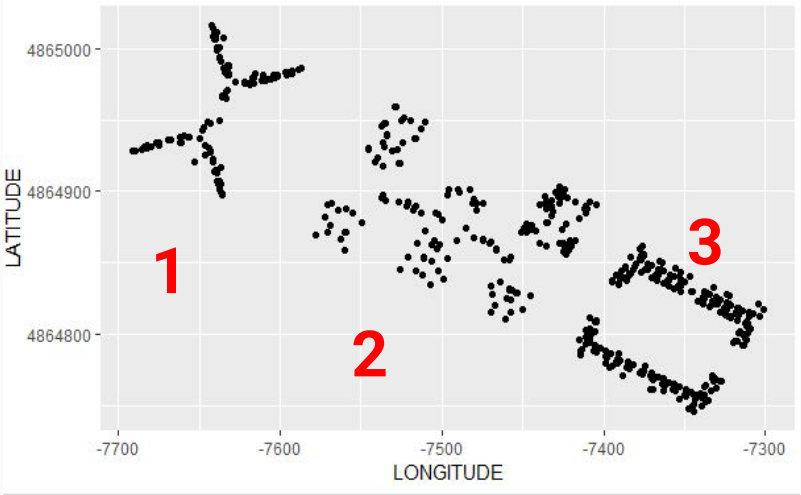
Maximilià Goldston Marí

WIFI ALLOCATION: KNN, bias and overfitting

Context

Our measurements

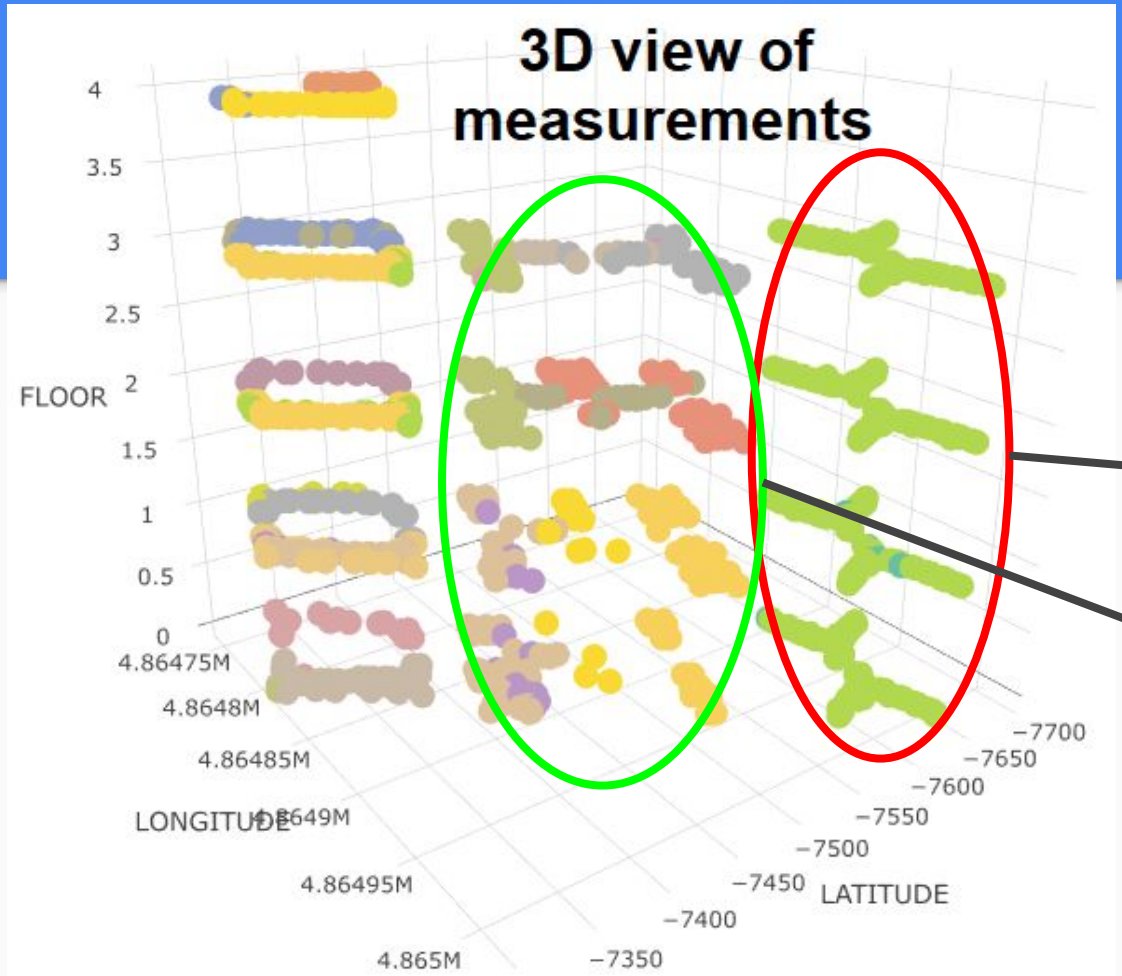
View from the top



520 Wireless Access Points (WAPs) in 3 buildings.

Through our phones, we measure the received signal strength indicator (RSSI) received at a specific point in space of each WAP.

Use these measures to build a system to predict position in the buildings



905 spaces
18 users
16 phones
2 types of measurement: Center of room / door of the room

Building 0: 2 different users measuring the same spaces

Buildings 1 and 2: 3 to 4 users per floor

Our goal

Our objectives: Use ML techniques that will help us predict the Building, the Floor, the Latitude and the Longitude of a new user.

Our scope: Build a model that is able to **generalize** for new and different measurements (different phones, user heights…).

How? **Preventing overfitting** and **mitigating bias** in our original data.

