A Project Report on

# TELUGU DATA CLASSIFICATION

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

## Bachelor of Technology

## In

## Computer Science and Engineering

Submitted by

V. Durga Bhavani
(20H51A05D4)

R. Narasimha
(21H55A0519)

V. Keerthana
(21H55A0524)

Under the esteemed guidance of

Ms. Komal Parashar
(Assistant Professor)

EXPLORE TO INVENT

**Department of Computer Science and Engineering**

**CMR COLLEGE OF ENGINEERING & TECHNOLOGY**
(UGC Autonomous)
*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A$^+$ Grade
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

**2023- 2024**

# CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## CERTIFICATE

This is to certify that the Major Project report entitled **"Telugu Data Classification"** being submitted by V.Durga Bhavani (20H51A05D4), R.Narasimha (21H55A0519), V.Keerthana (21H55A0524) in partial fulfillment for the award of Bachelor of Technology in **Computer Science and Engineering** is a record of bonafide work carried out under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

| | | |
|---|---|---|
| **Ms. Komal Parashar** | **Dr. Siva Skandha Sanagala** | |
| Assistant Professor | Associate Professor and HOD | |
| **Dept. of CSE** | **Dept. of CSE** | **EXTERNAL EXAMINER** |

# ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Ms. Komal Parashar, Assistant Professor** , Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**,** **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and Shri Ch Abhinav Reddy, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

| | |
|---|---|
| V. Durga Bhavani | 20H51A05D4 |
| R. Narasimha | 21H55A0519 |
| V. Keerthana | 21H55A0524 |

# TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# ABSTRACT

Telugu, being a highly morphologically rich Dravidian language, presents significant challenges in natural language processing (NLP) tasks. The proliferation of Telugu documents online necessitates efficient organization through automated categorization into predefined topics. This study focuses on the classification of Telugu data into distinct domains, including business, sports, entertainment, nation, and editorial, using machine learning classifiers such as Support Vector Machine (SVM), Naive Bayes, and Logistic Regression. The performance of these classifiers is evaluated based on metrics such as accuracy, precision, recall, and F1-score. Results indicate that the Naive Bayes model exhibits superior performance across these metrics. This research contributes to advancing Telugu text classification methodologies in the realm of machine learning, thereby facilitating improved organization and analysis of Telugu data available online.

# CHAPTER 1
# INTRODUCTION

# CHAPTER 1

# INTRODUCTION

### 1.1 Problem Statement

Telugu, being a complex Dravidian language, poses challenges in organizing its abundant online content. With the increasing usage of online platforms for sharing opinions and views in Telugu, there arises a need for accurately classifying Telugu documents into predefined topics like business, science, sports, etc. However, existing sentiment analysis techniques have not paid much attention to Telugu news articles. Therefore, there is a need to develop a robust methodology for text classification in Telugu to facilitate organizing, identification of user preferences, personalized content delivery, market analysis, and better engagement with targeted users.

### 1.2 Research Objective

**1. Collect and preprocess a Telugu news dataset:**

- **Data Collection**: Obtain a comprehensive Telugu news dataset from reliable sources such as Kaggle, news websites, or academic repositories.
- **Data Preprocessing**:
  - **Text Cleaning**: Remove special characters, punctuation marks, and irrelevant symbols from the text.
  - **Tokenization**: Split the text into individual words or tokens to prepare it for further processing.
  - **Stopword Removal**: Eliminate common words like articles, prepositions, and conjunctions that may not contribute much to classification.
  - **Stemming or Lemmatization**: Reduce words to their root form to normalize the vocabulary.

**2. Explore and implement feature extraction techniques:**

- **Count Vectorizing (One-Hot Encoding):** Count Vectorizing, also known as One-Hot Encoding, is a method used to represent words in numerical values as vectors. In this approach, each unique word in a corpus (collection of documents) is assigned a unique dimension. The vector representation of a word consists of zeros in all dimensions except for the one corresponding to that word, which contains a value of 1.

- **TF-IDF (Term Frequency-Inverse Document Frequency)**:
    - Calculate the importance of words based on their frequency in a document and their rarity across the entire corpus.
    - Term Frequency-Inverse Document Frequency (Tf-Idf) is another method used to convert words into numerical values for machine learning algorithms. It accounts for the importance of words in a document relative to their frequency across all documents in a corpus.
    - Tf (Term Frequency) measures how often a word appears in a document relative to the total number of words in that document: TF = (Number of times word W appears in a document) / (Total number of words in the document)
    - Idf (Inverse Document Frequency) measures the rarity of a word across all documents in the corpus. It is calculated as follows: IDF = log(Total number of documents / Number of documents containing word W)
- **N-grams**:
    - It refers to sequences of n consecutive words in a text document, where n can be any positive integer
    - Consider sequences of adjacent words to capture contextual information and improve classification accuracy.
        - ➢ Unigrams(1-grams)
        - ➢ Bigrams(2-grams)
        - ➢ Trigrams(3-grams)
        - ➢ N-Grams(N-grams)

- **Unigrams (1-grams):**
    - Unigrams are single words that appear in a text document.
    - Each word is treated as a separate feature in the feature vector.

  - Unigrams capture the presence or absence of individual words in the text.

- **Bigrams (2-grams):**
  - Bigrams are sequences of two consecutive words in a text document.
  - They capture the relationships between adjacent words in the text.

- **Trigrams (3-grams):**
  - Trigrams are sequences of three consecutive words in a text document.
  - They capture more complex relationships between words compared to bigrams.
  - Trigrams provide even more context and capture longer sequences of words.

**3. Evaluate and compare the performance of machine learning classification models:**

- **Naïve Bayes**:
  - A probabilistic classifier that assumes independence between features and calculates the probability of each class given the input features.
- **Support Vector Machine (SVM)**:
  - Strives to find the optimal hyperplane that separates data points of different classes with the maximum margin in a high-dimensional space.
- **Logistic Regression**:
  - A linear model used for binary or multiclass classification tasks, which calculates the probability of a particular class.

**4. Implement the classification pipeline and evaluate model performance:**

- **Model Training**:
  - Train the classification models using the preprocessed Telugu news dataset and the extracted features.
- **Model Evaluation**:
  - Assess the performance of each model using evaluation metrics such as accuracy, F1-score, precision, and recall.
  - Conduct cross-validation to ensure robustness and generalization of the models.

**5. Provide insights into the effectiveness of different classification models:**

- Analyze and interpret the results obtained from evaluating each classification model.
- Identify the strengths and weaknesses of each model in accurately categorizing Telugu documents into predefined topics.
- Provide recommendations for selecting the most suitable classification model based on performance metrics and specific use cases.

### 1.3 Project Scope

1. **Text Classification Methodology**: The project aims to develop and implement a text classification methodology tailored specifically for Telugu documents. This methodology will involve various stages such as data collection, preprocessing, feature extraction, model training, and evaluation.
2. **Focus on Telugu News Articles**: The primary focus of the project is to categorize Telugu news articles into predefined topics such as business, editorial, national, sports, and entertainment. This will involve collecting a dataset of Telugu news articles and annotating them with appropriate topic labels.
3. **Utilization of Modern ML Techniques**: The project will utilize modern machine learning techniques including Naïve Bayes, Support Vector Machine (SVM), and Logistic Regression, backed by Natural Language Processing (NLP) approaches. These techniques will be employed for feature extraction, model training, and classification.
4. **Evaluation of Model Performance**: The developed classification models will be evaluated using standard evaluation metrics such as accuracy, F1-score, precision, and recall. Cross-validation techniques may also be employed to ensure robustness and generalization of the models.

### Limitations:

1. **Limited Availability of Labeled Datasets**: One of the primary limitations is the scarcity of labeled Telugu news datasets. The availability of diverse and sufficiently large datasets may affect the performance and generalization capability of the classification models.

2. **Challenges in Text Preprocessing**: Telugu text presents challenges in accurate tokenization and preprocessing due to its complex morphology and script. Developing effective preprocessing techniques that handle issues such as word segmentation and morphological variations is crucial but may be challenging.
3. **Model Performance Variability**: The performance of classification models may vary depending on factors such as the quality of input features, the size of the dataset, and the tuning of hyperparameters. Achieving optimal performance across different topics and datasets may be challenging.
4. **Focus Solely on Text Classification**: The project focuses solely on text classification tasks and does not address other aspects of natural language understanding such as sentiment analysis or text generation. Therefore, the insights and solutions provided by the project are limited to the scope of text classification.

# CHAPTER 2

# BACKGROUND WORK

# CHAPTER 2

# BACKGROUND WORK

## LITERATURE WORK

Deepu et al., n.d. [1] proposed a rule-based approach for opinion classification of Malayalam motion picture audits, tending to challenges stemming from client input containing spelling botches. Sahu et al. [2] focused on classifying Odia movie reviews using supervised classification techniques. Moreover, J.Sultana et al., [3] compared conventional Profound Learning and ML approaches for opinion expectation on instructive information, finding MLP to abdicate the finest results. S.S. Mukku et al., [4] displayed a system for Telugu opinion investigation utilizing Doc2Vec models prepared with different ML procedures At long last, D Naga et al., [5] examined n-gram include selection's effect on news article content classification utilizing semi-supervised learning strategies, highlighting SVM's prevalence. The creators of the current think about propose to analyze Telugu news assumptions utilizing machine learning strategies to address the require for assumption investigation in Telugu news.

Kamal Sarkar et al., [6] and colleagues did some study about feeling analysis using a multinomial Naïve Bayes classifier enhanced with fancy determination characteristics. They focused on looking at feelings in tweets written in Bengali and Hindi, showing how their method can work in different languages. The Multinomial Naïve Bayes thing is great for handling jobs about text figures. The determination chat included deeper feeling analysis, underlining how smart their process is. The work by Sarkar and the gang gives us some good ideas for studying feelings in various languages. This study is super important in the international social media world because looking at feelings helps in knowing what people think across different languages.

Reddy Naidu et al., [7] presented a novel two-phase estimation research approach in their sentiment analysis investigation for Telugu e-News. This technique uses Telugu SentiWordNet, a lexical resource specific to the language, to categorize phrases found in Telugu e-News articles. The authors' use of a two-phase process points to a sophisticated and specialized method for handling the complexities of sentiment analysis in Telugu. Naidu et al. substantially advance sentiment analysis research by concentrating on regional language subtleties and offering a system that works with Telugu e-news information.

Their research contributes to our knowledge of sentiment dynamics in regional languages and may help tailor sentiment analysis techniques to a different type of linguistic circumstances. There is potential for improving sentiment analysis methods through more investigation of their methodology.

## 2.1 Rule Based Approach

### 2.1.1 Introduction .

Deepu et al., n.d. [1] proposed a rule-based approach for opinion classification of Malayalam motion picture audits, tending to challenges stemming from client input containing spelling botches. Sentiment analysis, the computational study of analyzing people's opinions and attitudes from written text, has gained significant attention due to the proliferation of user-generated content on social media platforms. However, analyzing sentiment in languages with complex structures and high variability, such as Malayalam, presents unique challenges, particularly when dealing with user feedback containing spelling errors.In this context, Deepu et al. (n.d.) proposed a novel approach to address these challenges specifically within the domain of Malayalam movie reviews. Their research focused on developing a rule-based method tailored to the linguistic nuances of Malayalam, aiming to accurately classify opinions despite variations in language usage and spelling errors commonly found in informal text data.

Their approach involved several key steps, including preprocessing the text to handle spelling mistakes, extracting relevant linguistic features, and developing a set of rules based on linguistic patterns specific to Malayalam. These rules were then applied to automatically classify reviews as positive, negative, or neutral sentiments.

Evaluation of their method likely encompassed standard metrics to assess its performance, such as accuracy, precision, recall, and F1-score. Additionally, the study discussed the effectiveness and limitations of the approach, providing valuable insights for future research in sentiment analysis, particularly in languages with similar challenges.

### 2.1.2 Merits , Demerits and Challenges Merits

1. **Handling Spelling Errors:** The approach addresses a common challenge in sentiment analysis by preprocessing the text to handle spelling mistakes, improving the robustness of the classification process.
2. **Tailored Approach:** Deepu et al.'s rule-based method is specifically designed for the Malayalam language, allowing for more accurate sentiment analysis in movie reviews written in this language.Linguistic Patterns. By leveraging linguistic patterns specific to Malayalam, the method captures the nuances of the language, leading to more precise sentiment classification.
3. **Automated Classification:** The rule-based approach enables automated classification of movie reviews into positive, negative, or neutral sentiments, saving time and effort compared to manual labeling.
4. **Linguistic Patterns:** By leveraging linguistic patterns specific to Malayalam, the method captures the nuances of the language, leading to more precise sentiment classification.

**Demerits**

1. **Limited Generalizability:** The rule-based approach may have limited generalizability to other languages or domains outside of Malayalam movie reviews, as it heavily relies on linguistic patterns specific to this context.
2. **Dependency on Rules:** The effectiveness of the method heavily depends on the quality and coverage of the rules developed. Inadequate rules may lead to inaccurate classification results.
3. **Subjectivity:** Sentiment analysis is inherently subjective, and the accuracy of the classification may vary based on individual interpretations of the text, which may not always align with the intended sentiment.

**Challenges**

1. **Data Availability:** Obtaining sufficient annotated data for training and evaluating the sentiment analysis model, especially in languages with limited resources like Malayalam, can be challenging.

2. **Scalability:** Adapting the rule-based approach to handle large volumes of data and scaling it to accommodate evolving linguistic patterns and user behaviors over time presents scalability challenges.
3. **Evaluation Metrics:** Selecting appropriate evaluation metrics to assess the performance of the sentiment analysis model, considering the nuances of sentiment classification and language-specific characteristics, requires careful consideration.

### 2.1.3 Implementation

1. **Data Collection:** Gather a dataset of Malayalam movie reviews from various sources such as movie review websites, social media platforms, or online forums. Ensure the dataset is representative and covers a diverse range of sentiments.
2. **Preprocessing:** Preprocess the text data to handle spelling errors, tokenization, and normalization. This step may involve using spell-checking algorithms, word segmentation tools, and text normalization techniques specific to the Malayalam language.
3. **Feature Extraction:** Extract relevant linguistic features from the preprocessed text data. Features may include sentiment-bearing words, phrases, grammatical structures, and contextual cues indicative of positive or negative sentiment.
4. **Rule Set Development:** Develop a set of rules based on linguistic patterns and domain-specific knowledge relevant to Malayalam movie reviews. These rules should capture the sentiment indicators, key phrases, and linguistic cues associated with positive, negative, or neutral opinions.
5. **Opinion Classification:** Apply the developed rule set to automatically classify movie reviews into positive, negative, or neutral sentiments. Implement algorithms to match the extracted features with the predefined rules and assign sentiment labels accordingly.
6. **Evaluation:** Evaluate the performance of the sentiment analysis model using standard metrics such as accuracy, precision, recall, and F1-score. Compare the model's predictions against manually annotated data or ground truth labels to assess its effectiveness.
7. **Analysis and Refinement:** Analyze the effectiveness and limitations of the approach based on evaluation results. Identify areas for improvement and refinement in the rule set

or feature extraction techniques. Iterate on the implementation to enhance the accuracy and robustness of the sentiment analysis method.

