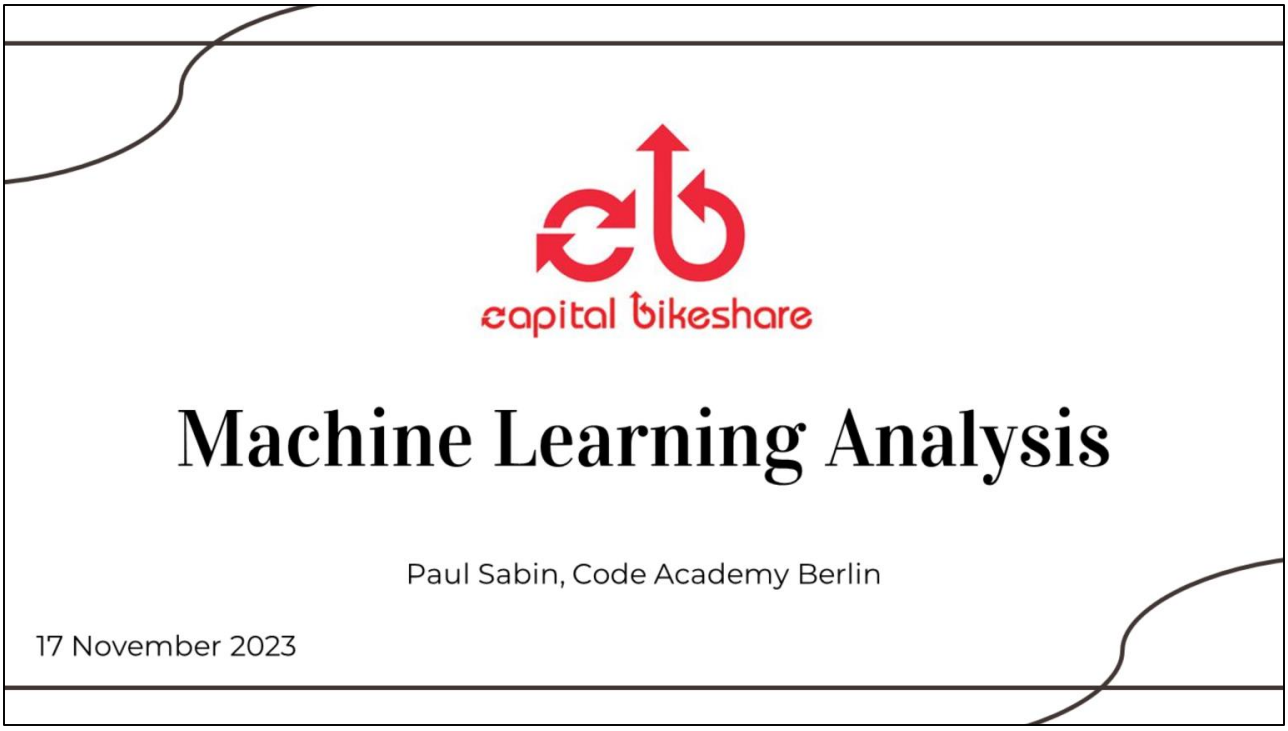


Full slide deck + brief speaker notes below

Contents

01
Recap

About Capital Bikeshare (CaBi)

