Language Detection using Machine Learning

A PROJECT REPORT

Submitted by

PRAVEEN M (2116210701193)

in partial fulfillment for the award

of the degreeof

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING

COLLEGEANNA UNIVERSITY,

CHENNAI

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE,CHENNAI

BONAFIDE CERTIFICATE

Certified that this Thesis titled **“Language Detection using Machine Learning** ” is the bonafide work of “**PRAVEEN M (210701193)**” who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr . S Senthil Pandi M.E., Ph.D.,
SUPERVISOR
Professor
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Chennai - 602 105
Submitted to Project Viva-Voce Examination held on___________________

Internal Examiner **External Examine**r

ABSTRACT

This project focuses on the development and comparison of machine learning models for language detection using a variety of algorithms: Support Vector Machines (SVM), Naive Bayes, Logistic Regression, and k-Nearest Neighbours (KNN). The study includes preprocessing steps such as tokenization and numerical conversion to create appropriate feature representations for each model. Training and testing are conducted using datasets containing text samples from different languages. The evaluation of each model involves standard criteria like precision,F1-score,accuracy,recall allowing for a comprehensive comparison. Additionally, computational efficiency and suitability for real-world language detection tasks are considered. Parameter optimization techniques are explored to enhance model performance and generalization. This comparative analysis aims to present the positives and negatives of each algorithm in accurately detecting the language of input text. The findings will assist in choosing the most suitable model based on specific project requirements, computational constraints, and deployment scenarios. The project's outcomes are relevant to various applications requiring language identification, including multilingual content processing, language-specific services, and global communication platforms. By leveraging machine learning methodologies, this research contributes to advancing language detection accuracy and efficiency in diverse linguistic contexts.

Keywords—Language Detection, k-Nearest Neighbours, Support Vector Machines(SVM), Logistic Regression , Naïve Bayes.

I. SECTION

INTRODUCTION

Language detection, an integral aspect of modern information processing systems, involves identifying the language of the given text. This task holds significant importance in various domains, including international communication, content filtering, and cross-border data analysis. Reliable language detection systems are essential for enabling effective communication and information retrieval across linguistic barriers.

This research paper will explore and compare the effectiveness of traditional machine learning algorithms for language detection. Specifically, we will investigate four classical algorithms : Logistic Regression, Naive Bayes, k-Nearest Neighbors (KNN) and Support Vector Machines (SVM). These algorithms have been widely studied and applied in text classification tasks, making them suitable candidates for language detection purposes.

The primary objective of this study is to assess the performance of these algorithms in accurately identifying the language of input text. We will follow a systematic approach, starting with data preprocessing techniques such as tokenization and feature extraction. The preprocessed data will then be used to train and evaluate the language detection models. In addition to evaluating the accuracy of language detection, we aim to analyze the robustness and scalability of each algorithm. Factors such as computational efficiency, memory usage, and model complexity will be considered to understand the practical feasibility of deploying these models in real-world applications. Furthermore, we will explore the impact of varying dataset sizes and language distributions on model performance.

Through this comparative analysis, we aim to suggest intel on the strengths and limitations of traditional machine learning approaches for language detection tasks. The findings of this research will be valuable for developers, practitioners, and researchers seeking to implement language detection systems without relying on advanced natural language processing techniques.

This research endeavors to evaluate the efficiency of traditional machine learning models in language detection across a multitude of languages. The primary objectives include analyzing the robustness and scalability of each algorithm when confronted with large-scale language detection tasks. Additionally, the study aims to explore how varying dataset sizes and language distributions impact the algorithms' performance and generalizability. Furthermore, it seeks to compare the computational efficiency and memory usage of these algorithms, thereby assessing their practical suitability for real-world deployment. By furnishing actionable insights and guidelines, this research endeavors to aid decision-makers in selecting the most appropriate machine learning algorithms for language detection tasks, considering their unique project requirements and constraints.