### 2.1.4. Block Diagram

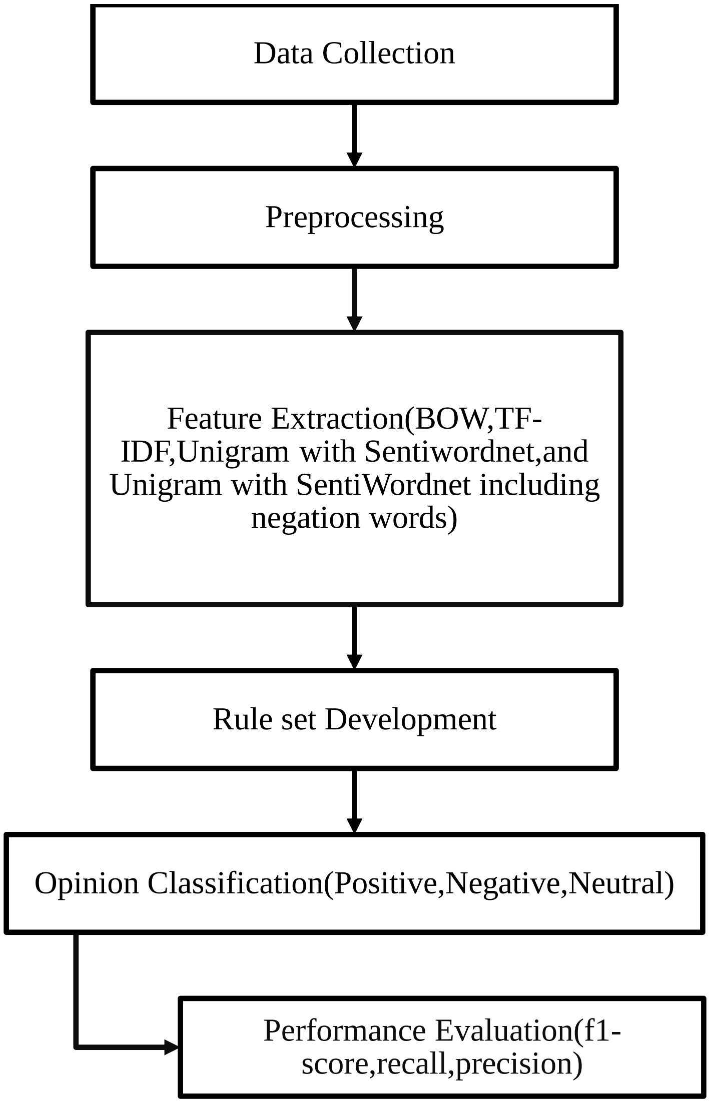

Data Collection
Preprocessing
Feature Extraction(BOW,TF-IDF,Unigram with Sentiwordnet,and Unigram with SentiWordnet including negation words)
Rule set Development
Opinion Classification(Positive,Negative,Neutral)
Performance Evaluation(f1-score,recall,precision)

**Fig 2.1.4.1 Block Diagram of Rule Based Approach**

## 2.2 Supervised Learning

### 2.2.1 Introduction

Sahu et al. [2] focused on classifying Odia movie reviews using supervised classification techniques. Supervised learning is a branch of machine learning where algorithms learn patterns and relationships from labeled training data. These algorithms can then make predictions or decisions without human intervention based on the learned patterns. In the context of their study, Sahu et al. applied supervised classification techniques to analyze and categorize Odia movie reviews. By providing labeled examples of movie reviews, the algorithms were trained to recognize patterns in the text data associated with different sentiment categories, such as positive, negative, or neutral.

The fundamental concept behind supervised learning is that the algorithm learns to map input features (such as words or phrases in the movie reviews) to the corresponding output labels (sentiment categories) based on the provided training data. Once trained, the model can generalize these learned patterns to make predictions on new, unseen data, allowing for automated sentiment analysis of Odia movie reviews without the need for human intervention.

### 2.2.2 Merits, Demerits and Challenges

**Merits**

1. **Language-specific Analysis:** Focusing on Odia movie reviews allows for sentiment analysis tailored to a specific language, catering to the linguistic nuances and cultural context of Odia speakers.
2. **Cultural Relevance:** By analyzing sentiments expressed in Odia movie reviews, the project contributes to a deeper understanding of cultural preferences, perceptions, and attitudes towards cinema within the Odia-speaking community.
3. **Supervised Learning:** Utilizing supervised classification techniques enables the algorithm to learn from labeled data, potentially leading to more accurate sentiment classification by leveraging existing patterns and relationships in the data.
4. **Automated Analysis:** The project facilitates automated sentiment analysis of Odia movie reviews, providing a scalable solution for processing large volumes of text data without the need for manual annotation.

**Demerits:**

1. **Limited Dataset:** Availability of a sufficient and diverse dataset of labeled Odia movie reviews may be a challenge, potentially limiting the effectiveness of the supervised learning approach due to inadequate training data.
2. **Subjectivity in Labeling:** The process of labeling movie reviews with sentiment categories may involve subjective judgments, leading to potential inconsistencies or inaccuracies in the labeled dataset.
3. **Language Complexity:** Odia, like many languages, may exhibit complex linguistic structures and expressions that pose challenges for natural language processing tasks such as sentiment analysis, potentially affecting the accuracy of the classification model.

**Challenges:**

1. **Data Collection:** Acquiring a sufficient amount of labeled Odia movie reviews for training the classification model may be challenging due to limited availability of annotated datasets.
2. **Feature Representation:** Selecting and representing relevant features from the text data, especially in a language-specific context like Odia, requires careful consideration to capture meaningful linguistic patterns and sentiment indicators.
3. **Model Evaluation:** Assessing the performance of the classification model on Odia movie reviews involves selecting appropriate evaluation metrics and conducting rigorous testing to ensure the model's accuracy and reliability.

### 2.2.2 Implementation

1. **Data Collection:** Obtain a dataset of Odia movie reviews from reliable sources. This dataset should include a sufficient number of reviews labeled with sentiment categories (positive, negative, or neutral).
2. **Data Preprocessing:** Clean and preprocess the text data to remove noise and irrelevant information. This may involve tasks such as tokenization, lowercasing, removing stopwords, and stemming or lemmatization.

3. **Feature Extraction:** Extract relevant features from the preprocessed text data. Common techniques include bag-of-words representation, TF-IDF (Term Frequency-Inverse Document Frequency), and word embeddings like Word2Vec or GloVe.
4. **Splitting the Dataset:** Divide the dataset into training and testing sets. The training set will be used to train the classification model, while the testing set will be used to evaluate its performance.
5. **Model Selection**: In their study, Sahu et al. implemented supervised learning classifiers for sentiment analysis of Odia movie reviews. The classifiers they used included:
   i. **Logistic Regression**
   ii. **Naive Bayes**
   iii. **Support Vector Machine (SVM):**
6. **Model Training:** Train the selected classification model using the training data. The model will learn to associate the extracted features with the corresponding sentiment labels.

### 2.2.4 Block Diagram

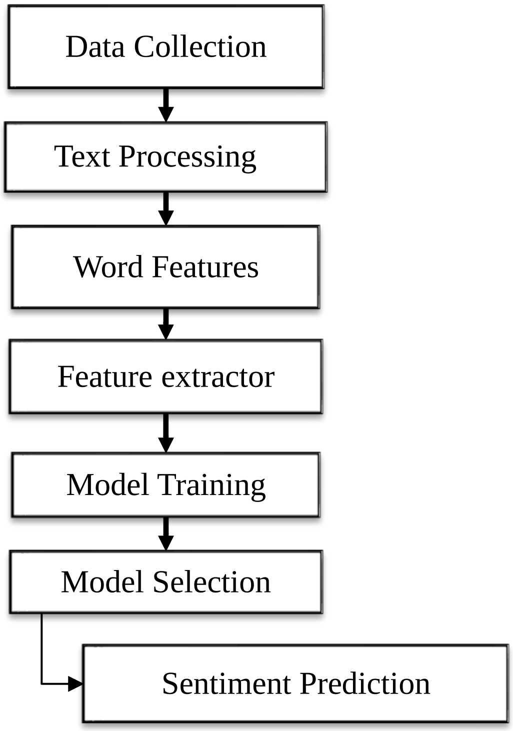


**fig 2.2.4.1 Block Diagram of Supervised Learning**

## 2.3. Deep Learning

### 2.3.1 Introduction

J.Sultana et al., [3] compared conventional Profound Learning and ML approaches for opinion expectation on instructive information, finding MLP to abdicate the finest results. In their studies conducted in 2020, Jabeen Sultana and her team delved into the realm of sentiment analysis of educational tweets, recognizing the increasing significance of social media platforms in educational discourse. With the advent of digital communication, social media has become a prominent space for educators, students, policymakers, and researchers to engage in discussions, share resources, and express opinions. Within this context, understanding the sentiment conveyed in educational tweets has emerged as a crucial area of study.

Traditional sentiment analysis techniques often struggle to accurately capture the nuances present in educational tweets due to their informal language, varied expressions, and contextual complexities. Recognizing this limitation, the studies conducted by Sultana and her team sought to leverage deep learning methods as a promising approach to unraveling sentiment dynamics within educational tweets. Deep learning techniques, known for their ability to automatically learn intricate patterns and representations from data, offer a powerful means to analyze and classify sentiments in textual data.

By focusing on sentiment analysis of educational tweets, Sultana et al. aimed to address the need for more nuanced insights into the attitudes, perceptions, and sentiments prevalent within the educational community on social media platforms. These insights hold significant implications for educators, policymakers, and researchers, as they can inform decision-making processes, facilitate targeted interventions, and foster a more responsive educational environment. Through their research, Sultana and her team sought to contribute towards advancing the understanding of sentiment analysis in educational contexts, particularly in the digital age where social media platforms play a pivotal role in shaping educational discourse.

### 2.3.2 Merits,Demerits and Challenges

**Merits**

1. **Accurate Classification:** Deep learning methods have shown promising results in accurately classifying sentiment in text data, including tweets. Their ability to capture complex patterns and relationships within the data can lead to more precise sentiment analysis results.
2. **Handling Nuanced Language:** Educational tweets often contain nuanced language and context-specific terminology. Deep learning models, with their ability to learn representations from data, can effectively handle these nuances, resulting in more nuanced sentiment analysis.
3. **Scalability:** Deep learning methods are highly scalable, making them suitable for analyzing large volumes of educational tweets. They can efficiently process and classify vast amounts of data, enabling comprehensive sentiment analysis across diverse educational topics and trends.
4. **Adaptability:** Deep learning models can adapt to changing linguistic patterns and user behaviors over time. This adaptability allows them to maintain performance in dynamic environments, ensuring robust sentiment analysis even as language use evolves.

**Demerits**

1. **Data Intensity:** Deep learning methods often require large amounts of labeled data for training, which may be challenging to obtain in the context of sentiment analysis of educational tweets. The process of manually labeling tweets can be time-consuming and resource-intensive.
2. **Overfitting:** Deep learning models, particularly complex ones, are susceptible to overfitting, where the model learns to memorize the training data rather than generalize patterns. Overfitting can lead to poor performance on unseen data and reduced generalization ability.
3. **Interpretability:** Deep learning models are often regarded as black boxes, making it challenging to interpret how they arrive at their predictions. Understanding the reasoning behind the model's sentiment classification decisions may be difficult, limiting the interpretability of the results.

**Challenges**

1. **Informal Language:** Educational tweets often contain informal language, abbreviations, and slang, which can pose challenges for sentiment analysis. Deciphering the sentiment conveyed in such tweets accurately requires robust natural language understanding capabilities.
2. **Class Imbalance:** The distribution of sentiment classes in educational tweets may be imbalanced, with one class dominating over others. Class imbalance can affect the model's ability to learn effectively, leading to biased predictions towards the majority class.
3. **Domain Specificity:** Educational discourse encompasses a wide range of topics and domains, each with its unique language and sentiment characteristics. Adapting deep learning models to different educational contexts and domains while maintaining performance can be challenging.
4. **Ethical Considerations:** Analyzing sentiment in educational tweets raises ethical considerations regarding user privacy, consent, and data usage. Ensuring compliance with ethical guidelines and regulations is essential to protect user rights and maintain trust in the sentiment analysis process.

### 2.3.3 Implementation

1. **Data Collection:** Gather a dataset of educational tweets from various sources, including Twitter, educational forums, and relevant online platforms. Ensure the dataset is diverse and representative of different educational topics and sentiments.
2. **Preprocessing**: Preprocess the text data to remove noise, such as special characters, emojis, and URLs. Tokenize the text into words and apply stemming or lemmatization to normalize the text. Additionally, remove stopwords and handle spelling errors specific to the language used in educational tweets.
3. **Feature Extraction:** Extract relevant features from the preprocessed text data, such as word embeddings or contextual representations using pre-trained deep learning models like Word2Vec, GloVe, or BERT. These features capture semantic information and contextual relationships among words, facilitating sentiment analysis.
4. **Model Development:** Build a deep learning model for sentiment analysis, such as a

recurrent neural network (RNN), long short-term memory (LSTM) network, or convolutional neural network (CNN). Design the model architecture to take input features and predict the sentiment label (positive, negative, or neutral).

5. **Training:** Split the dataset into training, validation, and testing sets.Train the deep learning model on the training data using techniques like stochastic gradient descent (SGD) or Adam optimizer. Tune hyperparameters such as learning rate, batch size, and number of epochs to optimize model performance.
6. **Evaluation:** Evaluate the trained model's performance on the validation set using metrics such as accuracy, precision, recall, and F1-score. Fine-tune the model based on validation results to improve its accuracy and generalization ability.
7. **Testing:** Evaluate the final model on the testing set to assess its performance on unseen data. Analyze the confusion matrix and visualize sentiment predictions to gain insights into model behavior and performance.

### 2.3.4 Block Diagram

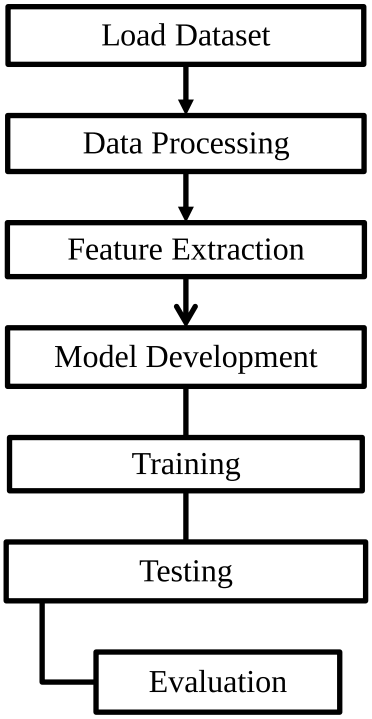


**fig 2.2.4.1 Block Diagram of Deep Learning**

# CHAPTER 3
# PROPOSED SYSTEM

# CHAPTER 3

# PROPOSED SYSTEM

## 3.1 OBJECTIVE OF PROPOSED MODEL

The main objective of Telugu text classification is essential for efficient information management and accessibility within the Telugu-speaking community. Its primary objective lies in organizing and categorizing Telugu language content into relevant topics such as nation, entertainment, sports, business, and more. By employing advanced machine learning and natural language processing techniques, this classification system aims to streamline data retrieval, enabling users to quickly find information of interest. Moreover, it facilitates targeted content recommendations, enhances search functionalities, and supports various applications tailored to the Telugu language domain, ultimately improving user experience and promoting effective communication and utilization of Telugu textual data.

## 3.2 ALGORITHMS USED IN PROPOSED METHODOLOGY

The algorithms used for Telugu data classification are Naïve Bayes, Logistic Regression and Support Vector Machine (SVM).

**Multinomial Naive Bayes**

Naive Bayes, a classification algorithm rooted in Bayes' theorem, assumes feature independence to compute class probabilities based on input features. It estimates the likelihood of observing features given each class, multiplied by prior probabilities of the classes. However, encountering unseen feature values in test data can lead to zero probabilities, undermining prediction reliability. To counteract this, Laplace smoothing is used, augmenting feature counts with a small constant to prevent zero probabilities. This adjustment ensures smoother probability estimates, enhancing model robustness and enabling better generalization to unseen data. Consequently, Laplace smoothing is pivotal in refining Naive Bayes' predictive performance and adaptability across various classification tasks.

