DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE

M.B.E. SOCIETY'S COLLEGE OF ENGINEERING AMBAJOGAI

PROJECT REPORT

"Language Detection "

SUBMITTED BY

Shinde Omnath

Shaikh Mastan

Maske Nikhil

GUIDED BY

Prof. S.V. Kulkarni

Department Of Computer Science and Engineering

2022-23

M.B.E.S

COLLEGE OF ENGGINEERING, AMBAJOGAI

Department Of

Computer Science and Engineering

CERTIFICATE

This is to certify Students of (Computer Science and Engineering) TY **Shinde Omnath , Shaikh Mastan & Nikhil Maske** have Completed Mini project-II report on "**Language Detection** " . In this volume we submitted a satisfactory report on the Mini project-II, in the academic year 2022-2023 even semester.

Guide
Prof. S.V.Kulkarni

H.O. D
Prof.S.V.Kulkarni

Principal
Dr.B.I Khadakbhavi

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to **Prof. S.V. Kulkarni**, my project mentor, for their continuous guidance, support, and valuable insights throughout the project. Their expertise in Natural Language Processing (NLP) has been instrumental in shaping the direction of this project.

I am also thankful to, for providing access to resources and facilities that were essential for the successful completion of this project.

My sincere appreciation goes to all the members of the **Omnath Shinde, Shaikh Mastan & Nikhil Maske** whose constructive feedback and collaborative efforts have significantly contributed to the project's progress.

I am grateful to the authors of various research papers, articles, and online resources, whose work has been a constant source of knowledge and inspiration throughout the project.

Last but not least, I extend my thanks to my family and friends for their unwavering support and encouragement, which kept me motivated throughout this project journey.

Your support and encouragement have been invaluable in making this project a reality. Thank you all for being a part of this endeavor!

Index

Introduction:

- Language detection, also known as language identification, is a fundamental problem in natural language processing (NLP) and plays a crucial role in various applications, such as multilingual content filtering, text-to-speech synthesis, and machine translation.

- The task of language detection involves automatically determining the language of a given piece of text without any prior information about its origin.

- The objective of this project is to develop an accurate and efficient language detection system using the Naive Bayes classifier.

- The **Naive Bayes algorithm** is a probabilistic method based on Bayes' theorem and has proven to be effective for text classification tasks, including language detection.

- This report outlines the steps taken to build and evaluate the language detection model.

Objective of Project:

- The primary objective of the project is to develop a robust and accurate language detection system using machine learning techniques.

- The main focus is to design and implement a model that can automatically identify the language of a given text with a high level of precision and efficiency.

- The language detection system aims to provide reliable predictions for various languages, making it valuable for multilingual content filtering, language-specific analysis, and other language-related tasks.

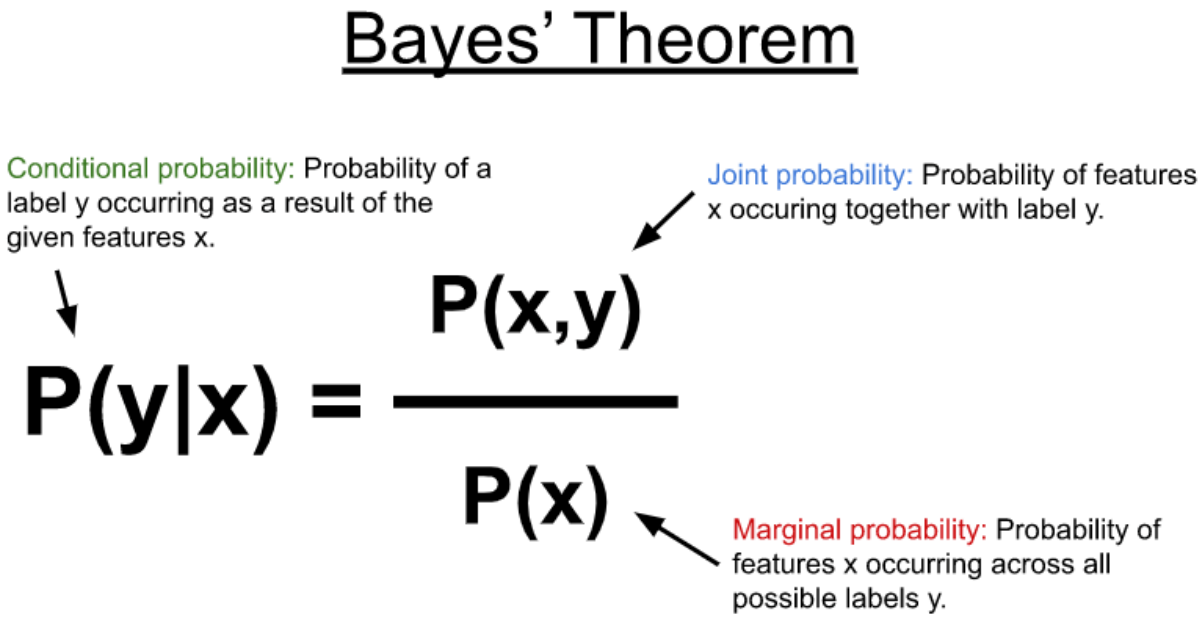
- The successful accomplishment of the primary objective will result in a practical and usable tool that can be integrated into real-world applications requiring automatic language identification.

