Bike Sharing Demand Prediction with Decision Trees and Neural Network

Alon Itach *alon.itach@mail.huji.ac.il*
Ben Cohen *ben.cohen3@mail.huji.ac.il*
Franziska Wehrmann *franzisk.wehrmann@mail.huji.ac.il*
Kfir Gil *kfir.gil@mail.huji.ac.il*

***Abstract*—This project shows how to model and predict the demand of bikes at stations of a public bike sharing system. The aim is to predict for every station the demand of bikes throughout the day under different circumstances.**
The problem is modeled as a categorical decision problem where the attributes contain information about time, location, type of weekday and weather. Due to the categorical nature of the attributes, it is possible to build decision trees on historical data and use those trees to predict the future. In order to prevent overfitting, a variety of different trees were created. Those multiple predictions are then combined via a Neural Network to one final prediction.
Even though the prediction has only an accuracy of about 70%, the incorrect predictions are very close to the actual values and therefore can still be seen as a successful prediction. 70% of the incorrect predictions are only off by 1.

I. Introduction

Public bikesharing systems emerged in many cities around the world and become more and more popular. It combines multiple benefits like going by bike for recreational or health purposes, not being stuck in traffic, high flexibility (especially when combined with public transport), environmental reasons, not having to take care of your own bike. Those reasons especially apply in big cities and gained in people's priorities in the last years.
But it is difficult to find the perfect balance between a sufficient supply of bikes to keep the customers happy and having too many bikes that litter up the city [1]. It also lies in the interest of the bike sharing company to perform economically: to have enough bikes for every arriving customer, but to have just enough bike in order to save maintenance costs.
This article first explains which data sources were used to build a model. Section III explains which algorithms are used to solve the problem and why, followed by the display and discussion of the prediction in the next section and the discussion of some problems and further outlook.

II. Modeling of the Problem

We are interested in the demand of bikes in the next time window on a specific station. Here the data of Capital Bikeshare, a bikesharing provider in Washington D.C., was used. [2]
We get this information through processing of the chronological log book of trips, where every row represents the activity of one bike - when and at which station it got picked up and when and where it got returned [3]. Counting how many bikes were picked up at one station during a certain time window gives exactly the information that we are interested in.
To get a more precise prediction, not only time, but also other attributes should be taken into account. An explanation of the attributes follows below:

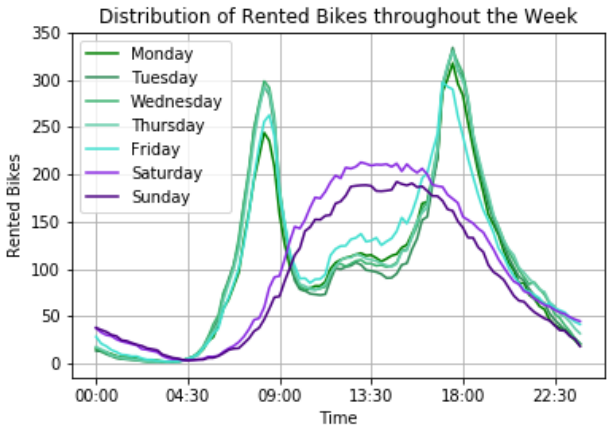

Fig. 1. Distribution of bike trips throughout the week in 15 minutes intervals. Averaged over the year of 2018. The distribution shows clearly that there is remarkable difference in the behavior on working days (Monday to Friday, green-blue shades) and on the weekend (Saturday and Sunday, purple shades). During the weekdays, the Distribution of trips peaks in the morning at around 8:30 and in the evening at around 17:30, whereas on the weekend the trips peak at noon around 13:30.

A. Time

Since we want to predict the demand of bikes for the next time window, it is worth it to explore the distribution of bike trips over time. Figure 1 shows how many bikes got picked up around that time at any station across the whole city (15 minute intervals). For this graph only the trip history of year 2018 was used.
The graph shows clearly, that the distribution of bike pickups on workdays (Monday to Friday) is different to the distribution on the weekends (Saturday and Sunday).
During the week, the graph shows a high activity in the mornings (sharp peak around 8:30) as well as in the evening (sharp peak around 17:30). On the weekend the activity is the highest around noon, but does not peak as sharply as during the week. This might be due to people that use the bike to commute to and from work on workdays and use the bike for leisure on the weekend.
With this graph it becomes clear that not only time is an

important attribute, but also the day of the week should be part of the model.