Before building our models

Normalizing our data

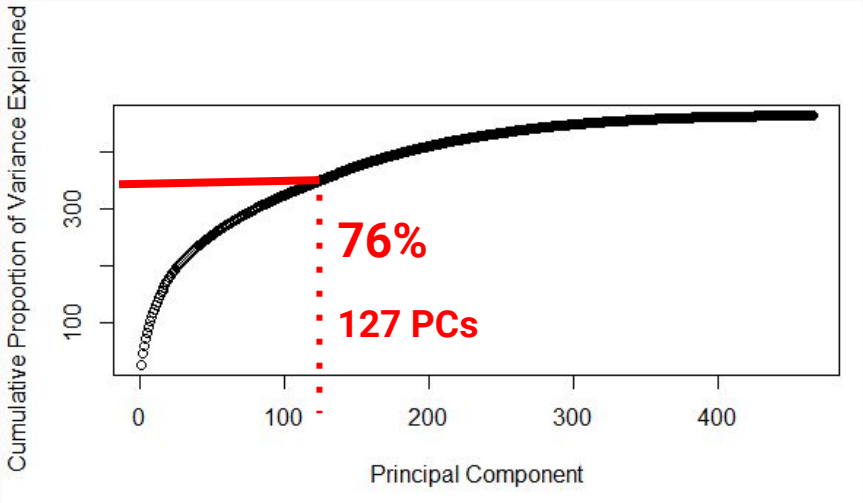
2 possible approaches

We're interested in generalizing for new measurements.

Normalizing WAPs (columns):
-Causes features to be equally weighted. Give same importance to every WAP.

Normalizing measurements (rows):
-Causes measurements to come closer. Considering there's proportionality in RSSI between phones, this mitigates the effect of different Users and Phones making the measurements.

Preprocessing
Principal Component Analysis (PCA)



For preprocessing we decide to get the Principal Components that have Eigenvalues >1.
An *eigenvalue* > 1 indicates that PCs account for more variance than accounted by one of the original variables in standardized data.

These components explain 76% of the variance:

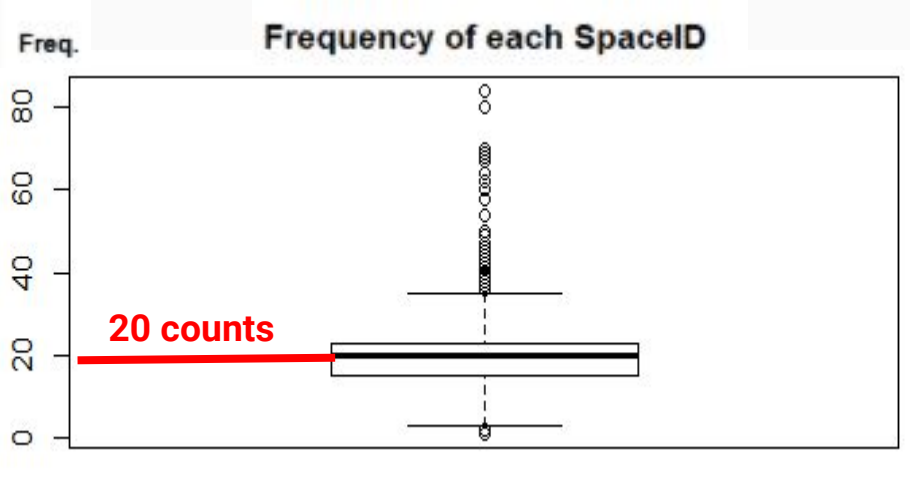
127 PCs


Predicting the
building of the user

Bias in building

We get the same number of measurements for each building in order to prevent bias towards building 3.

Number of neighbors



20 measurements belong to a single space ID.

When building a KNN model, on average, the 20 first neighbors will belong to the same spaceID.

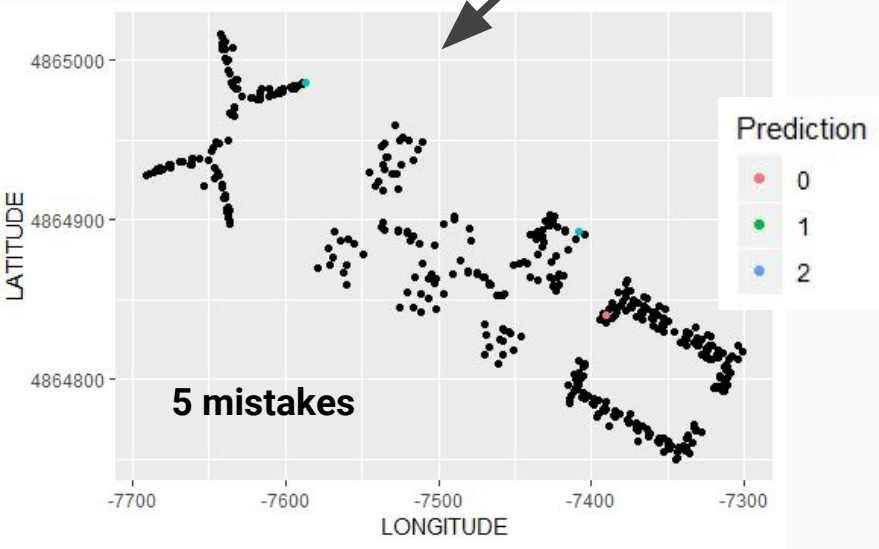
For building, we want that on avg. each SpaceID is influenced by 2 more.

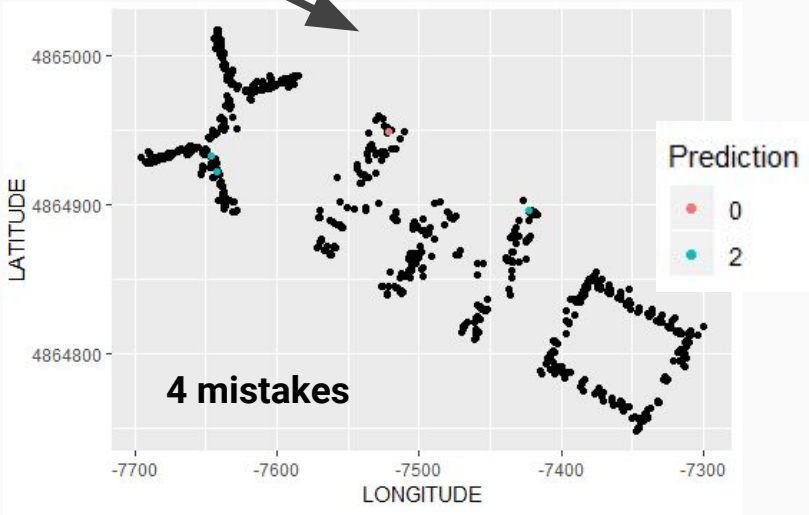
We choose 60 as the number of neighbors.

