**CS689A PROJECT REPORT**

**ON**

# SENTIMENT ANALYSIS ON HINDI TEXT

**Course Instructor:**

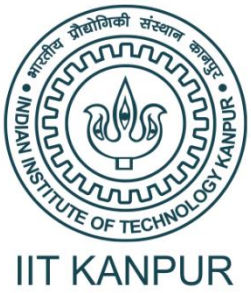
**Prof. Arnab Bhattacharya**


**Submitted by:**

**Allan Robey 22111007**

**Avnish Tripathi 22111014**

**Divyesh Tripathi 22111020**



**Department of Computer Science and Engineering**


**2022-2023**

# Contents

## ABSTRACT

Sentiment Analysis has been gaining a lot of attention since the last few years. Sentiment Analysis is a text classification task which is currently a vital research area in the field of web-based mining. To seek out other people's opinions prior to making a decision is normal human behaviour. There are many documents accessible that convey perspectives on many topics. However, the greatest difficulty lies in deciphering these materials in order to generate usable knowledge.

There are enormous works in the field of sentiment analysis that are available in English. There hasn't been much effort done on Indian languages, though. Sentiment analysis in Hindi is necessary since opinion-rich sites have exploded in Hindi during the past few years.

In this project, we've divided Hindi Text into positive, negative and neutral categories. For the purpose of annotating the data, the deep learning model such as Vader, Roberta, Bert and TS were used and their accuracies were compared. For feature matrix generation, four techniques have been incorporated: TF-IDF, Bag of words and BM25 and their accuracies on different classifiers such as Support Vector Machine, Gaussian Naïve Bayes' and Logistic Regression.

# 1. INTRODUCTION

The technique of extracting and comprehending the sentiments described in the text is known as sentiment analysis of document. The proliferation of information on social media platforms like Facebook, Twitter, and LinkedIn has offered users new ways to voice their opinions about specific goods, people, and locations. The user's opinion is always shown as textual data. Millions of text messages are sent every day via social media or online retailers. It is vitally important to look into and analyse the sentiment of the opinion. To ascertain if an opinion is positive, negative, or neutral, text analytics and NLP with artificial intelligence capabilities are applied. Opinion mining and sentiment analysis are independent of any specific platform or domain. It propagates through all social media platforms. It is particularly beneficial for the expansion of many businesses and organisations in the fields of healthcare, management, the economy, and countless others. Analysis of sentiment is additionally offering business intelligence that may be used to make smart, effective decisions. Analysis of Sentiment and categorization of sentiment are the two techniques for opinion mining. However, both have unique, autonomous characteristics, yet occasionally both can be used in place of each other. Sentiment classification depicts the emotional tenor of putting the document's class labels on, or segment. Sentiment orientation denotes the subjectivity-based polarity of an opinion, whether it is true or wrong. Determining whether the given text or reviews data is subjective or objective in nature is a process called subjective analysis.

Most of the research in this domain has been focused on English language. However, today there is an explosive growth of information on the web in native Indian languages such as Hindi, Marathi, Gujarati, Malayalam, Tamil, etc. This data will be of no use if they are not classified on the basis of their sentiments. This has motivated us to perform Sentiment Analysis of Hindi Text in our project.

**Challenges in performing sentiment analysis of Hindi text are as follows:**

- When compared to English, Hindi is a free-order language with a rich morphology. It signifies that the word order in Hindi is flexible, with the subject, object, and verb appearing in any sequence, in contrast to English, which has a fixed word order in which the subject is always followed by the verb and then the object. When determining the polarity of a text, word order is crucial.
- There is an unavailability of well annotated standard corpus.

- There is scarcity of resources in Hindi language, which makes it difficult to collect data and create the necessary annotated datasets.
- Furthermore, this language lacks effective taggers and parsers.

