Rideindego Bike Sharing Data

Lei Ma

Outline

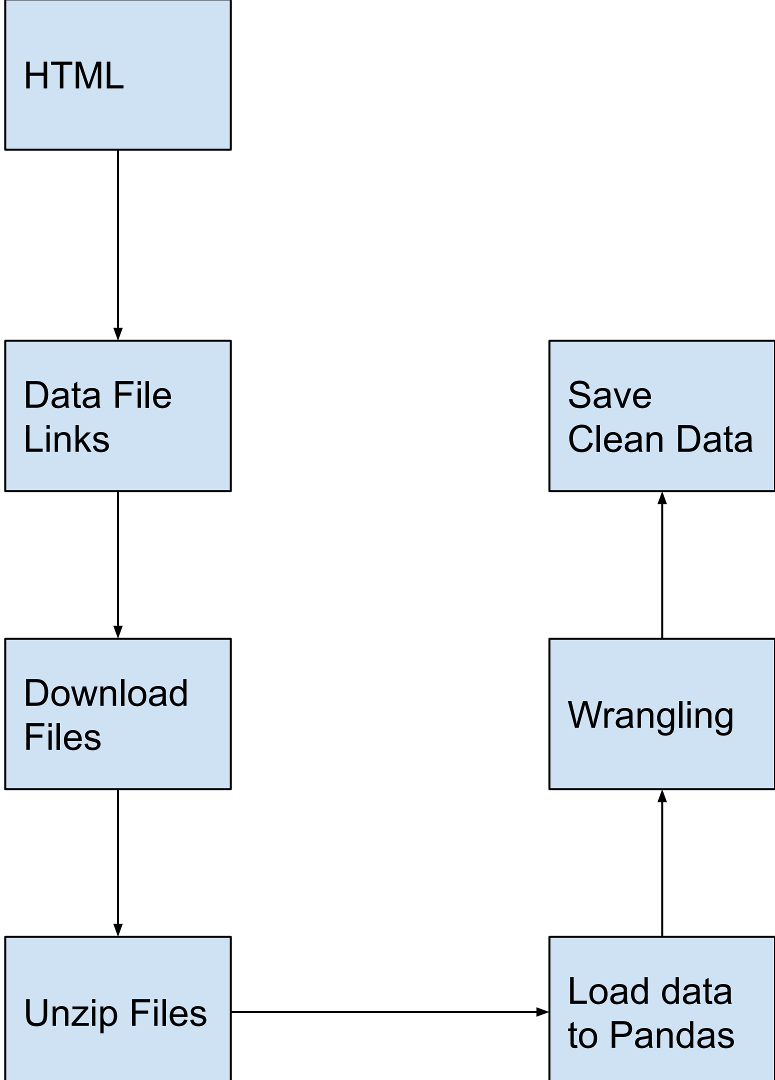
1. Data Retrieval and Wrangling
2. EDA
3. Regression
4. Improvements
   a. Classification
   b. Include Weather Data
   c. NN
5. What did I Learn
6. Further Improvements

Data Retrieval and Wrangling

Code: `python app/get_indego_data.py`

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: hour of day, day of the week, month of year



HTML

Data File Links

Download Files

Unzip Files

Load data to Pandas

Wrangling

Save Clean Data

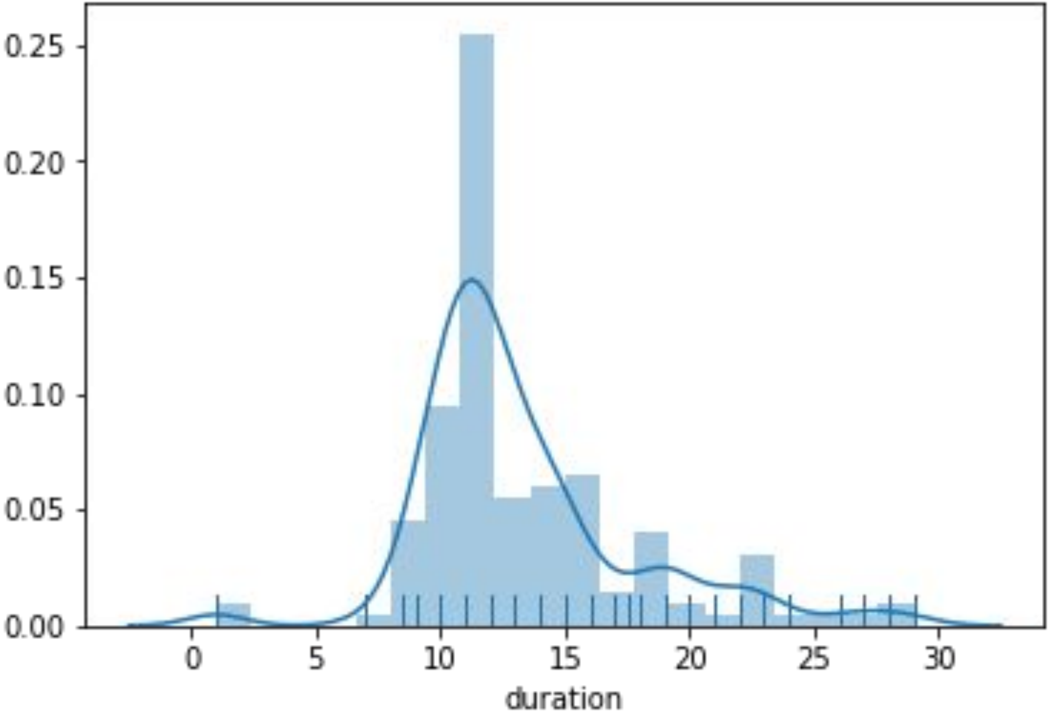
EDA

Two types of bikes:

