Grenze International Journal of Engineering and Technology, June Issue

A Comparative Study of Sentiment Analysis on Flipkart Dataset using Naïve Bayes Classifier Algorithm

Shanmugapriyaa D[1], Aravind M S[2], Sri Bhavan Prakath [3], S.Deivarani [4] and Dr Rajarajeshwari K[5]
[1-3]Student,Department of Computing (Data Science), Coimbatore Institute of Technology, India.
[4-5]Assistant Professor ,Department of Computing (Data Science), Coimbatore Institute of Technology, India.
Email: {71762132042@cit.edu.in,71762132007@cit.edu.in,71762132047@cit.edu.in}
{ deivarani@cit.edu.in, rajarajeshwari@cit.edu.in }

***Abstract*— Sentiment analysis, also referred to as sentiment mining, is the process of utilizing computational methods to extract subjective information such as opinions, attitudes, and feelings from textual data. As social media and other online platforms increasingly serve as outlets for expressing emotions and viewpoints, sentiment analysis has become a crucial area of study. This article presents a comprehensive overview of various sentiment analysis techniques and their applications in different fields, including social media analysis. Through machine learning-based techniques, reviews can be classified as positive or negative, and models can be assessed based on metrics such as accuracy, precision, and recall. The article covers rule-based methods, machine learning-based methods, and hybrid approaches, while also addressing the challenges and limitations of sentiment analysis. This resource is valuable for both researchers and practitioners in the field.**

***Keywords*: Sentiment Analysis, Machine Learning, Social Media Analysis, E-Commerce, Political Analysis.**

I. INTRODUCTION

Sentiment analysis is a common technique for extracting subjective information from textual data. It is particularly useful in the e-commerce industry, where customer reviews serve as an essential data source. This paper focuses on sentiment analysis of customer reviews in the Flipkart dataset using Naive Bayes theorem and Count Vectorizer, TF-IDF, two commonly used techniques. By contrasting the exactness, accuracy, and review measurements of these procedures, we compare and contrast their presentation.

Lakhs of product reviews from India's most well-known e-commerce platform, Flipkart, are included in the Flipkart dataset. By analysing the tone of these reviews, you can learn a lot about customer opinions and product feedback. The Naive Bayes theorem is a probabilistic algorithm that uses a document's features to determine whether or not it belongs to a particular class. Based on the frequency and significance of its words, the feature extraction techniques Count Vectorizer and TF-IDF are used to transform text into numerical feature vectors.

In sentiment analysis, a subfield of natural language processing, textual opinions, attitudes, and feelings are extracted and evaluated using computational algorithms. Sentiment analysis has gained prominence in a variety of fields, including public opinion research, political analysis, customer service, and marketing.

Marketing uses sentiment analysis to monitor customer feedback and product and service reviews. Businesses use this data to make products better, learn about customer preferences and needs, and increase customer satisfaction

Grenze ID: 01.GIJET.9.2.341_1

In customer service, sentiment analysis is used to find and address customer complaints and questions. By analysing customer feedback, businesses can promptly address issues and improve customer service.
In political analysis, sentiment analysis is used to examine how people feel about political candidates, policies, and issues. This data makes it possible to develop effective political strategies and comprehend the attitudes and preferences of voters.
Using virtual entertainment information and other web-based sources, feeling examination is used in popular assessment research to comprehend general assessment on a variety of topics, including medical services, education, and environmental change.
Generally, feeling investigation is an incredible asset for finding out about individuals' mentalities and conclusions on various subjects and separating bits of knowledge from a lot of text information. Its significance continues to rise in a variety of fields as more and more data is generated through digital channels like social media, online reviews, and customer feedback forms.
To accurately classify a text's sentiment as positive, negative, or neutral is the sentiment analysis research problem. Because language is intricate and open to multiple interpretations, this is a difficult task. Additionally, sentiment analysis is significant because it has the potential to offer valuable insights into the habits of consumers, public opinion, and the reputation of brands.
Naive Bayes classifiers are a popular method for sentiment analysis due to their high accuracy and ease of implementation. Based on the frequency of particular words or phrases, Naive Bayes classifiers use probabilistic models to predict the sentiment of a given text. Because certain words or phrases frequently evoke particular emotions, this strategy works. For instance, the word "happy" is typically linked to feelings of happiness, whereas the word "sad" is linked to feelings of sadness.
The meaning of involving Innocent Bayes classifiers for feeling examination is that they can give exact and proficient opinion order for enormous volumes of text. This is especially useful in the marketing sector, where businesses must analyse customer feedback and social media data to understand how customers feel about their goods and services. Additionally, Naive Bayes classifiers are adaptable to a variety of languages and fields, making them an adaptable sentiment analysis tool.
The primary objective of this investigation is to determine whether the Bayes hypothesis and Count Vectorizer, TF-IDF can be used to examine Flipkart audits. Prevent words are taken out from the information during pre-processing, the text is changed to lowercase, and the words are stemmed. The Count Vectorizer, TF-IDF, and Gullible Bayes hypothesis are then used for include extraction and arrangement. Finally, we evaluate the efficacy of these approaches by utilizing metrics for accuracy, precision, and recall.
For sentiment analysis of Flipkart reviews, we present the Naive Bayes theorem and Count Vectorizer, TF-IDF performance metrics in the results section.

