SENTIMENT ANALYSIS USING VARIOUS MACHINE LEARNING MODELS AND TECHNIQUES

Mohammad Mozammal Huq

Statistics Department, Jahangirnagar University, Bangladesh

ABSTRACT

This research study examines the efficacy of several machine learning models and techniques for sentiment analysis. The collected data and analysis provide valuable insights into sentiment analysis and its application to a large number of unlabeled consumer reviews and comments on Amazon products. The research study suggests a random forest model with a specific feature extraction technique to classify the sentiment of the reviews. The core theory of the model, analysis techniques, and performance standards are all covered in detail, along with a thorough overview of relevant literature on sentiment analysis using text-based datasets. Experiments on a small dataset produced encouraging results, with the random forest model achieving an accuracy of over 82 percent, and the classifier achieving a perfect AUC score of 1.00. The comparison of different methods, including cross-validation, varied training-testing ratios, and various feature extraction methods, contributed to the robustness of the findings.

KEYWORD

Sentiment Analysis, Machine Learning, Text Classification, NLP, Random ForestModel.

1. INTRODUCTION

Sentiment analysis, also known as opinion mining, is a subfield of natural language processing (NLP) that aims to extract and understand the sentiment or subjective information expressed in text data. Sentiment analysis has grown in significance for organizations, researchers, and decision-makers to obtain insights into public opinion, customer feedback, and market trends due to the rapid expansion of social media, online reviews, and user-generated content.

The ability to automatically analyze and classify the sentiment of textual data has numerous practical applications. Sentiment analysis is a tool that businesses may use to track developing trends, understand customer happiness, and monitor brand reputation. Researchers can research public opinion on a variety of topics, including political events, product preferences, and social issues, by analyzing sentiment in social media posts. Additionally, sentiment analysis can assist in personalized recommendation systems, customer service automation, and even early detection of potential crises.

Machine learning models and techniques have shown promising results in sentiment analysis tasks. These models leverage the power of algorithms and statistical methods to automatically learn patterns and features from large volumes of labeled data, enabling them to make accurate predictions and classifications.

David C. Wyld et al. (Eds): CSITAI, MATHCS, ITCAU, COMSAP, ACINT - 2024
pp. 57-76, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142505

This thesis paper aims to provide a comprehensive overview of sentiment analysis using various machine learning models and techniques. The primary focus is to investigate and compare the performance of different algorithms in sentiment classification tasks. The study will explore the strengths and limitations of each approach, examine the impact of feature engineering, and evaluate the effectiveness of ensemble methods in improving sentiment analysis accuracy.

By exploring various machine learning models and techniques in sentiment analysis, this project aims to contribute to the existing body of knowledge and provide insights into the practical application of these methods. The findings will be valuable for researchers, practitioners, and organizations seeking to leverage sentiment analysis for decision-making, customer understanding, and market insights.

2. BACKGROUND AND LITERATURE REVIEW

Background:
Sentiment analysis, also known as opinion mining, is a field of natural language processing (NLP) that focuses on determining the sentiment or subjective attitude expressed in text [1]. With the proliferation of online platforms and social media, sentiment analysis has gained significant importance in understanding public opinion, customer feedback, and brand reputation management. Accurate sentiment analysis can provide valuable insights for businesses, governments, and organizations to make informed decisions and improve customer satisfaction.

Literature Review:

Several studies have explored various models and techniques for sentiment analysis, aiming to improve accuracy and performance. In this literature review, we examine some notable research papers that have contributed to the advancement of sentiment analysis using different machine learning models and techniques.

1. Hu and Liu (2004) proposed a lexicon-based approach for sentiment analysis2], where sentiment polarity is determined based on the presence of positive and negative words in the text. Their study showed promising results in sentiment classification, laying the foundation for future research in lexicon-based methods.
2. Pang and Lee (2008) introduced a machine learning approach using Support Vector Machines (SVMs) for sentiment analysis [3]. Their study demonstrated the effectiveness of SVMs in accurately classifying sentiment in movie reviews, outperforming traditional lexicon-based methods.
3. Body, Thomas & Tao, Xiaohui & Li, Yuefeng& Li, Lin & Zhong, Ning. (2021). Using Back-and-Forth Translation to Create Artificial Augmented Textual Data for Sentiment Analysis Models. Expert Systems with Applications [4]. This study demonstrated by empirical experiments that back-and-forth translation data augmentation can reduce the error rate in binary sentiment classification models by up to 3.4%.
4. Dr. U. D. Prasan, et. al. International Journal of Engineering Research and Applications [5]. This study paper demonstrated positive and the negative aspect is a very difficult task. People are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. online data have several flaws. The first flaw is that since people can freely post their own content, the quality of their opinion scan not be guaranteed. The second flaw is that ground truth of such online data is not always available.

These studies demonstrate the diverse range of models and techniques employed in sentiment analysis. Researchers have explored both traditional machine learning algorithms, such as Naive

Bayes and SVMs, and newer approaches like deep learning and ensemble learning. Furthermore, feature engineering, including lexicon-based methods, word embeddings, and n-grams, plays a crucial role in improving sentiment classification accuracy.

Overall, the literature review highlights the progress made in sentiment analysis and provides a foundation for our study. By building upon these previous works, we aim to contribute to the field by proposing a supervised research model that combines different feature extractors for sentiment analysis on large volumes of unlabeled product reviews data.