The outcomes of this study has the advantage of providing more efficient and accurate language detection solutions, thereby enhancing cross-cultural communication, content filtering, and global data analysis capabilities across diverse linguistic contexts.

II. SECTION

LITERATURE SURVEY

A. OVERVIEW

The literature review section of this research paper provides an extensive examination of prior studies and methodologies concerning language detection utilizing traditional machine learning (ML) algorithms. It explores the efficacy of Logistic Regression, Naive Bayes, k-Nearest Neighbors (KNN) and Support Vector Machines (SVM) in language detection tasks, stressing the significance of feature engineering, model selection, and algorithmic performance across diverse linguistic and textual contexts. Moreover, it investigates various feature representation techniques, encompassing hybrid methods such as word and character n-grams, word embeddings, and semantic similarity approaches.

In addition to algorithmic effectiveness, the review evaluates the influence of dataset characteristics such as size, language diversity, and noise on model performance. It sheds light on challenges like imbalanced datasets and code-switching in multilingual environments, recognizing the complexity inherent in real-world language data. Furthermore, the review delves into scalability and efficiency considerations in language detection models, discussing their implications for practical applications across domains such as social media analysis and content filtering.

By synthesizing insights gleaned from these sources, the literature review aims to offer a comprehensive understanding of language detection utilizing traditional ML algorithms. It seeks to identify existing gaps and opportunities for further research and development in this burgeoning field, with the ultimate goal of advancing the state-of-the-art in language detection technology.

B. COMPARISON WITH OTHER APPROACHES

In comparison to traditional machine learning models such as Logistic Regression, Naive Bayes, k-Nearest Neighbors (KNN) and Support Vector Machines (SVM) other advanced models present both advantages and challenges in the context of language detection tasks. Deep learning models such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) excel in capturing sophisticated patterns and semantic representations from text data, making them suitable for large-scale language processing. However, they require substantial amounts of annotated data and computational resources for training, limiting their practicality in resource-constrained environments. Ensemble methods such as Gradient Boosting and Random Forests Machines showcase robustness against overfitting and noise in data, yet they can be computationally intensive and lack interpretability.

Transformer models like BERT and GPT have revolutionized natural language processing by capturing contextual information effectively, but their high computational cost during training and inference poses challenges for real-time applications. Probabilistic models and rule-based systems offer interpretability and domain-specific rule handling, but they may struggle with capturing complex non-linear relationships and adapting to diverse language variations. Ultimately, the selection of particular model depends on criteria like dataset size, computational resources, interpretability needs, and specific application requirements in language detection.

III.SECTION

METHODOLOGY

A. DATASET COLLECTION

The dataset utilized for language detection in this research paper was obtained from Kaggle, a platform for data science enthusiasts and practitioners. The dataset comprises a diverse range of text samples representing set of languages like Tamil, English, Hindi, Malayalam, Kannada, Russian, Italian, Portuguese, Spanish, Swedish, German, Turkish, Dutch, Greek, Danish and Arabic.The data collection process involved retrieving text samples from various authentic sources such as language-specific websites, publicly available text corpora, news articles, social media posts, and literature repositories. Measures were taken to ensure dataset quality and authenticity, including filtering out duplicate entries, irrelevant content, and noise, as well as conducting data integrity checks for accurate language labeling. The use of the Kaggle dataset enabled a comprehensive analysis of machine learning models for language detection across multiple languages.