Balance

System/stations

Location

Affects ratio of bikes taken/returned

Rebalancing

Motivate & Bike Angels

4

Balance of whole system, balance of individual stations

02
Machine Learning Goal

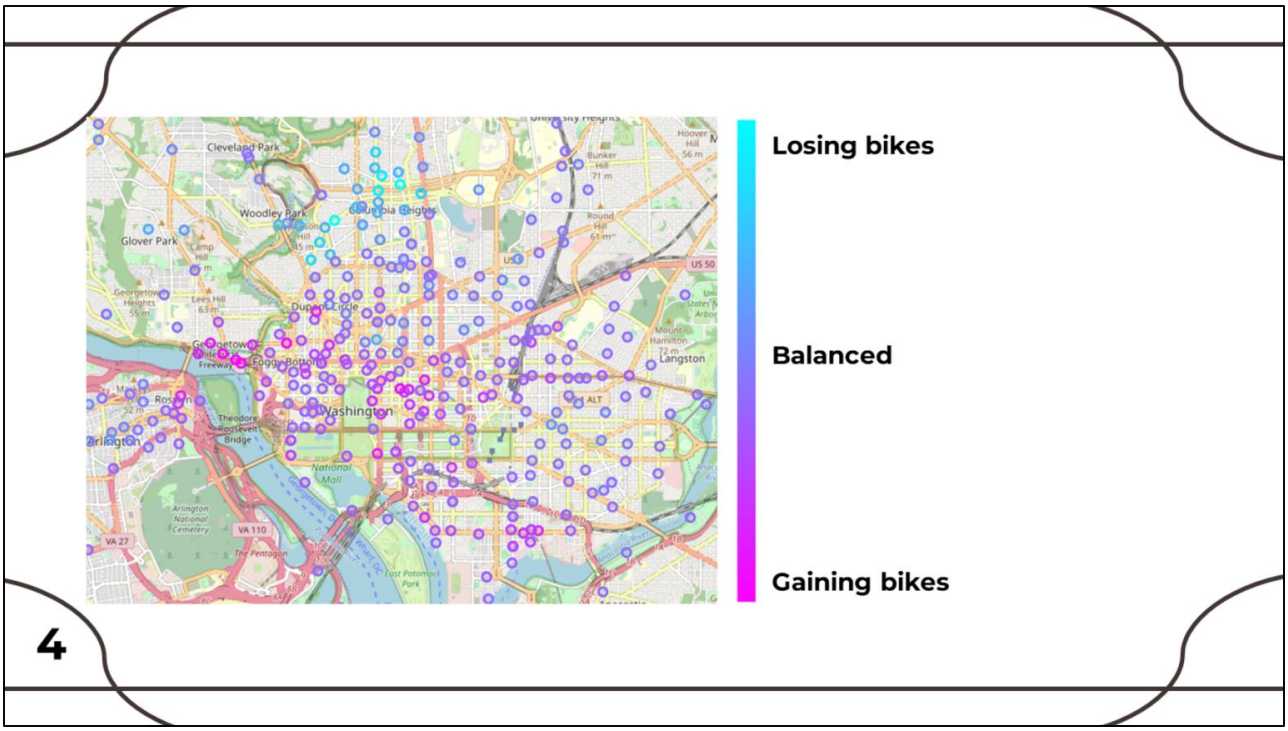
Predict (im)balance of a proposed new station

A new station will be opened at a certain location. Based on coordinates, can we predict:

- Ratio of bikes taken/returned
- Net total bikes taken/returned

5

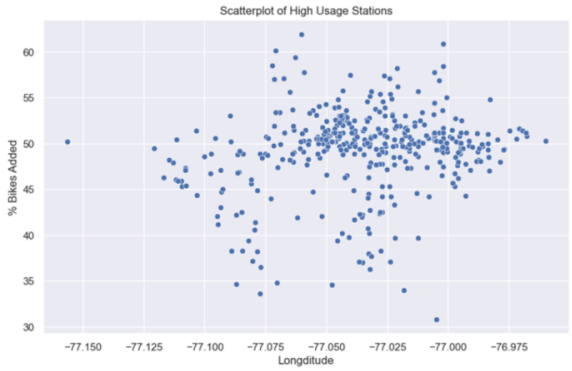
[Slide]
The second figure is a little harder to predict, because it also involves the total amount of station usage

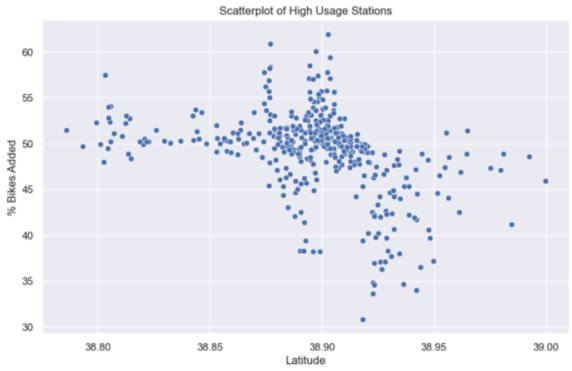


Here's a screenshot of a map similar to the one I created for my EDA. Each circle represents a station, and the colour is mapped to the ratio of bikes added, so light blue circles are stations that lose a lot of bikes and are in danger of becoming empty, darker blue/purple circles are well balanced stations, which are the majority, and bright pink circles are stations that gain a lot of bikes and therefore often get full up and "dock-block" riders.
The ML models can't see the map or the features of Washington DC, but can they nevertheless extract useful information from the latitude and longitude data?

