Code Documentation

This model is used to predict INR against USD given a set of currencies and the date. Two endpoints, ‘currencyprediction’ and ‘currencytraining’ handle prediction and training the model respectively. Input data can be in form of CSV file or JSON.

The **artifacts** folder contains the **model.pkl** **file, raw data, train data and test data**. During the data ingestion phase, raw, train and test data will be created, which can then be used to train the model. While training the model for the first time, model.pkl file will be created which will then be used for prediction. For further prediction calls, the same model.pkl file will be used and in case the need arises, training can be done again which will replace the existing pkl file with the new file.

The **logs** folder contains detailed logs documenting the entire code execution and can be used to figure out in which stage the error occurs. Logs are systematically named using the date and time of code execution.

The **src** folder contains the entire code responsible for execution of the model. MLProject folder contains two subfolders, **components and pipeline** and other files like **exception, logger and utils**.

The **exception** file contains code which handles custom exceptions for better error reporting. The custom exception returns a formatted string containing the error message, filename and line number.

The **logger** file is responsible for creating logs.

The **utils** file contains a list of commonly used functions which will be implemented in other parts of the code. Currently, it contains **save\_object, load\_object and evaluate\_models** functions.

The components folder contains three files, **data\_ingestion, data\_transformation and model\_trainer**.

**Data ingestion** reads data which is passed as a dataframe to the the intiate\_data\_ingestion function and creates the respective files in the artifacts folder.

**Data transformation** is responsible for converting non-numerical data into numerical data for both the training set and test set.

**Model trainer** is used to train the data using a variety of models along with hyperparameter tuning. The best model which gives the lowest RMSE is saved as model.pkl file.

The **pipeline** folder contains the **prediction\_pipeline** file which makes predictions on a given set of features using the loaded model.

The **templates** folder contains a basic **index.html** file which will be rendered when hitting the base route of our API.

**app.py** is the file which contains both the endpoints implemented using Flask.

**middleware.py** contains middleware which allows only authorised users to access the endpoints.

**requirements.txt** contains the list of dependencies.

**setup.py** is responsible for install and package the dependencies mentioned in requirements.txt

**template.py** contains the list of files.