Machine Learning Engineer Nanodegree

Capstone Project: Bicycle-Sharing System Analysis and Trip duration Prediction

---

Spyridon Kostis, September 2020

1. Problem Statement

The bike-share service is among the most popular, alternative ways to move through cities. Year by year, the number of cities that a person can find a bike to share, via specific applications, are increasing. In this project, I am working with data from FordBike, a bike-share service company, operating in the city of San Francisco. In this project, I present an Machine-Learning model that predicts the time a rider will use a bike, by predicting the ride duration. The model takes as inputs various values, like the time of the day, the starting station, the age of the biker etc. and predicts, based on its training, the duration of the ride is about to take place.

I think this is a very useful real-life project, production oriented, that can help bike owner companies to have better planning for its bikes fleet. By knowing the duration, the company, among others, can answer questions like: What is the most possible station for a rider to arrive? How many bikes will I have in a specific area in the future? What is the need for bikes in this area?

2. Analysis

Data Exploration

This is a preview of the current dataset followed by the description of the basic characteristics of the variables.

duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name
52185	2019-02-28 17:32:10.1450	2019-03-01 08:01:55.9750	21.0	Montgomery St BART Station (Market St at 2nd St)	37.789625	-122.400811	13.0	Commercial St at Montgomery St
42521	2019-02-28 18:53:21.7890	2019-03-01 06:42:03.0560	23.0	The Embarcadero at Steuart St	37.791464	-122.391034	81.0	Berry St at 4th St
61854	2019-02-28 12:13:13.2180	2019-03-01 05:24:08.1460	86.0	Market St at Dolores St	37.769305	-122.426826	3.0	Powell St BART Station (Market St at 4th St)
36490	2019-02-28 17:54:26.0100	2019-03-01 04:02:36.8420	375.0	Grove St at Masonic Ave	37.774836	-122.446546	70.0	Central Ave at Fell St
1585	2019-02-28 23:54:18.5490	2019-03-01 00:20:44.0740	7.0	Frank H Ogawa Plaza	37.804562	-122.271738	222.0	10th Ave at E 15th St

	duration_sec	start_station_id	start_station_latitude	start_station_longitude	end_station_id	end_station_latitude	end_station_longitude	bike_id
count	183412.000000	183215.000000	183412.000000	183412.000000	183215.000000	183412.000000	183412.000000	183412.000000
mean	726.078435	138.590427	37.771223	-122.352664	136.249123	37.771427	-122.352250	4472.906375
std	1794.389780	111.778864	0.099581	0.117097	111.515131	0.099490	0.116673	1664.383394
min	61.000000	3.000000	37.317298	-122.453704	3.000000	37.317298	-122.453704	11.000000
25%	325.000000	47.000000	37.770083	-122.412408	44.000000	37.770407	-122.411726	3777.000000
50%	514.000000	104.000000	37.780760	-122.398285	100.000000	37.781010	-122.398279	4958.000000
75%	796.000000	239.000000	37.797280	-122.286533	235.000000	37.797320	-122.288045	5502.000000
max	85444.000000	398.000000	37.880222	-121.874119	398.000000	37.880222	-121.874119	6645.000000

The dataset contains the following columns:

Trip Duration (seconds) (int64)
Start Time and Date (obj)
End Time and Date (obj)
Start Station ID (float64)
Start Station Name (obj)
Start Station Latitude (float64)
Start Station Longitude (float64)
End Station ID (float64)
End Station Name (obj)
End Station Latitude (float64)
End Station Longitude (float64)
Bike ID (int64)
User Type (Subscriber or Customer) (obj)
Member birth year (float64)
Gender (obj)
Bike share for all trip (obj)

Quality and tidiness issues

The dataset hasn't got any duplicated data, but there are many missing values. I decided to delete the rows with the missing data.