03
Setting up ML Models

Looking for correlation



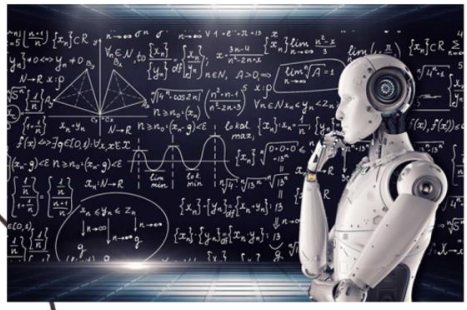


6

So first of all I just wanted to see whether any correlation would be visible to the naked eye by plotting on these two graphs longitude versus ratio of bikes added, and the same with latitude. As you can see, there does appear to be some pattern here, so I decided it was worth exploring this topic further. However, it would be difficult to express the patterns you see here as a line, a curve or even a number of curves. So I was unsure how far I would get with linear regression or polynomial regression methods.

Steps…

1) Trained 10 models with lat/long as input variables
2) Attempted to predict ratio of bikes added, net total
3) Trimmed data set to omit least used stations
4) Focused on most promising models
5) Performed hyperparameter tuning



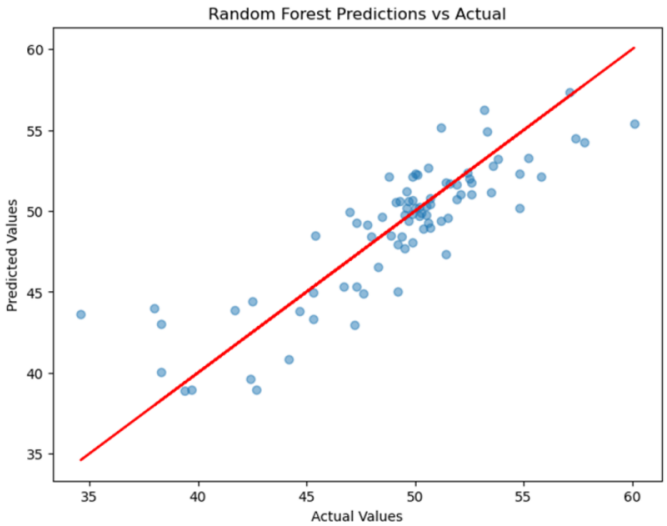
04

ML Results

Predicting ratio of bikes added

Random Forest:

Mean Absolute Error: 1.83
Mean Squared Error: 5.78
Root Mean Squared Error: 2.40
R-squared: 0.73



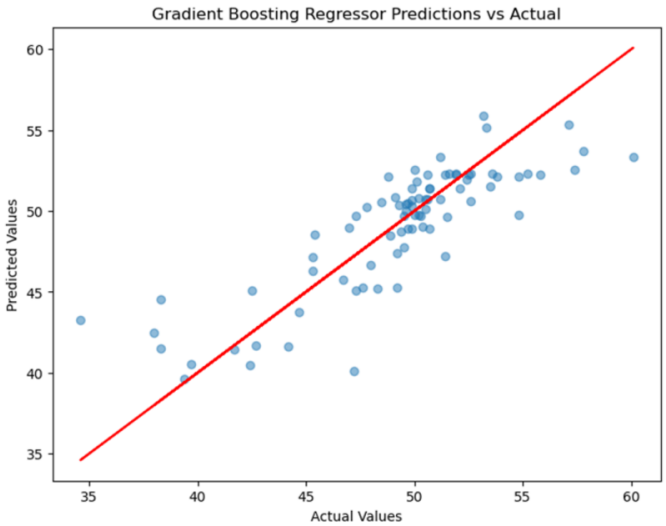
6

Ratio of bikes added is a measure of how balanced a specific station is. 50/50 = perfectly balanced. <50 = losing bikes. >50 gaining bikes.
Each dot represents a station. Many stations are clustered around the 50 mark. Lowest ratio around 35, highest around 60 – very problematic stations that require a lot of rebalancing.
The model correctly identifies stations that are balanced, those that lose bikes and those that gain bikes almost every time. However, it is not very accurate when it comes to the degree of imbalance. For stations that are extremely unbalanced, the model often gives a more conservative estimate.