**Multinomial Logistic Regression**

Multinomial logistic regression is a statistical method used to predict the outcome of a categorical dependent variable with more than two levels. Unlike binary logistic regression which predicts only two possible outcomes, multinomial logistic regression can handle multiple outcome categories. It extends logistic regression by modeling the logarithm of the odds that a particular outcome belongs to a particular category, given the values of the predictor variables. In multinomial logistic regression, probabilities are estimated for each category of the dependent variable, and the category with the highest probability is chosen as the predicted output for a given set of predictor variables. This is typically achieved through the SoftMax function, which normalizes the logits (log-odds) into probabilities, ensuring that the probabilities sum up to 1 across all categories. The output category with the highest probability is then selected as the predicted class.

**Support Vector Machine (SVM)**

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that best separates the data points into different classes in a high-dimensional space. SVM aims to maximize the margin, which is the distance between the hyperplane and the nearest data points (support vectors) from each class. By maximizing the margin, SVM not only effectively classifies the training data but also generalizes well to unseen data, reducing the risk of overfitting. SVM can handle linearly separable as well as non-linearly separable data by using different kernel functions, such as linear, polynomial, radial basis function (RBF), or sigmoid kernels, which map the input data into higher-dimensional feature spaces where the data might become linearly separable. SVM is widely used in various fields, including text classification, image recognition, and bioinformatics, owing to its flexibility, effectiveness, and ability to handle high-dimensional data.

## 3.3 DESIGNING

### 3.3.1 UML DIAGRAM

#### A. Class Diagram

A class diagram is a visual representation of the structure and relationships of classes within a system, often used in software engineering to model the static structure of object-oriented systems. It depicts the classes in the system, along with their attributes, methods, and associations between classes. Each class represents a distinct entity or concept in the system, encapsulating its data and behavior. Attributes represent the state or properties of a class, while methods represent the operations or behaviors that the class can perform. Associations between classes illustrate how classes are related to each other, such as composition, aggregation, or inheritance. Class diagrams serve as a blueprint for designing and understanding the architecture of a software system, aiding in communication between stakeholders and guiding the implementation process.

- The name is derived from the first / most prominent component.
- The attributes of the class are included in the pivot.
- At the bottom, the techniques and actions that the class may use or reject are shown.

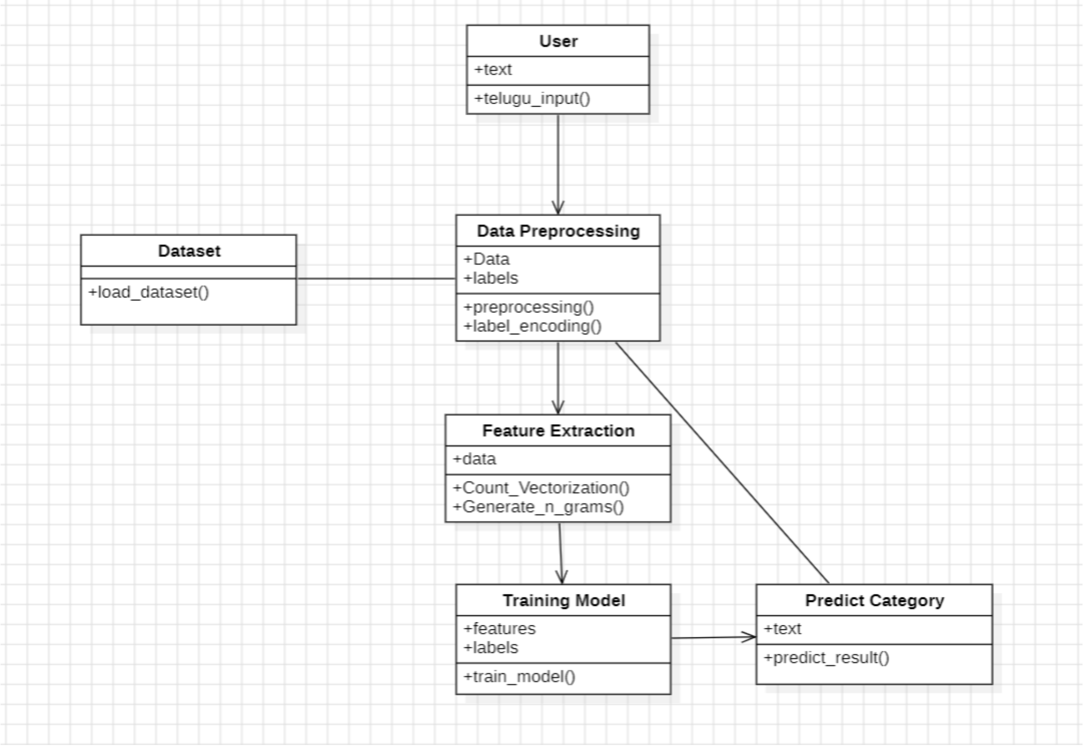


**Fig 3.3.1.1: Class Diagram**

## B. Use Case Diagram

A use case diagram is a graphical representation that illustrates the interactions between users (actors) and a system to achieve specific goals or tasks. It showcases the various ways users interact with the system, typically focusing on the system's functionalities from a user's perspective. Actors are external entities, such as users or other systems, that interact with the system. Use cases represent specific tasks or functionalities the system provides to the actors. Arrows indicate the direction of communication between actors and use cases. Use case diagrams help stakeholders understand the system's behavior and requirements in a straightforward and visual manner, aiding in communication and requirements analysis during the software development process.

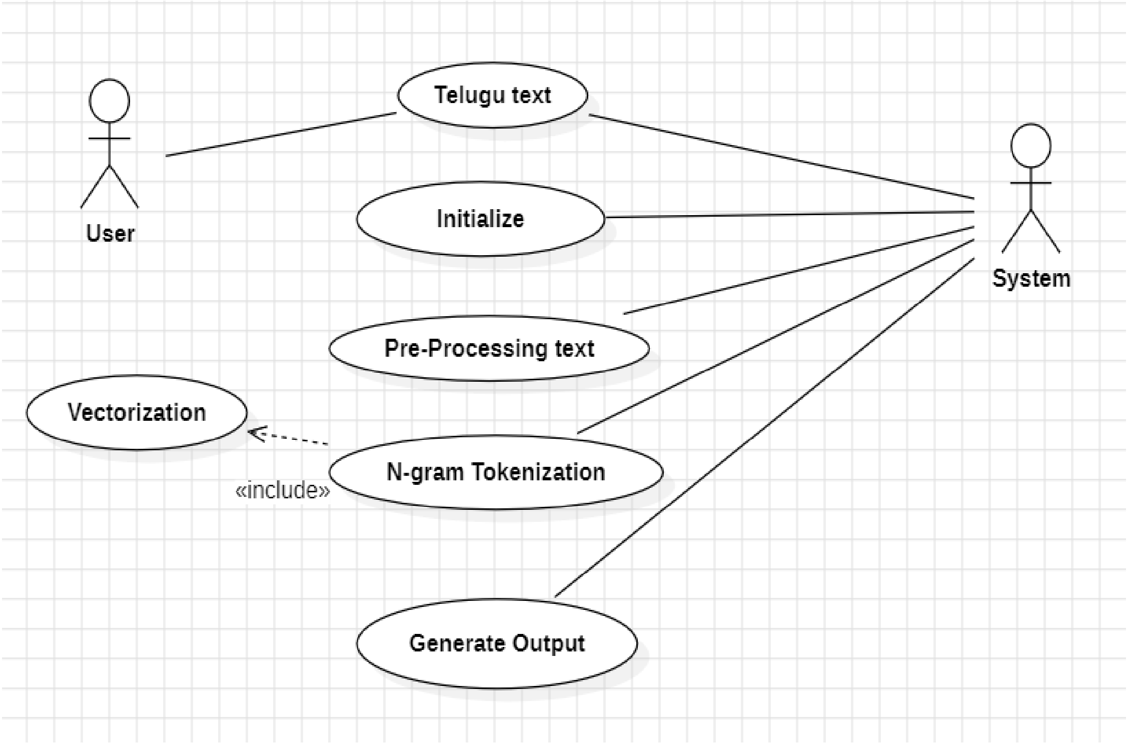


**Fig 3.3.1.2: Use Case Diagram**

## C. Sequential Diagram

A definition of a relationship that shows how the processes work, and in which case, is called a sequence graph. The collection graph shows the combination of items arranged in chronological order. It shows the objects and categories associated with the situation, as well as the collection of symbols that are exchanged between texts that are thought to make everything right. Sequential diagrams are usually related to the recognition of cases of ongoing work in the Functional View system. Follow-up form is sometimes called event graphs, specific scenarios, and time charts.

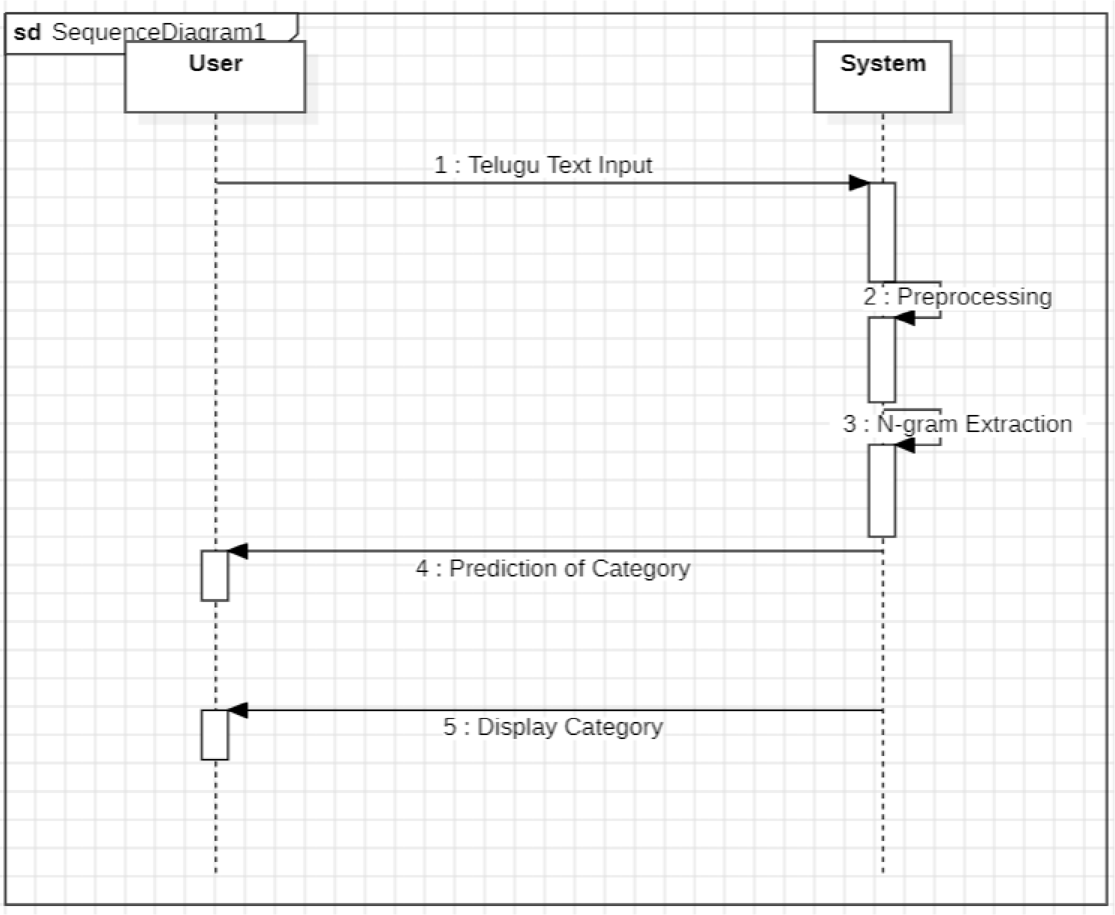


**Fig 3.3.1.3: Sequential Diagram**

### D. Collaboration Diagram

A collaboration diagram organizes the relationships between several entities. There are numbered interactions to help keep track of what's going on, making it easy to see how things progress. It is possible to identify all potential connections between entities with the help of the cooperation diagram.

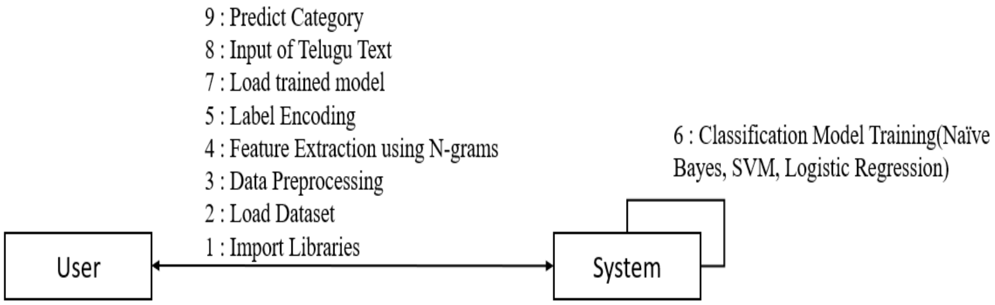


**Fig 3.3.1.4: Collaboration Diagram**

### E. Activity Diagram

The proposed system's behavior is shown in terms of activities in an activity diagram. Activities are the parts of a model that represent a series of operations. Other actions, the availability of things, or external occurrences may all serve as triggers for an activity.

