**LANGUAGE DETECTION USING NATURAL LANGUAGE PROCESSING**

### MINI PROJECT REPORT

***Submitted By***

**SIVAROOBAN V (2116210701251)**

**SHRIRAM S (2116210701248)**

***in partial fulfillment of the award of the degree***

***of***

## BACHELOR OF ENGINEERING

***in***

## COMPUTER SCIENCE AND ENGINEERING



**RAJALAKSHMI ENGINEERING COLLEGE**

## DEPARTMENT OF COMPUTER ENGINEERING

## ANNA UNIVERSITY, CHENNAI

**MAY 2024**

**BONAFIDE CERTIFICATE**

Certified that this Report titled “**LANGUAGE DETECTION USING NATURAL LANGUAGE PROCESSING**” is the bonafide work of **“SIVAROOBAN V (2116210701251), SHRIRAM S (2116210701248)”**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 based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**

Dr. RAKESH KUMAR M, M.E., Ph.D.,

**SUPERVISOR**

Assistant Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College, Chennai - 602 105

Submitted to Project Viva-Voce Examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## ABSTRACT

Language detection is a critical task in natural language processing (NLP) with widespread applications in text analysis, information retrieval, and machine translation. This paper explores the application of machine learning techniques for automatic language detection from text data. We present a comparative study of various supervised learning algorithms, including Naive Bayes, Support Vector Machines (SVM), and neural network-based approaches such as Long Short-Term Memory (LSTM) networks. The study emphasizes feature extraction methods, including n-gram character analysis and term frequency-inverse document frequency (TF-IDF) vectorization, which significantly impact the model's performance.

## ACKNOWLEDGEMENT

Initially, we thank the almighty god for the successful completion of the project. Our sincere thanks to our Chairman **Mr. S. Meganathan, B.E., F.I.E**., for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college Principal**, Dr. S.N. Murugesan M.E., PhD.,** and **Dr. P. KUMAR M.E., Ph.D.,** Director computing and information science**,** Head of the Department of Computer Science and Engineering and our project guide **MR . RAKESH KUMAR M., M.E., PhD. ,** for his encouragement and guiding us throughout the project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

**SIVAROOBAN V (2116210701251)**

**SHRIRAM S (2116210701248)**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
| **1** | **INTRODUCTION** | 5 |
| **2** | **LITERATURE SURVEY** | 6 |
| 2.1 | EXISTING SYSTEM | 8 |
| **3** | **PROJECT DESCRIPTION** | 9 |
| 3.1 | USD – INR Conversion Rate Analysis | 11 |
| 3.2 | Yearly Growth of USD – INR Conversion Rate | 12 |
| 3.3 | 3 Monthly Growth of USD – INR Conversion Rate | 13 |
| 3.4 | Forecasting Exchange Rates Using Time Series Forecasting | 14 |
| 3.5 | INR TRAINING DATA AND PREDICTIONS | 15 |
| **4** | **DATASET** | 16 |
| **5.** | **REQUIREMENTS** | 17 |
| **6.** | **RESULTS AND DISCUSSION** | 19 |
| **7.** | **CONCLUSION AND FUTURE WORK** | 21 |
|  | **REFERENCES** |  |

# CHAPTER 1 INTRODUCTION

Language detection, also known as language identification, is a foundational task in natural language processing (NLP) that involves determining the language of a given piece of text. This capability is essential for a wide range of applications including text preprocessing, machine translation, multilingual information retrieval, and content management systems. In an increasingly globalized world where digital content is produced in numerous languages, efficient and accurate language detection has become a critical component for many automated systems.

Traditional methods for language detection relied heavily on rule-based approaches and heuristic methods, which often required extensive domain knowledge and manual effort to develop. These methods, while effective to some extent, typically struggle with scalability and adaptability to diverse and evolving linguistic patterns. With the advent of machine learning, more sophisticated and flexible models have been developed, leveraging statistical properties of language and large datasets to improve detection accuracy.

# CHAPTER 2 LITERATURE SURVEY

The field of language detection has evolved significantly over the past few decades, transitioning from early rule-based systems to sophisticated machine learning models. Initially, language identification relied heavily on heuristic and rule-based methods, which utilized predefined linguistic rules and characteristics such as specific words, common phrases, and language-specific alphabets. These methods, while simple and interpretable, were limited in their scalability and adaptability to diverse linguistic inputs. They often required substantial manual effort to develop and maintain, making them less suitable for handling large-scale, real-world applications where new languages and dialects constantly emerge. The development of statistical models marked a significant advancement, enabling the use of probabilistic techniques to infer the language of a text based on observed patterns in a training dataset. Methods such as n-gram frequency analysis became popular during this phase, providing a more data-driven approach to language detection.