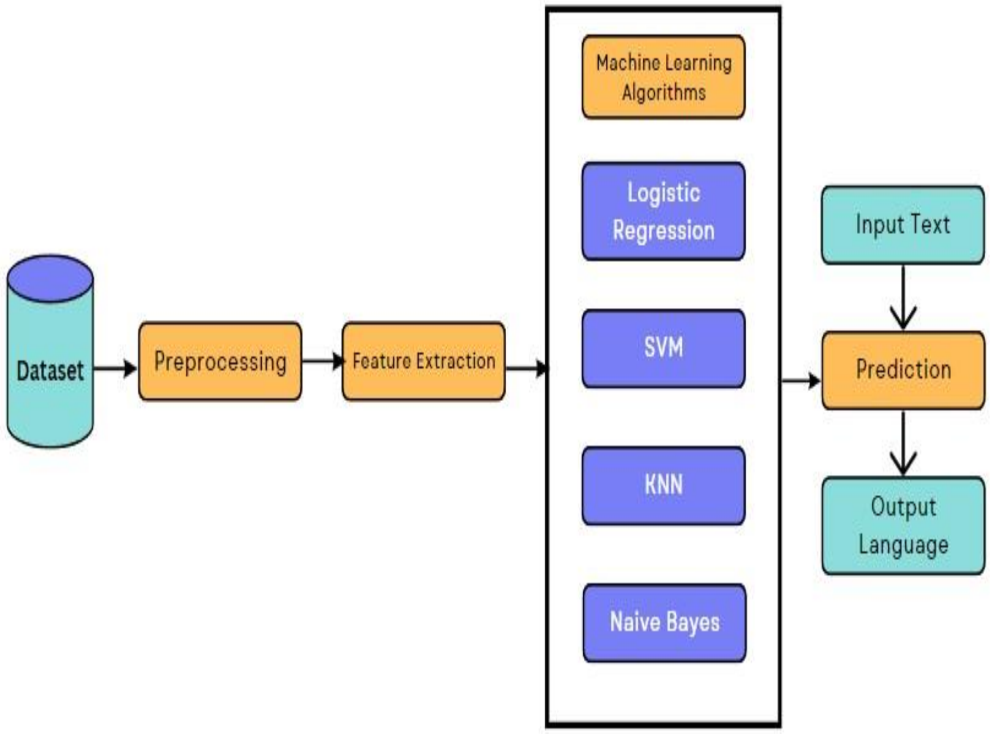
B. PREPROCESSING PROCEDURES

1. Text Cleaning: Text cleaning removes noise and irrelevant information from raw text data by eliminating special characters and converting text to lowercase for uniformity.
2. Tokenization: Tokenization breaks down text into meaningful segments, such as individual tokens, facilitating feature extraction for machine learning models.
3. Stop word Removal: Stop word removal filters out common stop words like "the" and "is" from tokenized text, reducing noise and improving data quality.
4. Normalization: Normalization techniques like stemming and lemmatization reduce word variations, enhancing consistency and improving model generalization.

5. Numerical Conversion: Numerical conversion transforms preprocessed text into numerical representations using TF-IDF vectorization, enabling efficient model processing.

6. Data Splitting: Partitioning pre processed data into training (80%) and testing (20%) sets ensures unbiased evaluation on unseen samples.

7. Data Encoding: Data encoding converts language labels into numerical format using label encoder technique, facilitating supervised learning tasks for language detection.