Predicting ratio of bikes added

Gradient Boosting Regressor:

Mean Absolute Error: 1.89
Mean Squared Error: 6.44
Root Mean Squared Error: 2.54
R-squared: 0.70



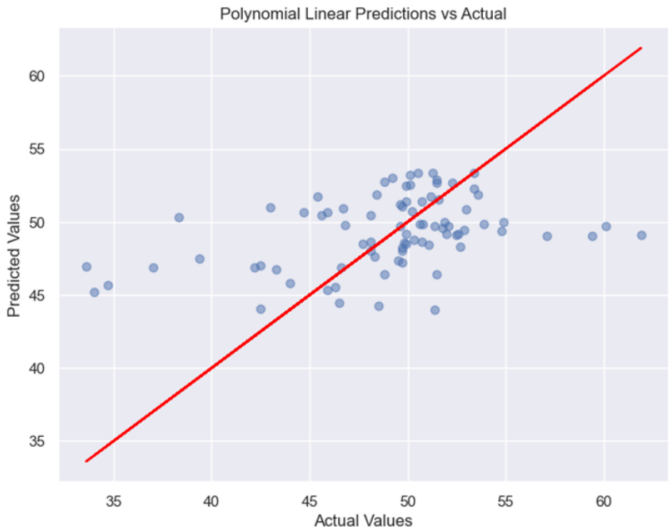
6

Gradient boosting regressor performs similarly well to the random forest. You can see that the dots reveal a trend that aligns, to some extent, with the diagonal line of perfect predictions. Again, the more extreme actual values tend to have a larger error. My intuition here is that both these models are doing a good job of identifying patches, within the coordinates data, where stations are very likely to gain or lose bikes. However, the prediction is then based on a mean value for that geographical area. That's why we see a horizontal arrangement of the blue dots on this plot: the models are making very similar guesses for groups of stations. The stations with more extreme actual ratios are likely influenced by additional factors that are not being taken into account at all by these models, which have been trained exclusively on positional data.

Predicting ratio of bikes added

Polynomial Linear Regression:

Mean Absolute Error: 3.48
Mean Squared Error: 22.51
Root Mean Squared Error: 4.74
R-squared: 0.14



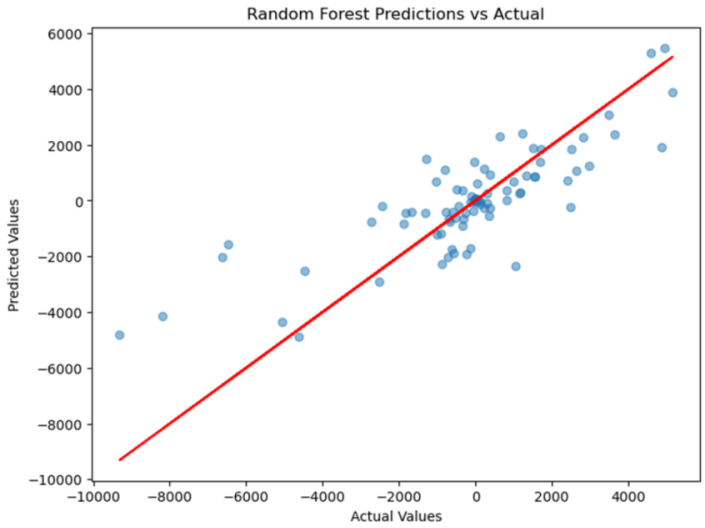
6

Let me just quickly show you, by contrast, a model that was far less effective – polynomial linear regression. It does correctly identify most of the stations that are losing bikes, but the predictions are way to close to 50% and you can see that all the predictions are in this horizontal arrangement, meaning that it's predicting very similar values for large groups of stations.

Predicting net total bikes taken/returned

Random Forest:

Mean Absolute Error: 1032
Mean Squared Error: 2.2m
Root Mean Squared Error: 1495
R-squared: 0.66



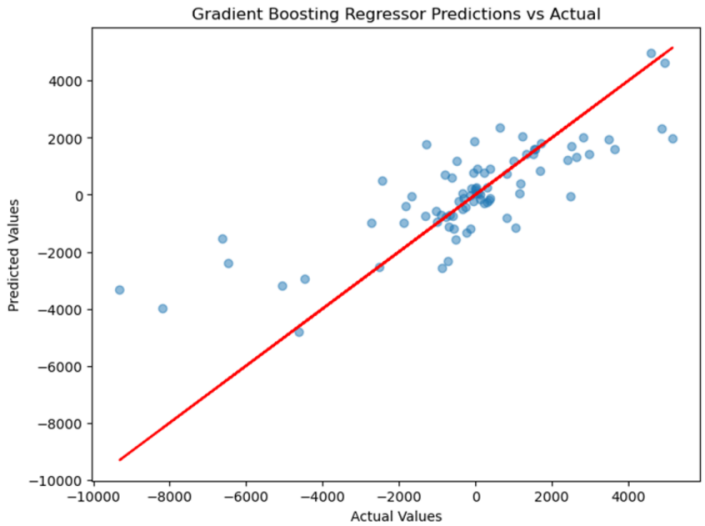
6

Now we look at the measure that's a little more difficult to predict, namely the net total of bikes taken and returned. Why is the scale so large? Because it covers the whole 2.5 year period of the data set.
We have an r-squared value of 0.66, and we could express this in another way: roughly two thirds of the variation we observe in the net total of bikes taken and returned can be explained by the independent variables in the model, namely the latitude and longitude of the station.