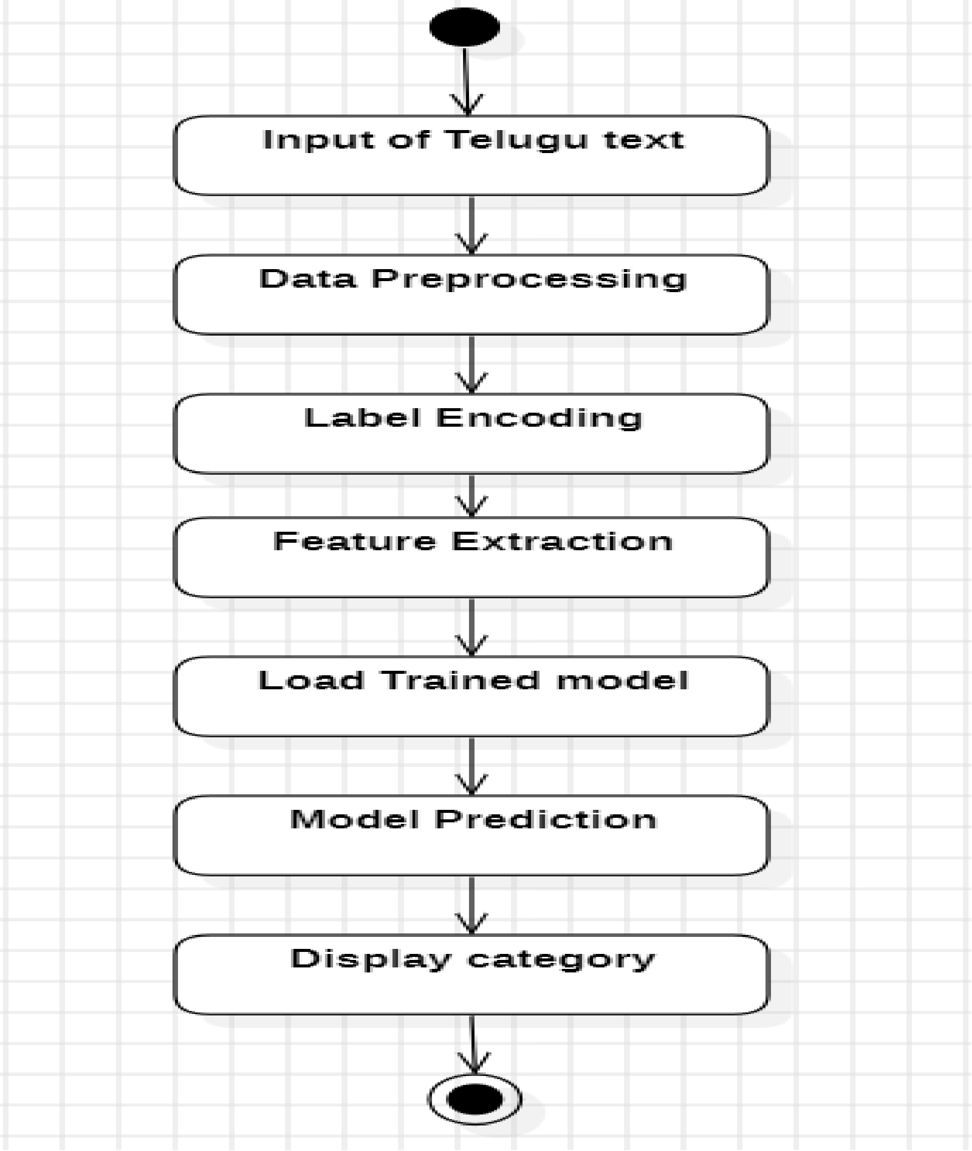


**Fig 3.3.1.5: Activity Diagram**

### F. Data Flow Diagram

DFD diagrams are used to depict the flow of data inside a system. Process, details, external dimensions, structure, and digital data are all included in this process and their corresponding elements and components. Using a DFD, you can see the flow of data through the system and how it is monitored at each step. As data moves from integration to harvesting, a visual framework depicts the progression of information dissemination and advancement. To address the structure, you may employ DFD at any level of consideration.

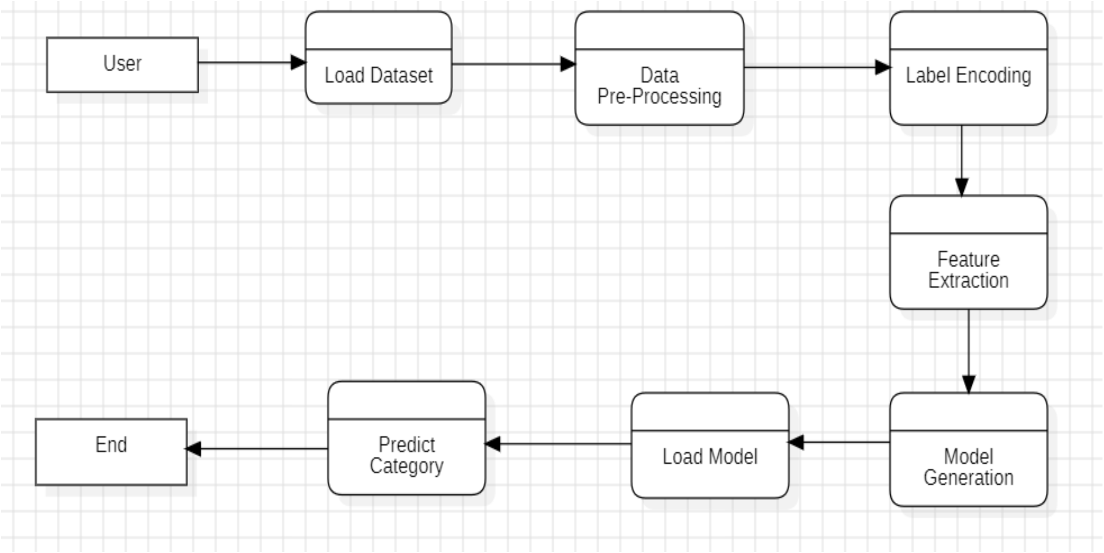
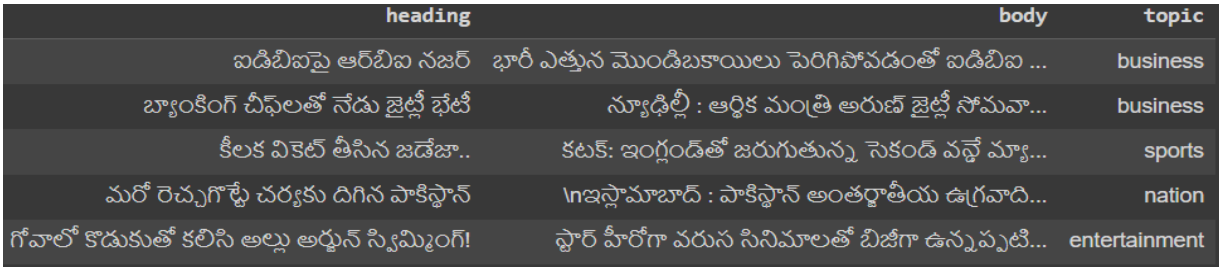


**Fig 3.3.1.6: Data Flow Diagram**

## 3.4 STEPWISE IMPLEMENTATION AND CODE

**DATASET:**

Text classification is a crucial aspect and influences strongly on social practices. Most of the news articles generated in Telugu have not received much attention in the sentiment analysis community. These reasons led to choose the news dataset. The Telugu News dataset was collected by Kaggle (SRK, 2020). This Telugu news statements belonging to five areas and class labels used (Business, Editorial, National, Sports, Entertainment).



| heading | body | topic |
|---|---|---|
| ఐడిబిఐపై ఆర్‌బిఐ నజర్ | భారీ ఎత్తున మొండిబకాయిలు పెరిగిపోవడంతో ఐడిబిఐ ... | business |
| బ్యాంకింగ్ చీఫ్‌లతో నేడు జైట్లీ భేటీ | న్యూఢిల్లీ : ఆర్థిక మంత్రి అరుణ్ జైట్లీ సోమవా... | business |
| కీలక వికెట్ తీసిన జడేజా.. | కటక్: ఇంగ్లండ్‌తో జరుగుతున్న సెకండ్ వన్డే మ్యా... | sports |
| మరో రెచ్చగొట్టే చర్యకు దిగిన పాకిస్థాన్ | \nఇస్లామాబాద్ : పాకిస్థాన్ అంతర్జాతీయ ఉగ్రవాది... | nation |
| గోవాలో కొడుకుతో కలిసి అల్లు అర్జున్ స్విమ్మింగ్! | స్టార్ హీరోగా వరుస సినిమాలతో బిజీగా ఉన్నప్పటి... | entertainment |

**Fig 3.4.1 Some samples of Telugu news dataset**

**PRE-PROCESSING:**

Pre-Processing step is the first crucial step for removing the cleaning the data so that model can understand the effectively to yield a high-performance result. The pre-processing steps includes

- Removes special characters and punctuations.
- Sklearn: It is Label Encoder assigns numbered values to categories.

**FEATURE EXTRACTION**

The collected preprocessing data undergoes N-gram analysis, encompassing uni-gram, bigram, tri-gram, 4-gram, and 5-gram generation, employing Laplace smoothing. Subsequently, count vectorization is applied to the N-grams to evaluate their efficiency in classification tasks. Through analysis, it is determined that 2-gram generation yields superior performance compared to other N-grams. Therefore, for subsequent model training processes, 2-gram count vectorization is selected. This approach enhances the effectiveness of feature representation, consequently improving the overall performance of the classification model.

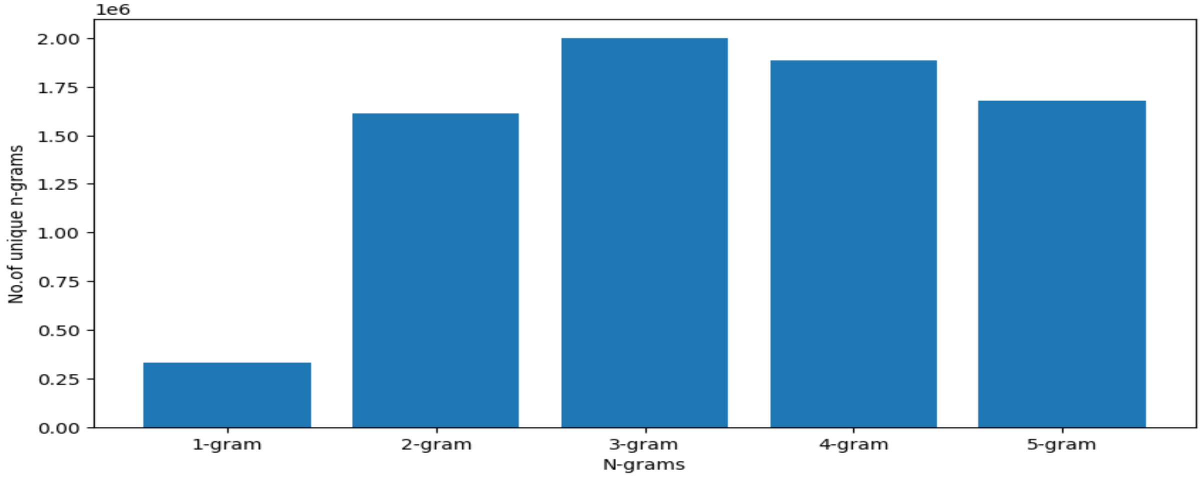


**Fig 3.4.2 Number unique N- grams**

**CLASSIFICATION MODELS TRAINING**

The 2-gram count vectors extracted during feature extraction serve as the input for training three different classification models: Naïve Bayes, Logistic Regression, and Support Vector Machine (SVM). By utilizing these count vectors, each model learns to distinguish patterns and relationships within the Telugu textual data represented by 2-gram features.

**PREDICTION**

The trained model is loaded and predictions of the category based on the Telugu text provided by the user. The prediction are made by the trained classification models – Naïve Bayes, Support Vector Machine (SVM) and Logistic Regression.

**CODE:**

- Importing Libraries

  ```
  import nltk
  import pandas as pd
  import matplotlib.pyplot as plt
  import numpy as np
  import re
  import string
  from sklearn.model_selection import train_test_split
  import seaborn as sns
  import os
  ```

- **Loading Dataset**

  ```
  train_path = "/content/drive/MyDrive/train_telugu_news.csv"
  telugu_news_df = pd.read_csv(train_path)
  telugu_news_df
  ```

- **Downloading indic-nlp-library**

  ```
  pip install indic-nlp-library
  ```

- **Pre-Processing**

  ```
  telugu_news_df.isna().sum()
  telugu_news_df[telugu_news_df["heading"].isna() == True]
  del telugu_news_df["heading"]
  del telugu_news_df["SNo"]
  ```

```
telugu_news_df["topic"].unique()
topic_dic = {}
c = 0
for un in telugu_news_df["topic"].unique():
    if un not in topic_dic:
        topic_dic[un] = c
        c += 1
topic_dic
inv_topic_dict = {v: k for k, v in topic_dic.items()}
def func_topic(s):
    return topic_dic[s]
telugu_news_df["topic"] = telugu_news_df["topic"].apply(func_topic)
date_df = telugu_news_df["date"]
del telugu_news_df["date"]
import sys
from indicnlp import common

# The path to the local git repo for Indic NLP library
INDIC_NLP_LIB_HOME=r"indic_nlp_library"

# The path to the local git repo for Indic NLP Resources
INDIC_NLP_RESOURCES=r"indic_nlp_resources"

# Add library to Python path
sys.path.append(r'{}\src'.format(INDIC_NLP_LIB_HOME))

# Set environment variable for resources folder
common.set_resources_path(INDIC_NLP_RESOURCES)
```

```
from indicnlp.tokenize import sentence_tokenize

indic_string = telugu_news_df["body"][0]
# Split the sentence, language code "hi" is passed for hingi
sentences=sentence_tokenize.sentence_split(indic_string, lang='te')
# print the sentences
for t in sentences:
    print(t)
telugu_news_df["body_processed"] = telugu_news_df["body"].str.replace('\u200c', '')
telugu_news_df["body_processed"] = telugu_news_df["body_processed"].str.replace('\n',
'')
telugu_news_df["body_processed"] = telugu_news_df["body_processed"].str.replace('\t',
'')
telugu_news_df["body_processed"] =
telugu_news_df["body_processed"].str.replace('\xa0', '')
PUNCT = string.punctuation

def remove_punctuation(text):
    return text.translate(str.maketrans('', '', PUNCT))

from indicnlp.tokenize import sentence_tokenize
tot_telugu_text1 = ""

for t in telugu_news_df["body_processed"]:
  tot_telugu_text1 += t
tot_sentences = sentence_tokenize.sentence_split(tot_telugu_text1, lang='te')
print(len(tot_sentences))
tot_telugu_text = ""
```

```
c = 1
for t in telugu_news_df["body_processed"]:
  tot_telugu_text += t
  c += 1

from indicnlp.tokenize import indic_tokenize
vocab_dic  = {}
tokenized_text = []
heap_arr = []
for t in indic_tokenize.trivial_tokenize(tot_telugu_text):
    tokenized_text.append(t)
    heap_arr.append(len(vocab_dic))

    if t not in vocab_dic:
      vocab_dic[t] = 1

    else:
      vocab_dic[t] += 1
```

- **N-Gram analysis of the entire corpus**

```
from nltk.util import ngrams

bigrams_telugu_vocab = {}

for sen in tot_sentences_proc:
  tokens = indic_tokenize.trivial_tokenize(sen)
  bigram = list(ngrams(tokens, 2))

  for big in bigram:
    if tuple(big) not in bigrams_telugu_vocab:
      bigrams_telugu_vocab[tuple(big)] = 1

    else:
      bigrams_telugu_vocab[tuple(big)] += 1
bigrams_telugu_vocab = {k: v for k, v in sorted(bigrams_telugu_vocab.items(), key=lambda item: item[1], reverse = True)}

trigrams_telugu_vocab = {}

for sen in tot_sentences_proc:
  tokens = indic_tokenize.trivial_tokenize(sen)
  trigram = list(ngrams(tokens, 3))

  for trig in trigram:
    if tuple(trig) not in trigrams_telugu_vocab:
      trigrams_telugu_vocab[tuple(trig)] = 1

    else:
      trigrams_telugu_vocab[tuple(trig)] += 1

trigrams_telugu_vocab = {k: v for k, v in sorted(trigrams_telugu_vocab.items(), key=lambda item: item[1], reverse = True)}
four_grams_telugu_vocab = {}

for sen in tot_sentences_proc:
  tokens = indic_tokenize.trivial_tokenize(sen)
  fourgram = list(ngrams(tokens, 4))
  for fourg in fourgram:
    if tuple(fourg) not in four_grams_telugu_vocab:
      four_grams_telugu_vocab[tuple(fourg)] = 1

    else:
      four_grams_telugu_vocab[tuple(fourg)] += 1
four_grams_telugu_vocab = {k: v for k, v in sorted(four_grams_telugu_vocab.items(), key=lambda item: item[1], reverse = True)}
five_grams_telugu_vocab = {}
for sen in tot_sentences_proc:
  tokens = indic_tokenize.trivial_tokenize(sen)
  fivegram = list(ngrams(tokens, 5))

  for fiveg in fivegram:
    if tuple(fiveg) not in five_grams_telugu_vocab:
      five_grams_telugu_vocab[tuple(fiveg)] = 1

    else:
      five_grams_telugu_vocab[tuple(fiveg)] += 1

five_grams_telugu_vocab = {k: v for k, v in sorted(five_grams_telugu_vocab.items(), key=lambda item: item[1], reverse = True)}
```