## 2. RELATED WORK

- The collected tweets had no particular format, no specific sentiment labels but were of a particular size. The tools like R, Rapid Miner's ALYLIEN extension were used. The output was deemed valid based on the results, and decisions were taken. [1]
- Sentiment analysis were performed on patient records. The performance of the model was evaluated by using lexicon scoring and machine learning models. It gave better understanding and better visualization of patient records than existing models at that time. [2]
- The opinions were classified as positive or negative by observing and monitoring the threshold values with the help of HADOOP and it gave an accuracy of 96% for the tweets. [3]
- A sentiment analysis mechanism for product assessments has been suggested. The complete procedure is divided into three parts. The techniques of classification Gaussian Naïve Bayesian, Random Forest and support vector machines models are chosen for categorisation. [4]
- Detecting the stance of tweets on social media has also been performed using sentiment analysis. The goal of the paper is to ascertain the author's viewpoint on a particular subject. They suggested fusing feature vector and sentiment data in this study. The suggested paradigm performs well for brief papers and tweets. A separate collection of classifiers, including the support vector machine (SVM) technique, were used to test the model. The system performed better when evaluated for various tasks. [5]
- In this study, text blob, a Python library, was used to carry out the sentimental analysis. Facebook, Twitter, and news websites are the three venues that were used to examine these sentiments. ANN is used to classify the tweets (Artificial Neural Network). The tweets were gathered using the Twitter API, and they were then categorised using the Naive Bayes algorithm. R programming has been used to examine the findings. For a huge amount of data in a short length of time, an accuracy of 70% to 89% is reached.[6]

## 3. PROBLEM STATEMENT

The amount of information available on the internet is growing rapidly all around the world right now. Information provision is not the main issue; rather the inputs are knowledge-starved. People expressing their opinions on social media in their native languages like Hindi has also increased rapidly. People tend to express their views on various fields such as movies, reviews of products, places, etc in their regional languages like Hindi has increased tremendously. So, there is an urgent need to analyse these data so as to determine the opinion of the people based on the sentiment of the text and gain knowledge related to the respective fields. It can then help the business people by tackling the customer's opinion about the particular product so that they can modify changes if it is required. In order to perform sentiment analysis, there is a need of a well annotated data. However, there is a scarcity of resources in Hindi language, which makes it difficult to collect data and create the necessary annotated datasets. The project aims to create a well annotated corpora for Hindi language as well as determine the most efficient method for feature matrix generation. The project also aims at determining the most efficient method for classifying the Hindi text into different classes such as positive, negative and neutral based on the sentiment.

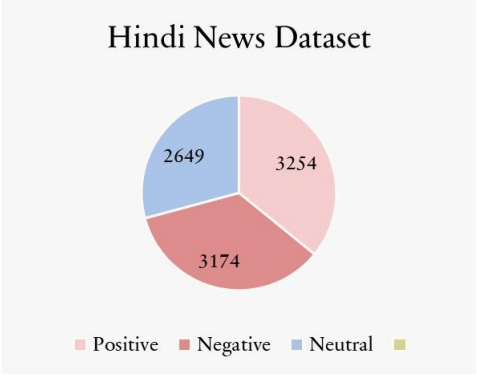
# 4. METHODOLOGY

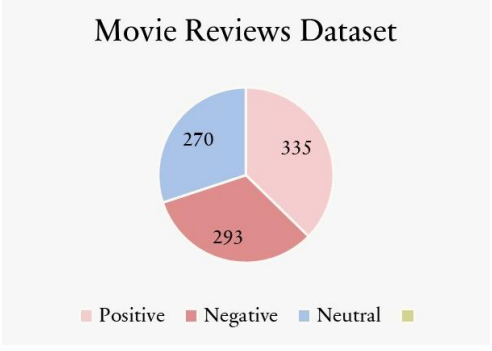
## 4.1 DATASET

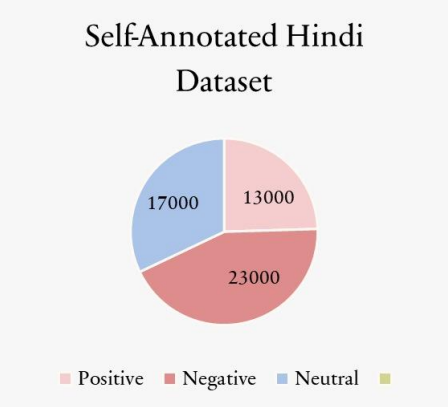
- We have used 3 different types of datasets in our project.
- Firstly, we have used dataset of 9077 labelled sentences in order to determine the most efficient deep learning model to annotate the Hindi dataset which comprises of 1,00,000 sentences.
- The annotated Hindi dataset comprising of 1,00,000 sentences and movie reviews dataset from Kaggle comprising of 898 reviews of which 335 reviews were positive, 293 were negative reviews and 270 reviews were neutral. Both of these datasets were used to perform classification task.

