SENTIMENT ANALYSIS ON AMAZON REVIEWS

NATURAL LANGUAGE PROCESSING PROJECT

GHADEER ALHARBI

Abstract

In this technological age, we are all familiar with online shopping and the feedback associated with it. Any business needs to keep track of customer reviews, ratings, and feedback in order to understand what their views are on the product. In order to keep up with customer needs, advice, and corrections, reviews are the only way to keep the product up to date. There are millions of reviews associated with a single type of product that a company sells, and the company needs to employ a proper channel for working with it, so that they can filter out the information about how the product performs on the market or what a customer thinks about it. This paper focuses on the use of sentiment analysis process models to determines whether the sentiment expressed is positive, negative, or neutral. Two models were utilized for this purpose, namely Textblob tool and Naive Bayes Multinomial algorithms. The performance of both models was evaluated using accuracy. The results indicate that the performance of Naive Bayes algorithm is outperformed the Textblob.

Table of Contents

List of Tables

List of Figures

Table of Equations

Chapter 1: Overview

1.1 Introduction

Natural language processing (NLP) for the computational representation and analysis of human language has received much attention. It is important for businesses that deal with large amounts of unstructured text, whether emails, social media conversations, online chats, survey responses, and many other forms of data. It has expanded its applications into areas such as sentiment analysis, machine translation, data mining, spam detection, medicine, abstraction, and query response. NLP is a subfield of artificial intelligence and linguistics that tries to assist computers in recognizing written words or phrases in human speech. It is intended to make the user's life easier by allowing them to speak with the computer in natural language. These are broken into two parts: learning one or more natural languages and developing a natural language that facilitates text comprehension and creation. Linguistics is the study of language, which encompasses phonology, morphology of word structure, syntax of sentence structure, lexical semantics, and pragmatics of comprehension [1].

Every day we come across various products in our lives, on the digital medium we swipe across hundreds of product choices under one category. It will be tedious for the customer to make selection. Here comes 'reviews' where customers who have already got that product leave a rating after using them and brief their experience by giving reviews. As we know ratings can be easily sorted and judged whether a product is good or bad. But when it comes to sentence reviews, we need to read through every line to make sure the review conveys a positive or negative sense that it comes the role of sentiment analysis. Sentiment analysis is a widely used text classification tool that examines a message and determines whether the sentiment expressed is positive, negative, or neutral. It's crucial for businesses to comprehend people's emotions because customers can now express their thoughts and feelings more openly than ever before. It is challenging for humans to manually evaluate each line and identify the emotion behind the users' experiences. With the help of technology, however, customer feedback can now be automatically analyzed, from survey responses to social media conversations. This way, brands can listen attentively to their customers and customize their products and services to meet their needs. However, the test data used to train these machine learning models often contains highly imbalanced classes, which renders them unsuitable for training [2].

1.2 Issues

The main problem addressed in sentiment analysis of Amazon reviews using NLP techniques is how to accurately classify the sentiment of user-generated content on the platform, which can be a mixture of positive, negative, and neutral opinions. Another problem that can be addressed the presence of noisy or irrelevant text data, such as product names, brand mentions, stop wards, punctuations, or unrelated comments, which can affect the accuracy of sentiment classification. The objective in this case is to develop NLP techniques and preprocessing methods using python that can effectively filter out irrelevant text data and improve the accuracy of sentiment analysis results.

1.3 Goal of project

The goal of the project is to develop machine learning models (NLP model) using python that can effectively analyze large volumes of text data and accurately classify reviews based on their sentiment (positive, negative, neutral), thereby providing valuable insights to businesses about customer opinions and preferences. Another objective is to explore and compare different NLP techniques and machine learning algorithms to identify the most effective approach for sentiment analysis of Amazon reviews. Ultimately, sentiment analysis used to help businesses make data driven decisions and improve their products and services based on customer feedback.
Our approach aims to create and compare a hybrid machine learning model (NLP model) that achieves a balanced and impartial classification of reviews. The flowchart shown in Figure 1.1 outlines our proposed approach [3] and all contribution in this project by me.

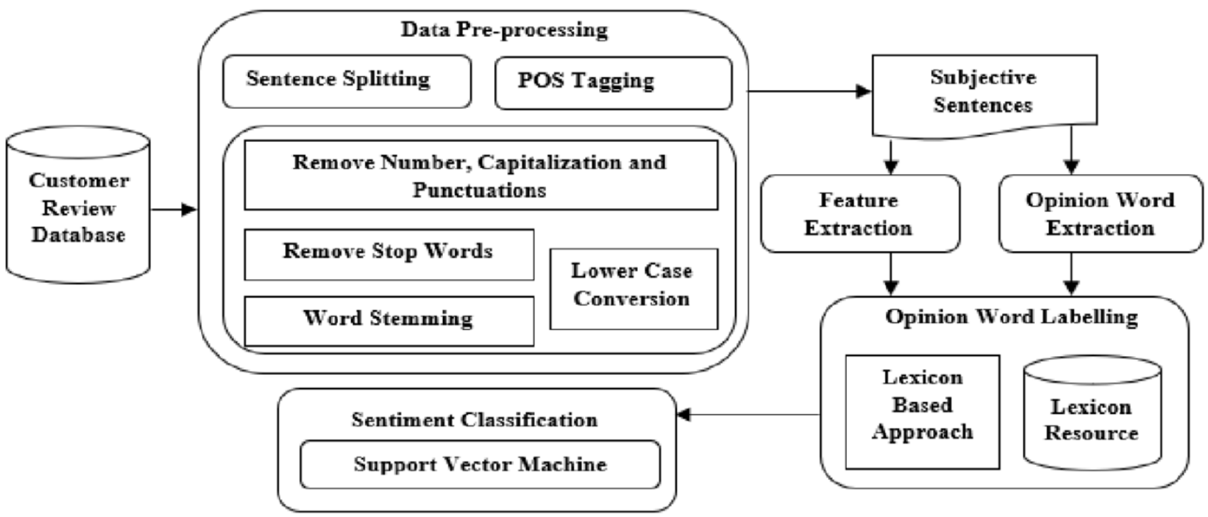

Figure 1.1 Proposed Flow Diagram.[3]