- **Using n grams and laplace smoothening**

```
from indicnlp.tokenize import sentence_tokenize
indic_string = telugu_news_df["body"][0]
# Split the sentence, language code "hi" is passed for hingi
sentences=sentence_tokenize.sentence_split(indic_string, lang='te')
for t in sentences:
    print(t)

# function to build a n-gram vocabulary
def build_n_gram_vocab(n, tot_sentences_proc):
  if n > 1:
    n_grams_telugu_vocab = {}
    for sen in tot_sentences_proc:
      tokens = indic_tokenize.trivial_tokenize(sen)
      ngram = list(ngrams(tokens, n))
      for ngm in ngram:
        if tuple(ngm) not in n_grams_telugu_vocab:
          n_grams_telugu_vocab[tuple(ngm)] = 1
        else:
          n_grams_telugu_vocab[tuple(ngm)] += 1
    n_grams_telugu_vocab = {k: v for k, v in sorted(n_grams_telugu_vocab.items(),
key=lambda item: item[1], reverse = True)}
  else:
    n_grams_telugu_vocab = {}
    for sen in tot_sentences_proc:
      tokens = indic_tokenize.trivial_tokenize(sen)
      ngram = list(ngrams(tokens, 1))
      for ngm in ngram:
        if ngm not in n_grams_telugu_vocab:
          n_grams_telugu_vocab[ngm] = 1
        else:
          n_grams_telugu_vocab[ngm] += 1
    n_grams_telugu_vocab = {k: v for k, v in sorted(n_grams_telugu_vocab.items(),
key=lambda item: item[1], reverse = True)}
  return n_grams_telugu_vocab
```

- **Pre-processing of test data**

```
test_path = "/content/drive/MyDrive/test_telugu_news.csv"
test_news_df = pd.read_csv(test_path)
test_news_df.head()
y_test = test_news_df["topic"].apply(func_topic)
test_news_df["body_processed"] = test_news_df["body"].str.replace('\u200c', ")
test_news_df["body_processed"] = test_news_df["body_processed"].str.replace('\n', ")
test_news_df["body_processed"] = test_news_df["body_processed"].str.replace('\t', ")
test_news_df["body_processed"] = test_news_df["body_processed"].str.replace('\xa0', ")
test_news_df["body_processed"] = test_news_df["body_processed"].apply(lambda text: remove_punctuation(text))
categories = [i for i in range(5)]
test_text = []
for t in test_news_df["body_processed"]:
  test_text.append(t)
```

- **Feature Extraction**

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
categories = [i for i in range(5)]
text_topic = []
for i in range(5):
  curr_text = ""

  for text in telugu_news_df[telugu_news_df["topic"] == i]["body_processed"]:
    curr_text += text
    curr_text += " "
  text_topic.append(curr_text)
```

- **Tokenization of Test data**

```
def get_all_vocab(tot_text):
  dic = {}
  for t in indic_tokenize.trivial_tokenize(tot_text):
    if t not in dic:
      dic[t] = 1
    else:
      dic[t] += 1
  return dic

x_train = text_topic
y_train = categories
import regex
from indicnlp.tokenize import indic_tokenize
def custom_analyzer(text):
    words = regex.findall(r'\w{1,}', text) #extract words of at least 2 letters
    for w in words:
        yield w
```

- **Model Training (Naïve Bayes, SVM, Logistic Regression)**

```
import joblib
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

clf = MultinomialNB()
```

```
clf.fit(x_train_features, y_train)
joblib.dump(clf, '/content/drive/MyDrive/naive_bayes.pkl')
lr_classifier = LogisticRegression()
lr_classifier.fit(x_train_features, y_train)
joblib.dump(lr_classifier, '/content/drive/MyDrive/logistic_regression.pkl')
svm_classifier = SVC()
svm_classifier.fit(x_train_features, y_train)
joblib.dump(svm_classifier, '/content/drive/MyDrive/svm_classifier.pkl')
```

- Loading Naïve bayes model

```
# Load the trained Naive Bayes model from Google Drive
c = joblib.load('/content/drive/MyDrive/naive_bayes.pkl')
# Load the trained SVM model from Google Drive
svm = joblib.load('/content/drive/MyDrive/svm_classifier.pkl')
# Load the trained logistic regression model from Google Drive
lr = joblib.load('/content/drive/MyDrive/logistic_regression.pkl')
```

- **Accuracy Results**

```
from sklearn.metrics import classification_report
#Classification Report for Naive Bayes
x_test_features = count_vec.transform(x_test)
y_pred_test = c.predict(x_test_features)
target_names = list(inv_topic_dict.values())
print(classification_report(y_test, y_pred_test, target_names=target_names))
#Classification Report for Logistic Regression
y_pred_test = lr.predict(x_test_features)
print(classification_report(y_test, y_pred_test, target_names=target_names))
#Classification Report for SVM
y_pred_test = svm.predict(x_test_features)
```

```
print(classification_report(y_test, y_pred_test, target_names=target_names))
```

- **Prediction**

```
#Prediction by logistic regression model
test_text = """హీరో శ్రీ విష్ణు, ప్రియదర్శి, రాహుల్ రామకృష్ణ హీరోలుగా.. 'హుషారు'
ఫేమ్ శ్రీ హర్ష కొనుగంటి దర్శకత్వం వహించిన అవుట్ అండ్ అవుట్- ఎంటర్‌టైనర్
'ఓం భీమ్'"""
predicted_class = predict_text_sample(test_text, inv_topic_dict, lr, count_vec)
print("Predicted class is:", predicted_class)

#Prediction by SVM model
test_text = """హీరో శ్రీ విష్ణు, ప్రియదర్శి, రాహుల్ రామకృష్ణ హీరోలుగా.. 'హుషారు'
ఫేమ్ శ్రీ హర్ష కొనుగంటి దర్శకత్వం వహించిన అవుట్ అండ్ అవుట్- ఎంటర్‌టైనర్
'ఓం భీమ్ '"""
predicted_class = predict_text_sample(test_text, inv_topic_dict, svm, count_vec)
print("Predicted class is:", predicted_class)

