Studying the Effect of Vectorization Techniques in Mix-Code (Hinglish Language) on Open-Source Data Using Machine Learning and Transfer Learning Methodology.

2022

by

Murthy S Routhula
Under the supervision of
Dr. Abhishek Kaushik

School of Informatics and Creative Arts
Department of Computing Science and Mathematics

Acknowledgment

This paperwork is not supported by any organization. This is intended for only a knowledge-gaining basis, and I can take initiative for this future work. Special gratitude and thanks I give to my project coordinator Dr. Abhishek Kaushik, Lecturer for M.Sc. Data Analytics, Department of Computer Science and Mathematics, Dundalk Institute of Technology, Dundalk, Ireland for simulating suggestions and encouragement, helping me in this research.

I am grateful to all the lecturers from M.Sc. Data Analytics Course, Dr. Jack McDonnell, Dr. Peadar Grant, Dr. Siobhan Connolly Kernan, and Dr. Rajesh Jaiswal for their valuable lectures and teachings about Data Analytics skills during all the lectures, labs, and projects.

Table of Contents

Page No.

Abstract:

One of the popular virtual learning sources in the present world is YouTube which has been accessed by billions of Internet users. Due to its popularity, the number of YouTubers has increased. Generally, people show their intentions about the videos posted on YouTube through comments. India has a population of 1.4 billion (India Population (2022) - Worldometer 2022) and has nearly 121 languages and 270 mother tongues (Jo Hartley 2021). Hindi is one of the most spoken languages in India. Indians mostly use Mix-Code language in commenting i.e., Hinglish which is the combination of Hindi and English languages. This project is useful in analyzing the Mix-Code YouTube comments given by users for the videos posted by YouTubers. It helps in knowing the intention of users according to the video content and helps YouTubers to post videos with better quality and content. Different Vectorization techniques using Bag-of-Words models and Pre-trained Transformer models are applied to the datasets to transfer comments to features. Both parametric and non-parametric models are trained using these vectorized datasets along with labels which include different classes like Questions, Suggestions, Gratitude, etc. Different combinations of Vectorizers and Machine Learning Models are conducted to pick the best model based on the different evaluation methods for the Hinglish Mix-code. Standard Scaled Verloop BERT Hinglish (Sentence Transformer) – SVM Model has shown the best performance for Kabita's Dataset and Nisha's Dataset with 82% and 78% accuracies respectively.

Keywords:

Natural Language Processing, Sentimental Analysis, YouTube, Internet, Mix-Code, Hinglish, Machine Learning, Vectorization, Evaluation methods.

1. Introduction

YouTube is an online video-sharing social media platform that started on 14th February 2005 and is owned by Google on(Matthew Johnston 2022) November 13, 2006. It has billions of monthly users who watch videos for billions of hours collectively for their requirements. As it is one of the best learning and research platforms, it has expanded into mobile platforms too (William L. Hosch 2022). The videos on YouTube include short films, movies, documentaries, cooking channels, educational and technological related, etc. Everyone has their food preferences. Especially international students who have habituated to home food learn to cook food themselves using YouTube videos. Due to this reason, many YouTubers started doing videos based on cooking different cuisines which some channels are very popular for their unique content. To know about the viewers' intentions and feedback on the videos, they must manually read the comments and prepare for the next video and improve. This will take a lot of time if comments are more than hundreds. This project can help in finding the nature of the comment user has given for the uploaded video instead of manual reading. This will be achieved by training the model with different types of comments with labels to understand the patterns and predict the new comments label.

This Project comes under Sentimental Analysis using Natural Language Processing popularly known as NLP. NLP started in the 1950s and is supported by Alan Turing's article titled "Computing Machinery and Intelligence" popularly known as "Turing Test" which automates the assumptions and generation of Natural Language (Natural Language Processing - Ela Kumar - Google Books n.d.). "*Sentimental Analysis which is also called opinion mining is Natural Language Processing technique used to determine whether the text data is positive or negative or neutral*" (Sentiment Analysis Guide 2020). These texts may be extracted from different comments, reviews, paragraphs, etc. It is mainly applied to social media, surveys, customer services, etc. In NLP as the natural language is processed which is stored in the form of documents or tables, the main words are extracted and used to get the opinion of the text. These words are converted to vectorized forms using different vectorization methods as mathematical calculations can be done on numerical data. This vectorized data will be

trained to Machine Learning (ML) model. Generally, Classification models are integrated into the Natural Language Processing processes. This is because different texts should be classified based on the nature of the text data which may be positive or negative or neutral. As labels will be provided for training the model, Supervised learning will be applied in this project.

Machine Learning (ML) is a term introduced by Arthur Samuel in 1952 while he was writing the computer program to play checkers game (A Short History of Machine Learning -- Every Manager Should Read n.d.). It involves mainly two types of learning namely Supervised and Unsupervised.

- In Supervised Learning, the Machine Learning models are trained on data called training data that consists of already assigned labels. Then the model is tested using test data to check the prediction capacity. The evaluation is conducted based on the actual test results and predicted results to check the accuracy of the models.
- In Unsupervised Learning, no labels will be provided, and the data will be clustered based on the patterns recognized in the model. In this project, the data has Mix-Code textual comments, and labels were assigned based on the type of comment, Supervised Learning models are trained with the vectorized Mix-Code text along with the labels.

Mix-Code languages consist of two or more language varieties while using. This type of language can be usually observed in general conversation, the local language, comments, reviews, etc. Hinglish is one of its types and it is a mix of Hindi and English Languages as shown in Figure 1. Red colour font words belong to Hindi language vocabulary and blue colour font words belong to English vocabulary. They are both used to form a meaningful sentence whose meaning can be seen. The data consists of most of these types of comments. There are some challenges in analyzing the Mix-Code languages as stop words in Natural Language Processing should be given manually depending on our requirements. Some of the other Mix-Code languages can be noted in Table 1.

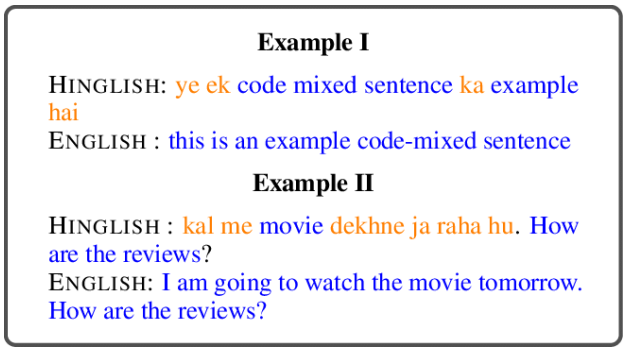

Figure. 1. Hinglish Mix-Code Language. *Source:* (Srivastava and Singh 2021)

Mix-Code	Languages
Benglish	Bengali and English
Chinglish	Chinese and English
Denglisch	Deutsch (German) and English
Dunglish	Dutch and English
Greeklish	Greek and English
Poglish	Polish and English
Porglish	Portuguese and English
Spanglish	Spanish and English
Svorsk	Swedish and Norwegian
Tanglish	Tamil and English

Table. 1. Mix-Code Language Types (Uma Gunturi 2020)

The flow of this project includes cleaning data like removing special characters, smiley symbols, etc. Different types of vectorizations are planned on the data namely TF-IDF, Term Frequency (TF), Count Vectorizer, BERT transformers, GPT and XLM. Supervised learning is applied to all the transformed data vector forms with different classification models like Logistic Regression, K-Nearest Neighbors, Naïve Bayes, Decision Trees, Random Forests, Support Vector Machine, etc. This Report is divided into 7 sections namely Introduction, Literature Review, Methodology, Evaluation, Ethical Considerations, Conclusion and Future Work, and References. The problem statement, the structure of the report, research questions, and research motivation is discussed in the Introduction. The background research works, methods, and influenced works are mentioned in the Literature review. The methodology of how the project has been planned and detailed steps of implementation are discussed in the Methodology section. The description of data and pre-processing steps are mentioned in Data Exploration and Pre-Processing. The Evaluation of different models and final results are described in the Results section. The Ethical methods regarding the project and data are discussed in Ethical Considerations. The project conclusions and hypothesis explanation are discussed in the Conclusion and Future work section. The work references are added in the References section.

Research Questions

1. Which vectorizer techniques can be effectively used for Machine Learning models on Hinglish Mix-Code?
2. Which parametric or non-parametric model is the best performing model on Hinglish data?
3. Is Principal Component Analysis (PCA) and Independent Component Analysis (ICA) on the Machine Learning models help in getting good results for Mix-Code models?

2. Literature Review

This section briefly discusses the literature survey and background studies done for this sentimental analysis.

Data Pre-processing

Data pre-processing includes data cleaning, feature extraction, etc. Data cleaning consists of the removal of stop words, line breaks, emojis, etc. The feature extraction methods used in this analysis are count vectorizer, TF-IDF, term frequency, and transformers like BERT, GPT, and XLM. Kumar et al. used a TF-IDF vectorizer to extract features from Amazon's electronic items dataset and input them into the SVM algorithm (Kumar and Subba 2020). Irawaty et al. made the vectorizations comparison to analyze YouTube comments on Nokia products. They have used TF-IDF, Count vectorizer, and hashing vectorizer for vectorization. They have used K-Nearest Neighbor, SVM to classify. Their evaluation results show that TFIDF with SVM has good accuracy of 97.5% than other combinations (Irawaty et al. 2020). Shah et al. have conducted a Sentimental Analysis of Marglish comments on YouTube cookery channels. Marglish is the mixed code of Marathi and English. They have achieved the best accuracy of 62.68% for the combination of the Count Vectorizer and Multilayer Perceptron. The best models they suggested for Marglish datasets are Multilayer Perceptron and Bernoulli Naïve Bayes (Shah et al. 2020). Aro et al. analyzed the effect of removing stopwords on text data classification of SMS spam datasets. They have modeled using a Decision tree and Multinomial Naïve Bayes. They have found that the removal of stopwords has no effect on the classification effect of text mining but reduced the confidence level of prediction (Aro et al. 2019). AbdulNabi et al. used deep learning for spam mail detection. They have used BERT (Bidirectional Encoder Representations from Transformers) transformer which was pre-trained and fine-tuned to separate spam mails from non-spam. Then they were trained and tested by Machine Learning algorithms using two separate datasets (AbdulNabi and Yaseen 2021). Devika et al. worked on extracting the key phrases from social data using the sentence transformer of the BERT model. As BERT can enhance the performance in Natural Language Processing tasks and extract typical phrases

in tweets, their model of BERT with sentence transformer gave an accuracy of 86% which is higher than their other models (Devika et al. 2021). Qu et al. did the emotion classification of Spanish language data with XLM-RoBERTa for word embedding and the transformer encoder for feature extraction. The extracted features are given to the TextCNN model as inputs (Qu et al. 2021). Kadriu et al. used a Bag of words and word analogies for Albanian text classification. The text has been classified using two approaches, one is converting the text into vector space and the second is using FastText for hierarchical classification. For classification, the bag of words model gave the best evaluation result. For multi-label text, FastText gave better performance. Overall, using the bag of words model gave 94% of accuracy (Kadriu et al. 2019).

Mix-Codes

Mix codes are combinations of different languages in conversations. The data used in this analysis consist of Hinglish mix code which is a combination of Hindi and English. This language is mostly used in India in casual conversations and commenting on social networks. Agarwal et al. worked on Hinglish dialogue generation. They have used mBART multilingual sequence-to-sequence transformers for Hinglish dialog generation which sets new benchmarks for mix codes dialog generation tasks (Agarwal et al. 2021). Kumar et al. used neural networks and transfer learning for cyberbullying detection on mixed code data. They have included typography learned using Machine Learning Processing along with English and Hindi languages. They have combined those features to the unified level which gives the unique distribution advantage without increasing the input space dimensionality (Kumar and Sachdeva 2020). Mundra et al. evaluated text representation methods to detect cyber harmful content on social media. The data considered for analysis is in the Hindi and English mix-code popularly known as Hinglish. In their analysis, it is found that character-based embedding is working well for noisy data. This model also worked better than pre-trained word embedding (Mundra and Mittal 2021). Singh et al. conducted a Sentiment Analysis on social media mix-code content which is in the Hindi and Punjabi languages. The labels of the data include positive, negative, and neutral based on the words in the text. They have used the N-gram approach applied to the sentence (Singh and Goyal 2020). Bansal et al. experimented with Sentiment Analysis on English Punjabi mix-code social media data. They have collected data through Twitter and Facebook APIs. They have used a pipeline Dictionary vectorizer and an N-gram approach (Bansal et al. 2020).

Machine Learning

Machine Learning is the branch of Artificial Intelligence that is helpful in predictions on data. Both Supervised and Unsupervised Learning are useful in sentimental analysis. Unsupervised is used to cluster or separate the data based on patterns while Supervised Learning is used to train the model based on the outputs which help in future prediction. Bhavitha et al. applied Machine Learning algorithms to subjective data to get the intention behind the text whether it is positive, negative, or neutral regarding the newly launched product. They have got 85% of accuracy on supervised learning techniques than unsupervised learning techniques (Bhavitha et al. 2017). Agrawal et al. evaluated supervised and unsupervised learning techniques in sentimental analysis. They have evaluated based on the accuracy, benefits, and disadvantages of every mechanism. They have got good metrics for supervised models when compared to unsupervised models (Agrawal et al. 2021). Bansal et al. experimented with Sentiment Analysis on English Punjabi mix-code social media data. Machine Learning models used are Decision tree, Gaussian Naïve Bayes, and Logistic Regression. The evaluation metrics of Logistic Regression are better with an accuracy of 86.63% and an F1 score of 88% when compared with other models (Bansal et al. 2020). Harfoushi et al. analyzed Twitter data which consists of opinions of individuals, images, and tweets. They have implemented Azure Machine Learning models like SVM and Logistic regression. The results confirmed that Microsoft Azure Algorithms can be used to build effective models when compared to the traditional way of modeling in data analytics (Harfoushi et al. 2018). Thelwall M has checked if there is an effect of

gender bias in Machine Learning for Sentiment Analysis. He has trained and tested the models using three sets of datasets of hotel and restaurant reviews. His study declares that mixed-gender datasets are preferring the opinion of women. Conclusions are that the training of the model on the same gender improves the performance of the model less than adding additional data on both genders' data (Thelwall 2018). Valencia et al. predicted the price movement of cryptocurrencies using Machine Learning and Sentiment Analysis. Models like Neural Networks, Support Vector Machines and Random Forest have been implemented based on the data from Twitter and the market to analyze the price movement of Bitcoin, Ripple, Ethereum, and Litecoin. Results indicate that using Machine Learning price prediction can be possible and Neural networks are better in performance than other models (Valencia et al. 2019). Swaminathan et al. modeled hate speech identification based on the Dravidian mix-code. They have used Machine Learning, deep learning, and ensemble models. For sentiment classification, they have trained and tested the models like Naïve Bayes, Decision tree, Random Forest, Long Short-Term Memory, and AdaBoost. For Hate speech and offense content identification, they have used the models Naïve Bayes, Decision tree, Random Forest, Long Short-Term Memory, SVM, and Gated Recurrent Unit. The F1 scores obtained for Naïve Bayes and Long Short-Term Memory are 61% and 60% respectively. For hate speech identification, subtask A of LSTM gave an F1 score of 50.02% and subtask B of the ensemble approach gave an F1 score of 24.26% (Swaminathan et al. 2020).

Sentiment Analysis

Sentiment Analysis is the process of extracting the intentions from the text computationally along with identifying and categorizing the opinions. It is a sub-field of Natural Language Processing to get the positive or negative or neutral opinion of the text. Fang et al. conducted sentimental analysis on online product review data from Amazon.com. They have analyzed sentence-level categorization and review-level categorization (Fang and Zhan 2015). Serrano-Guerrero et al. worked on a comparative analysis of some free web services. They analyzed using the reviews based on three different collections and analyzed each tool (Serrano-Guerrero et al. 2015). Williams et al. investigated the effect of idioms in Sentiment Analysis. They evaluated models based on precision, recall, and F1-score. The statistical significance of improvement was confirmed using McNemar's test (Williams et al. 2015). Nguyen et al. built a model for analyzing stock movement using sentiments from social media. They have achieved better accuracy while analyzing the 18 stocks using one-year transactions than the historical price method and human sentiment method (Nguyen et al. 2015). Alsaffar et al. performed Sentiment Analysis on the Malay language using K-Nearest Neighbor. They have used Lexicon based approach which derives the intention from text based on the words' semantic orientation. Their hybrid method outperforms the of-the-art unigram baseline method (Alsaffar and Omar 2015).

Transfer Learning

When compared to the training of the model from the scratch by Holderrieth et al., there is performance gain in the model when executed using Transfer Learning by feature usage and similarity in task and distribution in the population (Holderrieth et al. 2021). Due to an imbalance in the data collected from European Ancestry about clinical omics, transfer learning was developed with multi-ethnic data by Gao and Cui. The accuracy of Cox neural network models was improved for ethnic groups that have data imbalance (Gao and Cui 2021). Davchev et al. combined the old prices of Bitcoin to predict the next day's price using Time Series Forecasting and Auto-Regressive methods along with the sentiment extracted from the financial blogs. NLP models with Transfer Learning methodologies have benefitted with good performance in sentiment extraction when compared to the standard extraction models (Davchev et al. 2021). Pan J has worked on a dimensions reduction framework for Transfer Learning which reduces the distance between domains by preserving the properties of the data. He has applied this framework to applications namely cross-domain WiFi

localization and cross-domain text classification. In his method, he proposes a spectral feature alignment algorithm for cross-domain learning where knowledge of the domain is available. The result of the experiment shows that this model is more advantageous than the state-of-the-art algorithms in real-world datasets (Pan 2010). Shahin and Almotairi have worked on the Arabic sign language recognition system which depends on Transfer Learning. A good recognition system of Arabic sign language is proposed which has an accuracy of 99.52% which was later applied to other sign language recognition systems (Shahin and Almotairi 2019). Singh and Lefever have worked on the sentimental analysis of Hinglish tweets using a supervised classifier and Transfer Learning. The cross-lingual embeddings have improved the metrics with F1-Score 0.635 when compared to mono-lingual metrics with F1-Score 0.616. The Transfer Learning results yielded the F1-Score of 0.556 which is almost equal to the supervised learning metrics (Singh and Lefever 2020). Arora et al. have worked on cross-lingual Transfer Learning models for the dataset that contains English, German, French, Spanish, and Spanglish. They have observed that XLM-R models have a good performance on the above-mentioned cross-lingual. They found that it is possible to make model performance better by using only monolingual data (Arora et al. 2020). Zhao et al. have proposed the Transfer Learning model to detect new and unseen network attacks by using known attack information. They have worked on HeTL which relies on hyperparameters and CeHTL which is known for the clustering approach. Their result has proved that HeTL and CeHTL have good performance improvement when compared to their baseline models like KNN, Decision Trees, Random Forests, etc (Zhao et al. 2019).

Hyperparameter Tuning

Hyperparameter Tuning is used to increase the performance of the models by selecting the best parameters based on the algorithm type. Elgeldawi et al. have worked on the Arabic sentiment classification problem of positive, negative, and neutral sentiments. They have used different hyperparameters for each classifier namely Logistic Regression, Random Forest, Decision Tree, Support Vector Machines, Ridge Classifier and Naïve Bayes. After calculating the score of each model before and after hyperparameter tuning, they got good accuracy of 95.6208 by SVM classifier with Bayesian Optimization (Elgeldawi et al. 2021). Hoque and Aljamaan have worked on stock price forecasting evaluated by Mean Absolute Percentage Error and Root Mean Square Error. They have evaluated the models like Decision trees, KNN, SVM, Kernel Ridge Regression, etc before and after hyperparameter tuning. The SVM has the best performance of any other forecasting model and KNN and Decision Trees have no improvement in performance metrics after hyperparameter tuning (Hoque and Aljamaan 2021). Zhang et al. have worked on Alzheimer's Disease prediction in the early stage. The data includes age, sex, education, etc along with MRI information. They have applied the SVM algorithm with hyperparameters with 100 times repeat and 5-fold cross-validation for this use case. They have achieved an increasing 96% efficiency with the selected hyperparameters of SVM (Zhang et al. 2021). Wazirali has worked on a Cyber-attack detection system that depends on hyperparameters tuning and cross-validation techniques. KNN model as main model and 5-fold cross-validation has been performed with different nearest neighbors as a hyperparameter. The comparison of the Intrusion Detection System with KNN algorithms, the result demonstrates that the proposed approach has good performance (Wazirali 2020). Ottoni and Novo have conducted Deep Learning to recognize vegetation images in buildings. They have mainly used Convolutional Neural Networks for their use case. They adjusted the hyperparameters with rigorous search which helped them in getting an accuracy score of 90% at the test stage. Their hyperparameter tuned model has achieved 97.8% of correct classification for positive class (Ottoni and Novo 2021). Stuke et al. have conducted the computational chemistry experiment for different hyperparameter tuning types like Grid Search CV, and Random Search CV on Bayesian Optimization for Ridge Regression model. While increasing the number of hyperparameters, Bayesian Optimization and Random Search CV are more efficient in computational time when compared to the Grid Search CV. Also, the metrics like accuracy have increased for Bayesian Optimization and Random Search CV while delivering (Stuke et al. 2021).

3. Methodology

In this section, the methods and flow of sentimental analysis that is conducted are discussed. The flow of the project is divided into different sections as below

1. Data Collection: The data is collected from the UCI website (UCI Machine Learning Repository: Youtube cookery channels viewers comments in Hinglish Data Set n.d.). The data contains the comments received by the two YouTube cookery channels namely, Nisha Madhulika's Cooking channel and Kabita's Kitchen. The data consists of labels divided into 7 categories as shown in Table 2.
2. Data Preprocessing: The raw data consists of many line breaks and smiley symbols. They are removed in the preprocessing stage.
3. Data Visualization: The Visualization Analysis is carried out to analyze labels, stop words, hashtags, word counts, character counts, numerical values present, etc.
4. Vectorization: The processed data is converted to vector form datasets using different vectorization techniques like Term Frequency-Inverse Document Frequency (TF-IDF), Term Frequency (TF), Count Vectorizer, BERT Transformers, etc.
5. Feature Scaling: Different Scaling techniques are applied to check the effect of scaling on the Machine Learning evaluation results.
6. Machine Learning: The Machine Learning models are trained and tested with the vectorized datasets. Different cross-validation techniques are used for each model. The training data is 70% and the testing data is 30%. The dimension reduction technique like Principal Component Analysis and Information separation technique like Independent Component Analysis is performed.
7. Evaluation: As the Sentimental analysis is based on the classification type of supervised learning, the evaluation is done based on Precision, Recall, F1 Score, Confusion matrix, Classification report, Accuracy, Area Under Curve, etc.
8. Hyperparameter Tuning: It is useful in finding a set of optimal values for the parameters of each model. Best parameters are found using Random Search CV and Grid Search CV.
9. Hypothesis Testing: It is useful in finding the performance of the model statistically. Paired T-Test is conducted between the best models to finalize one among them.
10. Results: The best results for the research question will be fixed based on the evaluation results of the different Machine Learning models applied to different vectorized datasets.

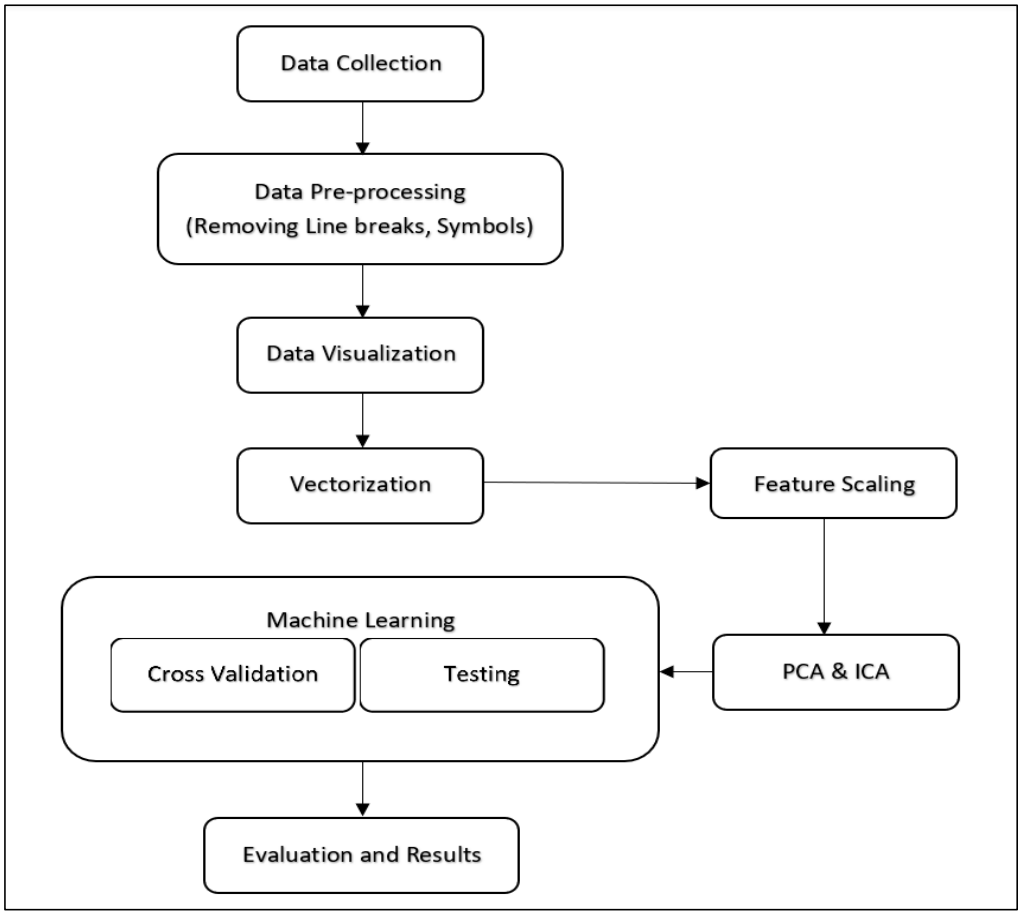

Figure. 2. Flow of Methodology

3.1 Data Collection

The two datasets are of two YouTube Cookery channels taken from the UCI website. The channels are India's popular cooking channels namely NishaMadhulika and Kabita's Kitchen. Each dataset consists of 4900 rows. Each row has a comment given by the user and the type of user intention through the comment. As the comments on YouTube resemble conversational type language and some comments are given by users who don't have good knowledge of English typing, some spellings might be wrong. The comments were clustered and labeled using the unsupervised learning method Density-Based Spatial Clustering of Applications with Noise (DBSCAN) after collecting the YouTube comments through its API in March 2019 (Kaur et al. 2019).

The dataset labels were classified into 7 categories based on the viewers' intentions. Those 7 categories include Gratitude, About Recipe, About Video, Praising, Hybrid, Undefined, Suggestion, or Query. The number of rows of each dataset was divided equally according to those 7 labels as shown in Table 3. The description of each label can be seen below.

Label 1 – Gratitude

This Label indicates that the comment is the gratitude shown by the viewer to the YouTuber.

Examples:

1. Thank you so much for putting this detailed video
2. thank u mam
3. thank you didi

Label 2 – About Recipe

This Label indicates that the comment is the review given by the viewer about the recipe and how good it is and tastes.

Examples:

1. This is a perfect biryani recipe
2. Nice recipe, that was so simple yet delicious
3. 2 good Mam very nice recipe

Label 3 – About Video

This Label indicates that the comment is the review given by the viewer about the video how good it is and playtime.

Examples:

1. AMAZING! Maine ye video dekhkar dum biryani banana sikha hai
2. very nice video mam, Great video!
3. nice video

Label 4 – Praising

This Label indicates that the comment is the review given by the viewer praising the chef and admiring him.

Examples:

1. the way u cook, it's really looking so beautiful
2. Very nice cooking style
3. Super your recipes are amazing

Label 5 – Hybrid

This Label indicates that the comment includes two or more qualities of labels. For example, the viewer expresses his views about the recipe and video in the same comment.

Examples:

1. Thakuuu soo mch mam u r such a talented
2. Nice Aunty ji.....kaun se oil ka use karna hoga??
3. hello nisha,ive tried ur alo paratha n it was just awesome,i just love u n ofcourse ur recipes.

Label 6 – Undefined

This Label indicates that the comment doesn't come under any of the other labels like praising or showing gratitude or querying about recipes or videos.

Examples:

1. I am hungry
2. Who try this please one like
3. Happy new year aanti

Label 7 - Suggestion or Query

This Label indicates that the comment is the question or suggestion by the viewer about the recipe.

Examples:

1. Atta flour means wheat flour?
2. Can we grate the potatoes mam?
3. Kya stafing me Magi masala dal sakte he

Labels	Nisha Madhulika Dataset	Kabita's Kitchen Dataset
Label-1	700	700
Label-2	700	700
Label-3	700	700
Label-4	700	700
Label-5	700	700
Label-6	700	700
Label-7	700	700
Total Comments	4900	4900

Table. 2. Distribution of Labels in the Datasets

3.2 Data Preprocessing

YouTube comments given by users consist of many spelling mistakes and special characters. This is because the comments resemble the common conversation type language. To make the data efficient for modeling, preprocessing is done on both datasets. Pre-processing includes the removal of special characters, smiley symbols, numbers, line breaks, converting text to lowercase, stop words, etc. Tokenization is done before vectorization.

Special characters include punctuation marks. Smiley symbols are generally used on social media to replicate the expressions. So, they are removed. Line breaks occur if the user tries to write 2 different reviews in the same comment. All the text is converted to lowercase to attain equality in the strings while performing the vectorization. Stop words are the most used words in sentences. For example, stop words are like 'at', 'is', 'was', 'if', etc. But these stop words should be configured according to the use case. As the comments used for analysis are of Hinglish mix-code language, we should manually add stop words according to our requirements. Tokenization means the splitting of sentences into keywords, phrases, etc called Tokens by removing spaces, punctuations, etc.

3.3 Data Visualization

The main purpose of this data visualization is to analyze the data and understand it more clearly. It provides a well-organized visual representation of data to easily analyze and interpret the understanding. The distribution of labels, stop words, hashtags, word counts, character counts, numerical values present, etc in the data are analyzed using visualizations. This is achieved by plotting the graphs like Boxplots, Count plots, etc. using matplotlib or seaborn libraries.

3.3.1 Kabita's Kitchen Dataset

Classes Distribution

The Number of Comments distributed for each Class or Label.