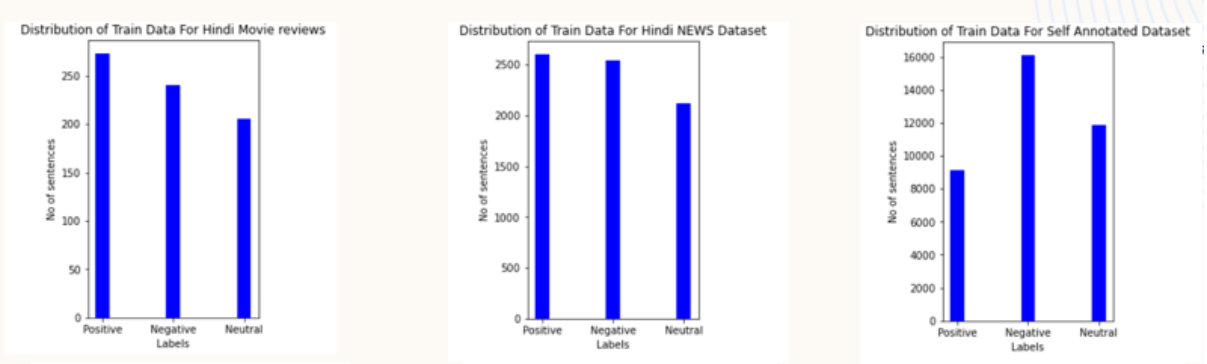
| **Dataset** | **No of Sentences** | **Purpose** |
|---|---|---|
| News Dataset in Hindi | **9077** | To determine the most efficient model for data annotating task |
| Movie Reviews in Hindi<br>https://www.kaggle.com/datasets/disisbig/hindi-movie-reviews-dataset | **898** | To perform classification task |
| Hindi Text | **1,00,000** | Annotated using the determined most efficient deep learning model to perform the classification task |

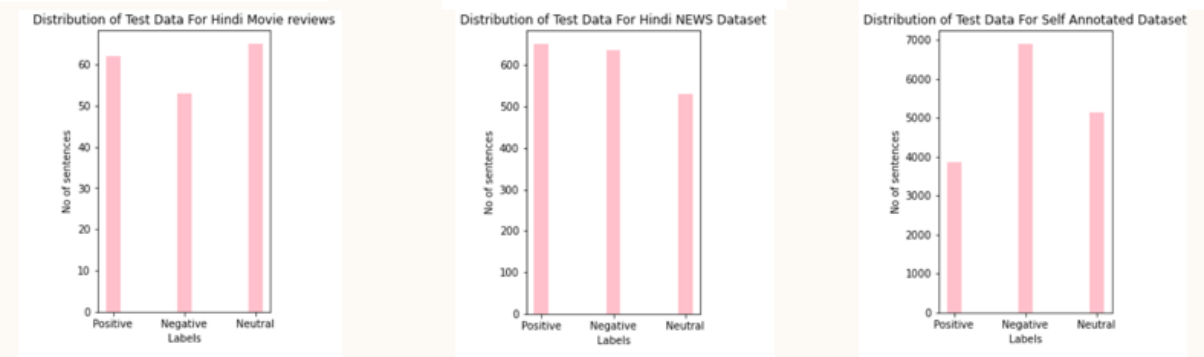
The figure below indicates the distribution of different dataset used in this project-




Self-Annotated Hindi Dataset
17000
13000
23000
Positive
Negative
Neutral


Distribution of Train Data For Hindi Movie reviews
Distribution of Train Data For Hindi NEWS Dataset
Distribution of Train Data For Self Annotated Dataset
No of sentences
Positive
Negative
Neutral
Labels


Distribution of Test Data For Hindi Movie reviews
Distribution of Test Data For Hindi NEWS Dataset
Distribution of Test Data For Self Annotated Dataset
No of sentences
Positive
Negative
Neutral
Labels

