LANGUAGE DETECTION AND TRANSLATION

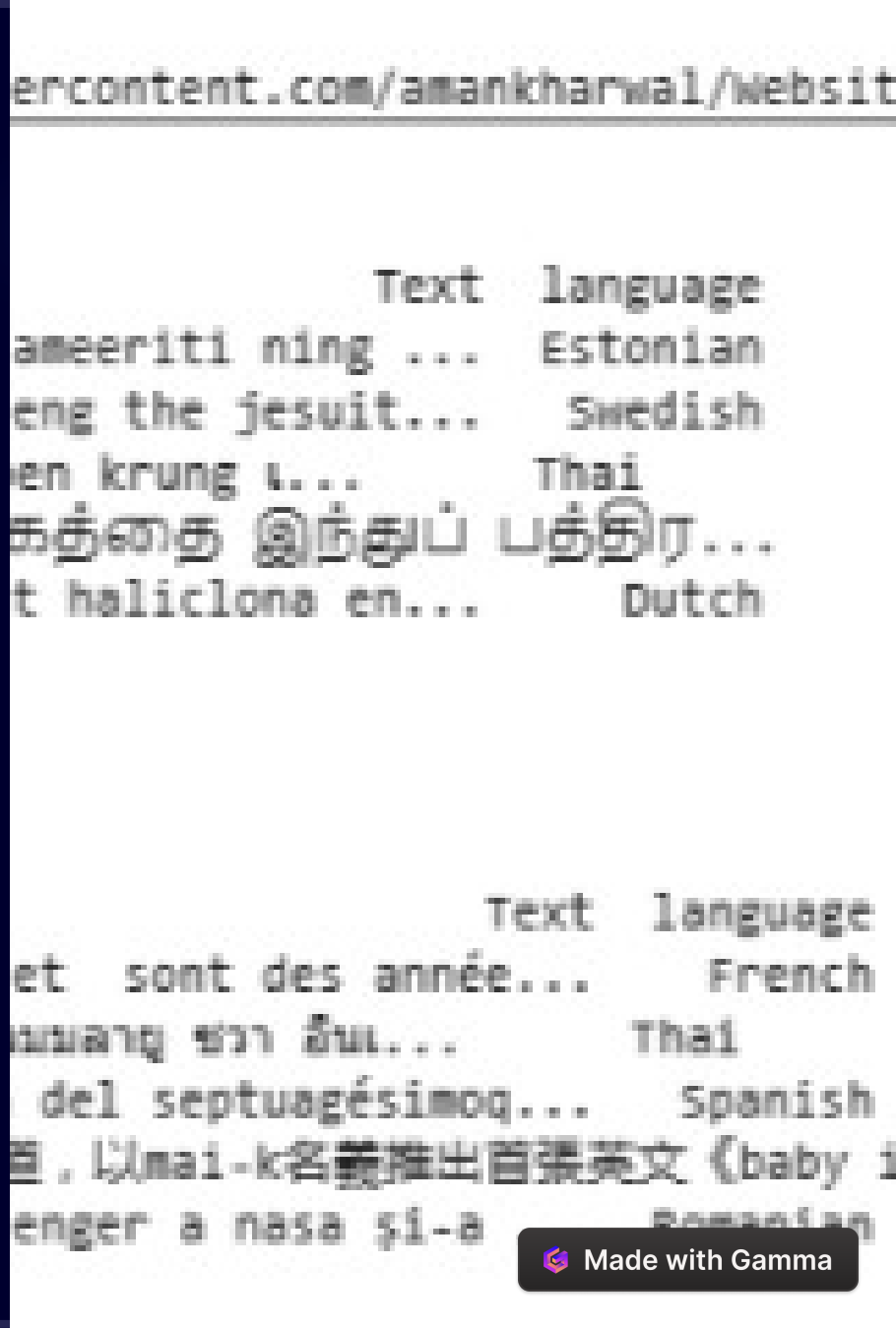
In this presentation, we will explore the fascinating world of language detection and translation. Join us as we delve into the complexities and possibilities of cross-cultural communication.