Firstly, I prepared the data for the prediction phase. I created some more columns, in order to help the prediction algorithm to make more precise predictions:

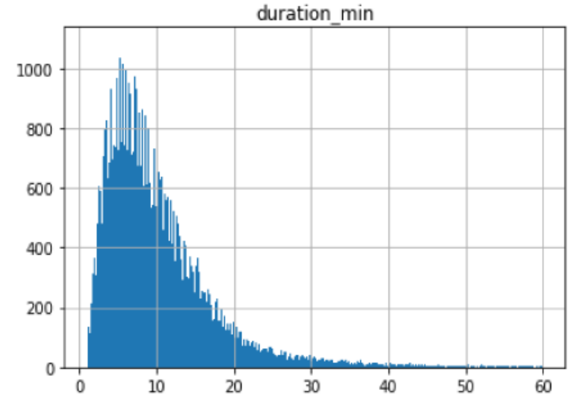
1. I made a column called 'Customer_age' in order to make the data more user friendly.
2. I transformed the ride duration from seconds to minutes.
3. I made a new categorical column named 'Customer_decate' ,separating the ages by decade.
4. I separated the start date and month. "hour start", 'day_start','month start' to help our analysis.
5. I created a column called “day_period” to separate the hours in 4 categories. Morning, Midday, Afternoon, After midnight.

