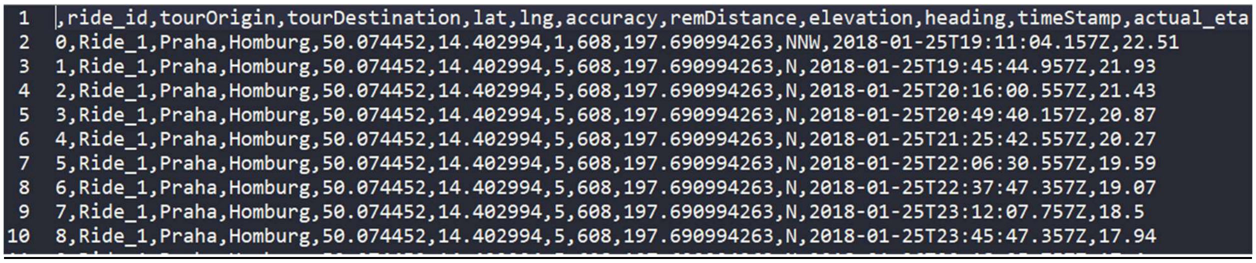
Dataset Description

The documentation gives a brief overview of the problem and the dataset.

SNo.	Column Name	Description
1.	ride_id	
2.	tourOrigin	The origin city of the ride
3.	tourDestination	The destination city of the ride.
4.	lat	The latitude of the current position.
5.	lng	The longitude of the position.
6.	accuracy	Accuracy of the measurement.
7.	remDistance	The remaining distance (in km) in the trip.
8.	elevation	The elevation in the route which the vehicle has encountered.
9.	heading	The direction in which the vehicle is heading.
10.	timestamp	The date and time of the measurement
11	**actual_eta**	Actual remaining time to destination in fraction of hours. This is the field that you need to predict for a large fraction of records in df_test.csv (see below)

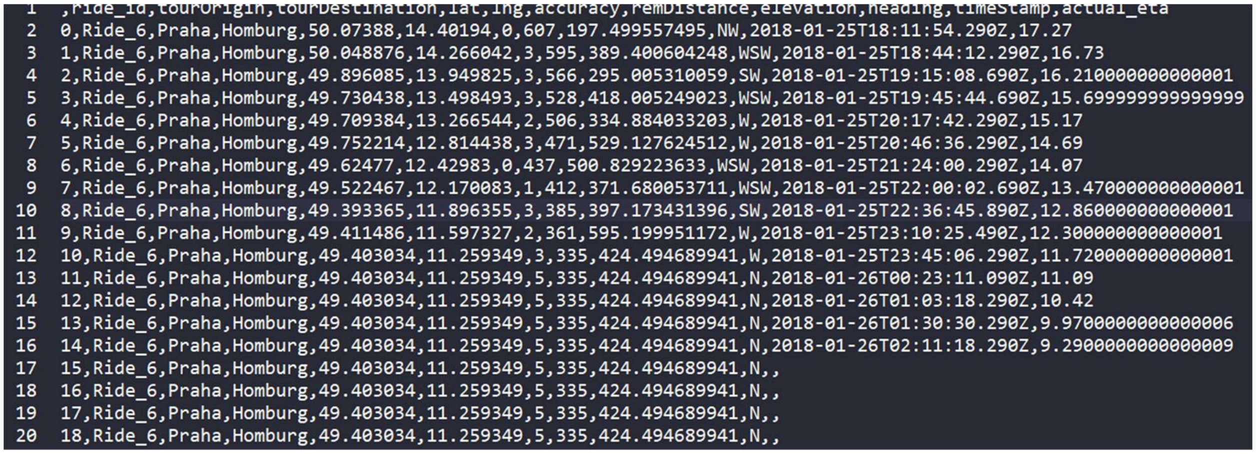
Dataset Splits

df_train.csv



```
,ride_id,tourOrigin,tourDestination,lat,lng,accuracy,remDistance,elevation,heading,timeStamp,actual_eta
0,Ride_1,Praha,Homburg,50.074452,14.402994,1,608,197.690994263,NNW,2018-01-25T19:11:04.157Z,22.51
1,Ride_1,Praha,Homburg,50.074452,14.402994,5,608,197.690994263,N,2018-01-25T19:45:44.957Z,21.93
2,Ride_1,Praha,Homburg,50.074452,14.402994,5,608,197.690994263,N,2018-01-25T20:16:00.557Z,21.43
3,Ride_1,Praha,Homburg,50.074452,14.402994,5,608,197.690994263,N,2018-01-25T20:49:40.157Z,20.87
4,Ride_1,Praha,Homburg,50.074452,14.402994,5,608,197.690994263,N,2018-01-25T21:25:42.557Z,20.27
5,Ride_1,Praha,Homburg,50.074452,14.402994,5,608,197.690994263,N,2018-01-25T22:06:30.557Z,19.59
6,Ride_1,Praha,Homburg,50.074452,14.402994,5,608,197.690994263,N,2018-01-25T22:37:47.357Z,19.07
7,Ride_1,Praha,Homburg,50.074452,14.402994,5,608,197.690994263,N,2018-01-25T23:12:07.757Z,18.5
8,Ride_1,Praha,Homburg,50.074452,14.402994,5,608,197.690994263,N,2018-01-25T23:45:47.357Z,17.94
```

