Ixigo Assignment: By Anubhav Kesari

Developing a train-delay prediction model for the Indian Railways trains.

- Step 1 - Basic Understanding of the process

Basic Understanding of the Process -

The amount of the delay that a train could have at any station during its journey can be expressed be as a result of many factors

- Train Factors
  - What's the max speed that train can run
  - What's the train cat? - Local,Express,Superfast ?
- Spatial Factors
  - How busy is the route that the train is going through
  - Is the time of travel the peak time.
- Temporal Factors
  - How much it has been already been delayed in the previous stations

So to build more confidence about the above factors and their importance in the delay estimation , I started doing EDA of the data.

- Step 2 Basic EDA of the data

Various EDAs to analyse the various variables in the input dataset and their impact on delay estimation was carried out . Clearly shown in the notebook.

- Step 3 - Defining the features for each station wrt to train

The **input_feature** at current station n consists of :

- TrainCode: Signifying the train
- ArrivalDelay_n : at station wrt actual
- DepartureDelay_n : at station wrt wrt actual
- Distance_n: distance from nth station to n+1th station
- Planned_running_time_n : planned running time for n+1th station

- Day_n - day of the week at station n
- hour_n - of the week at station n

Defining the Deep Learning Problem

- It's a times series data because
    - The current delay depends on various factors of previous stations which would be obviously happened in time before now

- I thought of doing this in a seq2one fashion by taking n-1th station and nth station feature vectors to get n+1th delay, but this can't be used as such in test time. And doing it in seq2one prediction style

- Each train (distinct running train) would be considered as a sample)

Scaling Data

- At first I thought of doing the MinMaxScaler , but either MinMaxScaler or StandardScaler,both are very sensitive to outliers , so I tried giving my choice to RobustScaler (an implementation in sklearn,that uses only a certain percentile of data to get mean and std.

Evaluating performance - Strategies

- Mean absolute error (mae) delay to calculate the loss function of the LSTM Net
-

Forecasting Strategies

- For this , I did literature surveys around which strategy would be best .

- Predicting one step at a time looks good , but can lead to degradation of the model due to accumulation of the model's errors.

- I have tried to apply both the recursive and hybrid approach

Time Window Selection

- I first tried with window of just 2 i.e. features at station n-1 and n to get delay at n+1
- Then I extended this to 5 window problem i.e. n to n-4

Test Set Prediction Strategy

I had two options in my mind to predict the delays of a train

- 1 . Use the departure delay of last station of previous running instance of the same train to forecast next instance's arrival at starting station

- 2. Predict the arrival delay of the previous running instance at the starting station (the distance being the whole length of journey) to calculate the arrival delay.

- Use down trains arrivals at various stations as features in the down train predictions

To understand the inferences

- I tried to compare it with baseline delays of the training time.

Improvements:

- Use of ensemble modelling
- Trying of 2nd & 3rd strategy in predicting the test set
- More analysis in post training part
    - Analysis how my model works train wise
- Due to the paucity of time, I gave the baseline predictions as outputs ,as there was some error ,but If given time,I could try my suggested one.