MULTILINGUAL TEXT CLASSIFIER

Multilingual Text Classifier has two parts -

- **Classification** : Text classification involves assigning a label or category to a text document based on its content.
- **Multilingual** : Multilingual text classification means the model can handle and understand text in multiple languages. This is achieved through techniques like using language models that are trained on a large amount of multilingual text data.

Aim : Creating a multilingual text classifier can help organizations automate the process of language identification and text categorization for various applications.

This folder consists of 2 other folders:

1. Implementation
2. Model

Implementation - consists of code, results of implementation of the code (csv, png files and analysis graph) and the PPT file which contains the overview of the project.

Model - consists of the actual model which is used for execution and obtaining the results.

After opening this folder navigate to:

Model -> models -> sentiment_model -> checkpoint-1875

Copy the path after opening “ checkpoint-1875 “

Under “Model Loaders” section of the code, paste the copied path in the variables “tokenizer” and “model”

```
tokenizer = AutoTokenizer.from_pretrained("your path", local_files_only=True)
```

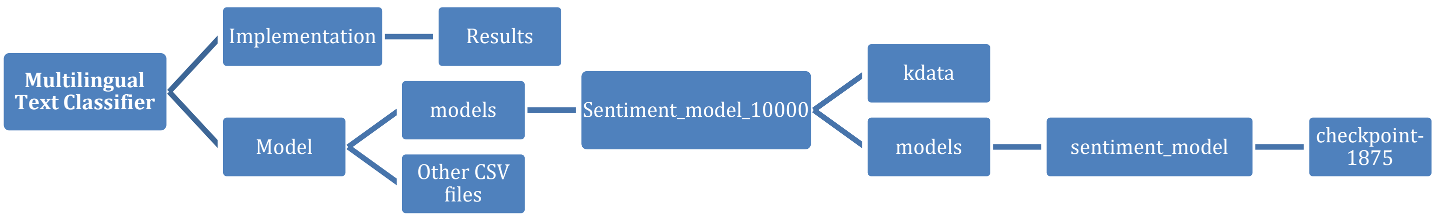
- This line loads a pre-trained tokenizer from a local directory.
- AutoTokenizer.from_pretrained() is a Hugging Face method that automatically selects the appropriate tokenizer class based on the saved configuration.
- The path points to a specific checkpoint directory (checkpoint-1875) where the tokenizer files are stored.
- local_files_only=True ensures it only looks for the tokenizer locally (won't try to download from the internet).

```
model = AutoModelForSequenceClassification.from_pretrained("your path")
```

- This line loads a pre-trained sequence classification model from the same local directory.
- AutoModelForSequenceClassification.from_pretrained() automatically selects the correct model architecture for sequence classification tasks.
- The model was likely fine-tuned for sentiment analysis (since it's a sequence classification model stored in a "Sentiment_Model" directory).

Note: These 2 lines of code are the only major changes required since the model is already trained.
Other necessary requirements - proper installation of python (correct version), using the correct environment (if you use one), selecting the correct interpreter in the VScode (if you are using VScode), installing the correct extensions before using VScode (such as Python , Python debugger, etc) and opening or loading the folder where your project is located, into VScode before executing the code.

Folder structure :



- **'Implementation' folder has been uploaded on GitHub**
- **'Model' folder can be found at:** https://drive.google.com/drive/folders/1eJlljZnqsMbBd5rgvWKo_dmygR4JFh8b?usp=