

algorithm's operation. Naive Bayes forecasts future probabilities based on previously collected data or experience. One of the Naive Bayes Classification's characteristics is the presence of independent input variables that presume the presence of an articular feature from a class that is mutually independent of other features.

Alec co utilized a variety of machine learning methods. Numerous machine learning methods are available, including Naive Bayes, Maximum entropy, and Support vector machine. Janice M. Weibe discussed the categorization of documents and sentences. He gathered evaluation data for a variety of product categories, including automobiles, banking, movies, and travel. He classified the terms as good or negative. Additionally, he computed the text's total positive or negative score. If the text has more positive than negative terms, it is deemed positive; otherwise, it is considered negative. Hu and Liu (2004) pioneered the development and implementation of a system that generates a feature-based summary from customer feedback. The suggested system carries out three functions: To extract product attributes from customer reviews, association rule mining is performed; WordNet is used to forecast the semantic orientations of opinion terms; and a feature-based summary is generated. Few methods for feature-based summarization have been suggested over the last two decades. The summarizers are used in a variety of areas, including product reviews, film reviews, local service evaluations, and hotel reviews.

III. PROPOSED ALGORITHM

The classification technique Naive Bayes is based on Bayes' theorem. The Naive Bayes Classifier's primary characteristic is a very strong independence assumption between condition and event. When two distinct occurrences occur, Bayes's theorem allows the conditional probability to be decomposed as follows:

$$P(A/B) = \frac{P(A) \cdot P(B/A)}{P(B)}$$

For text classification at the word feature level, the Naive Bayes assumption of attribute independence works effectively. When the number of characteristics is large, the independence assumption enables each attribute's parameters to be learnt independently, considerably simplifying the learning process.

3.1. Data collection

In this study, we utilized the reference dataset "Restaurant Reviews.tsv" for analyzing restaurant reviews. These evaluations are written in simple language and contain some slang and colloquial language. It includes both good and negative evaluations that are unique from one another. Classification models may be taught to understand the user's sentiment. We gathered data from 1,000 reviews of a restaurant and their associated feelings. There are two columns in this dataset. The first column contains text data denoted by the term "Review," while the second column has binary values denoted by the term "Liked." For example, if a review is favorable to the restaurant, as in a positive review, the associated mood is specified as "1". On the other hand, if a review is unfavorable to restaurants, like in a negative review, it is classified as "0."

```
data = pd.read_csv('Restaurant_Reviews.tsv', sep='\t', quoting=3)
```

Architecture :

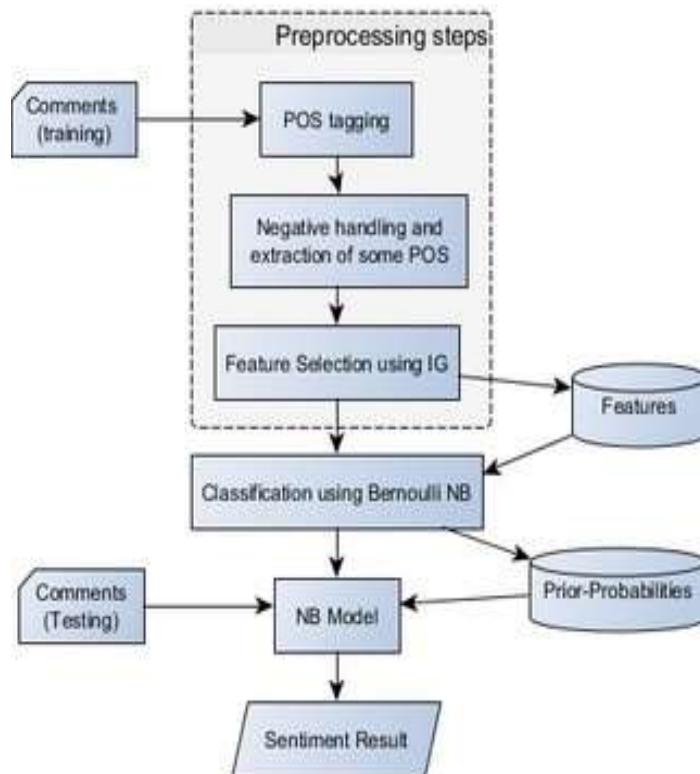


Figure 1. Architecture

3.2. Data preprocessing

Given that we are analyzing text data for sentiment analysis, data preparation is critical to ensure that the model understands the input. Text data is densely packed with noise. As a result, it's difficult to sanitize the texts intelligently. Pre-processing data considerably decreases the size of the input text documents. It occurs in a series of steps. Each review begins with a preprocessing phase that eliminates all ambiguous information, such as stop words, numeric, and special characters. We imported the NLTK library and from it, stop words and Porter Stemmer for eliminating stop words, numeric, and special characters. NLTK (Natural Language Toolkit) is the industry-standard Python framework for working with human language data. It provides intuitive interfaces to over 50 corpora and lexical resources, including WordNet, as well as a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, as well as wrappers for industrial-strength natural language processing libraries.