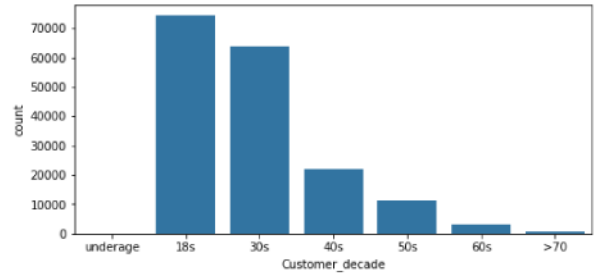
Exploratory analysis

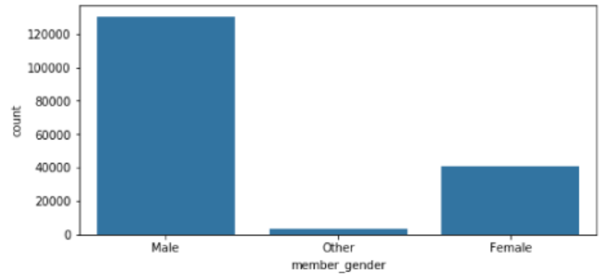
The columns of the dataset, as formatted after the data cleaning process.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 174952 entries, 0 to 183411
Data columns (total 20 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   duration_min             174952 non-null  float64
 1   start_station_id         174952 non-null  float64
 2   start_station_name       174952 non-null  object
 3   start_station_latitude   174952 non-null  float64
 4   start_station_longitude  174952 non-null  float64
 5   end_station_id           174952 non-null  float64
 6   end_station_name         174952 non-null  object
 7   end_station_latitude     174952 non-null  float64
 8   end_station_longitude    174952 non-null  float64
 9   bike_id                  174952 non-null  int64
 10  user_type                174952 non-null  object
 11  member_gender            174952 non-null  object
 12  bike_share_for_all_trip  174952 non-null  object
 13  Customer_age             174952 non-null  float64
 14  general_runtime          174934 non-null  category
 15  Customer_decade          174951 non-null  category
 16  hour_start               174952 non-null  int64
 17  day_start                174952 non-null  int64
 18  month_start              174952 non-null  int64
 19  day_period               174952 non-null  category
dtypes: category(3), float64(8), int64(4), object(5)
memory usage: 24.5+ MB
```

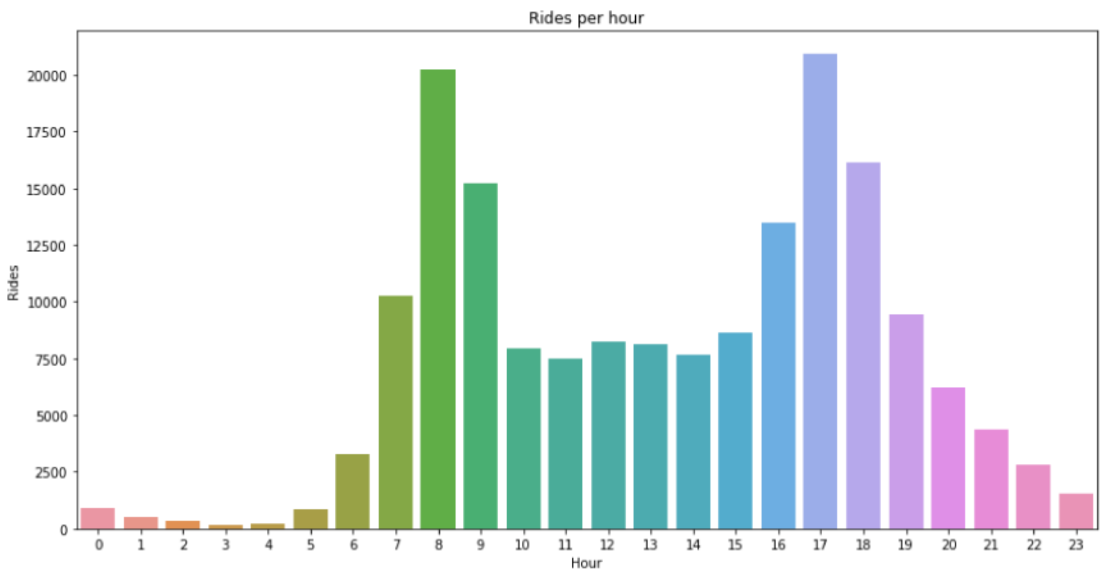
I'm most interested in figuring out how every different biker group (gender, age, customer-subscriber) behaves and how the bike use changes through time.

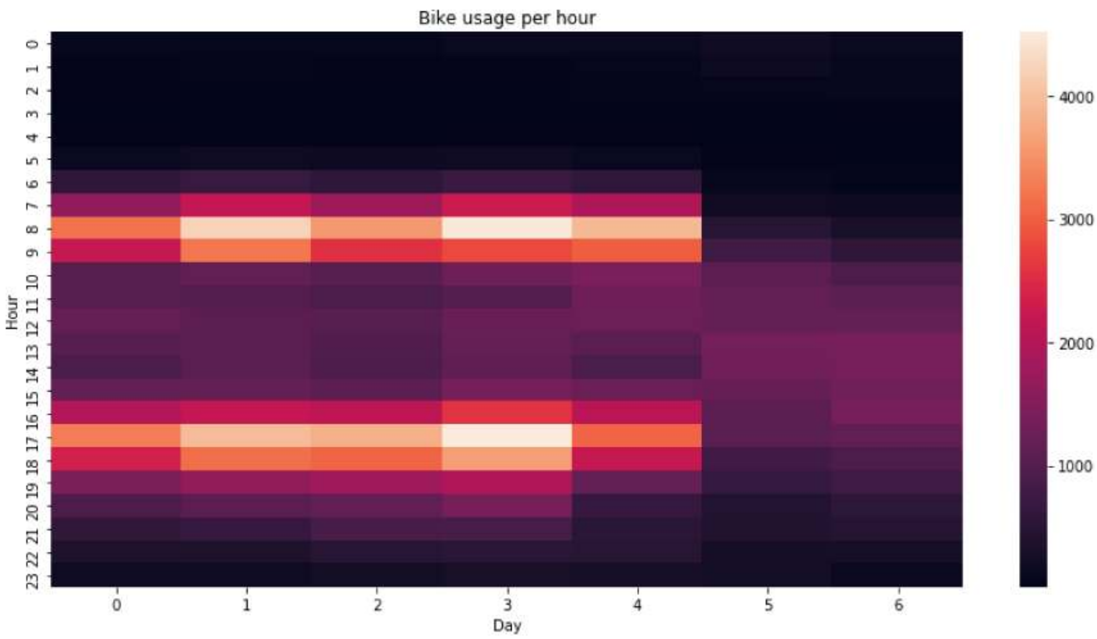


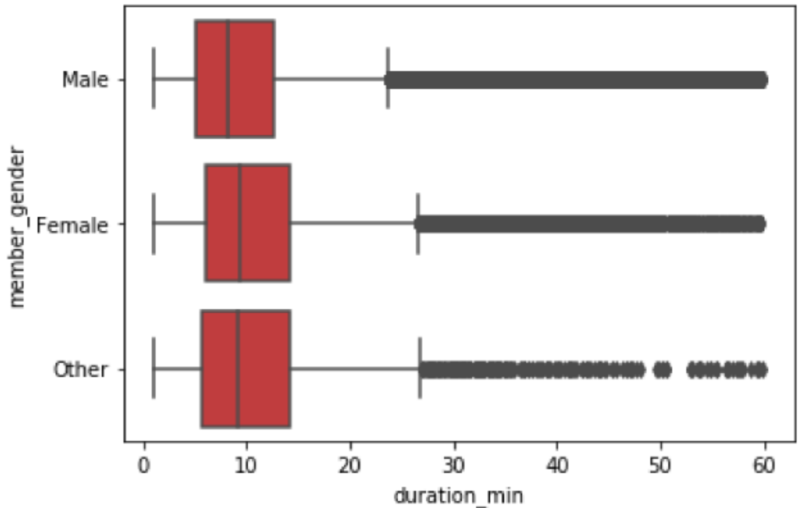




- We can see that the majority of the rides are between 3 and 10 minutes long.
- According to the genre, the riders are mainly men.
- The bikers are mainly young with the majority in their 20s-30s.
- We can see that the busiest hours of a day are between 07:00 and 10:00 in the morning and 16:00 and 19:00 in the afternoon. We can assume that this indicates that bikes are used by the city's residents to move to and from their workplaces. That is more obvious at the following heatmap, which indicates that at the weekends, the rides are noticeable less.







This graph indicates the mean ride duration per genre for riders younger than 60 years old. We can see that men make shorter journeys.

3.Algoriths and Metrics.

Algorithms

Our main algorithm will be XGBoost .

"XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate way. The same code runs on a major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples."

Source: https://xgboost.readthedocs.io/en/latest/index.html

“Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.”

Source: https://en.wikipedia.org/wiki/Gradient_boosting

For our benchmark model, we will use the Linear Learner Algorithm.

“Linear models are supervised learning algorithms used for solving either classification or regression problems. For input, you give the model labeled examples (x, y). x is a high-dimensional vector and y is a numeric label. The algorithm learns a linear function and maps a vector x to an approximation of the label y.”