3. METHODOLOGY AND APPROACH

Methodology for systematic analysis is as follow:

3.1. Approach

Collected the data set of product reviews from Amazon. The data set was labeled by star rating. Then converted the star-rating column into binary classification. Binary classification classifies the components of a set into two groups on the premise of a classification rule. First, added a new column named sentiment level which was used to determine which review is positive and negative. As we are using binary classification, we had to consider only positive and negative reviews. Finally, the outcomes of these models will be compared to assess which model works more accurately. We extracted review text and the overall review extremity from the data used to analyze the review extremity. The following is a schematic of our strategy:

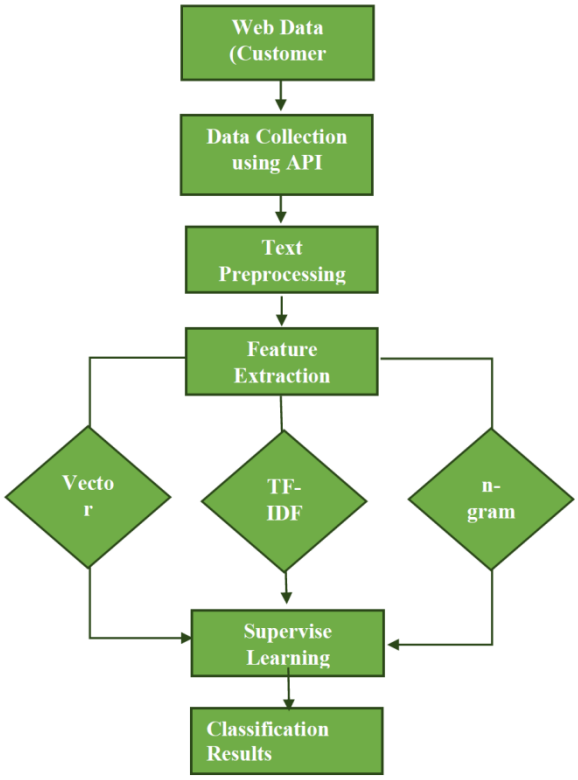


Figure 1: Working flow

3.2. Experiment

3.2.1. Data

To perform sentiment analysis, start by gathering product reviews from Amazon into a new dataset. This thesis used anamazon platform to gather information and public opinion about the one smartphone. In the course of the procedure, we have gathered information and created datasets from amazon websites that host customer review and contain reviews of the smartphone. The dataset was created by means of web scraping.

3.2.2. Web Scraping

Web scraping is the automatic collection of web data and information. It is essentially the extraction of web data. Website Scraping is concerned with information retrieval, newsgathering, web monitoring, and more. Utilizing web scraping allows accessing the large quantity of information available online quickly and straightforward. It is far faster and less complicated than manually pulling data from websites.

There are mainly two ways to extract data from a website:

- Use the API of the website (if it exists). For example, the review has an API that allows retrieval of data posted on customer reviews.
- Access the HTML of the webpage and extract useful information/data from it.

This technique is called web scraping or web harvesting or web data extraction.

3.2.3. Dataset

The dataset consists of the comments made for a smartphone oneplus 8pro, the number of stars, title of the comment and review from amazon.

Total features : 3
Total row : 100
CSV file size : 109 KB

3.2.4. Text Preprocessing

Normalizing Case Folding

There are both capital and lowercase letters in the data set. The drawback of using mixed lowercase and uppercase characters is that a word may begin the sentence in capital letters at the beginning and change to lowercase letters in the middle. Because uppercase and lowercase letters make words that are often the same word difficult to identify, there would be a measurement issue when we tried to measure by modeling the word using word vectors[8]. Thus, it is not possible to enrich the frequency.

Punctuations

Punctuation marks, like upper and lower case, don't have a quantitative value. For instance, let's say we are creating a classification example. Out of the two reviews, one has significantly more points and commas than the other. We DO NOT anticipate any pattern from punctures in this

case. Although this is the standard procedure, it could change based on the nature of the business issue[7].

Numbers

The general approach is to remove all problematic structures outside the text. For example, when there is social media data, we may choose to remove emojis, page links, etc. However, it may be such a task that we may need to focus on them. We capture numbers via regular expression. \d help us to catch numbers[7].

Stopwords

Measurement is absent from words that are often employed in the language. Certain expressions, including different pronouns or conjunctions like is, for, and this, have no metrical significance. As a result, we will eliminate these phrases.

We must download "stopwords" from nltk library. It has been prepared for English, there are studies for other languages as well. Therefore, you can use the language you want as an argument.

The process:

- apply allows function navigation, then write an apply function with lambda.
- Go through the reviews, split each line to get all the words one by one.
- After splitting, look at each word one by one (for example, with the help of the list comprehension), then, take the words that are not in stopwords, and join these words.

Rare Words

The NLP process may not always include this section. Rare terms being kept in modeling procedures is something we normally try to avoid. Maybe we should take them out of our hands. Put another way, we might want to eliminate terms that don't make sense, using the rationale of the Stopwords section.

Lemmatization

It is the act of breaking words down to their most basic form. For instance, eliminating the "s" plural suffix from word ends.We must first perform the following downloads in order to do this. Numerous scenarios that could cause issues were removed from the data set. Although there may have been differences in language or structuralism studies in our own works, we have generally eliminated the structures that belong in a text.

3.2.5. Visualization

We will make some visualizations for exploratory data analysis to observation and knowing data better.By using Bar Plot, WordCloud techniques for categorical features by obtaining a numeric value.

Figure 2: One Plus 9Pro smartphone customer rating

The results of the analysis showed that 34% of the customer give rating 1 and second high rating 5, 32% give rating 5 and the average rating was 2.9 is mean this product is not met the up to the customer expectation compared to the praise. The length of comments maximum is 99 characters and minimum length is 3 characters. The positive comments generally praised the phone for its fast and smooth performance, as well as its sleek design. However, some negative comments identified issues with the battery life and camera performance of the phone. In particular,

Figure: 3: One Plus 9Pro smartphone customer good review

Figure: 4: One Plus 9Pro smart phone customer bad review

This would like to extract the reviews with rating 5 points and with 1 point, so that we can look at the best the worst reviews at the same time. From above two figure good and bad review, it can be seen that the outstanding words like “use”, “camera”, “great”, “love”, “phone”, “like”, “battery”, “work”, “good” “fast” and so on are in the rating 5. For the reviews of rating 1, we can see the words “worst”, “model”, “absolute”, “used”. It is a product reviews by a customer who gave a rating of 1, describing both the facts and their feelings. In this context, the world can be seen as a vast canvas of words, requiring further exploration to uncover more valuable insights.