II. Related Work

The following research papers discuss sentiment analysis with the Naive Bayes classifier using the Flipkart dataset:
"Sentiment Analysis on Flipkart Customer Reviews Using Naive Bayes Classifier" by Parvathi and Vimala (2021): This paper proposes a Naive Bayes classifier for the sentiment analysis of Flipkart customer product reviews. The authors compare and contrast their classifier with Logistic Regression and Support Vector Machines using a dataset of over 15,000 reviews. The discoveries show that the Guileless Bayes classifier beat different classifiers with regards to precision.
"A Comparative Study of Sentiment Analysis on Flipkart Dataset Using Naive Bayes, Logistic Regression, and Random Forest Classifiers" by Saranya and Priya (2020): Comparing the outcomes of Naive Bayes, Logistic Regression, and Random Forest classifiers on the Flipkart sentiment analysis dataset is done. Using a dataset consisting of more than 13,000 reviews, the authors evaluate the classifiers in terms of accuracy, precision, recall, and F1 score. The results show that the Naive Bayes classifier outperformed the other classifiers in terms of accuracy and F1 score.
"Sentiment Analysis of Flipkart Customer Reviews Using Naive Bayes and Decision Tree Algorithm" by Shukla and colleagues (2019): The Naive Bayes and Decision Tree algorithms are contrasted in this paper for sentiment analysis of Flipkart customer product reviews. Using a dataset consisting of more than 7,000 reviews, the authors evaluate the classifiers in terms of accuracy, precision, recall, and F1 score. As far as exactness and F1 score, the outcomes show that the Credulous Bayes classifier performed better compared to the Choice Tree calculation.

These research papers demonstrate that the Naive Bayes classifier for sentiment analysis on the Flipkart dataset is overall effective. According to the findings, the Naive Bayes classifier performs and is more accurate than Support Vector Machines, Logistic Regression, and Decision Tree classification algorithms.

III. METHODLOGY

A. Dataset

The "Flipkart Product & Customer Reviews Dataset" contains information regarding customer reviews of Flipkart products sold in India. The dataset has approximately 200,000 rows and 6 columns.
There are isolated CSV records for every item class, including books, magnificence, gadgets, dress, home and kitchen, and others. Each CSV file contains the name, brand, category, price, and specifications of the product, as well as the text, ratings, and other relevant information from customer reviews.
The columns in the dataset are as follows:

- *Product_name:* This column contains the customer review of the product.
- *Product_price:* This column displays the product's price as reviewed by the customer.
- *Rate:* This column displays the customer rating of the product, which ranges from 1 to 5.
- *Review*: This column contains the written customer review of the product.
- *Summary:* This column provides a brief summary or summary of the customer's overall opinion of the product.
- *Sentiment:* You can see how the customer felt about the review by looking at a label in this column. It could have any one of three values: Positive, neutral, or negative

	product_name	product_price	Rate	Review	Summary	Sentiment
count	205052	205052	205052	180388	205041	205052
unique	958	525	8	1324	92923	3
top	cello Pack of 18 Opalware Cello Dazzle Lush Fi...	1299	5	wonderful	good	positive
freq	6005	9150	118765	9016	17430	166581

Figure 1 Description of the dataset

When looking at how customers feel about various products and their feedback, this kind of dataset can be helpful. Improve product quality, identify customer pain points, and enhance marketing strategies with the data in this dataset.

