Rideindego Bike Sharing Data

Lei Ma

Get Data

Code: python app/get_indego_data.py

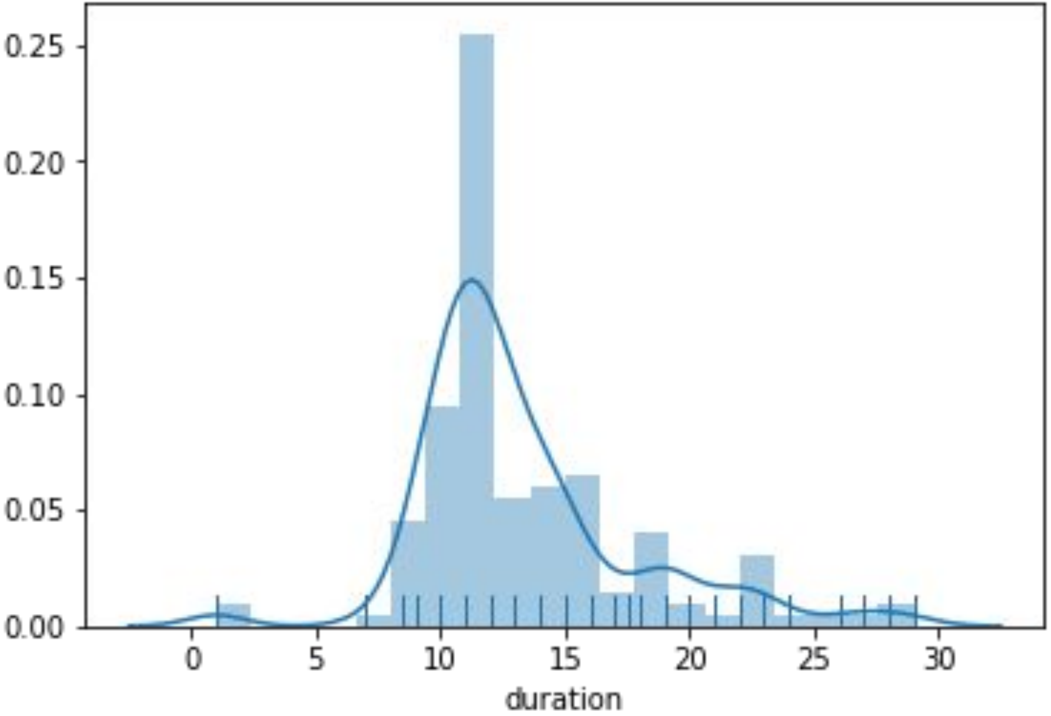
Parse HTML -> Download zip files + Unzip -> concat all files into one -> Transform data -> Save in specified path

Data transformations:

1. Backfill for bike_type: they introduced ‘electric’ type in 2018 q3. I have to back fill this using ‘standard’
2. bike_id has null values: fillna using 0
3. duration <= 2017 Q1 are in seconds: transform to minutes
4. Added columns: week of day, hour of day, month of year