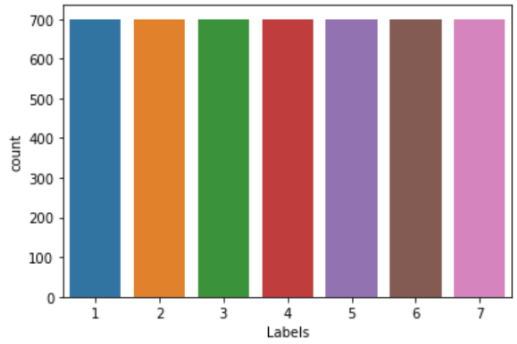

Figure. 3. Classes Distribution in Kabita's Data

Stopwords Distribution

The Number of Stopwords present in each comment.

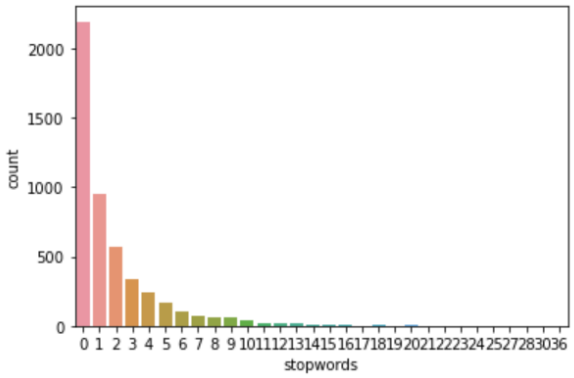

Figure. 4. Stopwords Distribution in Kabita's Data

Upper Words Distribution

The Number of Upper character words present in each comment

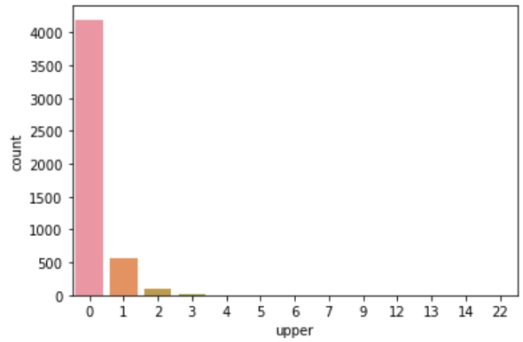

Figure. 5. Upper Words Distribution in Kabita's Data

Hashtags Distribution

The Number of Hashtags present in each comment

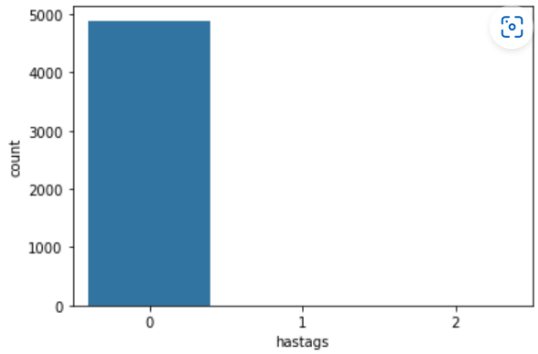

Figure. 6. Hashtags Distribution in Kabita's Data

Word Count in Comments

The Number of Words in each comment

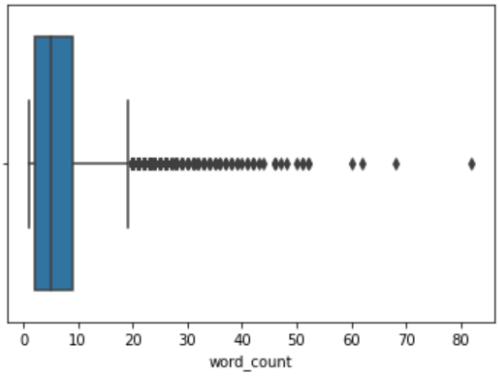

Figure. 7. Word Count of Comments in Kabita's Data

Character Count in Comments

The Number of Characters in each comment.

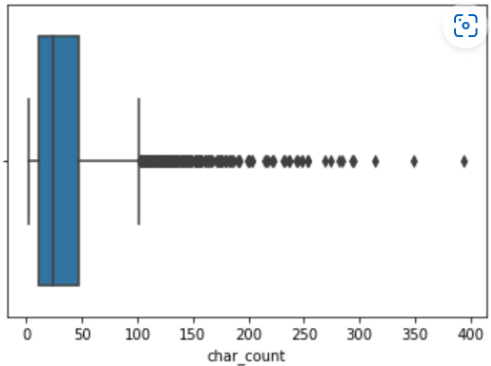

Figure. 8. Character Count of Comments in Kabita's Data

Average Word Count in Comments

The Average number of words in the comments (Total number of characters without white spaces divided by the total number of words).

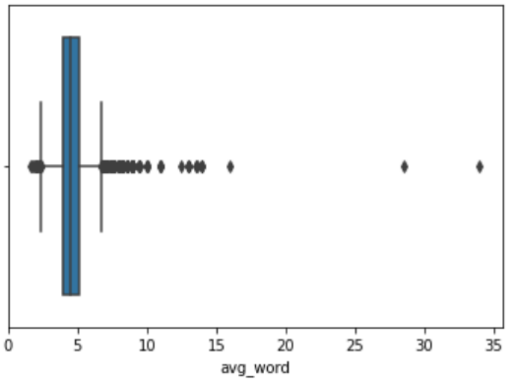

Figure. 9. Average Word Count of Comments in Kabita's Data

Stopwords Distribution for each Label

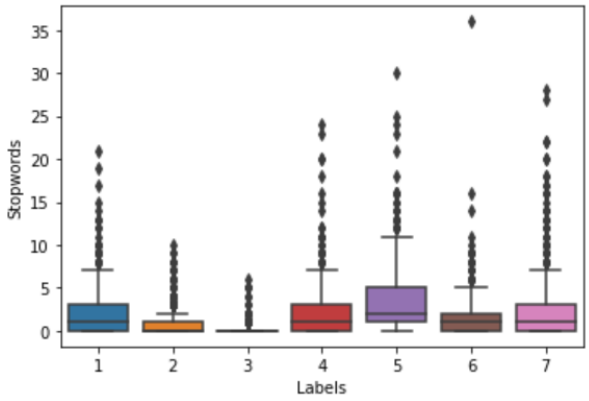

Figure. 10. Stopwords Distribution for each Label in Kabita's Data

Word Count of Comment for each Label

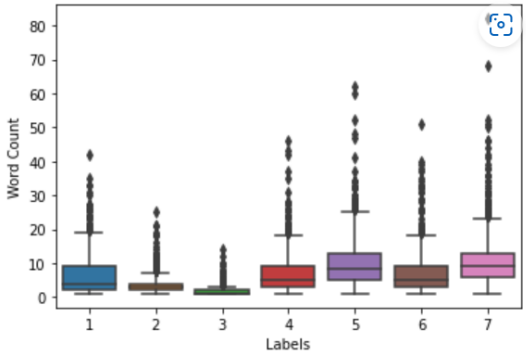

Figure. 11. Stopwords Distribution for each Label in Kabita's Data

Relation between Word Count and Character Count

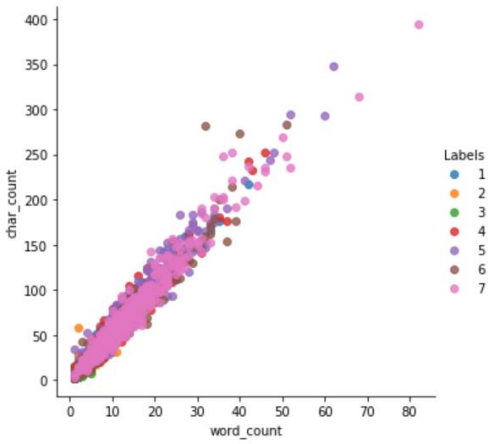

Figure. 12. Relation between Word Count and Character Count in Kabita's Data

Relation between Character Count and Average Words

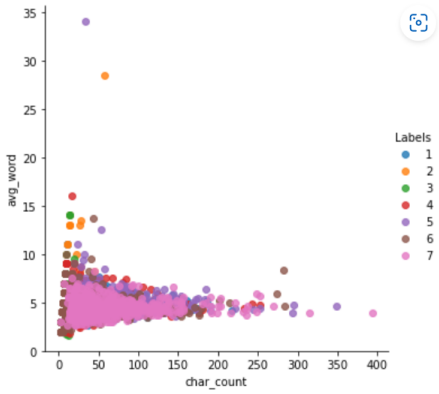

Figure. 13. Relation between Character Count and Average Words in Kabita's Data

3.3.2 Nisha's Dataset

Classes Distribution

The Number of Comments distributed for each Class or Label.

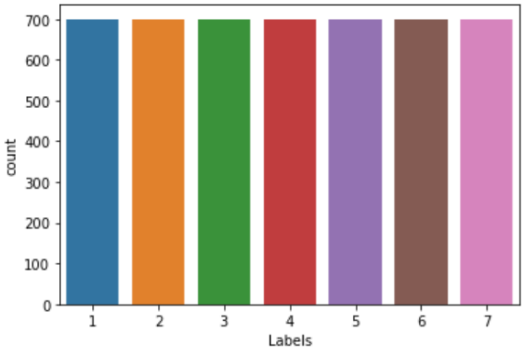

Figure. 14. Classes Distribution in Nisha's Data

Stopwords Distribution

The Number of Stopwords present in each comment.

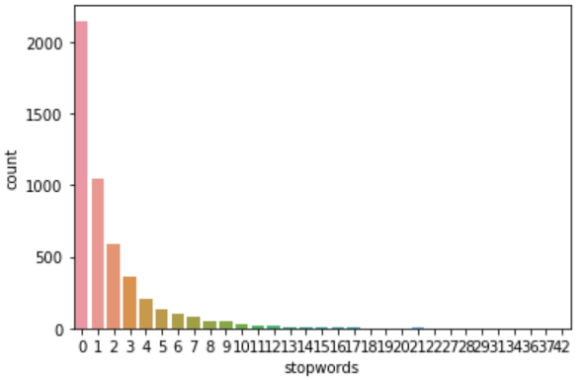

Figure. 15. Stopwords Distribution in Nisha's Data

Upper Words Distribution

The Number of Upper character words present in each comment

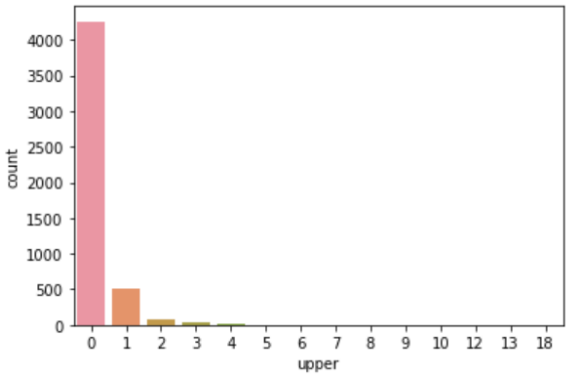

Figure. 16. Upper Words Distribution in Nisha's Data

Hashtags Distribution

The Number of Hashtags present in each comment

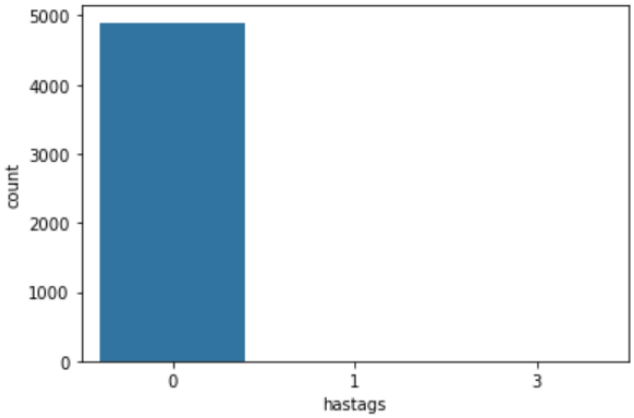

Figure. 17. Hashtags Distribution in Nisha's Data

Word Count in Comments

The Number of Words in each comment

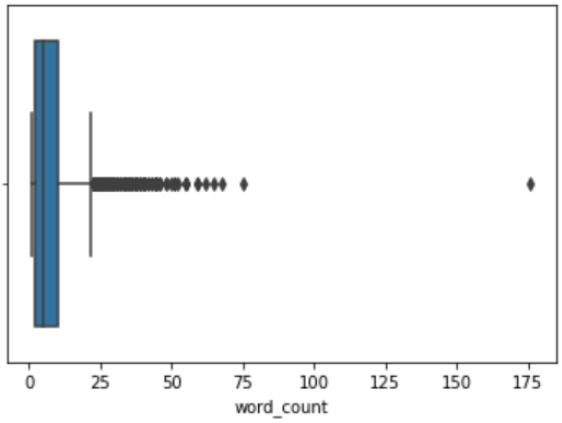

Figure. 18. Word Count of Comments in Nisha's Data

Character Count in Comments

The Number of Characters in each comment.

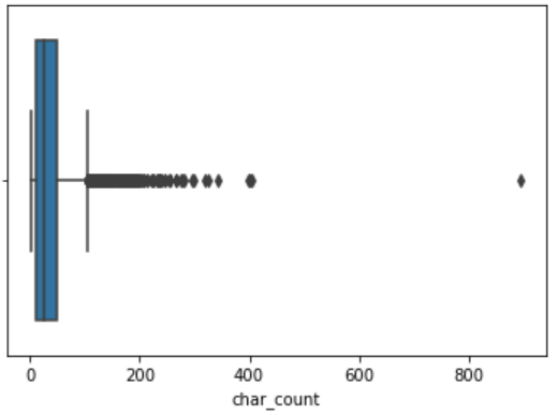

Figure. 19. Character Count of Comments in Nisha's Data

Average Word Count in Comments

The Average number of words in the comments (Total number of characters without white spaces divided by the total number of words).

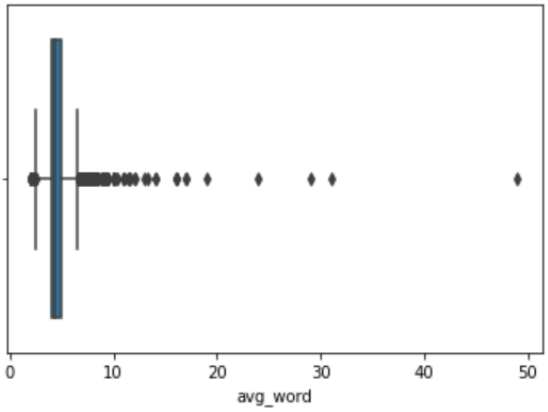

Figure. 20. Average Word Count of Comments in Nisha's Data

Stopwords Distribution for each Label

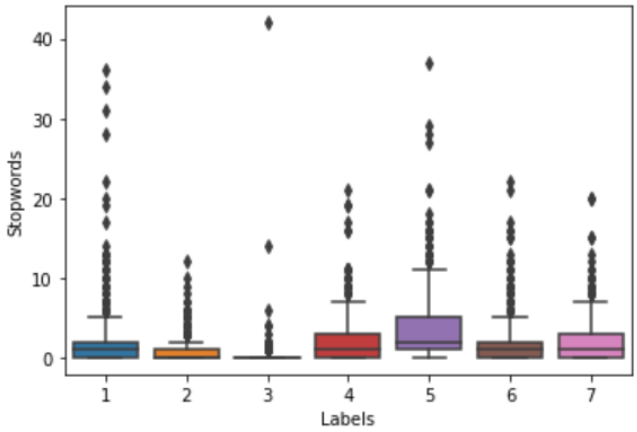

Figure. 21. Stopwords Distribution for each Label in Nisha's Data

Word Count of Comment for each Label

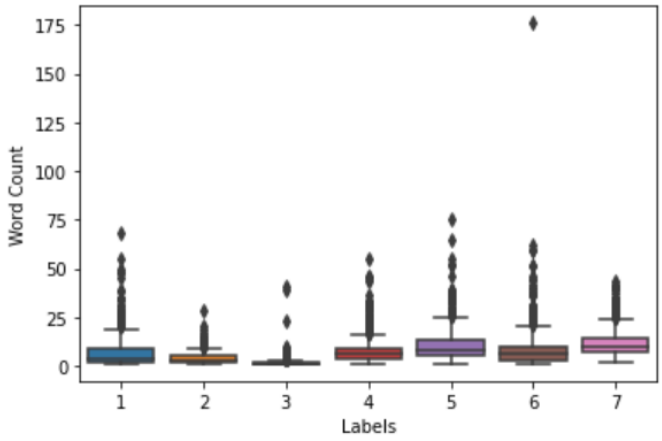

Figure. 22. Stopwords Distribution for each Label in Nisha's Data

Relation between Word Count and Character Count

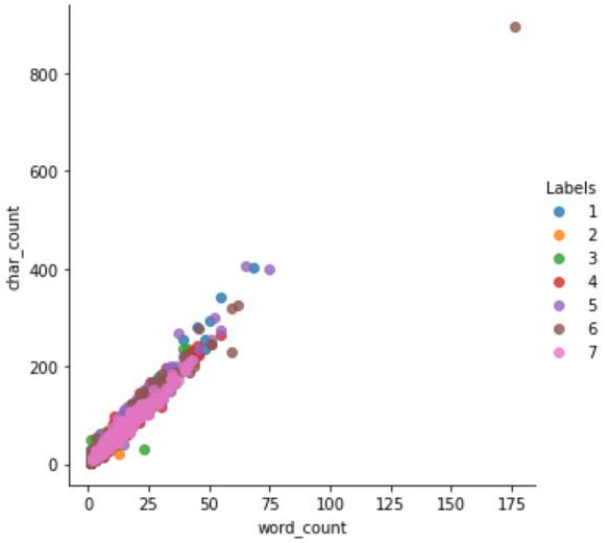

Figure. 23. Relation between Word Count and Character Count in Nisha's Data

Relation between Character Count and Average Words

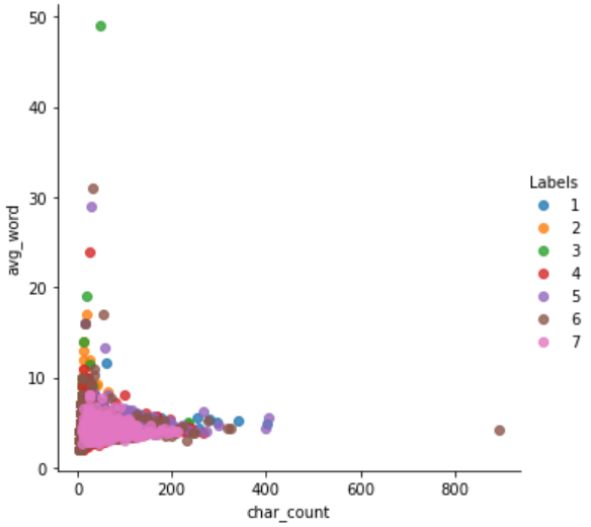

Figure. 24. Relation between Character Count and Average Words in Nisha's Data

3.4 Vectorization

In Machine Learning, while working with categorical data, we need to convert them to numerical as the statistical calculation can be done only on numerical values. For this requirement, there are numerous methods to convert categorical data into numerical data. Some of the methods are dummies creation, Values assignment, Vectorization, etc. In vectorization, the text is tokenized and converted into vectors called Feature Extractions. One of the best methods for this feature extraction is Bag of Words. In the Bag of Words model, the grammar and order of words won't be considered instead it will keep the count of word repetition. The Example of the Bag of words application can be seen in Table 4. As Bag of words feature extraction is best for classification models, this method of feature extraction is applied before modeling.

Normal text	This Project is based on Natural Language Processing. Natural Language Processing is formerly called NLP.

Bag of Words model	BoW1 = {
	"This":1, "Project":1, "is":2, "based":1, "on":1, "Natural":2,
	"Language":2, "Processing":2, "formerly":1, "called":1,
	"NLP":1
	}

Table. 3. Bag of Words Example

The Bag of Word models used for the analysis is Term Frequency – Inverse Document Frequency (TF-IDF) Vectorizer, Term Frequency (TF) Vectorizer, and Count Vectorizer.

3.4.1 Term Frequency – Inverse Document Frequency Vectorizer

The approach in this method is that the words that are more common in one text and less common in other texts should be given high weights. For this method also, the first step will be tokenization. TF-IDF value of each word in the text will be calculated.

TF value can be calculated by,

$$TF = \frac{Frequency\ of\ the\ word\ in\ the\ sentence}{Total\ number\ of\ words\ in\ the\ sentence}$$

IDF value can be calculated by,

$$IDF = \log\left(\frac{Total\ number\ of\ sentences\ (documents)}{Number\ of\ sentences\ (documents)\ containing\ the\ word}\right)$$

$$TF - IDF = TF * IDF$$

TF value of word changes from document to document but IDF value of word remains constant as it depends on the total number of documents

3.4.2 Term Frequency Vectorizer

It is the value of TF from the TF-IDF vector without IDF value. The Term frequency of words will be calculated by dividing the frequency of words in the sentence by the total number of words. The value of the word which is repeated more will be given preference.

3.4.3 Count Vectorizer

It calculates the value by one-hot encoding which means the value depends on the number of times the word repeats in the text. For every occurrence of the word in the text, the value will be incremented by 1. If the word is not present in the feature, it will be added. The example of count vectorization is explained in Table 5.

Normal text		Hi, how are you? Are you fine?				
Count Vectorization	**Indexing**	are	fine	hi	how	you
		0	1	2	3	4
	Vector values	2	1	1	1	2

Table. 4. Count vectorization

In the field of Data Science, there is a great use of transfer learning as a pre-trained neural network is taken based on the known task and used by fine-tuning it for the required specific model. Along with the vectorizers, word embeddings of transformers like BERT, GPT, and XLM are used to convert the comments to vector formats. Word embeddings mean converting words to vectors in lower-dimensional space. By this, we can use mathematical operations on the numerical form of words in Machine Learning. Transformers are the deep learning encoder-decoder model which uses the self-attention mechanism weighting the parts of input data. They are increasing their choice for Natural Language Processing replacing other deep learning models like Recurrent Neural networks (RNN), Long Short-Term Memory (LSTM), etc. Both Sentence Transformers and Fine-Tuned Transformers are used to convert Comments to Vector forms. For Sentence Transformer models, Pre-Trained Transformers like BERT Base (sentence-transformers/bert-base-nli-mean-tokens · Hugging Face n.d.), Ganesh BERT Hinglish (ganeshkharad/gk-hinglish-sentiment · Hugging Face n.d.), Narasimha Distil BERT Hinglish (Narasimha/hinglish-distilbert · Hugging Face n.d.), Verloop BERT Hinglish (verloop/Hinglish-Bert · Hugging Face n.d.), GPT (Muennighoff/SGPT-125M-mean-nli · Hugging Face n.d.), and XLM (sentence-transformers/stsb-xlm-r-multilingual · Hugging Face n.d.) is used to convert text to vectors. Vectors derived using Fine-Tuned Transformers are using BERT Base (bert-base-uncased · Hugging Face n.d.), BERT Hinglish (Verloop BERT), GPT Base (gpt2 · Hugging Face n.d.), GPT Hinglish (impyadav/GPT2-FineTuned-Hinglish-Song-Generation · Hugging Face n.d.), and XLM Base (xlm-mlm-en-2048 · Hugging Face n.d.).

i. BERT Model

It is a Bidirectional Encoder Representation from Transformers. It is a pretrained transformer model well-suitable for Natural Language Processing. Here it is used to extract high-quality features from text data and use them for classification analysis. It has an advantage over the

Word2Vec models because it captures the differences like polysemy and context. For example, the word “bank” in “robbing the bank” and “fishing by the bank” has two different word embeddings in the BERT model when compared to Word2Vec models.

BERT works on the attention mechanism that learns the context between the words in a text. The Transformer has two levels namely Encoder that reads the input and the Decoder that produce the prediction mechanism for the task. Due to the importance of the generation of the language models, BERT’s goal is of encoding mechanism. As it is bidirectional, the encoder of the BERT model reads the input both Left-to-Right and Right-to-Left. Therefore, being a non-directional transformer, BERT learns the context of the input in all directions. The input is a sequence of tokens that are converted to vectors in the neural network. For learning the context in the sentence along with the prediction, BERT uses two types of processes namely Masked LM and Next Sentence Prediction.

In Masked LM, 15% of words in the input are replaced with masked tokens. Then the model tries to predict the masked words based on the non-masked words’ context in the order of sentence. The level in the process includes the addition of the classification layer on the encoder output, transforming the output vectors into vocabulary features by multiplying the vectors with an embedding matrix, and probability calculation of each word using softmax.

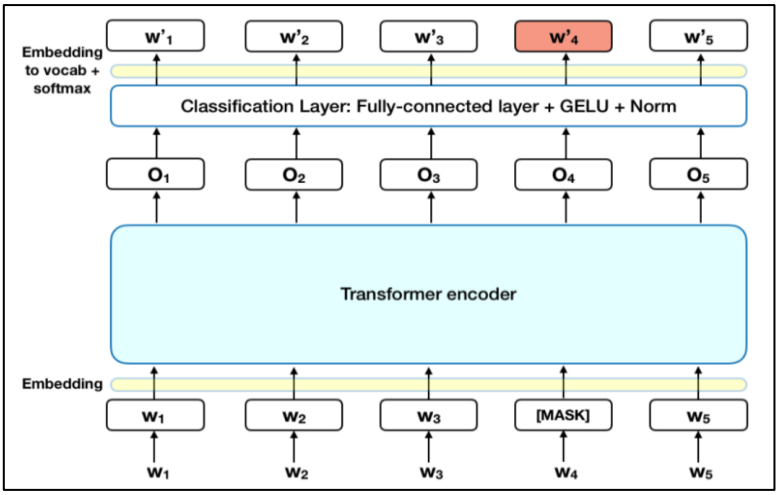

Figure. 25. BERT Masked LM Architecture (BERT Explained: State of the art language model for NLP | by Rani Horev | Towards Data Science n.d.)

In Next Sentence Prediction, the next sentence is predicted based on the first sentence. The first half of the sentence is taken as the input and some part in the next sentence is taken randomly. A [CLS] token is added at the beginning of the first sentence and a [SEP] token is added at end of each sentence. Sentence embedding to indicate A or B is attached to each token along with the positional token to know the position of the word in a sentence. To predict the second sentence based on the first one, the input goes through the BERT model. The output token that has [CLS] token is transformed into a 2x1 vector in the classification layer. The probability of the next sentence will be calculated using softmax.

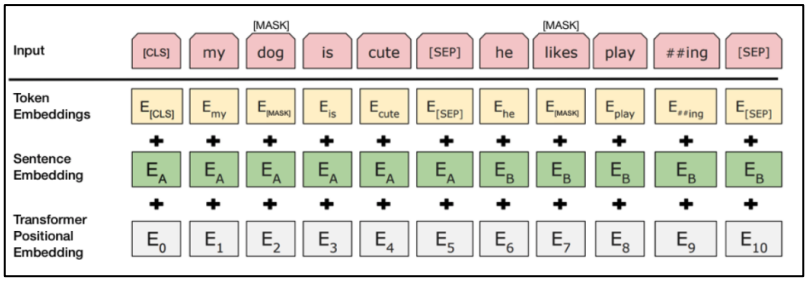

Figure. 26. BERT Next Sentence Prediction Architecture (BERT Explained: State of the art language model for NLP | by Rani Horev | Towards Data Science n.d.)

BERT can be used for various types of language task requirements like Next sentence classification in sentiment analysis, Question and Answer tasks, Named Entity Recognition tasks, etc.

ii. GPT Model

It is a Generative Pre-trained Transformer model by OpenAI. It performs Natural Language Processing tasks like answering questions, and the relation between text fragments, etc. Using generative pre-training, the model improves the understanding of language. GPT used in this project is used for word embedding in a 768-dimensional state. It is built based on Transformer Decoder blocks. It consists of 12 layers of Transformers and 12 independent attention mechanisms for each transformer. Each combination of transformer and attention mechanism is one linguistic property that the model captures.

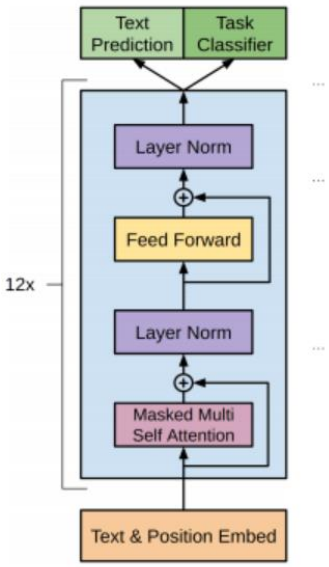

Figure. 27. Improving Language Understanding by GPT (Examining the Transformer Architecture | by James Montantes | Towards Data Science n.d.)

We can say GPT is the sophisticated and advanced version of the mobile keyboard app for the next word prediction feature. It was trained on 40GB of web data and based on the model dimensions and Decoder blocks, the GPT is classified into GPT-2 Small (768 Dimensions and 12 Decoder Blocks), GPT-2 Medium (1024 Dimensions and 24 Decoder Blocks), GPT-2 Large (1280 Dimensions and 36 Decoder Blocks), GPT-2 Extra Large (1600 Dimensions and 48 Decoder Blocks). The major difference in the self-attention layer in GPT is that it masks the future tokens by not changing the words like BERT. It gives the output one token at a time, and it

will be an input to the next token. The tokens are produced in the form of a sequence called "Auto-Regression".

Initially, the model will have only one token as input, and it is processed in all layers producing a vector. Based on the highest probability of scores among the vocabulary present in the model, the next word will be predicted. This word will be the input to the next token. While producing the vector in the top decoder block, the vector will be multiplied by the embedding matrix which refers to the word embedding in the model. The result is the score for each word in the vocabulary of the model. Tokens are usually created by the model using Byte Pair Encoding (The Illustrated GPT-2 (Visualizing Transformer Language Models) – Jay Alammar – Visualizing Machine Learning one concept at a time. n.d.).

iii. XLM Model

It is Cross-Lingual Language Model after enhancing the BERT Model. It is a pre-trained transformer for the objectives like casual language modeling, masked language modeling, and translation language modeling. XLM is used for word embedding for this project. It uses the mechanism of Dual Language Training with BERT to learn the relation between the text inputs in different languages. XLM works on Multilingual Classification Tasks and Machine Translation Models.