- **standard bikes**
- ~~electric bikes~~

EDA

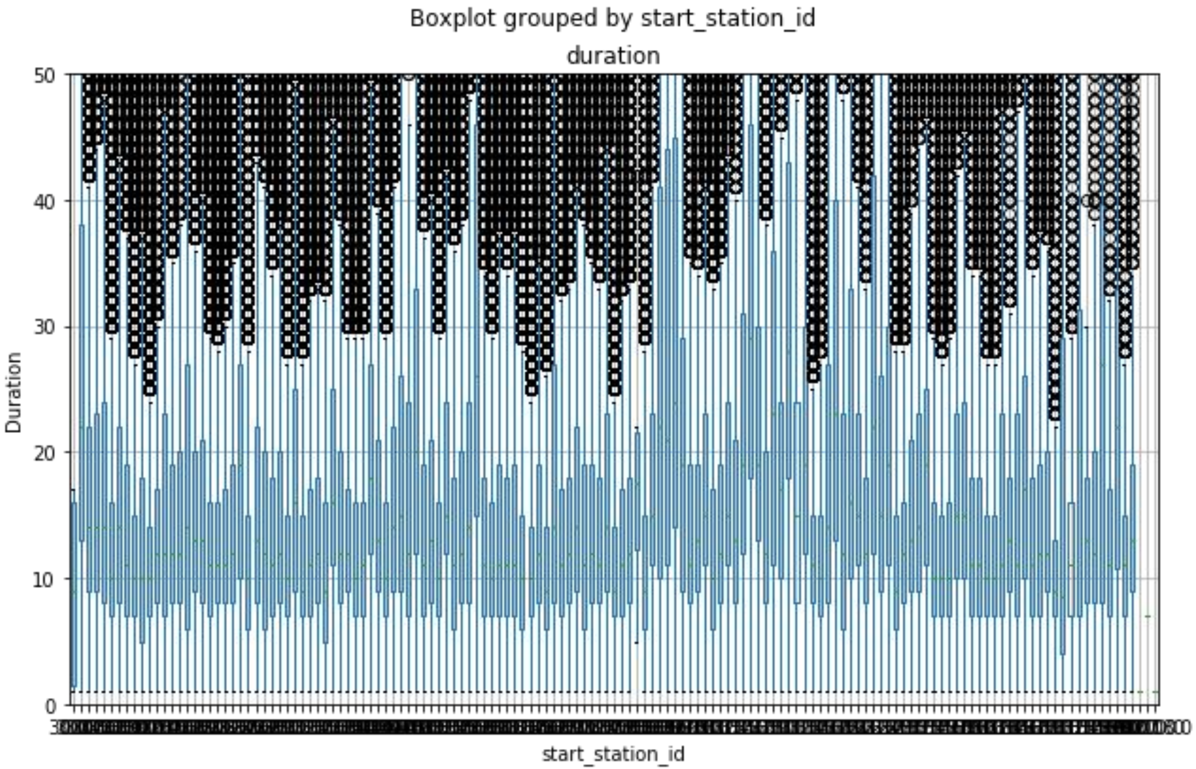
Duration is around 12 min



EDA

Bike stations is related to geolocation.

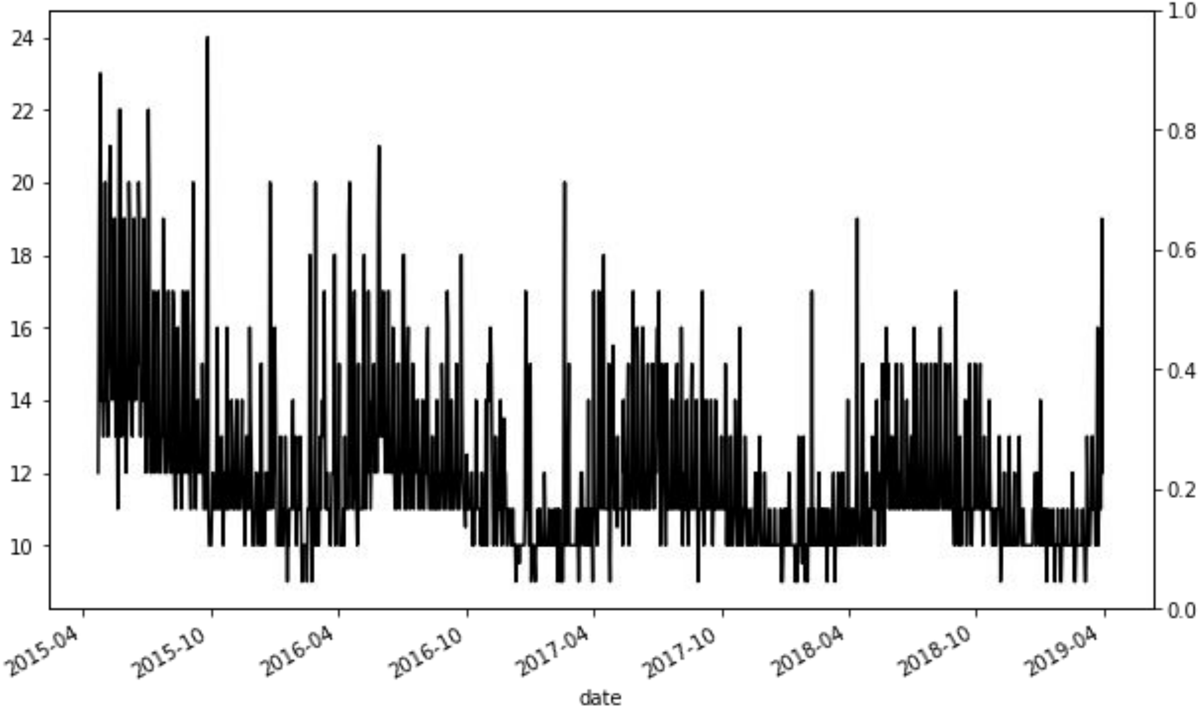
We will leave this one out for now.



