Predicting nationality base on names report

Problem statement:

The value of this project shines when you want to predict their nationality base on only the names. It can detect does their name match with given country. This also be useful when you want to fill the missing value in country column given a name of that person.

In my personal case, the idea of creating this project is I want fill the missing value base on the name of casts in Netflix movies and tv-show dataset.

[Netflix Movies and TV Shows | Kaggle](https://www.kaggle.com/datasets/shivamb/netflix-shows)

This will provide valuable insight if we want to analyze the nationality of cast in a movie and conclude which country have the most actor/actress in the dataset.

The method:

Here is the workflow:

Cleaned JP dataset

Save model Japan.pt

Filter the raw\_japan dataset

Fake dataset for Japan

JP, FR, VN dataset

LSTM model on 3 language

Fake dataset for Vietnam

Save model Vietnam.pt

Filter the raw\_ Vietnam dataset

Cleaned VN dataset

Cleaned FR dataset

Filter the raw\_ France dataset

Save model France.pt

Fake dataset for France

For cleaning the dataset, I first applied some text preprocessing technique like Unicode to ascii transformation, remove nan value, translate the language like japanese, korean, Russian created by Faker library.

For each language fake dataset, I will extract the value that has largest number of character and padding all other value with “-“ so that it will have the same length. It is necessary for training in batch so the input size will have the same dimension in a batch.

The dataset I used for training on fake\_dataset is the purpose language against English. When I train fake dataset for English, it will use several language against English like Japan, Inida, Mexico,…

After obtained trained model for each languages. I use these model to filter each raw\_language csv file and save the filtered as clean\_language.csv

Finally, I create a large LSTM model to classify all language on cleaned dataset. The result I got for 11 languages is 85.6%, each clean\_datasets has about 10k rows.

Future improvement:

I will consider to train the large LSTM model with larger cleaned dataset.

I will use tokenizer in order to split the given word into tokens with appropriate vocab\_size with this dataset. In this project, I use the vocab = (a,b,c,d,e,…) that split raw text by letters.