XLM used Byte Pair Encoding (BPE) which splits the text into common words across different languages and helps in sharing vocabulary between languages. Each training set consists of the same words in different languages. As XLM has the BERT architecture, the model can get the context between the words from one language and predicts the tokens in another language. In addition, the model also contains Language ID along with Sentence ID and Positional ID.

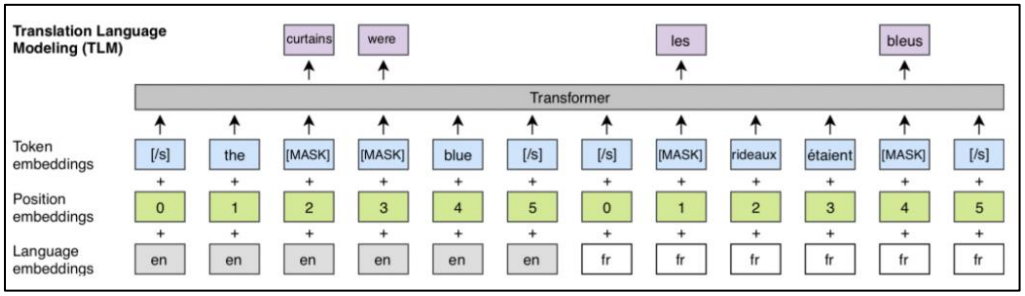

Figure. 28. XLM Translation Language Modeling Architecture (XLM — Enhancing BERT for Cross-lingual Language Model | by Rani Horev | Towards Data Science n.d.)

The BPE processing helps leverage data from other languages to low-level languages. The sentences in one language are translated to another language and back to the main language using Back translation.

3.5 *Feature Scaling*

In raw data, the values will range widely which will make Machine Learning algorithms work abnormally. So, scaling of data is needed to normalize the features of the data. Scaling of the data should be normally done in pre-processing steps of modeling. There are different types of scaling techniques like Min-Max Scaling, Standard Scaling, Normalize Scaling, Binary Scaling, etc.

3.5.1 Min-Max Scaling: It shrinks the data to the given range of values without losing the shape of the original distribution. By default, it will scale the data in the range of 0 to 1. The scaling of data between the required range of values (a,b) is generally done by the below formula.

$$x^1(Scaled) = \frac{(b-a)(x-x_{min})}{x_{max}-x_{min}} + a$$

3.5.2 Standard Scaling: The Standard distribution is mainly achieved by standard scaling. The scaled value is the result of the difference between the actual value and the mean value of the feature divided by the standard deviation of the feature.

$$x^1(Scaled) = \frac{x-\bar{x}}{\mu}$$

3.5.3 Normalize Scaling: Normalizer is mainly used to control the size of vector to avoid numerical instabilities due to outliers. It shrinks the data between 0 to 1. It is mostly useful for regression than classification.

$$x^1(Normalized) = \frac{x-x_{min}}{x_{max}-x_{min}}$$

3.5.4 Binary Scaling: It is the technique of scaling where the threshold should be provided. The values less than or equal to the threshold will be changed to 0 and values greater than the threshold will be changed to 1. The default threshold for Binarizer is 0.

3.6 Machine Learning

Machine Learning is a branch of Artificial Intelligence where the predictions are made for future data by the algorithms based on the patterns of the data we feed while training. Machine Learning algorithms are divided into 4 types based on the data of prediction.

a. Supervised learning: In this type, the models are trained with both Inputs and desired outputs of the data. The training data will be in the form of a matrix with the desired output in vector form called labels. One label might be the output of multiple input types. Supervised learning is further divided into Regression and Classification. In regression, the output labels are numerical data types and in classification, output labels are Categorical data types. The algorithm keeps on improving the accuracy and predictions over time based on the data.
b. Unsupervised Learning: These models are used if the data consists of no labels to predict the output. The main purpose of this learning is to group or cluster the data based on the patterns and similarities recognized by the algorithm. Unsupervised learning is further divided into 2 types namely Clustering and Association rules. K-Means, Hierarchical, etc are important clustering types. Association rules help to find the relations and co-occurrences between features in data.
c. Semi-Supervised Learning: It involves both unsupervised and Supervised learning models. The data which consists of no labels are clustered and provided labels using unsupervised learning. Now the data is mapped with labels and trained using supervised learning models to predict unknown future data. Based on the accuracy, the supervised learning model is again trained along with the test data.

d. Reinforcement Learning: In this type, the model will depend on the sequence of decisions while training. The goal is to reduce the error and increase the success accuracy based on the error scenarios. The model always tries to learn from the random trails themselves.

The data for the sentimental analysis has already been labeled. So, Supervised learning models are applied to predict the user's intention through his comment. The labels of the data are considered as categorical as they are assigned to the sentiment types. The classification algorithms are modeled according to this analysis's response variable data type. Based on the parameters, the supervised classification algorithms are divided into 2 types i.e., parametric, and non-parametric. Parametric models require fixed parameters and are not flexible. In non-parametric models, the parameters are not fixed. Due to this, the features increase with training data. The various parametric and non-parametric models are mentioned in Table 6. In this use case, both parametric and non-parametric algorithms are used.

Parametric models	Logistic Regression Bernoulli Naïve Bayes Gaussian Naïve Bayes Multinomial Naïve Bayes
Non-parametric models	Decision tree Random forest K-Nearest Neighbors (KNN) Support Vector Machines (SVM)

Table. 5. Parametric and Non-parametric models

Logistic Regression

Logistic Regression (LR) is one of the supervised learning algorithms which is suitable mostly for binary classification. It works on the principle of predicting the probability of an outcome or observation or event. It is one of the easiest and most suitable for linearly separated datasets. It uses the Sigmoid function to calculate predictions and probabilities. The prediction of the class depends on the predefined threshold on the sigmoid function graph. The Sigmoid function of the Logistic Regression can be defined as below equation lr1 where 'e' is the base of natural algorithms and x is the numerical value that needs to be transformed.

$$f(x) = \frac{1}{1+e^{-x}} \quad \rightarrow \text{(lr1)}$$

Based on the equation lr1, the equation of Logistic regression can be represented as below equation lr2 where x is input, y is the predicted value, b0 is the intercept term, and b1 is the coefficient of x.

$$y = \frac{e^{(b_0+b_1x)}}{1+e^{(b_0+b_1x)}} \quad \rightarrow \text{(lr2)}$$

The assumption of Logistic Regression includes a large sample size, no extreme outliers in the dataset, observation of data are independent of each other, and no collinearity between independent and target variables. Binary, Multinomial and ordinary are types of Logistic Regression (Logistic Regression: Equation, Assumptions, Types, and Best Practices n.d.).

Naïve Bayes

Naïve Bayes is a classifier-type Machine Learning model that depends on the Bayes theorem. Bayes theorem. Using the Bayes theorem, we can find the probability of A happening when B occurred. The assumption for Bayes's theorem is that all the targets have equal importance for the outcome.

$$p(y|X) = \frac{p(X|y)p(y)}{p(X)} \quad \rightarrow \text{Bayes Equation}$$

Types of Naïve Bayes models include Multinomial Naïve bayes, Bernoulli Naïve Bayes and Gaussian Naïve Bayes. The multinomial Naïve Bayes model is mainly used for the classification of multi-class document type problems. Bernoulli Naïve Bayes is used for Boolean-type predicted class problems. Gaussian Naïve Bayes model is used mostly for continuous target variable that follows gaussian distribution (Naive Bayes Classifier. What is a classifier? | by Rohith Gandhi | Towards Data Science n.d.).

Decision Tree

Decision Trees are one of the non-parametric models of Machine Learning for supervised classification. The value is predicted using the decision rules learned by the model according to the independent variables. The decision of the tree depends on the maximum depth of the model. More the depth, more the complex the model because of complex decisions, and can be noticed in Figure 27. It can do multi-class classifications based on the probability and the index. Visualizations of trees and their simple understanding are the advantage of Decision Trees. The predicting data is logarithmic of several observations used for the training. Validation of the model can be done using statistical tests. Over-fitting due to over complexion of trees is the disadvantage of Decision Trees. Due to its sensitivity to stability, small changes in the data change the decision of the tree. Balancing the dataset is necessary as there will be biases in the model if the class dominates others.

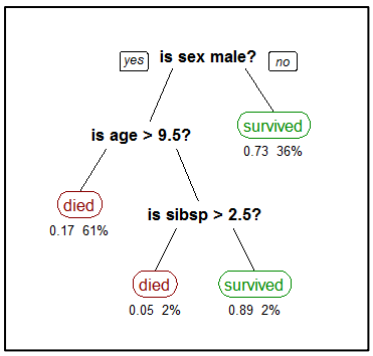

Figure. 29. Decision Tree Classification Example (Decision Trees in Machine Learning | by Prashant Gupta | Towards Data Science n.d.)

Random Forest

Random Forest can be used as a classification or regression model. The complexity of Random Forest depends on the number of trees used in the model. The decision of the predicted output depends on the voting system after all the tree outputs are derived based on different combinations. Feature importance is pretty good in this model. It is a divide and conquers approach which is called an ensemble method. The trees are divided based on the information gain, Gini Index, and gain ratio. The working of the Random Forest model can be observed in Figure 28. High accuracy is the advantageous feature of Random Forest as the decision is based on all the individual decisions. Due to the consideration of average predictions, it avoids over-fitting. It handles Null values by taking the median or weighted average by itself. Slow functionality is one of the disadvantages as it processes multiple decision trees. Interpreting is a difficult task due to its complex structure.

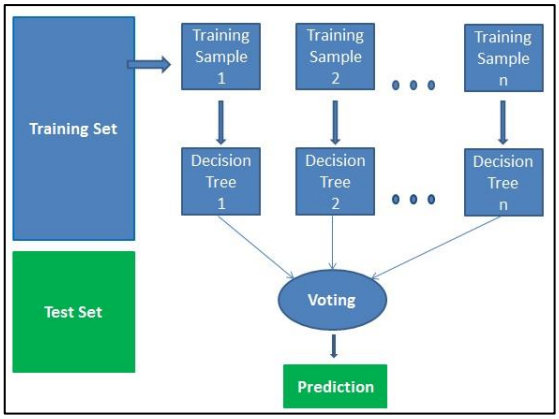

Figure. 30. Functionality of Random Forest (Sklearn Random Forest Classifiers in Python Tutorial | DataCamp n.d.)

K – Nearest Neighbors (KNN)

KNN predicts the class of the data based on the distance between the surrounding train points to the test point. Based on the k value selected which is the number of neighbors that are needed to be taken into consideration, the class of test data will be predicted. The probabilities of test data are calculated by the model to check to what class it belongs. The class holding the highest probability will be selected. For regression models, the mean is taken for assigning the predicted continuous output. The distance calculated by default in the KNN model is Euclidean distance. The value of K is one of the hyperparameters of the KNN model. For each value of K, the model should be fitted and tested to calculate the metrics of the models. The K value which contains good evaluation metrics will be selected to train total data. As the data needs to be fitted every iteration, it is computationally expensive and slow. Thus, it is called Lazy Learning Algorithm. Generally, the value of K is taken using the elbow method which is the plot between the K value and the Accuracy of each model with the K value. The boundary line which the model takes based on the K value can be seen in Figure 29.

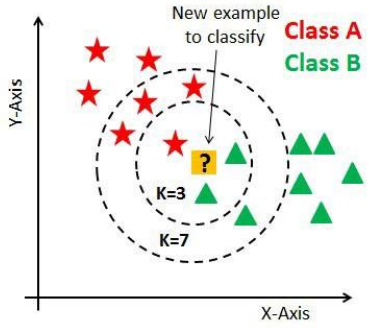

Figure. 31. KNN model with boundaries based on K value (K-Nearest Neighbor. A complete explanation of K-NN | by Antony Christopher | The Startup | Medium n.d.)

Support Vector Machine (SVM)

SVM is one of the most preferred algorithms in Machine Learning due to its best results with less computation power. Its main objective is to find the hyperplane in an N-dimensional space to classify the data. N is the number of features in the data. Even though there can be multiple hyperplanes to divide the data, the plane with the maximum margin between both the classes. Future data can be classified with more accuracy when the plane margin is maximum. The hyperplane concept can be easily understood in Figure 30. Based on the number of features in the dataset, the hyperplane will be taken. For example, if the number of features is 2, the line is the hyperplane. If the number of features

is 3, the 2-dimensional plane will be the hyperplane. It can be seen in Figure 31. As the position and size of the hyperplane are influenced by the points nearer to it, these vectors are useful for maximizing the margin of the hyperplane and helping in building the SVM model.

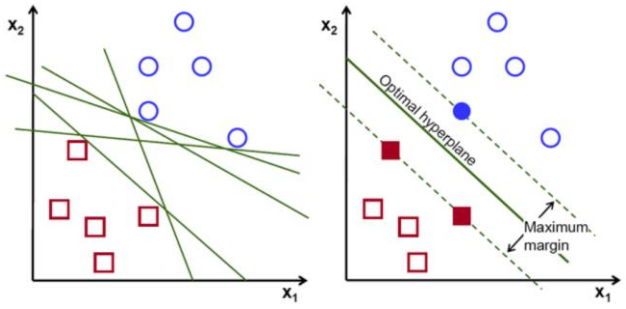

Figure. 32. Explanation of Hyper Plane in SVM (Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science n.d.)

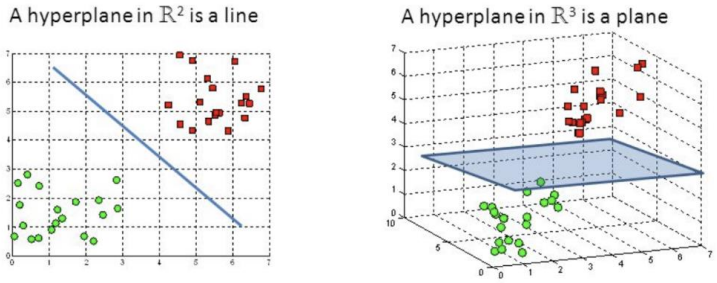

Figure. 33. Hyperplanes of SVM concerning Features of Data (Left Diagram-2 Features, Right Diagram-3 Features) (Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science n.d.)

SVM contains the Hyperparameters like Kernal, Gamma, C value, etc. A kernel is the function type that classifies the data by its shape. There are four types of kernels namely RBF (Radial Basis Function), Linear, Polynomial, and Sigmoid. RBF which is also called Gaussian Kernel is used to classify when there is no prior information about data. The linear kernel is suitable when the data is large and linearly separable. The polynomial kernel can be used if the data is not linear, and one class is the boundary for another class. Gamma is the coefficient of kernel function which tries to fit exactly which may lead to over-fitting. The c value is the penalty parameter that is used to make decision boundaries and classify data correctly. SVM is advantageous if the data has more features. It is also effective if the features are more than data points. It is least preferable if the data is large because it takes more time for training. Probability estimates cannot be extracted directly but can get them using the probability parameter. Noise affects the performance of the SVM models.

Component Analysis

The dimensional reduction techniques like Principal Component Analysis (PCA) and Information separation techniques like Independent Component Analysis (ICA) are applied to observe their effect on the prediction accuracy. PCA is used to reduce the dimensions of the data without losing the information. It is used to find the features that are applicable for maximum variance in the data. All the features obtained after applying PCA are orthogonal to each other. Generally, ICA will be preferred to do after PCA. ICA is used to separate information to be maximally independent. ICA is used to find the hidden factors in the features. The assumptions for applying ICA should be variables are non-gaussian and independent.

Testing of the data is done after modeling and training the data using parametric and non-parametric models. For testing cross-validation methods are performed. Cross-validation techniques like Test-train split and k-fold are performed to check the accuracies for different models. The data used for the testing is planned to be 30%. Based on the test results the overfitting and underfitting of models are evaluated.

3.7 Cross Validation

Validating the stability of the Machine Learning model is always a necessary task. We need some sort of assurance that models are performing well without over-fitting or under-fitting, picking the patterns, and leaving the noise while training. Cross Validation is the method used to statistically calculate the stability of Machine Learning models. It is the process to decide whether the results after testing the model are based on the relation between variables and the description of data. There will be a clear understanding of training error which is the differences between original responses and predicted responses. Some of the common types of Cross-Validation techniques are K-Fold, Stratified K-Fold, Leave-P-Out, Random Test Train Splits, etc.

In K-Fold cross-validation, the data will be divided into 'k' parts depending on the value of k. Each time one part in the k parts will be the test set and the remaining k-1 parts are merged by the model for training. The same process will be executed for all the k sets in the model. There will be k number of test scores at the end of model execution and can be used for checking the effectiveness of the model. Generally, 5 and 10 are the preferred k values for K-Fold Cross Validation. There might be some imbalance in the distribution of the classes in the dataset. For example, if there are double negative classes than positive classes in the Binary classification then an imbalance in the data takes place. In that case, Stratified K-Fold can be used. In Stratified K-Fold, each fold will be divided approximately with the same percentage of classes.

In Leave-P-Out Cross Validation, p number of observations are taken as a test set and the remaining data will be taken as a training set for the model. This process is repeated until all the combinations in the data are completed. The scores are taken into observation to check the performance of the model. If the p-value is 1, then each row will be taken as test input and the model will be trained total data size times until all the observations are tested. This is a very expensive and exhaustive method of Cross-Validation techniques.

Random Test Train Splits or Shuffle Splits is another type of Cross-Validation technique that slightly resembles the K-Fold method. Unlike K-Fold, this model divides the data into random test train split in each iteration instead of dividing folds at first. At each iteration, data is shuffled and split randomly for training and testing. Based on the number of iterations the training and testing of the model are done.

3.8 Evaluation

The evaluation metrics considered for the sentimental analysis based on classification are Accuracy, Precision, Recall, F1 Score, Classification Report, Confusion Matrix, and Area under Curve (AUC). All these metrics are derived for all the combinations of Vectorizations, Scaling techniques, Algorithms, Cross-validations, and Models with Hyperparameters.

a. Accuracy: It is the metric that calculates how accurately the algorithm classifies the points correctly. In classification accuracy is calculated on True Positives, True Negatives, False Positives, and False Negatives.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Neagatives}$$

b. Precision: It is one of the model performance indicators of the classification models. The positive prediction of the model is evaluated by this metric. It is calculated by True positives and False positives predicted by the model.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

c. Recall: It is the number of true positives found by the model. It is calculated by using True positives and False negatives.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

d. F1 Score: It is used to calculate the test accuracy of the model. It is calculated using the Precision and Recall of the model by taking the harmonic mean of them. Its highest possible value is 1.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

e. Classification Report: It is one of the performance evaluation metrics that include the model's Precision, Recall, F1 score, and Support for each class present in the target variable. Support is the number of actual class occurrences in the dataset

f. Confusion Matrix: It is the metric used to evaluate the predictions done by the model. The True positives, False positives, True Negatives, and False Negatives can be derived from this matrix. The number of rows and columns of the matrix depends on the number of classes in the response variable.

		Predicted class	
		Yes	**No**
Actual class	**Yes**	True Positive	False Negative
	No	False Positive	True Negative

Table. 6. Confusion Matrix Table of Classification in Supervised Learning.

g. Area Under Curve (AUC) - Receiver operating characteristic (ROC): It measures the ability of the classifier to differentiate between classes at various thresholds. The AUC-ROC function takes the outcomes to form the test dataset and predicts the probabilities of classes. ROC is the curve of probability and AUC is like the summary and area under the probability curve indicating the capacity of the model to distinguish the classes. Specificity and Sensitivity are used in finding the AUC-ROC curve. True Negative rate is called Specificity. True positive rate is called Sensitivity. Higher AUC-ROC indicates that the model is better at distinguishing the classes.

3.9 Hypothesis Testing

One of the common methods in selecting models in Machine Learning is comparing their performance between them. Evaluation is commonly done using metrics and cross-validation methods like K-Fold. The Mean test scores are compared after the validations to pick the best model and it is hard to know whether the mean scores are leading us in the right way. So, it is suggested to do statistical tests to quantify the test scores and finalize the assumptions along with cross-validation scores. The Null Hypothesis for the Statistical test is No difference in the models' performances. The Alternate Hypothesis is that there will be differences between models' performances. Based on the p-value obtained after the statistical test, the rejection or acceptance of the Null hypothesis takes place. This Hypothesis testing will boost confidence in the interpretation of the results (Statistical Significance Tests for Comparing Machine Learning Algorithms n.d.).

The Five times repeatable 2-Fold Cross-Validation Paired T-Test is conducted as the data used in the algorithms is the same. The Mlxtend library is used to carry the 5*2 Paired T-Test. P-value is the

evidence or significance level for the null hypothesis in Machine Learning. The significance level taken during the statistical tests is 5%. The Null hypothesis is taken with the assumption that there are no anomalies between the independent variables and dependent variables. To support the tests, the mean score of 10-Fold Cross-Validation is taken and plotted using Boxplot to visually check the difference between the mean scores of the Machine Learning models. Then the best model is taken by considering both Paired T-Test and 10-Fold Cross-Validation Scores to avoid Type-I errors (False Positive) and Type-II errors (False Negative).

3.10 Hyperparameter Tuning

The performance of Machine Learning models depends mostly on Feature Engineering and selecting the parameters based on the type of algorithm used for training and testing. Hyperparameter Tuning is useful in getting the best parameters of the model before training the model. Examples of Hyperparameters of models are K value for KNN, Kernel types, C value, Gamma value, etc for SVM, Estimators, Maximum Depth, etc for Random Forests, Penalty, C value, Maximum Iterations, etc for Logistic Regression, etc.

Models in Machine Learning are mainly affected by two factors namely Bias error and Variance error. Bias error also called Under-fitting is due to simplifying the assumptions of the model and Variance error also called Over-fitting is due to the randomness in the training set. The two errors can be controlled by the complexity of the model and managing the size of training data. Good hyperparameters help avoid under-fitting where train and test errors are high and over-fitting where train error is low and the test error is high. Random Search CV and Grid Search CV are the two best algorithms for Hyperparameter tuning (Hyperparameter Tuning | Evaluate ML Models with Hyperparameter Tuning n.d.).

Models will go as estimators in both Grid Search CV and Random Search CV methods. Grid Search CV is the basic hyperparameter tuning that uses all the parameter combinations given in the grid. It is the computationally expensive method for finding hyperparameters. For example, if 4 values are given for the C value and 4 Kernel values in the parameters dictionary for the SVM model, it will take all 16 combinations to find the best parameters. Random Search CV overcomes the computationally expensive limitation of Grid Search CV. In Random Search CV, not all parameter combinations will be applied but a fixed number of sampled parameters and combinations are performed through the n_iter parameter.

In this project, the Coarse to Fine-tuning method is used for finding the best Hyperparameters. Coarse Fine-tuning is the preferable method for finding the best parameters during Hyperparameter tuning. It uses both the capabilities of Random Search and Grid Search CV. In this method, Random Search CV is applied to find the suitable parameters within a large range of values then Grid Search is applied in the smaller range where we get the suitable values through Random Search. This method is continued until the best parameters are found.

4. Results

The results obtained after modeling and testing are compared between different parametric, and non-parametric models based on cross-validations, scaling techniques, dimensional reduction, Information separation techniques, and Models with Hyperparameters. The results are justified based on different evaluation methods for all the combinations of techniques and models of supervised learning classification. As per the practical evaluation results and theoretical concepts from section 3, the best model is considered. Section 4.1 contains the evaluation metrics of Kabita's Dataset and Section 4.2 contains the evaluation metrics of Nisha's dataset. Sections 4.1.1 and 4.1.2 contains the metrics of Machine Learning models of Kabita's dataset without Scaling, Component Analysis. Sections 4.2.1 and 4.2.2 contains the metrics of Machine Learning models of Nisha's dataset without Scaling,

Component Analysis. Section 4.1.3 and 4.1.4 contains the metrics of the Scaling and Component Analysis models of Kabita's dataset. Section 4.2.3 and 4.2.4 contains the metrics of the Scaling and Component Analysis models of Nisha's dataset.

Based on the best metrics of Normal models (without Scaling and Component Analysis), Scaled models, Component Analysis models, and 4 models each from both Kabita's and Nisha's Dataset are selected and Hyperparameter Tuning is done to improve the performance of models. Hypothesis testing is conducted using Paired T-Test to check whether the selected models are correct. Area Under Curve (AUC) is derived to check how good the model is between the dataset classes and Receiver Operating Characteristic (ROC) is plotted to check it visually.

4.1 Kabita's Kitchen Dataset

4.1.1 Bag of Word Models

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.76	0.75	0.76
Gaussian Naïve Bayes	0.57	0.57	0.57	0.54
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.70	0.71	0.70	0.70
SVM (Linear)	0.76	0.77	0.76	0.76
KNN (4 Neighbors)	0.56	0.60	0.56	0.55
Decision Tree	0.70	0.70	0.70	0.70
Random Forest	0.74	0.74	0.74	0.74

Table. 7. TF-IDF Vectorized Models and Metrics of Kabita's Dataset.

Count Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.76	0.75	0.75
Gaussian Naïve Bayes	0.53	0.55	0.53	0.50
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.71	0.72	0.71	0.71
SVM (Linear)	0.75	0.76	0.75	0.75
KNN (3 Neighbors)	0.60	0.68	0.60	0.58
Decision Tree	0.68	0.69	0.68	0.68
Random Forest	0.72	0.73	0.72	0.72

Table. 8. Count Vectorized Models and Metrics of Kabita's Dataset.

TF (Term Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.76	0.75	0.75
Gaussian Naïve Bayes	0.56	0.57	0.56	0.53
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.72	0.72	0.72	0.71
SVM (RBF)	0.76	0.77	0.76	0.76
KNN (3 Neighbors)	0.60	0.65	0.60	0.59
Decision Tree	0.68	0.69	0.68	0.68
Random Forest	0.74	0.74	0.74	0.74

Table. 9. TF Vectorized Models and Metrics of Kabita's Dataset.

4.1.2 Pre-Trained Transformer Models

BERT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.78	0.78	0.78	0.78
Gaussian Naïve Bayes	0.55	0.58	0.55	0.54
Bernoulli Naïve Bayes	0.53	0.56	0.53	0.51
Multinomial Naïve Bayes	0.51	0.55	0.51	0.49
SVM (Poly)	0.76	0.77	0.76	0.76
KNN (7 Neighbors)	0.70	0.69	0.70	0.69
Decision Tree (max depth-10)	0.61	0.61	0.61	0.61
Random Forest (max depth-18)	0.73	0.74	0.73	0.73

Table. 10. BERT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Ganesh BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.54	0.54	0.54	0.53
Gaussian Naïve Bayes	0.28	0.25	0.28	0.20
Bernoulli Naïve Bayes	0.28	0.22	0.28	0.21
Multinomial Naïve Bayes	0.27	0.31	0.27	0.20

SVM (Linear)	0.54	0.54	0.54	0.53
KNN (6 Neighbors)	0.47	0.46	0.47	0.45
Decision Tree (max depth-7)	0.46	0.50	0.46	0.45
Random Forest (max depth-14)	0.55	0.56	0.55	0.55

Table. 11. Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.76	0.76	0.76	0.76
Gaussian Naïve Bayes	0.54	0.55	0.54	0.53
Bernoulli Naïve Bayes	0.53	0.53	0.53	0.52
Multinomial Naïve Bayes	0.48	0.49	0.48	0.47
SVM (Linear)	0.75	0.75	0.75	0.75
KNN (7 Neighbors)	0.67	0.69	0.67	0.66
Decision Tree (max depth-6)	0.56	0.58	0.56	0.56
Random Forest (max depth-17)	0.71	0.72	0.71	0.71

Table. 12. Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Verloop BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.79	0.79	0.79	0.79
Gaussian Naïve Bayes	0.59	0.60	0.59	0.57
Bernoulli Naïve Bayes	0.59	0.61	0.59	0.58
Multinomial Naïve Bayes	0.56	0.57	0.56	0.55
SVM (Poly)	0.79	0.79	0.79	0.79
KNN (6 Neighbors)	0.69	0.70	0.69	0.68
Decision Tree (max depth-9)	0.54	0.55	0.54	0.55
Random Forest (max depth-16)	0.73	0.74	0.73	0.73

Table. 13. Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

GPT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.76	0.75	0.75
Gaussian Naïve Bayes	0.56	0.59	0.56	0.56
Bernoulli Naïve Bayes	0.55	0.57	0.55	0.55
Multinomial Naïve Bayes	0.55	0.57	0.55	0.54
SVM (RBF)	0.74	0.75	0.74	0.74
KNN (7 Neighbors)	0.66	0.67	0.66	0.65
Decision Tree (max depth-7)	0.54	0.56	0.54	0.55
Random Forest (max depth-13)	0.69	0.70	0.69	0.69

Table. 14. GPT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

XLM Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.76	0.76	0.76	0.76
Gaussian Naïve Bayes	0.60	0.62	0.60	0.60
Bernoulli Naïve Bayes	0.59	0.61	0.59	0.59
Multinomial Naïve Bayes	0.56	0.58	0.56	0.56
SVM (RBF)	0.77	0.78	0.77	0.77
KNN (7 Neighbors)	0.68	0.67	0.68	0.67
Decision Tree (max depth-8)	0.60	0.62	0.60	0.60
Random Forest	0.72	0.73	0.72	0.72

Table. 15. XLM Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned BERT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.69	0.69	0.69	0.69
Gaussian Naïve Bayes	0.45	0.50	0.45	0.44
Bernoulli Naïve Bayes	0.46	0.47	0.46	0.44
Multinomial Naïve Bayes	0.42	0.45	0.42	0.41

SVM (RBF)	0.68	0.69	0.68	0.68
KNN (6 Neighbors)	0.57	0.58	0.57	0.56
Decision Tree	0.44	0.44	0.44	0.44
Random Forest	0.60	0.61	0.60	0.59

Table. 16. Fine-Tuned BERT Base Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned BERT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.67	0.67	0.67	0.67
Gaussian Naïve Bayes	0.48	0.53	0.48	0.46
Bernoulli Naïve Bayes	0.47	0.51	0.47	0.46
Multinomial Naïve Bayes	0.46	0.49	0.46	0.44
SVM (RBF)	0.67	0.68	0.67	0.67
KNN (8 Neighbors)	0.63	0.65	0.63	0.62
Decision Tree (max depth-8)	0.51	0.53	0.51	0.51
Random Forest	0.64	0.65	0.64	0.64

Table. 17. Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned GPT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.76	0.77	0.76	0.77
Gaussian Naïve Bayes	0.54	0.55	0.54	0.52
Bernoulli Naïve Bayes	0.53	0.53	0.53	0.52
Multinomial Naïve Bayes	0.50	0.50	0.50	0.49
SVM (Linear)	0.74	0.74	0.74	0.74
KNN (6 Neighbors)	0.48	0.48	0.48	0.48
Decision Tree (max depth-9)	0.51	0.54	0.51	0.52
Random Forest (max depth-13)	0.68	0.69	0.68	0.68

Table. 18. Fine-Tuned GPT Base Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned GPT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.78	0.79	0.78	0.78
Gaussian Naïve Bayes	0.53	0.55	0.53	0.52
Bernoulli Naïve Bayes	0.52	0.55	0.52	0.52
Multinomial Naïve Bayes	0.52	0.56	0.52	0.52
SVM (Linear)	0.76	0.76	0.76	0.76
KNN (5 Neighbors)	0.47	0.48	0.47	0.47
Decision Tree (max depth-10)	0.52	0.52	0.52	0.52
Random Forest	0.69	0.71	0.69	0.70

Table. 19. Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned XLM Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.51	0.52	0.51	0.51
Gaussian Naïve Bayes	0.42	0.46	0.42	0.42
Bernoulli Naïve Bayes	0.39	0.43	0.39	0.39
Multinomial Naïve Bayes	0.40	0.42	0.40	0.39
SVM (RBF)	0.54	0.55	0.54	0.54
KNN (5 Neighbors)	0.46	0.47	0.46	0.46
Decision Tree (max depth-12)	0.37	0.38	0.37	0.37
Random Forest (max depth-17)	0.47	0.51	0.47	0.48

Table. 20. Fine-Tuned XLM Base Vectorized Models and Metrics of Kabita's Dataset.