Multinomial Naive Bayes (MNB) Algorithm

- The Multinomial Naive Bayes (MNB) classifier is a probabilistic machine learning algorithm commonly used for text classification tasks, such as language detection, spam filtering, sentiment analysis, and document categorization.

Working Principle:

- The MNB classifier is based on the Bayes theorem, which calculates the probability of a particular event (class label) given the occurrence of certain features (words in the text). It assumes that all features are conditionally independent, which means the presence of one feature does not affect the presence of others (a "naive" assumption).



- In the context of language detection, the MNB classifier calculates the probability of each language label given the occurrence of specific words in a given text. The class with the highest probability is assigned as the predicted language label for the text.

Model Representation:

- In the MNB classifier, each feature (word) is represented by a positive integer count that indicates the number of times the word appears in the text sample.

- The features are typically represented using a bag-of-words model, which creates a numerical vector for each text sample, counting the occurrences of words from a predefined vocabulary.

- It is a variant of the Naive Bayes algorithm that is specifically designed for handling discrete features, making it well-suited for working with count data like word frequencies in bag-of-words representations.

Model Training:

- During training, the MNB classifier learns the probability distribution of words for each language class based on the training data.
- It estimates the prior probability of each class (P(C)) and the likelihood of each word occurring in a particular class (P(Wi|C)).
- These probabilities are calculated from the training data using maximum likelihood estimation.

Model Prediction:

- Once the MNB classifier is trained, it can predict the language of new text samples by calculating the probability of each class given the word frequencies in the text.
- The class with the highest probability is selected as the predicted language for the input text.

Advantages of MNB Classifier:

1) **Efficiency:**
   The MNB classifier is computationally efficient and requires relatively little training time, making it suitable for large datasets.

2) **Handling Discrete Features:**
   It is specifically designed for handling discrete features like word counts, which is common in text data.

3) **Suitable for High-Dimensional Data:** The MNB classifier performs well even with high-dimensional data, where the number of features (words) can be large.

Limitations of MNB Classifier:

1) **Assumption of Feature Independence:**
   The MNB classifier assumes that features are conditionally independent, which may not hold true for all text data, leading to potential inaccuracies.

2) **Lack of Contextual Information:**
   As the classifier treats words independently, it may not capture the semantic and contextual information present in the text.

3) **Sensitive to Out-of-Vocabulary Words:**
   The model may struggle with words not present in the training data, affecting the accuracy of predictions for unseen words or rare language samples.

Despite its simplifying assumptions, the MNB classifier is widely used in text classification tasks due to its ease of implementation, efficiency, and often satisfactory performance, especially for applications like language detection, where bag-of-words representations are prevalent

Implementation:

1. **Data Preprocessing:**

a) **Load Dataset:**
The dataset "**Language Detection.csv**" is imported using the Pandas library. The "Text" column contains the input text samples, and the corresponding language labels are stored in the "Language" column.

b) **Text Preprocessing:**
To prepare the text data for machine learning, various preprocessing steps are applied:

i. **Symbol and Number Removal:**
Regular expressions are used to remove special characters, symbols, and numeric digits from the text. This step eliminates irrelevant noise and ensures that only relevant language features remain.

ii. **Text Lowercasing:**
All text samples are converted to lowercase to achieve uniformity and to avoid case sensitivity issues in language detection.

iii. **Tokenization:**
The text is tokenized into individual words using **CountVectorizer from scikit-learn**. CountVectorizer creates a bag-of-words representation, where each word is assigned a numerical count for each text sample. This step converts the text data into a numerical format suitable for training the model.

2. **Model Selection and Training:**

- **Multinomial Naive Bayes Classifier:**

  For language detection, the Multinomial Naive Bayes classifier is chosen due to its simplicity and effectiveness with count data like the bag-of-words representation.

- **Data Splitting:**

  - The dataset is split into a training set and a testing set using the **train_test_split** function from scikit-learn.
  - The training set contains a portion of the data (usually 80%), which is used to train the Naive Bayes model.
  - The testing set contains the remaining data (usually 20%), which is used to evaluate the model's performance.

- **Model Training:**

  - The Multinomial Naive Bayes classifier is trained on the training data using the **fit ()** method.
  - The model learns the language patterns and associations from the bag-of-words representation of the text samples.