KNN MODEL:
-CV split: 10, repetition: 2
-Neighbors: 60

Test set (15%):
Accuracy: 0.9982
Kappa: 0.9966

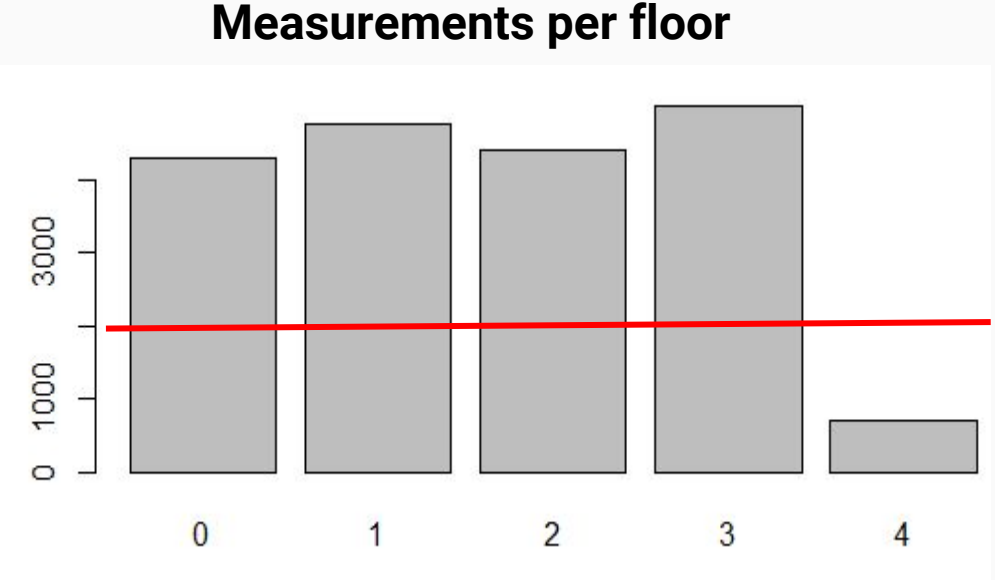
Validation set
Accuracy: 0.9964
Kappa: 0.9943





Predicting the floor of the user

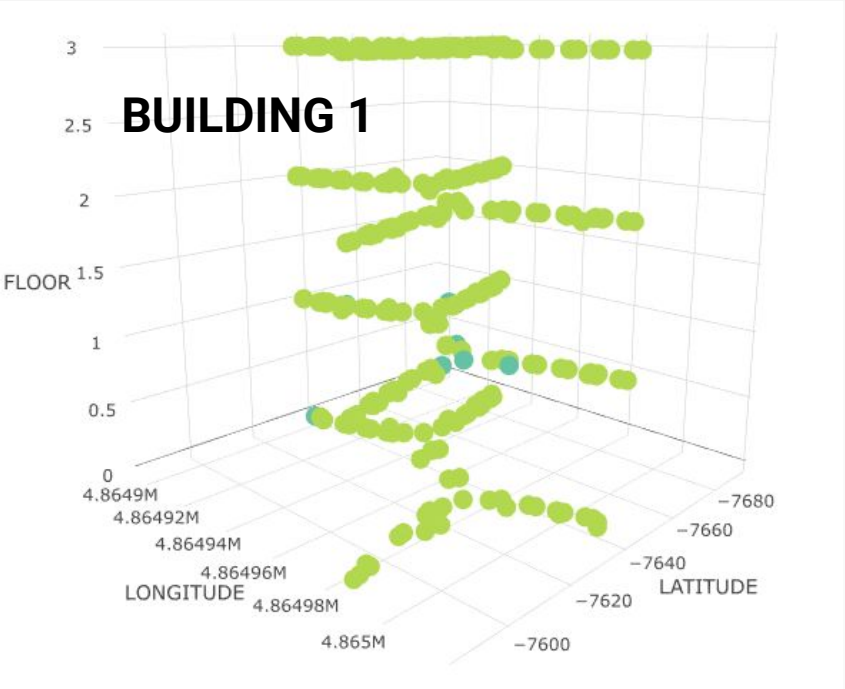
Bias in floor

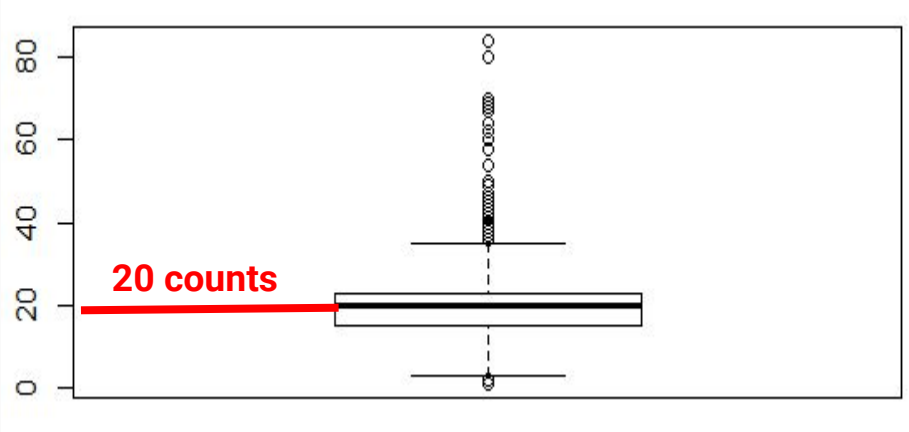


When making this we get a reduction of 5% in the accuracy of our model in the test set.

SpaceID Bias

We choose 60 neighbors.
3 different Space IDs.
Improves our generalization.
Probably 2 out of the 3 neighbors are in the same floor.