### 4.2 DATA LABELLING PROCESS

In order to annotate to Hindi text data different pre-trained deep learning models such as Vader, Roberta, Bert and TextBlob were used. The accuracies of the following deep learning model were checked on dataset of 9077 labelled sentences. They were labelled as positive, negative and neutral. The accuracies of the pre-trained deep learning model are shown in the table below:

| SR.NO | PRE-TRAINED MODEL | ACCURACY |
|---|---|---|
| 1. | Vader Sentiment Scoring | 62.89% |
| 2. | Roberta Pre-trained Model | 73.56% |
| 3. | bert-base-multilingual-uncased-sentiment | 65.28% |
| 4. | Text Blob | 48.53% |

From the above table, it can be clearly observed that Roberta pre-trained model had the best performance. Thus, it has been chosen as the pre-trained deep learning model in order to annotate the 1,00,000 Hindi sentences as positive, negative and neutral based on their sentiment. In order to annotate the data, the Hindi text were initially translated to English using GoogleTranslator from deep_translator. Then the pre-trained deep learning model was applied on the annotated data. Out of 1,00,000 sentences 53,000 sentences were considered of which comprises of 13,0000 sentences belonging to positive class, 23,000 sentences belonging to negative class and 17,000 sentences belonging to negative class.

### 4.3 DATA PRE-PROCESSING

In this phase, we have eliminated any punctuation, digits, one-length words, and stop words from each sentence that do not improve the classification's accuracy.

### 4.4 FEATURE MATRIX GENERATION

We generate the feature matrix using the TF-IDF, Bag of words approach, BM25 and n-gram model after data pre-processing is complete**.**

#### (a) TF-IDF –

We used the scikit-learn library's TfidfVectorizer() function in our code. It is utilised to turn a group of unprocessed documents into a matrix of TF-IDF features. The purpose of employing TF-IDF instead of the raw frequencies of occurrence of a token in a specific document is to

scale down the influence of tokens that occur very frequently in a given corpus and are thus empirically less informative than features that appear in a small proportion of the training corpus.

**(b) Bag of Words –**

Every algorithm we use in NLP operates on numbers. Our text cannot be entered into the algorithm directly. As a result, the text is pre-processed using the Bag of Words model, which creates a bag of words from it and keeps track of how many times the most common words are used overall. Using a table that shows the number of words that correspond to each word, this model may be seen. Prior to analysis, the data will be pre-processed, meaning that all capitalization, non-word characters, and punctuation will be removed. Having gathered the words that appear most frequently in our text, we will establish a dictionary as our word bank. Next, each sentence will be tokenized to words. We will now look up each word in the statement in our dictionary to see if it is present. In that case, we add one to its count. If not, it is added to our dictionary and its count is set to 1. In order to determine whether a word in a sentence is often or not, we now build a vector. If a term occurs frequently in the sentence, we set it to 1, otherwise we set it to 0.

**(c) BM25 –**

BM25 is a simple Python package and it has been used to index the data. Firstly, word-by-word breakdown of the statement is performed such that each word can be regarded separately. The special characters and stop words are removed from the text. Then the text is tokenized and BM25 is executed.

### 4.5 CLASSIFICATION

In order to perform classification techniques such as Gaussian Naïve Bayes, Logistic Regression and Support Vector Machine.

(a) **Gaussian Naïve Bayes –**

Naive Bayes is a classification method that relies on the independence of predictors assumption and is based on Bayes' Theorem. It makes the supposition that a certain feature's presence in a class is unconnected to the existence of any other feature. The number of parameters required by Naive Bayes-based classifiers is linear in the number of features or predictors in a learning task, making them extremely scalable.

**(a) Logistic Regression –**

The cumulative logistic distribution is the logistic function used in logistic regression to estimate probabilities and determine the relationship between the categorical dependent variable and one or more independent variables. Although the logistic function is used to alter the predictions, logistic regression is a linear function. A dataset with one or more independent variables that affect the outcome can be analysed statistically using this technique.

**(b) Support Vector Machine –**

A supervised learning model called a support vector machine (also known as a support vector network) contains different learning algorithms to study classification and regression issues. It is also referred to as a binary classifier since it looks for a hyperplane that can distinguish two classes of data with the greatest degree of precision. SVM creates a (N-1) dimensional hyperplane to divide a set of two-type points in N-dimensional space into two groups.