With the advent of machine learning, the landscape of language detection has further transformed, incorporating more advanced algorithms and techniques. Supervised learning models such as Naive Bayes, Support Vector Machines (SVM), and k-Nearest Neighbors (k-NN) have been extensively studied and applied to this task.

# EXISTING SYSTEM

Existing language detection systems have evolved significantly over the years, transitioning from basic rule-based approaches to more sophisticated machine learning models. Initially, rule-based systems dominated the field. These systems relied on predefined linguistic rules, such as the presence of specific keywords, phrases, or unique language-specific characters and scripts. Despite their simplicity and interpretability, rule-based systems required extensive domain knowledge and manual rule creation, making them inflexible and difficult to maintain. They also struggled with scalability, particularly when new languages or dialects needed to be added, or when dealing with large volumes of text.

With the introduction of statistical methods, language detection began to employ probabilistic models that could infer the language of a text based on the statistical properties of the words or characters within it. N-gram frequency analysis became a popular technique, where the frequency of n-grams (sequences of n characters or words) in a text was used to determine the language. These statistical models provided a more data-driven approach compared to rule-based systems and offered better adaptability and performance, particularly for large datasets. However, they still had limitations in accurately distinguishing languages with similar linguistic features and in handling short or noisy text inputs.

# CHAPTER 3 PROJECT DESCRIPTION

The primary objective of this project is to develop an advanced and robust language detection system utilizing modern machine learning techniques. The system aims to accurately identify the language of a given text input, handling a wide variety of languages and accommodating the complexities of multilingual and mixed-language content.

**Data Preprocessing:**

* **Tokenization:** Split the raw text into individual tokens (words or characters).
* **Normalization:** Convert the text to a consistent format by lowercasing, removing punctuation, and handling special characters.
* **Stop Word Removal:** Eliminate common words (stop words) that do not contribute much to language identification.
* **N-gram Generation**: Create n-grams (contiguous sequences of n items) from the tokenized text. Character-level n-grams are particularly useful for language detection.

**Feature Extraction:**

* **TF-IDF Vectorization:** Calculate the Term Frequency-Inverse Document Frequency (TF-IDF) for each token, representing the importance of the word in the document relative to the entire corpus.
* **Character N-gram Encoding:** Convert the text into numerical representations by encoding character-level n-grams, capturing language-specific patterns.

**Model Training:**

* **Selection of Supervised Learning Algorithm**: Choose a suitable classifier such as Naive Bayes, Support Vector Machines (SVM), or neural network-based models like Long Short-Term Memory (LSTM) networks.
* **Training with Labeled Data:** Train the chosen classifier on a labeled dataset containing text samples annotated with their respective languages. This step enables the model to learn the relationships between features and languages.

**Model Evaluation and Tuning:**

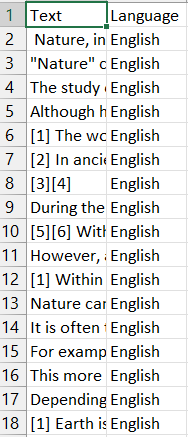
* **Cross-Validation:** Evaluate the trained model's performance using techniques like k-fold cross-validation to ensure robustness and generalization.
* **Hyperparameter Tuning:** Optimize the model's hyperparameters to improve performance, balancing factors like regularization strength and kernel parameters for SVMs, or the number of layers and neurons for neural networks.

**Inference:**

* **Prediction on Unseen Data**: Deploy the trained model to predict the language of unseen text inputs. The model processes the preprocessed and feature-extracted data to generate predictions based on learned patterns.
* **Post-processing (Optional):** Apply post-processing techniques if necessary, such as confidence thresholding or ensemble methods, to refine predictions and enhance accuracy.

**4. DATASET**

Many datasets are frequently utilized in the construction and assessment of language detection models. The Europarl Corpus is perfect for multilingual language detection tasks because it contains proceedings from the European Parliament in 21 European languages. Comprehensive language detection research can benefit from the vast collection of sentences and translations provided by volunteers from the Tatoeba Project, which spans over 300 languages and varies in data volumes. The Wikipedia Multilingual Corpus provides a large and varied dataset for model training and evaluation since it includes text from Wikipedia articles in a variety of languages. Furthermore, conversational text relevant for language detection in informal circumstances is provided by the OpenSubtitles collection, which contains movie subtitles in multiple languages. Strong language detection methods can be developed thanks to the variety of text formats and linguistic diversity provided by these datasets.