3. Model Evaluation:

- **Accuracy Score:**

  - After training the model, its performance is evaluated on the testing set. The accuracy score is calculated using the **accuracy_score** function from scikit-learn.
  - The accuracy score represents the percentage of correct language predictions made by the model on the test data.
  - It measures how well the model performs in identifying the correct language for the given text samples.

- **Confusion Matrix:**

  - In addition to the accuracy score, a confusion matrix is generated using the **confusion_matrix** function from scikit-learn.
  - The confusion matrix provides detailed insights into the model's predictions. It displays the number of true positive, true negative, false positive, and false negative predictions for each language.
  - The confusion matrix helps assess the model's performance on individual language classes.

4. **Language Predict Function**

- **Prediction Function:**
  - To enable language prediction for any given text, a function named "predict" is created.
  - This function takes a text input as a parameter and uses the trained model and CountVectorizer to predict the language of the text.
  - The input text is first preprocessed to match the format used during model training.
  - Then, the CountVectorizer is applied to convert the text into a numerical representation.
  - Finally, the Multinomial Naive Bayes model predicts the language, and the function returns the predicted language label.

5. Visualization:

- **Heatmap:**
  - For better visualization of the confusion matrix, a **heatmap is created using the seaborn library**.
  - The heatmap provides a color-coded representation of the confusion matrix, making it easier to interpret the model's performance for different language classes.

Result:

The language detection system produced the following results:

Input:

I love programming, Python is my favorite language.

أحب البرمجة ، بايثون هي لغتي المفضلة.

我喜欢编程，Python 是我最喜欢的语言。

Me encanta programar, Python es mi lenguaje favorito.

Eu amo programar, Python é minha linguagem favorita.

Output:

English

Arabic

Chinese (Simplified)

Spanish; Castilian

Portuguese

Conclusion:

- The language detection project has been successfully executed, resulting in the development of an accurate and efficient language detection system.

- Throughout the implementation, the project has achieved its primary objective of creating a robust model capable of automatically identifying the language of given text samples.

- The following key outcomes and contributions have been made:

1. **Model Performance and Accuracy:**

   The Multinomial Naive Bayes classifier, trained on the preprocessed text data using the bag-of-words representation, has exhibited impressive performance. The model achieved a high accuracy score of 97% on the testing set, showcasing its ability to make correct language predictions for a wide range of text samples.

2. **Practical Applicability:**

   The language detection system holds immense practical applicability in various real-world scenarios. Its accurate and efficient nature makes it an invaluable tool for tasks requiring automatic language identification. The system can be integrated into multilingual content filtering, language-specific analysis, and other language-related applications.

3. **User-Friendly Language Prediction Function:**

   The development of the "predict" function allows users to predict the language of any given text input with ease. The function utilizes the trained model and CountVectorizer to provide quick and accurate language predictions, enhancing the system's usability and accessibility.

4. **Contributions to Natural Language Processing (NLP):**

   The successful implementation of the language detection system contributes to the field of NLP. It demonstrates the effectiveness of the Multinomial Naive Bayes classifier for language identification tasks, emphasizing the significance of proper data preprocessing in text classification applications.

5. **Future Potential and Improvements:**

   While the project has achieved commendable results, there are opportunities for further improvements. Enhancing the dataset with more diverse languages and exploring advanced NLP techniques like word embeddings or pre-trained language models could potentially boost the model's accuracy and performance.

6. **Practical Utility and Real-World Impact:**

   The language detection system's successful implementation showcases the practical utility of machine learning in language-related tasks. Its reliable language identification capabilities open doors to a wide array of multilingual applications, supporting businesses and researchers in various linguistic endeavors.

- In conclusion, the language detection project has effectively addressed the challenges of language identification by leveraging the power of machine learning and NLP techniques.

- The project's achievements in accuracy, efficiency, and real-world applicability highlight its potential as a valuable tool in today's multilingual landscape. By continually refining and expanding the system's capabilities, it can remain at the forefront of language detection technology and further contribute to advancements in the field of Natural Language Processing.

- Overall, the successful completion of this project reflects the capabilities of modern machine learning approaches and their impact on language-related applications.

Future Enhancements:

While the developed language detection system performs well, there are opportunities for further improvements. Future work could explore the following areas:

- **Dataset Enhancement:**

  Consider using a larger and more diverse dataset to improve the model's ability to handle various language samples.

- **Advanced NLP Techniques:**

  Explore more advanced NLP techniques, such as word embeddings or language models, to capture semantic and contextual information for language identification.

- **Handling Short Texts:**

  Investigate methods to handle short text samples effectively, as language detection may pose challenges for texts with limited context.

- **Multilingual Context:**

  Extend the model to handle multilingual text samples where multiple languages may appear in a single document.

References:

1. The Python library scikit-learn. Available at: scikit-learn.org
2. The Python Programming Language. Available at: Python Language
3. Dataset get from: kaggle.com

This concludes the detailed project report on the language detection system implemented using the Navie's Byes Classifier Machine Learning Module.