EDA

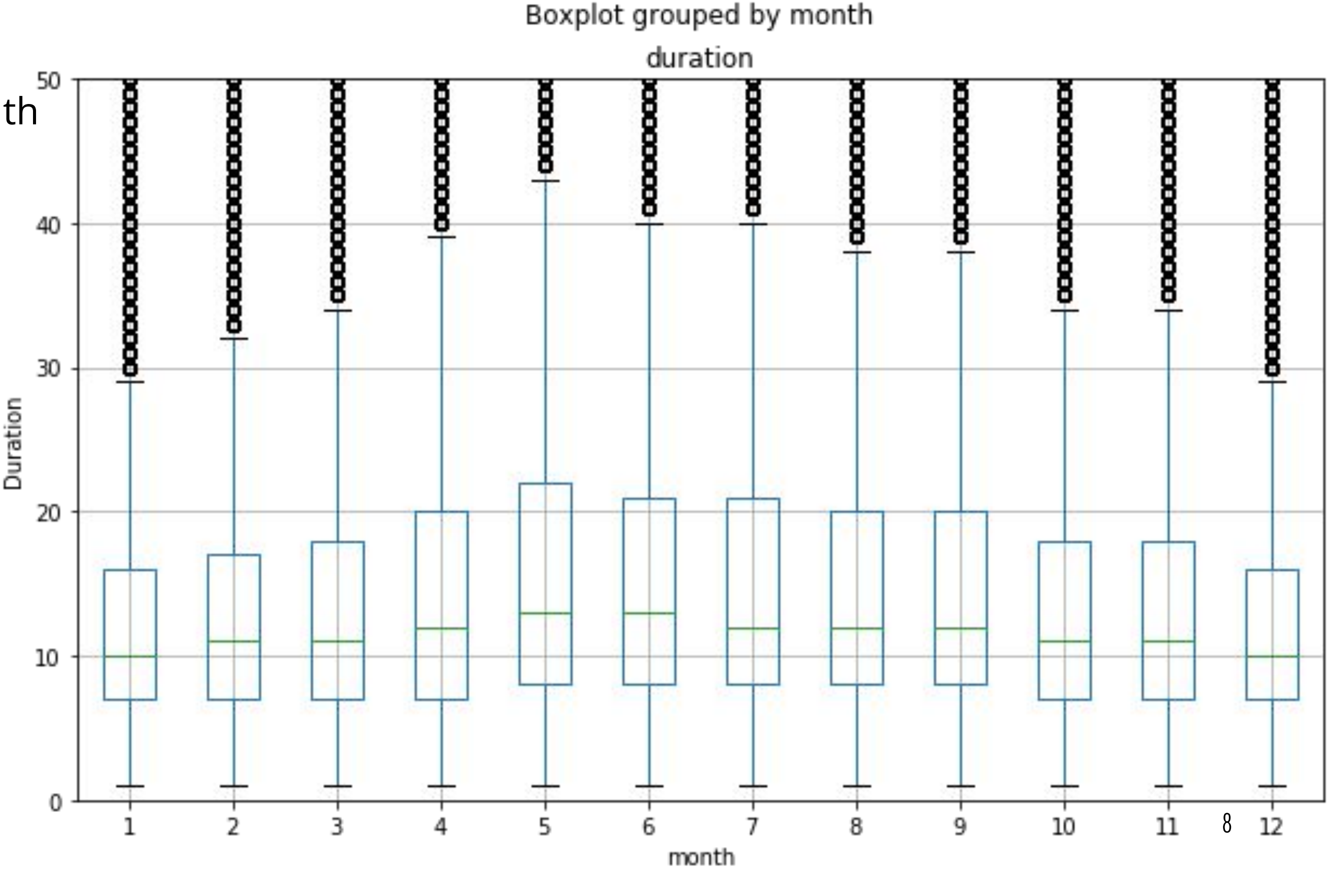
Seasonality: duration is longer in late spring and early summer. (~15 degrees)