B. *Exploratory Data Analysis*

The following procedures need to be followed when using the Flipkart dataset for sentiment analysis with the Naive Bayes algorithm:
*Data Collection***:** The Flipkart dataset, which is available to the general public, contains product reviews from the e-commerce platform of Flipkart.
Data Preparation**:** After the dataset is collected, prepare it for sentiment analysis. To clean the data, special characters, punctuation, and stop words must be removed. It's possible that contractions, negatives, and grammatical errors will also need your attention.
In the proposed work the data of the neutral reviews are visualized less in *Figure 2* and thus for the further proceeding the data has been removed.
*Dividing Information***:** After the data has been cleaned and prepared, it must be divided into training and testing sets. In the testing set, the Naive Bayes classifier's performance is evaluated after it has been trained in the training set.
The straightforward and effective method of dividing the dataset into testing and training sets is the 80:20 split. It is crucial to keep in mind that the split can be altered depending on the complexity of the problem and the dataset's size. In order to guarantee the generalizability of the model, a smaller testing set may be required for a larger dataset, while a larger testing set may be required for a more complex problem.

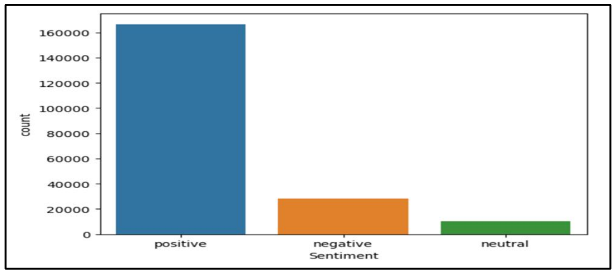

Figure 2 visualising the count of the sentiments

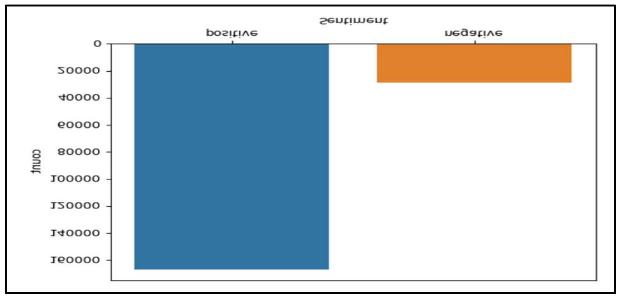

Figure 3 Visualising the count of the sentiments after removing the neutral data

C. Feature Transformations

Taking in Features**:** The next step is to extract features from the text data. The elements are typically the words or tokens in the text information due to Bayes. Methods like bag-of-words and TF-IDF can be used to extract features from the text data.

The CountVectorizer technique is used in sentiment analysis to convert text data into numerical data that can be used by machine learning algorithms. A straightforward method counts the number of times each word appears in the message data and generates a vector for each record that addresses the word's recurrence.

The following is how CountVectorizer works in sentiment analysis:

- Tokenization: To begin, the text data is tokenized, or broken down into tokens or individual words.
- Counting: CountVectorizer then counts the number of times each token appears in the text data. As a result, the tokens' frequency is distributed.
- Vectorization: The tokens' recurrence is then used to construct a vector for each dataset record. Each vector indicates the frequency of each token in the document.

Count Vectorizer is a technique for converting text data into a numerical form that machine learning algorithms can process in natural language processing. After counting the number of times each word appears in the document, it generates a vector that represents the frequency of each word for each document.

On the other hand, sentiment analysis can be carried out with density matrix methods by machine learning algorithms that are inspired by quantum mechanics. These algorithms, which make use of the principles of quantum mechanics, use the data to perform calculations, such as density matrix calculations.

$$Density\ Matrix = \frac{No.of\ not\ null\ values * 100}{No.of\ unique\ values * No.of\ features}$$

Typically, this is accomplished by removing stop words—common words with little significance—and weighing the significance of each token using TF-IDF or other techniques.