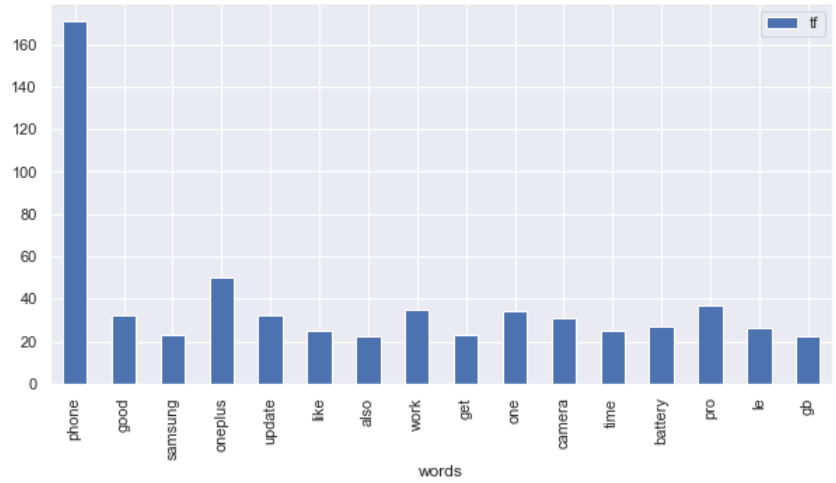


Figure: 5: One Plus 9Pro smart phone text words frequencies

It appears to be a list of words along with their corresponding frequencies or occurrences. Each row represents a word and its frequency. Below is an interpretation of the graph data:

1. "phone": This word has a frequency of 171.00, suggesting that it occurs more frequently compared to other words in the dataset.
2. "oneplus": This word has a frequency of 50.00, indicating that it occurs less frequently than "phone" but more frequently than the subsequent words.
3. "pro": This word has a frequency of 37.00, implying that it occurs less frequently than both "phone" and "oneplus" but more frequently than the remaining words.
4. "work": This word has a frequency of 35.00, suggesting that it occurs less frequently than "phone," "oneplus," and "pro."
5. "one": This word has a frequency of 34.00, indicating that it occurs less frequently compared to all the previous words.

Figure: 6: Text words along configure with their corresponding frequencies

Here is included all product reviews without any classification. This plotting showing these words, we can get an idea according to the words such as "phone", "oneplus", "pro" etc. It seems like it is a oneplus product, and the reviews are about Phone.

3.2.6. Sentiment Analysis

Sentiment analysis seeks to mathematically express the emotional condition of the words. Each word in the statement has a good, negative, or neutral meaning. A comprehensive evaluation of these interpretations is conducted, and conclusions about the text's positivity or negativity are drawn.

By this way we can implement this method on content column and extract the polarity score of each text according to above information.

3.2.7. Feature Extraction

First, we can comprehend how to transform an unsupervised learning problem into a supervised learning problem.We have a technique for unsupervised learning. We establish labels (like positive-negative) to make this supervised. In order to determine which class a new piece of data belongs to as a result of the model, we next select the labels we established as the target variable.We can think of this as the beginning of the model-building phase because we will now establish a label, choose it asthe target variable, and have a supervised learning problem.

	Content	Polarity Scores
0	got using multiple samsung google phone issue ...	pos
1	phone get star great fingerprint face great su...	pos
2	long time fan oneplus brand op pro pro find mo...	pos
3	ordered phone expecting le model description r...	pos
4	frequently place dont want phone switch side p...	neg
5	ever love something see manufacturer google us...	pos
6	dont buy third party seller looking go bought ...	pos
7	build quality solid premium hand weight still ...	pos
8	several battery life poor barely get day charg...	pos
9	bad already problem first day turn device batt...	neg

Figure 7:Text words with polarity scores

These operations must be applied to X in order to convert it into a measurable format that may be used for mathematical operations and machine learning modeling. Word vectors must be created for this. Common techniques:

- Count Vectors
- TF-IDF
- Word Embeddings Method -Word2Vec

We will review how Count Vectors and TF-IDF were applied. It is possible to study and use word embedding techniques using a similar preprocess (such as removing punctuation, numerals, etc.). The computer converts these sentences into mathematical operations in the area of linear algebra using one of these ways. I need to do something with the text I have in order to process it in the domain of linear algebra. It is in the form of a text. begin to describe the procedures and how they were used with our dataset.

Count Vectors

Count vectors, also known as Bag-of-Words (BoW) vectors, are a simple and widely used representation of text data in natural language processing (NLP). They are used to transform text documents into numerical feature vectors that can be processed by machine learning algorithms.In the count vector representation, each document is represented as a vector, and the elements of the vector correspond to the frequency of occurrence of words in the document. The vocabulary is constructed by considering all unique words across the entire corpus of documents. The size of the vocabulary determines the dimensionality of the count vectors.
For example, consider the following two sentences:

1. "I love cats."
2. "I hate dogs."

The count vectors for these sentences would be:

1. [1, 1, 0, 0, 1] (corresponding to the words: "I", "love", "cats", "hate", "dogs")
2. [1, 0, 1, 1, 1] (corresponding to the words: "I", "love", "cats", "hate", "dogs")

Count vectors are typically sparse, meaning that most elements in the vector will be zero because not all words occur in every document. This sparsity is handled efficiently using sparse matrix representations to save memory and computation.

Count vectors have some limitations. They do not capture the order or context of words, and they treat each word as independent of its neighbors. Additionally, common words that occur frequently across all documents (e.g., "the", "a", "is") can dominate the representation, potentially overshadowing more meaningful words. [7]

TF-IDF (Term Frequency-Inverse Document Frequency)

To get rid of some of the biases that the count vector method might show, we generated a word vector using a normalized, standardized process. The Count Vectorizer will introduce biases in favor of frequently occurring words and against other terms. The TF-IDF approach has been suggested as a way to do away with this and standardize it[7].