3.2.1.Importing Regular Expression:

A regular expression is a collection of characters that are mostly used to search for and replace patterns inside a string. When a machine stores a written language, it is often represented as a series of characters. Each word is a string. A string is a collection of words. Complete texts Additionally, there are strings, which include special characters for space and newlines. Occasionally, strings are prepared, for example, a "date string" such as 2002-06-23. Formatting whole texts is possible. For example, an email message with header fields followed by the message content. Texts may contain "markup," such as <abbrev>Phila</abbrev>, that contains information about the interpretation or display of certain sections of text. Thus, strings are common in language processing because they frequently include significant structures.

3.2.2.Stop-word elimination:

Stop-words are functional terms that appear often in the text's language (for example, "a," "the," "an," and "of" in the English language), making them unusable for categorization. We reduce stop-words by utilizing the Natural Language Toolkit package. We don't want these terms to eat up important storage space or processing time in our database. This is simply accomplished by keeping a list of words that you regard as stop words. In Python, NLTK stores a list of stopwords in 16 distinct languages.

3.2.3.Stemming:

Stemming is the process of reducing inflected words to their root (or stem) in information retrieval, such that similar terms map to the same stem. This approach decreases the number of words linked with each document organically, therefore reducing the feature space. In our tests, we employ a Porter stemming algorithm implementation. For instance, the English word "generalizations" would be stemmed as "generalizations → generalization → generalize → general → gener".

3.3. Bag of word model

A method for representing text data for machine learning algorithms, and the bag-of-words model aids us in this endeavor. The bag-of-words paradigm is straightforward to comprehend and use. It is a technique for extracting textual characteristics for use in machine learning algorithms. In machine learning, the act of transforming natural language processing text into numbers is referred to as vectorization. Potential features are extracted and translated into numerical format from the cleaned dataset. Vectorization is a method that converts textual input to numerical data. A matrix is generated via vectorization, with each column representing a feature and each row representing an individual review. In the first major phase of natural language processing, we not only cleaned all the reviews, but also built a corpus. Corpus is a term that refers to collections of texts. Our model corresponds to a corpus of 1000 cleansed reviews. We constructed the model bag from the corpus. The bag of models contains all of the corpus's distinct words. There are 1000 reviews in our corpus, and each review has one column for each unique term. Because 1000 reviews contain a large number of unique words, they have a large number of columns. We created a table with all the columns and a row count of 1000. We may simply eliminate duplicate values and data

redundancy by utilizing the Bag of Models. Each cell contains a number, which represents the frequency with which the relevant column appears in the review. For instance, the first column has the phrase "wow love location," which results in a value of 1 for the "wow" cell. However, the second row contains no instances of "wow," resulting in a value of "0" for the "wow" cell in the second row. This is the method through which the Bag Model was created.

3.4. Splitting Dataset

Splitting the data set into two halves is a critical component of the Machine Learning model.

3.4.1. Training Set

3.4.2. Evaluation Set

Machine Learning's primary objective is to generalize beyond the data examples used to train models. We wish to test the model to determine the quality of its pattern generalization on un-trained data. However, because future instances will have unknown target values and we will be unable to verify the accuracy of our predictions for future instances at this time, we will need to use some of the data for which we already know the answer as a proxy for future data, which will be referred to as our Test Set. When dealing with big datasets, the most common method is to divide them into training and test subsets, often with a ratio of 70-80% for training and 20-30% for testing. The Train test split function, which is loaded from the scikit-learn package, does this splitting randomly.

3.4.1. Training Set :

We've incorporated 80% of the data from 1000 reviews into our train set. Both the independent variable (x_{train}) and the dependent variable (y_{train}) are known in the training set.

3.4.2. Testing Set :

The test set contains 20% of the data from 1000 reviews, where the dependent variable is denoted by (x_{test}) and the independent variable is denoted by (y_{test}).

Splitting:

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
```

3.5. Fitting Algorithm to Training Dataset

Classification is a type of supervised learning, which occurs when a training set of properly recognized observations is provided. A classifier is an algorithm that performs classification, particularly in a concrete implementation. Occasionally, the word "classifier" refers to the mathematical function implemented by a classification algorithm that categorizes incoming data. It might be challenging to identify an excellent, or even a well-performing, machine learning method for a given dataset. We used trial and error to determine which algorithm produces the best results. We will demonstrate and discuss

the effectiveness of Bernoulli's Naive Bayes classifier in predicting whether restaurant reviews are positive or negative.