4.1.3 Scaling Models

a. Min-Max Scaling

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.73	0.73	0.73
Gaussian Naïve Bayes	0.57	0.57	0.57	0.54
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.69	0.70	0.69	0.69
SVM (Sigmoid)	0.73	0.74	0.73	0.73
KNN (6 Neighbors)	0.59	0.61	0.59	0.57
Decision Tree	0.70	0.70	0.70	0.70
Random Forest	0.74	0.74	0.74	0.74

Table. 21. Min-Max Scaled TF-IDF Vectorized Models and Metrics of Kabita's Dataset.

Count Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.69	0.70	0.69	0.69
Gaussian Naïve Bayes	0.54	0.55	0.54	0.51
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.68	0.68	0.68	0.67
SVM (Sigmoid)	0.72	0.74	0.72	0.73
KNN (7 Neighbors)	0.60	0.66	0.60	0.58
Decision Tree	0.65	0.67	0.65	0.65
Random Forest	0.69	0.71	0.69	0.69

Table. 22. Min-Max Scaled Count Vectorized Models and Metrics of Kabita's Dataset.

TF (Term Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.74	0.74	0.74	0.74
Gaussian Naïve Bayes	0.56	0.56	0.56	0.53
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.70	0.70	0.70	0.70
SVM (Sigmoid)	0.74	0.75	0.74	0.74
KNN	0.62	0.64	0.62	0.61

(3 Neighbors)				
Decision Tree	0.68	0.69	0.68	0.68
Random Forest	0.74	0.75	0.74	0.74

Table. 23. Min-Max Scaled TF Vectorized Models and Metrics of Kabita's Dataset.

BERT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.69	0.74	0.69	0.66
Gaussian Naïve Bayes	0.55	0.59	0.55	0.55
Bernoulli Naïve Bayes	0.23	0.38	0.23	0.20
Multinomial Naïve Bayes	0.53	0.55	0.53	0.51
SVM (RBF)	0.76	0.77	0.76	0.76
KNN (6 Neighbors)	0.69	0.68	0.69	0.68
Decision Tree (max depth-9)	0.53	0.54	0.53	0.53
Random Forest (max depth-13)	0.71	0.72	0.71	0.71

Table. 24. Min-Max Scaled BERT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Ganesh BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.36	0.50	0.36	0.30
Gaussian Naïve Bayes	0.28	0.23	0.28	0.20
Bernoulli Naïve Bayes	0.19	0.31	0.19	0.14
Multinomial Naïve Bayes	0.28	0.23	0.28	0.20
SVM (Poly)	0.41	0.44	0.41	0.38
KNN (8 Neighbors)	0.41	0.40	0.41	0.40
Decision Tree (max depth-6)	0.28	0.36	0.28	0.25
Random Forest (max depth-6)	0.45	0.45	0.45	0.44

Table. 25. Min-Max Scaled Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.74	0.72	0.71
Gaussian Naïve	0.55	0.56	0.55	0.54

Bayes				
Bernoulli Naïve Bayes	0.25	0.40	0.25	0.23
Multinomial Naïve Bayes	0.49	0.50	0.49	0.48
SVM (RBF)	0.75	0.76	0.75	0.76
KNN (6 Neighbors)	0.67	0.68	0.67	0.66
Decision Tree (max depth-7)	0.50	0.50	0.50	0.50
Random Forest (max depth-11)	0.68	0.69	0.68	0.68

Table. 26. Min-Max Scaled Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Verloop BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.64	0.74	0.64	0.61
Gaussian Naïve Bayes	0.54	0.63	0.54	0.54
Bernoulli Naïve Bayes	0.18	0.28	0.18	0.12
Multinomial Naïve Bayes	0.52	0.55	0.52	0.51
SVM (RBF)	0.72	0.78	0.72	0.72
KNN (8 Neighbors)	0.69	0.70	0.69	0.68
Decision Tree (max depth-5)	0.42	0.47	0.42	0.42
Random Forest (max depth-11)	0.67	0.70	0.67	0.68

Table. 27. Min-Max Scaled Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

GPT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.68	0.72	0.68	0.68
Gaussian Naïve Bayes	0.51	0.60	0.51	0.52
Bernoulli Naïve Bayes	0.20	0.35	0.20	0.16
Multinomial Naïve Bayes	0.54	0.56	0.54	0.54
SVM (RBF)	0.73	0.77	0.73	0.74
KNN (6 Neighbors)	0.66	0.66	0.66	0.64
Decision Tree (max depth-7)	0.43	0.46	0.43	0.44

Random Forest (max depth-13)	0.68	0.70	0.68	0.68

Table. 28. Min-Max Scaled GPT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

XLM Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.71	0.72	0.71	0.70
Gaussian Naïve Bayes	0.61	0.64	0.61	0.61
Bernoulli Naïve Bayes	0.22	0.31	0.22	0.17
Multinomial Naïve Bayes	0.57	0.59	0.57	0.57
SVM (RBF)	0.77	0.79	0.77	0.77
KNN (7 Neighbors)	0.68	0.68	0.68	0.68
Decision Tree (max depth-8)	0.50	0.53	0.50	0.51
Random Forest (max depth-17)	0.72	0.73	0.72	0.72

Table. 29. Min-Max Scaled XLM Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned BERT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.59	0.71	0.59	0.58
Gaussian Naïve Bayes	0.46	0.48	0.46	0.45
Bernoulli Naïve Bayes	0.25	0.47	0.25	0.22
Multinomial Naïve Bayes	0.44	0.48	0.44	0.43
SVM (RBF)	0.69	0.70	0.69	0.69
KNN (8 Neighbors)	0.56	0.58	0.56	0.56
Decision Tree (max depth-12)	0.37	0.39	0.37	0.37
Random Forest (max depth-16)	0.59	0.60	0.59	0.59

Table. 30. Min-Max Scaled Fine-Tuned BERT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned BERT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.62	0.66	0.62	0.61
Gaussian Naïve Bayes	0.48	0.53	0.48	0.47
Bernoulli Naïve Bayes	0.20	0.26	0.20	0.16
Multinomial Naïve Bayes	0.47	0.49	0.47	0.45
SVM (RBF)	0.70	0.70	0.70	0.70
KNN (8 Neighbors)	0.62	0.63	0.62	0.61
Decision Tree (max depth-5)	0.42	0.44	0.42	0.40
Random Forest (max depth-19)	0.63	0.64	0.63	0.63

Table. 31. Min-Max Scaled Fine-Tuned BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned GPT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.75	0.73	0.73
Gaussian Naïve Bayes	0.52	0.54	0.52	0.51
Bernoulli Naïve Bayes	0.23	0.38	0.23	0.20
Multinomial Naïve Bayes	0.50	0.50	0.50	0.48
SVM (RBF)	0.75	0.76	0.75	0.75
KNN (7 Neighbors)	0.67	0.67	0.67	0.66
Decision Tree (max depth-8)	0.44	0.44	0.44	0.43
Random Forest (max depth-15)	0.65	0.67	0.65	0.65

Table. 32. Min-Max Scaled Fine-Tuned GPT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned GPT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.77	0.78	0.77	0.77
Gaussian Naïve Bayes	0.52	0.55	0.52	0.53
Bernoulli Naïve Bayes	0.25	0.42	0.25	0.23
Multinomial	0.51	0.55	0.51	0.50

Naïve Bayes				
SVM (RBF)	0.79	0.80	0.79	0.79
KNN (5 Neighbors)	0.66	0.66	0.66	0.65
Decision Tree (max depth-9)	0.46	0.47	0.46	0.47
Random Forest (max depth-16)	0.68	0.70	0.68	0.68

Table. 33. Min-Max Scaled Fine-Tuned GPT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned XLM Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.51	0.54	0.51	0.52
Gaussian Naïve Bayes	0.41	0.44	0.41	0.41
Bernoulli Naïve Bayes	0.26	0.34	0.26	0.26
Multinomial Naïve Bayes	0.39	0.41	0.39	0.39
SVM (RBF)	0.54	0.57	0.54	0.54
KNN (5 Neighbors)	0.46	0.47	0.46	0.46
Decision Tree (max depth-12)	0.32	0.35	0.32	0.33
Random Forest (max depth-9)	0.46	0.50	0.46	0.47

Table. 34. Min-Max Scaled Fine-Tuned XLM Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

b. Normalized Scaling

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.76	0.75	0.76
Gaussian Naïve Bayes	0.57	0.57	0.57	0.54
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.69	0.70	0.69	0.69
SVM (Linear)	0.76	0.77	0.76	0.76
KNN (3 Neighbors)	0.56	0.62	0.56	0.55
Decision Tree	0.70	0.70	0.70	0.70
Random Forest	0.74	0.74	0.74	0.74

Table. 35. Normalize Scaled TF-IDF Vectorized Models and Metrics of Kabita's Dataset.

Count Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.76	0.75	0.75
Gaussian Naïve Bayes	0.56	0.57	0.56	0.53
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.70	0.71	0.70	0.70
SVM (RBF)	0.76	0.77	0.76	0.76
KNN (3 Neighbors)	0.60	0.65	0.60	0.59
Decision Tree	0.68	0.69	0.68	0.68
Random Forest	0.74	0.74	0.74	0.74

Table. 36. Normalize Scaled Count Vectorized Models and Metrics of Kabita's Dataset.

TF (Term Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.76	0.75	0.75
Gaussian Naïve Bayes	0.56	0.57	0.56	0.53
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.70	0.71	0.70	0.70
SVM (RBF)	0.76	0.77	0.76	0.76
KNN (3 Neighbors)	0.61	0.65	0.61	0.60
Decision Tree	0.68	0.69	0.68	0.68
Random Forest	0.74	0.74	0.74	0.74

Table. 37. Normalize Scaled TF Vectorized Models and Metrics of Kabita's Dataset.

BERT Base model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.70	0.70	0.70	0.70
Gaussian Naïve Bayes	0.56	0.58	0.56	0.55
Bernoulli Naïve Bayes	0.56	0.57	0.56	0.55
Multinomial Naïve Bayes	0.53	0.55	0.53	0.51
SVM (Poly)	0.77	0.77	0.77	0.77
KNN (6 Neighbors)	0.69	0.68	0.69	0.68
Decision Tree (max depth-7)	0.59	0.60	0.59	0.59

Random Forest (max depth-14)	0.73	0.74	0.73	0.73

Table. 38. Normalize Scaled BERT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Ganesh BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.33	0.28	0.33	0.27
Gaussian Naïve Bayes	0.28	0.23	0.28	0.20
Bernoulli Naïve Bayes	0.28	0.22	0.28	0.20
Multinomial Naïve Bayes	0.27	0.22	0.27	0.19
SVM (Poly)	0.33	0.28	0.33	0.26
KNN (8 Neighbors)	0.47	0.47	0.47	0.46
Decision Tree (max depth-10)	0.47	0.47	0.47	0.47
Random Forest (max depth-14)	0.56	0.57	0.56	0.56

Table. 39. Normalize Scaled Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.70	0.70	0.70	0.69
Gaussian Naïve Bayes	0.54	0.55	0.54	0.53
Bernoulli Naïve Bayes	0.54	0.55	0.54	0.54
Multinomial Naïve Bayes	0.49	0.50	0.49	0.47
SVM (Poly)	0.76	0.76	0.76	0.75
KNN (6 Neighbors)	0.66	0.67	0.66	0.65
Decision Tree (max depth-11)	0.57	0.58	0.57	0.57
Random Forest	0.70	0.70	0.70	0.70

Table. 40. Normalize Scaled Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Verloop BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.76	0.75	0.75
Gaussian Naïve Bayes	0.56	0.60	0.56	0.56
Bernoulli Naïve Bayes	0.56	0.58	0.56	0.55
Multinomial Naïve Bayes	0.55	0.57	0.55	0.54
SVM (Poly)	0.80	0.80	0.80	0.80
KNN (6 Neighbors)	0.69	0.70	0.69	0.67
Decision Tree (max depth-9)	0.53	0.56	0.53	0.54
Random Forest (max depth-16)	0.72	0.73	0.72	0.72

Table. 41. Normalize Scaled Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

GPT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.57	0.60	0.57	0.57
Bernoulli Naïve Bayes	0.54	0.57	0.54	0.54
Multinomial Naïve Bayes	0.54	0.56	0.54	0.54
SVM (Poly)	0.76	0.76	0.76	0.76
KNN (5 Neighbors)	0.67	0.67	0.67	0.66
Decision Tree	0.52	0.52	0.52	0.52
Random Forest	0.70	0.71	0.70	0.70

Table. 42. Normalize Scaled GPT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

XLM Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.74	0.74	0.74	0.74
Gaussian Naïve Bayes	0.62	0.63	0.62	0.61
Bernoulli Naïve Bayes	0.61	0.62	0.61	0.61
Multinomial Naïve Bayes	0.60	0.61	0.60	0.59
SVM (Poly)	0.79	0.79	0.79	0.78

KNN (6 Neighbors)	0.69	0.69	0.69	0.68
Decision Tree (max depth-8)	0.59	0.60	0.59	0.59
Random Forest (max depth-16)	0.73	0.74	0.73	0.73

Table. 43. Normalize Scaled XLM Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned BERT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.65	0.65	0.65	0.64
Gaussian Naïve Bayes	0.48	0.52	0.48	0.47
Bernoulli Naïve Bayes	0.47	0.49	0.47	0.45
Multinomial Naïve Bayes	0.45	0.48	0.45	0.44
SVM (Poly)	0.70	0.70	0.70	0.70
KNN (8 Neighbors)	0.57	0.57	0.57	0.57
Decision Tree (max depth-12)	0.43	0.45	0.43	0.44
Random Forest	0.59	0.61	0.59	0.59

Table. 44. Normalize Scaled Fine-Tuned BERT Base Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned BERT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.67	0.67	0.67	0.66
Gaussian Naïve Bayes	0.47	0.52	0.47	0.45
Bernoulli Naïve Bayes	0.46	0.51	0.46	0.45
Multinomial Naïve Bayes	0.47	0.50	0.47	0.45
SVM (Poly)	0.70	0.70	0.70	0.70
KNN (6 Neighbors)	0.62	0.62	0.62	0.61
Decision Tree (max depth-9)	0.50	0.50	0.50	0.50
Random Forest	0.65	0.66	0.65	0.65

Table. 45. Normalize Scaled Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned GPT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.37	0.41	0.37	0.34
Gaussian Naïve Bayes	0.50	0.53	0.50	0.48
Bernoulli Naïve Bayes	0.50	0.50	0.50	0.49
Multinomial Naïve Bayes	0.50	0.53	0.50	0.50
SVM (Poly)	0.37	0.41	0.37	0.34
KNN (3 Neighbors)	0.52	0.54	0.52	0.51
Decision Tree (max depth-11)	0.51	0.53	0.51	0.52
Random Forest (max depth-12)	0.68	0.69	0.68	0.67

Table. 46. Normalize Scaled Fine-Tuned GPT Base Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned GPT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.29	0.35	0.29	0.25
Gaussian Naïve Bayes	0.52	0.55	0.52	0.51
Bernoulli Naïve Bayes	0.51	0.54	0.51	0.51
Multinomial Naïve Bayes	0.52	0.55	0.52	0.50
SVM (Poly)	0.30	0.33	0.30	0.23
KNN (8 Neighbors)	0.54	0.53	0.54	0.53
Decision Tree (max depth-12)	0.51	0.51	0.51	0.51
Random Forest (max depth-20)	0.70	0.71	0.70	0.70

Table. 47. Normalize Scaled Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned XLM Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.52	0.53	0.52	0.52
Gaussian Naïve Bayes	0.41	0.44	0.41	0.41
Bernoulli Naïve Bayes	0.40	0.44	0.40	0.40
Multinomial	0.39	0.40	0.39	0.38

Naïve Bayes				
SVM (Poly)	0.54	0.57	0.54	0.55
KNN (7 Neighbors)	0.47	0.48	0.47	0.47
Decision Tree (max depth-15)	0.38	0.39	0.38	0.38
Random Forest (max depth-12)	0.47	0.51	0.47	0.47

Table. 48. Normalize Scaled Fine-Tuned XLM Base Vectorized Models and Metrics of Kabita's Dataset.

c. Standard Scaling

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.69	0.69	0.69	0.69
Gaussian Naïve Bayes	0.21	0.25	0.21	0.16
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.69	0.70	0.69	0.69
SVM (Linear)	0.70	0.70	0.70	0.70
KNN (5 Neighbors)	0.55	0.56	0.55	0.54
Decision Tree	0.68	0.68	0.68	0.67
Random Forest	0.74	0.75	0.74	0.74

Table. 49. Standard Scaled TF-IDF Vectorized Models and Metrics of Kabita's Dataset.

Count Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.71	0.71	0.71	0.71
Gaussian Naïve Bayes	0.21	0.27	0.21	0.16
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.68	0.68	0.68	0.68
SVM (Sigmoid)	0.73	0.74	0.73	0.73
KNN (8 Neighbors)	0.58	0.59	0.58	0.55
Decision Tree	0.68	0.69	0.68	0.68
Random Forest	0.72	0.73	0.72	0.71

Table. 50. Standard Scaled Count Vectorized Models and Metrics of Kabita's Dataset.

TF (Term Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic	0.71	0.71	0.71	0.71

Regression				
Gaussian Naïve Bayes	0.21	0.27	0.21	0.16
Bernoulli Naïve Bayes	0.71	0.72	0.71	0.71
Multinomial Naïve Bayes	0.70	0.71	0.70	0.70
SVM (Sigmoid)	0.71	0.71	0.71	0.71
KNN (7 Neighbors)	0.56	0.58	0.56	0.55
Decision Tree	0.67	0.68	0.67	0.67
Random Forest	0.74	0.74	0.74	0.74

Table. 51. Standard Scaled TF Vectorized Models and Metrics of Kabita's Dataset.

BERT Base model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.76	0.76	0.76	0.76
Gaussian Naïve Bayes	0.57	0.59	0.57	0.55
Bernoulli Naïve Bayes	0.54	0.55	0.54	0.52
Multinomial Naïve Bayes	0.53	0.55	0.53	0.51
SVM (RBF)	0.77	0.77	0.77	0.77
KNN (8 Neighbors)	0.69	0.68	0.69	0.68
Decision Tree (max depth-10)	0.60	0.60	0.60	0.59
Random Forest (max depth-16)	0.72	0.73	0.72	0.72

Table. 52. Standard Scaled BERT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Ganesh BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.59	0.60	0.59	0.58
Gaussian Naïve Bayes	0.28	0.23	0.28	0.20
Bernoulli Naïve Bayes	0.28	0.22	0.28	0.20
Multinomial Naïve Bayes	0.28	0.23	0.28	0.20
SVM (Linear)	0.60	0.61	0.60	0.60
KNN (6 Neighbors)	0.48	0.48	0.48	0.47
Decision Tree (max depth-6)	0.43	0.43	0.43	0.42
Random Forest	0.53	0.53	0.53	0.53

(max depth-12)				

Table. 53. Standard Scaled Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.75	0.75	0.75
Gaussian Naïve Bayes	0.54	0.55	0.54	0.54
Bernoulli Naïve Bayes	0.51	0.52	0.51	0.50
Multinomial Naïve Bayes	0.49	0.50	0.49	0.48
SVM (RBF)	0.77	0.78	0.77	0.77
KNN (6 Neighbors)	0.67	0.68	0.67	0.66
Decision Tree (max depth-13)	0.56	0.56	0.56	0.56
Random Forest	0.70	0.71	0.70	0.70

Table. 54. Standard Scaled Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Verloop BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.77	0.77	0.77	0.77
Gaussian Naïve Bayes	0.56	0.58	0.56	0.55
Bernoulli Naïve Bayes	0.52	0.54	0.52	0.51
Multinomial Naïve Bayes	0.52	0.55	0.52	0.51
SVM (RBF)	0.80	0.80	0.80	0.80
KNN (8 Neighbors)	0.68	0.70	0.68	0.67
Decision Tree (max depth-9)	0.54	0.55	0.54	0.54
Random Forest (max depth-20)	0.72	0.73	0.72	0.72

Table. 55. Standard Scaled Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

GPT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.76	0.76	0.76	0.76
Gaussian Naïve Bayes	0.55	0.57	0.55	0.55
Bernoulli Naïve Bayes	0.53	0.55	0.53	0.53
Multinomial Naïve Bayes	0.54	0.56	0.54	0.54
SVM (RBF)	0.77	0.77	0.77	0.77
KNN (8 Neighbors)	0.66	0.65	0.66	0.64
Decision Tree (max depth-6)	0.51	0.53	0.51	0.52
Random Forest (max depth-16)	0.70	0.70	0.70	0.69

Table. 56. Standard Scaled GPT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

XLM Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.76	0.76	0.76	0.76
Gaussian Naïve Bayes	0.60	0.62	0.60	0.60
Bernoulli Naïve Bayes	0.58	0.59	0.58	0.58
Multinomial Naïve Bayes	0.57	0.59	0.57	0.57
SVM (RBF)	0.79	0.79	0.79	0.79
KNN (6 Neighbors)	0.68	0.67	0.68	0.68
Decision Tree (max depth-7)	0.58	0.59	0.58	0.58
Random Forest (max depth-17)	0.73	0.75	0.73	0.74

Table. 57. Standard Scaled XLM Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned BERT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.68	0.68	0.68	0.68
Gaussian Naïve Bayes	0.45	0.49	0.45	0.43
Bernoulli Naïve Bayes	0.46	0.47	0.46	0.45
Multinomial	0.44	0.48	0.44	0.43

Naïve Bayes				
SVM (RBF)	0.70	0.70	0.70	0.70
KNN (7 Neighbors)	0.57	0.58	0.57	0.57
Decision Tree (max depth-17)	0.42	0.42	0.42	0.42
Random Forest	0.59	0.60	0.59	0.59

Table. 58. Standard Scaled Fine-Tuned BERT Base Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned BERT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.65	0.65	0.65	0.65
Gaussian Naïve Bayes	0.48	0.53	0.48	0.47
Bernoulli Naïve Bayes	0.46	0.49	0.46	0.45
Multinomial Naïve Bayes	0.47	0.49	0.47	0.45
SVM (RBF)	0.70	0.71	0.70	0.70
KNN (4 Neighbors)	0.61	0.62	0.61	0.60
Decision Tree (max depth-9)	0.47	0.49	0.47	0.47
Random Forest (max depth-13)	0.61	0.63	0.61	0.61

Table. 59. Standard Scaled Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned GPT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.74	0.74	0.74	0.74
Gaussian Naïve Bayes	0.51	0.52	0.51	0.50
Bernoulli Naïve Bayes	0.49	0.49	0.49	0.47
Multinomial Naïve Bayes	0.50	0.50	0.50	0.48
SVM (RBF)	0.76	0.77	0.76	0.76
KNN (8 Neighbors)	0.66	0.66	0.66	0.65
Decision Tree (max depth-19)	0.48	0.48	0.48	0.48
Random Forest (max depth-18)	0.67	0.68	0.67	0.67

Table. 60. Standard Scaled Fine-Tuned GPT Base Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned GPT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.78	0.79	0.78	0.78
Gaussian Naïve Bayes	0.52	0.55	0.52	0.51
Bernoulli Naïve Bayes	0.50	0.53	0.50	0.50
Multinomial Naïve Bayes	0.51	0.55	0.51	0.50
SVM (RBF)	0.80	0.80	0.80	0.80
KNN (5 Neighbors)	0.66	0.67	0.66	0.65
Decision Tree (max depth-13)	0.53	0.53	0.53	0.53
Random Forest (max depth-18)	0.69	0.71	0.69	0.69

Table. 61. Standard Scaled Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Kabita's Dataset.

Fine-Tuned XLM Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.51	0.51	0.51	0.51
Gaussian Naïve Bayes	0.41	0.44	0.41	0.41
Bernoulli Naïve Bayes	0.40	0.42	0.40	0.39
Multinomial Naïve Bayes	0.39	0.41	0.39	0.39
SVM (RBF)	0.53	0.55	0.53	0.53
KNN (4 Neighbors)	0.46	0.46	0.46	0.45
Decision Tree (max depth-18)	0.36	0.37	0.36	0.37
Random Forest (max depth-17)	0.47	0.50	0.47	0.48

Table. 62. Standard Scaled Fine-Tuned XLM Base Vectorized Models and Metrics of Kabita's Dataset.

4.1.4 Principal Component and Independent Component Analysis Models

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

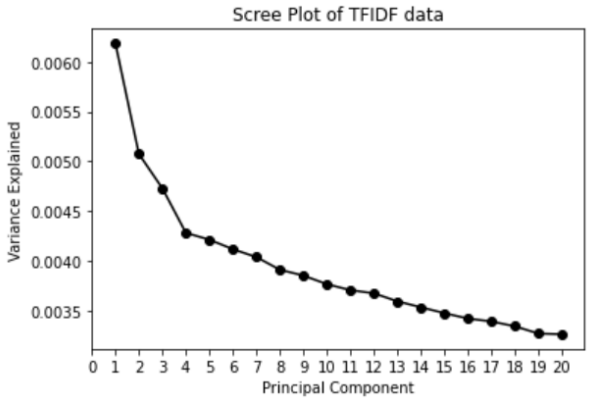

Figure. 34. Scree Plot for TF-IDF Vectors of Kabita's Dataset.

According to the Scree plot of Figure 24, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.38	0.41	0.38	0.31
Gaussian Naïve Bayes	0.36	0.34	0.36	0.32
Bernoulli Naïve Bayes	0.34	0.32	0.34	0.28
Multinomial Naïve Bayes	0.24	0.26	0.24	0.16
SVM (RBF)	0.47	0.44	0.47	0.44
KNN (8 Neighbors)	0.55	0.55	0.55	0.55
Decision Tree (max depth-10)	0.53	0.55	0.53	0.53
Random Forest (max depth-16)	0.59	0.60	0.59	0.59

Table. 63. TF-IDF Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Count Vectorized Model

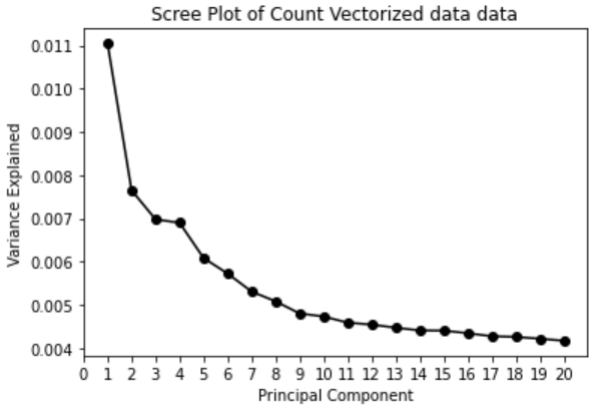

Figure. 35. Scree Plot for Count Vectors of Kabita's Dataset.

According to the Scree plot of Figure 25, 5 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.37	0.33	0.37	0.25
Gaussian Naïve Bayes	0.28	0.28	0.28	0.24
Bernoulli Naïve Bayes	0.36	0.29	0.36	0.27
Multinomial Naïve Bayes	0.20	0.13	0.20	0.11
SVM (RBF)	0.45	0.43	0.45	0.43
KNN (5 Neighbors)	0.53	0.53	0.53	0.53
Decision Tree (max depth-16)	0.53	0.54	0.53	0.54
Random Forest (max depth-11)	0.59	0.60	0.59	0.59

Table. 64. Count Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

TF (Term Frequency) Vectorized Model

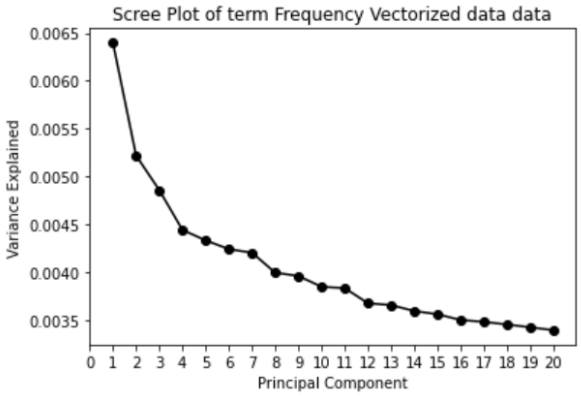

Figure. 36. Scree Plot for TF Vectors of Kabita's Dataset.