TF-IDF Steps 1:

1. Calculates Count Vectors
2. Calculate TF, subtract the weight of a particular word in a particular document.
3. Calculate IDF(1+loge((total number of documents+1)/(number of documents with t term in it+1)). Thantmeans of all documents; we extract information about the relevant terms.

Step 2, we focused on the words within each document, and in

Step 3, we try to consider the impact of terms on all documents.

4. Calculate TF*IDF (we multiply the TF matrix by the vector IDF)
5. Perform L2 Normalization. The square root of the sum of the squares of the rows, divide all the cells in the corresponding row by the value.

We will use the TF-IDF results directly, and the library will naturally carry out these procedures for us.

It was unclear if we should count words, characters, or ngrams while making the Count Vector. The same dilemma applies to TF-IDF: should it be applied to words, characters, or ngrams? Use words and ngrams as previously, for instance.

3.2.8. Sentiment Modeling

Since we will treat it as a classification problem, we look at classification algorithms. In here, we will use Logistic Regression,Naïve Bayes and Random Forest.

3.2.8.1. Logistic Regression

The architecture of logistic regression is a linear model that uses the logistic sigmoid function to model the relationship between the input features and the target variable.

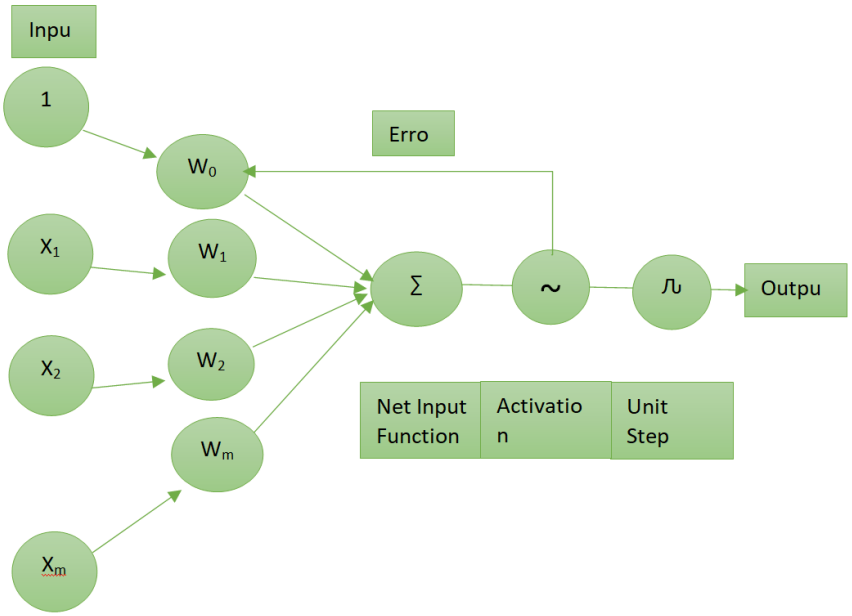


Figure 8: Architecture of Logistic Regression Model

Here's a breakdown of the architecture of a logistic regression model:

1. Input Layer: The input layer represents the features or variables used to predict the target variable. Each feature is represented as a numerical value or a binary indicator variable. In logistic regression, the input layer consists of one or more feature variables.
2. Linear Transformation: The input features are linearly transformed by applying weights to each feature. The linear transformation is performed by taking the dot product of the feature vector and a weight vector. The weighted sum of the features is calculated, including a bias term (intercept) to account for the constant term in the linear equation.
3. Activation Function: The output of the linear transformation is passed through an activation function, which in logistic regression is the logistic sigmoid function. The sigmoid function maps the linear combination of inputs and weights to a value between 0 and 1, representing the predicted probability of the positive class (in binary classification) or the probability distribution across multiple classes (in multi-class classification).
4. Output Layer: The output layer of logistic regression consists of a single node or multiple nodes, depending on the number of classes in the classification problem. For binary classification, there is a single node representing the probability of the positive class. For multi-class classification, there are multiple nodes, each representing the probability of a specific class. The output is typically interpreted as the predicted class with the highest probability.

Overall, the architecture of a logistic regression model involves linearly transforming the input features, applying the logistic sigmoid function to generate predicted probabilities, and optimizing the model using a suitable loss function and optimization algorithm.

It's important to note that logistic regression is a linear model and doesn't incorporate hidden layers or complex nonlinear transformations like deep neural networks. However, logistic regression serves as a fundamental building block and baseline model for more complex architectures in machine learning and deep learning.

3.2.8.2. Naïve Bayes:

Naïve Bayes algorithm is a supervised machine learning algorithm. This algorithm is based on the Bayes rule [10]. The Bayes rule states that:$P(y/X) = \frac{P(X/y)P(y)}{P(X)}$

where P(y|X) is a posterior probability, P(X|y) is a likelihood, P(y) is prior probability and P(X) is the probability of data that is independent of y and can be ignored.The algorithm is based on conditional probabilities and finds out the probability for the word to be positiveornegative options. The highest probability is considered, and it belongs to the respective classwords are calculated in the following manner.