Our model consists of four main steps: (I) Creating the Corpus, (II) Vectorizing and Extracting Features, (III) Oversampling the Minority Instances, and (IV) Training the Classification Algorithms. We have applied two machine learning algorithms and oversampling techniques to the model and conducted a comprehensive study to compare their effectiveness.

Chapter 2: Literature Reviews

The design of NLP-related solutions involves several key steps that are essential for creating effective and accurate natural language processing systems. First, the problem domain and the specific application of the NLP solution must be defined. This involves understanding the user's needs, the type of data that will be processed, and the expected outcomes of the NLP solution. Next, the data must be collected and pre-processed to ensure that it is in a format that NLP techniques can effectively analyze. This may involve cleaning, tokenization, and normalization of the data. After the data is prepared, appropriate NLP techniques and algorithms must be selected and applied. This may involve machine learning, deep learning, or statistical approaches, depending on the specific application. Once the NLP models are developed, they must be evaluated and refined to ensure accuracy and effectiveness. This may involve fine-tuning the models, testing their performance on different datasets, and optimizing their parameters. Finally, the NLP solution must be integrated into a larger system or application to ensure that it can be used effectively by end users. This may involve developing a user interface, integrating with other systems, and providing documentation and support for users. Overall, the design of NLP-related solutions requires a deep understanding of both the technical aspects of NLP and the specific needs and requirements of the end users [3].
NLP using Sentiment analysis techniques and it is a popular research area due to the abundance of user-generated content on the platform and the importance of understanding customer sentiment for businesses. Many studies have been conducted on this topic, using various machine learning algorithms such as Naive Bayes, Support Vector Machines (SVM), Random Forest, and deep learning techniques such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models [4]. These studies have shown that NLP techniques can effectively classify Amazon reviews as positive, negative, or neutral, with high accuracy. Furthermore, some studies have explored hybrid approaches that combine NLP techniques with feature selection and weighting methods to achieve even higher accuracy in sentiment classification. Overall, sentiment analysis of Amazon reviews using NLP techniques has proven to be a valuable tool for businesses to understand customer sentiment and improve their products and services [5]. The detection of sarcasm has become a prominent area of research in natural language processing in recent years. This research began in 2010 with the construction of a corpus extracted from social media sites such as Twitter and Amazon [6]. Several algorithms were proposed for detecting sarcastic words in this corpus, with one achieving an accuracy of 81% [7]. Other researchers in [8] have focused on detecting sarcasm in specific languages, such as Hindi and Czech, and have achieved high levels of accuracy. Some researchers have explored the use of behavioral and psychological aspects to detect sarcasm, while others have found that including additional information about the author, environment, and audience of a tweet can improve accuracy. Overall, these studies show that sarcasm can take many different forms and that it is possible to teach machines to understand it, particularly in the context of close relationships.

In a research paper published in [9], various classifiers and feature sets were tested for the automatic detection of sarcasm and nastiness. The study found that the features varied depending on the emoticon used and also highlighted the diverse forms sarcasm can take. Another study conducted in the same year by Ptáček et al. [10] proposed an effective approach for detecting sarcasm in both Czech and English languages, with the SVM classifier performing the best on the Czech dataset. In 2015, Rajadesingan et al. [11] contributed a unique approach called SCUBA, which uses behavioral and psychological aspects to detect sarcasm, making it particularly useful for real-world applications. Additionally, [12] showed that providing extra information such as author, environmental, and audience details in a tweet can improve accuracy in detecting sarcasm. These studies collectively demonstrate that it is possible to teach machines to understand the nuances of sarcasm, including its close association with personal relationships. In [13], a Parsing Based Lexical Generation Algorithm (PBLGA) was proposed for generating a lexicon to detect sarcasm, and the Interjection Word Start (IWS) algorithm was introduced to add a hyperbole feature for detection. The authors provided a comparative study between the two algorithms, with PBLGA achieving the highest precision score of 0.89, while IWS recorded the highest recall and fi-score of 0.96 and 0.9, respectively. Another approach based on multiple information sources was presented in [14], where the authors demonstrated that Support Vector Machine (SVM) outperformed Multinomial Naive Bayes (MNB) in a broad range of scenarios. However, Mukherjee and Bala [15] showed that the efficiency of MNB is better than fuzzy clustering methods. The classification between irony and sarcasm can be challenging since both generate similar emotions. Detailed studies were conducted on this classification, and the problem of ambiguity during detection was identified.

