93	0.92
F1-SCORE	0.89	0.89	0.92	0.92
ROC AUC SCORE	0.86588	0.86801	0.89711	0.89340

3.2 **The result of Support Vector Classifier**

LinearSVC was used to train the model with default parameters to classify the types of data balancing. Without class balancing, the SVC confusion matrix accurately predicted 6,257 positive, 228 negative, and 8 neutral classes. In addition, in the undersampling strategy, the performance of SVC changed slightly to 6,229 positive classes, 258 negative classes, and 9 neutral classes. In the SMOTE round, in the positive class, accuracy decreased to 6,170. The number of correct predictions in the neutral and negative classes increased to 19 and 304, respectively. The results of the combined SVC confusion matrix–undersampling and SMOTE model differed slightly, as shown in Table 4. The results for accuracy and the weighted F1 scores were similar for all sample strategies. However, the AUC scores increased marginally in each round (see Table 5).

Table 5. Support Vector Classifier Confusion Matrix

Support Vector Classifier			Predicted Label		
			Positive	Neutral	Negative
without Class Balancing	True label	Positive	6257	0	36
		Neutral	55	8	7
		Negative	202	0	228
with Under-Sampling	True label	Positive	6229	1	63
		Neutral	51	9	10
		Negative	172	0	258
with SMOTE	True label	Positive	6170	17	106
		Neutral	30	19	15
		Negative	120	6	304
with Under-Sampling and SMOTE	True label	Positive	6120	20	153
		Neutral	30	21	19
		Negative	118	6	306

Table7. SVC Performance Measures

SVC	No Balancing	Under-sampling	SMOTE	Hybrid
ACCURACY	0.96	0.96	0.96	0.95
F1-SCORE	0.95	0.95	0.95	0.95
ROC AUC SCORE	0.95809	0.95838	0.95873	0.95921

3.3 **The result of XGBoost**

The results of the confusion matrix of XGBoost showed 6,265 positive classes, six neutral classes, and 63 negative classes in the processing data without class balancing. In the undersampling case, the results showed 6,242 positive classes, 11 neutral classes, and 188 negative classes. The results of SMOTE showed 6,220 positive classes, 49 neutral classes, and 192 negative classes. Finally, the results of the undersampling and SMOTE successfully predicted 6,173 positive classes, 51 neutral classes, and 224 negative classes (see Table 6). XGBoost showed the least accurate results in the no-balancing class, which, in addition to F1 scores, improved slightly in the other samples. However, the results of the SMOTE method showed the highest ROC AUC (see Table 7).

Table 8. XGBoost Confusion Matrix

XGBoost			Predicted Label		
			Positive	Neutral	Negative
without Class Balancing	True label	Positive	6265	3	25
		Neutral	63	6	1
		Negative	292	0	138
with Under-Sampling	True label	Positive	6242	5	46
		Neutral	52	11	7
		Negative	241	1	188
with SMOTE	True label	Positive	6220	23	50
		Neutral	19	49	2
		Negative	224	14	192
with Under-Sampling and SMOTE	True label	Positive	6173	34	86
		Neutral	16	51	3
		Negative	190	16	224

XGBoost	No Balancing	Under-sampling	SMOTE	Hybrid
ACCURACY	0.94	0.95	0.95	0.95
F1-SCORE	0.93	0.94	0.95	0.95
ROC AUC SCORE	0.96641	0.96654	0.96923	0.96761

Table9. XGBoost Performance Measures

*4. **Conclusions***

This study aimed to develop a model for the accurate prediction of sentiment in customer reviews. Data on women's clothing were collected from Kaggle. The proposed model classified positive, negative, and neutral sentiments. The dataset was pre-processed in several steps using different methods to train and improve ML algorithms to reduce dimensionality. The accuracy of the ML classifiers, F1 scores, and ROC AUC were compared, as shown in Table 8. However, the results showed that the accuracy of all classifiers was similar in the different samples. The F1 scores improved in the SMOTE and hybrid sampling strategies. The highest ROC AUC score was shown in XGBoost with hybrid sampling. Therefore, the performance of XGBoost classifier with SMOTE sampling showed the

highest accuracy, the highest F1 score, and the highest ROC AUC score.

This study contributes to the empirical literature by improving the method used to identify sentiments in customer feedback. The results of this study may be used to improve customers' experiences. The class balancing techniques used in this study improved the performance of the model. Unlike most previous studies that have adopted binary classification, we used multiclass classification to conduct unbalanced sampling based on several methods. In future work, we will extend the present study by applying deep learning and transformer models. For example, the results of the present study could be improved by applying a hyperparameter tuning model.

References

[1] G. E. C. Golondrino, M. A. O. Alarcón, and W. Y. C. Muñoz, "Proposal of an Automated Tool for the Application of Sentiment Analysis Techniques in the Context of Marketing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 3, pp. 396–402, 2022.