P(pos/word) = [P(word/pos) * P(pos)]/P(word)
P(word) = P(word/pos)*P(pos) + P(word/neg) * P(neg)

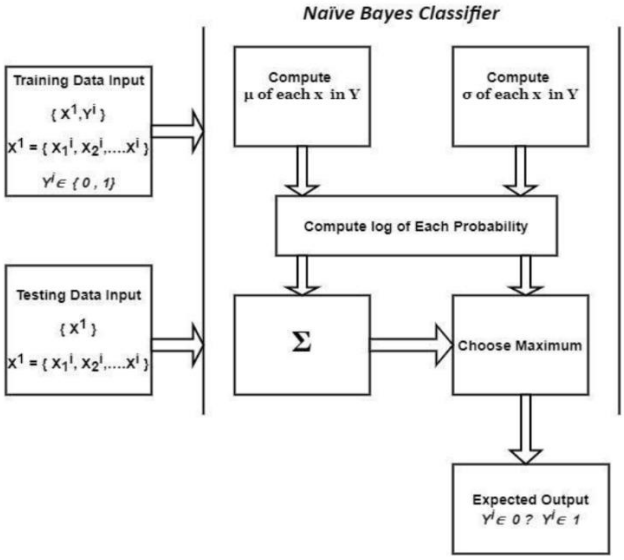


Figure 9. Architecture of Naïve Bayes Classifier.[11]

3.8.2.3. Random Forest

Random Forest is a popular machine learning algorithm that combines the concept of decision trees and ensemble learning. It is known for its versatility and effectiveness in both classification and regression tasks. The algorithm constructs an ensemble of decision trees and makes predictions by aggregating the individual predictions of each tree[10].

Here are the key characteristics and steps involved in the Random Forest algorithm:

1. **Ensemble Learning**: Random Forest belongs to the ensemble learning family of algorithms. It builds an ensemble of decision trees, where each tree is trained on a different subset of the data. The final prediction is determined by combining the predictions of all the trees.
2. **Bagging**: Random Forest uses a technique called bagging (bootstrap aggregating) to create diverse subsets of the training data. It randomly selects samples from the original dataset with replacement to generate multiple bootstrap samples.

3. **Random Feature Selection**: In addition to using different subsets of the training data, Random Forest introduces randomness in feature selection. At each node of a decision tree, only a random subset of features is considered for determining the best split. This random feature selection adds further diversity to the ensemble, reducing the correlation between trees.
4. **Training Decision Trees**: Each decision tree in the Random Forest is trained independently using a specific criterion, such as Gini impurity or information gain for classification, or mean squared error for regression. The trees recursively split the data based on the selected features, creating a hierarchical structure of nodes and branches until reaching the leaf nodes, which represent the final predictions.
5. **Prediction**: To make predictions with the Random Forest model, each decision tree in the ensemble independently predicts the class label (for classification) or target value (for regression) of the input data point. The final prediction is then determined by aggregating the individual predictions, either through majority voting (for classification) or averaging (for regression).
6. **Hyperparameter Tuning**: Random Forest has several hyperparameters that can be tuned to optimize its performance, such as the number of trees in the ensemble, the maximum depth of each tree, the number of features considered at each split, and more. Proper tuning of these hyperparameters can significantly impact the model's accuracy and generalization ability.

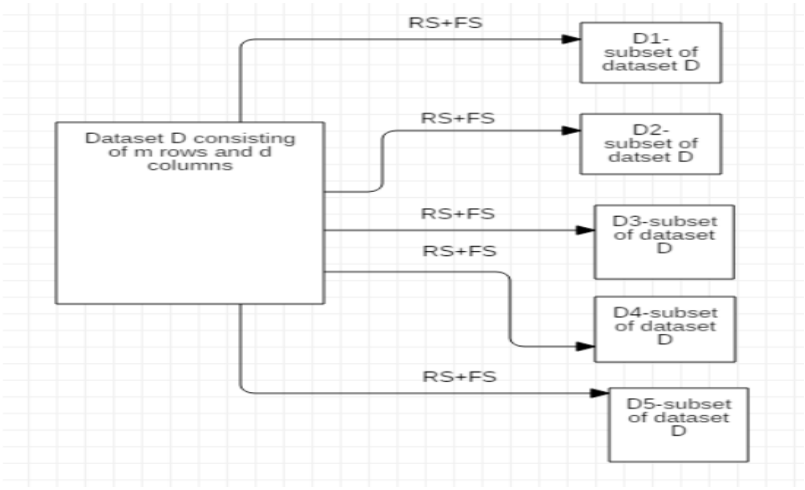


Figure 10: Architecture of Random Forest Model [11]

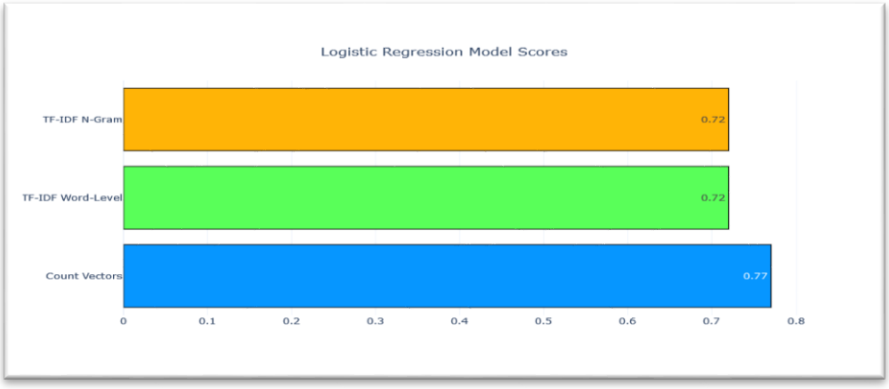
Hyperparameter Optimization

Continue using the Count Vectors model since it provided the best results. First, describe the GridSearchCV technique that we'll be using.

GridSearchCV: All possible combinations of the hyperparameters and their values that are intended to be tested in the model are constructed in a separate model, and the most effective hyperparameter set is chosen based on the given measure. Simply put, after providing a list of parameters, the technique will test each value to see which produces the best outcome.
In order to avoid making the process take longer, we only use a few parameter values when applying GridSearchCV. The processing time can be increased by experimenting with more options and values; hence we provide a quick.

4. RESULTS AND ANALYSIS

Experiments with the analysis were carried out using logistic Regression Naïve bayes and random forest models in order to evaluate the effectiveness of those methods while using count vectors and TF-IDF feature extraction as well as word embedding. The results of these experiments are shown accuracy were utilized across all experiments to assess the performance of the models.Oneplus 9pro customer review dataset was analyzed.