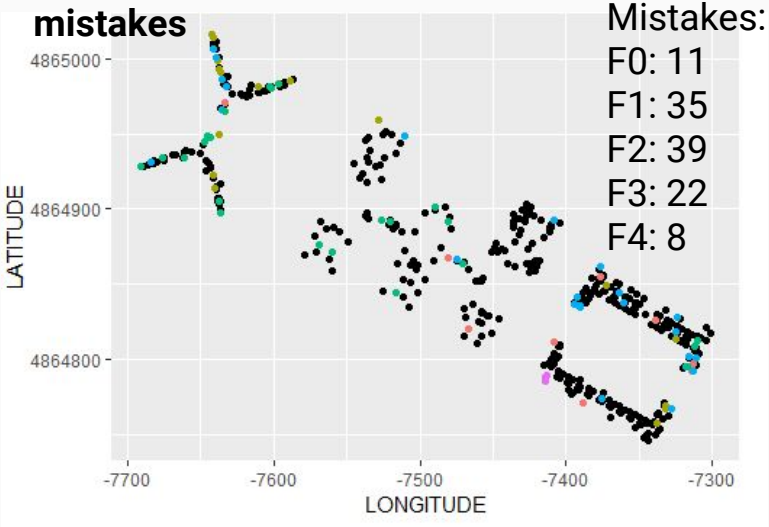


KNN MODEL:
-CV split: 10, repetition: 2
-Neighbors: 60

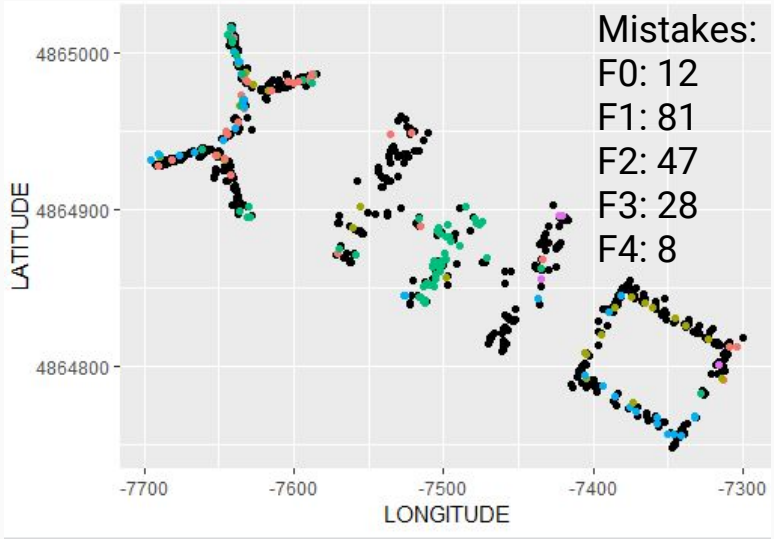
Test set (15%):
Accuracy: 0.960
Kappa: 0.948