EDA

Duration is around 12 min



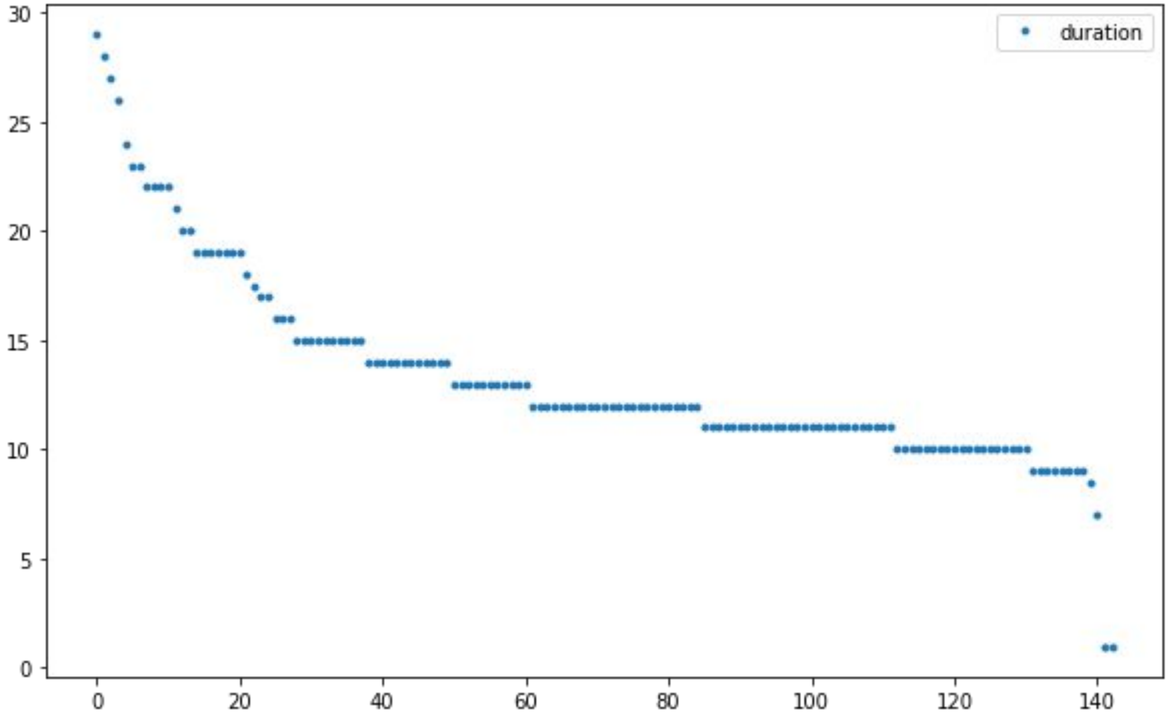
EDA

Bike stations matter (related to geolocation):

I have plotted the median duration for each station and sorted by the duration.

Horizontal axis shows the ordered stations.

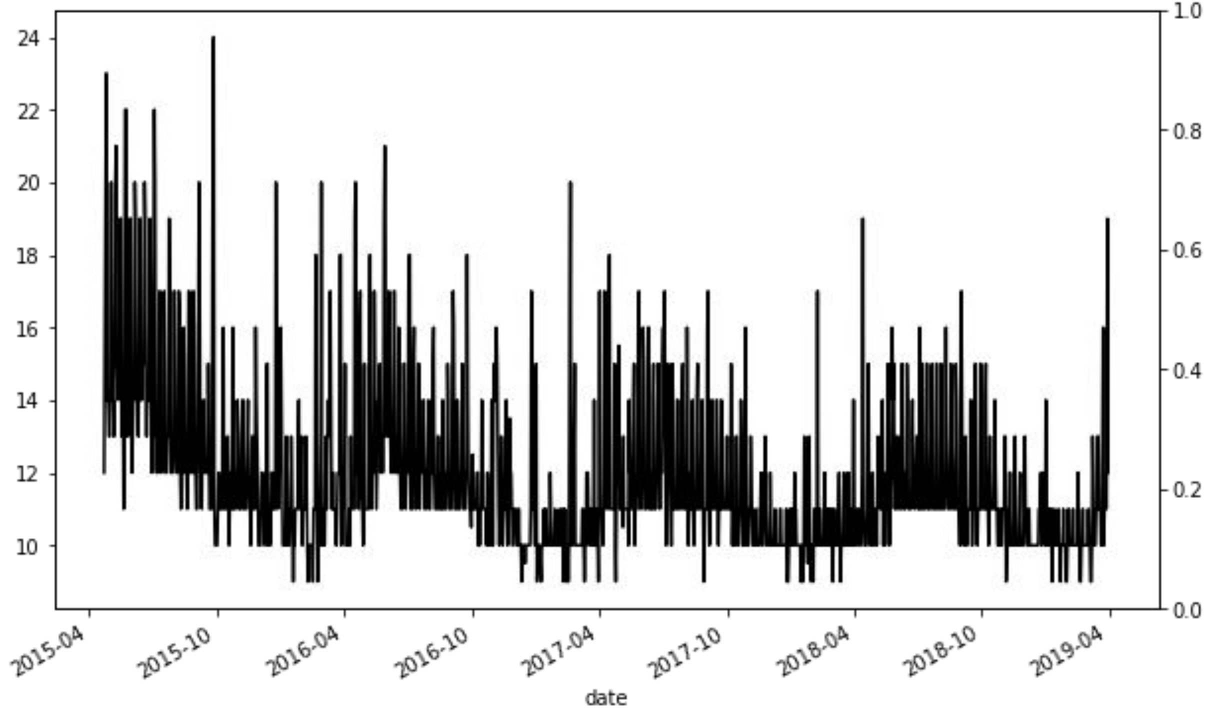
Vertical axis shows the median durations