In recent years, issues related to imbalanced data have been identified in the fields of sentiment analysis and sarcasm detection [16]. Studies conducted in [12] found a significant decrease in F1-scores for imbalanced sarcasm detection compared to balanced sarcasm detection. This suggests that a supervised learning model trained with imbalanced data can severely affect the performance of a classifier, resulting in incorrect classification of texts. To address this issue, Hazarika et al. [17] developed the CASCADE technique for sarcasm detection, which showed improvement on an imbalanced dataset. However, the literature survey revealed that only a few studies have been conducted to understand the effect of imbalanced classes in sarcasm detection. Therefore, the current article proposes a comprehensive study to address the issue of imbalanced class problems specifically in sarcasm detection. To study the issue of imbalanced classes in sarcasm detection, datasets containing sarcastic comments on various topics were collected. Various feature extraction techniques were used, followed by several synthetic minority oversampling methods to mitigate the problem of imbalanced classes. To test the effectiveness of these minority oversampling-based methods, well-known classification algorithms were trained and tested. The performance of the classifiers was measured using test-phase confusion matrix-based performance metrics. The results showed that the synthetic minority oversampling-based algorithms were highly effective in improving classifier performance in sarcasm detection. Table 1 show variety of NLP techniques and machine learning models for sentiment analysis of Amazon reviews.

Table 2.1 Summary of related works.

Referen ce	Finding or development	Technique used with the standard acronym
[18]	Compare the performance of various supervised and unsupervised machine learning techniques for sentiment analysis of Amazon reviews.	The authors found that the Support Vector Machine (SVM) classifier performed the best among the supervised techniques, while the Latent Dirichlet Allocation (LDA) model performed the best among the unsupervised techniques.
[19]	Used various machine learning techniques, including SVM, Naive Bayes, and Random Forest, to analyze Amazon reviews.	The authors found that SVM performed the best among the three techniques in terms of accuracy.
[20]	Utilized deep learning techniques, specifically Convolutional Neural Networks (CNNs), to analyze Amazon reviews.	The authors found that the CNN model achieved high accuracy in sentiment classification, outperforming traditional machine learning techniques.
[21]	Utilized various machine learning techniques, including Naive Bayes, SVM, and Random Forest, to classify Amazon reviews as positive, negative, or neutral.	The authors found that Naive Bayes performed the best among the three techniques, with an accuracy of 85.15%.
[22]	Using deep learning techniques, specifically Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs), to analyze Amazon product reviews	The authors found that the LSTM model achieved higher accuracy than the CNN model, with an overall accuracy of 87.46%.

Chapter 3: Research Methodology and Experiments

Using NLP techniques to Preprocess and Clean Text Dataset from what we learn in the course, Figure 3.1 shows flowchart of the developing sentiment analysis classification model [23]. Below is a step-by-step explanation of how to perform preprocessing and text analysis based on the course material

- **Collecting Data**: collecting data from its source and creating a Dataset Reprocessing: deleting the symbols and unwanted numbers, or letters of other languages
- **Filtering**: It includes several things: Spelling correction (Misspelling), delete Repeated Letters, Eliminate the words (Stop Words), which are words that do not indicate a meaning in themselves, such as: pronouns / nouns and nouns connected / prepositions and accusative ... etc.
- **Normalizing**: by replacing some letters with similar letters and deleting punctuation marks; the goal is for the text to appear in a single form that is easy to find in the dictionary used in the classification process.
- **Classifying by Machine Learning**: the process of classifying texts into positive, negative, or neutral.
- **Evaluation metrics**: As a classification problem, Sentiment Analysis uses the evaluation metrics of Precision, Recall, F-score, and Accuracy.
- **Visualize Results**: To visualize the results of Sentiment Analysis, such as graphs, histograms, and confusion matrices.
- **Model predicted**: using example to see how the model classify reviews.

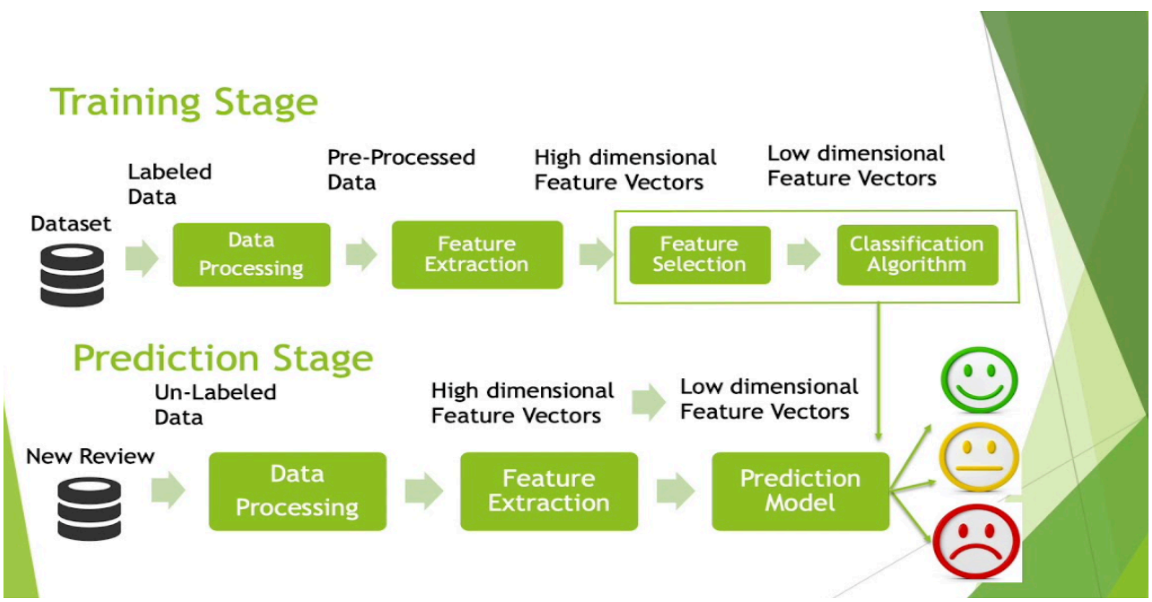

Figure 3.1 flowchart of a developing sentiment analysis classification model [23]