Validation set
Accuracy: 0.842
Kappa: 0.781

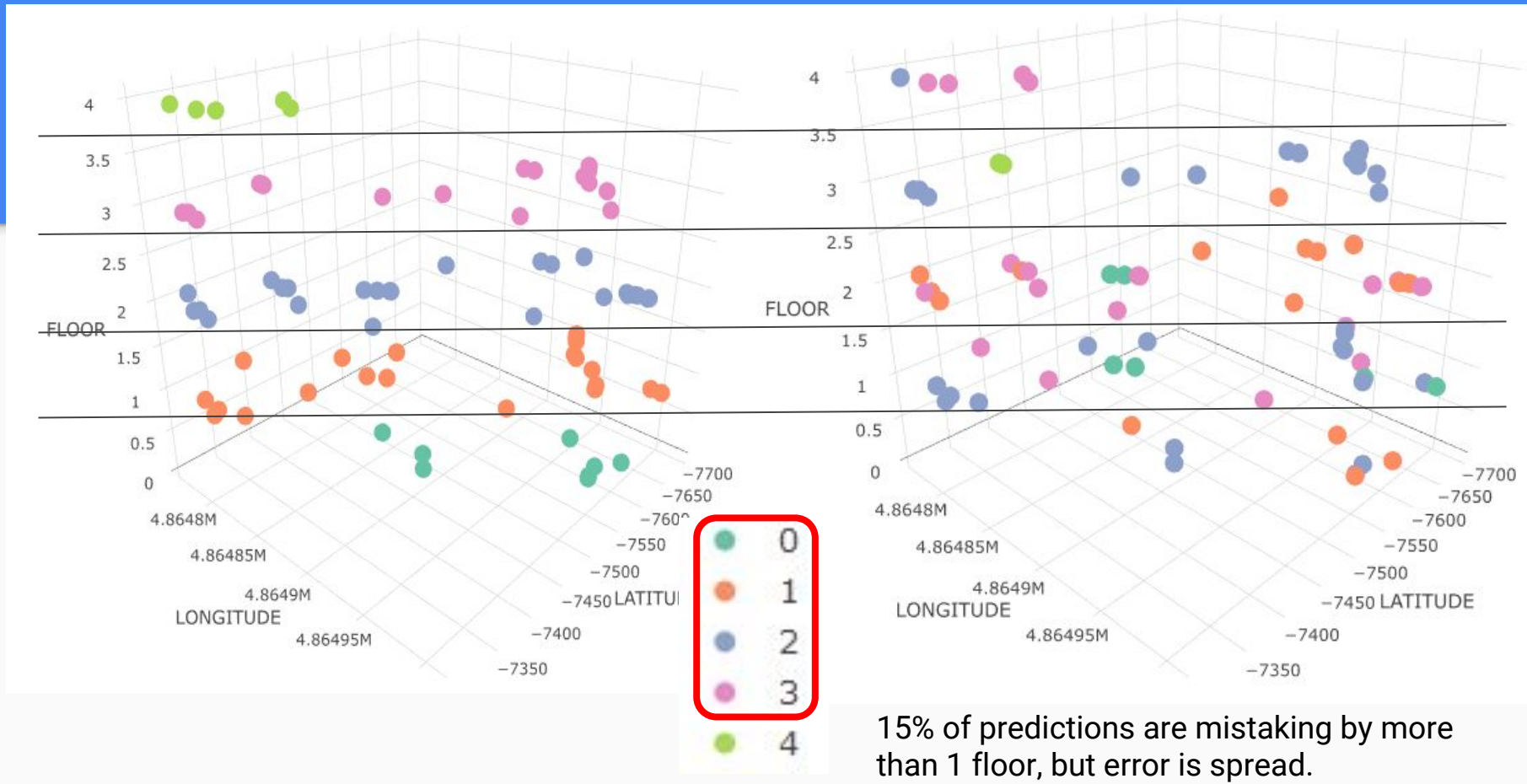
1 neighbor: 0.98, 20 mistakes



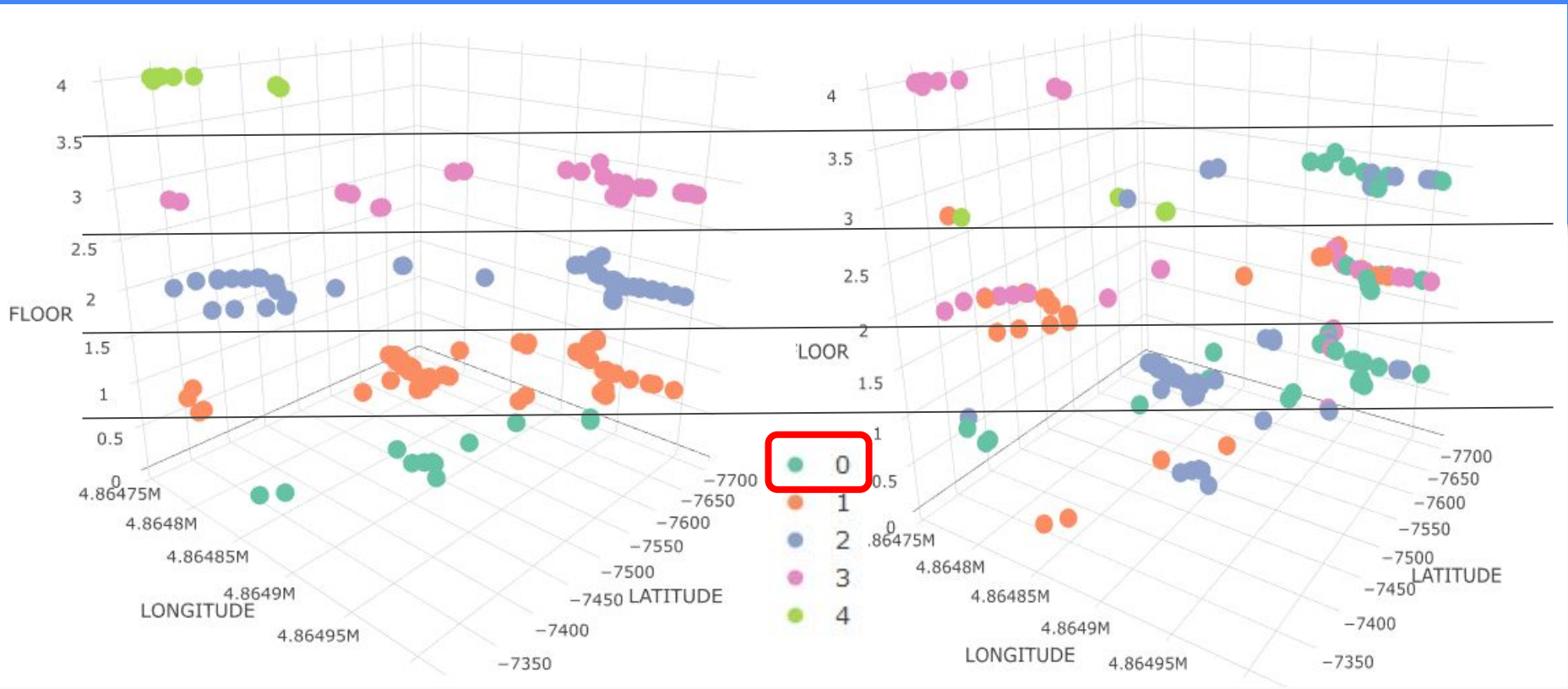
Mistakes:
F0: 11
F1: 35
F2: 39
F3: 22
F4: 8



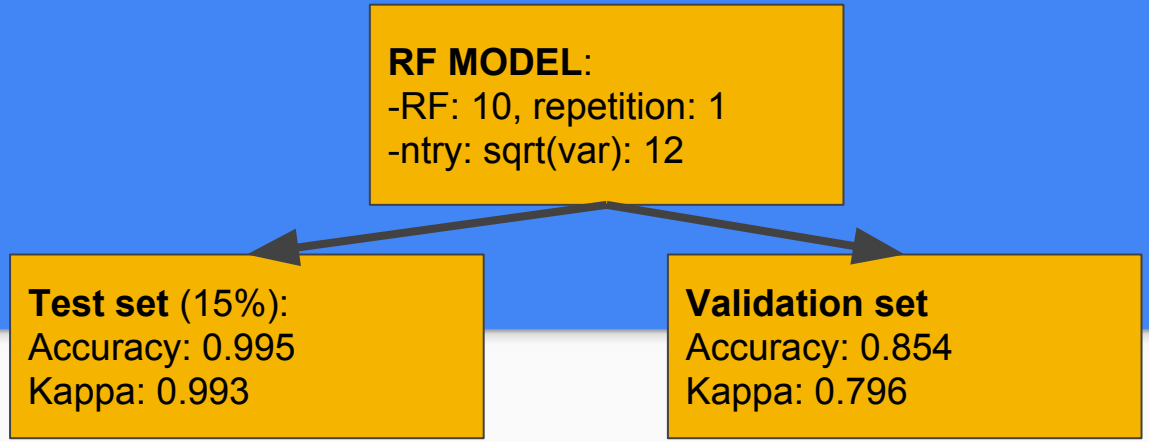
Mistakes:
F0: 12
F1: 81
F2: 47
F3: 28
F4: 8



15% of predictions are mistaking by more than 1 floor, but error is spread.