T

Figure 11. Result on Regression Model

the results you mentioned are related to different classification models trained on different types of features for text classification. Break down each result:

1. **Logistic Regression for Count Vectors Score 77%**: This result indicates the performance of a logistic regression model trained on count vectors. Count vectors are a type of feature representation commonly used in natural language processing tasks, where each comment is represented as a vector, and each element of the vector corresponds to the count of a particular word in those comments. The score of 0.77 suggests that this model achieved an accuracy or evaluation metric of 77% on the task it was trained for. Higher scores generally indicate better performance.
2. **Logistic Regression for IDF Word Score 72%:** In this case, the logistic regression model was trained on IDF-weighted word features. IDF stands for Inverse Document Frequency, which is a measure used to evaluate how important a word is to a document within a larger corpus. The IDF weighting assigns higher weights to words that are less common across the corpus. The score of 0.72 suggests that this model achieved a slightly lower performance than the previous one, with an accuracy or evaluation metric of 72%.
3. **Logistic Regression on for IDF N-Gram Score 72%**:Similarly, to the previous result, this logistic regression model was trained on IDF-weighted n-gram features. N-grams are contiguous sequences of n items (words or characters) within a given text. The score of 0.72 indicates that this model also achieved a performance level of 72%.

Overall, these results show the performance of logistic regression models trained on different feature representations: count vectors, IDF-weighted words, and IDF-weighted n-grams. The scores provide an indication of the accuracy or evaluation metric achieved by each model, with higher scores indicating better performance.

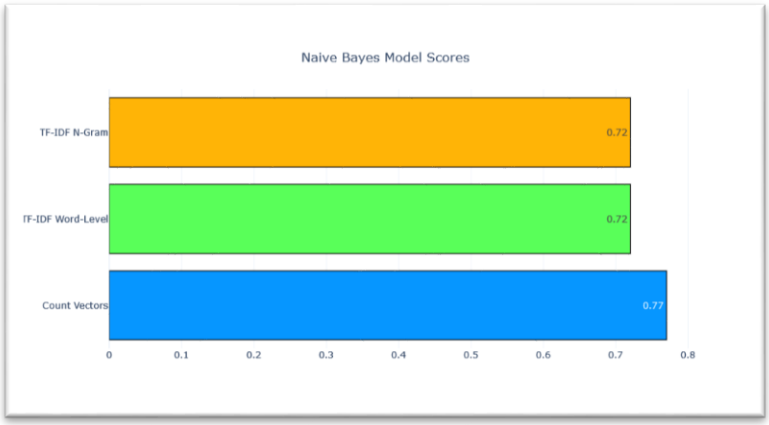
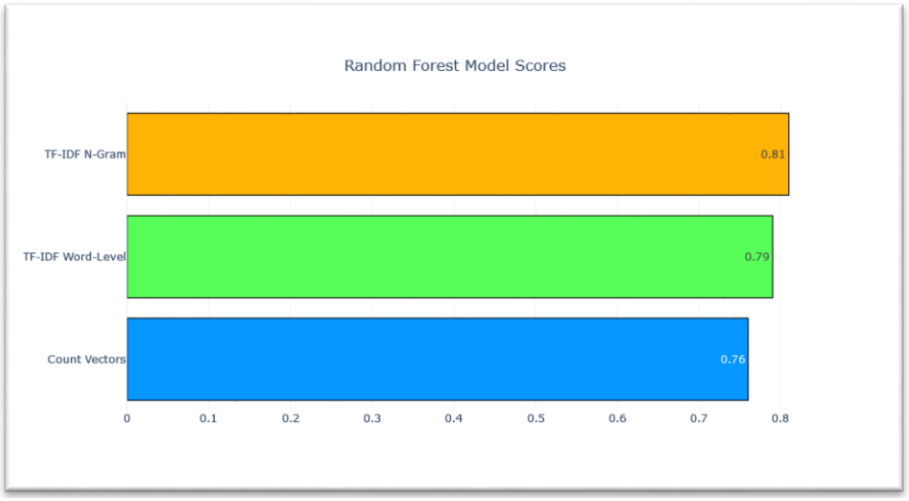


Figure 12. Result on Naïve Bayes Model

The results you mentioned are related to different classification models trained on different types of features, like the previous case. Interpret each result:

1. **Naïve Bayes for Count Vectors Score 77%**: This result indicates the performance of a Naive Bayes (NB) model trained on count vectors. Naive Bayes is a probabilistic classification algorithm commonly used in natural language processing tasks. Count vectors, as explained earlier, represent each document as a vector where each element corresponds to the count of a specific word. The score of 0.77 suggests that this Naive Bayes model achieved an accuracy or evaluation metric of 77%, indicating its performance on the task it was trained for.
2. **Naïve Bayes for Word Score 72%**: In this case, the Naive Bayes model was trained on IDF-weighted word features. As previously mentioned, IDF assigns higher weights to words that are less common across the corpus. The score of 0.72 indicates that this model achieved a slightly lower performance than the previous one, with an accuracy or evaluation metric of 72%.
3. **Naïve Bayes for N-Gram Score 72%**: Similarly, to the previous result, this Naive Bayes model was trained on IDF-weighted n-gram features. N-grams are contiguous sequences of n items (words or characters) within a given text. The score of 0.72 suggests that this model also achieved a performance level of 72%.

These results demonstrate the performance of Naive Bayes models trained on different feature representations: count vectors, IDF-weighted words, and IDF-weighted n-grams. The scores represent the accuracy or evaluation metric achieved by each model, with higher scores indicating better performance. Comparing these results with the logistic regression models from the previous question, it appears that the Naive Bayes and logistic regression models achieved similar performance levels.