According to the Scree plot of Figure 26, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.29	0.25	0.29	0.22
Gaussian Naïve Bayes	0.34	0.33	0.34	0.30
Bernoulli Naïve Bayes	0.33	0.23	0.33	0.24
Multinomial Naïve Bayes	0.28	0.35	0.28	0.23
SVM (RBF)	0.41	0.39	0.41	0.38
KNN (5 Neighbors)	0.55	0.55	0.55	0.55
Decision Tree (max depth-14)	0.55	0.55	0.55	0.55
Random Forest	0.59	0.61	0.59	0.59

(max depth-9)				

Table. 65. TF Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

BERT Base model (Sentence Transformer)

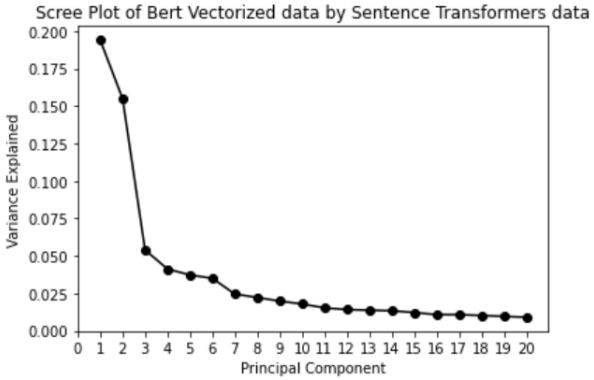

Figure. 37. Scree Plot for Count Vectors of Kabita's Dataset.

According to the Scree plot of Figure 27, 3 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.42	0.45	0.42	0.39
Gaussian Naïve Bayes	0.48	0.44	0.48	0.44
Bernoulli Naïve Bayes	0.45	0.41	0.45	0.42
Multinomial Naïve Bayes	0.41	0.39	0.41	0.37
SVM (RBF)	0.53	0.53	0.53	0.52
KNN (8 Neighbors)	0.54	0.53	0.54	0.53
Decision Tree (max depth-7)	0.53	0.54	0.53	0.53
Random Forest (max depth-14)	0.58	0.58	0.58	0.57

Table. 66. BERT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Ganesh BERT Hinglish Model (Sentence Transformer)

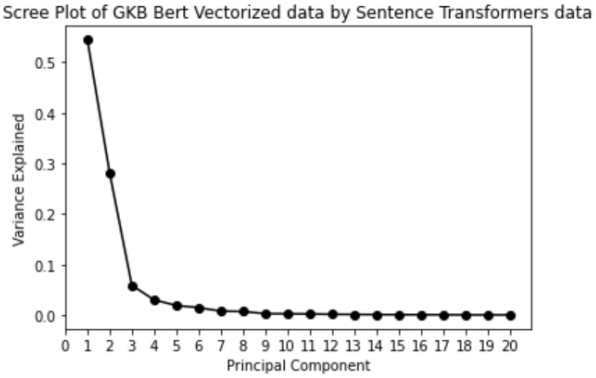

Figure. 38. Scree Plot for Ganesh BERT Hinglish Vectors (Sentence Transformer) of Kabita's Dataset.

According to the Scree plot of Figure 28, 3 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.29	0.16	0.29	0.17
Gaussian Naïve Bayes	0.29	0.21	0.29	0.21
Bernoulli Naïve Bayes	0.30	0.13	0.30	0.18
Multinomial Naïve Bayes	0.28	0.18	0.28	0.18
SVM (RBF)	0.33	0.29	0.33	0.27
KNN (6 Neighbors)	0.39	0.39	0.39	0.39
Decision Tree (max depth-12)	0.41	0.43	0.41	0.42
Random Forest (max depth-10)	0.45	0.46	0.45	0.45

Table. 67. Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

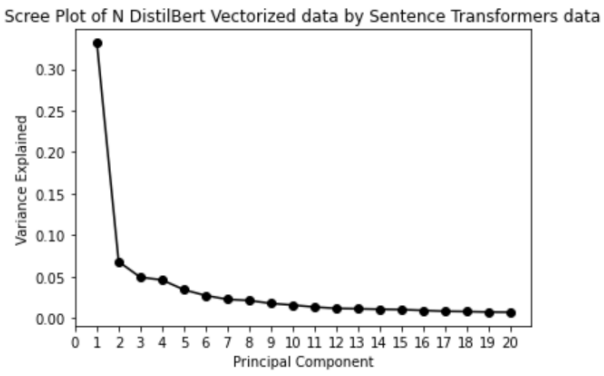

Figure. 39. Scree Plot for Narasimha Distil BERT Hinglish Vectors (Sentence Transformer) of Kabita's Dataset.

According to the Scree plot of Figure 29, 2 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.33	0.18	0.33	0.23
Gaussian Naïve Bayes	0.35	0.33	0.35	0.29
Bernoulli Naïve Bayes	0.28	0.16	0.28	0.21
Multinomial Naïve Bayes	0.26	0.27	0.26	0.21
SVM (RBF)	0.41	0.39	0.41	0.37
KNN (8 Neighbors)	0.44	0.43	0.44	0.44
Decision Tree (max depth-10)	0.44	0.45	0.44	0.44
Random Forest (max depth-8)	0.47	0.47	0.47	0.46

Table. 68. Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Verloop BERT Hinglish Model (Sentence Transformer)

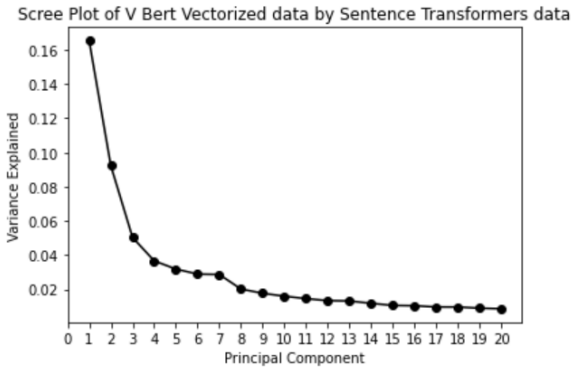

Figure. 40. Scree Plot for Verloop BERT Hinglish Vectors (Sentence Transformer) of Kabita's Dataset.

According to the Scree plot of Figure 30, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.44	0.35	0.44	0.36
Gaussian Naïve Bayes	0.50	0.49	0.50	0.48
Bernoulli Naïve Bayes	0.41	0.43	0.41	0.39
Multinomial Naïve Bayes	0.38	0.38	0.38	0.36
SVM (RBF)	0.58	0.57	0.58	0.57
KNN (5 Neighbors)	0.56	0.55	0.56	0.55
Decision Tree	0.53	0.54	0.53	0.53

(max depth-8)				
Random Forest (max depth-10)	0.58	0.57	0.58	0.57

Table. 69. Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

GPT Base Model (Sentence Transformer)

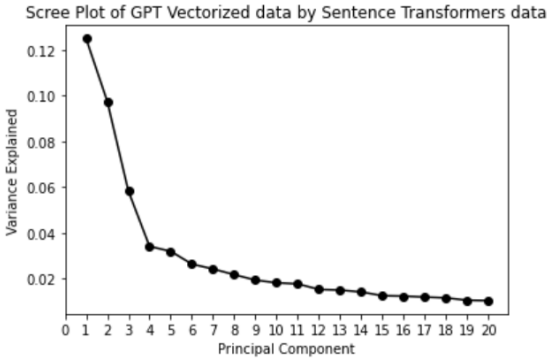

Figure. 41. Scree Plot for GPT Base Vectors (Sentence Transformer) of Kabita's Dataset.

According to the Scree plot of Figure 31, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.41	0.37	0.41	0.35
Gaussian Naïve Bayes	0.42	0.40	0.42	0.39
Bernoulli Naïve Bayes	0.39	0.42	0.39	0.36
Multinomial Naïve Bayes	0.38	0.39	0.38	0.37
SVM (RBF)	0.50	0.49	0.50	0.49
KNN (8 Neighbors)	0.47	0.46	0.47	0.47
Decision Tree (max depth-8)	0.48	0.48	0.48	0.47
Random Forest (max depth-15)	0.51	0.51	0.51	0.50

Table. 70. GPT Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

XLM Base Model (Sentence Transformer)

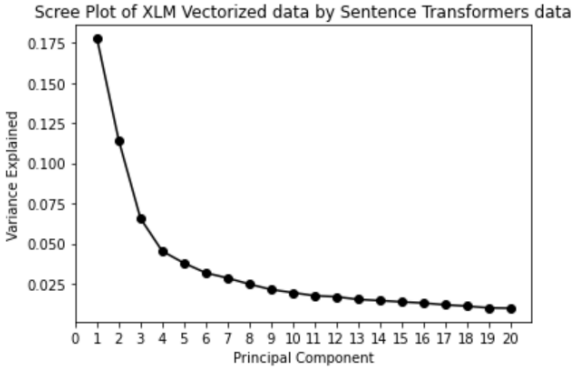

Figure. 42. Scree Plot for XLM Base Vectors (Sentence Transformer) of Kabita's Dataset.

According to the Scree plot of Figure 32, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.52	0.51	0.52	0.49
Gaussian Naïve Bayes	0.54	0.54	0.54	0.53
Bernoulli Naïve Bayes	0.43	0.44	0.43	0.41
Multinomial Naïve Bayes	0.49	0.50	0.49	0.49
SVM (RBF)	0.59	0.60	0.59	0.59
KNN (7 Neighbors)	0.57	0.56	0.57	0.56
Decision Tree (max depth-7)	0.55	0.56	0.55	0.55
Random Forest (max depth-12)	0.59	0.59	0.59	0.59

Table. 71. XLM Base (Sentence Transformer) Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Fine-Tuned BERT Base Model

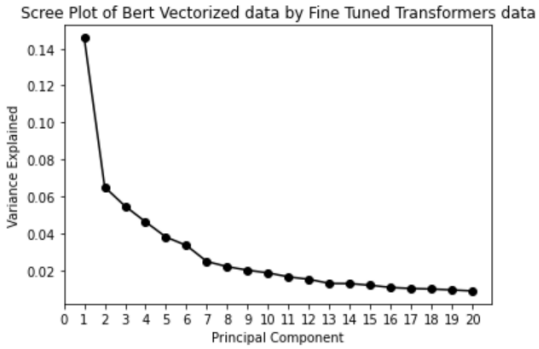

Figure. 43. Scree Plot for Fine-Tuned BERT Base Vectors of Kabita's Dataset.

According to the Scree plot of Figure 33, 7 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.36	0.37	0.36	0.30
Gaussian Naïve Bayes	0.35	0.35	0.35	0.32
Bernoulli Naïve Bayes	0.34	0.33	0.34	0.30
Multinomial Naïve Bayes	0.34	0.33	0.34	0.31
SVM (RBF)	0.43	0.43	0.43	0.41
KNN (5 Neighbors)	0.41	0.42	0.41	0.41
Decision Tree (max depth-12)	0.39	0.42	0.39	0.40
Random Forest (max depth-19)	0.47	0.48	0.47	0.47

Table. 72. Fine-Tuned BERT Base Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Fine-Tuned BERT Hinglish Model

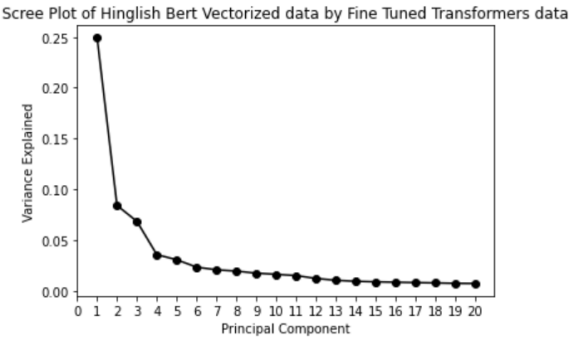

Figure. 44. Scree Plot for Fine-Tuned BERT Hinglish Vectors of Kabita's Dataset.

According to the Scree plot of Figure 34, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.41	0.41	0.41	0.31
Gaussian Naïve Bayes	0.42	0.40	0.42	0.38
Bernoulli Naïve Bayes	0.36	0.26	0.36	0.28
Multinomial Naïve Bayes	0.36	0.37	0.36	0.32
SVM (RBF)	0.49	0.47	0.49	0.46
KNN (8 Neighbors)	0.48	0.47	0.48	0.48
Decision Tree (max depth-6)	0.44	0.45	0.44	0.44
Random Forest	0.50	0.50	0.50	0.49

(max depth-13)				

Table. 73. Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Fine-Tuned GPT Base Model

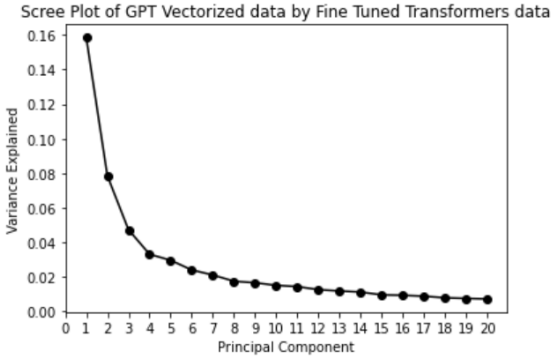

Figure. 45. Scree Plot for Fine-Tuned GPT Base Vectors of Kabita's Dataset.

According to the Scree plot of Figure 35, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.43	0.35	0.43	0.36
Gaussian Naïve Bayes	0.45	0.44	0.45	0.43
Bernoulli Naïve Bayes	0.36	0.31	0.36	0.29
Multinomial Naïve Bayes	0.37	0.33	0.37	0.33
SVM (RBF)	0.54	0.52	0.54	0.52
KNN (5 Neighbors)	0.52	0.52	0.52	0.52
Decision Tree (max depth-7)	0.50	0.51	0.50	0.50
Random Forest (max depth-10)	0.55	0.55	0.55	0.55

Table. 74. Fine-Tuned GPT Base Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Fine-Tuned GPT Hinglish Model

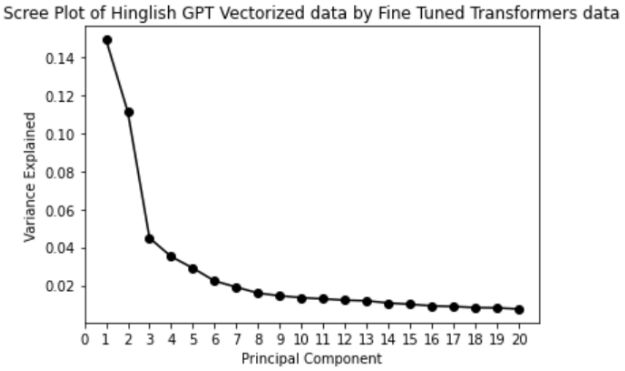

Figure. 46. Scree Plot for Fine-Tuned GPT Hinglish Vectors of Kabita's Dataset.

According to the Scree plot of Figure 36, 3 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.38	0.33	0.38	0.29
Gaussian Naïve Bayes	0.41	0.41	0.41	0.37
Bernoulli Naïve Bayes	0.35	0.26	0.35	0.28
Multinomial Naïve Bayes	0.33	0.33	0.33	0.31
SVM (RBF)	0.49	0.48	0.49	0.46
KNN (6 Neighbors)	0.46	0.45	0.46	0.46
Decision Tree (max depth-10)	0.47	0.47	0.47	0.47
Random Forest (max depth-10)	0.52	0.51	0.52	0.50

Table. 75. Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

Fine-Tuned XLM Base Model

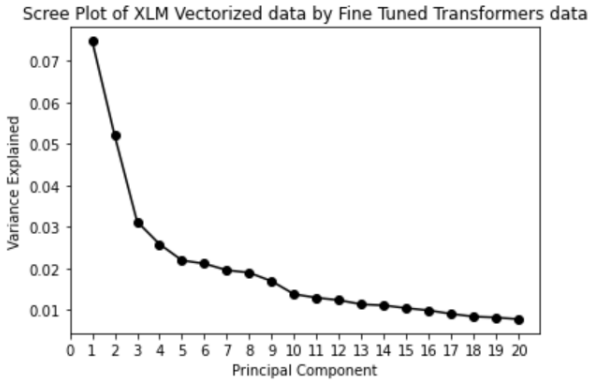

Figure. 47. Scree Plot for Fine-Tuned XLM Base Vectors of Kabita's Dataset.

According to the Scree plot of Figure 37, 5 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.35	0.35	0.35	0.32
Gaussian Naïve Bayes	0.32	0.36	0.32	0.31
Bernoulli Naïve Bayes	0.31	0.29	0.31	0.30
Multinomial Naïve Bayes	0.34	0.34	0.34	0.32
SVM (Poly)	0.37	0.41	0.37	0.37
KNN (8 Neighbors)	0.38	0.38	0.38	0.38
Decision Tree (max depth-12)	0.37	0.38	0.37	0.37
Random Forest (max depth-12)	0.42	0.44	0.42	0.42

Table. 76. Fine-Tuned XLM Base Vectorized Models and Metrics of Kabita's Dataset after PCA and ICA.

4.1.5 Hypothesis Testing of Models

Based on the metrics of Models without Scaling and Component Analysis, Scaled Models and Component Analysis applied Models, the best metrics are obtained for Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM Classification Model (Accuracy-80%), Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM Classification Model (Accuracy-80%), Standard Scaled Fine-Tuned GPT Hinglish Transformer - SVM Classification Model (Accuracy-80%), Verloop BERT Hinglish Sentence Transformer without Scaling – Logistic Regression Model (Accuracy-79%).

Some Models like SVM Classification and Logistic Regression of Normalize Scaled Verloop BERT Hinglish Sentence Transformer and Standard Scaled Fine-Tuned GPT Hinglish Transformer have metrics showing similar performance. To check the performance of the models statistically, a Hypothesis test is conducted between the models like Standardized Fine-Tuned GPT Hinglish Transformer's Logistic Regression and SVM and Normalized Fine-Tuned Verloop BERT Hinglish Sentence Transformer's Logistic Regression and SVM. The results of the Hypothesis test using Paired T-Test and Cross-Validation for the models are conducted.

Standard Scaled Fine-Tuned GPT Hinglish – Logistic Regression and SVM

The Paired T-Test is conducted for both the models for alpha 0.05 value. The t-statistic obtained is 1.45 and the p-value obtained is 0.207. As the p-value is greater than 0.05, the Null hypothesis is not rejected and there is no difference between the models' performance. To find the performance of the model more accurately, Mean Accuracies are calculated after k-fold cross-validation using 10-fold. The Accuracies are visualized using Boxplots which are shown in Figure 38. As SVM has good mean accuracy than Logistic Regression, SVM is considered as best Algorithm for the Standard Scaled Fine-Tuned GPT Hinglish dataset.

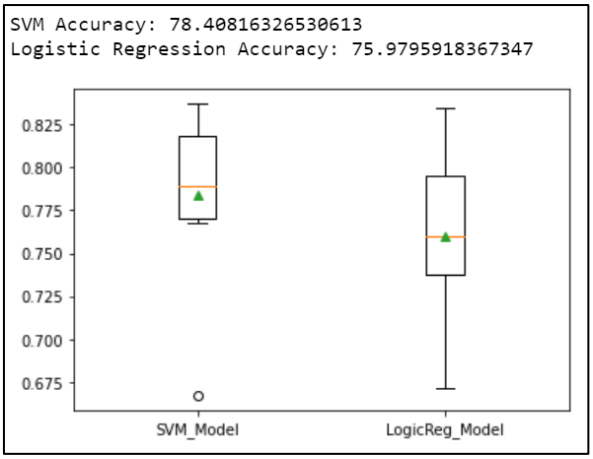

Figure. 48. Mean Accuracies of SVM and Logistic Regression

Normalize Scaled Verloop BERT Hinglish Sentence Transformer – Logistic Regression and SVM

The Paired T-Test is conducted for both the models for alpha 0.05 value. The t-statistic obtained is 9.930 and the p-value obtained is 0. As the p-value is less than 0.05, the Null hypothesis is rejected and there is a difference between the models' performance. To find the performance of the model more accurately, Mean Accuracies are calculated after k-fold cross-validation using 10-fold. The Accuracies are visualized using Boxplots which are shown in Figure 39. As SVM has good mean accuracy than Logistic Regression, SVM is considered as best Algorithm for the Standard Scaled Fine-Tuned GPT Hinglish dataset.

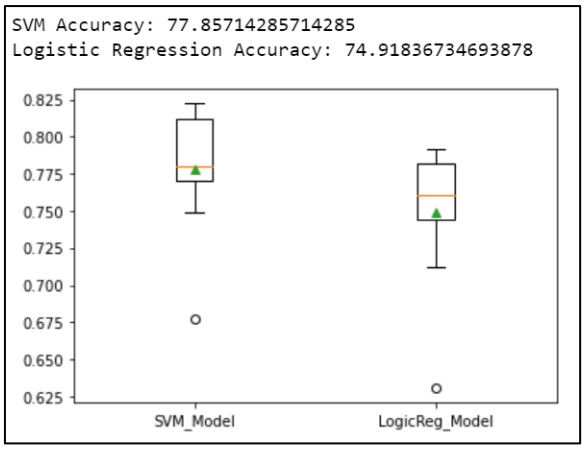

Figure. 49. Mean Accuracies of SVM and Logistic Regression

4.1.6 Hyperparameter Tuning

Hyperparameter Tuning is done for the 4 best models to check whether there is improvement in the performance of the models namely Normalize Scaled Verloop BERT Hinglish Sentence Transformer - SVM Model, Standard Scaled Verloop BERT Hinglish Sentence Transformer - SVM Model, Standard Scaled Fine-Tuned GPT Hinglish Transformer - SVM Model, Verloop BERT Hinglish Sentence Transformer without Scaling – Logistic Regression Model. Coarse to Fine Tuning is done while performing Hyperparameter Tuning. Grid Search CV is done after finding the Hyperparameters through Random Search CV. The best parameters found for the 4 models are mentioned in Table 78 along with the metrics.

Model and Parameters	Accuracy	Precision	Recall	F1-Score
Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM (C=10, gamma=1)	0.82	0.82	0.82	0.82
Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM (C=100)	0.82	0.82	0.82	0.82
Standard Scaled Fine-Tuned GPT Hinglish Transformer – SVM (C=10, gamma='auto')	0.81	0.81	0.81	0.81
Verloop BERT Hinglish Sentence Transformer – Logistic Regression (C=1, max_iter=3000)	0.79	0.79	0.79	0.79

Table. 77. Hyperparameter Tuned Best Models and Metrics of Kabita's Dataset.

4.1.7 AUC-ROC Curves

The Area Under Curve for the best models of Kabita's dataset is mentioned visually in the Receiver Operation Characteristic Curve plots. Based on the Visual Plots from Figure 40, Figure 41, Figure 42, and Figure 43, the results are clear that Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM, Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM, Standard Scaled Fine-Tuned GPT Hinglish Transformer – SVM, No Scaled Verloop BERT Hinglish Sentence Transformer – Logistic Regression are best models for Nisha's Dataset.

Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM

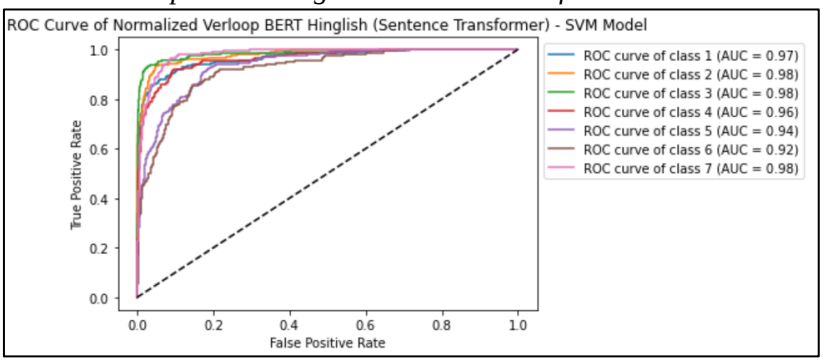

Figure. 50. ROC Curves for Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM of Kabita's Dataset

Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM

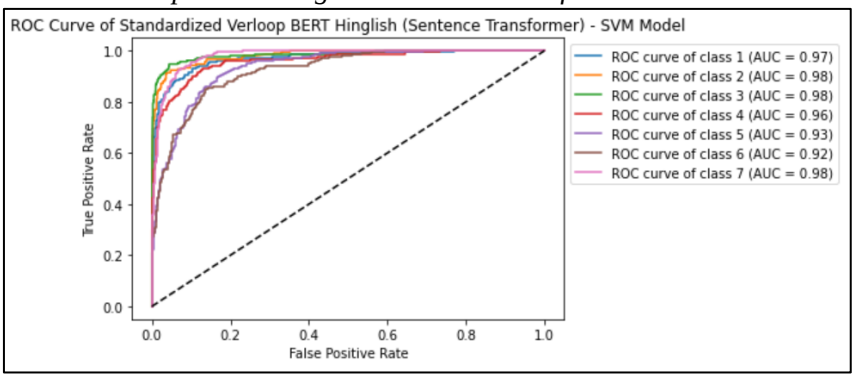

Figure. 51. ROC Curves for Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM of Kabita's Dataset

Standard Scaled Fine-Tuned GPT Hinglish Transformer – SVM

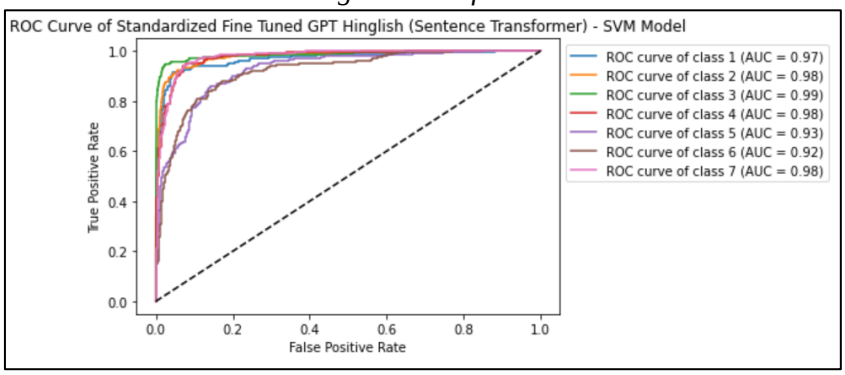

Figure. 52. ROC Curves for Standard Scaled Fine-Tuned GPT Hinglish Transformer – SVM of Kabita's Dataset

Verloop BERT Hinglish Sentence Transformer – Logistic Regression

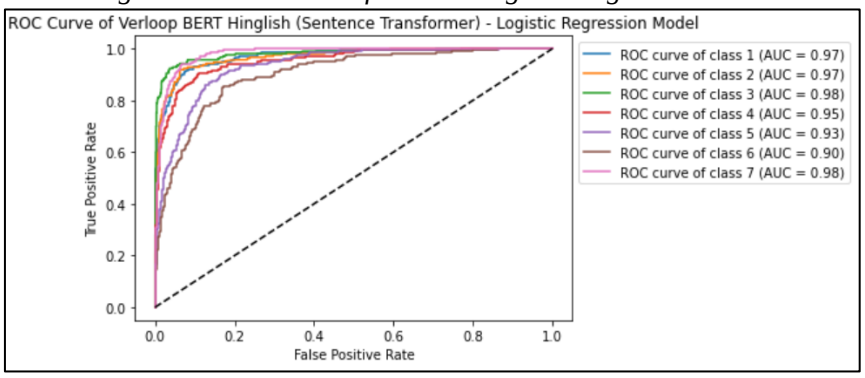

Figure. 53. ROC Curves for Verloop BERT Hinglish Sentence Transformer – Logistic Regression of Kabita's Dataset

4.2 Nisha's Dataset

4.2.1 Bag of Word Models

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.74	0.73	0.73
Gaussian Naïve Bayes	0.53	0.50	0.53	0.49
Bernoulli Naïve Bayes	0.70	0.70	0.70	0.69
Multinomial Naïve Bayes	0.69	0.68	0.69	0.68
SVM (Linear)	0.74	0.74	0.74	0.74
KNN (4 Neighbors)	0.52	0.55	0.52	0.50
Decision Tree	0.65	0.65	0.65	0.65
Random Forest	0.71	0.71	0.71	0.71

Table. 78. TF-IDF Vectorized Models and Metrics of Nisha's Dataset.

Count Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.73	0.73	0.73
Gaussian Naïve Bayes	0.47	0.47	0.47	0.43
Bernoulli Naïve Bayes	0.69	0.70	0.69	0.68
Multinomial Naïve Bayes	0.69	0.68	0.69	0.68
SVM (Linear)	0.74	0.74	0.74	0.74
KNN (5 Neighbors)	0.52	0.59	0.52	0.49

Decision Tree	0.63	0.64	0.63	0.63
Random Forest	0.68	0.69	0.68	0.67

Table. 79. Count Vectorized Models and Metrics of Nisha's Dataset.

TF (Term Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.50	0.48	0.50	0.46
Bernoulli Naïve Bayes	0.69	0.70	0.69	0.68
Multinomial Naïve Bayes	0.70	0.70	0.70	0.69
SVM (Linear)	0.72	0.73	0.72	0.72
KNN (5 Neighbors)	0.55	0.59	0.55	0.54
Decision Tree	0.65	0.65	0.65	0.65
Random Forest	0.71	0.71	0.71	0.71

Table. 80. TF Vectorized Models and Metrics of Nisha's Dataset.