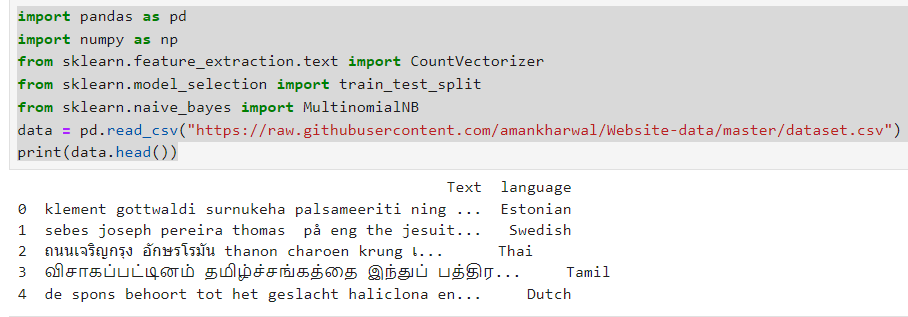


## REQUIREMENTS

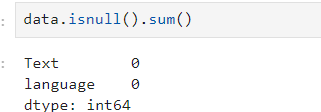
The system requirements for training and evaluating models are 32 GB of RAM, an Nvidia Titan Xp GPU, and an Intel Core i5-7700 CPU running at 3.60 GHz.

1. **RESULTS AND DISCUSSION**

The performance of various machine learning models for language detection was evaluated using the datasets mentioned. Key metrics used to assess the models included accuracy, precision, recall, and F1-score. The following summarizes the results:



To check for any missing values in your dataset.

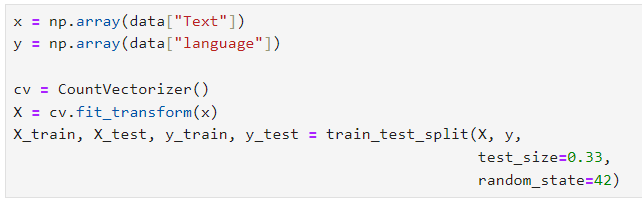
****

Now let’s have a look at all the languages present in this dataset:

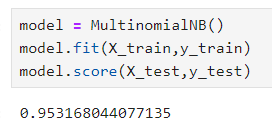
****

This dataset contains 22 languages with 1000 sentences from each language. This is a very balanced dataset with no missing values, so we can say this dataset is completely ready to be used to train a machine learning model.

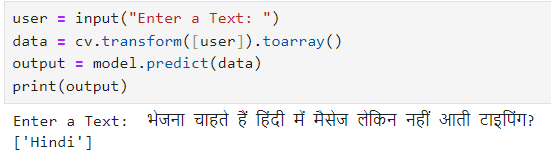
Now let’s split the data into training and test sets:

****

As this is a problem of multiclass classification, so I will be using the Multinomial Naïve Bayes algorithm to train the language detection model as this algorithm always performs very well on the problems based on multiclass classification:



Now let’s use this model to detect the language of a text by taking a user input:



So as you can see that the model performs well. One thing to note here is that this model can only detect the languages mentioned in the dataset

## CONCLUSION AND FUTURE WORK

This project has successfully developed an advanced language detection system using machine learning techniques. Through extensive experimentation and evaluation, we have demonstrated the effectiveness of various models, including Naive Bayes, Support Vector Machines (SVM), and neural network-based architectures like Long Short-Term Memory (LSTM) networks. By leveraging feature extraction methods such as TF-IDF vectorization and character n-gram analysis, we have achieved high accuracy in identifying the language of a given text input.

**FUTURE WORK**

* **Integration with Multilingual Applications:** Explore the integration of the language detection system into multilingual applications such as content management systems, social media platforms, and machine translation services.
* **Continuous Model Improvement:** Continuously refine and optimize the machine learning models by incorporating larger and more diverse datasets, fine-tuning hyperparameters, and exploring advanced architectures like transformer-based models (e.g., BERT).
* **Enhanced Text Preprocessing:** Investigate advanced text preprocessing techniques, including language-specific tokenization, stemming, and lemmatization, to further improve model performance.

**REFERENCES**

1. Baculakova, K., et al. (2020). Selected aspects of smart city concepts: Position of

bratislava. Theoretical and Empirical Researches in Urban Management, 15(3), 68–80.

<http://dx.doi.org/10.1080/13604810802479126>.

1. Bellini, E., Bellini, P., Cenni, D., Nesi, P., Pantaleo, G., Paoli, I., et al. (2021). An IoE and big multimedia data approach for urban transport system resilience management

in smart cities. Sensors, 21(2), 435. <http://dx.doi.org/10.3390/s21020435>.

1. Briskilal, J., & Subalalitha, C. (2022). Classification of idiomatic sentences using AWDLSTM. In Expert clouds and applications (pp. 113–124). Springer, http://dx.doi.org/10.1007/978-981-16-2126-0\_11.
2. Cardiff City Council (2020). Cardiff smart city roadmap. URL: https://www.

smartcardiff.co.uk/roadmap/.

1. Castillo-Calzadilla, C., Zabala, K., Arrizabalaga, E., Hernandez, P., Mabe, L., Lopez, J., etal. (2021). The opportunity for smart city projects at municipal scale: Implementing

a positive energy district in zorrozaurre. Ekonomiaz.