Source: https://docs.aws.amazon.com/sagemaker/latest/dg/linear-learner.html

Evaluation Metric

For the evaluation of our model predictions, we will use the Mean Absolute Error (MEA) metric. Given the fact that this is a regression problem and we would like to take into account deviations both pain and below the current price, I believe that the MAE metric is the most appropriate.

Mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of *Y* versus *X* include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. MAE is calculated as:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$ [1]

Source:https://wikimedia.org/api/rest_v1/media/math/render/svg/3ef87b78a9af65e308cf4aa9acf6f203efbdeded

4.Model Training Process

Preparing the data for the training.

Before separating the database for training and testing, I deleted the “start_station_name” column because I think that the starting station’s longitude and latitude columns are enough to determine the uniqueness of the starting point of the route, without the need for the column “start_station_name” that loads my database with a lot of extra,unhelpful, data.

After that, I transformed the categorical columns to numerical, using the get_dummies method and deleted any columns that reveal the duration of the ride ('end_station_latitude', 'end_station_longitude', 'general_runtime')

Taking in mind the number of data samples, I decided to keep 81% of the data as training dataset, 10% as testing dataset and 9% as validation.

	duration_min
duration_min	1
start_station_id	-0.0093
start_station_latitude	0.011
start_station_longitude	-0.039
bike_id	0.019
Customer_age	0.015
hour_start	0.003
day_start	0.017
member_gender_Female	0.023
member_gender_Male	-0.032
member_gender_Other	0.03
bike_share_for_all_trip_No	-0.012
bike_share_for_all_trip_Yes	0.012
Customer_decade_18s	-0.0045
Customer_decade_30s	-0.0049
Customer_decade_40s	0.0037
Customer_decade_50s	0.0091
Customer_decade_60s	0.0085
Customer_decade_>70	-0.0015
Customer_decade_underage	0.0097
day_period_aftermidnigt	-0.0035
day_period_afternoon	-0.00023
day_period_midday	0.011
day_period_morning	-0.0048
day_period_night	-0.0049

Variables correlation

The correlation between the duration and the other variables are not remarkably strong. This will affect our results negatively.

Model Training

Starting the model training process, I had to select the starting values for our model as below. For better results, I had used AWS Sagemaker's, hyperparameter tuning tool.

Base model parameters	Parameters tuning values
• max_depth=5 • eta=0.2 • gamma=4 • min_child_weight=6	• max_depth: 3 - 12 • eta: 0.05-0.5 • min_child_weight': 2- 8 • subsample: 0.5- 0.9

• subsample=0.8, • silent=0, • objective='reg:linear', • eval_metric='mae', • early_stopping_rounds=10, • num_round=500	• gamma: 0-10

Results

Comparing results with the benchmark model we can conclude that our model performs better from the Linear Learner algorithm.

Additionally, we can compare our model's performance with previous work on the same field. I refer to the "Bicycle-Sharing System Analysis and Trip Prediction" project. (https://arxiv.org/pdf/1604.00664.pdf)

The XGBoost's MEA

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, predictions)
```