```
from sklearn.naive_bayes import BernoulliNB
```

```
BEclassifier = BernoulliNB(alpha=0.8)
```

```
BEclassifier.fit(x_train, y_train)
```

3.6. Prediction of the Result

The Machine Learning system utilizes the training data to train models to recognize patterns, and the test data to assess the trained model's prediction ability. The machine learning system measures predictive performance by comparing predicted values on the evaluation data set to real values using a number of criteria. We will use Bernoulli's Naive Bayes method to forecast the test outcome in terms of the value of `y_pred_be`.

```
y_pred_be = BEclassifier.predict(x_test)
```

3.7. Evaluation Metrics

True Positive (TP):

A true positive (TP) is an outcome for which the model accurately predicts the positive class.

True Negative (TN):

A true negative (TN) is an outcome for which the model accurately predicts the negative class.

False Positive (FP):

A false positive occurs when the model forecasts a positive class erroneously.

False-Negative (FN):

A false negative occurs when the model forecasts a negative class erroneously.

Precision:

It is the ratio of genuine positives to false positives and vice versa. This value indicates the proportion of correctly recognized items in a class.

$$\text{Precision (P)} = \text{TP} / (\text{TP} + \text{FP})$$

Recall:

It's the proportion of true positives to the total number of true positives and false negatives. This defines how many items in a class are incorrectly categorized.

Recall (R) = TP / (TP+FN)

Accuracy:

It's calculated by multiplying the sum of all True positives and True Negatives by the total number of test cases. This metric assesses the overall accuracy of the classifier.

Accuracy = (TP+TN) / (TP+TN+FP+FN)

Confusion Matrix:

One technique for evaluating the performance of classifiers is to use a confusion matrix, in which the number of properly classified examples equals the sum of the matrix's diagonals; all other instances are wrongly categorised. When referring to the counts calculated in the confusion matrix, the following terminology is frequently used.

```
from sklearn.metrics import accuracy_score

from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

print(accuracy_score(y_test, y_pred_be))

print(confusion_matrix(y_test, y_pred_be))

print(classification_report(y_test, y_pred_be))
```

IV. RESULT

In the beginning, we analyzed the "Restaurant Review.tsv" file, which contains 1000 restaurant reviews. The overall number of positive and negative reviews is the same in this case, implying that 500 are favorable and 500 are negative. Operations like Data collection, Data Preparation, Bag of words model, Fitting Algorithm to Training Dataset are performed, and then Bernoulli's Naive Bayes technique is used to categorize a review as positive or negative, yielding an accuracy score of 77.5 percent.

Accuracy Score:

```
print(accuracy_score(y_test, y_pred_be))
0.775
```

Figure 2. Accuracy Score

Confusion Matrix:

```
print(confusion_matrix(y_test,y_pred_dst))
```

```
[[74 23]
 [35 68]]
```

Figure 3. Confusion Matrix

Classification Report:

```
print(classification_report(y_test,y_pred_dst))
```

	precision	recall	f1-score	support
0	0.68	0.76	0.72	97
1	0.75	0.66	0.70	103
accuracy			0.71	200
macro avg	0.71	0.71	0.71	200
weighted avg	0.71	0.71	0.71	200

Figure 4. Classification Report

V. CONCLUSION

After analysing a huge corpus of reviews, we conclude that Bernoulli's NB model outperforms competing methods on nearly every assessment criteria. Fitting and predicting the output takes relatively little time, which is why it may be utilised for real-time classification systems. We propose Bernoulli's Naive Bayes Classifier Model for sentiment analysis in this thesis. This model may be used to analyse the sentiment of any type of text data, including tweets, brand/product reviews, and vacation destination reviews. This model was tested on a dataset of 1000 restaurant reviews.

Sentiment analysis is critical for customers and service providers alike. Now, in the modern era of the internet and globalisation, both customers and service providers are curious about the general public's view of a certain brand/product/location, etc. It benefits the service provider since it includes a business component, but it also benefits customers because it assists them in selecting the finest product. We have concluded from our thesis work that the Bernoulli NB Classifier is an excellent machine learning model for sentiment analysis. It improves sentiment analysis prediction. It is a significant difficulty in the field of sentiment analysis to analyse a sarcastic review/text. A machine might be capable of detecting sarcasm. Future studies might concentrate on sarcastic expressions, which are notoriously difficult to comprehend, both for humans and computers. Another difficult issue is detecting spam content in user reviews. Finally, a review is designated as one of several kinds.

Future Scope -

Because the reviews are created with a mixture of real-life review data and sarcastic phrases, review mining is a difficult process. Furthermore, sentiment classification may be done at three levels. Only document-level sentiment classification is performed by this system. Following that, this system uses purely statistical approaches to get an edge. The accuracy of the results might be enhanced by combining the use of semantic resources such as WordNet and SentiWordNet with a statistical method. Furthermore, because this method only examines the sentiment classification of subjective comments, a subjectivity function that can identify whether a statement is an objective or subjective may be added.

Acknowledgment

I would like to express my deep gratitude to my professor, Dr. B. Indira Reddy, for guiding me through the process of writing this research paper.

REFERENCES

- [1] Danescu-Niculescu-Mizil, C.; Kossinets, G.; Kleinberg, J.; Lee, L. How opinions are received by online communities: A case study on amazon.com helpfulness votes. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 141–150.
- [2] Guo, Y.; Wang, Y.; Wang, C. Exploring the Salient Attributes of Short-Term Rental Experience: An Analysis of Online Reviews from Chinese Guests. *Sustainability* 2019, 11, 4290.
- [3] Pan, Y.; Zhang, J.Q. Born unequal: A study of the helpfulness of user-generated product reviews. *J. Retail.* 2011, 87, 598–612.
- [4] Nam, S.; Ha, C.; Lee, H. Redesigning In-Flight Service with Service Blueprint Based on Text Analysis. *Sustainability* 2018, 10, 4492.
- [5] Kim, W.G.; Li, J.J.; Brymer, R.A. The impact of social media reviews on restaurant performance: The moderating role of excellence certificate. *Int. J. Hosp. Manag.* 2016, 55, 41–51.
- [6] Baek, H.; Ahn, J.; Choi, Y. Helpfulness of online consumer reviews: Readers' objectives and review cues. *Int. J. Electron. Commer.* 2012, 17, 99–126.
- [7] Otterbacher, J. "Helpfulness" in online communities: A measure of message quality. In Proceedings of the Sigchi Conference on Human Factors in Computing Systems, Boston, MA, USA, 4–9 April 2009.
- [8] Pantelidis, I.S. Electronic meal experience: A content analysis of online restaurant comments. *Cornell Hosp. Q.* 2010, 51, 483–491.
- [9] Yan, X.; Wang, J.; Chau, M. Customer revisit intention to restaurants: Evidence from online reviews. *Inf. Syst. Front.* 2015, 17, 645–657.
- [10] Kwok, L.; Xie, K.L. Factors contributing to the helpfulness of online hotel reviews: Does manager response play a role? *Int. J. Hosp. Manag.* 2016, 28, 2156–2177.
- [11] Jia, S.S. Leisure Motivation and Satisfaction: A Text Mining of Yoga Centres, Yoga Consumers, and Their Interactions. *Sustainability* 2018, 10, 4458.
- [12] S Chandrakala and C Sindhu, (2012), "Opinion Mining and sentiment classification a survey", *ICTACT journal on soft computing*, volume: 03, issue: 01, pp. 420-427.

- [13] J. Kittler, (1998), "Combining classifiers: a theoretical framework", Pattern Analysis and Applications, 1, pp.18–27.
- [14] Freund, Y. and Schapire, R. (1996), "Experiments with a new boosting algorithm", In Proceedings of the Thirteenth International Conference on Machine Learning, Bari, Italy, pp.148-156.
- [15] Kim S., Han K., Rim H., and Myaeng S. H. (2006), "Some effective techniques for naïve bayes text classification", IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 11, pp. 1457-1466.
- [16] Nolan Miller, Paul Resnick, Richard Zeckhauser, (2005), "Eliciting Informative Feedback: The Peer-Prediction Method" management science Vol. 51, No. 9, pp. 1359–1373
- [17] Freund, Y. and Schapire, R. (1995), "A decision-theoretic generalization of on-line learning and an application to boosting", In proceedings of the Second European Conference on Computational Learning Theory, pp. 23-37.
- [18] Gayatree Ganu, Noémie Elhadad, Amélie Marian, (2009), "Beyond the Stars: Improving Rating Predictions using Review Text Content", Twelfth International Workshop on the Web and Databases, Providence, Rhode Island, USA
- [19] T. Ho, J. Hull, S. Srihari, (1994), "Decision combination in multiple classifier systems", IEEE Transactions on Pattern Analysis and Machine Intelligence, 16, pp. 66–75.
- [20] Mullen, Tony and Nigel Collier. (2004), "Sentiment analysis using Support Vector Machines with diverse information sources", In Dekang Lin and Dekai Wu, editors, Proceedings of EMNLP, PP. 412-418, Barcelona, Spain. Association for Computational Linguistics.