B. Location

Another aspect of modeling the activity on stations is to take its location into account. The theory is based on the purpose of why a person would rent a bike. Is it for commuting, then we would see a higher activity at stations that are around work places. Also certain areas of a city are more pleasant for bikerides. It is expected that close to parks the activity is higher and stations far outside the city are less active (too far to commute to work, need to use other way of transport to get inside the city, people are more likely to have their own bike if they want to go on a ride for leisure. We want to emphasize, that those points are merely the motivation of why we introduced the location feature, the reasons for different behavior on stations can not be read directly from the data).
Motivated by this theory, figure 2 shows the demand on the stations across the city on a normal Tuesday in May 2018. The exact calculation of the demand is explained in section II-D, for now it is enough to mention that a negative number (red) means that there are this amount of bikes missing at a station during this 90-minutes time interval, a positive number (blue) means that bikes are unused.
As expected we see stronger values in the center of the city. In the morning we also see that the outer part of the inner city is dark red, which means that more bikes got picked up at those stations than returned. 'People pick up a bike to commute to work, to the center.', in the center we see a few stations that are dark blue, which means that they have many unused bikes 'Many bikes arrived because people came to work'. In the evening those stations are dark red 'People take bikes to get home from work' and the outer inner city becomes more blue 'more bikes arriving than are picked up'.

Using the coordinates as categorical location attributes is not a good idea since there are far too many different values. Also the precise location of the station is not important to us, it is only important in which area the station is. Therefore we introduce a 'hierarchical' location splitting.
In the first layer 'L1' the map is split into a grid of 4x4 cells and all stations inside this cell belong to this L1-group. The grid is exactly the grid how it is displayed in figure 2. We see that not all of the cells contain stations, so we end up with even less than 16 values for the attribute L1.
Some of the cells contain many stations and as we see in the figure, the splitting in 4x4 cells is not granular enough. For the second layer 'L2' we pick those L1 areas that are dense and split them again in a 4x4 grid.
Instead of using the coordinates of the station, we now use the information to which L1 and L2 area the station belongs to. This is how we managed to downscale the dimension of the location attributes drastically without loosing too much information.

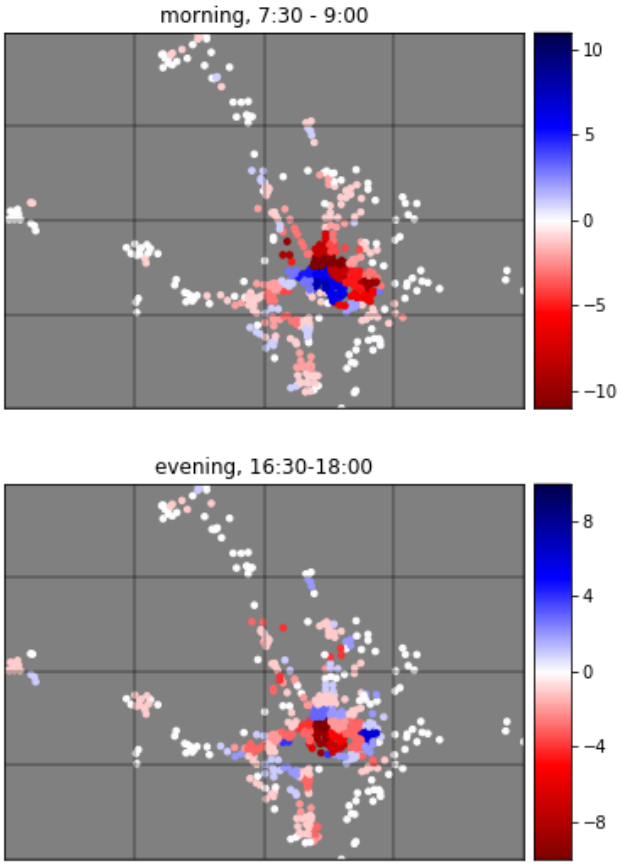

Fig. 2. Demand of bikes per station during morning and evening on a working day. Negative numbers (red) mean that in the next 90 minutes this station will need additional bikes, positive numbers (blue) indicate an existence of unneeded bikes.