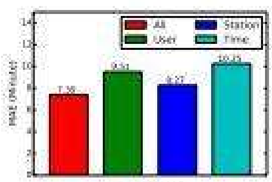
```
5.168976366353095
```

The Linears_Learner's MEA

```
: linear_mae=evaluate(linear_predictor, test_x_np, y_test, verbose=True)
  linear_mae
```

```
: 5.56932102661531
```

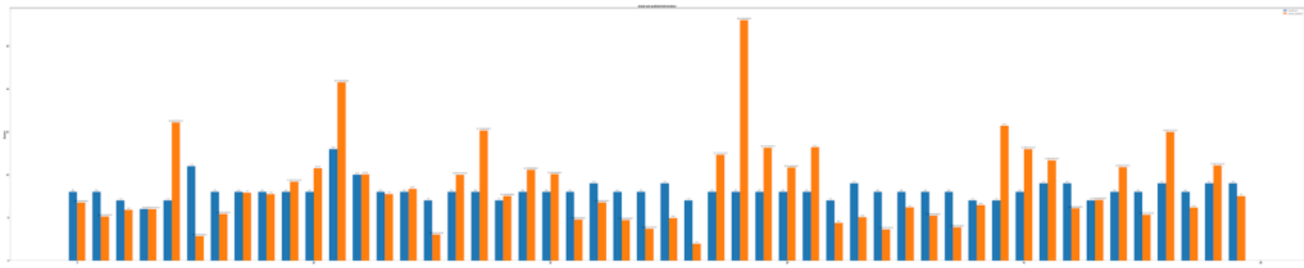
The compared's article MEA



(a) MAE

Model	MAE
XGBoost model	5.13
Benchmark model	5.56
Model from Compared Article	10.25

In the following graph, we can see that the predicted values (blue) do not have a big variance and are close to the mean value. This is more obvious when the actual value (orange) is extremely away from the mean and our model is not able to predict such unusual prices.



5.Ideas for model optimization.

We have already made the conclusion that our variables' correlations are not too strong. Additionally I believe that by using the Sagemaker's Huperparameters tuning tool we have already picked the best parameters for our model.

Taking those in mind, I think that the best way to move, for better results, is towards additional information to our dataset, that will help our model. That information can be weather information, because I assume that the weather conditions greatly affect the duration of the routes. Furthermore, taking in mind the history of the specific subscriber that is using the bike, will make our predictions much more accurate.

6.Ideas for real application of this project.

Knowing the duration of a single ride can give the owner company the ability to optimize its fleet management. At the same time, it can draw more conclusions, such as the possible destinations of this particular rider. Knowing this, can predict the needs of bikes in a specific area.

7.Conclusion.

Trying to make regression predictions, with so weak correlation between the variables, does not guarantee the best results. The result of our metric is not good enough for a real life application. Even Though, our model is performing better than the compared models. Taking that in mind, compared with the suggestions above for better optimization, I believe that this work is a good base to work on, with great potential in the future.