**(c) Kernalized Support Vector Machine**

The Support Vector Machine uses a set of mathematical operations that provide the kernel the ability to manipulate the data. In order for a non-linear decision surface to turn into a linear equation in a higher number of dimension spaces, Kernel Function often converts the training set of data.

## 5. EXPERIMENTAL SETUP AND RESULTS ANALYSIS

In this project, we have used three different datasets in Hindi language. The Hindi news dataset comprises of 9077 sentences of which 3174 were of negative class, 3254 were of positive class and 2649 were of neutral class respectively. The Hindi movies reviews dataset from Kaggle comprises of 898 sentences of which 293 were of negative class, 335 were of positive class and 270 were of neutral class respectively.

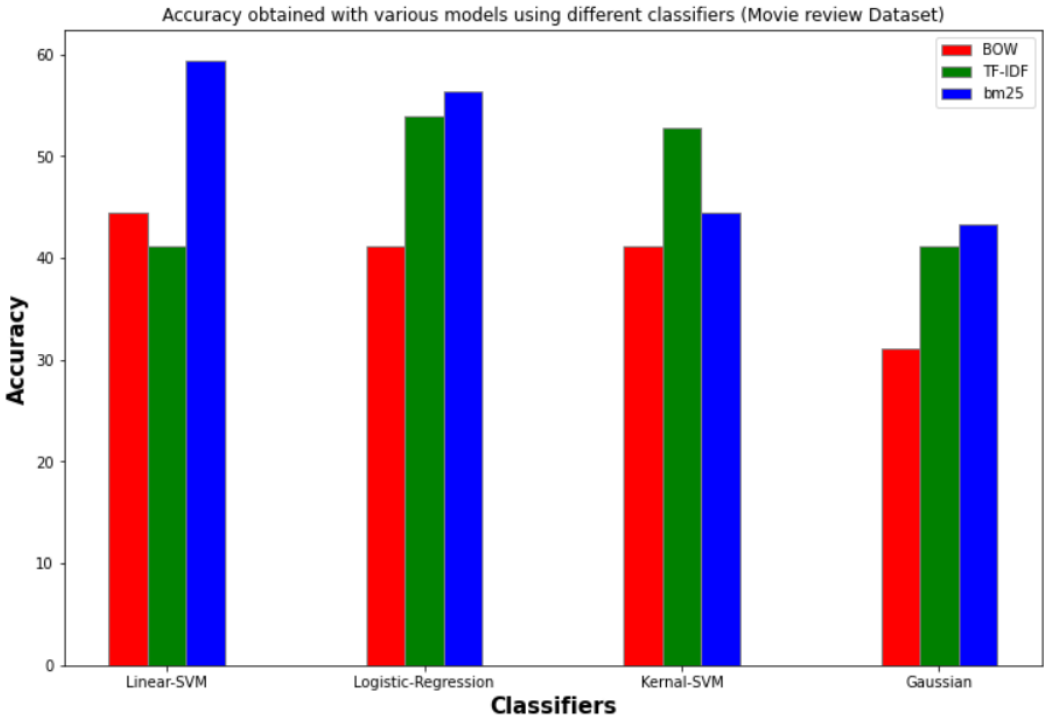
The pre-trained deep learning models such as Vader, Roberta, TextBlob and bert-base-multilingual-uncased-sentiment model were selected to annotate the Hindi unlabelled dataset. The performance of these deep learning models was determined on the Hindi News dataset. The pre-trained Roberta deep learning model has been determined as the most efficient model for annotating the Hindi Dataset comprising of 1,00,000 sentences as it gave the highest accuracy of 73 %. In order to validate the accuracy of the model we have also manually checked the predicted label of 1,000 sentences of which 869 were found to be correct.

In order to annotate the data, the sentences had to be initially translated to English language from Hindi language using GoogleTranslator from deep_translator. However, translating 1,00,000 sentences using GoogleTranslator is a tedious and cumbersome task as it is extremely time consuming. Hence, the data has been split sequentially into different parts and has been translated on multiple systems in order to speed-up the process of translating 1,00,000 Hindi sentences. Once the data has been translated to English, the translated data has been fed into the Roberta Deep learning model in order to annotate the data. As it an intensive task, it has carried out using GPU in order to enhance the speed-up.

