Report On The Winning Solution Of The “Traffic Jam: Predicting People's Movement into Nairobi” Challenge

1) Creating the Data:

The first step to solve of this problem is to create the train set. Since the data provides information about ticket transactions, we need to group it by ride_id to to find the right number of tickets per ride.

2) Data processing & features engineering

To improve the performance of this model, additional meaningful features were needed. Here is a list of handcrafted features:

- date features(year,day of week ,minute).
- is_weekend: is the current day a weekend or not ⇒ a binary value (0,1).
- diff_btw_0_1_next_bus: time interval separating this ride’s bus and the next bus.
- diff_btw_0_1_previous_bus: time interval separating this ride’s bus and the previous bus.
- diff_btw_0_2_next_bus: time interval separating this ride’s bus and the following bus to the next bus.
- diff_btw_0_2_previous_bus: time interval separating this ride’s bus and the bus before the previous bus.
- diff_btw_0_3_next_bus: time interval separating this ride’s bus and 3rd following it.
- is_holidays : I looked up every holiday in nairobi, this feature answers the question about whether the current date is a holiday or not ⇒ a binary value (0,1).
- is_after_tomorrow_holidays: is the day after tomorrow a holiday or not ⇒ a binary value (0,1).
- count_trip_per_X_min_travel_from : how many rides per X time coming from each city.
- count_trip_per_X_min: how many rides per X time is coming to Nairobi.
- travel_from_distance: the estimated distance by google map from each city to Nairobi .
- travel_from_time: the google maps estimated duration of a trip from each city to Nairobi .
- Speed: travel_from_distance/ travel_from_time.
- haversine_distance: using the longitude and latitude of each city to calculate the haversine distance between each city and Nairobi .
- is_rush_hour: is the arrival hour is a rush hour or not .
- 12H: is the arrival hour before 11:59 or not.
- T_mean,P0_mean: Weather data extracted from Weather_archive_in_Nairobi (a public weather data) https://rp5.ru/Weather_archive_in_Nairobi_(airport).

- Uber movement data: will be addressed in following sections.
- number_of_ticket_by_hour_mean: aggregate features by hour.
- number_of_ticket_by_travel_from_mean: aggregate features by travel_from .
- number_of_ticket_by_minute_mean: aggregate features by minute.
- number_of_ticket_by_hour_dayofweek_mean: aggregate features by day of week and hour.
- number_of_ticket_by_hour_travel_from_mean: aggregate features by travel_from and hour.
- number_of_ticket_by_dayofweek_travel_from_mean: aggregate features by travel_from and day of week.
- number_of_ticket_by_minute_travel_from_mean: aggregate features by minute travel from .
- number_of_ticket_by_travel_from_sum: the sum of ticket by travel from over the total number of ticket in the train set: equivalent to the ratio of population in each city .

3) Uber data

Uber provided us with powerful features like the time of ride between Kawangware, the first stop in the outskirts of Nairobi, to Afya Centre, the main bus terminal where most passengers disembark. Those features improved my score by ~ 0.12.

Uber data contains 4 types of data:

- Travel_Times_Daily: ride time statistics for each day (this data contains a lot of missing values as shown in the uber_data notebook. This data is unuseful)
- Hourly Aggregate data: ride time statistics per hour of day
- Monthly Aggregate: ride time statistics per Month
- Weekly Aggregate: ride time statistics per day of week

I Integrated those features in two ways :

1. I added to the departure time of the Monthly Aggregate and Weekly Aggregate data the estimated ride time by google map to get the new day of week and the new day of month. Then I merged the Monthly Aggregate and Weekly Aggregate with new arrival time (month, day of week)
2. I integrated the Hourly Aggregate data three times in the final data with different arrival hours
    a. I added the estimated ride time by google map to the departure time in the purpose of getting the arrival hour.
    b. I added 9 hours to the departure time
    c. I added 8 hours to the departure time

Note: As you mentioned in the challenge description, the routes from these 14 origins to the first stop in the outskirts of Nairobi takes approximately 8 to 9 hours from time of departure.

4) Learning

I used Light GBM + XGboost to train the data. I tuned their parameters using grid search. I also used K-fold with (lgbm , xgboost) over 10 fold to slightly improve my score. For the final result, the Xgboost gave better score than the LGBM.

5) Provided code

The solution files are divided into 3 folders along with the model file:

- data: folder holding the raw data, raw uber data and the weather data
- features: folder holding the processed data which contains:
    - final features: folder holding the processed data.
    - prepare_data.ipynb: notebook to create the train data
    - Uber_data.ipynb: notebook to process the raw uber data
    - weather.ipynb: notebook to process the raw weather data
    - features.ipynb: Notebook for feature engineering
- sub: folder holding the output of model
- Model: folder holding the notebook to train the processed data
- LIB: useful functions and Class

6) Excision order

1. Prepare_data.ipynb
2. Uber_data.ipynb
3. Weather.ipynb
4. Features.ipynb
5. Xgboost.ipynb (Model folder)