Introduction

- **Project Title: "Language Detection and Translation"**
- Our goal is to develop an advanced language detection and translation system for seamless multilingual conversations.
- **Team Members: Waqar Ali(Sec. 2), Muhammad Ali(Sec. 2), Nouman(Sec. 3).**

About Data Set

- **Dataset:** we collected the dataset from Kaggle.
- The data consists of 22,000 sentences in 22 Languages.
- Each Class consists of 1000 sentences and their relative labels(Language Name)
- **Link For Data Set:** https://raw.githubusercontent.com/amankharwal/Website-data/master/dataset.csv"



Visualization of Data Set

```
data = pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/dataset.csv")
print(data.head())
```

```
                                                Text  language
0  klement gottwaldi surnukeha palsameeriti ning ...  Estonian
1  sebes joseph pereira thomas  på eng the jesuit...   Swedish
2  ถนนเจริญกรุง อักษรโรมัน thanon charoen krung เ...      Thai
3  விசாகப்பட்டினம் தமிழ்ச்சங்கத்தை இந்துப் பத்திர...     Tamil
4  de spons behoort tot het geslacht haliclona en...     Dutch
```

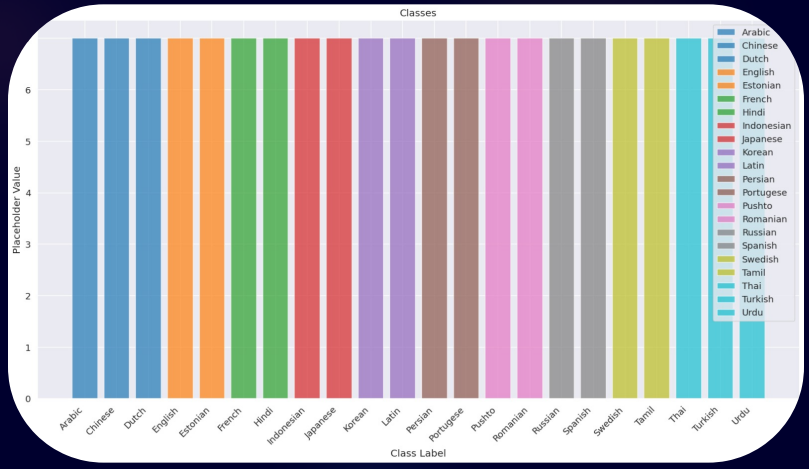
```
print(data.tail())
```

```
                                                Text  language
21995  hors du terrain les années  et  sont des année...    French
21996  ใน พศ  หลักจากที่เสด็จประพาสแหลมมลายู ชวา อินเ...      Thai
21997  con motivo de la celebración del septuagésimoq...   Spanish
21998  年月，當時還只有歲的她在美國出道，以mai-k名義推出首張英文《baby i like》，由...   Chinese
21999   aprilie sonda spațială messenger a nasa și-a ...  Romanian
```

Visualization of Data Set

```
a["language"].value_counts()

Estonian     1000
Swedish      1000
English      1000
Russian      1000
Romanian     1000
Persian      1000
Pushto       1000
Spanish      1000
Hindi        1000
Korean       1000
Chinese      1000
French       1000
Portugese    1000
Indonesian   1000
Urdu         1000
Latin        1000
Turkish      1000
Japanese     1000
Dutch        1000
Tamil        1000
Thai         1000
Arabic       1000
Name: language, dtype: int64
```



Model Architectures: Model Selection Recap

- We carefully selected models based on their performance and suitability for our project goals. We selected Multinomial Naïve Bayes Classifier from Sklearn.
- We selected countVectorizer for tokenization and for vectors.

Model Architectures: Detailed Model Architecture

- **A vectorizer**, in the context of natural language processing (NLP) and machine learning, is a tool or method used to convert text data into a numerical format that can be utilized by machine learning algorithms.
- The CountVectorizer works through the following steps:
- **Tokenization:** It breaks down each document into individual words or tokens.
- **Vocabulary Building:** It creates a vocabulary, which is a set of all unique words (or tokens) found in the entire collection of documents.
- **Counting:** It counts the occurrences of each word in each document and constructs a matrix where each row corresponds to a document, each column corresponds to a word in the vocabulary, and the values represent the counts of each word in each document.