Chapter 4: Dataset and Pre-processing

4.1 Data Extraction:

The data was acquired from Kaggle and the size 39.2+ KB, xwhich includes nine columns. It con sists of reviews of fine foods from amazon. The data span a period of more than 10 years, includi ng all 500,000 reviews up to October 2012. Reviews include product and user information, Score , time, summary, and a plain text review as show in Figure 4.1.1 analysis focus in the column Tex t and Score. Figure 4.1.2 show counting reviews by score which appear score 5 is the most.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Id                      500 non-null    int64
 1   ProductId               500 non-null    object
 2   UserId                  500 non-null    object
 3   ProfileName             500 non-null    object
 4   HelpfulnessNumerator    500 non-null    int64
 5   HelpfulnessDenominator  500 non-null    int64
 6   Score                   500 non-null    int64
 7   Time                    500 non-null    int64
 8   Summary                 500 non-null    object
 9   Text                    500 non-null    object
dtypes: int64(5), object(5)
memory usage: 39.2+ KB
None
```

Figure 4.1.1 Statistics of Dataset used.

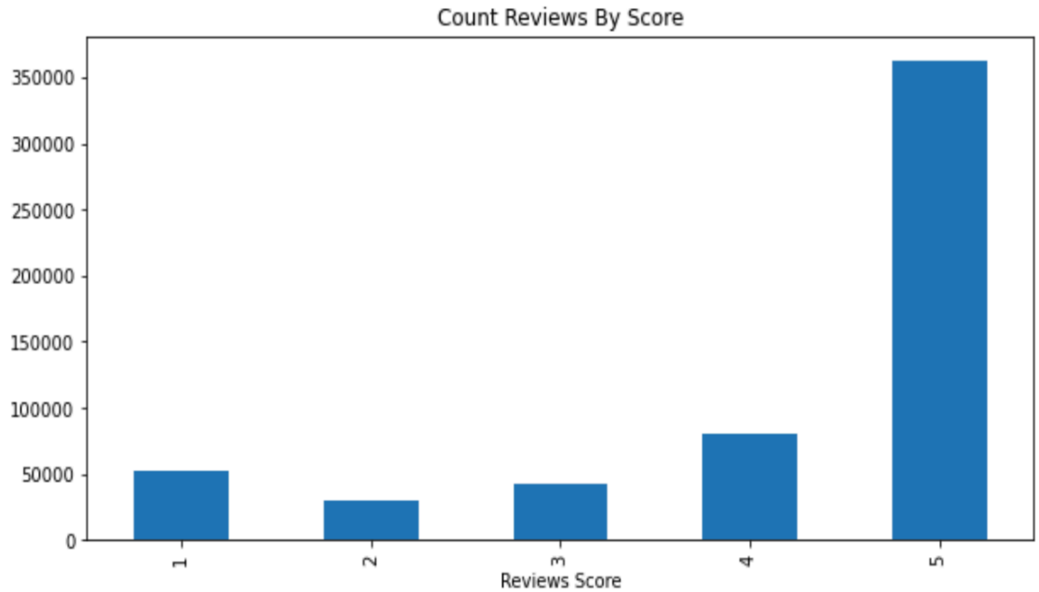

Figure 4.1.2 count reviews by score

4.2 Preprocessing:

4.2.1 Dataset:

Before sending the reviews to the model, we had to perform a considerable amount of preprocessing.

1- The data is clean and does not contain of null values
2- Rename and drops some columns that do not affect for the dataset.

4.2.2 Text :

Text cleaning or Text pre-processing is a mandatory step when we are working with text in Natural Language Processing (NLP). In real-life human writable text data contain various words with the wrong spelling, short words, special symbols, emojis, etc. we need to clean this kind of noisy text data before feeding it to the machine learning model.

Basic sentiment analysis of text reviews follows a straightforward process:

1- Doing some statistics on word frequency and character, special character stop ward and number. It can be of assistance to translators when calculating quotes and create a copy of the dataset figure 4.2.2 show the statistics on words frequency and character.
2- Tokenizing sentences to break text down into sentences, words, or other units.
3- Normalizing words: Stemmed and lemmatize the words.