4.2.2 Pre-Trained Transformer Models

BERT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.73	0.73	0.73
Gaussian Naïve Bayes	0.55	0.58	0.55	0.54
Bernoulli Naïve Bayes	0.53	0.56	0.53	0.52
Multinomial Naïve Bayes	0.51	0.54	0.51	0.49
SVM (Linear)	0.72	0.72	0.72	0.72
KNN (7 Neighbors)	0.66	0.66	0.66	0.65
Decision Tree (max depth-8)	0.55	0.56	0.55	0.55
Random Forest (max depth-15)	0.69	0.70	0.69	0.69

Table. 81. BERT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Ganesh BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.56	0.55	0.56	0.55
Gaussian Naïve Bayes	0.27	0.16	0.27	0.16
Bernoulli Naïve	0.27	0.19	0.27	0.19

Bayes				
Multinomial Naïve Bayes	0.26	0.18	0.26	0.18
SVM (Linear)	0.55	0.56	0.55	0.55
KNN (4 Neighbors)	0.46	0.46	0.46	0.46
Decision Tree (max depth-8)	0.48	0.49	0.48	0.48
Random Forest (max depth-15)	0.54	0.55	0.54	0.54

Table. 82. Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.71	0.71	0.71	0.71
Gaussian Naïve Bayes	0.52	0.53	0.52	0.51
Bernoulli Naïve Bayes	0.52	0.53	0.52	0.51
Multinomial Naïve Bayes	0.49	0.50	0.49	0.47
SVM (Linear)	0.71	0.71	0.71	0.71
KNN (6 Neighbors)	0.64	0.65	0.64	0.63
Decision Tree (max depth-13)	0.57	0.57	0.57	0.57
Random Forest (max depth-18)	0.68	0.68	0.68	0.68

Table. 83. Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Verloop BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.75	0.75	0.75
Gaussian Naïve Bayes	0.57	0.59	0.57	0.55
Bernoulli Naïve Bayes	0.57	0.59	0.57	0.56
Multinomial Naïve Bayes	0.53	0.53	0.53	0.52
SVM (Poly)	0.74	0.74	0.74	0.74
KNN (5 Neighbors)	0.66	0.67	0.66	0.64
Decision Tree (max depth-20)	0.56	0.55	0.56	0.55
Random Forest (max depth-16)	0.69	0.70	0.69	0.68

Table. 84. Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

GPT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.58	0.60	0.58	0.58
Bernoulli Naïve Bayes	0.56	0.57	0.56	0.56
Multinomial Naïve Bayes	0.56	0.57	0.56	0.55
SVM (RBF)	0.72	0.73	0.72	0.72
KNN (8 Neighbors)	0.63	0.61	0.63	0.61
Decision Tree (max depth-12)	0.50	0.50	0.50	0.50
Random Forest (max depth-17)	0.68	0.69	0.68	0.68

Table. 85. GPT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

XLM Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.61	0.62	0.61	0.61
Bernoulli Naïve Bayes	0.58	0.59	0.58	0.58
Multinomial	0.56	0.58	0.56	0.56

Naïve Bayes				
SVM (RBF)	0.73	0.74	0.73	0.73
KNN (7 Neighbors)	0.68	0.67	0.68	0.67
Decision Tree (max depth-17)	0.59	0.59	0.59	0.59
Random Forest (max depth-17)	0.70	0.71	0.70	0.71

Table. 86. XLM Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned BERT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.67	0.67	0.67	0.67
Gaussian Naïve Bayes	0.42	0.46	0.42	0.40
Bernoulli Naïve Bayes	0.43	0.44	0.43	0.41
Multinomial Naïve Bayes	0.42	0.42	0.42	0.40
SVM (RBF)	0.66	0.67	0.66	0.66
KNN (8 Neighbors)	0.54	0.54	0.54	0.52
Decision Tree (max depth-20)	0.42	0.42	0.42	0.42
Random Forest (max depth-14)	0.56	0.57	0.56	0.56

Table. 87. Fine-Tuned BERT Base Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned BERT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.66	0.65	0.66	0.65
Gaussian Naïve Bayes	0.45	0.49	0.45	0.44
Bernoulli Naïve Bayes	0.43	0.47	0.43	0.41
Multinomial Naïve Bayes	0.42	0.46	0.42	0.41
SVM (Poly)	0.66	0.66	0.66	0.66
KNN (7 Neighbors)	0.57	0.56	0.57	0.55
Decision Tree (max depth-8)	0.48	0.51	0.48	0.49
Random Forest (max depth-15)	0.61	0.61	0.61	0.61

Table. 88. Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned GPT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.73	0.73	0.73
Gaussian Naïve Bayes	0.52	0.55	0.52	0.51
Bernoulli Naïve Bayes	0.50	0.51	0.50	0.49
Multinomial Naïve Bayes	0.49	0.48	0.49	0.48
SVM (Linear)	0.69	0.69	0.69	0.69
KNN (7 Neighbors)	0.46	0.46	0.46	0.45
Decision Tree (max depth-11)	0.49	0.49	0.49	0.49
Random Forest (max depth-15)	0.65	0.66	0.65	0.65

Table. 89. Fine-Tuned GPT Base Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned GPT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.74	0.75	0.74	0.74
Gaussian Naïve Bayes	0.51	0.52	0.51	0.50
Bernoulli Naïve Bayes	0.51	0.53	0.51	0.51
Multinomial Naïve Bayes	0.50	0.52	0.50	0.49
SVM (Linear)	0.69	0.69	0.69	0.69
KNN (7 Neighbors)	0.46	0.46	0.46	0.45
Decision Tree (max depth-6)	0.48	0.54	0.48	0.49
Random Forest (max depth-16)	0.66	0.67	0.66	0.66

Table. 90. Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned XLM Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.50	0.50	0.50	0.50
Gaussian Naïve Bayes	0.40	0.42	0.40	0.39
Bernoulli Naïve Bayes	0.39	0.41	0.39	0.38
Multinomial Naïve Bayes	0.38	0.38	0.38	0.37
SVM (RBF)	0.52	0.54	0.52	0.52

KNN (7 Neighbors)	0.43	0.45	0.43	0.42
Decision Tree (max depth-12)	0.35	0.37	0.35	0.36
Random Forest (max depth-11)	0.46	0.49	0.46	0.46

Table. 91. Fine-Tuned XLM Base Vectorized Models and Metrics of Nisha's Dataset.

4.2.3 Scaling Models

a. Min-Max Scaling

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.71	0.71	0.71	0.71
Gaussian Naïve Bayes	0.52	0.50	0.52	0.48
Bernoulli Naïve Bayes	0.70	0.70	0.70	0.69
Multinomial Naïve Bayes	0.67	0.67	0.67	0.67
SVM (Sigmoid)	0.70	0.71	0.70	0.70
KNN (4 Neighbors)	0.55	0.55	0.55	0.53
Decision Tree	0.64	0.65	0.64	0.64
Random Forest	0.71	0.71	0.71	0.71

Table. 92. Min-Max Scaled TF-IDF Vectorized Models and Metrics of Nisha's Dataset.

Count Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.69	0.69	0.69	0.69
Gaussian Naïve Bayes	0.51	0.49	0.51	0.47
Bernoulli Naïve Bayes	0.69	0.70	0.69	0.68
Multinomial Naïve Bayes	0.66	0.66	0.66	0.65
SVM (Sigmoid)	0.71	0.72	0.71	0.71
KNN (5 Neighbors)	0.57	0.57	0.57	0.55
Decision Tree	0.61	0.63	0.61	0.61
Random Forest	0.64	0.65	0.64	0.64

Table. 93. Min-Max Scaled Count Vectorized Models and Metrics of Nisha's Dataset.

TF (Term Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.70	0.70	0.70	0.70
Gaussian Naïve Bayes	0.49	0.48	0.49	0.46
Bernoulli Naïve Bayes	0.69	0.70	0.69	0.68
Multinomial Naïve Bayes	0.69	0.69	0.69	0.69
SVM (RBF)	0.71	0.72	0.71	0.71
KNN (5 Neighbors)	0.58	0.58	0.58	0.56
Decision Tree	0.65	0.65	0.65	0.65
Random Forest	0.71	0.71	0.71	0.71

Table. 94. Min-Max Scaled TF Vectorized Models and Metrics of Nisha's Dataset.

BERT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.65	0.72	0.65	0.64
Gaussian Naïve Bayes	0.58	0.60	0.58	0.58
Bernoulli Naïve Bayes	0.22	0.29	0.22	0.17
Multinomial Naïve Bayes	0.54	0.57	0.54	0.53
SVM (RBF)	0.76	0.77	0.76	0.76
KNN (6 Neighbors)	0.67	0.66	0.67	0.66
Decision Tree (max depth-6)	0.55	0.59	0.55	0.56
Random Forest (max depth-16)	0.70	0.71	0.70	0.70

Table. 95. Min-Max Scaled BERT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Ganesh BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.30	0.61	0.30	0.26
Gaussian Naïve Bayes	0.28	0.26	0,28	0.17
Bernoulli Naïve Bayes	0.19	0.27	0.19	0.14
Multinomial Naïve Bayes	0.30	0.26	0.30	0.22
SVM (Poly)	0.48	0.53	0.48	0.48
KNN (7 Neighbors)	0.44	0.43	0.44	0.42

Decision Tree (max depth-7)	0.35	0.38	0.35	0.36
Random Forest (max depth-14)	0.43	0.44	0.43	0.40

Table. 96. Min-Max Scaled Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.66	0.70	0.66	0.66
Gaussian Naïve Bayes	0.54	0.55	0.54	0.54
Bernoulli Naïve Bayes	0.21	0.34	0.21	0.18
Multinomial Naïve Bayes	0.50	0.50	0.50	0.49
SVM (RBF)	0.72	0.73	0.72	0.72
KNN (8 Neighbors)	0.63	0.63	0.63	0.62
Decision Tree (max depth-7)	0.44	0.50	0.44	0.44
Random Forest (max depth-20)	0.68	0.69	0.68	0.68

Table. 97. Min-Max Scaled Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Verloop BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.64	0.66	0.64	0.60
Gaussian Naïve Bayes	0.54	0.64	0.54	0.54
Bernoulli Naïve Bayes	0.18	0.34	0.18	0.13
Multinomial Naïve Bayes	0.53	0.54	0.53	0.51
SVM (RBF)	0.73	0.76	0.73	0.73
KNN (8 Neighbors)	0.66	0.68	0.66	0.64
Decision Tree (max depth-6)	0.41	0.50	0.41	0.42
Random Forest (max depth-9)	0.68	0.71	0.68	0.68

Table. 98. Min-Max Scaled Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

GPT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.69	0.70	0.69	0.67
Gaussian Naïve Bayes	0.56	0.62	0.56	0.58
Bernoulli Naïve Bayes	0.19	0.35	0.19	0.16
Multinomial Naïve Bayes	0.57	0.58	0.57	0.57
SVM (RBF)	0.73	0.77	0.73	0.74
KNN (8 Neighbors)	0.63	0.62	0.63	0.61
Decision Tree (max depth-6)	0.45	0.50	0.45	0.44
Random Forest (max depth-20)	0.69	0.70	0.69	0.69

Table. 99. Min-Max Scaled GPT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

XLM Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.60	0.76	0.60	0.60
Gaussian Naïve Bayes	0.62	0.65	0.62	0.63
Bernoulli Naïve Bayes	0.22	0.40	0.22	0.20
Multinomial Naïve Bayes	0.59	0.60	0.59	0.58
SVM(RBF)	0.76	0.79	0.76	0.77
KNN (8 Neighbors)	0.67	0.66	0.67	0.66
Decision Tree (max depth-7)	0.55	0.57	0.55	0.55
Random Forest (max depth-11)	0.72	0.73	0.72	0.72

Table. 100. Min-Max Scaled XLM Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned BERT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.54	0.68	0.54	0.54
Gaussian Naïve Bayes	0.33	0.47	0.33	0.28
Bernoulli Naïve Bayes	0.21	0.35	0.21	0.16
Multinomial	0.43	0.46	0.43	0.41

Naïve Bayes				
SVM (RBF)	0.68	0.69	0.68	0.68
KNN (7 Neighbors)	0.55	0.56	0.55	0.54
Decision Tree (max depth-10)	0.34	0.36	0.34	0.34
Random Forest (max depth-19)	0.58	0.58	0.58	0.58

Table. 101. Min-Max Scaled Fine-Tuned BERT Base Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned BERT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.55	0.56	0.55	0.51
Gaussian Naïve Bayes	0.45	0.52	0.45	0.44
Bernoulli Naïve Bayes	0.21	0.26	0.21	0.16
Multinomial Naïve Bayes	0.43	0.48	0.43	0.41
SVM (RBF)	0.67	0.67	0.67	0.67
KNN (6 Neighbors)	0.57	0.56	0.57	0.55
Decision Tree (max depth-10)	0.42	0.44	0.42	0.43
Random Forest (max depth-16)	0.62	0.63	0.62	0.62

Table. 102. Min-Max Scaled Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned GPT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.57	0.70	0.57	0.58
Gaussian Naïve Bayes	0.52	0.56	0.52	0.52
Bernoulli Naïve Bayes	0.20	0.30	0.20	0.15
Multinomial Naïve Bayes	0.52	0.52	0.52	0.50
SVM (RBF)	0.73	0.74	0.73	0.73
KNN (8 Neighbors)	0.61	0.61	0.61	0.60
Decision Tree (max depth-6)	0.44	0.47	0.44	0.45
Random Forest (max depth-11)	0.65	0.66	0.65	0.65

Table. 103. Min-Max Scaled Fine-Tuned GPT Base Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned GPT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.73	0.72	0.72
Gaussian Naïve Bayes	0.54	0.55	0.54	0.55
Bernoulli Naïve Bayes	0.25	0.39	0.25	0.23
Multinomial Naïve Bayes	0.53	0.54	0.53	0.52
SVM (RBF)	0.75	0.76	0.75	0.75
KNN (7 Neighbors)	0.60	0.61	0.60	0.59
Decision Tree (max depth-7)	0.44	0.52	0.44	0.45
Random Forest (max depth-14)	0.67	0.68	0.67	0.67

Table. 104. Min-Max Scaled Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned XLM Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.48	0.51	0.48	0.48
Gaussian Naïve Bayes	0.39	0.41	0.39	0.38
Bernoulli Naïve Bayes	0.27	0.35	0.27	0.27
Multinomial Naïve Bayes	0.39	0.39	0.39	0.38
SVM (RBF)	0.52	0.54	0.52	0.53
KNN (8 Neighbors)	0.45	0.45	0.45	0.44
Decision Tree (max depth-14)	0.34	0.35	0.34	0.34
Random Forest (max depth-13)	0.47	0.47	0.47	0.47

Table. 105. Min-Max Scaled Fine-Tuned XLM Base Vectorized Models and Metrics of Nisha's Dataset.

b. Normalized Scaling

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.74	0.73	0.73
Gaussian Naïve Bayes	0.53	0.50	0.53	0.49
Bernoulli Naïve Bayes	0.70	0.70	0.70	0.69
Multinomial Naïve Bayes	0.67	0.67	0.67	0.67
SVM (Linear)	0.74	0.74	0.74	0.74
KNN (4 Neighbors)	0.52	0.56	0.52	0.50
Decision Tree	0.65	0.65	0.65	0.65
Random Forest	0.71	0.71	0.71	0.71

Table. 106. Normalize Scaled TF-IDF Vectorized Models and Metrics of Nisha's Dataset.

Count Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.50	0.48	0.50	0.46
Bernoulli Naïve Bayes	0.69	0.70	0.69	0.68
Multinomial Naïve Bayes	0.69	0.69	0.69	0.69
SVM (Linear)	0.72	0.73	0.72	0.72
KNN (5 Neighbors)	0.56	0.60	0.56	0.54
Decision Tree	0.65	0.65	0.65	0.65
Random Forest	0.71	0.71	0.71	0.71

Table. 107. Normalize Scaled Count Vectorized Models and Metrics of Nisha's Dataset.

TF (Term Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.50	0.48	0.50	0.46
Bernoulli Naïve Bayes	0.69	0.70	0.69	0.68
Multinomial Naïve Bayes	0.69	0.69	0.69	0.68
SVM (Linear)	0.72	0.73	0.72	0.72
KNN (5 Neighbors)	0.54	0.58	0.54	0.52
Decision Tree	0.65	0.65	0.65	0.65

Random Forest	0.71	0.71	0.71	0.71

Table. 108. Normalize Scaled TF Vectorized Models and Metrics of Nisha's Dataset.

BERT Base model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.71	0.71	0.71	0.71
Gaussian Naïve Bayes	0.56	0.58	0.56	0.55
Bernoulli Naïve Bayes	0.55	0.58	0.55	0.54
Multinomial Naïve Bayes	0.53	0.55	0.53	0.52
SVM (Poly)	0.75	0.76	0.75	0.75
KNN (8 Neighbors)	0.66	0.65	0.66	0.64
Decision Tree (max depth-8)	0.59	0.60	0.59	0.59
Random Forest (max depth-19)	0.72	0.73	0.72	0.72

Table. 109. Normalize Scaled BERT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Ganesh BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.35	0.34	0.35	0.26
Gaussian Naïve Bayes	0.28	0.18	0.28	0.17
Bernoulli Naïve Bayes	0.30	0.21	0.30	0.22
Multinomial Naïve Bayes	0.31	0.28	0.31	0.23
SVM (Poly)	0.35	0.32	0.35	0.25
KNN (8 Neighbors)	0.47	0.46	0.47	0.46
Decision Tree (max depth-7)	0.49	0.50	0.49	0.48
Random Forest (max depth-15)	0.56	0.57	0.56	0.56

Table. 110. Normalize Scaled Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.68	0.68	0.68	0.68
Gaussian Naïve Bayes	0.53	0.55	0.53	0.52
Bernoulli Naïve Bayes	0.53	0.55	0.53	0.52
Multinomial Naïve Bayes	0.51	0.51	0.51	0.50
SVM (Poly)	0.72	0.73	0.72	0.72
KNN (7 Neighbors)	0.62	0.63	0.62	0.61
Decision Tree (max depth-10)	0.55	0.55	0.55	0.55
Random Forest (max depth-19)	0.68	0.69	0.68	0.68

Table. 111. Normalize Scaled Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Verloop BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.74	0.74	0.74	0.74
Gaussian Naïve Bayes	0.58	0.60	0.58	0.57
Bernoulli Naïve Bayes	0.58	0.59	0.58	0.56
Multinomial Naïve Bayes	0.55	0.56	0.55	0.54
SVM (Poly)	0.77	0.78	0.77	0.77
KNN (8 Neighbors)	0.66	0.68	0.66	0.65
Decision Tree (max depth-10)	0.54	0.55	0.54	0.54
Random Forest (max depth-12)	0.71	0.71	0.71	0.71

Table. 112. Normalize Scaled Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

GPT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.61	0.63	0.61	0.61
Bernoulli Naïve Bayes	0.58	0.59	0.58	0.58
Multinomial	0.58	0.58	0.58	0.57

Naïve Bayes				
SVM (Poly)	0.75	0.75	0.75	0.75
KNN (8 Neighbors)	0.64	0.63	0.64	0.62
Decision Tree (max depth-8)	0.50	0.54	0.50	0.51
Random Forest (max depth-10)	0.70	0.70	0.70	0.70

Table. 113. Normalize Scaled GPT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

XLM Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.75	0.75	0.75
Gaussian Naïve Bayes	0.63	0.64	0.63	0.63
Bernoulli Naïve Bayes	0.61	0.62	0.61	0.61
Multinomial Naïve Bayes	0.60	0.61	0.60	0.59
SVM (Poly)	0.78	0.78	0.78	0.78
KNN (7 Neighbors)	0.68	0.67	0.68	0.67
Decision Tree (max depth-9)	0.58	0.59	0.58	0.58
Random Forest (max depth-18)	0.73	0.74	0.73	0.73

Table. 114. Normalize Scaled XLM Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned BERT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.65	0.65	0.65	0.64
Gaussian Naïve Bayes	0.46	0.47	0.46	0.44
Bernoulli Naïve Bayes	0.46	0.48	0.46	0.44
Multinomial Naïve Bayes	0.47	0.47	0.47	0.45
SVM (Poly)	0.68	0.69	0.68	0.68
KNN (8 Neighbors)	0.56	0.56	0.56	0.55
Decision Tree (max depth-14)	0.43	0.43	0.43	0.43
Random Forest (max depth-20)	0.59	0.59	0.59	0.58

Table. 115. Normalize Scaled Fine-Tuned BERT Base Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned BERT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.64	0.64	0.64	0.63
Gaussian Naïve Bayes	0.44	0.51	0.44	0.42
Bernoulli Naïve Bayes	0.44	0.52	0.44	0.43
Multinomial Naïve Bayes	0.43	0.49	0.43	0.42
SVM (Poly)	0.67	0.68	0.67	0.67
KNN (7 Neighbors)	0.59	0.58	0.59	0.57
Decision Tree (max depth-10)	0.49	0.49	0.49	0.49
Random Forest (max depth-14)	0.63	0.63	0.63	0.62

Table. 116. Normalize Scaled Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned GPT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.37	0.37	0.37	0.34
Gaussian Naïve Bayes	0.50	0.52	0.50	0.48
Bernoulli Naïve Bayes	0.53	0.53	0.53	0.52
Multinomial Naïve Bayes	0.51	0.52	0.51	0.51
SVM (Poly)	0.39	0.41	0.39	0.37
KNN (4 Neighbors)	0.50	0.51	0.50	0.50
Decision Tree (max depth-15)	0.48	0.48	0.48	0.48
Random Forest (max depth-17)	0.67	0.67	0.67	0.67

Table. 117. Normalize Scaled Fine-Tuned GPT Base Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned GPT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.34	0.38	0.34	0.30
Gaussian Naïve Bayes	0.54	0.56	0.54	0.54
Bernoulli Naïve Bayes	0.54	0.54	0.54	0.53
Multinomial	.54	0.55	0.54	0.53

Naïve Bayes				
SVM (Poly)	0.27	0.43	0.27	0.21
KNN (8 Neighbors)	0.53	0.52	0.53	0.51
Decision Tree (max depth-9)	0.52	0.54	0.52	0.53
Random Forest (max depth-14)	0.68	0.69	0.68	0.68

Table. 118. Normalize Scaled Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned XLM Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.53	0.52	0.53	0.52
Gaussian Naïve Bayes	0.41	0.41	0.41	0.40
Bernoulli Naïve Bayes	0.40	0.41	0.40	0.39
Multinomial Naïve Bayes	0.39	0.39	0.39	0.38
SVM (Poly)	0.54	0.55	0.54	0.54
KNN (8 Neighbors)	0.46	0.46	0.46	0.46
Decision Tree (max depth-12)	0.37	0.37	0.37	0.37
Random Forest (max depth-18)	0.47	0.48	0.47	0.47

Table. 119. Normalize Scaled Fine-Tuned XLM Base Vectorized Models and Metrics of Nisha's Dataset.

c. Standard Scaling

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.67	0.67	0.67	0.67
Gaussian Naïve Bayes	0.22	0.27	0.22	0.17
Bernoulli Naïve Bayes	0.70	0.70	0.70	0.69
Multinomial Naïve Bayes	0.67	0.67	0.67	0.67
SVM (Linear)	0.67	0.67	0.67	0.67
KNN (5 Neighbors)	0.52	0.53	0.52	0.52
Decision Tree	0.64	0.64	0.64	0.64
Random Forest	0.70	0.71	0.70	0.70

Table. 120. Standard Scaled TF-IDF Vectorized Models and Metrics of Nisha's Dataset.

Count Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.68	0.69	0.68	0.68
Gaussian Naïve Bayes	0.22	0.26	0.22	0.16
Bernoulli Naïve Bayes	0.69	0.70	0.69	0.68
Multinomial Naïve Bayes	0.66	0.66	0.66	0.65
SVM (Linear)	0.69	0.69	0.69	0.69
KNN (7 Neighbors)	0.55	0.56	0.55	0.53
Decision Tree	0.63	0.64	0.63	0.63
Random Forest	0.68	0.69	0.68	0.68

Table. 121. Standard Scaled Count Vectorized Models and Metrics of Nisha's Dataset.

TF (Term Frequency) Vectorized Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.67	0.67	0.67	0.67
Gaussian Naïve Bayes	0.22	0.26	0.22	0.17
Bernoulli Naïve Bayes	0.69	0.70	0.69	0.68
Multinomial Naïve Bayes	0.69	0.69	0.69	0.68
SVM (Sigmoid)	0.69	0.69	0.69	0.69
KNN (8 Neighbors)	0.54	0.55	0.54	0.53
Decision Tree	0.65	0.65	0.65	0.65
Random Forest	0.71	0.71	0.71	0.71

Table. 122. Standard Scaled TF Vectorized Models and Metrics of Nisha's Dataset.

BERT Base model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.74	0.73	0.74	0.74
Gaussian Naïve Bayes	0.57	0.59	0.57	0.56
Bernoulli Naïve Bayes	0.56	0.58	0.56	0.55
Multinomial Naïve Bayes	0.54	0.57	0.54	0.53
SVM (RBF)	0.76	0.76	0.76	0.76
KNN (8 Neighbors)	0.66	0.66	0.66	0.65
Decision Tree (max depth-8)	0.59	0.60	0.59	0.59

Random Forest (max depth-12)	0.71	0.71	0.71	0.71

Table. 123. Standard Scaled BERT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Ganesh BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.62	0.61	0.62	0.61
Gaussian Naïve Bayes	0.31	0.23	0.31	0.22
Bernoulli Naïve Bayes	0.30	0.22	0.30	0.21
Multinomial Naïve Bayes	0.30	0.26	0.30	0.22
SVM (Linear)	0.60	0.60	0.60	0.59
KNN (7 Neighbors)	0.47	0.46	0.47	0.46
Decision Tree (max depth-16)	0.36	0.39	0.36	0.36
Random Forest (max depth-16)	0.53	0.53	0.53	0.52

Table. 124. Standard Scaled Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.52	0.53	0.52	0.51
Bernoulli Naïve Bayes	0.50	0.50	0.50	0.49
Multinomial Naïve Bayes	0.50	0.50	0.50	0.49
SVM (RBF)	0.73	0.73	0.73	0.73
KNN (6 Neighbors)	0.63	0.63	0.63	0.62
Decision Tree (max depth-11)	0.54	0.55	0.54	0.54
Random Forest (max depth-20)	0.69	0.70	0.69	0.69

Table. 125. Standard Scaled Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Verloop BERT Hinglish Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.72	0.72	0.72	0.72
Gaussian Naïve Bayes	0.57	0.59	0.57	0.56
Bernoulli Naïve Bayes	0.54	0.55	0.54	0.53
Multinomial Naïve Bayes	0.53	0.54	0.53	0.51
SVM (RBF)	0.77	0.78	0.77	0.77
KNN (8 Neighbors)	0.66	0.66	0.66	0.64
Decision Tree (max depth-8)	0.52	0.54	0.52	0.52
Random Forest (max depth-18)	0.71	0.71	0.71	0.71

Table. 126. Standard Scaled Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

GPT Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.72	0.73	0.72
Gaussian Naïve Bayes	0.59	0.61	0.59	0.59
Bernoulli Naïve Bayes	0.57	0.58	0.57	0.57
Multinomial Naïve Bayes	0.57	0.58	0.57	0.57
SVM (RBF)	0.75	0.75	0.75	0.75
KNN (6 Neighbors)	0.63	0.62	0.63	0.61
Decision Tree (max depth-13)	0.51	0.52	0.51	0.52
Random Forest (max depth-15)	0.70	0.70	0.70	0.69

Table. 127. Standard Scaled GPT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

XLM Base Model (Sentence Transformer)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.75	0.75	0.75	0.75
Gaussian Naïve Bayes	0.63	0.64	0.63	0.63
Bernoulli Naïve Bayes	0.61	0.62	0.61	0.61
Multinomial Naïve Bayes	0.59	0.60	0.59	0.58
SVM (RBF)	0.78	0.78	0.78	0.78
KNN (8 Neighbors)	0.66	0.65	0.66	0.65
Decision Tree (max depth-7)	0.58	0.58	0.58	0.58
Random Forest (max depth-16)	0.73	0.74	0.73	0.73

Table. 128. Standard Scaled XLM Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned BERT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.63	0.63	0.63	0.63
Gaussian Naïve Bayes	0.44	0.45	0.44	0.41
Bernoulli Naïve Bayes	0.45	0.46	0.45	0.43
Multinomial Naïve Bayes	0.43	0.46	0.43	0.41
SVM (RBF)	0.68	0.68	0.68	0.68
KNN (7 Neighbors)	0.54	0.54	0.54	0.52
Decision Tree (max depth-18)	0.40	0.40	0.40	0.40
Random Forest (max depth-19)	0.60	0.61	0.60	0.60

Table. 129. Standard Scaled Fine-Tuned BERT Base Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned BERT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.63	0.63	0.63	0.63
Gaussian Naïve Bayes	0.47	0.52	0.47	0.46
Bernoulli Naïve Bayes	0.44	0.50	0.44	0.43
Multinomial Naïve Bayes	0.43	0.48	0.43	0.41
SVM (RBF)	0.68	0.68	0.68	0.68
KNN (8 Neighbors)	0.58	0.58	0.58	0.57
Decision Tree (max depth-9)	0.46	0.46	0.46	0.45
Random Forest (max depth-13)	0.62	0.62	0.62	0.62

Table. 130. Standard Scaled Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned GPT Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.71	0.71	0.71	0.71
Gaussian Naïve Bayes	0.54	0.56	0.54	0.53
Bernoulli Naïve Bayes	0.52	0.53	0.52	0.51
Multinomial Naïve Bayes	0.52	0.52	0.52	0.50
SVM (RBF)	0.75	0.75	0.75	0.75
KNN (7 Neighbors)	0.61	0.61	0.61	0.59
Decision Tree (max depth-9)	0.44	0.46	0.44	0.45
Random Forest (max depth-15)	0.69	0.69	0.69	0.68

Table. 131. Standard Scaled Fine-Tuned GPT Base Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned GPT Hinglish Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.72	0.73	0.72
Gaussian Naïve Bayes	0.53	0.54	0.53	0.52
Bernoulli Naïve Bayes	0.51	0.52	0.51	0.50
Multinomial	0.53	0.54	0.53	0.52

Naïve Bayes				
SVM (RBF)	0.76	0.76	0.76	0.76
KNN (7 Neighbors)	0.60	0.61	0.60	0.59
Decision Tree (max depth-8)	0.49	0.48	0.49	0.48
Random Forest (max depth-11)	0.68	0.68	0.68	0.67

Table. 132. Standard Scaled Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Nisha's Dataset.

Fine-Tuned XLM Base Model

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.50	0.50	0.50	0.50
Gaussian Naïve Bayes	0.41	0.41	0.41	0.40
Bernoulli Naïve Bayes	0.39	0.40	0.39	0.38
Multinomial Naïve Bayes	0.39	0.39	0.39	0.38
SVM (RBF)	0.54	0.54	0.54	0.54
KNN (8 Neighbors)	0.45	0.45	0.45	0.44
Decision Tree (max depth-20)	0.34	0.34	0.34	0.34
Random Forest (max depth-17)	0.47	0.47	0.47	0.46

Table. 133. Standard Scaled Fine-Tuned XLM Base Vectorized Models and Metrics of Nisha's Dataset.

4.2.4 Principal Component and Independent Component Analysis Models

TF-IDF (Term Frequency – Inverse Document Frequency) Vectorized Model

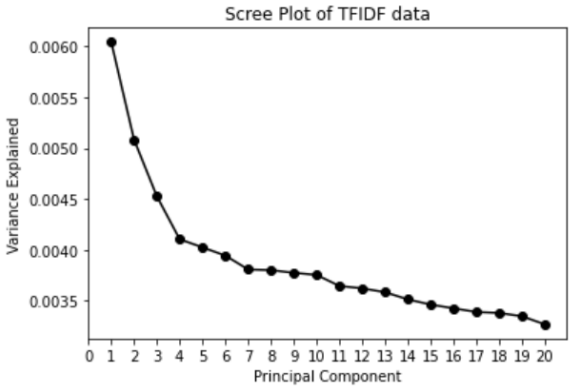

Figure. 54. Scree Plot for TF-IDF Vectors of Nisha's Dataset.