Model Architectures:

Detailed Model Architecture

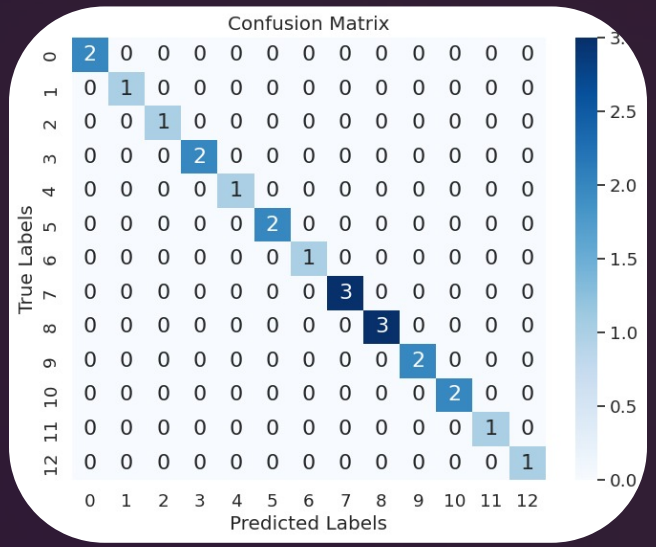
- **Multinomial Naive Bayes (NB) classifier**, as implemented in scikit-learn, does not involve an iterative optimization process with epochs, as is typical in some other machine learning models like neural networks. Instead, it follows a closed-form solution based on probability theory. Here's an overview of how the Multinomial Naive Bayes model is trained:
- **Training Process:**
  - **Data Preparation:**
  - **Input Data:** The training data consists of a set of documents (text) and their corresponding class labels.
  - **Feature Extraction:** The documents are typically represented as feature vectors, where each feature corresponds to a term (word) and the value represents the frequency of that term in the document.
- **Parameter Estimation:**
  - **Prior Probabilities:** The prior probability of each class is calculated based on the distribution of class labels in the training data. This is the probability of encountering each class without considering any features.
  - **Likelihood Estimation:** For each term in the vocabulary, the likelihood of observing that term given a class is estimated. This is done by calculating the conditional probability of the term occurring in documents of each class.
- **Model Training:** The Multinomial Naive Bayes model is trained by storing these estimated probabilities.
- **Making Predictions**: Once the model is trained, it can be used to make predictions on new, unseen data.
  - **Data Preparation:** The new documents are transformed into the same feature space as the training data.
- **Prediction:**
  - For each document, the model calculates the probability of it belonging to each class using Bayes' theorem. The class with the highest probability is assigned as the predicted class for that document.

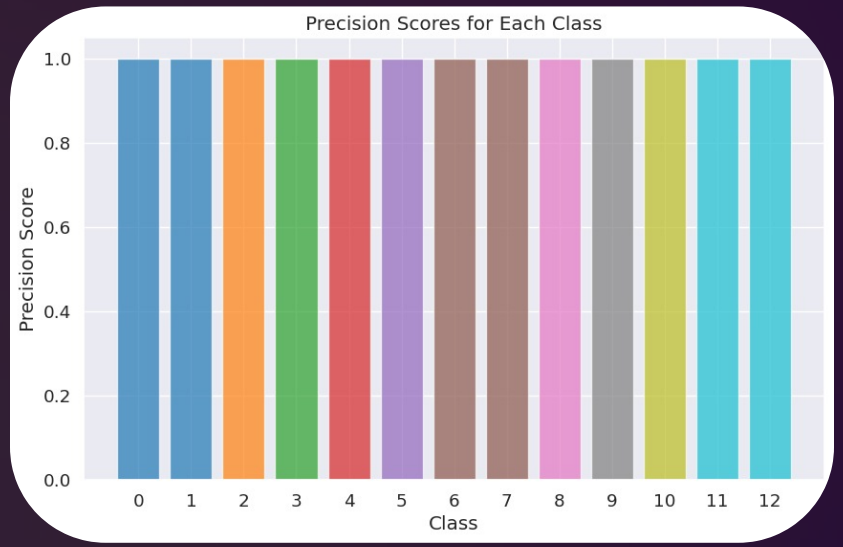
Model Architectures: Model Training Process