Time, Weekday and Location are probably the most important attributes for the prediction, but it is also worth it to consider other conditions that will have an influence on the decision of a person if he is going to rent a bike.

C. Weather Data

Clearly the weather has influence on the decision of if a person would go by bike or not. During unpleasant weather like snow, strong rain, thunderstorm, strong wind or extreme cold/heat, less people will be willing to use a bike.
We used an API to retrieve the historical weather at the closest airport. Even though the airport is a few miles outside of the city, the weather is still representative enough to be used for model.

D. Cluster of Stations and Calculation of Demand

Demand: Our main aim is to find a way to predict the demand for bikes at a station. In other words: If the returns of bikes and rents are equaling each other out, then we do not have to think about this station anymore and can not re-distribute bikes from here to somewhere else.

But returns of the bikes must happen before their pickups, in order to take this into account, the 90-minutes return window is shifted by 30 minutes earlier towards the 90-minutes pickups window.
Demand equals returns minus pickups. Negative number: need of bikes at this station, positive number: have more than enough bikes at the station.

Cluster of Stations: The motivation here is that a person that wants to rent a bike does not really care if he walks to the station on the left or on the right. In reality he would open the app to figure out which station in his vicinity has bikes available and would then start walking towards it. Therefore we from now on will not think about individual stations, but about clusters of stations. This also has the advantage that we can group together a few stations and therefore have less noisy data. Instead of looking at 560 stations, we look at 170 clusters. The clustering was done through simply dividing the city into a 32x32-grid and joining the stations inside a cell to one cluster.

Combined Demand for Cluster instead of Station: As we saw in section II-B and figure 2, the activity on stations in the center is extremely variant throughout the day, especially during rushhours. Additionally the stations in those areas are pretty dense, so that a cluster contains many stations (biggest cluster in the center has 19 stations, outside of the city most of the clusters have 1-3 stations). If the demand of the stations is summed up to the demand of the cluster, the demand at higher and active stations will be more extreme than the demand in smaller clusters.
To take the size of the cluster out of the prediction, the next step is to calculate the relative demand, which is demand_per_cluster divided by cluster_size. This results in decimal numbers, e.g. a cluster of size 3 with demand 10 has relative demand 3.33333.
Since we want to predict this variable, decimal numbers are very impractical, but using the next bigger integer leads to a good approximation. ($\lfloor -3.33333 \rfloor \rightarrow -4$ and $\lceil 3.33333 \rceil \rightarrow 4$).
When this rounded number is now multiplied again by the cluster_size to get the overall demand in this cluster, we would get a prediction of 4*3=12 pickups, which is still relatively close to the actual demand, which was 10.
To think about it for extreme values: Assume we had 4 pickups in a cluster of 5 stations, or 1 pickup in a cluster of 5 stations, then we get a decimal number x between 0 and 1. $\lceil x \rceil = 1$ in any case. Multiplied by the number_of_stations_in_cluster we get one pickup per station. It is an over-estimation but makes sense in the real world application. The service provider prefers to over-supply bikes to ensure a smooth service.

For the year 2018 this leads us to the distribution of demands that is shown in figure 3. The logarithmic scale emphasizes that demand zero and ± 1 are extremely frequent in the dataset. Demand zero is almost a magnitude higher than the next most frequent value +1. The graph shows a exponential decline from zero towards extreme values to both sides (negative and positive range).