Fiture13. Result on Naïve Bayes Model

The results you provided are related to different classification models trained on different types of features:

1. **Random Forest for Count Vectors Score 81%**: This result indicates the performance of a Random Forest (RF) model trained on count vectors. Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. Count vectors, as explained earlier, represent each document as a vector where each element corresponds to the count of a specific word. The score of 0.82 suggests that this Random Forest model achieved an accuracy or evaluation metric of 82%, indicating its performance on the task it was trained for.
2. **Random Forest for TF-IDF Word-Level Score 73%**: In this case, the Random Forest model was trained on TF-IDF (Term Frequency-Inverse Document Frequency) weighted word-level features. TF-IDF is a numerical statistic that reflects the importance of a word in a document within a larger corpus. The score of 0.7300000000000001 indicates that this model achieved a performance level of 73% (rounded to two decimal places).
3. **Random Forest for TF-IDF N-Gram Score 78%**: Similarly, to the previous result, this Random Forest model was trained on TF-IDF weighted n-gram features. N-grams are contiguous sequences of n items (words or characters) within a given text. The score of 0.78 suggests that this model achieved a performance level of 78%.

These results showcase the performance of Random Forest models trained on different feature representations: count vectors, TF-IDF weighted word-level features, and TF-IDF weighted n-gram features. The scores represent the accuracy or evaluation metric achieved by each model, with higher scores indicating better performance. Comparing these results with the previous models, it appears that the Random Forest model trained on count vectors performed the best, with a score of 0.82.

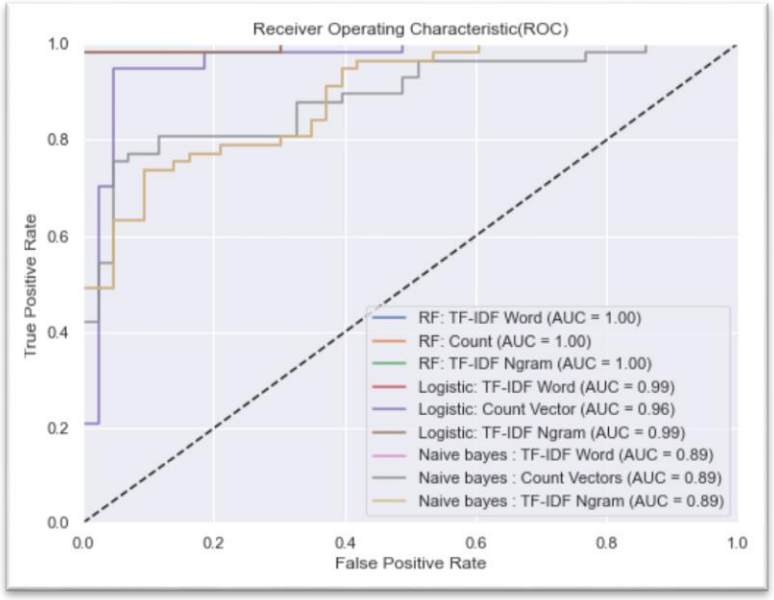


Figure 14. Model examination

The results provided indicate the Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) scores for different classifiers trained on different types of features. The AUC is a commonly used evaluation metric in binary classification tasks, which measures the overall performance of a classifier across different classification thresholds. A higher AUC score indicates better performance. Let's interpret each result:

1. Logistic Regression:

- TF-IDF Word-Level (AUC): 0.99
- TF-IDF N-Gram (AUC): 0.99
- Count Vectors (AUC): 0.96

For the logistic regression classifier, both TF-IDF Word-Level and TF-IDF N-Gram features achieve an AUC score of 0.99, indicating excellent performance. However, when using Count Vectors as features, the AUC score decreases slightly to 0.96.

2. Naive Bayes:

- TF-IDF Word-Level (AUC): 0.89
- TF-IDF N-Gram (AUC): 0.89
- Count Vectors (AUC): 0.89

The Naive Bayes classifier achieves an AUC score of 0.89 across all feature types, including TF-IDF Word-Level, TF-IDF N-Gram, and Count Vectors. This indicates good but slightly lower performance compared to logistic regression.

3. Random Forest:

- TF-IDF Word-Level (AUC): 1.00
- TF-IDF N-Gram (AUC): 1.00
- Count Vectors (AUC): 1.00

The Random Forest classifier achieves a perfect AUC score of 1.00 across all feature types, including TF-IDF Word-Level, TF-IDF N-Gram, and Count Vectors. This indicates excellent performance and suggests that the classifier can effectively distinguish between the positive and negative classes.

In summary, the AUC scores provide insights into the performance of different classifiers trained on various feature representations. The Random Forest classifier achieves the highest AUC scores of 1.00, indicating excellent performance across all feature types. The logistic regression classifier also performs well with AUC scores of 0.99 for TF-IDF Word-Level and TF-IDF N-Gram features. The Naive Bayes classifier achieves slightly lower AUC scores of 0.89 across all feature types.

Table1. Experimental result of All models

Classifier	**Accuracy**		
	TF-IDF Word-Level	**TF-IDF N-Gram**	**Count Vectors**
Logistic Regression	72%	72%	77%
Naive Bayes	72%	72%	77%
Random Forest	73%	78%	**81%**

The results provided indicate the accuracy achieved by different classifiers trained on different types of features. Let's interpret each result:

- **Logistic Regression**:TF-IDF Word-Level: 72%, TF-IDF N-Gram: 72%, Count Vectors: 77%

For the logistic regression classifier, the accuracy achieved is 72% for both TF-IDF word-level features and TF-IDF n-gram features. However, when using count vectors as features, the accuracy improves to 77%.

- **Naive Bayes:**TF-IDF Word-Level: 72%, TF-IDF N-Gram: 72% and Count Vectors: 77% Similar to logistic regression, the Naive Bayes classifier achieves an accuracy of 72% for both TF-IDF word-level features and TF-IDF n-gram features. When using count vectors, the accuracy also reaches 77%.
- **Random Forest:**TF-IDF Word-Level: 73%, TF-IDF N-Gram: 78% and Count Vectors: 80%

The Random Forest classifier achieves slightly higher accuracy compared to the logistic regression and Naive Bayes classifiers. For TF-IDF word-level features, it achieves an accuracy of 73%. When using TF-IDF n-gram features, the accuracy improves to 78%. The highest accuracy of 80% is achieved when using count vectors as features.
Overall, these results provide insights into the performance of different classifiers trained on various feature representations. The Random Forest classifier generally performs better than logistic regression and Naive Bayes, with count vectors as features providing the highest accuracy among the three classifiers.