EDA

Seasonality: duration is longer in summer

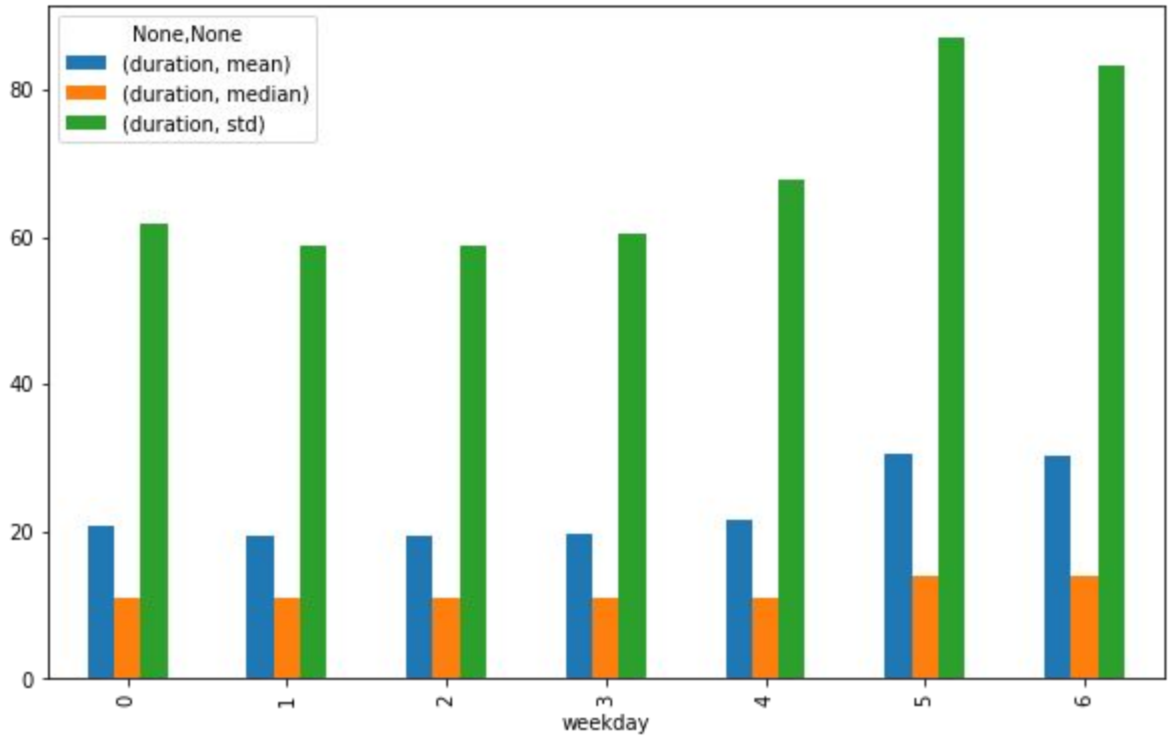
Vertical axis shows the median durations for each day



