The complete prediction is divided into 3 python scripts:

1: **Data_Preprocessing.py**: This script outputs the needed dataset to be able to feed into the model by using the dataset given after generating it from the servers (here it's like the time_between_two_tolls.csv) and saves the required output as '*Processed_df.csv*'

2: **Model_py**: This script uses this 'Processed_df.csv' as input to train the model with the optimum hyperparameters saves the trained model as '*finalized_model.sav*' using the pickle package python. The other files that are pickled are the list of vehicle number, toll_both number assignment dictionaries, as we need to convert the test dataset toll_both, vehicle_no fields in the exact same numbering system as we did with the train_df. This scripts takes some time as it trains the model.

Here the hyper-parameter optimization is also included by the user's choice. If chosen, we can perform the hyper parameter tuning by supplying the dictionary of parameters range in the terminal when asked for it. (for example {'hidden_layer_sizes': [(100,40),(200,50)], 'alpha':[3,5,7]}). (Parameters and their values) You can also change the responsible evaluation metric to choose (Here I used *neg_mean_squared_error*) to find the best parameters (Metrics and their explanation). *But this takes a lot of time according to the size of the input given.*

3: **Prediction_py**: This script is used to supply the test data and use it to predict the time of arrival for the remaining tolls using the saved model '*finalized_model.sav*'. The predicted results of the forthcoming toll_boths are saved in the csv file of '*predicted_eta.csv*'. This script takes practically negligible amount of time as it just predicts using the already trained model.