Phone Sentiment Analysis with Web Crawl Data

Tuomo Kareoja

IOT Analytics

November 28, 2019

Agenda

Background

- Mission: Locate device by its WiFi fingerprint
- Area to cover:
  - Three buildings of Universitat Jaume I with 4 or more floors and almost 110.000 m^2
  - More than 20 different users and 25 Android devices, with widely varying signal strengths
  - Area covered by 510 different WiFi access-points

Data Processing

- On top of WiFi signal strength also the phone model and OS very provided. These were not used in the models and dropped
- Missing signal for Wifi access-points was recoded from 100 to -110 so that it was smaller than the weakest actual signal in the dataset (-105)
- No scaling was done as all the models used where tree based
- outlier dropping was tried, but this led to worse performance, so all observations where kept in the analysis

Model Types Used

- Different variations of KNN models were tried as these usually perform well in in these locationing tasks and can predict multiple related targets (latitude, longitude, building, floor) all at once
  - K and radius model uses 3 closest observations by similarity of WiFi signals, but beyond most nearest neighbor it only considers observations which are within certain radius limit of this distance
  - KNN grouping model uses 2 closest observations by similarity of WiFi signals, but before training the models the WiFi signals strengths were grouped averaged for by room
- Second option was to create multiple models to predict different metrics with interconnected CatBoost models that would capture the connections between target values
  1. Separate model was created to predict each outcome from WiFi signals
  2. The predictions from previous models were added beside the Wifi signals and given to second layer of models that had the same goal as the first ones
  3. Lastly the just predictions from the second layer were given to a last layer of 4 models that made the final prediction

Model Evaluation

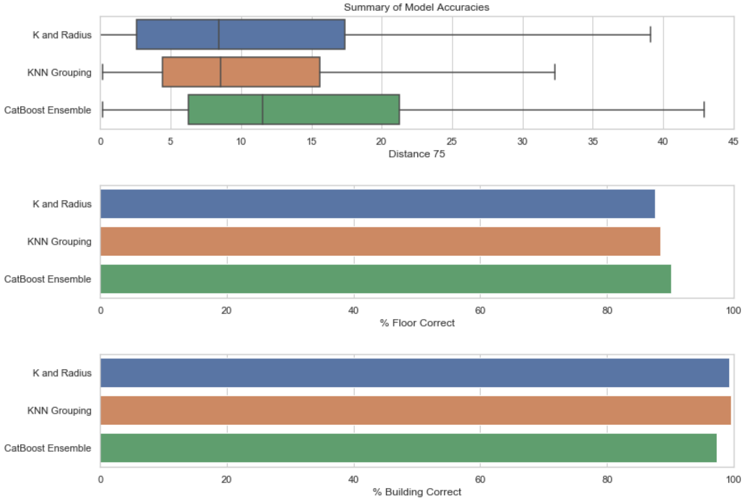
Models were evaluated by a single scoring metrics named Distance 75 that combines all the 4 target values. This metric calculates the manhattan distance between prediction and actual values in 3D spaces and adds a additional penalty for getting the building wrong:

Distance75 =

$$
\begin{aligned}
&\frac{1}{n} * \sum_{n=1}^{n} (|longitude_{predicted} - longitude_{actual}| \\
&+|latitude_{predicted} - latitude_{actual}| \\
&+4 * |floor\ number_{predicted} - floor\ number_{actual}| \\
&+50 * |building\ number_{predicted} - building\ number_{actual}|)
\end{aligned}
\tag{1}
$$

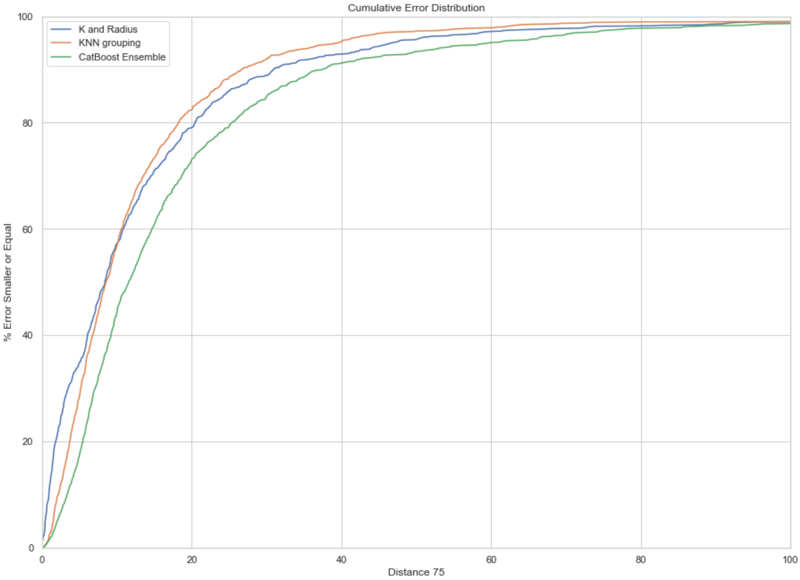
Overall Performance

- KNN models perform clearly much better than CatBoost
- Building is easy to predict, but floor is not



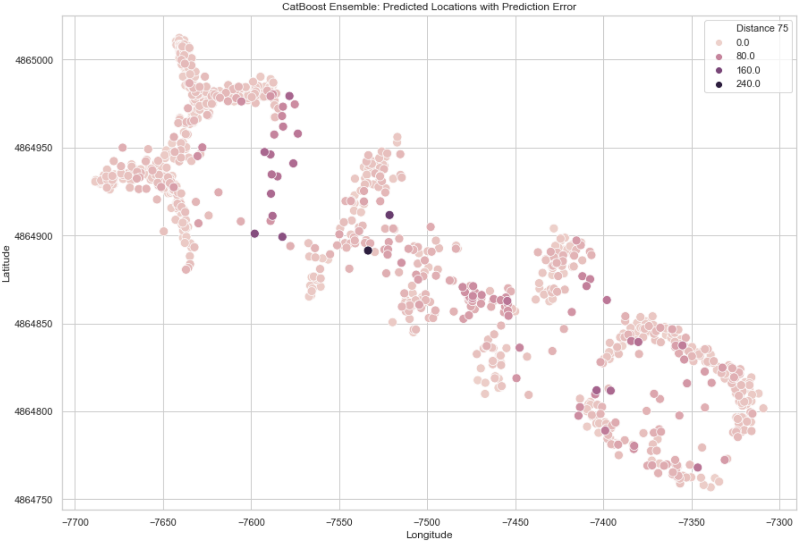
Error distribution

- K and radius model has a fatter tail than KNN grouping in errors



KNN models vs. CatBoost

- CatBoost makes predictions that are outside the buildings



KNN models vs. CatBoost

- KNN keeps to already observed values and makes no stupid errors

Conclusions

1. Hard to model as the model has to be able to predict multiple related targets (e.g. longitude and latitude)
2. Simple KNN model with some adjustment beats more complex models
3. Signal processing does not seem to help (maybe should be done for each OS separately)

The End

Questions?