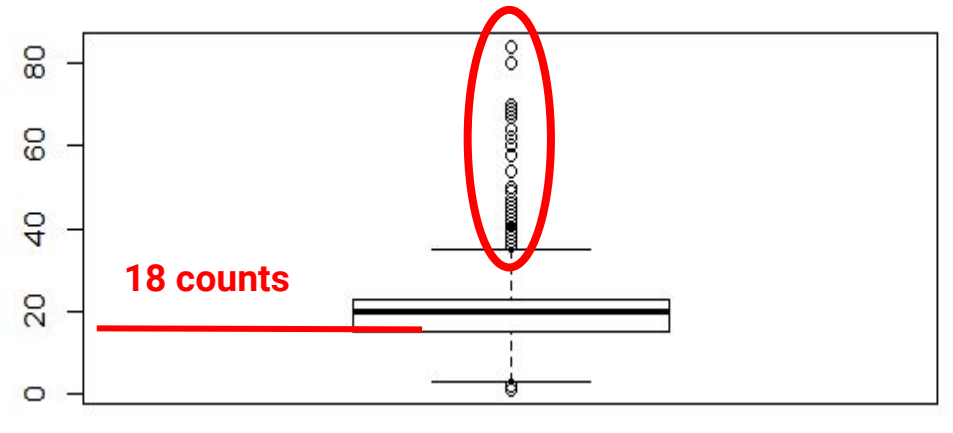


15% of errors are mistaking by more than 1 floor, but most of them belong to predicting Floor 0.


RF MODEL:
-RF: 10, repetition: 1
-ntry: sqrt(var): 12
Test set (15%):
Accuracy: 0.995
Kappa: 0.993
Validation set
Accuracy: 0.854
Kappa: 0.796

Predicting longitude and latitude

Bias



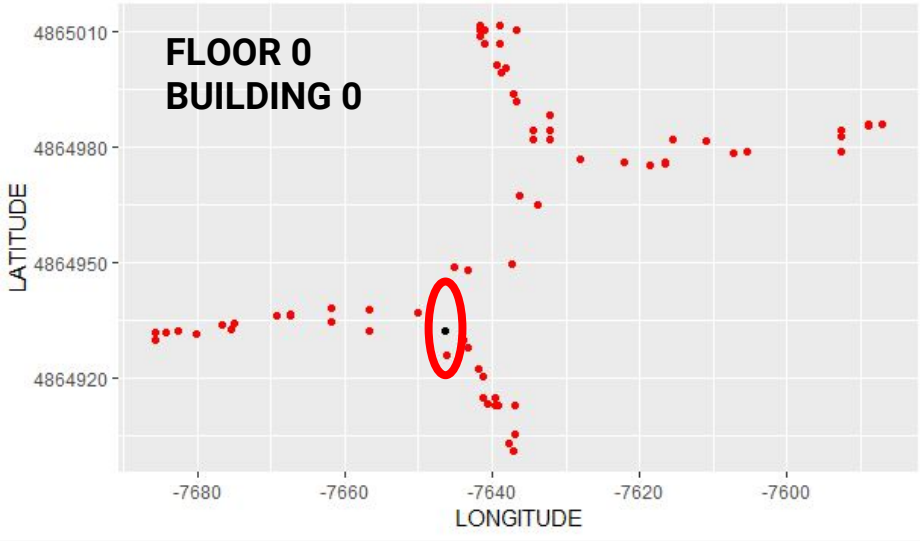
We reduce number of measurements per SPACE ID to 18.

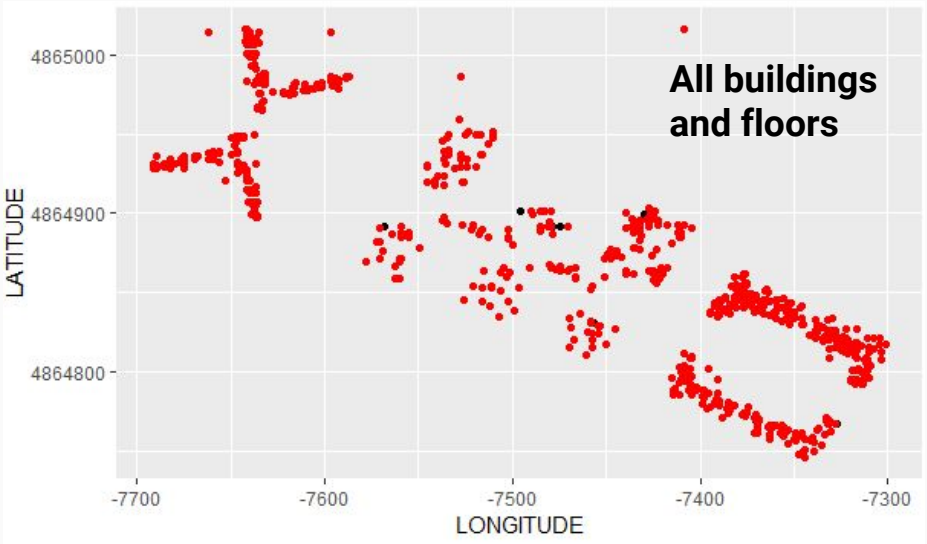
When doing this, MAE on test set drops more than 0.17m.