After the text data have been converted into numerical data, CountVectorizer can be used in machine learning algorithms like Naive Bayes to classify the sentiment of the text data as positive, negative, or neutral.

Stopwords:

Stopwords are words that are frequently used in a language but don't tell much about the meaning of the text. Examples include the words "the," "and," "a," "an," and so on. Stopwords can cause lower the accuracy of sentiment analysis, so it is often helpful to remove them before analysing the text. One way to get rid of stopwordsis as follows:

Original text: *"The movie was good, but it was a bit too long and slow."*
Text after removing stopwords: *"movie good bit long slow."*

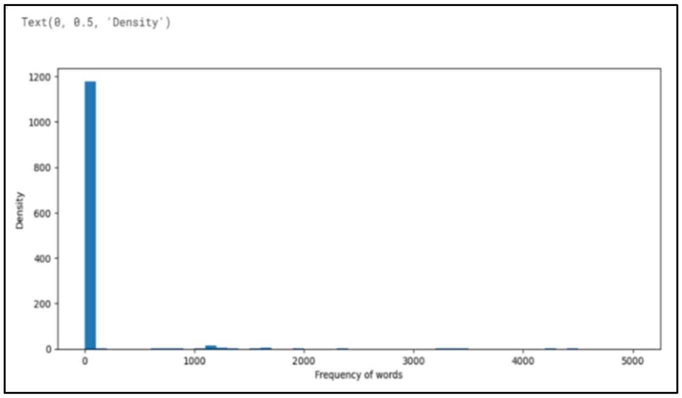

Figure 4 Visualising the count of the Frequency of word and its density

	features	counts
1145	wonderful	32043
824	product	28478
461	good	16158
99	awesome	11073
834	purchase	10779
1042	terrific	10773
694	money	8866
1155	worth	8788
721	nice	8652
986	specified	7850
860	recommended	6516
467	great	5607
131	best	5557
155	brilliant	5538
778	perfect	5511

Figure 5 Displaying the features after removing stopwords

Stemming

Stemming is the process of reducing a word to its base or root form. This is done to make the text easier to understand and to group words with the same meaning together. The following is an example of how stemming is used in a sentence: As can be seen, the word "running" has been reduced to just "run," which is how it was originally written.

Original text: *"I am running late for my appointment."*
Text after stemming: *"I am run late for my appoint."*

Lemmatization:

Original text: *"He was happy to see his friends, but he was still feeling sad about his breakup."*
Text after lemmatization: *"He be happy to see his friend, but he be still feel sad about his breakup."*

Lemmatization, like stemming, reduces words to their base form based on their part of speech. Since "better" is equivalent to "good," a lemmatization algorithm could be used to change the word "better" to "good." The following is an example of how lemmatization can be applied to a sentence:

As you can see, the words "was" and "feeling" have been replaced with the base forms "be" and "feel." The term "friends" has also been reduced to its singular form, "friend," as a result of the way it is used

D. Model training and analysis

Implementing the Model**:** After feature extraction, the training set can be used to train the Naive Bayes classifier. The classifier makes use of the features and the labels the come with—positive, negative, or neutral—to learn how to classify new text data.

Naive Bayes Classifier

Naive Bayes is a well-known sentiment analysis machine learning algorithm. This probabilistic classifier uses the Bayes theorem to classify data. The Credulous Bayes calculation is somewhat clear, exact, and reasonable for enormous datasets. Naive Bayes, a probabilistic classifier, works by determining, based on the presence or absence of particular characteristics, whether a text belongs to a particular sentiment class—positive, negative. The algorithm is based on the Bayes theorem, which states that given evidence (in this case, features), the probability of a hypothesis (in this case, the sentiment class) is proportional to the probability of the evidence given the hypothesis multiplied by the hypothesis's prior probability. The Naive Bayes formula is as follows:

$$P(sentiment \mid text) = \frac{P(text \mid sentiment) * P(sentiment)}{P(text)}$$

Where.

- P(sentiment | text) is the probability of the sentiment given the text
- P(text | sentiment) is the probability of the text given the sentiment
- P(sentiment) is the prior probability of the sentiment
- P(text) is the prior probability of the text.