4- Convert word to lowercase: It entails condensing all forms of a word into a single representation of that word. For instance, “Watched,” “watching,” and “watches” can all be normalized into “watch,” by condensing all forms of a word into a single form or lower case.
5- Removing stop words, punctuation, and spelling correction: Stop words are words that may be important in human communication but are of little value for machines. Like “if,” “but,” “or,” and so on.
6- Vectorizing text by turning the text into a numerical representation and assign a sentiment score to each phrase and component (-1 to +1).

ProductId	UserId	ProfileName	Score	Time	Summary	Text	Text_len	char_count	word_count	hashtags	stopwords	number_count
3001E4KFG0	A3SGXH7AUHU8GW	delmartian	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	263	263	49	0	21	0
i00813GRG4	A1D87F6ZCVE5NK	dll pa	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	190	190	31	0	12	0
000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	509	509	99	0	42	0
3000UA0QIQ	A395BORC6FGVXV	Karl	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...	219	219	43	0	15	0
3006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...	140	140	30	0	12	0

Figure 4.2.2 show the statistics on words frequency and character.

Chapter 5: Sentiment Analysis Models

Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral. The sentiment function of textblob returns two properties, polarity, and subjectivity. Polarity is float number which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjective sentences generally refer to personal opinion, emotion, or judgment whereas objective refers to factual information. Subjectivity is also a float number which lies in the range of [0,1].

5.1 Creating 'sentiment' column:

This is an important preprocessing phase; we are deciding the outcome column (sentiment of review) based on the overall score. If the score is greater than 3, we take that as positive and if the value is less than 3 it is negative If it is equal to 3, we take that as neutral sentiment. Figure 5.1.1 show the numbers of different sentiment categories/classes, and it clearly the number of positive sentiments is having high counts followed by negative and the last one neutral. Figure 5.1.2 the code def of create sentiment column.

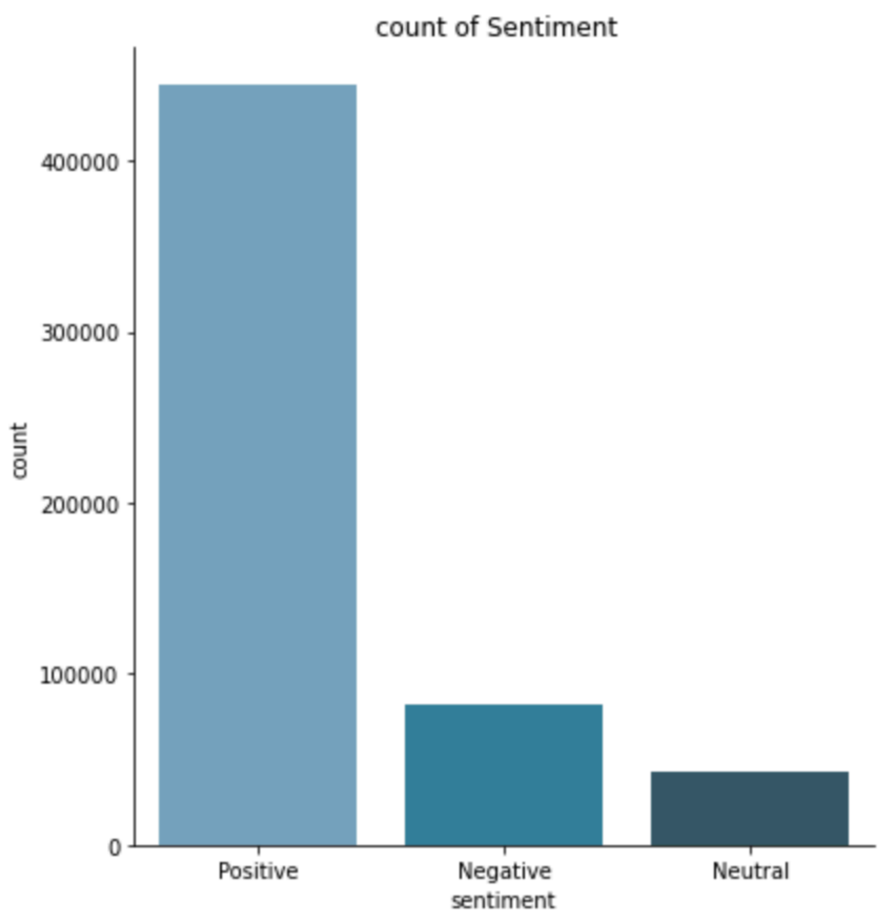

Figure 5.1.1 Show Numbers of different Sentiment Categories /Classes