LONGITUDE

KNN MODEL:
-CV split: 10, repetition: 2
-Neighbors: 1

Test set (15%):
RMSE: 6.17
R-squared: 0.997
MAE: 1.04m

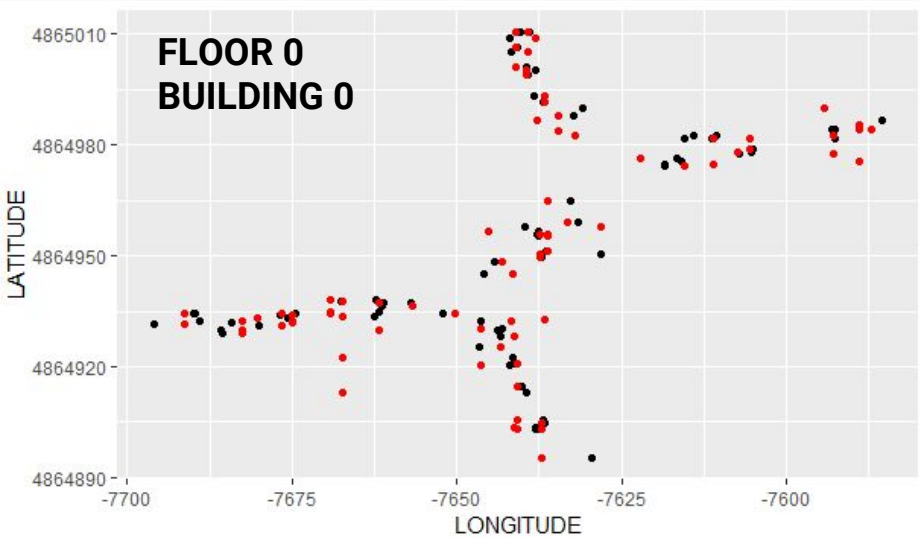


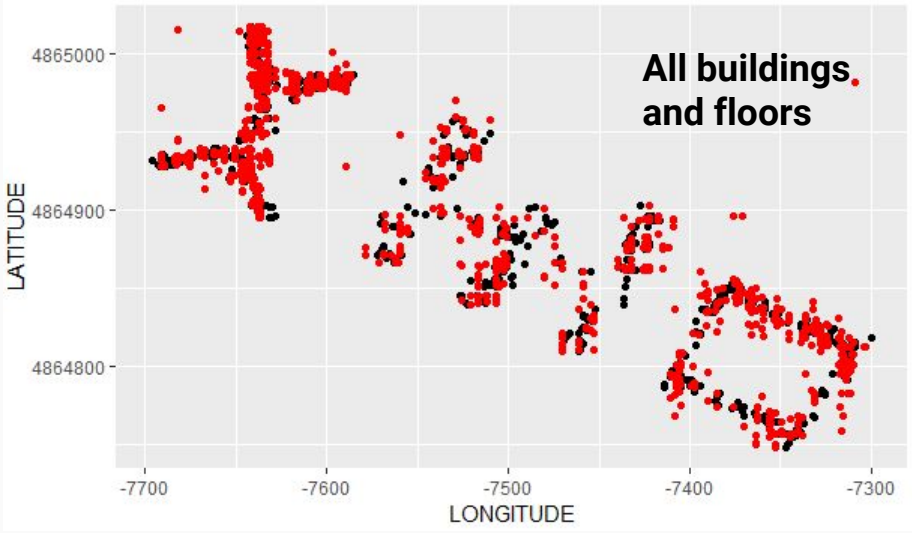


LONGITUDE

KNN MODEL:
-CV split: 10, repetition: 2
-Neighbors: 1

Validation set (15%):
RMSE: 19.56
R-squared: 0.973
MAE: 8.84m



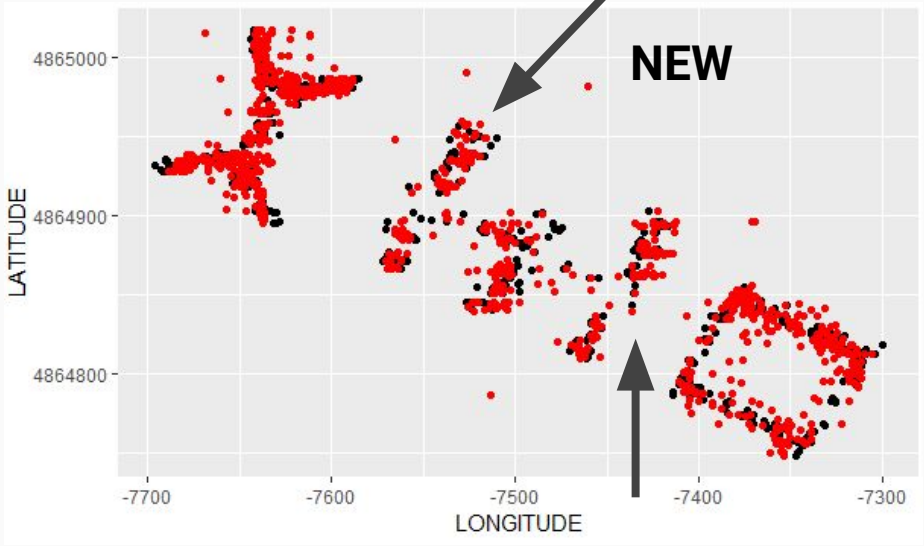


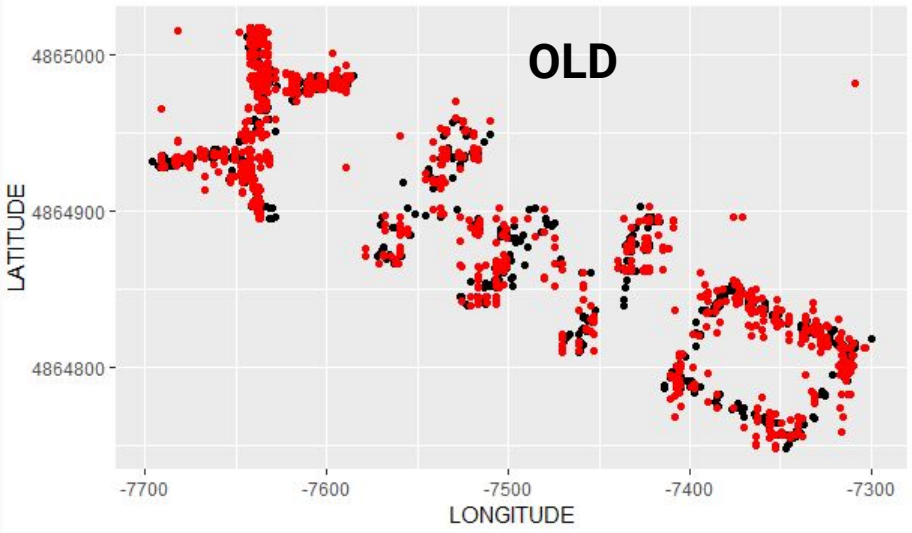
LONGITUDE

KNN MODEL:
-CV split: 10, repetition: 2
-Neighbors: 9

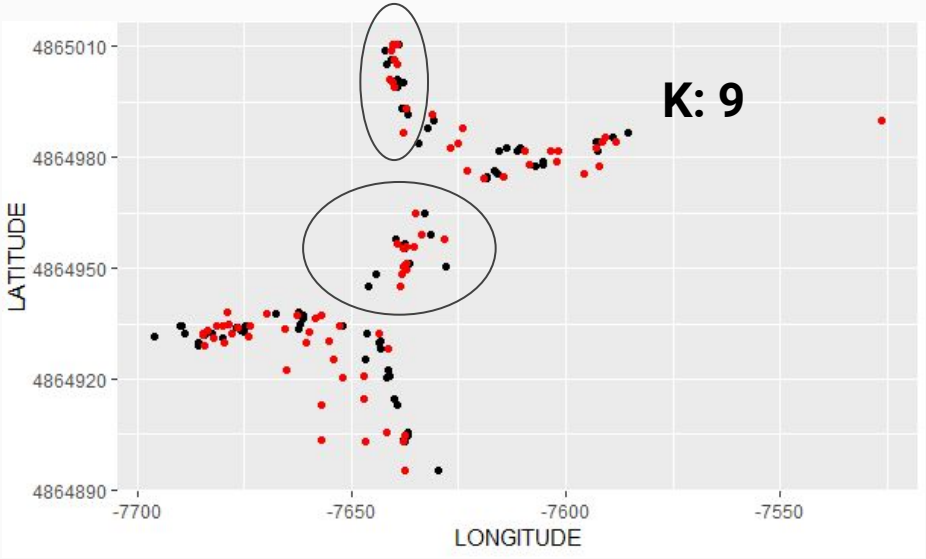
Test set (15%):
RMSE: 5.99
R-squared: 0.998