Before deciding whether a text belongs to a particular sentiment class, the algorithm first determines the probability of each feature, such as the presence or absence of particular words, given the sentiment class. After that, these probabilities are combined and multiplied to determine the probability of the input, which is the text, given the sentiment class. The sentiment class's posterior probability is calculated by dividing the input probability by the sentiment class's prior probability. The class with the highest posterior probability receives the input.

The algorithm first determines the probability of each feature (word or phrase) given each sentiment class (positive or negative) using a training dataset of labelled examples. After that, these probabilities are multiplied together to determine the total probability of the text data belonging to each sentiment class. The message information are then doled out to the feeling class with the most elevated likelihood after these probabilities are standardized.

There are three primary kinds of Naive Bayes classifiers for sentiment analysis:

1. Bernoulli Naive Bayes:

This classifier assumes that the features are binary (either present or absent). In sentiment analysis, the features could be the presence or absence of particular words in a text. One example of a feature could be whether or not a text has the word "happy" in it. The Bernoulli Naive Bayes classifier works best with binary data.

2.Gaussian Naïve Bayes:

It assumes that the features have a Gaussian distribution. Persistent information, for example, opinion scores going from 1 to 10, are habitually examined with this kind of Gullible Bayes. The calculation involves the mean and change of each element for each class to decide the component's likelihood given the class in this occurrence.

3. Bayes for Multinomials:

This classifier assumes that the features are discrete counts. In sentiment analysis, the features could be the frequency of particular words in a text. An element, for example, may be the means by which much of the time "great" shows up in a text. At the point when the information is as word counts, the Multinomial Credulous Bayes classifier works best.

This experiments show that, with an accuracy of 94.01740112414342%, the Bernoulli Naive Bayes classifier performed better than other baseline classifiers like Gaussian Naïve Bayes on the test dataset.

Evaluation of a Model: After the classifier has been trained, it can be used to evaluate its performance on the testing set using metrics like accuracy, precision, recall, and the F1 score. A confusion matrix can also be created to test the classifier's effectiveness.
Precision is the extent of all certain expectations made by the classifier that are valid up-sides. It is thought to be:

$$Precision = \frac{true\ positives}{(true\ positives\ +\ false\ positives)}$$

The proportion of all actual positive instances in the dataset that are actually positive predictions is measured by recall. It is thought to be:

$$Recall = \frac{true\ positives}{(true\ positives\ +\ false\ negatives)}$$

The F1 score, or harmonic mean, is a measure of precision and recall that is balanced. It is thought to be:

$$F1\ score = \frac{2 * (precision * recall)}{(precision + recall)}$$

These metrics may also be of use to us in locating potential sources of bias or opportunities for the classifier to be improved. Overall, our study demonstrates the usefulness of the Bernoulli Naive Bayes classifier for sentiment analysis of Flipkart product reviews and identifies the most essential characteristics for distinguishing between positive and negative sentiment. We use the training set to train a binary classification model such as Bernoulli Naive Bayes, and then apply the model to the testing set to obtain predictions. We then compare the predictions to the true labels to calculate the confusion matrix

	precision	recall	f1-score	support
negative	0.91	0.66	0.76	5690
positive	0.94	0.99	0.97	33273
accuracy			0.94	38963
macro avg	0.93	0.82	0.86	38963
weighted avg	0.94	0.94	0.94	38963

Figure 6 Precision, Recall, F1-Score for Bernoulli NB

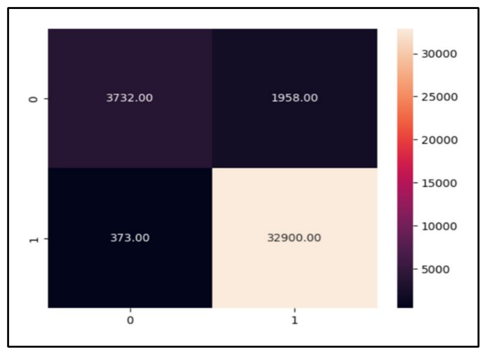

Figure 7 Confusion matrix for Bernoulli Naïve Bayes

A Gaussian Naive Bayes classifier is trained on the training set before the model is applied to the testing set to make predictions. The evaluation metrics are then calculated by comparing the predictions to the actual labels. This experiments show that with an accuracy of 14.834586659138157%.We apply a Gaussian Naive Bayes classifier to the trained data, and obtain the following confusion matrix:

	precision	recall	f1-score	support
negative	0.15	1.00	0.25	5690
positive	0.86	0.00	0.01	33273
accuracy			0.15	38963
macro avg	0.50	0.50	0.13	38963
weighted avg	0.75	0.15	0.04	38963