Predicting net total bikes taken/returned

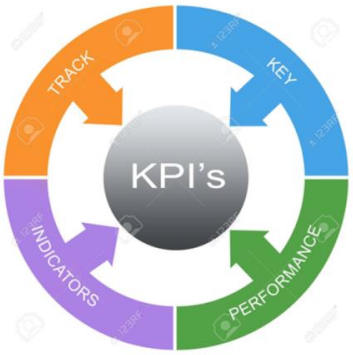
Gradient Boosting Regressor:

Mean Absolute Error: 1044
Mean Squared Error: 2.5m
Root Mean Squared Error: 1571
R-squared: 0.62



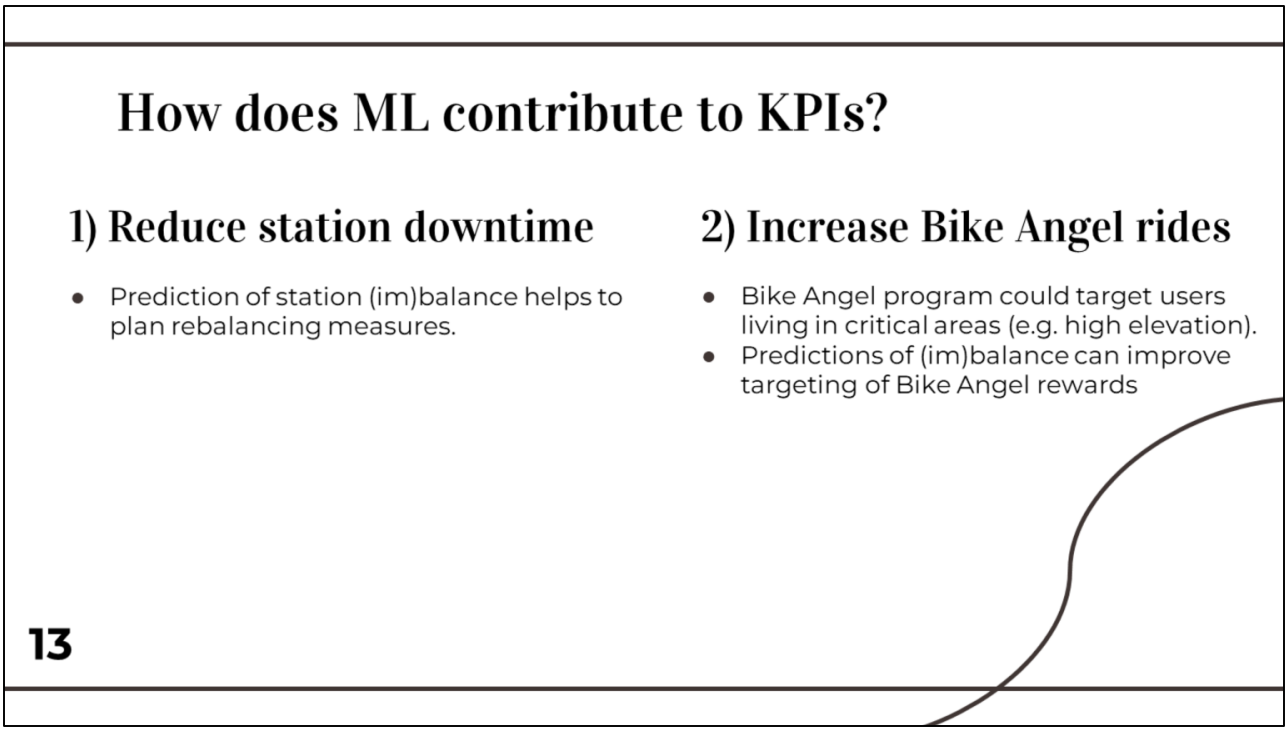
6

Quite a similar result. But, when I changed the random state on the train/test split, the gradient boosting regressor performed less consistently. This suggests that gradient boosting regressor is overfitting to the specific data it's trained on, whereas the random forest model maintains higher scores across different splits.