Vertical axis shows the median durations for each day



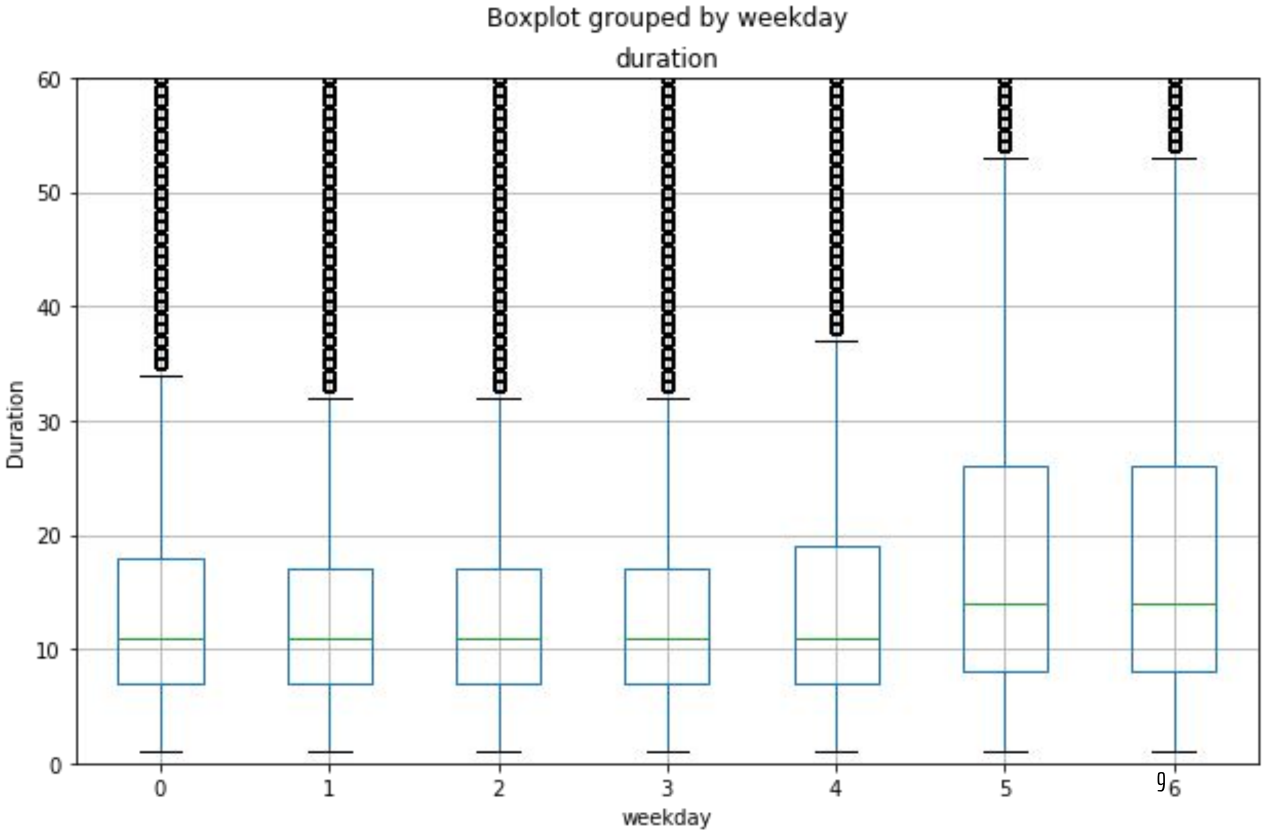
EDA

Duration for each month



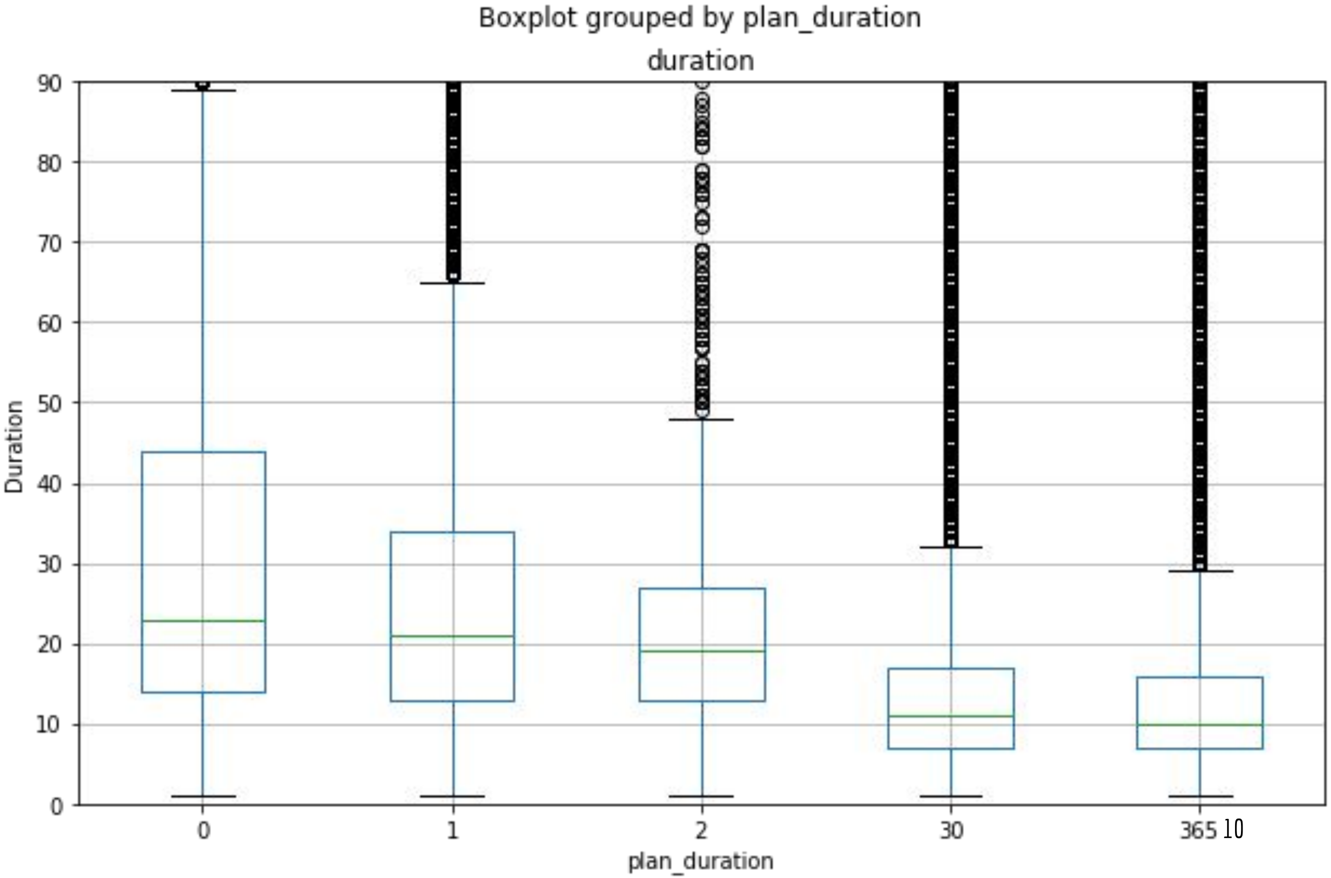
EDA

Duration is longer on weekends.