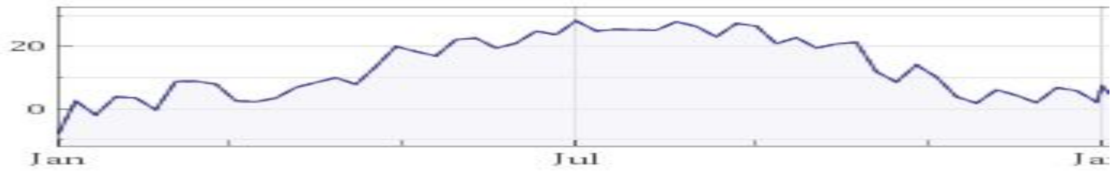
EDA

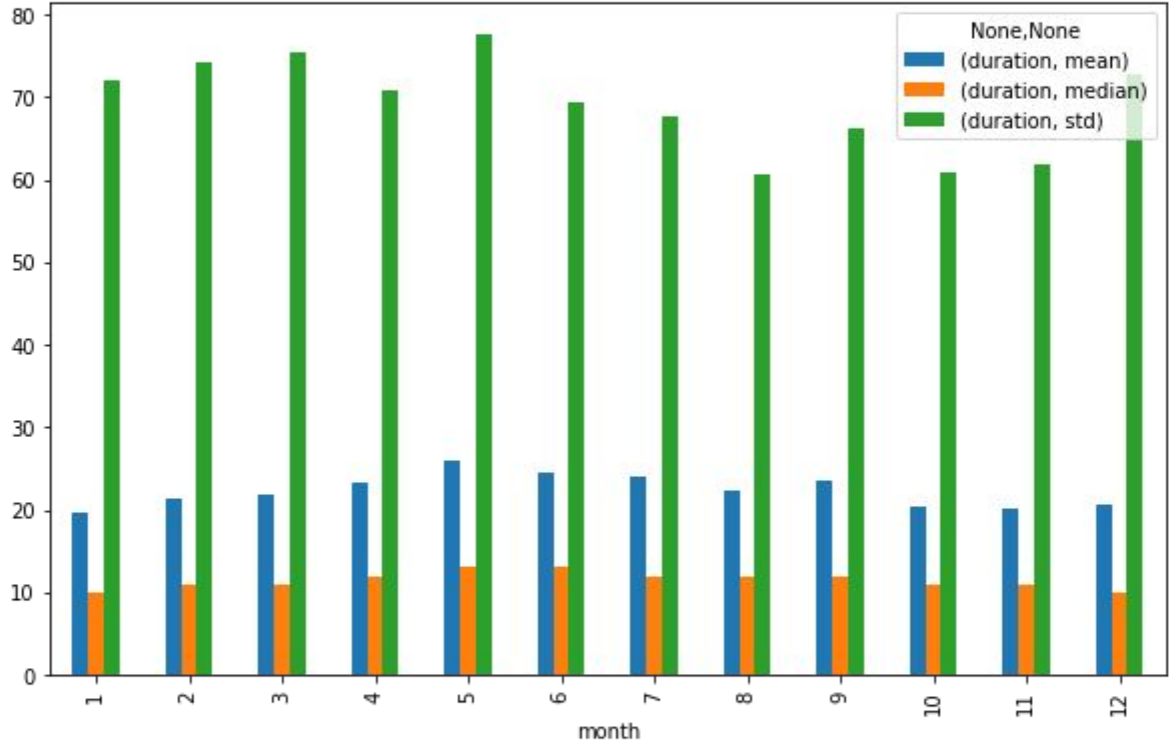
Duration is longer on weekends.

The bars are mean, median, std (standard deviation) of the durations.



