Language Detection Project Documentation

Overview

This project focuses on building a language detection model using machine learning techniques. The goal is to classify text into one of 22 different languages. The project utilizes the **Multinomial Naive Bayes** algorithm with two different text vectorization methods: **Bag of Words (BoW)** and **Term Frequency-Inverse Document Frequency (TF-IDF)**. The model is trained on a dataset containing 22,000 text samples, each labeled with its corresponding language.

Dataset

The dataset used in this project is stored in a CSV file named language.csv. It contains two columns:

- **Text**: The text samples in various languages.
- **Language**: The language label for each text sample.

The dataset includes 22 languages, each with 1,000 samples, making a total of 22,000 samples.

Dataset Statistics

- **Total Samples**: 22,000
- **Languages**: 22 (Estonian, Swedish, English, Russian, Romanian, Persian, Pushto, Spanish, Hindi, Korean, Chinese, French, Portugese, Indonesian, Urdu, Latin, Turkish, Japanese, Dutch, Tamil, Thai, Arabic)
- **Missing Values**: None

Project Workflow

1. Importing Libraries

The project uses the following libraries:

- **Pandas**: For data manipulation and analysis.
- **NumPy**: For numerical operations.
- **Scikit-learn**: For machine learning tasks, including model training, evaluation, and text vectorization.

2. Loading the Dataset

The dataset is loaded into a Pandas DataFrame, and the first few rows are inspected to understand its structure.

3. Data Preprocessing

- **Checking for Missing Values**: The dataset is checked for any missing values, and it is confirmed that there are none.
- **Dataset Shape**: The dataset contains 22,000 rows and 2 columns.

- **Language Distribution**: Each language has exactly 1,000 samples, ensuring a balanced dataset.

4. Splitting the Dataset

The dataset is split into training and testing sets using an 80-20 split:

- **Training Data**: 70% of the dataset (15,400 samples).
- **Testing Data**: 30% of the dataset (6,600 samples).

5. Text Vectorization

Two text vectorization techniques are used to convert text data into numerical format:

1. **Bag of Words (BoW)**: Converts text into a matrix of token counts.
2. **Term Frequency-Inverse Document Frequency (TF-IDF)**: Converts text into a matrix of TF-IDF features.

6. Model Building

The **Multinomial Naive Bayes** algorithm is used for classification. Two models are trained:

- **Model 1**: Using BoW vectorization.
- **Model 2**: Using TF-IDF vectorization.

7. Model Training

Both models are trained on the vectorized training data.

8. Model Evaluation

The models are evaluated on the testing data using **accuracy score** as the metric:

- **BoW Model Accuracy**: ~95.48%
- **TF-IDF Model Accuracy**: ~95.56%

9. Making Predictions

The trained models are used to predict the language of new text inputs. For example:

- Input: "L'apprentissage automatique change la façon dont nous comprenons les données."
- Predicted Language: French

10. Saving the Model

The trained BoW model is saved using the pickle library for future use.

Results

- Both the BoW and TF-IDF models achieved high accuracy (~95.5%), with the TF-IDF model performing slightly better.

- The models are capable of accurately predicting the language of text inputs across all 22 languages.

Conclusion

This project successfully demonstrates the use of machine learning for language detection. The Multinomial Naive Bayes algorithm, combined with text vectorization techniques like BoW and TF-IDF, proves to be effective for this task. The model can be further improved by experimenting with other algorithms, hyperparameter tuning, or using larger datasets.

Tools and Libraries Used

- **Pandas**: Data manipulation and analysis.
- **NumPy**: Numerical operations.
- **Scikit-learn**: Machine learning tasks (text vectorization, model training, evaluation).
- **Pickle**: Saving and loading the trained model.