df_test.csv



```
,ride_id,tourOrigin,tourDestination,lat,lng,accuracy,remDistance,elevation,heading,timeStamp,actual_eta
0,Ride_6,Praha,Homburg,50.07388,14.40194,0,607,197.499557495,NW,2018-01-25T18:11:54.290Z,17.27
1,Ride_6,Praha,Homburg,50.048876,14.266042,3,595,389.400604248,WSW,2018-01-25T18:44:12.290Z,16.73
2,Ride_6,Praha,Homburg,49.896085,13.949825,3,566,295.005310059,SW,2018-01-25T19:15:08.690Z,16.210000000000001
3,Ride_6,Praha,Homburg,49.730438,13.498493,3,528,418.005249023,WSW,2018-01-25T19:45:44.690Z,15.699999999999999
4,Ride_6,Praha,Homburg,49.709384,13.266544,2,506,334.884033203,W,2018-01-25T20:17:42.290Z,15.17
5,Ride_6,Praha,Homburg,49.752214,12.814438,3,471,529.127624512,W,2018-01-25T20:46:36.290Z,14.69
6,Ride_6,Praha,Homburg,49.62477,12.42983,0,437,500.829223633,WSW,2018-01-25T21:24:00.290Z,14.07
7,Ride_6,Praha,Homburg,49.522467,12.170083,1,412,371.680053711,WSW,2018-01-25T22:00:02.690Z,13.470000000000001
8,Ride_6,Praha,Homburg,49.393365,11.896355,3,385,397.173431396,SW,2018-01-25T22:36:45.890Z,12.860000000000001
9,Ride_6,Praha,Homburg,49.411486,11.597327,2,361,595.199951172,W,2018-01-25T23:10:25.490Z,12.300000000000001
10,Ride_6,Praha,Homburg,49.403034,11.259349,3,335,424.494689941,W,2018-01-25T23:45:06.290Z,11.720000000000001
11,Ride_6,Praha,Homburg,49.403034,11.259349,5,335,424.494689941,N,2018-01-26T00:23:11.090Z,11.09
12,Ride_6,Praha,Homburg,49.403034,11.259349,5,335,424.494689941,N,2018-01-26T01:03:18.290Z,10.42
13,Ride_6,Praha,Homburg,49.403034,11.259349,5,335,424.494689941,N,2018-01-26T01:30:30.290Z,9.9700000000000006
14,Ride_6,Praha,Homburg,49.403034,11.259349,5,335,424.494689941,N,2018-01-26T02:11:18.290Z,9.2900000000000009
15,Ride_6,Praha,Homburg,49.403034,11.259349,5,335,424.494689941,N,,
16,Ride_6,Praha,Homburg,49.403034,11.259349,5,335,424.494689941,N,,
17,Ride_6,Praha,Homburg,49.403034,11.259349,5,335,424.494689941,N,,
18,Ride_6,Praha,Homburg,49.403034,11.259349,5,335,424.494689941,N,,
```

The train data set is complete and contains all information.

In the test data set the rides timestamp and actual_eta are omitted for the second half of the ride. This is intentional: the first half of the ride could for instance be used to find out the general traffic conditions of that day.

Your task is to predict the actual_eta to destination for points for which timestamp and actual_eta are not provided. Note: you must **only** provide the ETA for the missing timestamps, not for the provided ones. This means that the y_pred.csv file should contain 1023 entries.

y_pred.csv

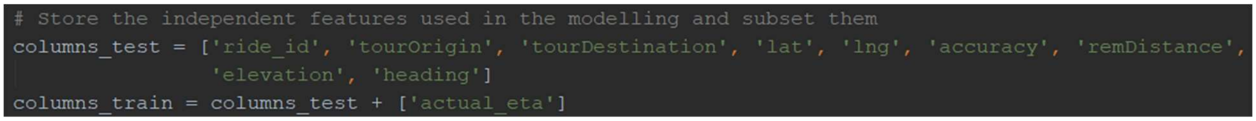
Predicted *subset* of df_test.csv. Must contain a list of values in which each item corresponds to actual_eta value in hours (for 30 minutes *eta* value would be 0.5 hrs) for each row in the test dataset which is omitted.

Baseline Model: Script

The modelling script is to enable the participants who are not well versed in data science but want to get their hands dirty with feature creation and engineering. It is in written in Python and can instance be used from the Cloudera workbench (or from your own Notebook).

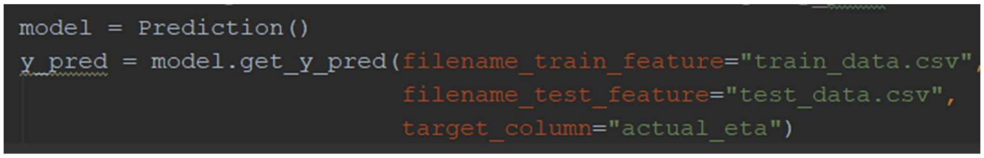
The script is documented, but here are some general tips:

- Create features for train and test dataset and save the results into train and test set as ".csv". It is better to save them in a file path and name other than the original dataset so that you always have the original data to create your features. Please be sure that the number of features are the same in train and test data with one extra column as actual_eta in test set.
- You need to add your added features in this list:



```
# Store the independent features used in the modelling and subset them
columns_test = ['ride_id', 'tourOrigin', 'tourDestination', 'lat', 'lng', 'accuracy', 'remDistance',
                'elevation', 'heading']
columns_train = columns_test + ['actual_eta']
```

- Pass the file-path as argument to the model to get y_pred and submit the y_pred values to the submission interface.



```
model = Prediction()
y_pred = model.get_y_pred(filename_train_feature="train_data.csv",
                          filename_test_feature="test_data.csv",
                          target_column="actual_eta")
```

- Note that any non-numeric feature will be removed before training of the baseline models

Roll your own ideas

Using the baseline script is of course just an option – feel free to try your own ideas!