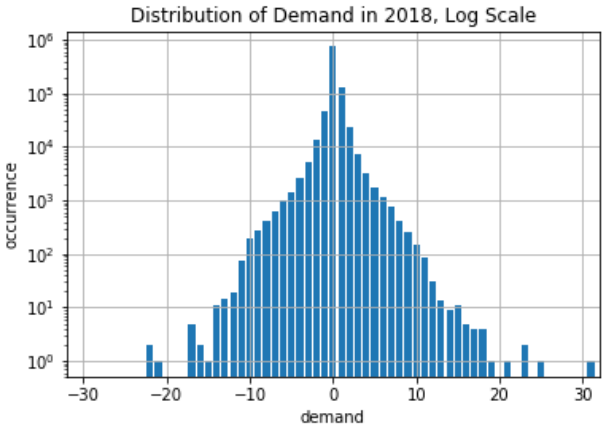

Fig. 3. Distribution of demands for year 2018. Note the logarithmic scale on the y-axis. We see an approximately exponential decline from zero towards extreme values to both sides (negative and positive range). But around 0 to ± 2 the decline is even steeper. We see that demand zero is almost a magnitude higher than the next values ± 1.

III. Algorithm

A. Decision Trees

Categorical nature of our data leads to the approach to build decision trees based on the past data to make predictions on the future.
Decision Trees are sensitive to overfitting, especially when they grow deep due to too many attributes and categories of the attributes. This issue can be bypassed by using the combined prediction of multiple smaller trees. Instead of growing a tree from all 13 attributes, we grow trees from the basic attributes (time and location) combined with one additional attribute each. And also some trees with two additional attributes. This results in 13 different trees.
Additionally, the trees are build according to different strategies:

Entropy: Entropy is a measure of the uncertainty of a random Variable, so in building the tree, the algorithm picks the attribute with the lowest entropy as next candidate to split the data.
Let X be a random variable with distribution $p(x)$ over values $\{x_1, ..., x_n\}$, the entropy of X is defined as:

$$H(X) = -\sum_x p(x)\, log_b\, p(x)$$

when $b = 2$, then the entropy is measured in bits.

Information Gain: Pick the attribute with the highest Information Gain as next candidate to split the data.
Let X be a random variables with distributions $p(x)$ and $p(y)$ over values $\{x_1, ..., x_n\}$ and $\{y_1, ..., y_m\}$. $H(X)$ be the

entropy of X. The conditional entropy of X given that Y is defined as:

$$H(X|Y) = \sum_y p(y)\, H(x|Y=y) = -\sum_{x,y} p(x,y)\, log_b\, p(x|y)$$

the mutual information between X and Y (measure of amount of information that variables X and Y hold about each other):

$$I(X,Y) = H(X) - H(X|Y)$$

The motivation for Information Gain is the following: Given a set of datapoints, the quality of each attribute is minus the average of the entropies of the nodes that are produced by the split of this attribute. In other words: it represents the information that is relevant to classification that is gained by the division.
Define the following:

- $Attr$: the set of attributes (features)
- Ex: set of training examples
- $val(x,a)$: the value of attribute a in example x
- $Goal\, \epsilon\, Attr$: the attribute that we want to predict
- for attribute a and value $v\, \epsilon\, val(a)$: $Ex_{a,v} = \{x\, \epsilon\, Ex | val(x,a) = v\}$

The Information Gain for an attribute a is defined by:

$$IG_{Ex}(Goal, a) = H_{Ex}(Goal) - H_{Ex}(Goal|a)$$

where $H_{Ex}(Goal|a) = \sum_{v\, \epsilon val(a)} \frac{|Ex_{a,v}|}{|Ex|}\, H_{Ex_{a,v}}(Goal)$.
The drawback of Information Gain is, that it prefers attributes with many possible values, because the more elements the sum has, the bigger it is likely to be. Even though many splits in one layer increase the purity of the tree quickly, it does not mean that those were good splits. Small trees are actually preferred over big ones, because big trees are likely to overfit.

