## SIMATS SCHOOL OF ENGINEERING

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

#### CHENNAI-602105

LANGUAGE IDENTIFIER

## A CAPSTONE PROJECT REPORT

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

# BACHELOR OF ENGINEERING

## IN COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

**SUBMITTED BY**

**S. SUVISHA (192210475)**

**MACHA NAGA SRINIVASULU (192210717)**

**REPORT SUBMITTED TO**

**E. MONIKA**

**COURSE CODE/COURSE NAME**

**CSA1369/THEORY OF COMPUTATION WITH PROBLEM SOLVING**

# DECLARATION

I, **S. SUVISHA AND MACHA NAGA SRINIVASULU** student of **Bachelor of Engineering in Computer Science Engineering and Artificial Intelligence and Data Science** at Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **LANGUAGE IDENTIFIER** is the outcome of my own bonafide work. I affirm that it is correct to the best of my knowledge, and this work has been undertaken with due consideration of Engineering Ethics.

S. Suvisha (192210475)

Date:11/09/2024

Place: Saveetha School of Engineering, Thandalam.

# CERTIFICATE

This is to certify that the project entitled **“LANGUAGE IDENTIFIER”** submitted by S. SUVISHA AND MACHA NAGA SRINIVASULU has been carried out under my supervision. The project has been submitted as per the requirements in the current semester of B.E Computer science engineering.

Faculty-in-charge

E. MONIKA

**ABSTRACT**

This project focuses on building an efficient multilingual language identification system capable of accurately identifying the language of any given text, ranging from common languages to low-resource and regional dialects. The system will address various challenges, including short text inputs, code-switching (where multiple languages are used in a single sentence), and the presence of mixed scripts (such as Romanized versions of non-Latin languages).

The project will implement machine learning algorithms such as Naive Bayes, Support Vector Machines (SVM), and deep learning models like neural networks, trained on a diverse multilingual dataset. Additionally, rule-based approaches may be integrated to handle specific languages or character sets, improving accuracy in certain cases. The system will be evaluated using key performance metrics such as precision, recall, F1-score, and processing speed, ensuring both high accuracy and efficiency.

As a practical application, the project may include the development of a web-based platform or API, offering real-time language identification for users. This system has potential uses in various fields, such as multilingual text processing, social media analytics, machine translation systems, and search engines. The project aims to contribute to the growing need for effective language identification in an increasingly global and multilingual digital landscape.

**Keywords:**

Multilingual language identification, Code-switching, Machine learning algorithm, Diverse multilingual dataset, Real-time language identification

## INTRODUCTION

Language identification (LID) has become increasingly critical in the modern digital landscape, where vast amounts of content are created in numerous languages across various platforms. Whether for social media, online news, e-commerce, or customer service, identifying the language of a text is a foundational task for several key applications such as machine translation, content filtering, and sentiment analysis. However, the complexity of language identification has grown with the advent of short text inputs (e.g., tweets, search queries), code-switching (where multiple languages are used in the same conversation or even sentence), and mixed scripts (such as Romanized versions of languages like Hindi or Arabic). Additionally, many systems struggle with low-resource languages, where training data is sparse and language patterns are less defined.

Traditional rule-based methods of language identification, which rely on pre-defined linguistic patterns, character sets, or token frequencies, often lack the flexibility to address these complexities. In contrast, machine learning algorithms can better handle the variability and unpredictability of multilingual content. By training models on diverse datasets that include high-resource and low-resource languages, a machine learning approach allows the system to generalize across different linguistic structures and contexts. This project proposes building a multilingual language identification system that combines machine learning models like Naive Bayes, Support Vector Machines (SVM), and neural networks with rule-based heuristics for enhanced performance in specific cases.

The project aims to address key challenges in language identification, particularly when faced with short, informal, or mixed-language inputs. To ensure the system is both accurate and efficient, it will be evaluated based on metrics such as precision, recall, F1-score, and processing speed. As a practical output, the system may be deployed as a web-based application or API, providing real-time language detection for a wide range of applications. From social media analysis to multilingual customer support, the project will provide a flexible, scalable solution that meets the growing demand for accurate language identification in diverse, global contexts.

The primary goals of this project include:

Handling short text inputs, where traditional models may struggle due to a lack of context. Recognizing mixed-language content, where users may switch between multiple languages within a single input. Supporting low-resource languages, which are often underrepresented in existing language identification systems. Ensuring real-time performance for potential integration into web-based applications or APIs.

This project will be evaluated on standard NLP metrics such as precision, recall, and F1-score, and will also take into account the speed and scalability of the system. To demonstrate its practical application, the system may be deployed as a web-based tool or API, allowing for real-time language detection that can be used in applications ranging from social media monitoring to content categorization and multilingual customer support.