#Prediction by Naive Bayes
test_text = """హీరో శ్రీ విష్ణు, ప్రియదర్శి, రాహుల్ రామకృష్ణ హీరోలుగా.. 'హుషారు'
ఫేమ్ శ్రీ హర్ష కొనుగంటి దర్శకత్వం వహించిన అవుట్ అండ్ అవుట్- ఎంటర్‌టైనర్
'ఓం భీమ్ '"""
predicted_class = predict_text_sample(test_text, inv_topic_dict, c, count_vec)
print("Predicted class is:", predicted_class)
```

## 3.5 BLOCK DIAGRAM

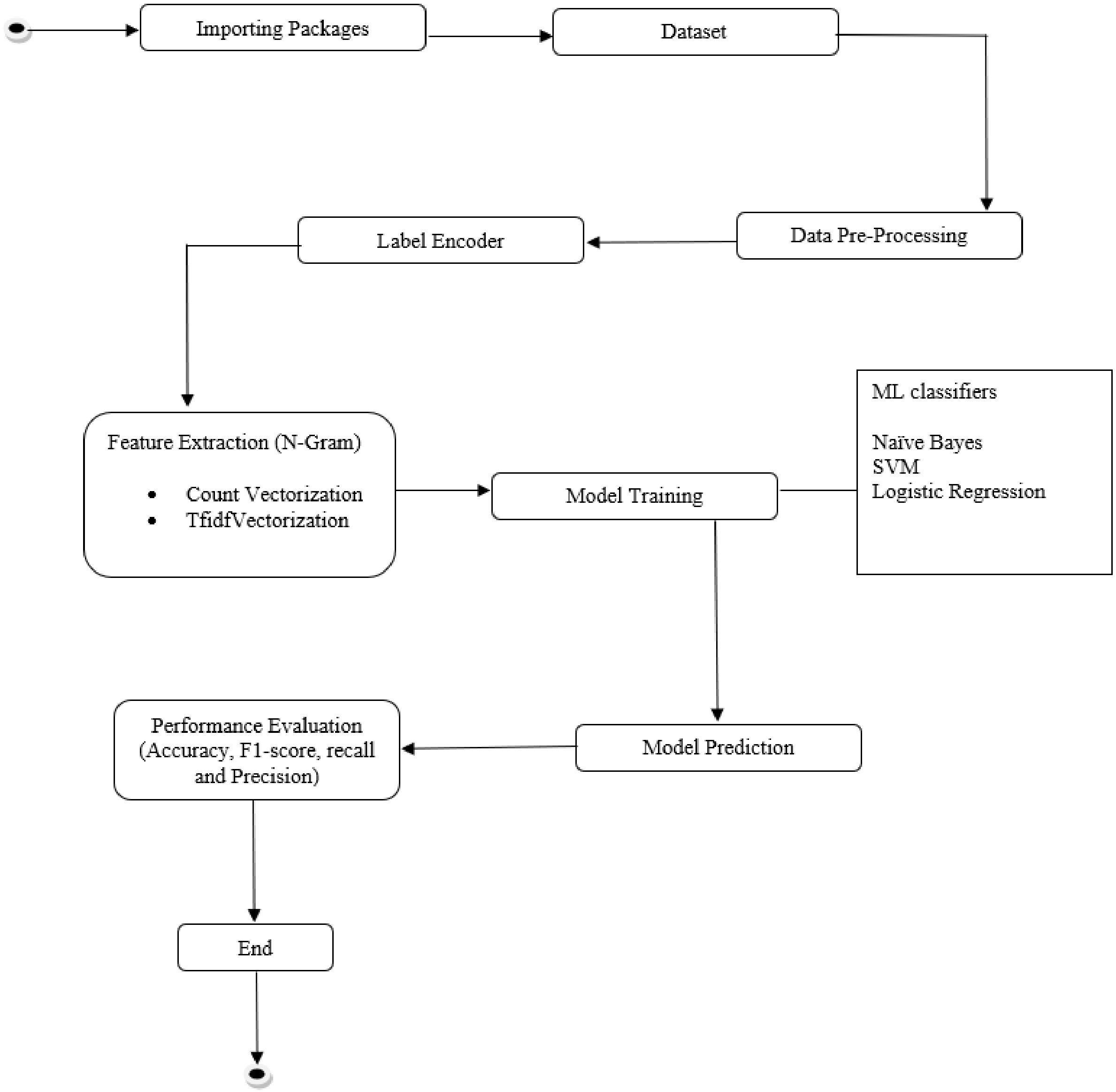


**Fig 3.5.1 Block Diagram of Proposed Methodology**

# CHAPTER 4
# RESULTS AND DISCUSSION

# RESULTS AND DISCUSSION

### Performance Evaluation of Naïve Bayes

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Business | 0.88 | 0.97 | 0.92 | 653 |
| Sports | 0.97 | 0.94 | 0.96 | 437 |
| Nation | 0.95 | 0.90 | 0.93 | 1673 |
| Entertainment | 0.96 | 0.98 | 0.97 | 1289 |
| Editorial | 0.80 | 0.83 | 0.82 | 277 |
| **accuracy** | | | 0.93 | 4329 |
| **macro avg** | 0.91 | 0.92 | 0.92 | 4329 |
| **weighted avg** | 0.94 | 0.93 | 0.93 | 4329 |

**Table 4.1: Performance Evaluation metrics for Naïve Bayes**

### Performance Evaluation of SVM

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Business | 0.89 | 0.83 | 0.86 | 653 |
| Sports | 0.97 | 0.85 | 0.91 | 437 |
| Nation | 0.85 | 0.93 | 0.89 | 1673 |
| Entertainment | 0.93 | 0.94 | 0.93 | 1289 |
| Editorial | 0.90 | 0.73 | 0.81 | 277 |
| **accuracy** | | | 0.88 | 4329 |
| **macro avg** | 0.91 | 0.86 | 0.88 | 4329 |
| **weighted avg** | 0.90 | 0.90 | 0.90 | 4329 |

**Table 4.2: Performance Evaluation metrics for SVM**

### Performance Evaluation of Logistic Regression

|  | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Business | 0.83 | 0.96 | 0.89 | 653 |
| Sports | 0.95 | 0.94 | 0.94 | 437 |
| Nation | 0.94 | 0.78 | 0.85 | 1673 |
| Entertainment | 0.97 | 0.93 | 0.95 | 1289 |
| Editorial | 0.47 | 0.86 | 0.61 | 277 |
| **accuracy** |  |  | 0.88 | 4329 |
| **macro avg** | 0.83 | 0.90 | 0.85 | 4329 |
| **weighted avg** | 0.90 | 0.88 | 0.88 | 4329 |

**Table 4.3: Performance Evaluation metrics for Logistic Regression**

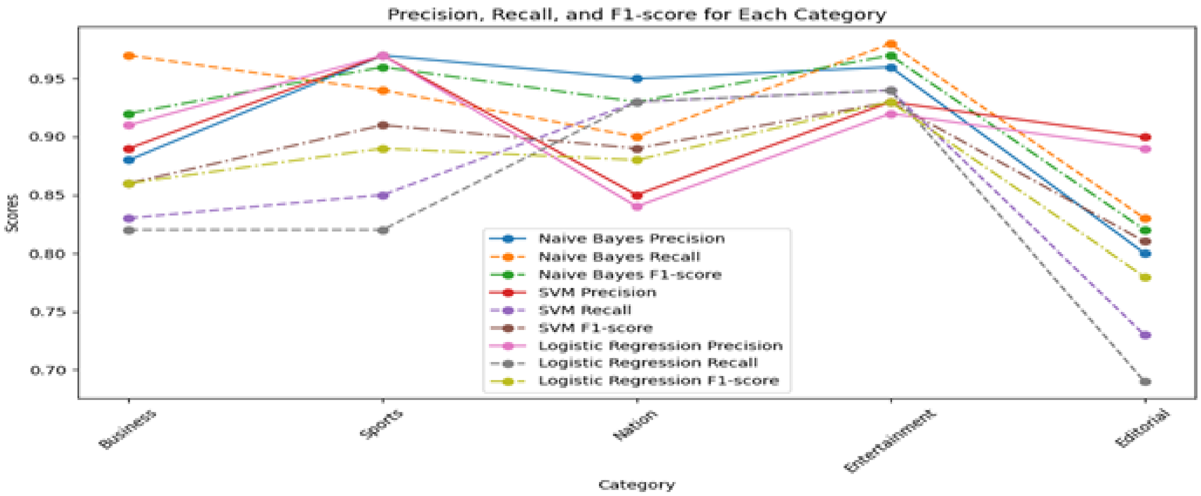


**Fig 4.1: graphical representation of precision, f1-score and recall of the Classification models on each of the category.**

The Table 4.4 interprets the data of comparison of accuracies between the Machine Learning Classification Models Naïve Bayes, SVM and Logistic Regression. We can observe that Naïve bayes is having high performance and outstanding accuracy than SVM and Logistic Regression.

| | **Accuracy** |
|---|---|
| Naïve Bayes | 93 |
| SVM | 88 |
| Logistic Regression | 88 |

**Table 4.4: Accuracy of the classification models**

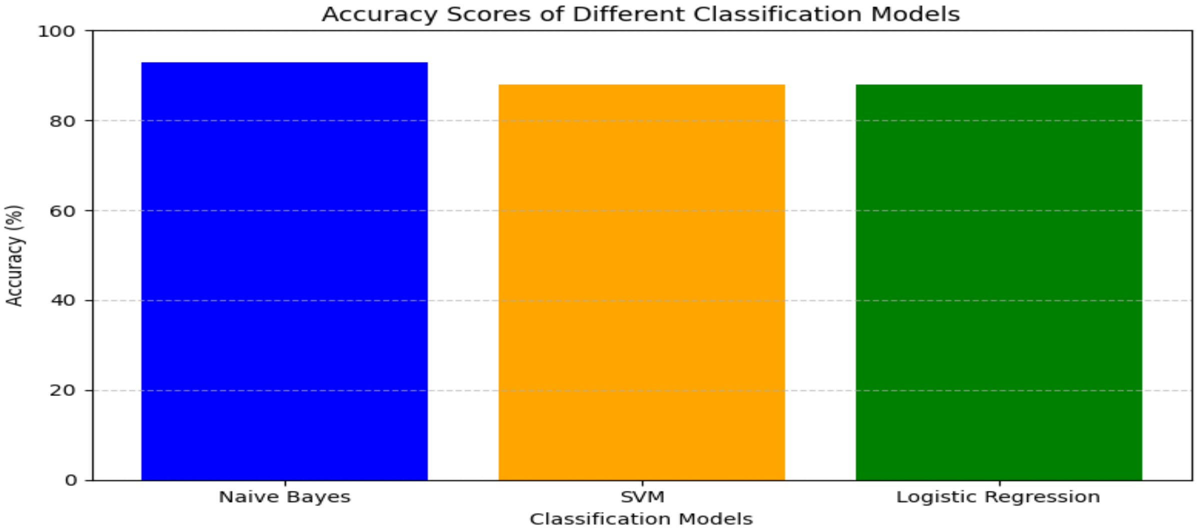


**Fig 4.1.2: Graphical representation of accuracy scores of the classification models SVM, Naïve Bayes, Logistic Regression**

# CHAPTER 5
# CONCLUSION

**CHAPTER 5**

# CONCLUSION

In conclusion, this study presents a comprehensive approach to categorizing Telugu documents into predefined topics using machine learning and natural language processing techniques. The research process involved data collection from the Telugu News dataset, pre-processing steps including special character removal and tokenization, and feature extraction using TF-IDF and count vectorization with n-grams. The classification models, including Naïve Bayes, SVM, and Logistic Regression, were trained and evaluated based on metrics such as accuracy, F1-score, precision, and recall. Naïve Bayes exhibited the highest accuracy at 93%, followed by SVM at 88%, and Logistic Regression at 88%.

Overall, Naïve Bayes emerged as the most effective model for classifying Telugu text data into predefined topics. However, future research may explore additional optimizations to enhance the performance of SVM and Logistic Regression models. This study contributes to advancing text classification methodologies in Telugu language processing, facilitating better organization and analysis of Telugu content online.

# REFERENCES

## REFERENCES

[1]. Nair, D. S., Jayan, J. P., Rajeev, R. R., & Sherly, E. (2014, September). SentiMa-sentiment extraction for Malayalam. In 2014 International conference on advances in computing, communications and informatics (ICACCI) (pp. 1719-1723). IEEE.

[2]. Sahu, S. K., Behera, P., Mohapatra, D. P., & Balabantaray, R. C. (2016). Sentiment analysis for Odia language using supervised classifier: an information retrieval in Indian language initiative. CSI transactions on ICT, 4, 111-115.

[3]. Sultana, J., Sadaf, K., & Jilani, A. K. (2021). Classifying Student's Academic Performance using SVM. Journal of Engineering and Applied Sciences, 8(2), 61-61.

[4]. Mukku, S. S., Choudhary, N., & Mamidi, R. (2016). Enhanced Sentiment Classification of Telugu Text using ML Techniques. SAAIP@ IJCAI, 2016, 29-34.

[5]. Sudha, D. N. (2021). Semi Supervised Multi Text Classifications for Telugu Documents. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(12), 644-648.

[6]. Sarkar, K. (2020). Heterogeneous classifier ensemble for sentiment analysis of Bengali and Hindi tweets. Sādhanā, 45(1), 196.

[7]. Naidu, R., Bharti, S. K., Babu, K. S., & Mohapatra, R. K. (2017, March). Sentiment analysis using telugu sentiwordnet. In 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 666-670). IEEE.

[8]. Samuel, J., Ali, G. M. N., Rahman, M. M., Esawi, E., & Samuel, Y. (2020). Covid-19 public sentiment insights and machine learning for tweets classification. Information, 11(6), 314.

[9]. Chattu, K., & Sumathi, D. (2023, July). Sentiment Classification of Low Resource Language Tweets Using Machine Learning Algorithms. In 2023 2nd International Conference on Edge Computing and Applications (ICECAA) (pp. 1055-1061). IEEE.

[10]. Sultana, J., Rani, M. U., Aslam, S. M., & AlMutairi, L. (2021). Predicting indian sentiments of COVID-19 using MLP and adaboost. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 706

[11]. Mukku, S. S. (2017). Sentiment Analysis for Telugu Language (Doctoral dissertation, PhD thesis, International Institute of Information Technology).

**GITHUB PROJECT LINK:**

**https://github.com/Chinni2103/Telugu_Data_Classification**

Volume: 12 Issue: III Month of publication: March 2024

DOI: https://doi.org/10.22214/ijraset.2024.59508

# Telugu Data Classification

Komal Parashar[1], V. Durga Bhavani[2], V. Keerthana[3], R. Narasimha[4]
*[1]Assistant Professor, [2, 3, 4]UG Students, Department of Computer Science & Engineering, CMR College Of Engineering & Technology, Hyderabad, Telangana*

***Abstract: Telugu is considered as the difficult languages which is morphologically rich when it comes to Dravidian languages. There are many Telugu documents available on Internet, it is important to organize the data by automatically by assigning a collection of text with predefined categories. Here the Telugu data is classified into multiple areas like business, sports, entertainment, nation, editorial is the main goal throughout this research work. This research work provides up an efficient model by adopting some ML classifiers such as SVM, Naive Bayes, and Logistic regression to perform some areas of classification on Telugu data. The results obtained by various machine-learning models are compared and an efficient model is discovered, and it is observed that the Naive Bayes model outperformed with reference to accuracy, precision, recall, and F1-score.***

***Keywords: Support Vector Machine (SVM), Naïve Bayes, Logistic Regression, ML classifiers***

## I. INTRODUCTION

These days, many people use online multimedia platforms such as blogs, online shopping review sites, feedback forums, and social networking sites. Share opinions and views in their native languages on Facebook, Twitter, WhatsApp, Instagram, LinkedIn about a specific topic. Here the text classification helps in the classification of the user opinions or views provided by the users into the domain it falls under and helps in organizing, identification of user preferences, personalized delivery of content, market analysis and facilitates the targeted users. Telugu, a really complex Dravidian language, poses challenges for organizing its abundant online content. To streamline this process, we aim to categorize Telugu documents into predefined topics like business, science, sports, etc. We'll employ modern techniques like Support Vector Machine (SVM); Naive Bayes; and Logistics Regression; backed by Natural Language Processing (NLP) in Machine Learning. By leveraging these methods, we seek to accurately classify Telugu text data and generate meaningful insights.