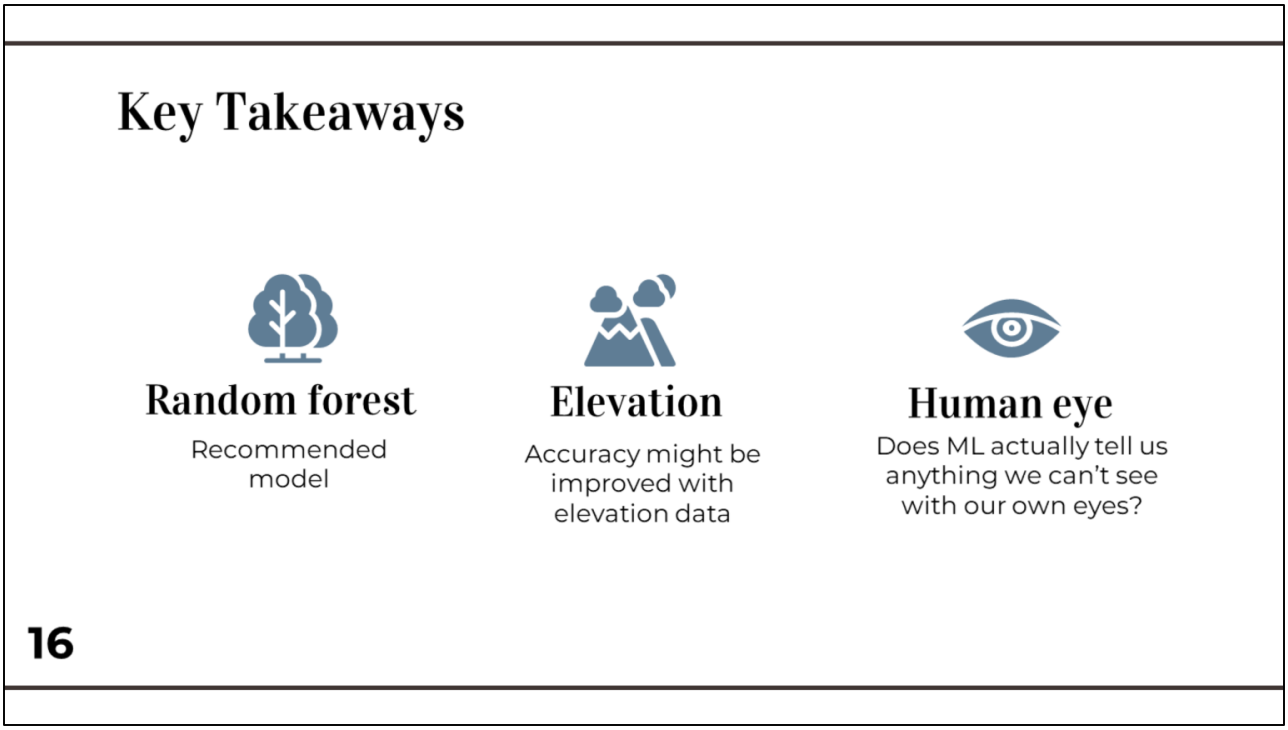
05

KPIs



Station downtime is defined as the length of time per month that a station is *completely full or empty*.
If we can already predict whether a new station or an existing station is going to remain in balance due to normal user activity or consistently lose or gain bikes over time, this helps Capital Bikeshare and the operator Motivate to plan its rebalancing measures more effectively. As you will recall, these include bike coralls for receiving bikes at busy stations and the transport of bikes by truck from full to empty stations.
Bike Angels example: If we know that a certain station tends to lose bikes throughout the day, we can incentivise Bike Angels to replenish that station with bikes as soon as it has several open docks.

06
Key Takeaways



Random forest performed most consistently when predicting the ratio and the net total.
We used latitude and longitude as data to feed the model, but perhaps we could improve accuracy if we also knew the elevation above sea level of each station.
The kind of trends we're talking about – for instance, that people are more likely to ride downhill – are already quite easily visible by marking the most imbalanced stations on a map of the city. I would suggest that right now the ML is of limited use, except for confirming what we can already observe from the data. However, if we add other independent variables to the training data in the future, perhaps the ML approach will reveal something we could not spot with our own eyes.

Thanks

Any more questions?

paul@sabinwords.com