Gain Ratio: To solve the problem of Information Gain that prefers attributes with many values, Gain Ratio is the normalization of Information Gain. Pick the attribute with the highest Gain Ratio as next candidate to split the data.
The Gain Ratio is then defined as:

$$IGR_{Ex}(Goal, a) = \frac{IG_{Ex}(Goal, a}{IV_{Ex}(a)}$$

Where $IV_{Ex}(a) = -\sum_{v\, \epsilon val(a)} \frac{|Ex_{a,v}|}{|Ex|}\, log\, \frac{|Ex_{a,v}|}{|Ex|}$ is the Intrinsic Value of the attribute a, which represents the potential information generated by dividing Ex in to subsets according to a's value.
Information Gain represents the proportion of the information that is generated by the split that is useful (i.e. that appears to help the classification)

Using those three kind of trees and our chosen variation of attributes, we get a total of 39 trees.

One very important point about the creation of the trees needs to be mentioned: In case that multiple datapoints lead to the same leaf, the most frequent value gets assigned to the leaf. For example if 8 datapoints lead to one leaf that have the demand (0,1,1,1,1,2,2,8) we assign demand 1 to the leaf, because it was the most frequent value. We also could have decided to choose the median (or the mean), but it occurred to us that taking the most frequent value models the reality the best, since those are the events that actually happened the most. This has the problem that extreme but rare values will not be modeled by the tree and will not be predicted. The comparison between the distribution of the actual demand (figure 3) to the distribution of the predicted demand (figure 6) shows exactly this issue.

We used random 50% of the data from January 2017 to June 2019 to build the trees.

The trees that we built are visualized with bokeh, instructions on how to display them are found here [5]. Unfortunately it is only possible to show trees that are small enough, since more complex trees need too much time to be visualized with bokeh, which is not a pleasant experience.

B. Combination of Trees

Instead of combining the prediction of the trees uniformly, we decided to expand the setup by a Neural Network, that takes as input the predictions of the 39 different trees and combines them to a single prediction.
This procedure has the advantage that more expressive trees are valued differently than less expressive trees. Also we expect the Neural Net to find a more complex connection between the tree predictions. It is possible that some combinations of trees work good together for low values and others work good together for high values. This is some non-linear function that a Neural Network should be able to figure out.

Architecture of Network:

- **Input Layer**: 1x39, number of predictions of the trees for one datapoint
- **First Layer**: 1x100 neurons, activation function: tanh
- **Second Layer**: 1x60 neurons, activation function: tanh
- **Output**: 1x58, where every position represents a demand-class and the value its likelihood. The predicted demand is therefore the demand-class with the highest likelihood. activation function: softmax where its range is defined by the maximum demand that occurred in the training set.
- **Optimization Function**: Adam
- **Epochs trained**: 100, **Batch Size**: 60

The hyper-parameters were tuned through gridsearch on batch size, epochs, optimization function, neurons of first layer, neurons of second layer. As measures of the performance we used accuracy and sparse categorical cross entropy as a loss function, which fits to our purpose since each predicted demand can be seen as a class on his own.

We use the remaining 50% of the data from January 2017 to June 2019 as test data (The part of the data that was not used to build the trees.) We predict the demand with the trees and then use 80% of the test data to train the Neural Network and 20% to test it.
To verify the performance of the net, we performed a 10-fold crossvalidation that gave a mean accuracy of 0.6896 and variance 0.0022. The low variance shows that the Neural Net is not overfitting.

Evaluation of the Accuracy

The accuracy of about 70% does not seem too good at first, but maybe it is good enough for our needs. A prediction is only counted as accurate, if it is exactly equal to the true value, but for us it might be enough to be close to the true value. Therefore it makes sense to look at the error size, that says by how much the prediction is off.
In figure 4 we see that almost 90% of the errors are only off by one or two. Which means that if we always provide two more bikes than the prediction suggests, the stations will have enough bikes in $69\% + (31\% \cdot 90\%) = 96.9\%$ of the cases. If one additional bike is provided, then in $69\% + (31\% \cdot 70\%) = 90.7\%$ of the cases. (probability of accurate prediction + (probability of inaccurate prediction * part of predictions that become accurate if additional bike is provided)

Fig. 4. Comparison of distribution of error size between uniform combination of trees and combination with Neural Net. Of the errors that the Neural Net makes, the error size is most of the times pretty low, whereas in the uniformly combination of threes, the error size is much higher.