```python
def f(row):

    '''This function returns sentiment value based on the Score ratings from the user'''

    if row['Score'] == 3.0:
        val = 'Neutral'
    elif row['Score'] == 1.0 or row['Score'] == 2.0:
        val = 'Negative'
    elif row['Score'] == 4.0 or row['Score'] == 5.0:
        val = 'Positive'
    else:
```

Figure 5.1.2 code def of create sentiment column.

5.2 Building the Proposed Models

In this study, it used one machine learning model and Textblob library to predicts the sentiment and get the accuracy. They are addressed in depth below.

5.3 TextBlob

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common NLP tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more [23]. TextBlob stands on the giant shoulders of Natural Language Toolkit and pattern and plays nicely with both [24].
Features

- Noun phrase extraction
- Part-of-speech tagging
- Sentiment analysis
- Classification (Naive Bayes, Decision Tree)
- Tokenization (splitting text into words and sentences)
- Word and phrase frequencies
- Parsing
- n-grams
- Word inflection (pluralization and singularization) and lemmatization
- Spelling correction
- Add new models or languages through extensions
- WordNet integration

5.3.1 Predicted the Sentiment Categories:

First predict sentiment categories from the textblob tool from the actual dataset, create function using textblob to get the sentiment categories and polarity to compare with actual dataset and get the accuracy. Figure 5.3.1 shows the number of sentiment vs polarity, and the greater number of positive reviews are having the most polarity rate. Figure 5.3.2 show the code for implementing this function.

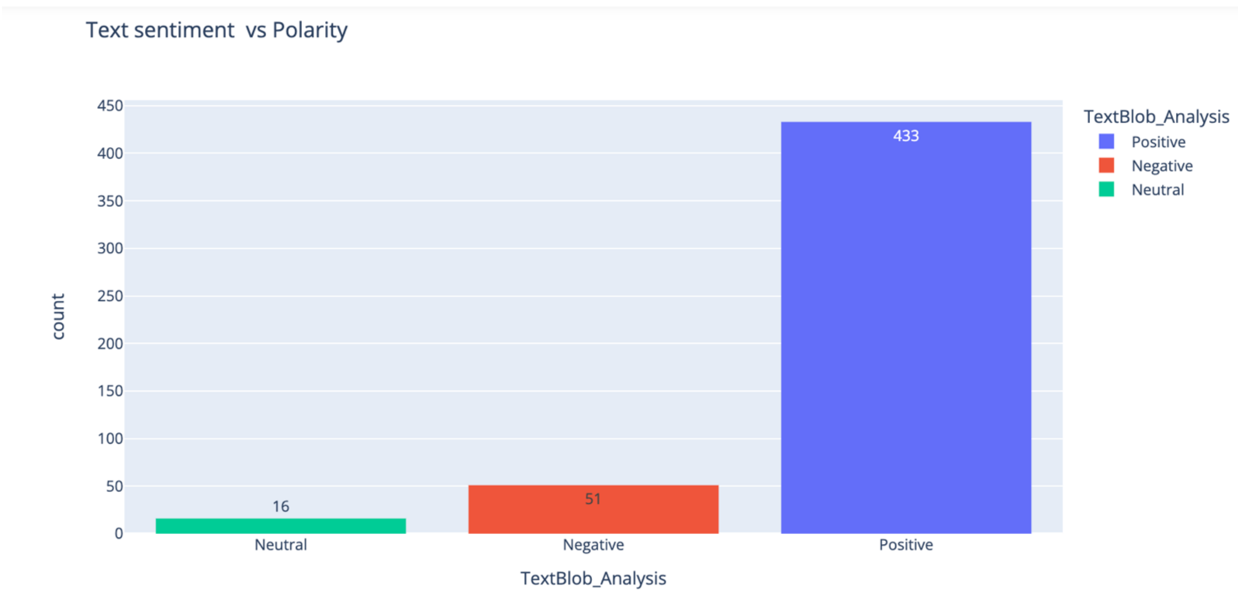

Figure 5.3.1 the count of sentiment vs polarity

```
: #def sentiment_analysis(Text):

def getSubjectivity(df11):
    return TextBlob(Text_new).sentiment.subjectivity

 #Create a function to get the polarity
def getPolarity(Text_new):
    return TextBlob(Text_new).sentiment.polarity

#print(df1[['Text_new','sentiment']].head(20))
 #Create two new columns 'Subjectivity' & 'Polarity'
df11['TextBlob_Subjectivity'] = df11['Text_new']#.apply(getSubjectivity)
df11['TextBlob_Polarity'] = df11['Text_new'].apply(getPolarity)
def getAnalysis(sentiment):
  if sentiment < 0:
     return 'Negative'
  elif sentiment == 0:
     return 'Neutral'
  else:
     return 'Positive'
df11 ['TextBlob_Analysis'] = df11 ['TextBlob_Polarity'].apply(getAnalysis )
print(df11.head(20))#
```

Figure 5.3.2 predict textblob code.

5.3.2 Results

In the accuracy compare actual (sentiment) and Predicted (TextBlob_analysis), and the results of accuracy score show the TextBlob accuracy 79%, the code and result shown in the figure 5.3.2

```
# Evaluate the model performance : accuracy

cm1 = metrics.accuracy_score(df11['sentiment'],df11['TextBlob_Analysis'])
cm1
```

```
0.792
```

Figure 5.3.2. the code of accuracy result.

5.3.3 Example of the Textblob performers different dataset:

```
TextBlob_Analysis = TextBlob("The food was great!")
print(TextBlob_Analysis.sentiment)
```

```
Sentiment(polarity=1.0, subjectivity=0.75)
```

Figure 5.3.3 code of perform and result of TextBlob.

5.4 Naive Bayesian classifier

Multinomial Naive Bayes algorithm is another method used in this report, which is a supervised machine learning algorithm that is mostly used in Natural Language Processing (NLP), like text classification. The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output. It is also part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category [26]. The equation 5.4 below shows Naive Bayesian formula.

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Equation 5.4 Naive Bayesian [26]

5.4.1 Model Training and Evaluation

Using the scikit-learn tool to train-test split as we learn in the course. Based on the model's analysis results, with the most efficient partition using 70% of the data for training and 30% for testing how well the model performed. Because this was a classification problem, using evaluation metrics such as classification accuracy, confusion matrix that represented the output as a probability to assess how well the two models performed [25].
Accuracy is used in classification problems to tell the percentage of correct predictions made by a model. Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made. We calculate it by dividing the number of correct predictions by the total number of predictions [25].
Confusion Matrix is a very popular measure used while solving classification problems. It can be applied to binary classification as well as for multiclass classification problems. An example of a confusion matrix for binary classification is shown in Table 5.4.1.

Table 5.4.1 example of matrix classification [25]

		Predicted	
Actual		Negative	Positive
	Negative	TN	FP
	Positive	FN	TP

Confusion matrices represent counts from predicted and actual values. The output "TN" stands for True Negative which shows the number of negative examples classified accurately. Similarly, "TP" stands for True Positive which indicates the number of positive examples classified accurately. The term "FP" shows False Positive value, i.e., the number of actual negative examples classified as positive; and "FN" means a False Negative value which is the number of actual positive examples classified as negative. One of the most commonly used metrics while performing classification is accuracy. The accuracy of a model (through a confusion matrix) is calculated using the given formula below [25]. Figure 3.9 shown the code.

$$\text{Accuracy} = \frac{\text{TN+TP}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}$$

Equation 5 4.2 accuracy formula [25]

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

# Step 1: Prepare the training data
x = df11['Text_new']
y = df11['sentiment']

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)

# Step 2: Vectorize the training and test data
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

# Step 3: Train the Naive Bayes classifier
clf = MultinomialNB()
clf.fit(X_train_vectorized, y_train)

# Step 4: Predict the labels for the test data
y_pred = clf.predict(X_test_vectorized)
```