Once the Hindi data has been annotated as positive, negative and neutral using the deep learning model then the accuracy obtained by various models such as TF-IDF, Bag of Words and BM25 using different classifiers such as Gaussian Naïve Bayes, Logistic Regression, SVM and Kernalized SVM has been analysed.

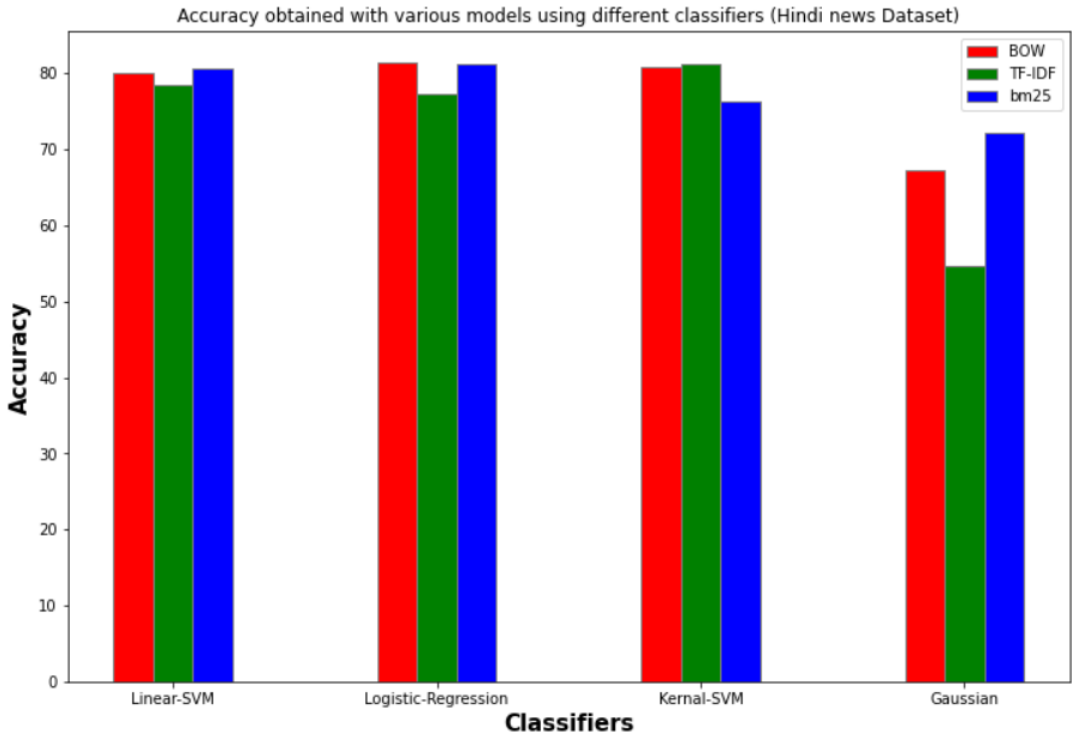
**Accuracy obtained with various models using different classifiers Movie Reviews**

| Models used | Gaussian Naïve Bayes | Logistic Regression | Support Vector Machine | Kernalized SVM |
|---|---|---|---|---|
| **Bag of Words** | 31.11% | 41.11 % | 44.44 % | 41.11 % |
| **TF-IDF** | 41.11 % | 53.33 % | 53.88 % | 52.77 % |
| **BM25** | 43.33 % | 58.33 % | 59.44 % | 44.44 % |



**Accuracy obtained with various models using different classifiers Hindi News Dataset**

| Models used | Gaussian Naïve Bayes | Logistic Regression | Support Vector Machine | Kernalized SVM |
|---|---|---|---|---|
| Bag of Words | 67.29 % | 81.49 % | 80.12 % | 80.78 % |
| TF-IDF | 54.625 % | 77.2 % | 78.48 % | 81.27 % |
| BM25 | 72.68 % | 81.27 % | 80.61 % | 76.21 % |