EDA



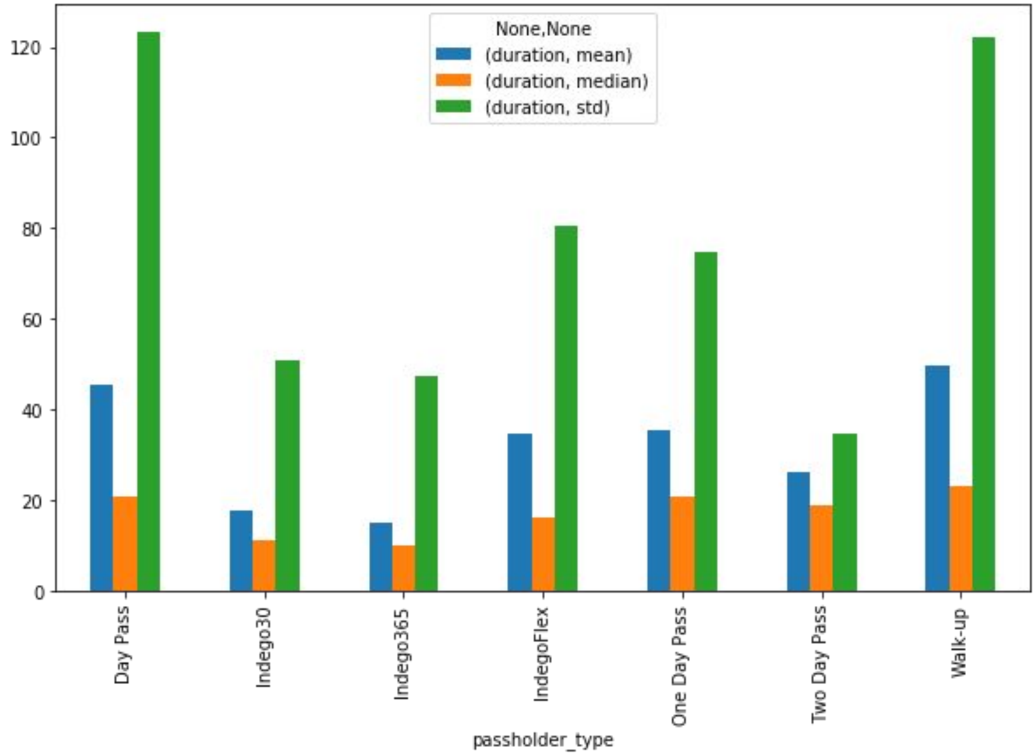


Cold and hot weathers leads to shorter durations.

The chart above shows the temperature in Philadelphia duration the year.

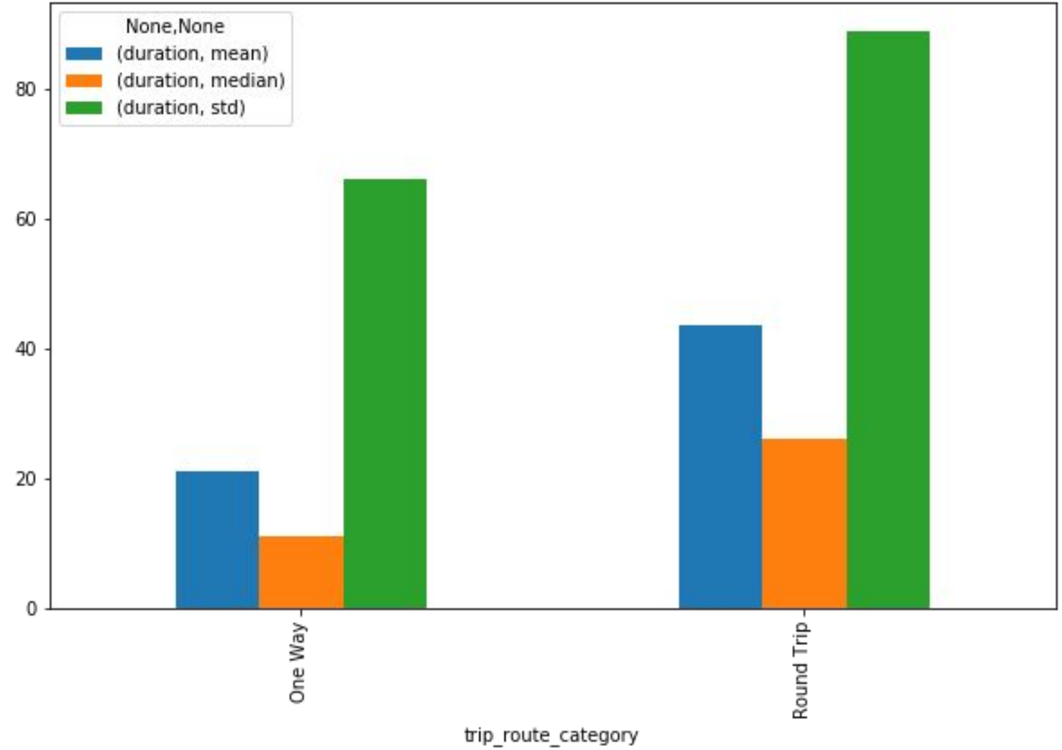
EDA

User plans matters



EDA

Trivial result: two way doubles the duration



Model

Ridge regression not working: almost like random predictions

R2 score:
0.036

1. OneHotEncoding
2. Using features: 'start_station_id', 'passholder_type', 'trip_route_category', 'hour', 'weekday', 'month'

Model

Decision tree not working

R2 score:
-0.4

Though not working, I have implemented it in the app: see the command on the right.