Figure 5.4.1 model training naive bayes code

5.4.2 Results and Discussions:

The results show that Naive Bayesian outperformed the TextBlob, and the accuracies obtained were above 85% and for textblob was 79%. The comparison of the performance of the two models is showing in Figure 5.4.2. However, for classification problems, it's important to obtain a confusion matrix and evaluate the F1 score rather than solely relying on the accuracy metric. Figure 5.4.3 shows the confusion matrix with ROC, and we check our f1 score and Figure 5.4.4 shows the classification report.

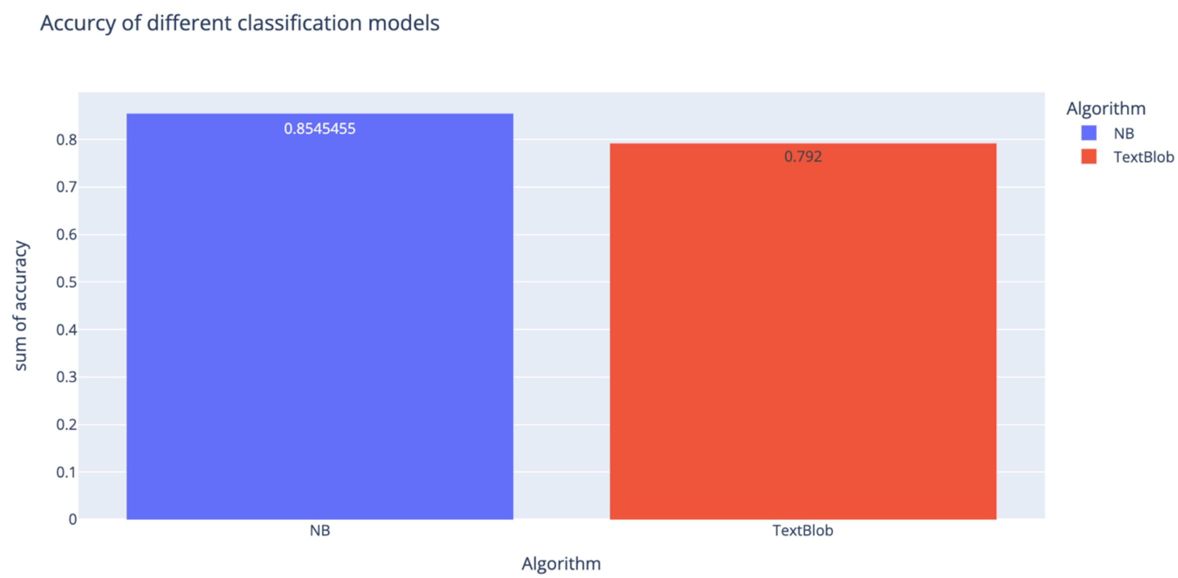

Figure 5.4.2 compare of two accuracy models.

```
Confusion Matrix:
[[ 17345   1434   7961]
 [  2786   3361   7967]
 [  4869   2611 139256]]
```

Figure 5.4.3 confusion matrix

Classification Report:

	precision	recall	f1-score	support
Negative	0.69	0.65	0.67	26740
Neutral	0.45	0.24	0.31	14114
Positive	0.90	0.95	0.92	146736
accuracy			0.85	187590
macro avg	0.68	0.61	0.64	187590
weighted avg	0.83	0.85	0.84	187590

Figure 5.4.4 classification report

Figure 3.12 shows the classification report. Given the significance of predicting positive, negative, and neutral reviews, we have taken all of them into account. We achieved a fairly high F1 score accuracy.

5.4.3 Example of the naïve bayes model performers in different dataset:

```
# Step 4: Predict the label of a new data string
new_data_string = "food is good thing"
X_new = vectorizer.transform([new_data_string])
predicted_label = clf.predict(X_new)

# Step 5: Print the predicted label
print("Predicted label:", predicted_label[0])
```

```
Predicted label: Positive
```

Figure 5.4.3 code of perform and result of Naive bayes.

Chapter 6: Conclusion and Future Works:

Sentiment analysis is a popular method of text classification that evaluates a message to determine whether the sentiment behind it is positive, negative, or neutral. This is particularly important for businesses, as customers now have more opportunities than ever to express their thoughts and feelings openly. It can be difficult for humans to comb through every line of feedback and identify the underlying emotions conveyed by the user. However, with the aid of technology, businesses can automatically analyze customer feedback, from survey responses to social media conversations, and use the insights gained to better understand their customers' needs and preferences. This allows companies to listen attentively to their customers and tailor their products and services accordingly.