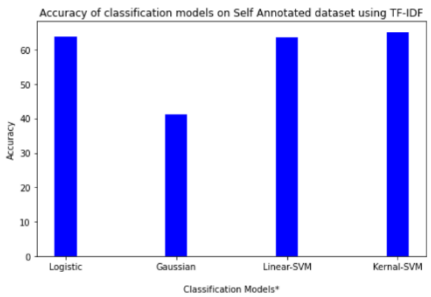
**Accuracy obtained with various models using different classifiers Self-Annotated Hindi Dataset**

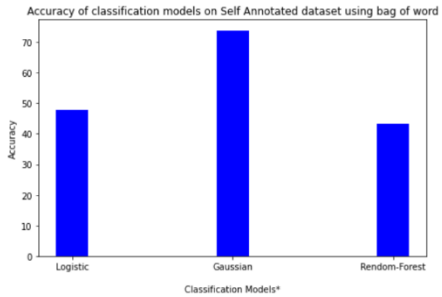
1. **TF-IDF**

| Classifier | Accuracy |
|---|---|
| Gaussian Naïve Bayes | 41.11 % |
| Logistic Regression | 63.79 % |
| Linear SVM | 63.52 % |
| Kernelized SVM | 64.88 % |

2. **Bag of Words**

| Classifier | Accuracy |
|---|---|
| Gaussian Naïve Bayes | 73.56 % |
| Logistic Regression | 47.68 % |
| Random Forest | 43.39 % |



## 6. CONCLUSION

In this project we have focused majorly on analysing three aspects of sentiment analysis. Firstly, collecting the Hindi dataset from various sources and creating a well annotated Hindi dataset. Secondly, determining the most efficient deep learning model to annotate the data among pre-trained deep learning models such as Vader, Roberta, TextBlob and bert-base-multilingual-uncased-sentiment model. The results have proven that Roberta is the most efficient deep-learning model to annotate the Hindi data among the mentioned models. Thus, in this project we have successfully annotated 1,00,000 Hindi sentences using Roberta pre-trained model. Finally, most efficient method of featurization among Bag of words, TF-IDF and BM25 for various classifiers such as Gaussian Naïve Bayes, Logistic Regression, SVM and Kernelized SVM has been determined successfully. The best classifier to perform sentiment analysis of Hindi text for different methods of featurization has also been determined successfully.

## 7. FUTURE SCOPE

- The project has been limited to perform sentiment analysis on Hindi language only.
- The project can be extended to perform sentiment analysis on many unrecognized and low-resourced Indian languages in the future.
- The size of the Hindi corpora can be extended to incorporate more data.
- The project can be extended to incorporate and determine other efficient methods in order to annotate the data.

## REFERENCES

[1] R. Thilagavathi and M. K. Krishnakumari, “Tamil English language sentiment analysis system.” [Online]. Available: www.ijert.org. [1] K.M.A. Kumar, N. Rajasimha, M. Reddy, A. Rajanarayana, K. Nadgir, Analysis of users’ sentiments from Kannada web documents, Proc. Comput. Sci. 54 (2015) 247– 256, doi:10.1016/j.procs.2015.06.029.

[2] T.S. Raghavendra, K.G. Mohan, Web mining and minimization framework design on sentimental analysis for social tweets using machine learning, Proc. Comput. Sci. 152 (2019) 230–235, doi:10.1016/j.procs.2019.05.047.

[3] K.M.A. Kumar, N. Rajasimha, M. Reddy, A. Rajanarayana, K. Nadgir, Analysis of users’ sentiments from Kannada web documents, Proc. Comput. Sci. 54 (2015) 247– 256, doi:10.1016/j.procs.2015.06.029.

[4] Xing Fang ,Justin Zhan ,"Sentiment Analysis using product review data",Springer: Journal of Big data",2015,North Carolina A& T State university,Greensboro,NC,USA.

[5] A. Sharma, U. Ghose, Sentimental analysis of Twitter data with respect to general elections in India, Proc. Comput. Sci. 173 (2020) 325–334 no. 2019, doi:10.1016/j.procs.2020.06.038.

[6] S. Paliwal, S.Kumar Khatri, M. Sharma, Sentiment analysis and prediction using neural networks, in: Proceeding of the International Conference on Inventive Research in Computing Applications ICIRCA, 2018, pp. 1035–1042, doi:10.1109/ICIRCA.2018.8597358. no. Icirca2018.