According to the Scree plot of Figure 44, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.37	0.31	0.37	0.30
Gaussian Naïve Bayes	0.37	0.38	0.37	0.33
Bernoulli Naïve Bayes	0.34	0.23	0.34	0.25
Multinomial Naïve Bayes	0.32	0.34	0.32	0.32
SVM (RBF)	0.49	0.48	0.49	0.48
KNN (7 Neighbors)	0.52	0.51	0.52	0.51
Decision Tree (max depth-20)	0.51	0.51	0.51	0.51
Random Forest (max depth-12)	0.57	0.57	0.57	0.57

Table. 134. TF-IDF Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Count Vectorized Model

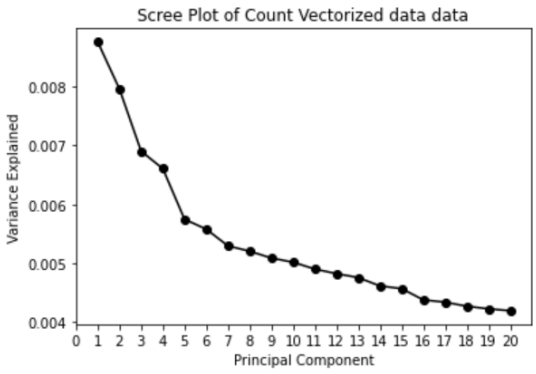

Figure. 55. Scree Plot for Count Vectors of Nisha's Dataset.

According to the Scree plot of Figure 45, 7 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.39	0.34	0.39	0.29
Gaussian Naïve Bayes	0.32	0.35	0.32	0.29
Bernoulli Naïve Bayes	0.37	0.34		0.34
Multinomial Naïve Bayes	0.17	0.09	0.17	0.08
SVM (RBF)	0.47	0.46	0.47	0.45
KNN (8 Neighbors)	0.56	0.55	0.56	0.55
Decision Tree (max depth-11)	0.55	0.55	0.55	0.55
Random Forest (max depth-10)	0.61	0.61	0.61	0.60

Table. 135. Count Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

TF (Term Frequency) Vectorized Model

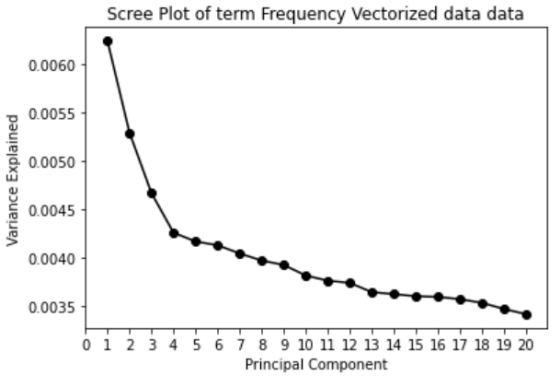

Figure. 56. Scree Plot for TF Vectors of Nisha's Dataset.

According to the Scree plot of Figure 46, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.38	0.31	0.38	0.30
Gaussian Naïve Bayes	0.35	0.37	0.35	0.31
Bernoulli Naïve Bayes	0.37	0.31	0.37	0.32
Multinomial Naïve Bayes	0.20	0.24	0.20	0.16
SVM (RBF)	0.49	0.48	0.49	0.48
KNN (8 Neighbors)	0.52	0.51	0.52	0.51
Decision Tree (max depth-9)	0.51	0.51	0.51	0.51
Random Forest (max depth-15)	0.57	0.57	0.57	0.57

Table. 136. TF Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

BERT Base model (Sentence Transformer)

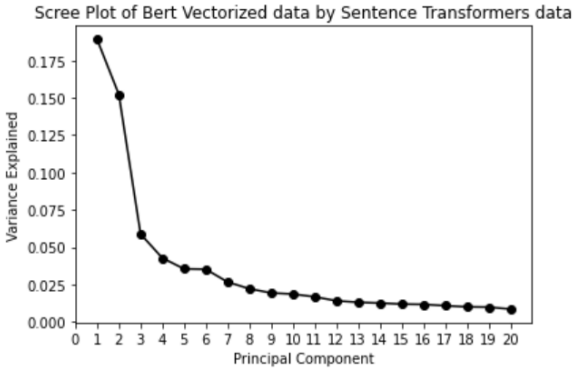

Figure. 57. Scree Plot for BERT Base (Sentence Transformer) Vectors of Nisha's Dataset.

According to the Scree plot of Figure 47, 3 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.45	0.40	0.45	0.39
Gaussian Naïve Bayes	0.48	0.44	0.48	0.43
Bernoulli Naïve Bayes	0.44	0.39	0.44	0.40
Multinomial Naïve Bayes	0.44	0.42	0.44	0.41
SVM (RBF)	0.53	0.50	0.53	0.50
KNN (7 Neighbors)	0.53	0.51	0.53	0.51
Decision Tree (max depth-8)	0.52	0.53	0.52	0.52
Random Forest (max depth-8)	0.56	0.56	0.56	0.55

Table. 137. BERT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Ganesh BERT Hinglish Model (Sentence Transformer)

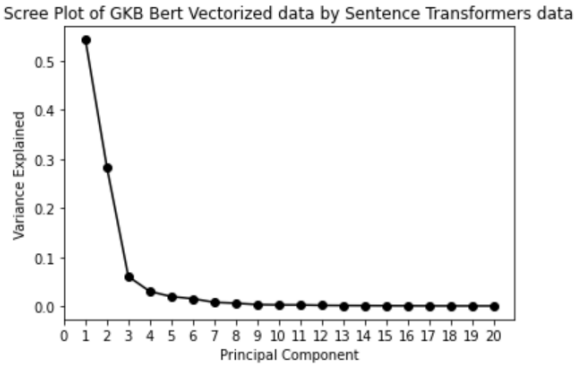

Figure. 58. Scree Plot for Ganesh BERT Hinglish (Sentence Transformer) Vectors of Nisha's Dataset.

According to the Scree plot of Figure 48, 3 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.32	0.17	0.32	0.20
Gaussian Naïve Bayes	0.33	0.34	0.33	0.23
Bernoulli Naïve Bayes	0.33	0.14	0.33	0.20
Multinomial Naïve Bayes	0.29	0.19	0.29	0.20
SVM (RBF)	0.34	0.24	0.34	0.24
KNN (4 Neighbors)	0.39	0.38	0.39	0.38
Decision Tree (max depth-8)	0.40	0.41	0.40	0.39
Random Forest (max depth-11)	0.43	0.44	0.43	0.43

Table. 138. Ganesh BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Narasimha Distil BERT Hinglish Model (Sentence Transformer)

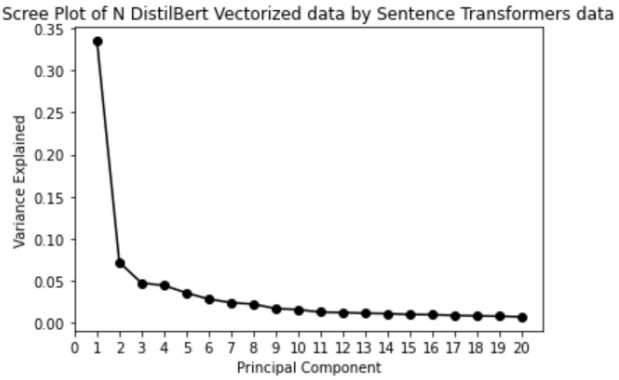

Figure. 59. Scree Plot for Narasimha Distil BERT Hinglish (Sentence Transformer) Vectors of Nisha's Dataset.

According to the Scree plot of Figure 49, 2 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.33	0.37	0.33	0.26
Gaussian Naïve Bayes	0.35	0.33	0.35	0.32
Bernoulli Naïve Bayes	0.33	0.19	0.33	0.24
Multinomial Naïve Bayes	0.27	0.22	0.27	0.22
SVM (RBF)	0.39	0.37	0.39	0.36
KNN (8 Neighbors)	0.40	0.39	0.40	0.39
Decision Tree (max depth-9)	0.43	0.45	0.43	0.43

Random Forest (max depth-10)	0.45	0.45	0.45	0.44

Table. 139. Narasimha Distil BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Verloop BERT Hinglish Model (Sentence Transformer)

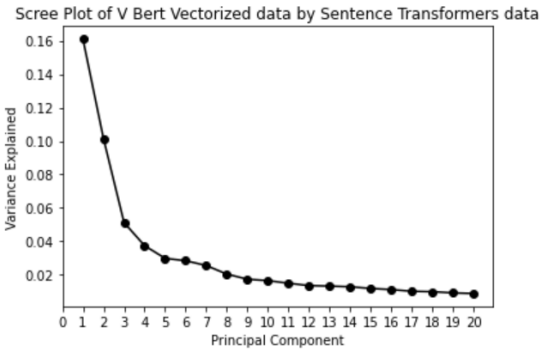

Figure. 60. Scree Plot for Verloop BERT Hinglish (Sentence Transformer) Vectors of Nisha's Dataset.

According to the Scree plot of Figure 50, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.45	0.38	0.45	0.37
Gaussian Naïve Bayes	0.52	0.50	0.52	0.49
Bernoulli Naïve Bayes	0.39	0.34	0.39	0.34
Multinomial Naïve Bayes	0.34	0.36	0.34	0.30
SVM (RBF)	0.58	0.56	0.58	0.56
KNN (8 Neighbors)	0.54	0.54	0.54	0.54
Decision Tree (max depth-11)	0.51	0.51	0.51	0.51
Random Forest (max depth-10)	0.57	0.57	0.57	0.57

Table. 140. Verloop BERT Hinglish (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

GPT Base Model (Sentence Transformer)

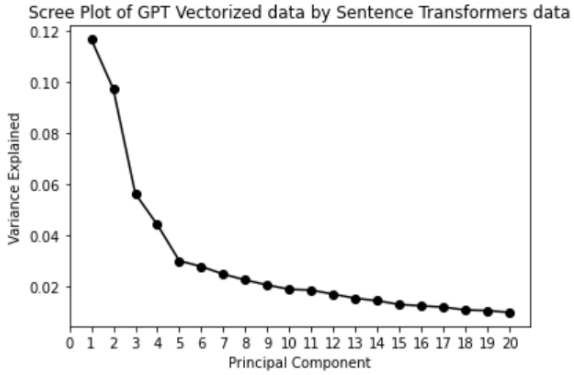

Figure. 61. Scree Plot for GPT Base (Sentence Transformer) Vectors of Nisha's Dataset.

According to the Scree plot of Figure 51, 5 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.44	0.39	0.44	0.39
Gaussian Naïve Bayes	0.47	0.45	0.47	0.44
Bernoulli Naïve Bayes	0.36	0.35	0.36	0.34
Multinomial Naïve Bayes	0.41	0.43	0.41	0.40
SVM (RBF)	0.55	0.54	0.55	0.53
KNN (6 Neighbors)	0.49	0.48	0.49	0.48
Decision Tree (max depth-7)	0.49	0.51	0.49	0.49
Random Forest (max depth-11)	0.55	0.55	0.55	0.55

Table. 141. GPT Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

XLM Base Model (Sentence Transformer)

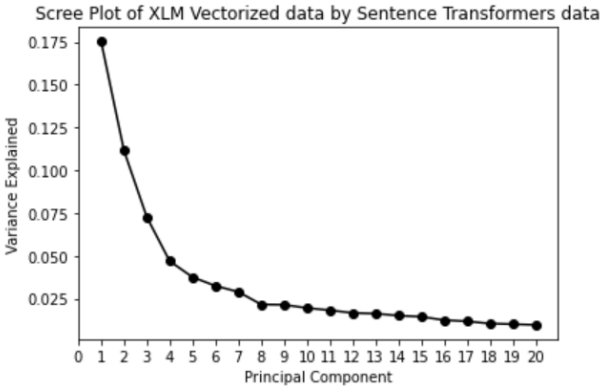

Figure. 62. Scree Plot for XLM Base (Sentence Transformer) Vectors of Nisha's Dataset.

According to the Scree plot of Figure 52, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.50	0.52	0.50	0.48
Gaussian Naïve Bayes	0.54	0.54	0.54	0.52
Bernoulli Naïve Bayes	0.47	0.51	0.47	0.46
Multinomial Naïve Bayes	0.49	0.48	0.49	0.46
SVM (RBF)	0.58	0.59	0.58	0.58
KNN (8 Neighbors)	0.55	0.54	0.55	0.54
Decision Tree (max depth-8)	0.55	0.56	0.55	0.55
Random Forest (max depth-14)	0.60	0.60	0.60	0.60

Table. 142. XLM Base (Sentence Transformer) Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Fine-Tuned BERT Base Model

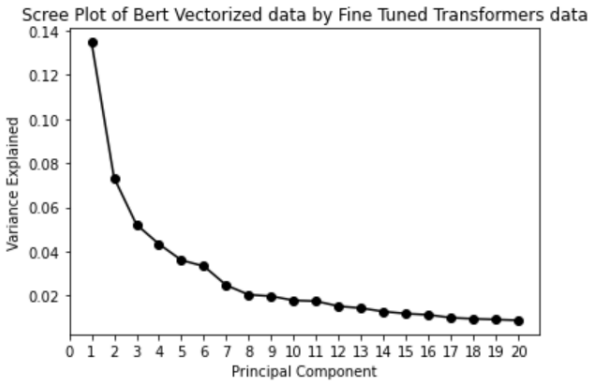

Figure. 63. Scree Plot for Fine-Tuned BERT Base Vectors of Nisha's Dataset.

According to the Scree plot of Figure 53, 7 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.36	0.34	0.36	0.29
Gaussian Naïve Bayes	0.37	0.38	0.37	0.34
Bernoulli Naïve Bayes	0.32	0.31	0.32	0.29
Multinomial Naïve Bayes	0.33	0.37	0.33	0.31
SVM (RBF)	0.44	0.47	0.44	0.41
KNN (7 Neighbors)	0.41	0.42	0.41	0.41
Decision Tree (max depth-11)	0.40	0.40	0.40	0.40

Random Forest (max depth-12)	0.47	0.48	0.47	0.46

Table. 143. Fine-Tuned BERT Base Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Fine-Tuned BERT Hinglish Model

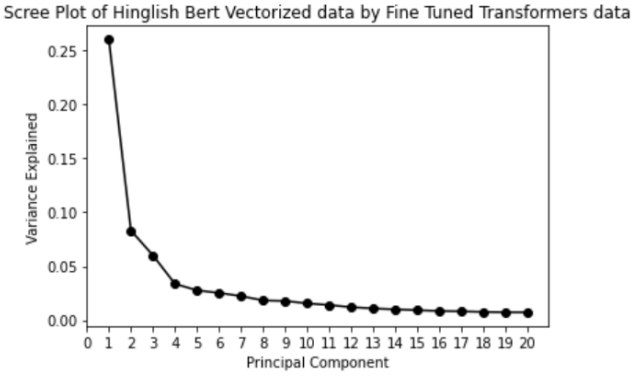

Figure. 64. Scree Plot for Fine-Tuned BERT Hinglish Vectors of Nisha's Dataset.

According to the Scree plot of Figure 54, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.42	0.44	0.42	0.34
Gaussian Naïve Bayes	0.44	0.44	0.44	0.41
Bernoulli Naïve Bayes	0.35	0.29	0.35	0.31
Multinomial Naïve Bayes	0.29	0.27	0.29	0.25
SVM (RBF)	0.49	0.47	0.49	0.47
KNN (6 Neighbors)	0.45	0.44	0.45	0.44
Decision Tree (max depth-8)	0.47	0.48	0.47	0.47
Random Forest (max depth-11)	0.53	0.52	0.53	0.51

Table. 144. Fine-Tuned BERT Hinglish Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Fine-Tuned GPT Base Model

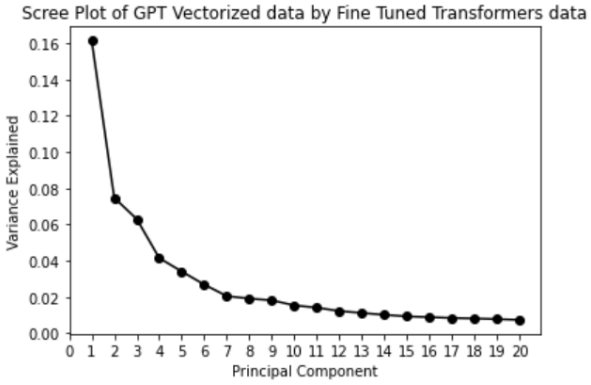

Figure. 65. Scree Plot for Fine-Tuned GPT Base Vectors of Nisha's Dataset.

According to the Scree plot of Figure 55, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.41	0.38	0.41	0.34
Gaussian Naïve Bayes	0.44	0.43	0.44	0.40
Bernoulli Naïve Bayes	0.33	0.25	0.33	0.26
Multinomial Naïve Bayes	0.40	0.39	0.40	0.37
SVM (RBF)	0.51	0.49	0.51	0.48
KNN (8 Neighbors)	0.47	0.46	0.47	0.46
Decision Tree (max depth-9)	0.46	0.46	0.46	0.45
Random Forest (max depth-11)	0.51	0.51	0.51	0.50

Table. 145. Fine-Tuned GPT Base Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Fine-Tuned GPT Hinglish Model

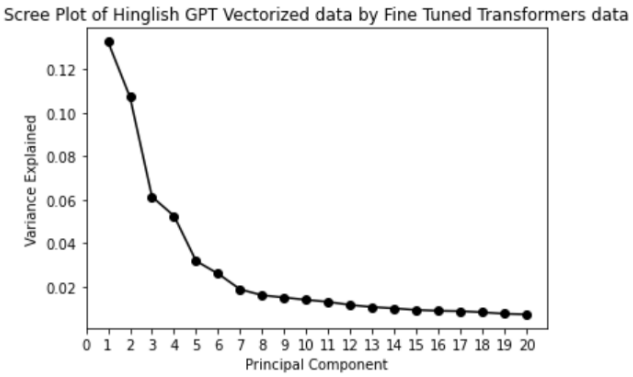

Figure. 66. Scree Plot for Fine-Tuned GPT Hinglish Vectors of Nisha's Dataset.

According to the Scree plot of Figure 56, 5 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.41	0.38	0.41	0.34
Gaussian Naïve Bayes	0.41	0.42	0.41	0.37
Bernoulli Naïve Bayes	0.36	0.33	0.36	0.33
Multinomial Naïve Bayes	0.33	0.31	0.33	0.30
SVM (RBF)	0.53	0.53	0.53	0.51
KNN (8 Neighbors)	0.49	0.48	0.49	0.49
Decision Tree (max depth-10)	0.47	0.46	0.47	0.46
Random Forest (max depth-11)	0.55	0.54	0.55	0.54

Table. 146. Fine-Tuned GPT Hinglish Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

Fine-Tuned XLM Base Model

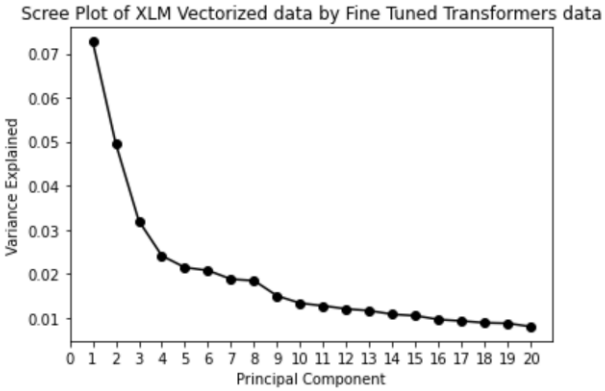

Figure. 67. Scree Plot for Fine-Tuned XLM Base Vectors of Nisha's Dataset.

According to the Scree plot of Figure 57, 4 Components are selected for Dimension Reduction.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.33	0.29	0.33	0.28
Gaussian Naïve Bayes	0.31	0.31	0.31	0.29
Bernoulli Naïve Bayes	0.27	0.19	0.27	0.22
Multinomial Naïve Bayes	0.26	0.26	0.26	0.23
SVM (Poly)	0.35	0.35	0.35	0.33
KNN (7 Neighbors)	0.34	0.33	0.34	0.33
Decision Tree (max depth-7)	0.36	0.37	0.36	0.34

Random Forest (max depth-11)	0.37	0.38	0.37	0.37

Table. 147. Fine-Tuned XLM Base Vectorized Models and Metrics of Nisha's Dataset after PCA and ICA.

4.2.5 Hypothesis Testing of Models

Based on the metrics of Models without Scaling and Component Analysis, Scaled Models and Component Analysis applied Models, the best metrics are obtained for Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM Classification Model (Accuracy-77%), Normalize Scaled XLM Base Sentence Transformer – SVM Classification Model (Accuracy-78%), Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM Classification Model (Accuracy-77%), Standard Scaled XLM Base Sentence Transformer - SVM Classification Model (Accuracy-78%).

Some Models like SVM Classification and Logistic Regression of Normalize Scaled XLM Base Sentence Transformer and Standard Scaled XLM Base Sentence Transformer have metrics showing similar performance. To check the performance of the models statistically, a Hypothesis test is conducted between the models like Standardized XLM Base Sentence Transformer's Logistic Regression and SVM and Normalized XLM Base Sentence Transformer's Logistic Regression and SVM. The results of the Hypothesis test using Paired T-Test and Cross-Validation for the models are conducted.

Standard Scaled XLM Base Sentence Transformer – Logistic Regression and SVM

The Paired T-Test is conducted for both the models for alpha 0.05 value. The t-statistic obtained is 2.99 and the p-value obtained is 0.03. As the p-value is less than 0.05, the Null hypothesis is rejected and there is a difference between the models' performance. To find the performance of the model more accurately, Mean Accuracies are calculated after k-fold cross-validation using 10-fold. The Accuracies are visualized using Boxplots which are shown in Figure 58. As SVM has good mean accuracy than Logistic Regression, SVM is considered as best Algorithm for the Standard Scaled Fine-Tuned GPT Hinglish dataset.

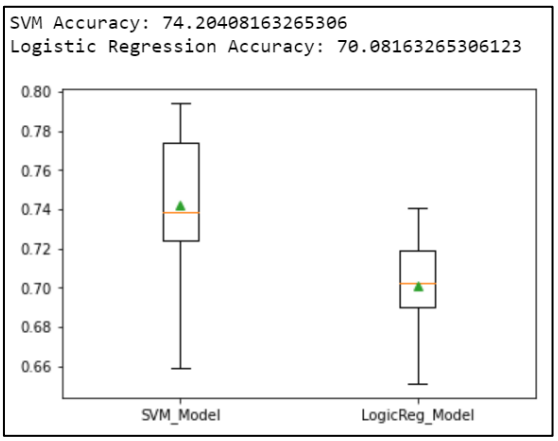

Figure. 68. Mean Accuracies of SVM and Logistic Regression

Normalize Scaled XLM Base Sentence Transformer – Logistic Regression and SVM

The Paired T-Test is conducted for both the models for alpha 0.05 value. The t-statistic obtained is 1.318 and the p-value obtained is 0. As the p-value is less than 0.245, the Null hypothesis cannot be rejected and there is no difference between the models' performance. To find the performance of the model more accurately, Mean Accuracies are calculated after k-fold cross-validation using 10-fold. The Accuracies are visualized using Boxplots

which are shown in Figure 59. As SVM has good mean accuracy than Logistic Regression, SVM is considered as best Algorithm for the Standard Scaled Fine-Tuned GPT Hinglish dataset.

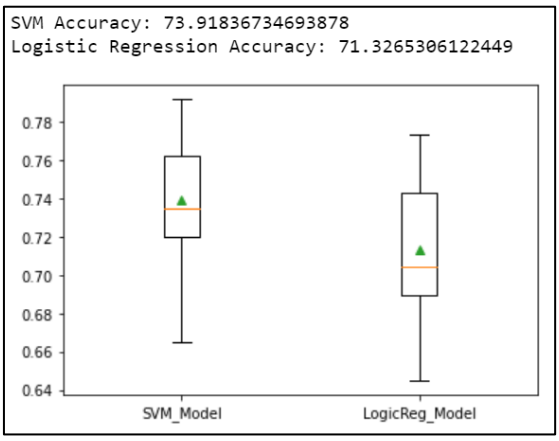

Figure. 69. Mean Accuracies of SVM and Logistic Regression

4.2.6 Hyperparameter Tuning

Hyperparameter Tuning is done for the 4 best models to check whether there is improvement in the performance of the models namely Normalize Scaled Verloop BERT Hinglish Sentence Transformer - SVM Model, Standard Scaled Verloop BERT Hinglish Sentence Transformer - SVM Model, Normalize Scaled XLM Base Sentence Transformer - SVM Model, Standard Scaled XLM Base Sentence Transformer - SVM Model. Coarse Fine-Tuning is done while performing Hyperparameter Tuning. Grid Search CV is done after finding the Hyperparameters through Random Search CV. The best parameters found for the 4 models are mentioned in Table 78 along with the metrics.

Model and Parameters	Accuracy	Precision	Recall	F1-Score
Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM (C=10)	0.77	0.77	0.77	0.77
Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM (C=100, gamma=0.001)	0.78	0.78	0.78	0.78
Normalize Scaled XLM Base Sentence Transformer – SVM (C=100, gamma=0.1)	0.78	0.77	0.78	0.77

Standard Scaled XLM Base Sentence Transformer - SVM (C=1, gamma='auto')	0.78	0.78	0.78	0.78

Table. 148. Hyperparameter Tuned Best Models and Metrics of Nisha's Dataset.

4.2.7 AUC ROC Curves

The Area Under Curve for the best models of Nisha's dataset is mentioned visually in the Receiver Operation Characteristic Curve plots. Based on the Visual Plots from Figure 60, Figure 61, Figure 62, and Figure 63, the results are clear that Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM, Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM, Normalize Scaled XLM Base Sentence Transformer – SVM, Standard Scaled XLM Base Sentence Transformer – SVM are best models for Nisha's Dataset.

Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM

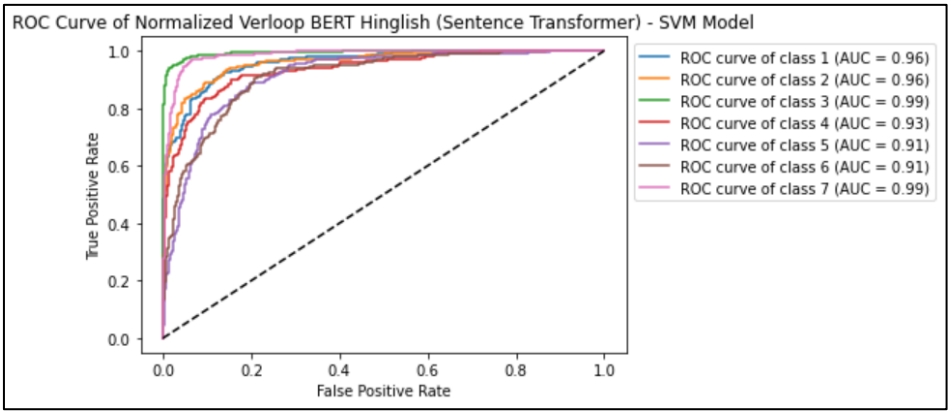

Figure. 70. ROC Curves for Normalize Scaled Verloop BERT Hinglish Sentence Transformer – SVM of Nisha's Dataset

Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM

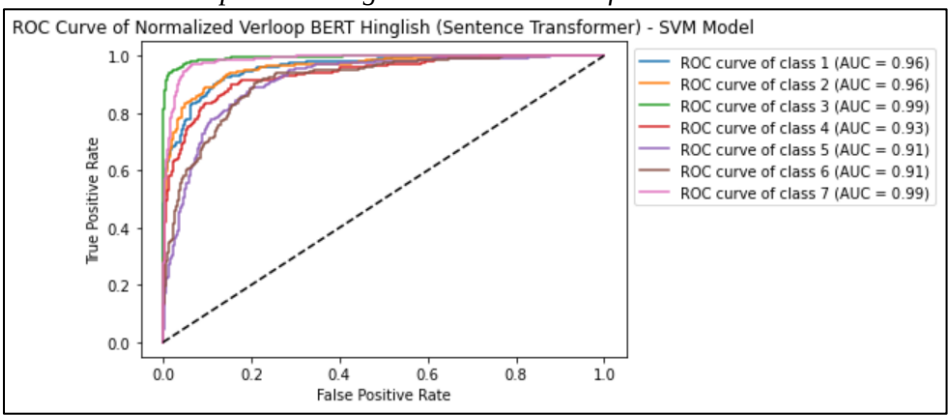

Figure. 71. ROC Curves for Standard Scaled Verloop BERT Hinglish Sentence Transformer – SVM of Nisha's Dataset

Normalize Scaled XLM Base Sentence Transformer – SVM

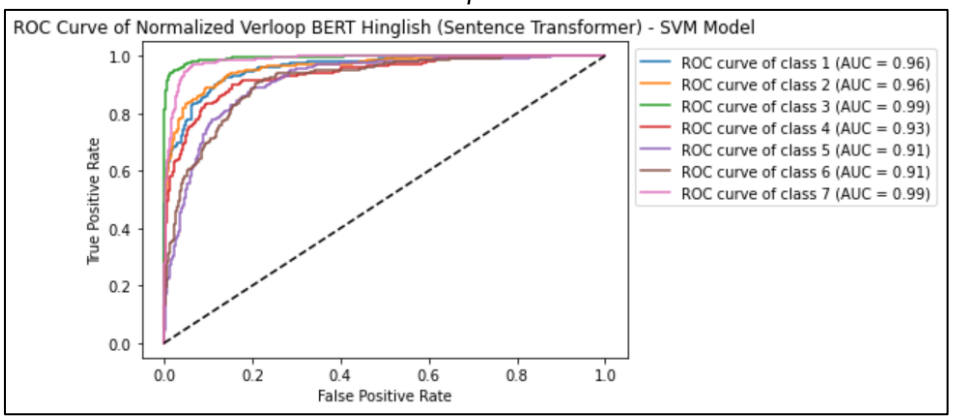

Figure. 72. ROC Curves for Normalize Scaled XLM Base Sentence Transformer – SVM of Nisha's Dataset

Standard Scaled XLM Base Sentence Transformer – SVM

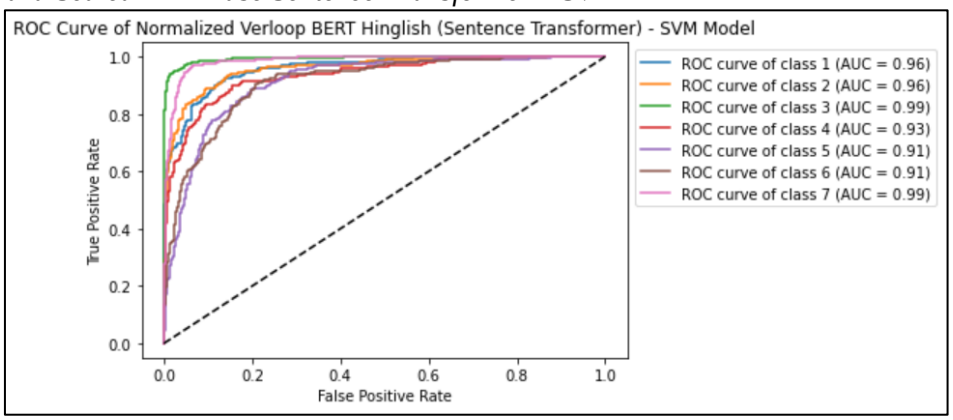

Figure. 73. ROC Curves for Standard Scaled XLM Base Sentence Transformer – SVM of Nisha's Dataset

4.3 Final Models Saving and Testing

Based on the evaluation metrics like Accuracy, Precision, Recall, F1-Score, and AUC-ROC after Hyperparameter tuning, final models are trained with 100% dataset and tested with external data.