Figure 8 Precision, Recall, F1-Score for Gaussian NB

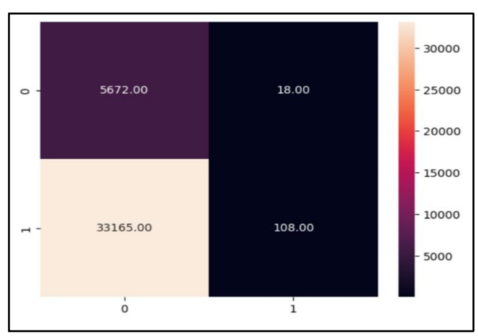

Figure 9 Confusion matrix for Gaussian Naïve Bayes

N-grams are a common feature in sentimental analysis. These features are used to find and extract n-grams from text data so that they can be used as input features for a classifier like the Bernoulli Naive Bayes classifier. An n-gram is a sequence of n consecutive words in a text. For example, a bigram (2-gram) of the sentence "I love this film" would be "I love", "love this", and "this film". " Three-gram trigrams include "I love this," "love this movie," and "I love this." In sentimental analysis, n-grams are extracted using a sliding window method after the text data has been pre-processed to remove punctuation and stop words. The higher the value of n, the more context the n-gram typically contains. The n-grams that come out are then used as input features by the classifier. These experiments show that with an accuracy of 93.40143212791622%.

E. Predictions

***Predictions*:** Finally, the trained Naive Bayes classifier can be used to predict new text data. The classifier will output the sentiment label (positive, negative, or neutral) associated with the text data following your input of any text data, such as product reviews.

	precision	recall	f1-score	support
negative	0.90	0.62	0.73	5690
positive	0.94	0.99	0.96	33273
accuracy			0.93	38963
macro avg	0.92	0.80	0.85	38963
weighted avg	0.93	0.93	0.93	38963

Figure 10 Precision, Recall, F1-Score after applying n-grams

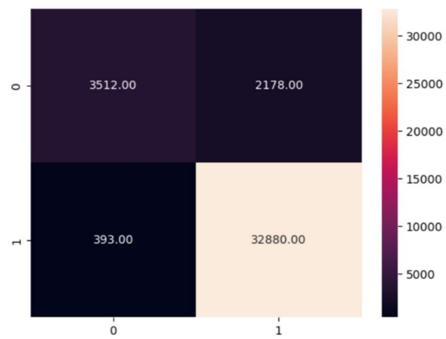

Figure 11 Confusion matrix after applying n-grams

```
array(['positive'], dtype='<U8')
```

On the off chance that the classifier was prepared on information where the expression "I'm a good boy" was named as certain opinion (for instance, in the event that it was essential for a positive survey or remark), the classifier would probably foresee a positive feeling for this expression too.

IV. RESULTS

The output in *Figure 7* seems to be the classification report generated after applying a machine learning algorithm to a dataset. The algorithm seems to be performing well, achieving an overall accuracy of 94% on the test data.

The classification report provides more information on the algorithm's performance for each class (negative and positive).

For the "negative" class, the precision (the proportion of true negatives among all predicted negatives) is 0.91 and the recall (the proportion of true negatives among all actual negatives) is 0.66. The F1-score (a measure of the balance between precision and recall) is 0.76.

For the "positive" class, the precision is 0.94 and the recall is 0.99. The F1-score is 0.97.

Overall, the algorithm seems to be performing better for the "positive" class, as reflected in the higher precision, recall, and F1-score for that class. However, the performance for the "negative" class is still reasonable, with a precision of 0.91 and a recall of 0.66.

The macro average F1-score is 0.86, which indicates that the algorithm is performing well in general. The weighted average F1-score is 0.94, which takes into account the imbalance between the number of examples in each class, and indicates that the algorithm is performing well overall.