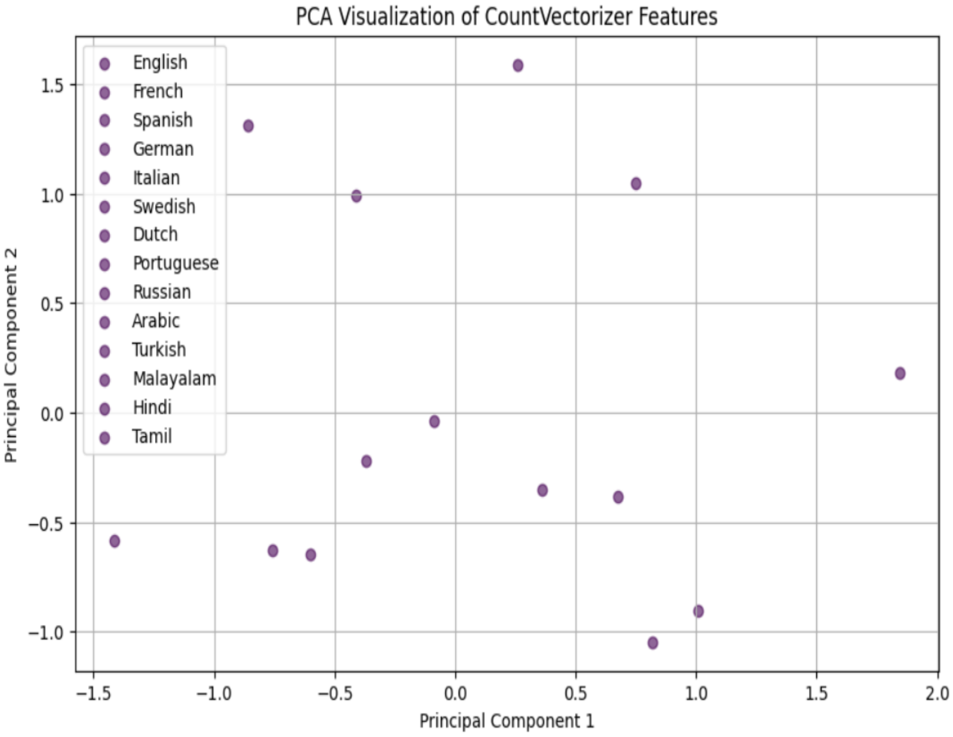
Model Architecture

C. FEATURE EXTRACTION

Feature extraction is a crucial preprocessing step in language detection, where raw text is transformed into a structured representation of various features.

In this study, we employed CountVectorizer from the sklearn.feature_extraction.text module to extract features from the text data. CountVectorizer converts the text into small tokens and makes it into a vocabulary of known words highlights them as vectors(documents),with dimension and frequency correlation of words in document. By fitting CountVectorizer to the training data and transforming both the training and testing data, we obtained bag-of-words representations for each text sample.

These representations capture the presence and frequency of words in the text, providing a numerical format suitable for machine learning algorithms. Additionally, the vocabulary learned from the training data offers insights into the unique words observed in the corpus. CountVectorizer provides a simple yet effective approach to feature extraction, facilitating the accurate detection of languages in text data.



D. MACHINE LEARNING ALGORITHMS

1)SUPPORT VECTOR MACHINES

Support Vector Machines (SVM) serve as a cornerstone in language detection problems because of their capability to handle high-dimensional feature spaces and nonlinear decision boundaries. In our project, SVMs were employed to discern language patterns from the features extracted using CountVectorizer. By identifying the optimal hyperplane that separates text samples of different languages, SVMs ensure robust and accurate classification results. The utilization of this enables effective language detection even in scenarios with complex language distributions, contributing to the overall success of our classification framework.

$$f(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

2)LOGISTIC REGRESSION

Logistic Regression stands out as a versatile and interpretable algorithm for language detection, capable of handling both multi-class classification and binary tasks. Leveraging features extracted via CountVectorizer, Logistic Regression models the relationship between input features and the probability of each language class. Through optimization techniques such as gradient descent, Logistic Regression effectively learns the underlying language patterns present in the text data. In our project, Logistic Regression offers a transparent and efficient solution for language detection, providing intel for specific decisions and enhancing the interpretability of our classification framework.

$$P(y = 1|\mathbf{x}) = \frac{1}{1+e^{-(\mathbf{w}\cdot\mathbf{x}+b)}}$$

3)K-NEAREST NEIGHBOURS

k-Nearest Neighbours (KNN) presents a simple yet effective approach to language detection, particularly suitable for scenarios with non-linear decision boundaries. By measuring the similarity of features between text samples, KNN accurately classifies text samples into their respective languages. Leveraging features extracted via CountVectorizer, KNN offers robust and adaptable language detection capabilities. In our project, KNN's flexibility enables it to adapt to varying data distributions, making it a valuable component of our language detection framework.