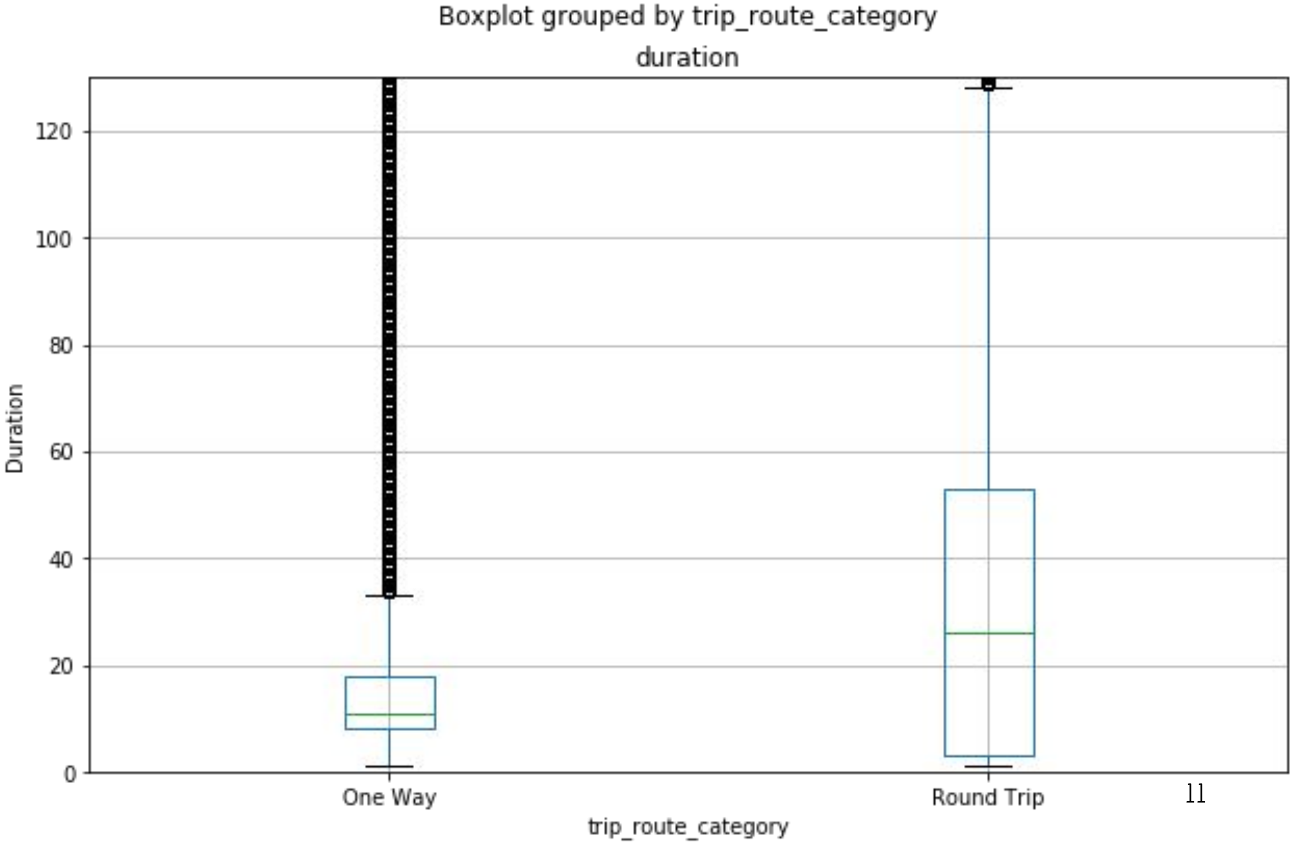
EDA

User plans matters



EDA

Trivial result: round trip doubles the duration



Regression

Ridge regression not working

R^2 score:
0.036

Not really much better than random guess.

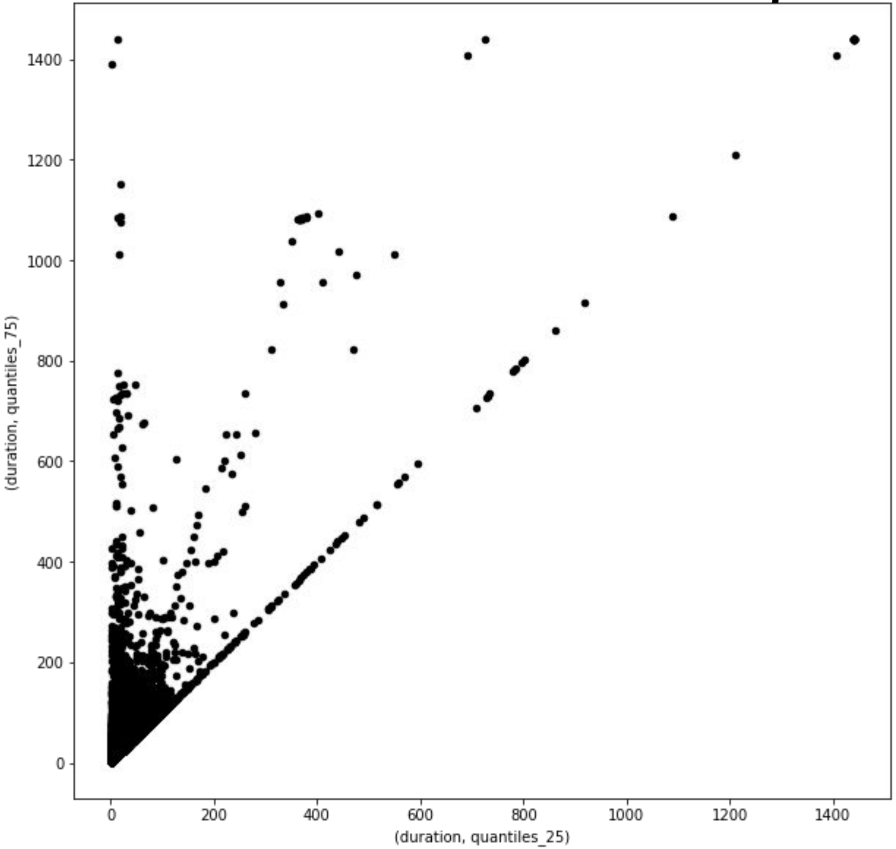
1. OneHotEncoding
2. Using features: 'passholder_type', 'trip_route_category', 'hour', 'weekday', 'month'

Improvement: Why aren’t the models working?