Figure 4 also justifies the choice of a Neural Network to combine the Decision Trees over a uniform combination of them. In case of the Neural Network, already 70% of the incorrectly predicted demands are only off by 1, another 20% are off by two, decreasing exponentially. The error sizes of the uniform combination of trees only declines slowly and almost linearly. This means that even if the Neural Network and the uniform combination of the Decision Trees had the same accuracy, then the Neural Net is not as much incorrect as the uniform combination of Trees is. But additionally the accuracy of the uniform combination of Trees is even less.

IV. Prediction

We now have the tools to predict the demand for future days. We created a dataset for a workday and a weekend day in September 2019, for good and bad weather each, therefore we used the weather data of some day in September in 2018.

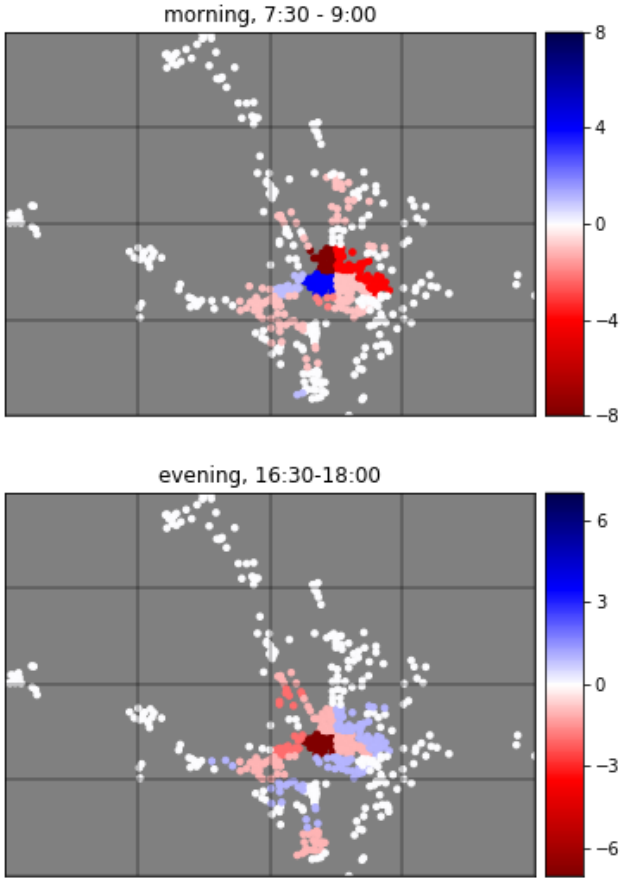

Fig. 5. Predicted demand of bikes per station during morning and evening on a working day in September 2019 with good weather. Negative numbers (red) mean that in the next 90 minutes this station will need additional bikes, positive numbers (blue) indicate an existence of unneeded bikes. The predicted demand around the rushhours is similar to the historical demand in figure 2.

The whole results are visualized via bokeh. An interactive map shows how the demand changes over time on weekday or workday, during good and bad weather [**?**], but figure 5 shows one input set: the result of the predicted demand on a workday during the morning and evening rushhour. The demand behaves similar to the historical demand (see figure 2), we again see more bikes arriving in the center during the morning and bikes going out of the city in the evening. Be aware that the only location information is, whether a bike belongs to L1 and L2 area or not. This is why the colored areas are bigger than in the plot of the historical data, where we looked at the clusters of stations. (The L1 and L2 attributes are explained at the bottom of section II-B. Imagine a grid of 4x4 (L1) or 16x16 (L2) drawn over the map, all

stations inside a cell belong to this L1/L2 area.)