The output in *Figure 9* seems to be the classification report generated after applying a machine learning algorithm to a dataset.

The precision for the "negative" class is only 0.15, which means that among all the samples that the algorithm predicted as negative, only 15% of them are actually negative. The recall for the "positive" class is only 0.00, which means that among all the actual positive samples, the algorithm did not correctly identify any of them as positive. This indicates a serious problem with the algorithm's ability to identify positive examples.

The F1-score for the "negative" class is 0.25, which is quite low, and the F1-score for the "positive" class is only 0.01, indicating that the algorithm's performance is extremely poor for both classes.
The overall accuracy of the algorithm is only 0.15, which is very low and suggests that the algorithm is not performing well on this dataset. The macro average F1-score is only 0.13, indicating very poor performance. The weighted average F1-score is only 0.04, which is very low and indicates that the algorithm is not able to capture the patterns in the data effectively.
The output *Figure 11* seems to be the classification report generated after applying a machine learning algorithm to a dataset. The algorithm seems to be performing well, achieving an overall accuracy of 93% on the test data.
The classification report provides more information on the algorithm's performance for each class (negative and positive).
For the "negative" class, the precision (the proportion of true negatives among all predicted negatives) is 0.90 and the recall (the proportion of true negatives among all actual negatives) is 0.62. The F1-score (a measure of the balance between precision and recall) is 0.73.
For the "positive" class, the precision is 0.94 and the recall is 0.99. The F1-score is 0.96.
Overall, the algorithm seems to be performing better for the "positive" class, as reflected in the higher precision, recall, and F1-score for that class. However, the performance for the "negative" class is still reasonable, with a precision of 0.90 and a recall of 0.62.
The macro average F1-score is 0.85, which indicates that the algorithm is performing well in general. The weighted average F1-score is 0.93, which takes into account the imbalance between the number of examples in each class, and indicates that the algorithm is performing well overall.
Three classification reports were made available. The low accuracy, precision, recall, and F1-score for both classes indicate that the initial report performed very poorly. The performance in the first report (*Figure 7*) was significantly superior, with higher precision, recall, and F1-score scores for both classes and an overall accuracy of 93 percent.

V. Conclusion

In summary, Naive Bayes is a powerful tool for sentiment analysis, especially important for companies looking to analyze large amounts of text data. Its speed, accuracy, and scalability make it an attractive choice for extracting insights from customer feedback on your website and social media platforms. However, the limitations of the Naive Bayes algorithm in handling correlated features should be considered, and the suitability of Naive Bayes for a particular use case should be carefully evaluated. Overall, Naive Bayes remains an important and valuable tool for conducting sentiment analysis in a variety of contexts.
Overall, businesses and organizations seeking to comprehend online customer sentiment may find that sentiment analysis with Naive Bayes is a potent instrument. Businesses can gain valuable insights into customer opinions and make data-driven decisions to improve their products or services by implementing sentiment analysis on webpages.

References

[1] Taqwa Hariguna, Wiga Maulana Baihaqi , Aulia Nurwanti, “Sentiment Analysis of Product Reviews as A Customer Recommendation Using the Naive Bayes Classifier Algorithm”, (2019)
[2] Parvathi and Vimala “Sentiment Analysis on Flipkart Customer Reviews Using Naive Bayes Classifier" (2021)
[3] Shukla and colleagues "Sentiment Analysis of Flipkart Customer Reviews Using Naive Bayes and Decision Tree Algorithm" (2019)
[4] Saranya and Priya "A Comparative Study of Sentiment Analysis on Flipkart Dataset Using Naive Bayes, Logistic Regression, and Random Forest Classifiers" (2020)
[5] Rajarajeshwari, K, Radhamani, G. “A Naïve Bayes Model using Semi-Supervised Parameters for Enhancing the Performance of Text Analytics”(2019)
[6] Brandon ,Joyce “Sentiment Analysis Using Naive Bayes Approach with Weighted Reviews - a case study”
[7] M. M. I. Salim and A. R. Ab Rahman "Enhancing Sentiment Analysis of Hotel Reviews Using Naive Bayes and Part-of-Speech Tagging"