[2] X. Lin, "Sentiment Analysis of E-commerce Customer Reviews Based on Natural Language Processing," *ACM Int. Conf. Proceeding Ser.*, pp. 32–36, 2020.

[3] S. Yi and X. Liu, "Machine learning based customer sentiment analysis for recommending shoppers, shops based on customers' review," *Complex Intell. Syst.*, vol. 6, no. 3, pp. 621–634, 2020.

[4] M. Raheem, M. Marong, N. K. Batcha, and R. Mafas, "Sentiment Analysis in E-Commerce: A Review on The Techniques and Algorithms," *J. Appl. Technol. Innov.*, vol. 4, no. 1, p. 6, 2020.

[5] P. Mehta and S. Pandya, "A review on sentiment analysis methodologies, practices and applications," *Int. J. Sci. Technol. Res.*, vol. 9, no. 2, pp. 601–609, 2020.

[6] R. K. Arya, M. T. C. Science, and R. Jindal, "N-Gram With Naive Bayes Classifier," vol. 11, no. 6, pp. 749–763, 2020.

[7] S. Behdenna, F. Barigou, and G. Belalem, "Document Level Sentiment Analysis: A survey," *EAI Endorsed Trans. Context. Syst. Appl.*, vol. 4, no. 13, p. 154339, 2018.

[8] A. Zaffar and S. M. A. Hussain, "Modeling and prediction of KSE – 100 index closing based on news sentiments: an applications of machine learning model and ARMA (p, q) model," *Multimed. Tools Appl.*, vol. 81, no. 23, pp. 33311–33333, 2022.

[9] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, pp. 1–19, 2021.

[10] H. Pandita and N. Kumar Gondhi, "A literature survey of sentiment analysis based on E-commerce reviews," *Proc. - 5th Int. Conf. Comput. Methodol. Commun. ICCMC 2021*, no. Iccmc, pp. 1767–1772, 2021.

[11] H. T. Duong and T. A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Comput. Soc. Networks*, vol. 8, no. 1, pp. 1–17, 2021.

[12] J. Guerreiro and P. Rita, "How to predict explicit recommendations in online reviews using text mining and sentiment analysis," *J. Hosp. Tour. Manag.*, vol. 43, no. November 2018, pp. 269–272, 2020.

[13] Nicapotato, "Women's E-Commerce Clothing Reviews," *[online] Kaggle*, 2018. .

[14] P. Poomka, N. Kerdprasop, and K. Kerdprasop, "Machine Learning Versus Deep Learning Performances on the Sentiment Analysis of Product Reviews," *Int. J. Mach. Learn. Comput.*, vol. 11, no. 2, pp. 103–109, 2021.

[15] S. Dey, S. Wasif, D. S. Tonmoy, S. Sultana, J. Sarkar, and M. Dey, "A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews," *2020 Int. Conf. Contemp. Comput. Appl. IC3A 2020*, no. February, pp. 217–220, 2020.

[16] K. Zahoor, N. Z. Bawany, and S. Hamid, "Sentiment analysis and classification of restaurant reviews using machine learning," *Proc. - 2020 21st Int. Arab Conf. Inf. Technol. ACIT 2020*, no. November, 2020.

[17] M. Zaffar *et al.*, "A hybrid feature selection framework for predicting students performance," *Comput. Mater. Contin.*, vol. 70, no. 1, pp. 1893–1920, 2021.

[18] K. Divya, B. S. Siddhartha, N. M. Niveditha, and B. M. Divya, "An Interpretation of Lemmatization and Stemming in Natural Language Processing," *J. Univ. Shanghai Sci. Technol.* , vol. 22, no. 10, p. 351, 2020.

[19] R. Patel and K. Passi, "Sentiment Analysis on Twitter Data of World Cup Soccer Tournament Using Machine Learning," *Internet of Things*, vol. 1, no. 2, pp. 218–239, 2020.

[20] P. Mukherjee, Y. Badr, S. Doppalapudi, S. M. Srinivasan, R. S. Sangwan, and R. Sharma, "Effect of Negation in Sentences on Sentiment Analysis and Polarity Detection," *Procedia Comput. Sci.*, vol. 185, no. June, pp. 370–379, 2021.

[21] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on the sentiment analysis," *Procedia Comput. Sci.*, vol. 152, no. July, pp. 341–348, 2019.

[22] C. Padurariu and M. E. Breaban, "Dealing with data imbalance in text classification," *Procedia Comput. Sci.*, vol. 159, pp. 736–745, 2019.

[23] B. K. Shah, A. K. Jaiswal, A. Shroff, A. K. Dixit, O. N. Kushwaha, and N. K. Shah, "Sentiments Detection for Amazon Product Review," *2021 Int. Conf. Comput. Commun. Informatics, ICCCI 2021*, 2021.

[24] Q. Zheng, C. Yang, Li‹Shundong, and F. Li, *E-Commerce Strategy*. Springer, 2014.

[25] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informatics*, vol. 17, no. 1, pp. 168–192, 2018.