The predicted demand differs from the historical demand in the way that that the minimal and maximal demand values are smaller than the maximal values in the historical demand. In this specific case it could be that during the displayed days was not that much activity in general, but we also see this over all the predictions.
Figure 6 shows the distribution of the predicted demand. If we compare it to the distribution of the historical demand in figure 3, we notice that the historical demand ranges until around ± 20, whereas the predicted demand only ranges until around ± 12. (In the distribution of the historical demand we only see data from 2018, in the distribution of the predicted demand we see random 20% of the data from 2017-June 2019. Even though the graphs do not show the exact same data, it can assumed to be similar.)
The reason why we do not manage to predict high values is the design of our trees. As explained at the bottom of section III-A, if during the creation of the trees the case occurs that multiple datapoints lead to the leaf that have different demands, then the most frequent value will be assigned to the leaf. Therefore extreme values, that are generally not frequent, won't be able to appear in the tree unless they are the only values that arrive at a leaf.
But the prediction still represents the historical data quite nicely, since the distribution of the demand stays pretty much the same. It has a similar form but does not range as wide as before.

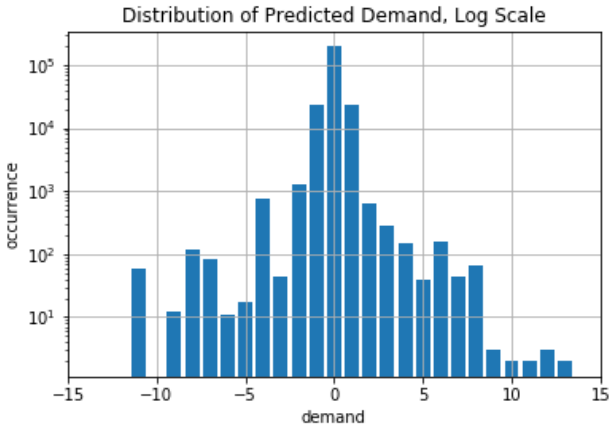

Fig. 6. Distribution of predicted demand, compare to distribution of demand of historical data in figure 3. The distribution again declines exponentially from zero to both sides to higher numbers. The range of predicted demand is remarkably smaller than of the historical demand. But the proportions are similar, which means that the prediction gives a similar distribution of results to the data that we were trying to model.

V. Problems and Outlook

Try to model human behavior.
A huge problem of this kind of project, and what should always be kept in mind in such a model is, that it is extremely difficult to model human behavior and the accuracy of the predictions needs to be seen in this context. We only used information about time, location, type of day and weather to predict the demand on stations that is only dependent on human behavior.
What if a road was blocked or there were some news that people should not go into a certain area? What if there was a huge one-time-event in some area of the town and many people wanted to come by bike. What if there is a general moodyness because of a local incident in the city and people do not feel energetic enough to go by bike?
There are multiple factors that could have influence on bike rentals, some of them could be modeled with additional attributes (like construction sides), others are probably impossible to feed into the model.

Elimination of extreme values through decision trees.
This phenomenon is described at the end of section III-A. It would be worth it to try different rules for deciding for the most representable demand in such a case. Other opportunities are for example to take the mean, or better median, or to filter out some events first. Since this project focuses on the actual algorithm of the decision trees, we decided not to focus on this choice too much here.

Choice of areas pretty arbitrary.
The choice of areas as explained in section II-B was based on the decision making of 'how can we decide which areas are different, if we do not actually know anything about the city'. We just cut the city into different pieces, from big to small. But it would make much more sense to find out which areas are actually similar to each other like business district, area with mainly flats of young workers, family homes...

More relevant attributes.
It would also be worth it to expand the data by more relevant attributes. Our model so far uses time, location, weather and type of weekday. But also information like Census data is likely to help the prediction, or police reports like blocked roads, or to know which stations are close to famous commuting spots (work, park, nice road along the river, ...)

References

[1] The Bike-Share Oversupply in China: Huge Piles of Abandoned and Broken Bicycles.: https://www.theatlantic.com/photo/2018/03/bike-share-oversupply-in-china-huge-piles-of-abandoned-and-broken-bicycles/556268/
[2] https://www.capitalbikeshare.com/
[3] Historical Trip Data of Bikerides in Washington D.C.: https://s3.amazonaws.com/capitalbikeshare-data/index.html
[4] Information about Stations, used for Location of Stations: https://opendata.dc.gov/datasets/capital-bike-share-locations?geometry=-164.18%2C-29.382%2C164.18%2C29.382
[5] GitHub repository of the project. How to run visualizations is explained in the README: https://github.com/kfir1g/AI-Project