(75 percentile, 25 percentile) for each group of features dimensions (grouping by all features).

Information is still diffused using these features.

1. ~~Not enough data (137 records for each group on avg)~~
2. Bad feature selection
3. Bad model



Improvement: Models and Features

Information is dispersed

Use classifications (granular data):

1. Segmentation of the duration
2. Treat the problem as a classification problem. Kind of cheating but we could at least get some information out of the data.

Add more features:

1. Weather: rain, temperature, wind, humidity, etc;
2. Geolocations: if would be almost like cheating to predict the time using the actual geolocations of the start and end stations. But the start stations are some information we could use. There is the station data as geojson, which has address info etc.
3. Holiday data: Christmas, etc.

Improvement: Classification

Model: Decision Tree

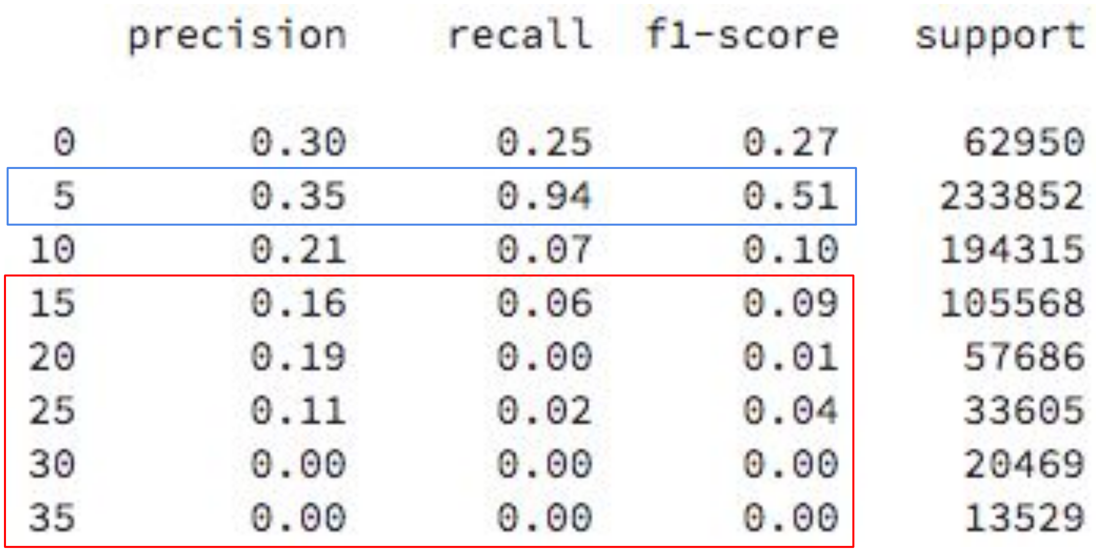
Parameters:

- Max depth: 6

Accuracy: 0.318

feature importance:

1. passholder_type: 0.19
2. trip_route_category: 0.14
3. hour: 0.33
4. weekday: 0.16
5. month: 0.18



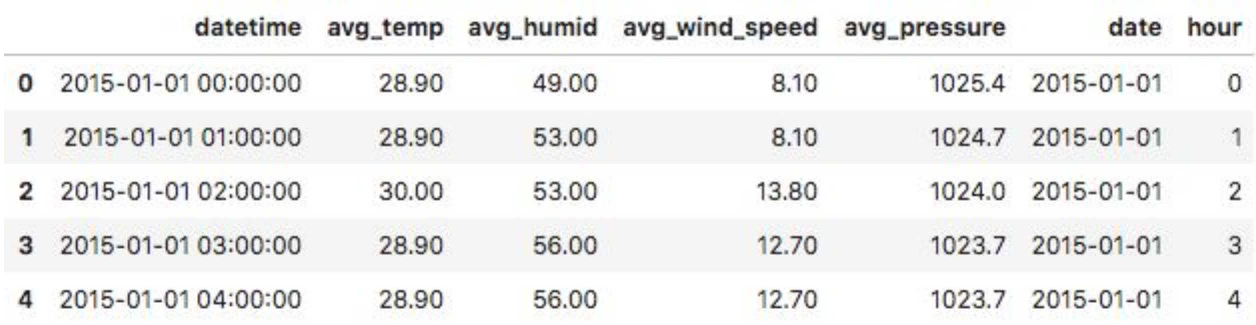
```
      precision    recall  f1-score   support

 0         0.30      0.25      0.27     62950
 5         0.35      0.94      0.51    233852
10         0.21      0.07      0.10    194315
15         0.16      0.06      0.09    105568
20         0.19      0.00      0.01     57686
25         0.11      0.02      0.04     33605
30         0.00      0.00      0.00     20469
35         0.00      0.00      0.00     13529
```

Improvement: Add Weather Data

Data Source:
http://www.climate.psu.edu/data/ida/index.php?t=3&x=faa_hourly&id=KPHL

Preprocessing:

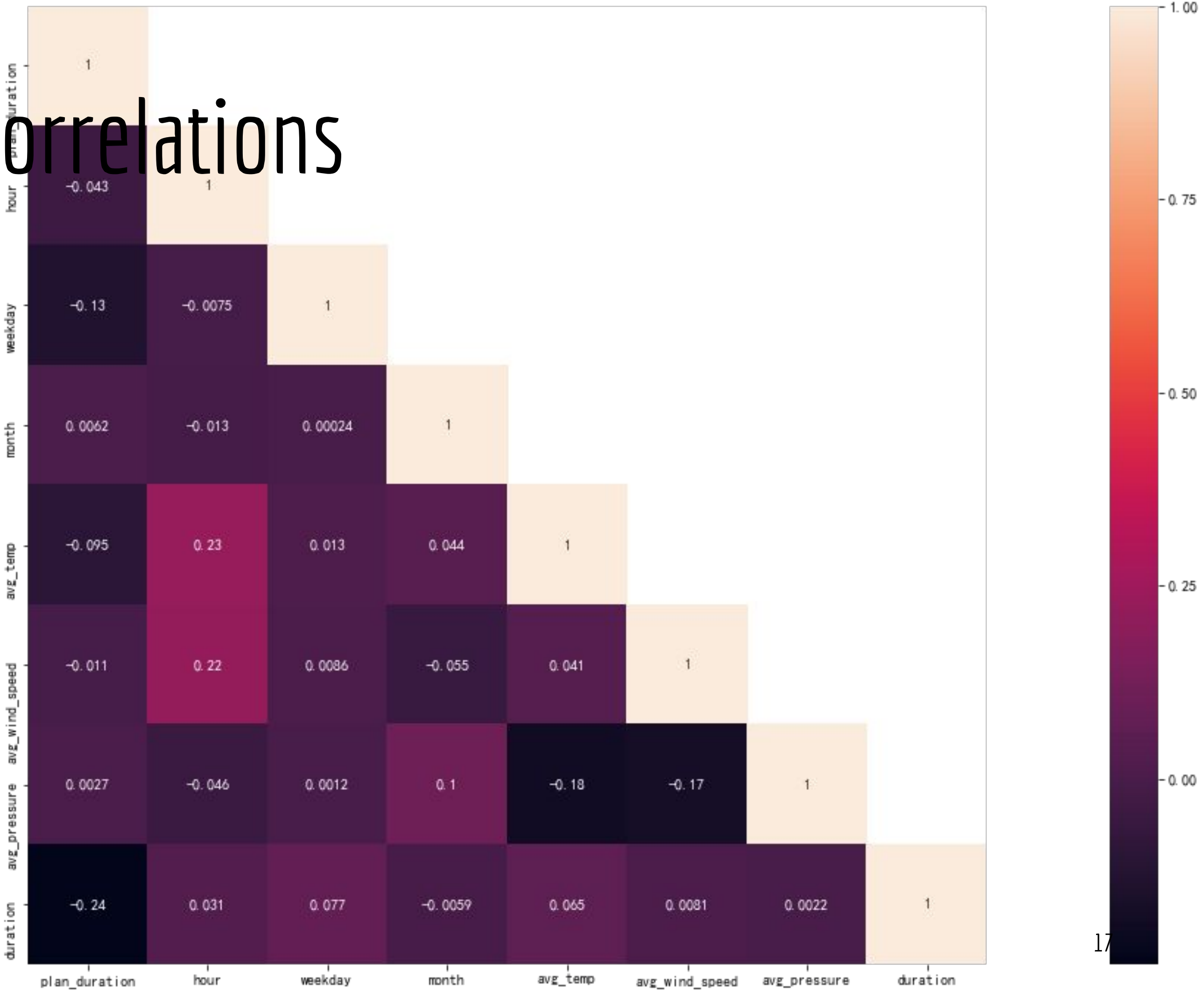
1. Add rows for date and hour



	datetime	avg_temp	avg_humid	avg_wind_speed	avg_pressure	date	hour
0	2015-01-01 00:00:00	28.90	49.00	8.10	1025.4	2015-01-01	0
1	2015-01-01 01:00:00	28.90	53.00	8.10	1024.7	2015-01-01	1
2	2015-01-01 02:00:00	30.00	53.00	13.80	1024.0	2015-01-01	2
3	2015-01-01 03:00:00	28.90	56.00	12.70	1023.7	2015-01-01	3
4	2015-01-01 04:00:00	28.90	56.00	12.70	1023.7	2015-01-01	4

Improvement: Correlations

Kendall tau



Improvement: Classification with Weather Data

Model: Decision Tree

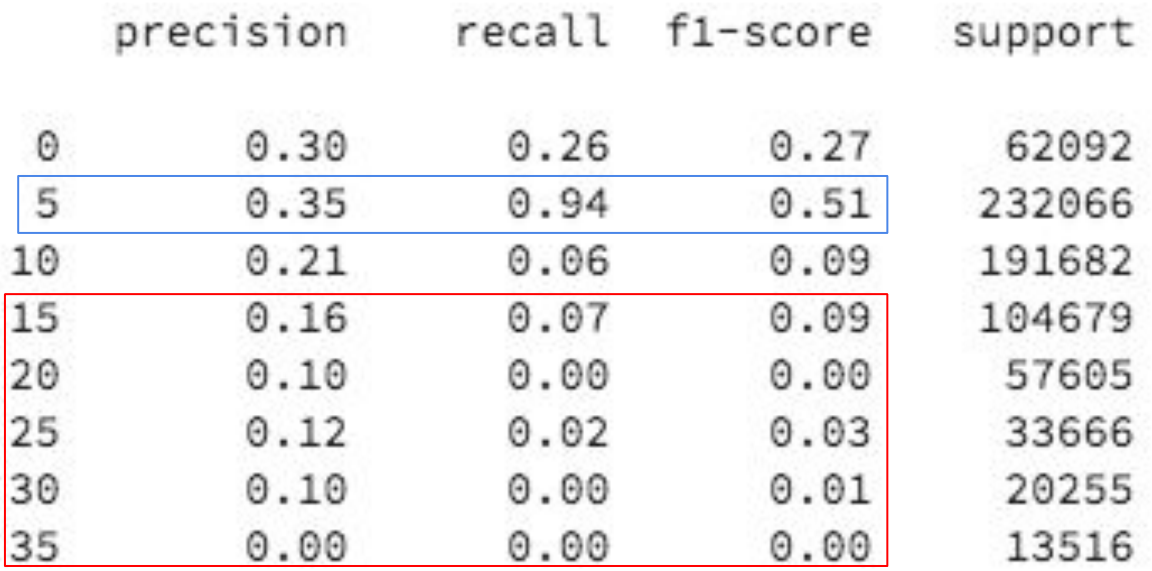
Parameters:

- Max depth: 7

Accuracy: 0.324

Feature importance:

1. passholder_type 0.16
2. trip_route_category 0.12
3. hour 0.034
4. weekday 0.13
5. month 0.1
6. avg_temp 0.21
7. avg_wind_speed 0.14
8. avg_pressure 0.1



```
         precision    recall  f1-score   support

 0            0.30      0.26      0.27     62092
 5            0.35      0.94      0.51    232066
10            0.21      0.06      0.09    191682
15            0.16      0.07      0.09    104679
20            0.10      0.00      0.00     57605
25            0.12      0.02      0.03     33666
30            0.10      0.00      0.01     20255
35            0.00      0.00      0.00     13516
```

Improvement: Does NN Work?