4.3.1 Kabita's Kitchen Dataset

The final model for Kabita's dataset is Standard Scaled Verloop BERT Sentence Transformer – SVM Model. The Evaluation metrics of the model are shown in Table 77. The model is trained with total data and saved as a "sav file" using Pickle. The model is loaded and tested with random comments to check the predictions for the comments given. The predictions of Kabita's model are mentioned in Table 150.

Comment	Prediction
Thankyou for the video bilkul pasand karta hun aapki videos	Gratitude
Mai try kiya aaj delicious recipe : About Recipe	About Recipe
video ki clarity bahot achi h	Praising Chef
bahot beautiful dikh rahe ho	Praising Chef
video achi h aur aap bhi	Undefined

yeh saal mein who is watching this video comment like karo	Undefined
kaun si company chilli powder use kar rahe ho	Suggestion/Query

Table. 149. Predictions of Kabita's Dataset Model.

4.3.2 Nisha's Dataset

The final model for the Nisha dataset is Standard Scaled Verloop BERT Sentence Transformer – SVM Model. The Evaluation metrics of the model are shown in Table 149. The model is trained with total data and saved as a "sav file" using Pickle. The model is loaded and tested with random comments to check the predictions for the comments given. The predictions of Kabita's model are mentioned in Table 151.

Comment	Prediction
Thankyou for the video bilkul pasand karta hun aapki videos	Gratitude
Mai try kiya aaj delicious recipe : About Recipe	About Recipe
video ki clarity bahot achi h	Praising Chef
bahot beautiful dikh rahe ho	Praising Chef
video achi h aur aap bhi	Praising Chef
yeh saal mein who is watching this video comment like karo	Undefined
kaun si company chilli powder use kar rahe ho	Suggestion/Query

Table. 150. Predictions of Kabita's Dataset Model.

5. Ethical Considerations

5.1 Harms and Benefits of the project

Like any other Technology and invention, Natural Language processing also has benefits and harms based on the projects it is being implemented on. As per the Confusion matrix of ethics, this research is ethically implemented through good methods yielding good results. The Project which is implemented by Analysing sentiments on YouTube comments has more benefits when compared to disadvantages as it saves a lot of time and manual tasks. Previously and in some present channels, the comments on YouTube are being examined for knowing the review emotion given by subscribers or viewers through manual reading and commenting. But this project which has Natural Language Processing integration helps in predicting the type of comment that the viewer has given, and the advanced model helps in giving a reply to the comment based on the emotion.

Figure. 74. Sentimental Analysis of YouTube comments. *Source:* (Ripul Agarwal 2020)

The Ethical challenges of this project include the applications that can use the implemented models on data which is either for the good or bad purpose of sentiment mining. The manual comment reviewers of social media channels or streamers will have their roles at risk as this model can replace their

positions as it will predict the comment emotion in less time instead, they can upgrade themselves by developing advanced techniques for this model.

5.2 Harms and Benefits linked to the data

5.2.1 Benefits

Helps in understanding the different vectorization techniques which remove the meaning of the words for analysis by the model and instead, convert strings to numerical forms to make the model understand the patterns in data. Making an understanding of different models and the evaluation results on the data taken helps in preferring models when more data is added for training instead of starting from the initial stage. Analyzation on mix codes like Hinglish (Hindi + English), Marglish (Marathi + English), Tenglish (Telugu + English), etc. which are realistic in conversations, speech recognition systems, etc. by using this type of Natural Language processing model.

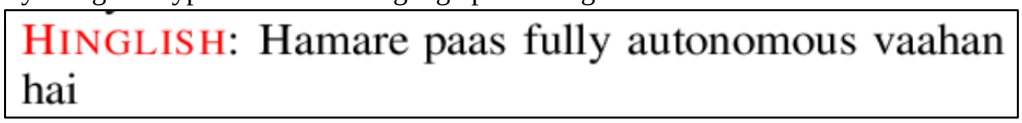

Figure. 75. Hinglish Text. *Source:* (Vivek Srivastava 2021)

5.2.2 Harms

Harms include the risk of not utilizing the data with consistency. The present data is transparent and consistent as the type of comments concerning emotions is equal in number. It also includes the reusability of data in the future as data used once can't be used again for training the model. There will be no concern regarding privacy for input data as the data is open-source and security policies are strictly followed while storing the data that including models, code snippets, procedures, etc.

5.3 Ethical Challenges with Dissertation

5.3.1 Collection of Data and Usage

Data is collected from UCI Machine Learning Repository. Data is open-sourced and is cited in the references as working on the same. Data is collected from Hinglish comments on two famous YouTube channels of Nisha Madhulika and Kabita. Data used for the Natural Language Processing project consists of Questions, Praise, Suggestions, Gratitude, About the Recipes, and Videos. It doesn't include any human personal information but only the comments of the people for the work mentioned on YouTube's channels. Data is not shared or sold to any third parties nor republished as it is only for study and research purposes. As the data is open source, no consent forms or privacy policies are attached to it. In case of any further research continued with the data produced by our models, they should cite this paper in their research work. The terms of our data policy will be clear and understandable indirect way, and they will be mentioned along with the report instead of clicking-through or buttons response. We can't make any changes to the data on the open-source, and we have the chance to modify it according to our use case.

5.3.2 Data Storage, Security, and Stewardship

Data is stored in such a way that its copy is available along with the if it got missed from a system. The method of remote storage will be mentioned in the report. Version control is applied to the data of research to track the changes and to revert if necessary. On systems, data is protected with a password locker. The further plan is to implement data anonymization and make the data encrypted so that even if a data breach happens, data

can be read. There is no risk in data storage for a long period as our case study doesn't include any personal information. The data is thoroughly examined and updated according to the requirement of the models in the research. Permissions for data including modifications, deletions, etc. are not given to any other people as the project is single-handled.

5.3.3 Data Hygiene and Relevance

The data collected is semi-structured and consists of sentences to analyze the type of comment. The datasets include two files of two YouTube channel comments of 4900 rows each. They have undergone different vectorization techniques after data wrangling. Different data is extracted from it like Hashtags, Numbers, Average words in the sentence, etc. to analyze the data using some visualizations. The sentences are converted to word vectors which will be Numerical data and Models are built on those vectorized data for predictions. Models include Parametric and Non-parametric Algorithms and Cross-validation is applied to the data for all Algorithms to check the correct accuracy and finalize the model for prediction. Data of models and evaluation results don't contain any sensitive information like passwords, Access codes, etc. When coming to data integrity, it will be consistent in all systems irrespective of the platform. The bias of data concerning gender or race won't be applicable here as it includes the type of comments, but no independent variable includes the nominal data. The labels of the response variable are 7 unique types and equally distributed with 700 rows each for each dataset. There will be no expiry for data in the Natural Language Processing as the synonyms of sentences don't change over time and more data can be added in the future to increase the training of the model.

5.3.4 Identifying and Addressing Harmful Bias

As all labels are considered and distributed equally in the response variable, there will be no bias in the models reducing underfitting increases the prediction level of unknown data. The non-bias nature of the datasets resembles consisting of 4900 rows each containing 7 types of 700 labels equally distributed.

5.3.5 Validation and Testing of Data Models

The challenges facing this include the finding of vectorization methods for mixed codes as mixed codes can be noticed only during normal conversation or giving comments in the native language. The stop words set for Hindi and English can't be found according to our use case so created stop words data manually based on labels taken in the target variable. To make the model more perfect in Natural language processing, the model should be continuously trained with more data for increasing accuracy in predictions. Different Algorithms are to be used for different vectorization techniques to finalize the Algorithm used for final training using cross-validation techniques.

5.4 SWOT Analysis

SWOT analysis is used to get aware of the factors while making decisions and strategies implemented (Stephen J. Bigelow 2022), it is applied to this research model for analyzing its Strengths, Weaknesses, Opportunities, and Threats. The SWOT points according to the thesis are mentioned below.

Strengths (Internal Positives)

- Unstructured text data can be vectorized and analyzed (Amanda Porter 2022). The unstructured data might be from documents, No-SQL databases, charts, etc. This data can be converted into structured data that can help analyze.
- Accurate than human analysis. The problems like Spam filtering, Name Entity Recognition (NER), etc can be solved using pre-trained and straightforward methods.
- Better understanding of market and customer satisfaction (Rachel Wolff 2020). Customer surveys can be analyzed using NLP and Deep Learning models by transferring the data into a binary or multi-class problem.
- Saves money and time (Shemmy Majewski 2020). Email routing is one of the best examples. Emails based on the departments or types can be classified using NLP and Classification methods of Machine Learning.

Weaknesses (Internal Negatives)

- Training takes a lot of time. Because of the complexity of the data collected from sources like Wikipedia, Web Scraping, etc pattern recognition takes a lot of time.
- Difficult to get 100% accuracy (Ximena Bolaños 2020). The main challenge in Machine Learning or Deep Learning is not what to learn. It is what not to learn. 100% Accuracy indicates overlearning. A model should be able to predict new data patterns but not remember training data.
- Ambiguity in phrases, Words with different contexts have different meanings. If sarcasm enters the input, unlike humans it is difficult for the machine to predict whether it is real or a joke.
- Low resource languages and mixed codes stop words need to be introduced manually (Inés Roldós 2020). Stop words are the most common words which don't add any context to the sentence. They are available in huge numbers in any language. A list of stop words should be prepared based on the requirement.

Opportunities (External Positives)

- Application of NLP in Education (Burstein 2009) includes the verification of Academic writings, Sentiment Analysis, etc. It helps the faculty and tutors to know the knowledge of the students based on their assignments.
- Predictive texts, Search results, Email filters, etc. (Natural Language Processing (NLP) Examples | Tableau n.d.). In Applications like Search results, the cross-lingual models help in the way to get the results if the input is in mix-code.
- Comments Analysis, Social Media Monitoring, Recruitment, etc. (Abhishek Sharma 2020). Analysis of Survey info which is from sources like social media is one of the main applications of the NLP. CV mining is one of the common procedures in job recruitment that is based on NLP.
- Intelligence gathering on financial stocks and marketing research, Report Auto-generation (Ilia Lorin 2020). Financial documents are the type of unstructured data. Risk Analysis, Financial Sentiments, etc can be the input to NLP models to forecast the stock movements based on the vocabulary in the corpus.

Threats (External Negatives)

- Ambiguous and vague models as they can't recognize the meaning and are unclear (Pamela Fox 2018). General models like Bag-of-Words can't find the context present in the input as they mostly depend on mathematical findings and formulae.
- Biasness of Human speech is getting stored in the machines where they show the same nature. Due to the differences shown by the people and mirroring them in the surveys and documents, the models are learning the context without bothering about the good or bad. Racism, Regionalism, and religious differences are some examples of human biases.
- Loss of manual task jobs due to automated NLP applications. Developing automated solutions decreases manual work therefore the need for manual workers slowly disappears. Instead, manual workers can upgrade themselves in the development of automation.

6. Conclusions and Future Work

YouTube is one of the popular mediums for learning and gaining knowledge about new things. It also acts as an entertainment network apart from the learnings. Many videos will be uploaded on YouTube on daily basis. Many people as a part of their daily activity, like to try and learn new cooking recipes and new cuisines. Due to this, YouTubers need to focus on the quality of the content based on the users' requirements and reviews. This use case helps the cooking channel admins in adding the content supported by the users in the videos. The main aim of this sentimental analysis is to find the best combination of vectorizers, scaling techniques, and Machine Learning models on the user comments. Based on the evaluation metrics, of all the Vectorizers taken in this project, Verloop BERT Hinglish Vectorizer which has BERT Transformer Architecture in the backend and pre-trained on Hinglish data suits more while converting the Hinglish comments into vector forms. Both Normalize and Standard Scalers help increase the performance of the model, but Standard scaler has more weightage in scaling methods for the taken datasets. SVM yielded the best results when compared to other parametric and non-parametric models for the Hinglish data. Component Analysis helps recognize the patterns in the data without loss of information. But in this project, there is no improvement in the performance of the models after applying Component analysis for both Kabita and Nisha's datasets.

The future work for this analysis includes the implementation of deep learning and neural network models on the same datasets and evaluating them for the best model. Analysis should include animations and emojis in future work. Other channel types like educational, music, sci-fi, etc topics will be covered for the sentimental analysis. Integrating Database can be useful for storing and rapid search of data. In case data becomes huge, integrating Apache Spark can be useful as it has an in-built Machine Learning library. The saved 'sav' file after final model training can be deployed along with the web interface developed using Python web-development frameworks like Flask or Django. This project can be extended to the Named Entity Recognition (NER) model to extract the necessary information from the comments.

7. References

A Short History of Machine Learning -- Every Manager Should Read. Available from: https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/?sh=eb5887015e78 [accessed 26 May 2022].

AbdulNabi, I. and Yaseen, Q. (2021). Spam email detection using deep learning techniques. In: *Procedia Computer Science*. Elsevier B.V., pp.853–858.

Abhishek Sharma. (2020). *Applications Of Natural Language Processing (NLP)* [online]. Available from: https://www.analyticsvidhya.com/blog/2020/07/top-10-applications-of-natural-language-processing-nlp/ [accessed 11 April 2022].

Agarwal, V., Pooja Rao, S.B. and Jayagopi, D.B. (2021). Towards Code-Mixed Hinglish Dialogue Generation. In: *International Conference Recent Advances in Natural Language Processing, RANLP*. Incoma Ltd, pp.7–15.

Agrawal, S.C., Singh, S. and Gupta, S. (2021). Evaluation of Machine Learning Techniques in Sentimental Analysis. In: *2021 5th International Conference on Information Systems and Computer Networks, ISCON 2021*. Institute of Electrical and Electronics Engineers Inc.

Alsaffar, A. and Omar, N. (2015). Integrating a Lexicon based approach and K nearest neighbour for Malay sentiment analysis. *Journal of Computer Science*, 11(4), pp.639–644.

Amanda Porter. (2022). *What are the advantages of Natural Language Processing in AI? - Capacity* [online]. Available from: https://capacity.com/enterprise-ai/faqs/what-are-the-advantages-of-natural-language-processing-nlp/ [accessed 11 April 2022].

Aro, T.O., Dada, F., Oluwagbemiga Balogun, A. and Oluwasogo, S.A. (2019). Stop Words Removal on Textual Data Classification. *International Journal of Information Processing and Communication (IJIPC*, 7(1), pp.1–9.

Arora, A., Shrivastava, A., Mohit, M., Sainz-Maza, L., Facebook, L. and Facebook, A.A. (2020). *Cross-lingual Transfer Learning for Intent Detection of Covid-19 Utterances*. Available from: https://fb.me/covid_.

Bansal, N., Goyal, V. and Rani, S. (2020). Experimenting Language Identification for Sentiment Analysis of English Punjabi Code Mixed Social Media Text. *International Journal of E-Adoption*, 12(1), pp.52–62.

BERT Explained: State of the art language model for NLP | by Rani Horev | Towards Data Science. Available from: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270 [accessed 29 August 2022].

bert-base-uncased · Hugging Face. Available from: https://huggingface.co/bert-base-uncased [accessed 26 August 2022].

Bhavitha, B.K., Rodrigues, A.P. and Chiplunkar, N.N. (2017). Comparative study of Machine Learning techniques in sentimental analysis. In: *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2017*. Institute of Electrical and Electronics Engineers Inc., pp.216–221.

Burstein, J. (2009). Opportunities for natural language processing research in education. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Davchev, J., Mishev, K., Vodenska, I., Chitkushev, L., Trajanov, D. and Dimitar, T. (2021). *Bitcoin Price Prediction using Transfer Learning on Financial Micro-blogs*. Available from: https://www.ceeol.com/search/article-detail?id=978526.

Decision Trees in Machine Learning | by Prashant Gupta | Towards Data Science. Available from: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 [accessed 31 August 2022].

Devika, R., Vairavasundaram, S., Mahenthar, C.S.J., Varadarajan, V. and Kotecha, K. (2021). A Deep Learning Model Based on BERT and Sentence Transformer for Semantic Keyphrase Extraction on Big Social Data. *IEEE Access*, 9, pp.165252–165261.

Elgeldawi, E., Sayed, A., Galal, A.R. and Zaki, A.M. (2021). Hyperparameter tuning for Machine Learning algorithms used for arabic sentiment analysis. *Informatics*, 8(4).

Examining the Transformer Architecture | by James Montantes | Towards Data Science. Available from: https://towardsdatascience.com/examining-the-transformer-architecture-part-1-the-openai-gpt-2-controversy-feceda4363bb [accessed 30 August 2022].

Fang, X. and Zhan, J. (2015). Sentiment analysis using product review data. *Journal of Big Data*, 2(1).

ganeshkharad/gk-hinglish-sentiment · Hugging Face. Available from: https://huggingface.co/ganeshkharad/gk-hinglish-sentiment [accessed 26 August 2022].

Gao, Y. and Cui, Y. (2021). *Multi-ethnic Survival Prediction: Transfer Learning with Cox Neural Networks*. Available from: https://gdc.

gpt2 · Hugging Face. Available from: https://huggingface.co/gpt2 [accessed 26 August 2022].

Harfoushi, O., Hasan, D. and Obiedat, R. (2018). Sentiment Analysis Algorithms through Azure Machine Learning: Analysis and Comparison. *Modern Applied Science*, 12(7), p.49.

Holderrieth, P., Gong, W., Smith, S.M. and Peng, H. (2021). *Transfer learning works in neuroimaging via feature re-use*.

Hoque, K.E. and Aljamaan, H. (2021). Impact of hyperparameter tuning on Machine Learning models in stock price forecasting. *IEEE Access*, 9, pp.163815–163830.

Hyperparameter Tuning | Evaluate ML Models with Hyperparameter Tuning. Available from: https://www.analyticsvidhya.com/blog/2021/04/evaluating-machine-learning-models-hyperparameter-tuning/ [accessed 27 August 2022].

Ilia Lorin. (2020). *Natural Language Processing (NLP) Use Cases in Business - MobiDev* [online]. Available from: https://mobidev.biz/blog/natural-language-processing-nlp-use-cases-business [accessed 11 April 2022].

impyadav/GPT2-FineTuned-Hinglish-Song-Generation · Hugging Face. Available from: https://huggingface.co/impyadav/GPT2-FineTuned-Hinglish-Song-Generation [accessed 26 August 2022].

India Population (2022) - Worldometer. (2022). *worldometers* [online]. Available from: https://www.worldometers.info/world-population/india-population/ [accessed 8 June 2022].

Inés Roldós. (2020). *Major Challenges of Natural Language Processing (NLP)* [online]. Available from: https://monkeylearn.com/blog/natural-language-processing-challenges/ [accessed 11 April 2022].

Irawaty, I., Andreswari, R. and Pramesti, D. (2020). Vectorizer Comparison for Sentiment Analysis on Social Media Youtube: A Case Study. In: *2020 3rd International Conference on Computer and Informatics Engineering, IC2IE 2020*. Institute of Electrical and Electronics Engineers Inc., pp.69–74.

Jo Hartley. (2021). *The Languages of India: What Languages are Spoken in India?* [online]. Available from: https://www.berlitz.com/blog/indian-languages-spoken-list [accessed 8 June 2022].

Kadriu, A., Abazi, L. and Abazi, H. (2019). Albanian Text Classification: Bag of Words Model and Word Analogies. *Business Systems Research*, 10(1), pp.74–87.

Kaur, G., Kaushik, A. and Sharma, S. (2019). Cooking is creating emotion: A study on hinglish sentiments of youtube cookery channels using semi-supervised approach. *Big Data and Cognitive Computing*, 3(3).

K-Nearest Neighbor. A complete explanation of K-NN | by Antony Christopher | The Startup | Medium. Available from: https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4 [accessed 1 September 2022].

Kumar, A. and Sachdeva, N. (2020). Multi-input integrative learning using deep neural networks and transfer learning for cyberbullying detection in real-time code-mix data. In: *Multimedia Systems*. Springer.

Kumar, V. and Subba, B. (2020). A tfidfvectorizer and SVM based sentiment analysis framework for text data corpus. In: *26th National Conference on Communications, NCC 2020*. Institute of Electrical and Electronics Engineers Inc.

Logistic Regression: Equation, Assumptions, Types, and Best Practices. Available from: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/ [accessed 31 August 2022].

Matthew Johnston. (2022). *7 Companies Owned by Google's Parent Company Alphabet (GOOGL),* [online]. Available from: https://www.investopedia.com/investing/companies-owned-by-google/ [accessed 8 June 2022].

Muennighoff/SGPT-125M-mean-nli · Hugging Face. Available from: https://huggingface.co/Muennighoff/SGPT-125M-mean-nli [accessed 26 August 2022].

Mundra, S. and Mittal, N. (2021). Evaluation of text representation method to detect cyber aggression in hindi english code mixed social media text. In: *ACM International Conference Proceeding Series*. Association for Computing Machinery, pp.402–409.

Naive Bayes Classifier. What is a classifier? | by Rohith Gandhi | Towards Data Science. Available from: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c [accessed 31 August 2022].

Narasimha/hinglish-distilbert · Hugging Face. Available from: https://huggingface.co/Narasimha/hinglish-distilbert [accessed 26 August 2022].

Natural Language Processing - Ela Kumar - Google Books. Available from: https://books.google.ie/books?hl=en&lr=&id=FpUBFNFuKWgC&oi=fnd&pg=PP2&dq=history+of+natural+language+processing&ots=GFy26LlyPw&sig=Qw6__PkPsebesXomRymAy6PXRsI&redir_esc=y#v=onepage&q=alan&f=false [accessed 26 May 2022].

Natural Language Processing (NLP) Examples | Tableau. Available from: https://www.tableau.com/learn/articles/natural-language-processing-examples [accessed 11 April 2022].

Nguyen, T.H., Shirai, K. and Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, 42(24), pp.9603–9611.

Ottoni, A.L.C. and Novo, M.S. (2021). A Deep Learning Approach to Vegetation Images Recognition in Buildings: A Hyperparameter Tuning Case Study. *IEEE Latin America Transactions*, 19(12), pp.2062–2070.

Pamela Fox. (2018). *Expressing an algorithm | AP CSP (article) | Khan Academy* [online]. Available from: https://www.khanacademy.org/computing/ap-computer-science-principles/algorithms-101/building-algorithms/a/expressing-an-algorithm [accessed 11 April 2022].

Pan, J. (2010). *Feature-Based Transfer Learning With Real-World Applications* [unpublished]. .

Qu, S., Yang, Y. and Que, Q. (2021). Emotion classification for spanish with xlm-roberta and textcnn. In: *CEUR Workshop Proceedings*. CEUR-WS, pp.94–100.

Rachel Wolff. (2020). *7 Benefits of Natural Language Processing (NLP)* [online]. Available from: https://monkeylearn.com/blog/nlp-benefits/ [accessed 11 April 2022].

Ripul Agarwal. (2020). *Sentiment Analysis of YouTube Comments | Analytics Steps* [online]. Available from: https://www.analyticssteps.com/blogs/sentiment-analysis-youtube-comments [accessed 11 April 2022].

sentence-transformers/bert-base-nli-mean-tokens · Hugging Face. Available from: https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens [accessed 26 August 2022].

sentence-transformers/stsb-xlm-r-multilingual · Hugging Face. Available from: https://huggingface.co/sentence-transformers/stsb-xlm-r-multilingual [accessed 26 August 2022].

Sentiment Analysis Guide. (2020). *Monkey Learn* [online]. Available from: https://monkeylearn.com/sentiment-analysis/ [accessed 8 June 2022].

Serrano-Guerrero, J., Olivas, J.A., Romero, F.P. and Herrera-Viedma, E. (2015). Sentiment analysis: A review and comparative analysis of web services. *Information Sciences*, 311, pp.18–38.

Shah, S.R., Kaushik, A., Sharma, S. and Shah, J. (2020). Opinion-mining on marglish and devanagari comments of youtube cookery channels using parametric and non-parametric learning models. *Big Data and Cognitive Computing*, 4(1), pp.1–19.

Shahin, A.I. and Almotairi, S. (2019). *Automated Arabic Sign Language Recognition System Based on Deep Transfer Learning*.

Shemmy Majewski. (2020). *7 Key Benefits Of Using Natural Language Processing In Business* [online]. Available from: https://dlabs.ai/blog/7-key-benefits-of-using-natural-language-processing-in-business/ [accessed 11 April 2022].

Singh, M. and Goyal, V. (2020). Sentiment Analysis of {E}nglish-{P}unjabi Code-Mixed Social Media Content. In: *Proceedings of the 17th International Conference on Natural Language Processing (ICON): System Demonstrations*. NLP Association of India (NLPAI), pp.24–25. Available from: https://aclanthology.org/2020.icon-demos.9.

Singh, P. and Lefever, E. (2020). *Sentiment Analysis for Hinglish Code-mixed Tweets by means of Cross-lingual Word Embeddings*.

Sklearn Random Forest Classifiers in Python Tutorial | DataCamp. Available from: https://www.datacamp.com/tutorial/random-forests-classifier-python [accessed 1 September 2022].

Srivastava, V. and Singh, M. (2021). Quality Evaluation of the Low-Resource Synthetically Generated Code-Mixed Hinglish Text. In: *INLG 2021 - 14th International Conference on Natural Language Generation, Proceedings*.

Statistical Significance Tests for Comparing Machine Learning Algorithms. Available from: https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/ [accessed 27 August 2022].

Stephen J. Bigelow. (2022). *What Is a SWOT Analysis? Definition and Examples - TechTarget* [online]. Available from: https://www.techtarget.com/searchcio/definition/SWOT-analysis-strengths-weaknesses-opportunities-and-threats-analysis [accessed 11 April 2022].

Stuke, A., Rinke, P. and Todorovic, M. (2021). Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization. *Machine Learning: Science and Technology*, 2(3).

Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science. Available from: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 [accessed 1 September 2022].

Swaminathan, S., Ganesan, H.K. and Pandiyarajan, R. (2020). HRS-TECHIE@Dravidian-CodeMix and HASOC-FIRE2020: Sentiment analysis and hate speech identification using Machine Learning, deep learning and ensemble models. In: *CEUR Workshop Proceedings*. CEUR-WS, pp.241–252.

The Illustrated GPT-2 (Visualizing Transformer Language Models) – Jay Alammar – Visualizing Machine Learning one concept at a time. Available from: https://jalammar.github.io/illustrated-gpt2/ [accessed 31 August 2022].

Thelwall, M. (2018). Gender bias in Machine Learning for sentiment analysis. *Online Information Review*, 42(3), pp.343–354.

UCI Machine Learning Repository: Youtube cookery channels viewers comments in Hinglish Data Set. Available from: https://archive.ics.uci.edu/ml/datasets/Youtube+cookery+channels+viewers+comments+in+Hinglish [accessed 8 April 2022].

Uma Gunturi. (2020). *A Primer on Code Mixing & Code Switching! | by Uma Gunturi | Medium* [online]. Available from: https://umagunturi789.medium.com/a-primer-on-code-mixing-code-switching-9bbde2a15e57 [accessed 11 June 2022].

Valencia, F., Gómez-Espinosa, A. and Valdés-Aguirre, B. (2019). Price movement prediction of cryptocurrencies using sentiment analysis and Machine Learning. *Entropy*, 21(6).

verloop/Hinglish-Bert · Hugging Face. Available from: https://huggingface.co/verloop/Hinglish-Bert [accessed 26 August 2022].

Vivek Srivastava. (2021). *A representative Hinglish sentence and the corresponding parallel... | Download Scientific Diagram* [online]. Available from: https://www.researchgate.net/figure/A-representative-Hinglish-sentence-and-the-corresponding-parallel-Hindi-English-sentences_fig1_352432102 [accessed 11 April 2022].

Wazirali, R. (2020). An Improved Intrusion Detection System Based on KNN Hyperparameter Tuning and Cross-Validation. *Arabian Journal for Science and Engineering*, 45(12), pp.10859–10873.

Williams, L., Bannister, C., Arribas-Ayllon, M., Preece, A. and Spasić, I. (2015). The role of idioms in sentiment analysis. *Expert Systems with Applications*, 42(21), pp.7375–7385.

Ximena Bolaños. (2020). *Natural Language Processing with Machine Learning* [online]. Available from: https://www.encora.com/insights/natural-language-processing-with-machine-learning [accessed 11 April 2022].

XLM — Enhancing BERT for Cross-lingual Language Model | by Rani Horev | Towards Data Science. Available from: https://towardsdatascience.com/xlm-enhancing-bert-for-cross-lingual-language-model-5aeed9e6f14b [accessed 29 August 2022].

xlm-mlm-en-2048 · Hugging Face. Available from: https://huggingface.co/xlm-mlm-en-2048 [accessed 26 August 2022].

YouTube | History, Founders, & Facts | Britannica. Available from: https://www.britannica.com/topic/YouTube [accessed 26 May 2022].

Zhang, F., Petersen, M., Johnson, L., Hall, J. and O'Bryant, S.E. (2021). Accelerating Hyperparameter Tuning in Machine Learning for Alzheimer's Disease With High Performance Computing. *Frontiers in Artificial Intelligence*, 4.

Zhao, J., Shetty, S., Pan, J.W., Kamhoua, C. and Kwiat, K. (2019). Transfer learning for detecting unknown network attacks. *Eurasip Journal on Information Security*, 2019(1).