- We converted the text into **numpy** array and fed to model for training.
- we stored the text in a variable x_train and labels for text in y_train and fed to model.
- **Train Test Split:**
  - we splat the data as 90 % for training and 10% for Testing Model

Evaluation Metrics and Results: Chosen Evaluation Metrics

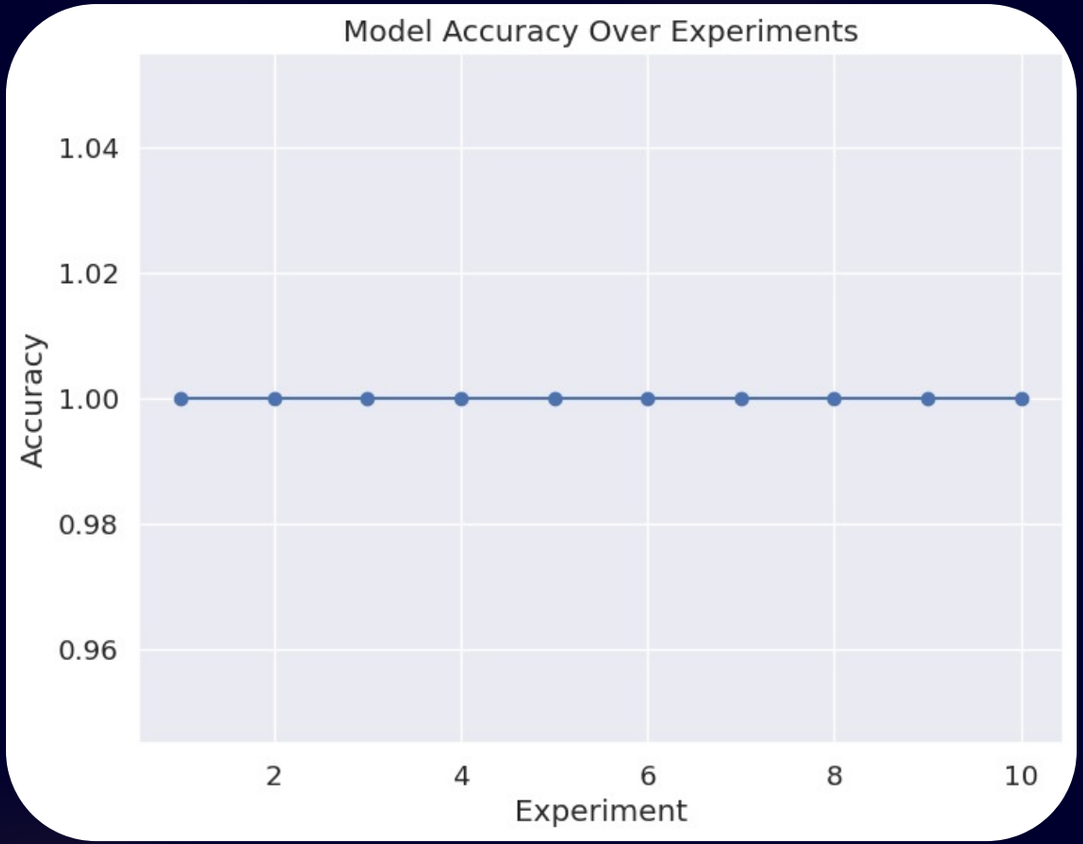
- We carefully selected evaluation metrics, including confusion matrix and precision-recall curve, to accurately measure our model's performance.
- We will justify why these metrics are appropriate for our project, ensuring reliable assessment of our system.





Evaluation Metrics and Results: Results and Discussion

- **Model Accuracy Over Experiments**



Translation

For translation we used free source translation library by Google.

known as gooletrans

Demo Of App

Here is a Demo Video of our app:

Our app Can Detect 22 Languages present in Data Set

Link for Demo Vedio: https://drive.google.com/file/d/1q5xFklnhclWqkovg3FuvODlB1TDT_8Pk/view?usp=sharing

Conclusion

- Through our project, we have achieved significant milestones in the development of advanced language detection and translation systems.
- We reflect on our learnings and their broader applications in the field of AI, highlighting the importance of breaking language barriers for a more connected world.

Thanks For Your Attention...