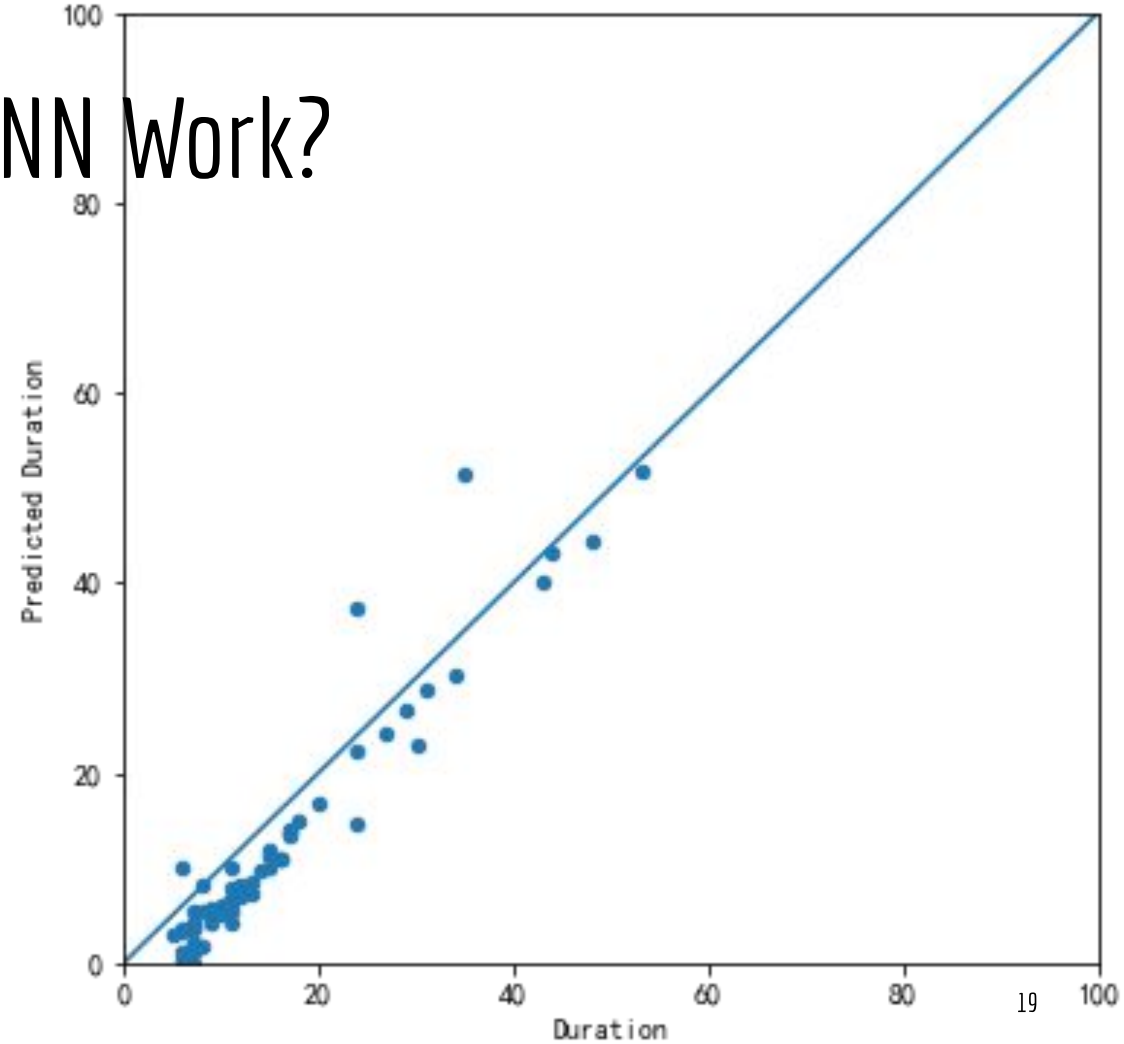
Model: NN (Feed forward)

Parameters:

- Layers: 2
- Dimensions: 50, 100 respectively
- Activation func.: relu

R^2 Score: 0.888

Reference:
https://github.com/yashu-seth/pytorch-tabular



What Did I Learn

Domain

1. (TRIVIAL) Everything from EDA
    a. Duration ~ 12min
    b. Seasonality: ride longer in late spring + early summer
    c. Weekend: longer duration
2. Temperature doesn’t improve the decision tree models a lot. (rain might)

About the models:

1. NN works fine: probably means we have to reengineer features for the tree-based method to work well
2. Need to break down the NN to check the focus. I might need to transform to another space for features.

Other Improvements

ETL

1. Robust ET(L): Transformation should be done using line separated json instead of pandas.
2. Reproducibility and easy management: Dockerized/serverless pipeline.

Predictions:

1. Should consider the following factors:
    a. Weather: rain
    b. Geolocations
    c. Holiday data

References

1. https://github.com/yashu-seth/pytorch-tabular
2. https://github.com/Lanbig/bike-sharing-prediction
3. Finseth, P. (2018). Evaluating Machine Learning Methods for City Bike Demand Prediction in Oslo Lasse Drevland, (June).