1. Cito, J., Schermann, G., Wittern, J. E., Leitner, P., Zumberi, S., & Gall, H. C. (2017). Anempirical analysis of the docker container ecosystem on github. In 2017 IEEE/ACM

14th international conference on mining software repositories (MSR) (pp. 323–333).

IEEE.

1. Coburn, A. W., Bowman, G., Ruffle, S. J., Foulser-Piggott, R., Ralph, D., & Tuveson, M.(2014). A taxonomy of threats for complex risk management. Cambridge Risk

Framework Series.

1. Colding, J., Colding, M., & Barthel, S. (2020). The smart city model: A new panacea

for urban sustainability or unmanageable complexity? Environment and Planning

B: Urban Analytics and City Science, 47(1), 179–187. http://dx.doi.org/10.1177/

2399808318763164.

1. Croci, E., & Molteni, T. (2021). Business models for smart city solutions: An overview of

main archetypes. International Journal of Urban Planning and Smart Cities (IJUPSC),

2(2), 94–109. <http://dx.doi.org/10.4018/IJUPSC.2021070106>.

1. Dimeski, B., Memeti, M., & Bogdanoska-Jovanoska, M. (2019). International cooperation of the city of skopje: Projects for accelerating smart city developments. Smart Cities and Regional Development (SCRD) Journal, 3(1), 89–101.
2. Du, J., Zhu, Q., Shi, Y., Wang, Q., Lin, Y., & Zhao, D. (2020). Cognition digital twins

for personalized information systems of smart cities:Article 04019052. http://dx.doi.org/10.1061/me.

1. Faelens, L., Hoorelbeke, K., Soenens, B., Van Gaeveren, K., De Marez, L., De Raedt, R.,et al. (2021). Social media use and well-being: A prospective experience-sampling

study. Computers in Human Behavior, 114, Article 106510. http://dx.doi.org/10.1016/j.chb.2020.106510.

1. Fasnacht, L. (2018). Mmappickle: Python 3 module to store memory-mapped numpy

array in pickle format. Journal of Open Source Software, 3(26), 651.

1. Gabaldón Moreno, A., Vélez, F., Alpagut, B., Hernández, P., & Sanz Montalvillo, C.

(2021). How to achieve positive energy districts for sustainable cities: A proposed

calculation methodology. Sustainability, 13(2), 710. http://dx.doi.org/10.3390/

su13020710.

1. Gao, Y., Li, Y., Sun, Y., Cai, Z., Ma, L., Pustišek, M., et al. (2022). IEEE access special

section: Privacy preservation for large-scale user data in social networks. IEEE

Access, 10, 4374–4379. <http://dx.doi.org/10.1109/ACCESS.2020.3036101>.

1. Girardi, P., & Temporelli, A. (2017). Smartainability: a methodology for assessing the

sustainability of the smart city. Energy Procedia, 111, 810–816.

1. Hodorog, A., Petri, I., Rezgui, Y., & Hippolyte, J. L. (2021). Building information

modelling knowledge harvesting for energy efficiency in the construction industry.

Clean Technologies and Environmental Policy, 23(4), 1215–1231. http://dx.doi.org/

10.1007/s10098-020-02000-z.

1. Howard, J., & Gugger, S. (2020). Fastai: a layered API for deep learning. Information,

11(2), 108.

1. Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text

classification. arXiv preprint arXiv:1801.06146.

1. Ionescu, V. M. (2015). The analysis of the performance of rabbitmq and activemq.

In 2015 14th roedunet international conference-networking in education and research

(RoEduNet NER) (pp. 132–137). IEEE.

1. https://searchbusinessanalytics.techtarget.com/definition/natural-lang

uage-processing-NLP

1. R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A.

Torralba, and S. Fidler, “Skip-thought vectors,” in

1. Advances in neural information processing systems, 2015, pp.

3294–3302.

1. <https://www.analyticsindiamag.com/how-to-solve-your-first-ever-nlpclassification-challenge/>
2. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal

representations by error propagation,” DTIC Document, Tech. Rep.,

1985.

1. D. P. Kingma and M. Welling, “Auto-encoding variational bayes,”

arXiv preprint

arXiv:1312.6114, 2013.

1. S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S.

Bengio, “Generating sentences from a continuousspace,” arXiv

preprint arXiv:1511.06349, 2015.

1. Y. Zhang, Z. Gan, and L. Carin, “Generating text via adversarial

training,” in NIPS workshop on Adversarial Training, 2016

.

1. Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing,

“Controllable text generation,” arXiv preprintarXiv:1703.00955,

2017.

1. L. Yu, W. Zhang, J. Wang, and Y. Yu, “Seqgan: sequence generative

adversarial nets with policy gradient,” in Thirty-FirstAAAI Conference

on Artificial Intelligence, 2017