$$\hat{y} = \mathrm{mode}(y_1, y_2, \ldots, y_k)$$

4)NAÏVE BAYES

Naive Bayes classifiers offer a computationally efficient and scalable solution for language detection tasks, making them well-suited for large-scale text classification endeavors. Despite the simplifying assumption of feature independence, Naive Bayes classifiers effectively model the conditional probabilities of each language given the features extracted from the text data. In our project, Naive Bayes classifiers, coupled with features extracted via CountVectorizer, demonstrate reliable and consistent language detection performance. The probabilistic nature of

Naive Bayes facilitates robust classification outcomes, augmenting the effectiveness of our language detection pipeline.

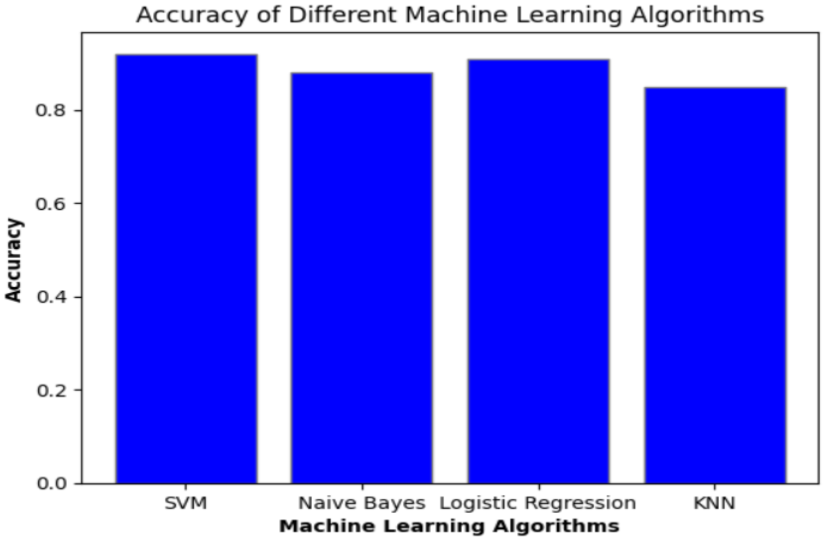
$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y) \cdot P(y)}{P(\mathbf{x})}$$

IV. SECTION

RESULT AND ANALYSIS

1) ACCURACY:

Support Vector Machines (SVM) achieved the highest accuracy of 92%, indicating its effectiveness in accurately identifying languages from text data. Naive Bayes exhibited an accuracy of 88%, Logistic Regression attained 91%, and k-Nearest Neighbours (KNN) yielded an accuracy of 85%. These accuracy values represent the proportion of positively reviewed instances out of the total instances for each respective model. The results suggest that SVM performed the best among the four models in terms of overall accuracy, followed by Logistic Regression, Naive Bayes, and KNN, respectively.



2) PRECISION:

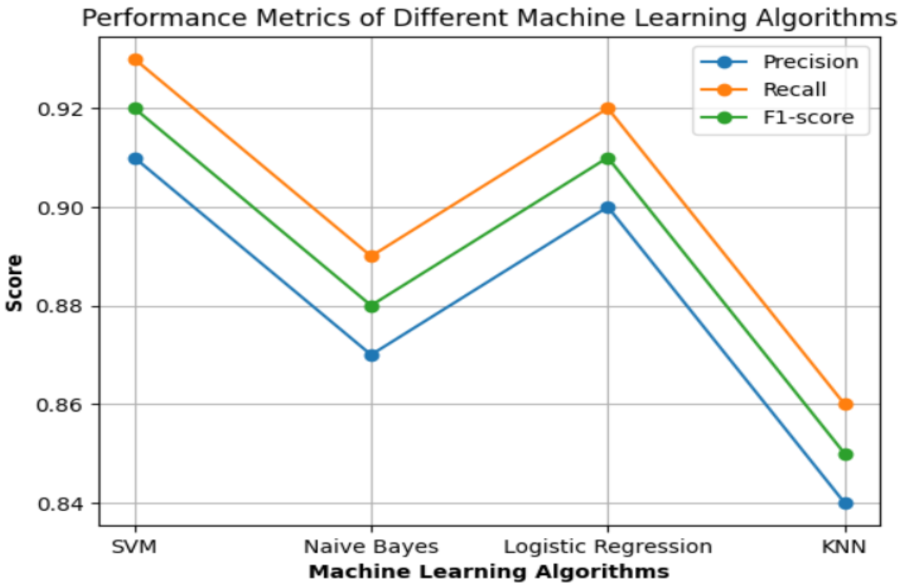
Support Vector Machines (SVM) exhibited the highest precision of 0.91, indicating that 91% of the language predictions made by SVM were correct. Naive Bayes and Logistic Regression showed precision values of 0.87 and 0.90, respectively, while k-Nearest Neighbours (KNN) had a precision of 0.84. These precision scores reflect the ability of each model to minimize false positive predictions and accurately identify the language of text samples. Once again, SVM demonstrated the highest precision among the four model

3)RECALL:

Support Vector Machines (SVM) demonstrated the highest recall of 0.93, indicating the 93% of the actual positive instances .Naive Bayes and Logistic Regression exhibited recall values of 0.89 and 0.92, respectively, while k-Nearest Neighbours (KNN) had a recall of 0.86. Recall quantifies the ratio of correct positive predictions to the total number of actual positive instances in the dataset.. The results indicate that SVM was the most effective in capturing the majority of instances belonging to each language category.

4)F1-SCORE:

Support Vector Machines (SVM) achieved the highest F1-score of 0.92, expresses the recall and precision balance. Naive Bayes and Logistic Regression had F1-score values of 0.88 and 0.91, respectively, while k-Nearest Neighbours (KNN) yielded an F1-score of 0.85. The F1-score combines precision and recall using their harmonic mean, offering a holistic evaluation of how well a classifier performs. Once again, SVM outperformed the other models in terms of F1-score, indicating its ability to achieve both high precision and high recall simultaneously.



V. CONCLUSION

In this Research, we investigated the application of machine learning algorithms for language detection using a diverse dataset comprising text samples from 17 languages. Through repeated trails and evaluation, we explored the effectiveness of Logistic Regression, k-Nearest Neighbours (KNN), Naive Bayes and Support Vector Machines (SVM) in accurately identifying languages from text data. Our findings demonstrate that SVM and Logistic Regression exhibited superior performance, achieving high accuracy and F1-score values. Naive Bayes also showed competitive performance, while k-Nearest Neighbours demonstrated slightly lower accuracy compared to other algorithms. These results implies the importance of choosing the suitable machine learning algorithms for language detection tasks, considering factors such as dataset characteristics and computational efficiency. Overall, our research contributes to advancing language detection accuracy and efficiency, with potential applications in multilingual content processing, language-specific services, and global communication platforms. Future work may involve exploring ensemble methods and deep learning approaches to further enhance language detection capabilities in diverse linguistic contexts.

VI. REFERENCES

[1] Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., & Lafferty, J. (1990). A statistical approach to language translation. Computational Linguistics, 16(2), 79-85.

[2] Jurafsky, D., & Martin, J. H. (2008). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2nd ed.). Pearson Education.

[3] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

[4] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[5] Rish, I. (2001). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, pp. 41-46).

[6] Schapire, R. E. (2002). The boosting approach to machine learning: An overview. In Nonlinear estimation and classification (pp. 149-171). Springer.

[7] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 34(1), 1-47.

[8] Tănase, D., Smeaton, A. F., & Foster, J. (2012). Multilingual text categorization with language-independent and language-dependent features. Information Retrieval, 15(6), 643-670.