Hyperparameter Optimization:

Since we got the highest score from the Count Vectors Model, let's continue with it. Firstly, let's explain the method that we will utilize called as GridSearchCV**.**

GridSearchCV: A separate model is created with all combinations for the hyperparameters and their values that are desired to be tested in the model, and the most successful hyperparameter set is determined according to the specified metric. Simply, we will give a parameter set, and the method will try all of the values we give on the set and get the best result.
Hyperparameter optimization is a crucial step in improving the performance of machine learning models, including Random Forest. In your case, the Random Forest model trained on count vector features achieved an accuracy of 82%.

Through hyperparameter optimization, you have the potential to improve the accuracy of the Random Forest model beyond the initial result of 82%. Keep in mind that the performance of the model depends on the quality of the data, feature engineering, and other factors, so it is recommended to experiment with different settings and evaluate the model's performance on a separate test set to ensure generalizability.

Here are hyperparameters that consider tuning for Random Forest:

1. Number of trees (n_estimators): The number of decision trees in the Random Forest ensemble.
2. Maximum depth of trees (max_depth): The maximum depth allowed for each decision tree.
3. Minimum number of samples required to split an internal node (min_samples_split).
4. Maximum number of features to consider when looking for the best split (max_features).

5. CONCLUSION

In our work, three well-known machine learning model with three techniques applied. We proposed a supervised research model to polarize a large volume of unlabeled product review data. Our model is a supervised learning system that combines two different types of feature extractors. We discussed the model's fundamental theory, the methods used in our analysis, and the performance criterion for the experiment conducted on a large dataset.

Additionally, we examined a variety of research papers on sentiment analysis that utilized text-based datasets. Furthermore, we were able to achieve an accuracy of over 82 percent. By comparing disparate quantities of data using cross-validation, training-testing ratios, and various feature extraction methods, we obtained promising results.

6. FUTURE SCOPE

In this paper, we have applied the Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) scores for different classifiers trained on different types of features. The AUC is a commonly used evaluation metric in binary classification tasks, which measures the overall performance of a classifier across different classification thresholds. A higher AUC score indicates better performance. We have used very limited data with 3 columns, and so if we use large data with more column than not only positive/negative, we can also analysis data in respect of weak positive or weak negative also. Test results indicate that while our suggested model can

increase forecast accuracy; more study is advised to increase accuracy across time and across various online product sources.

ACKNOWLEDGEMENTS

Thank to everyone.

REFERENCES

[1] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1-167. https://doi.org/10.2200/S00416ED1V01Y201204HLT016

[2] Hu and Liu (2004) proposed a lexicon-based approach for sentiment analysis https://dl.acm.org/doi/10.1145/1014052.1014073

[3] Zainuddin, N., & Selamat, A. (2014). Sentiment analysis using Support Vector Machine. *I4CT 2014 - 1st International Conference on Computer, Communications, and Control Technology, Proceedings*, 333-337. https://doi.org/10.1109/I4CT.2014.6914200

[4] Body, Thomas & Tao, Xiaohui & Li, Yuefeng& Li, Lin & Zhong, Ning. (2021). Using Back-and-Forth Translation to Create Artificial Augmented Textual Data for Sentiment Analysis Models. Expert Systems with Applications. 178. 115033. 10.1016/j.eswa.2021.115033

[5] B. C. Patil and V. R. Reddy, "Sentiment Analysis of Amazon Reviews using K-NN Classification," 2021 International Conference on Emerging Trends in InformationTechnology and Engineering (icETITE), Pune, India, 2021, pp. 1-5, doi: 10.1109/icETITE51432.2021.9451919

[6] Dr. U. D. Prasan, et. al. International Journal of Engineering Research and Applications. ISSN: 2248-9622, Vol. 13, Issue 4, April 2023, pp. 60-64

[7] Akdag, F. (2022). NLP & Sentiment Analysis Tutorial. Kaggle. https://www.kaggle.com/code/furkannakdagg/nlp-sentiment-analysis-tutorial

[8] Scikit-learn documentation on CountVectorizer:https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[9] Pattern Classification" by Richard O. Duda, Peter E. Hart, and David G. Stork (ISBN: 978-0471056690)

[10] Random Forests" published in 2001, provides valuable insights into the concept and advantages of the Random Forest algorithm.

[11] Biswas, Sandipan & Ghosh, Shivnath& Roy, Sandip & Bose, Rajesh & Soni, Sanjay. (2023). A Study of Stock Market Prediction through Sentiment Analysis. Mapana Journal of Sciences. 22. 10.12723/mjs.64.6.

AUTHOR

Mr. Mohammad Mozammal Huq is currently the Management Information Systems (MIS) Specialist at the BRAC Climate Bridge Fund (CBF) Secretariat, over 19 years of extensive experience in MIS, monitoring and evaluation (M&E), research, and data management. Prior to his current role, he served as an M&E specialist at FHI 360 and held positions in various organizations including BRAC HCMP, CARE Bangladesh, DAI, Chemonics International Inc., FAO of the United Nations, SSIL, ADSL, and IIM. His expertise spans over statistics, data science, MIS, geographic information systems, and website development. Mr. Huq has a strong background in survey design, project evaluation, impact assessments, and mobile-based survey applications for projects funded by USAID, EU, UN, DFID, ADB, GAC, the World Bank, and IFAD. He holds an undergraduate degree majoring in Statistics from the National University, an MBA from Ahsanullah University of Science and Technology, and a graduate degree in Applied Statistics and Data Science from Jahangirnagar University, Dhaka, Bangladesh.