Our approach aims to develop and compare one a hybrid machine learning model (NLP model) and Textblob tool that achieves a balanced and impartial classification of products reviews. The results show that Naive bayes outperformed Textblob model, and all the accuracies obtained were 85%. The model can be improved and made more practical by applying PCA (Principal Component Analysis) to the active learning process to fully automate data labeling. This model can be incorporated with programs that interact with customers seeking product scores. Because we used a large dataset, we can use the model on local market sites to improve accuracy and usability. Finally, we will try to generalize this model to all types of text-based reviews and comments.

Reference:

1. Chomsky, N. (2014). Aspects of the Theory of Syntax (Vol. 11). MIT press.
2. Hu, M., & Liu, B. (2004, August). Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 168-177).
3. Korovkinas, K., Danėnas, P., & Garšva, G. (2019). SVM and k-means hybrid method for textual data sentiment analysis. Baltic Journal of Modern Computing, 7(1), 47-60.
4. Katić, T., & Milićević, N. (2018, September). Comparing sentiment analysis and document representation methods of amazon reviews. In 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY) (pp. 000283-000286). IEEE.
5. Banerjee, A., Bhattacharjee, M., Ghosh, K., & Chatterjee, S. (2020). Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media. Multimedia Tools and Applications, 79(47-48), 35995-36031.
6. Davidov, D., Tsur, O., & Rappoport, A. (2010, August). Enhanced sentiment learning using twitter hashtags and smileys. In Coling 2010: Posters (pp. 241-249).
7. Davidov, D., Tsur, O., & Rappoport, A. (2010, July). Semi-supervised recognition of sarcasm in Twitter and Amazon. In Proceedings of the fourteenth conference on computational natural language learning (pp. 107-116).
8. Bharti SK, Babu KS, Raman R (2017) Context-based sarcasm detection in Hindi comments. 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), Bangalore, pp 1–6.
9. Joshi, A., Bhattacharyya, P., Carman, M. J., Joshi, A., Bhattacharyya, P., & Carman, M. J. (2018). Sarcasm detection using contextual incongruity. Investigations in Computational Sarcasm, 93-118.
10. Bamman D, Smith NA (2014) Contextualized sarcasm detection on twitter. In: Proceedings of the Ninth International AAAI conference on Web and Social Media, UK, pp 574–577.
11. Bouazizi, M., & Ohtsuki, T. (2015, December). Sarcasm detection in twitter:" all your products are incredibly amazing!!!"-are they really?. In 2015 IEEE global communications conference (GLOBECOM) (pp. 1-6). IEEE.
12. Ghosh K, Banerjee A, Chatterjee S, Sen S (2019) Imbalanced twitter sentiment analysis using minority oversampling. In: IEEE 10th International Conference on Awareness Science and Technology (iCAST). IEEE, pp 1–5.
13. Ling, J., & Klinger, R. (2016). An empirical, quantitative analysis of the differences between sarcasm and irony. In The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29–June 2, 2016, Revised Selected Papers 13 (pp. 203-216). Springer International Publishing.
14. Bharti SK, Babu KS, Jena SK (2015) Parsing-based sarcasm sentiment recognition in twitter data. In: Proceeding ASONAM '15 Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, France, pp 1373–1380.
15. Bouazizi, M., & Ohtsuki, T. O. (2016). A pattern-based approach for sarcasm detection on twitter. IEEE Access, 4, 5477-5488.
16. Ptáček T, Habernal I, Hong J (2014) Sarcasm detection on Czech and english twitter. In: Proceedings of COLING (2014), The 25th international conference on computational linguistics: technical papers, Dublin, pp 213–223.

17. Han J, Kamber M, Pei J (2012) Data mining concepts and techniques, third edn. Morgan Kaufmann publishers. Elsevier, pp 330–343.
18. Salmony, M. Y. A., & Faridi, A. R. (2021, April). Supervised Sentiment Analysis on Amazon Product Reviews: A survey. In 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM) (pp. 132-138). IEEE.
19. Wassan, S., Chen, X., Shen, T., Waqar, M., & Jhanjhi, N. Z. (2021). Amazon product sentiment analysis using machine learning techniques. Revista Argentina de Clínica Psicológica, 30(1), 695.
20. Shah, B. K., Jaiswal, A. K., Shroff, A., Dixit, A. K., Kushwaha, O. N., & Shah, N. K. (2021, January). Sentiments detection for amazon product review. In 2021 International conference on computer communication and informatics (ICCCI) (pp. 1-6). IEEE.
21. Gope, J. C., Tabassum, T., Mabrur, M. M., Yu, K., & Arifuzzaman, M. (2022, February). Sentiment Analysis of Amazon Product Reviews Using Machine Learning and Deep Learning Models. In 2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE) (pp. 1-6). IEEE.
22. Brownfield, S., & Zhou, J. (2020). Sentiment analysis of Amazon product reviews. In Software Engineering Perspectives in Intelligent Systems: Proceedings of 4th Computational Methods in Systems and Software 2020, Vol. 2 4 (pp. 739-750). Springer International Publishing.
23. https://textblob.readthedocs.io/en/dev/
24. https://www.nltk.org/
25. https://www.kaggle.com/code/benroshan/sentiment-analysis-amazon-reviews
26. J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.
27. https://www.sciencedirect.com/topics/engineering/confusion-matrix
28. https://www.saedsayad.com/naive_bayesian.htm