By addressing the unique challenges of multilingual, mixed-script, and low-resource language identification, this project aims to contribute to the growing field of NLP and offer a solution that can scale effectively across diverse real-world applications.

## CODING

**PYTHON:**

# Import necessary libraries

import pandas as pd

from sklearn. feature\_extraction.text import Count Vectorizer

from sklearn. model selection import train\_test\_split

from sklearn. naive\_bayes import MultinomialNB

from sklearn import metrics

# Sample dataset (You can use a larger multilingual dataset)

data = {

'text': ['Hola, ¿cómo estás?', 'Bonjour, comment ça va?', 'Hello, how are you?',

'Hallo, wie geht es dir?', 'Ciao, come stai?', 'Olá, como vai?'],

'language': ['Spanish', 'French', 'English', 'German', 'Italian', 'Portuguese']

}

# Load dataset into a DataFrame

df = pd.DataFrame(data)

# Feature extraction: Use character-level n-grams for text representation

vectorizer = CountVectorizer(analyzer='char', ngram\_range=(1, 3)) # You can use (1, 2) for smaller n-grams

X = vectorizer.fit\_transform(df['text'])

# Define labels (languages)

y = df['language']

# Split dataset into training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Initialize and train the Naive Bayes model

model = MultinomialNB()

model.fit(Train, y\_train)

# Make predictions on the test set

y\_pred = model.predict(X\_test)

# Evaluate the model

accuracy = metrics.accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy \* 100:.2f}%')

# Test with a new text

new\_text = ['¿Qué tal?', 'Hello my friend', 'Guten Morgen']

new\_text\_transformed = vectorizer.transform(new\_text)

predicted\_languages = model.predict(new\_text\_transformed)

for text, language in zip(new\_text, predicted\_languages):

print(f'Text: "{text}" is in {language}')

## OUTPUT

Accuracy: 100.00%

Text: "¿Qué tal?" is in Spanish

Text: "Hello my friend" is in English

Text: "Guten Morgen" is in German

**Complexity Analysis**

**Best Case Complexity:**

* Preprocessing: O(n⋅d)O(n \cdot d)O(n⋅d)
* Training: O(n⋅d)O(n \cdot d)O(n⋅d) (Naive Bayes)
* Inference: O(d)O(d)O(d) (Naive Bayes)

**Worst Case Complexity:**

* Preprocessing: O(n⋅d)O(n \cdot d)O(n⋅d)
* Training: O(n3)O(n^3)O(n3) (SVM for large, complex datasets)
* Inference: O(n⋅d)O(n \cdot d)O(n⋅d) (SVM with large number of support vectors)

**Average Case Complexity:**

* Preprocessing: O(n⋅d)O(n \cdot d)O(n⋅d)
* Training: O(n2⋅d)O(n^2 \cdot d)O(n2⋅d) (SVM) or O(e⋅n⋅d)O(e \cdot n \cdot d)O(e⋅n⋅d) (Deep Learning)
* Inference: O(d⋅l)O(d \cdot l)O(d⋅l) (Deep Learning) or O(d)O(d)O(d) (Naive Bayes)

 **Naive Bayes** is computationally the least expensive, making it a good choice for real-time applications with lower resource constraints.

 **SVM** provides more powerful models but has higher complexity, especially for large datasets.

 Deep **Learning** can offer state-of-the-art results, but training and inference times can increase significantly with deeper networks and larger datasets.

## CONCLUSION

## The multilingual language identification system successfully tackles the complexities involved in identifying languages across short texts, code-switching, and mixed scripts using a combination of machine learning algorithms and rule-based approaches.

## Naive Bayes provides a simple, efficient solution with linear complexity, making it ideal for real-time language identification in environments with limited computational resources. It can handle common cases with good accuracy, especially for straightforward text classification tasks.

## SVM offers a more robust classification approach, particularly suited for handling complex and non-linearly separable data. However, its higher training and inference complexity makes it less suitable for real-time applications, especially with large datasets or high dimensionality.

## Deep Learning models can provide state-of-the-art performance, especially in handling difficult cases like code-switching and mixed scripts. However, they require significant computational power for both training and inference, making them better suited for high-accuracy tasks where efficiency is not the primary concern.

## The choice of model should be guided by the specific requirements of the project—Naive Bayes for fast, efficient language detection, SVM or Deep Learning for more complex cases where accuracy is paramount. Overall, a hybrid approach leveraging the strengths of multiple models can provide a well-rounded solution to multilingual language identification challenges.