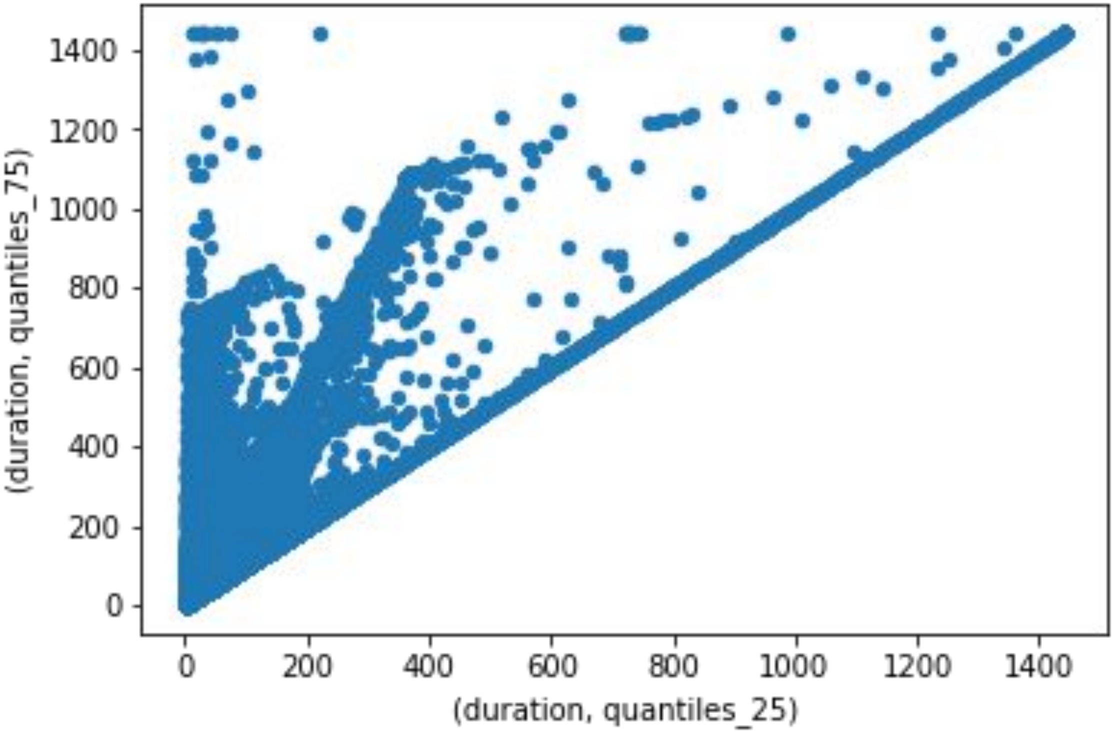
```
python app/prediction.py -f '{"passholder_type":"Indego30","trip_route_category":"One Way","hour":11,"weekday":0,"month":8, "bike_type": "standard"}'
```

Why aren’t the models working?

Bad feature selection:

The figure on the right shows the scatter plot for 75 percentile and 25 percentile for each group of features dimensions (grouping by all features).

It clearly shows that information is still diffused using these features and predictions can’t be good.



Improvements

ETL

1. Transformation should be done using line separated json instead of pandas. Line separated json will make it more robust. Pandas transformations fail on one single line of error.

Model:

I did not come up with a working model since the features I selected are not the best ones. Grouping by all the features produces large standard deviations.

0. Should check other features
1. Should consider the following factors:
   1. Weather: rain, temperature;
   2. Geolocations: I created a geolocation tool using OSM data.
github.com/emptymalei/geoeconomics-with-data
   3. Holiday data

Backup

1. About API creations: Since this was mentioned in the challenge, I have created many API in the past. I can create RESTFUL and serverless APIs. I have a very simple toy of serverless API: https://github.com/InterImm/marsapi-serverless But I also created very complicated serverless APIs during work.