## II. LITERATURE SURVEY

Deepu et al., n.d. [1] proposed a rule-based approach for opinion classification of Malayalam motion picture audits, tending to challenges stemming from client input containing spelling botches. Essentially, Sanjib et al.,[2] created a framework utilizing administered classification methods to classify Odia motion picture audit estimations as positive and negative. In differentiate, S.S. Mukku et al., [3] displayed a system for Telugu opinion investigation utilizing Doc2Vec models prepared with different ML procedures Moreover, J.Sultana et al., [4] compared conventional Profound Learning and ML approaches for opinion expectation on instructive information, finding MLP to abdicate the finest results. At long last, D Naga et al., [5] examined n-gram include selection's effect on news article content classification utilizing semi-supervised learning strategies, highlighting SVM's prevalence. The creators of the current think about propose to analyze Telugu news assumptions utilizing machine learning strategies to address the require for assumption investigation in Telugu news.

Kamal Sarkar et al., [6] and colleagues did some study about feeling analysis using a multinomial Naïve Bayes classifier enhanced with fancy determination characteristics. They focused on looking at feelings in tweets written in Bengali and Hindi, showing how their method can work in different languages. The Multinomial Naïve Bayes thing is great for handling jobs about text figures. The determination chat included deeper feeling analysis, underlining how smart their process is. The work by Sarkar and the gang gives us some good ideas for studying feelings in various languages. This study is super important in the international social media world because looking at feelings helps in knowing what people think across different languages.

Reddy Naidu et al., [7] presented a novel two-phase estimation research approach in their sentiment analysis investigation for Telugu e-News. This technique uses Telugu SentiWordNet, a lexical resource specific to the language, to categorize phrases found in Telugu e-News articles. The authors' use of a two-phase process points to a sophisticated and specialized method for handling the complexities of sentiment analysis in Telugu. Naidu et al. substantially advance sentiment analysis research by concentrating on regional language subtleties and offering a system that works with Telugu e-news information.

Their research contributes to our knowledge of sentiment dynamics in regional languages and may help tailor sentiment analysis techniques to a different type of linguistic circumstances. There is potential for improving sentiment analysis methods through more investigation of their methodology.

Samuel et al., [8] and his colleagues conducted a study that somehow connects to something about Coronavirus Tweets in the field of social media analytics. Their research on the Calculated Relapse and some other fancy Naïve Bayes techniques resulted in good precision, especially when they're dealing with the tiny Tweets. Because Twitter, is about short messages, so being brief is very important. This paper gives a guide for making tweet categorization methods better for the occurrences which happens like right now, you know, the epidemic.

## III. EXISTING SYSTEM METHODOLOGY

In the existing implementation Telugu news is translated into English using the Google translation library available in Python. Then determined the sentiment scores using various tagging techniques and mark them as - positive/negative. Then, an attempt was made for classifying polarity value of Telugu news statements using several ML classifiers namely Naive Bayes, Random Forest, Passive-Aggressive Classifier, Perceptron, and SVM (Support Vector Machine). Here, the models are created for classification. One is a binary class and the other is a multiclass model. In binary classification, the system categorizes sentiment into positive polarity or negative polarity. Meanwhile, in the multiclass classification task, the system further categorizes the sentiment into business, editorial, entertainment, nation, and sports. Results were implemented using test data against performance parameters.

### *A. Implementation of Existing System*

*1)* Telugu news is translated into English
*2)* Sentiment analysis is performed using different models
*3)* The classifiers SVM, Random Forests and Naïve Bayes used for the multiclass task
*4)* Training the Classifier Models
*5)* Testing the trained classifier models
*6)* Analysis of model performance

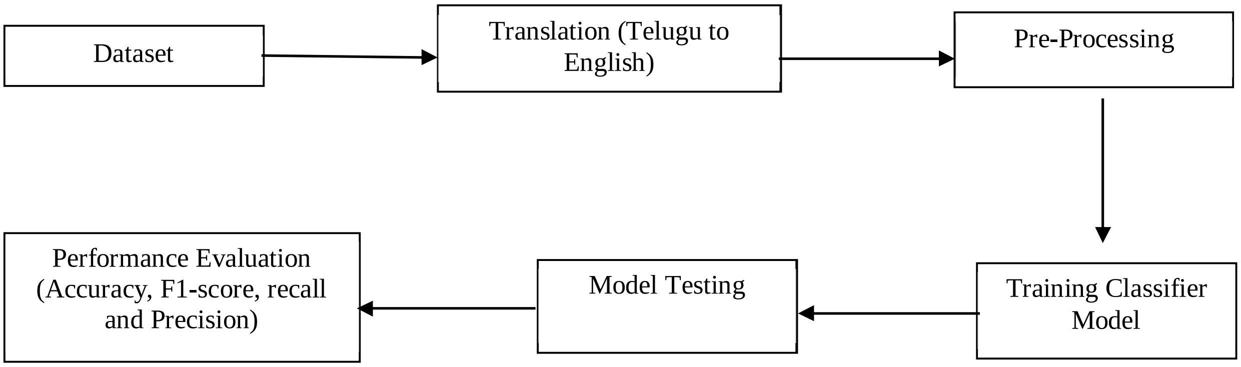
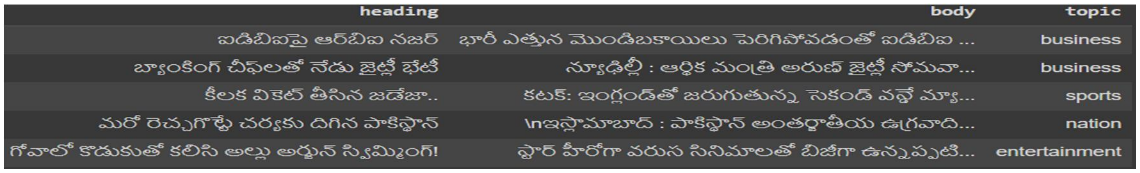

Dataset
Translation (Telugu to English)
Pre-Processing
Training Classifier Model
Model Testing
Performance Evaluation (Accuracy, F1-score, recall and Precision)

Fig. 1 Flow Diagram of Existing System

## IV. PROPOSED METHODOLOGY

### *A. Dataset*

Text classification is a crucial aspect and influences strongly on social practices. Most of the news articles generated in Telugu have not received much attention in the sentiment analysis community. These reasons led to choose the news dataset. The Telugu News dataset was collected by Kaggle (SRK, 2020). This Telugu news statements belonging to five areas and class labels used (Business, Editorial, National, Sports, Entertainment).



| heading | body | topic |
|---|---|---|
| ఐడిబిఐపై ఆర్బిఐ నజర్ | భారీ ఎత్తున మొండిబకాయిలు పెరిగిపోవడంతో ఐడిబిఐ ... | business |
| బ్యాంకింగ్ చీఫ్‌లతో నేడు జైట్లీ భేటీ | న్యూఢిల్లీ : ఆర్థిక మంత్రి అరుణ్ జైట్లీ సోమవా... | business |
| కీలక వికెట్ తీసిన జడేజా.. | కటక్: ఇంగ్లండ్‌తో జరుగుతున్న సెకండ్ వన్డే మ్యా... | sports |
| మరో రెచ్చగొట్టే చర్యకు దిగిన పాకిస్థాన్ | \nఇస్లామాబాద్ : పాకిస్థాన్ అంతర్జాతీయ ఉగ్రవాది... | nation |
| గోవాలో కొడుకుతో కలిసి అల్లు అర్జున్ స్విమ్మింగ్! | స్టార్ హీరోగా వరుస సినిమాలతో బిజీగా ఉన్నప్పటి... | entertainment |

Fig. 2 Some samples of Telugu news dataset

### *B. Pre-Processing*

Pre-Processing step is the first crucial step for removing the cleaning the data so that model can understand the effectively to yield a high-performance result. The pre-processing steps includes

*1)* Removes special characters
*2)* Sklearn: It is Label Encoder assigns numbered values to categories.
*3)* Tokenization: Breaking down the text into tokens and creating vocabulary.

### *C. Feature Extraction*

Feature extraction: **-** Used in converting raw data into computer understandable that is into numerical format. The Feature Extraction includes

*1)* Vectorization: Converting the text into numerical values using TF-IDF (Term Frequency Inverse Document Frequency) and count vectorization.
*2)* Divides text into feature matrix using n-grams (uni-grams, bi-grams, tri-grams, 4-grams, 5-grams) which is the sequence of words.

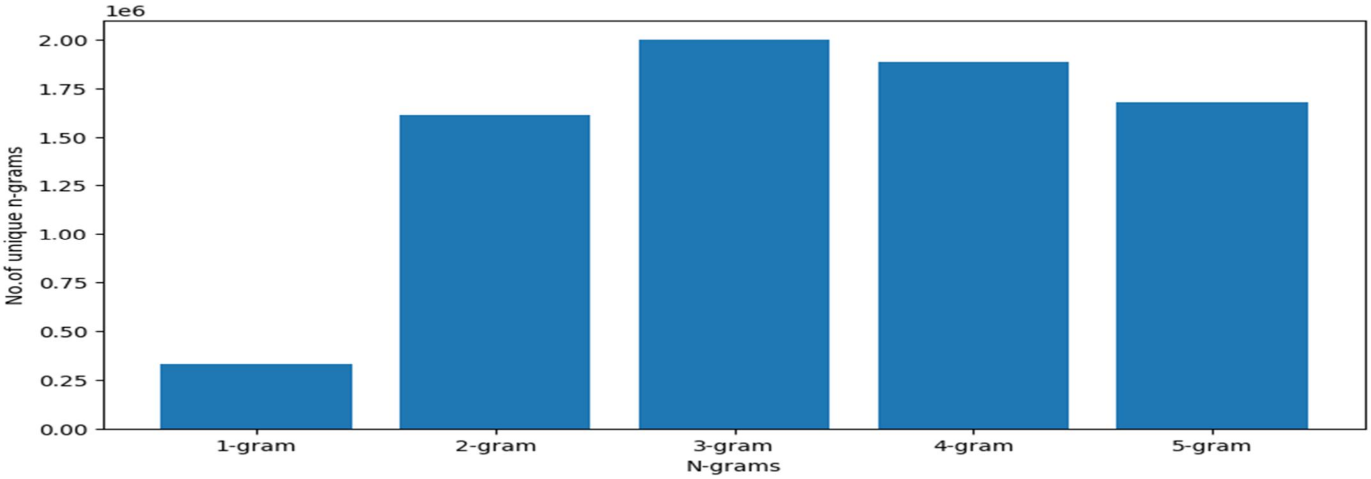


Fig. 3 Number unique N- grams

### *D. Classification Models*

In our extend we used mainly three ML classification models those are Naïve Bayes, SVM, and Logistic Regression. A robust methodology which is simple and effective in classifying the Telugu text into different types of categories. Here, Naïve Bayes is very efficient classification technique for the small range of dataset and it calculates the probability for each category base on the input extracted features. Equally, SVM strikes to get the optimal hyperplane separating data points of various categories with the largest margin and efficiently captures the relationships in the high dimensional space. Logistic regression is also called as linear model which measures the probability of the binary output which can be further extended to classify multiple categories.

### *E. Implementation*

The Flowchart starts with the importing of the packages such as indic-nlp library for the Indian Language understanding. Then, the dataset is extracted from the Kaggle for the Telugu data classification and processing data and giving a numerical value to each topic with the sklearn library- Label Encoder so that the model can understand and yield efficient results. After the Label encoding the feature extraction of N-gram using Count vectorization and TfidVectorization which is used for counting the frequency of occurrence of the n-gram (uni-gram, bigram, tri-gram, 4-gram, 5-gram). The vectors are generated in the feature extraction is kept for training the classification models Naïve bayes, SVM and Logistic Regression. After the models are trained the trained model is used for the prediction and performance of these models are being evaluated. The Evaluation of the trained is provided by the evaluation metrics Accuracy, F1-score, Precision and Recall.

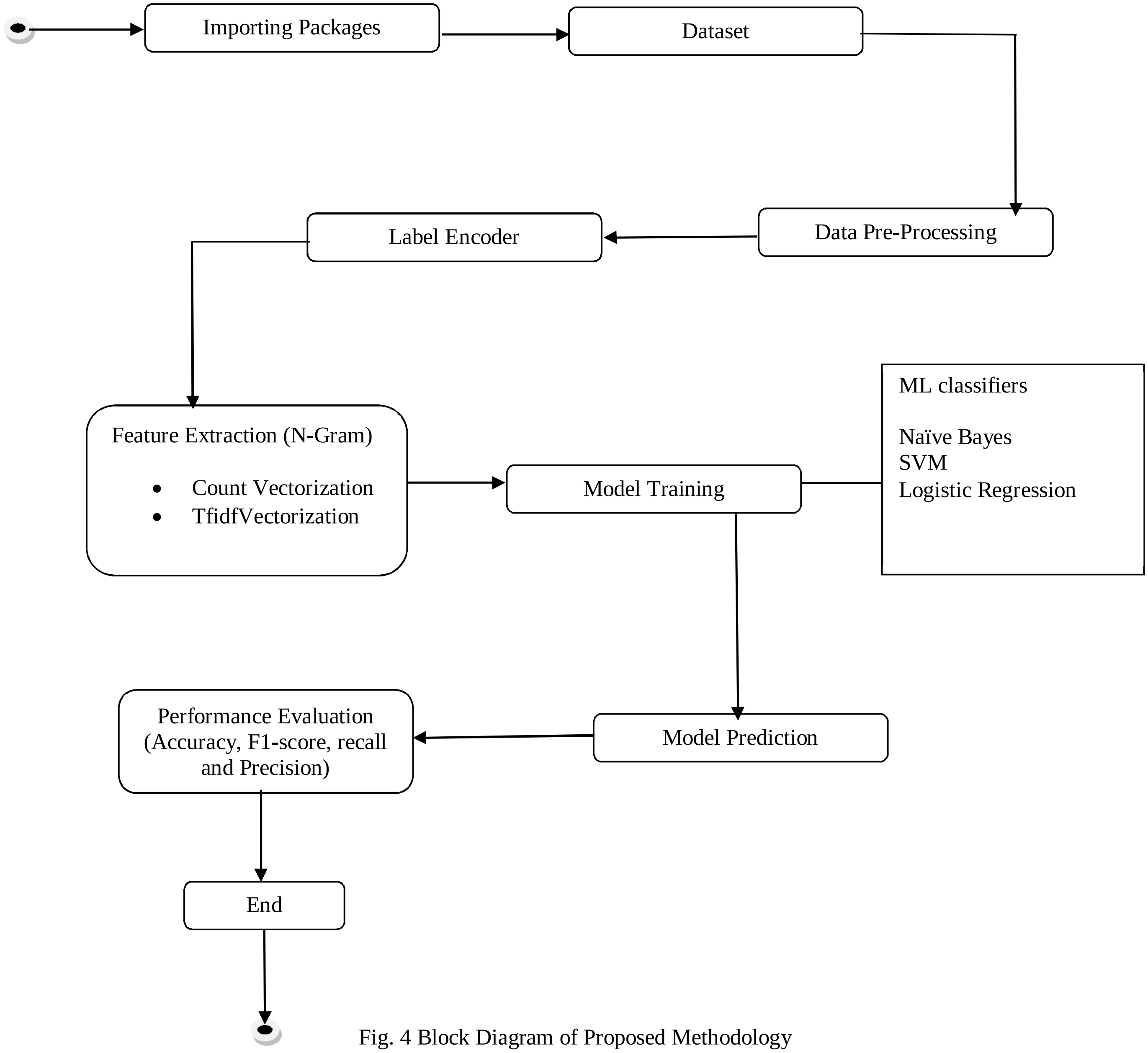


Fig. 4 Block Diagram of Proposed Methodology

## V. PERFORMANCE EVALUATION

### *A. Performance Evaluation of Naïve Bayes*

Table 1: Performance Evaluation metrics for Naïve Bayes

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Business | 0.88 | 0.97 | 0.92 | 653 |
| Sports | 0.97 | 0.94 | 0.96 | 437 |
| Nation | 0.95 | 0.90 | 0.93 | 1673 |
| Entertainment | 0.96 | 0.98 | 0.97 | 1289 |
| Editorial | 0.80 | 0.83 | 0.82 | 277 |
| accuracy | | | 0.93 | 4329 |
| macro avg | 0.91 | 0.92 | 0.92 | 4329 |
| weighted avg | 0.94 | 0.93 | 0.93 | 4329 |

### *B. Performance Evaluation of SVM*

Table 2: Performance Evaluation metrics for SVM

| B | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Business | 0.89 | 0.83 | 0.86 | 534 |
| Sports | 0.97 | 0.85 | 0.91 | 380 |
| Nation | 0.85 | 0.93 | 0.89 | 1296 |
| Entertainment | 0.93 | 0.94 | 0.93 | 1058 |
| Editorial | 0.90 | 0.73 | 0.81 | 195 |
| accuracy | | | 0.90 | 3463 |
| macro avg | 0.91 | 0.86 | 0.88 | 3463 |
| weighted avg | 0.90 | 0.90 | 0.90 | 3463 |

### *C. Performance Evaluation of Logistic Regression*

Table 3: Performance Evaluation metrics for Logistic Regression

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Business | 0.91 | 0.82 | 0.86 | 534 |
| Sports | 0.97 | 0.82 | 0.89 | 380 |
| Nation | 0.84 | 0.93 | 0.88 | 1296 |
| Entertainment | 0.92 | 0.94 | 0.93 | 1058 |
| Editorial | 0.89 | 0.69 | 0.78 | 195 |
| accuracy | | | 0.89 | 3463 |
| macro avg | 0.91 | 0.84 | 0.87 | 3463 |
| weighted avg | 0.89 | 0.89 | 0.89 | 3463 |

## VI. RESULTS AND DISCUSSION

The Table 4 interprets the data of comparison of accuracies between the Machine Learning Classification Models Naïve Bayes, SVM and Logistic Regression. We can observe that Naïve bayes is having high performance and outstanding accuracy than SVM and Logistic Regression.

Table 4: Accuracy of the classification models

| | Accuracy |
|---|---|
| Naïve Bayes | 93 |
| SVM | 90 |
| Logistic Regression | 89 |

Figures

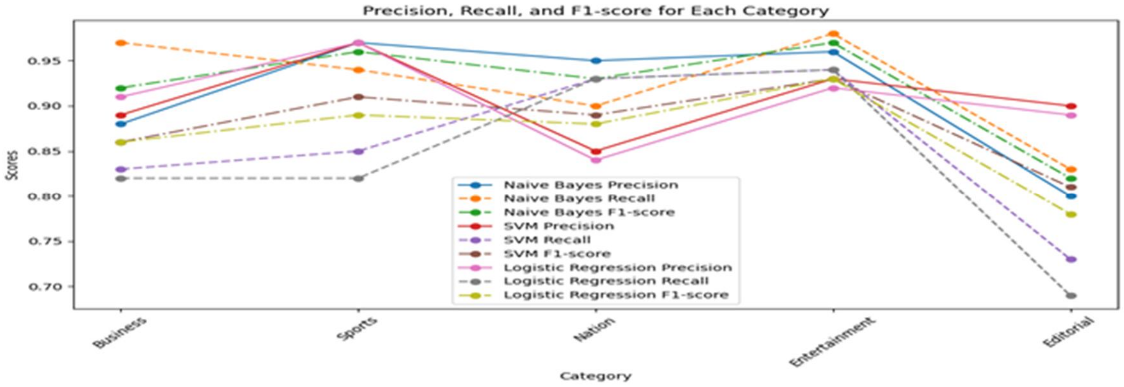


Fig. 5 graphical representation of precision, f1-score and recall of the Classification models on each of the category.

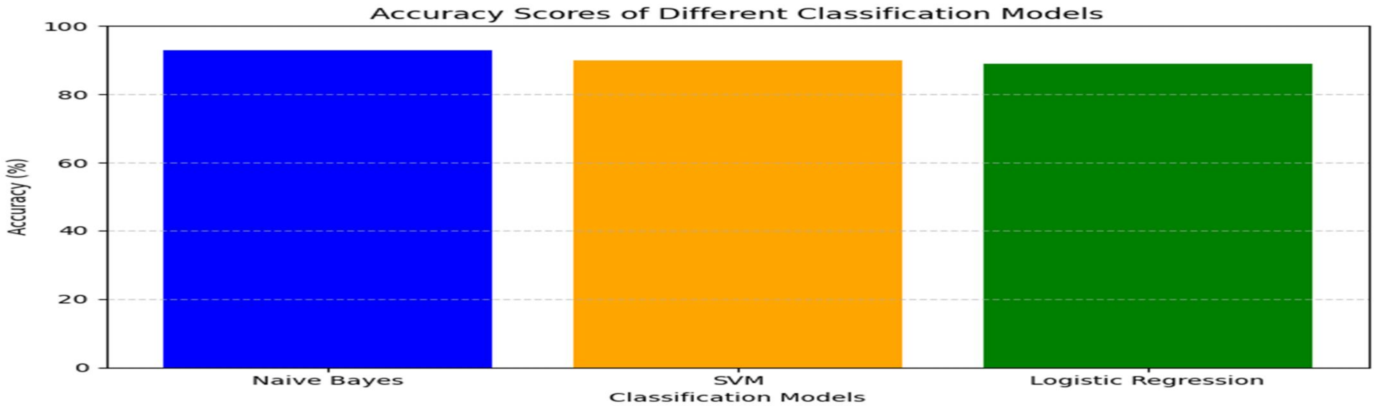


Fig. 6 Graphical representation of accuracy scores of the classification models SVM, Naïve Bayes, Logistic Regression

## VII. CONCLUSION

Our research work helps in the classifying the Telugu text into the categories such as Business, sports, nation, Editorial and Entertainment. The Telugu news dataset is used for classification of categories using the ML classification models like Naïve Bayes, Logistic Regression and SVM. The Naïve Bayes performance outperforms Logistic Regression and SVM.

## REFERENCES

[1] Nair, D. S., Jayan, J. P., Rajeev, R. R., & Sherly, E. (2014, September). SentiMa-sentiment extraction for Malayalam. In 2014 International conference on advances in computing, communications and informatics (ICACCI) (pp. 1719-1723). IEEE.

[2] Sahu, S. K., Behera, P., Mohapatra, D. P., & Balabantaray, R. C. (2016). Sentiment analysis for Odia language using supervised classifier: an information retrieval in Indian language initiative. CSI transactions on ICT, 4, 111-115.

[3] Mukku, S. S., Choudhary, N., & Mamidi, R. (2016). Enhanced Sentiment Classification of Telugu Text using ML Techniques. SAAIP@ IJCAI, 2016, 29-34.

[4] Sultana, J., Sadaf, K., & Jilani, A. K. (2021). Classifying Student's Academic Performance using SVM. Journal of Engineering and Applied Sciences, 8(2), 61-61.

[5] Sudha, D. N. (2021). Semi Supervised Multi Text Classifications for Telugu Documents. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(12), 644-648.

[6] Sarkar, K. (2020). Heterogeneous classifier ensemble for sentiment analysis of Bengali and Hindi tweets. Sādhanā, 45(1), 196.

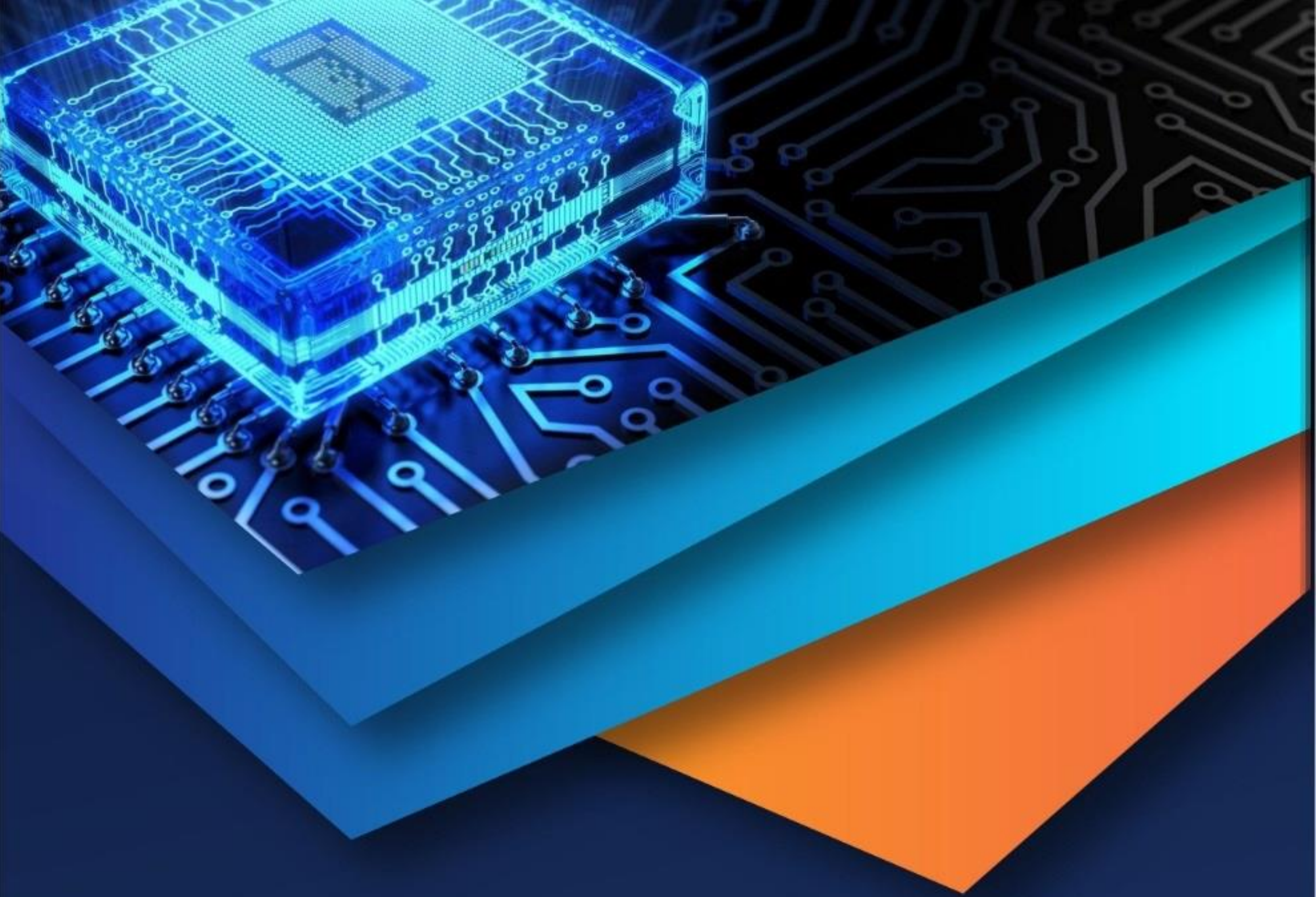
[7] Naidu, R., Bharti, S. K., Babu, K. S., & Mohapatra, R. K. (2017, March). Sentiment analysis using telugu sentiwordnet. In 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 666-670). IEEE.

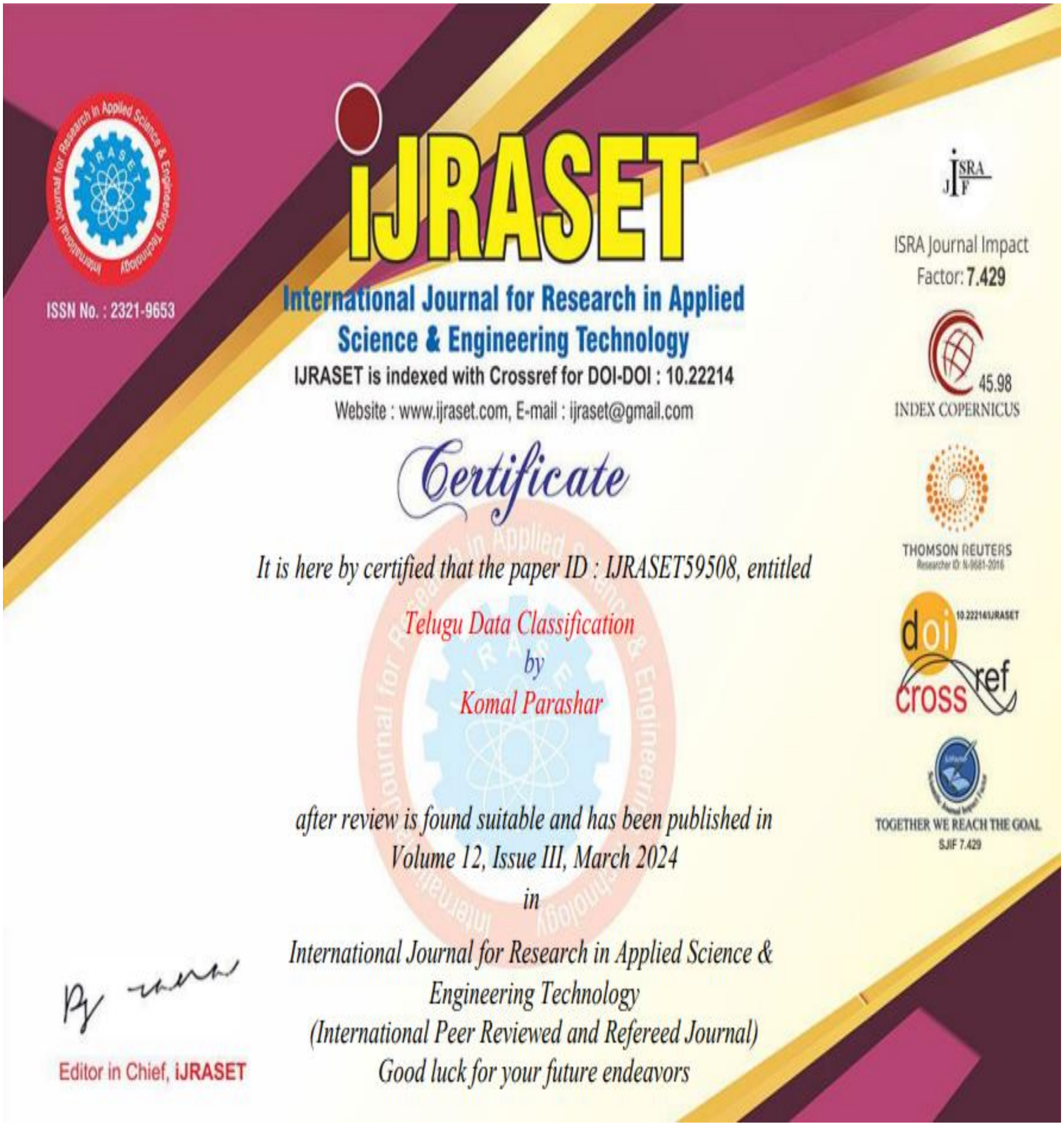
[8] Samuel, J., Ali, G. M. N., Rahman, M. M., Esawi, E., & Samuel, Y. (2020). Covid-19 public sentiment insights and machine learning for tweets classification. Information, 11(6), 314.

[9] Mukku, S. S. (2017). Sentiment Analysis for Telugu Language (Doctoral dissertation, PhD thesis, International Institute of Information Technology).

[10] Chattu, K., & Sumathi, D. (2023, July). Sentiment Classification of Low Resource Language Tweets Using Machine Learning Algorithms. In 2023 2nd International Conference on Edge Computing and Applications (ICECAA) (pp. 1055-1061). IEEE.

[11] Sultana, J., Rani, M. U., Aslam, S. M., & AlMutairi, L. (2021). Predicting indian sentiments of COVID-19 using MLP and adaboost. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 706-714.





ISSN No. : 2321-9653

# iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL
SJIF 7.429

## Certificate

*It is here by certified that the paper ID : IJRASET59508, entitled*

*Telugu Data Classification*
*by*
*Komal Parashar*

*after review is found suitable and has been published in*
*Volume 12, Issue III, March 2024*
*in*

*International Journal for Research in Applied Science &*
*Engineering Technology*
*(International Peer Reviewed and Refereed Journal)*
*Good luck for your future endeavors*

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL

SJIF 7.429

*Certificate*

*It is here by certified that the paper ID : IJRASET59508, entitled*

*Telugu Data Classification*

*by*

*V. Durga Bhavani*

*after review is found suitable and has been published in*

*Volume 12, Issue III, March 2024*

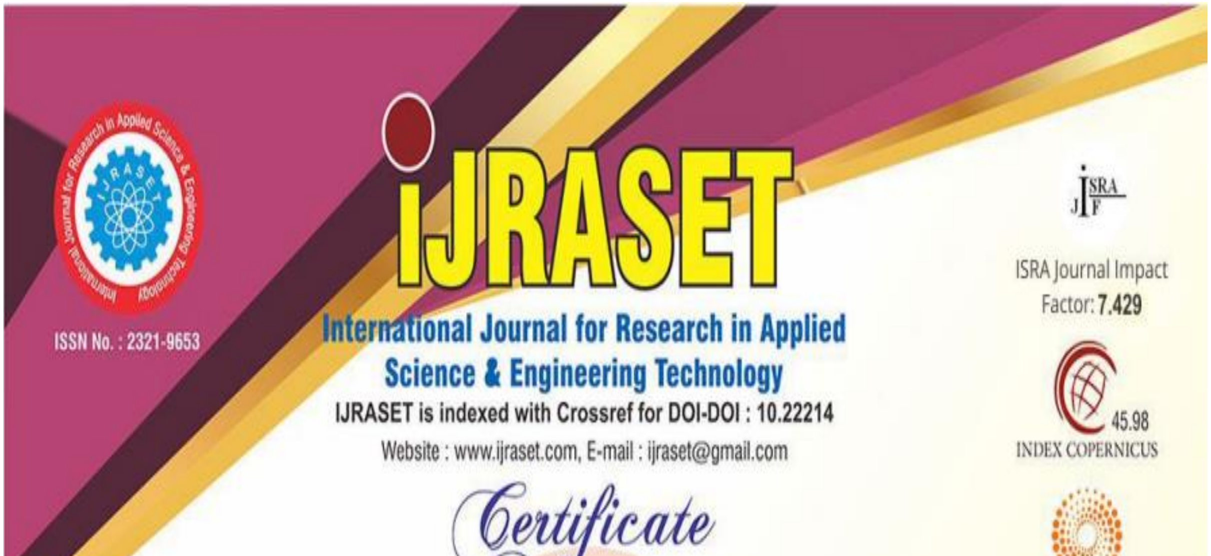
*in*

*International Journal for Research in Applied Science & Engineering Technology*

*(International Peer Reviewed and Refereed Journal)*

*Good luck for your future endeavors*

Editor in Chief, iJRASET



ISSN No. : 2321-9653

# iJRASET

**International Journal for Research in Applied Science & Engineering Technology**

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

doi 10.22214/IJRASET crossref

TOGETHER WE REACH THE GOAL SJIF 7.429

## Certificate

*It is here by certified that the paper ID : IJRASET59508, entitled*

*Telugu Data Classification*

*by*

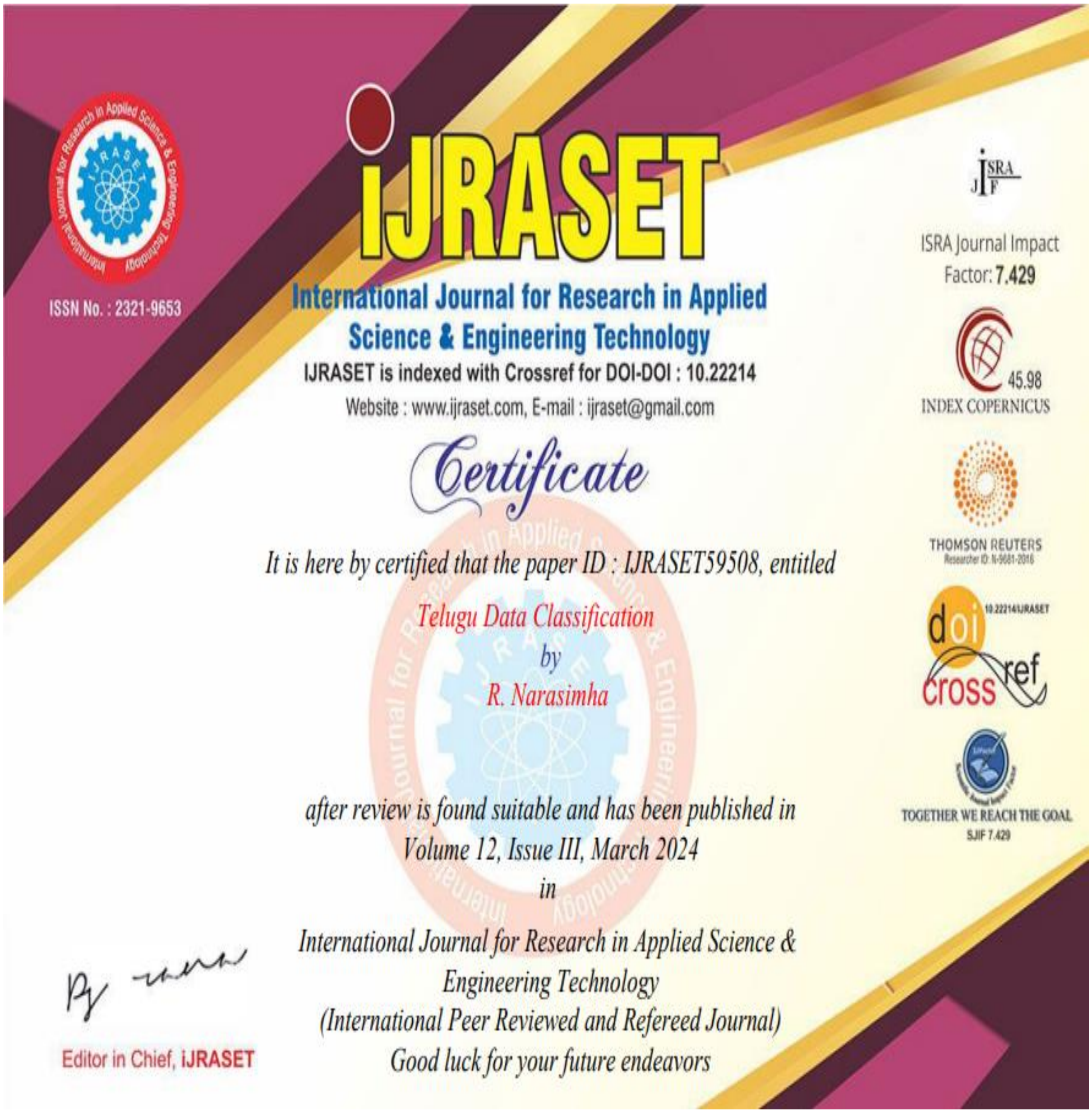
*V. Keerthana*

*after review is found suitable and has been published in*
*Volume 12, Issue III, March 2024*

*in*

*International Journal for Research in Applied Science &*
*Engineering Technology*
*(International Peer Reviewed and Refereed Journal)*
*Good luck for your future endeavors*

Editor in Chief, IJRASET



ISSN No. : 2321-9653

# iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

## Certificate

*It is here by certified that the paper ID : IJRASET59508, entitled*

*Telugu Data Classification*

*by*

*R. Narasimha*

*after review is found suitable and has been published in*
*Volume 12, Issue III, March 2024*
*in*
*International Journal for Research in Applied Science & Engineering Technology*
*(International Peer Reviewed and Refereed Journal)*
*Good luck for your future endeavors*

Editor in Chief, iJRASET