MAE: 3.08m

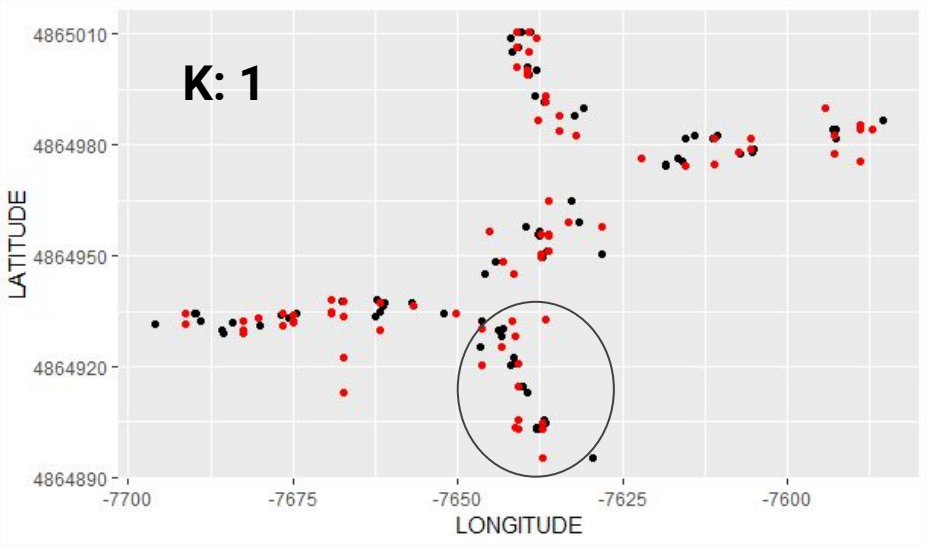
Validation set (15%):
RMSE: 14.65
R-squared: 0.985
MAE: 8.03m ↓ 0.81m





LONGITUDE


K: 9
4865010
4864980
4864950
4864920
4864890
LATITUDE
-7700
-7650
-7600
-7550
LONGITUDE


K: 1
4865010
4864980
4864950
4864920
4864890
LATITUDE
-7700
-7675
-7650
-7625
-7600
LONGITUDE

LATITUDE

KNN MODEL:
-CV split: 10, repetition: 2
-Neighbors: 9

LATITUDE WITH WAPS

Test set (15%):
RMSE: 5.24
R-squared: 0.99
MAE: 2.65m

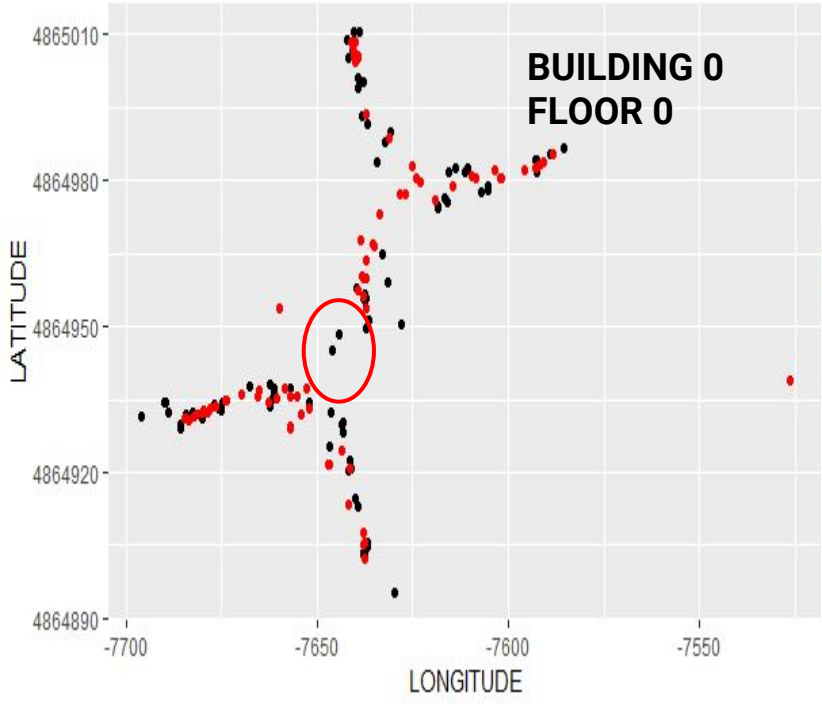
Validation set (15%):
RMSE: 15.62
R-squared: 0.951
MAE: 8.34m

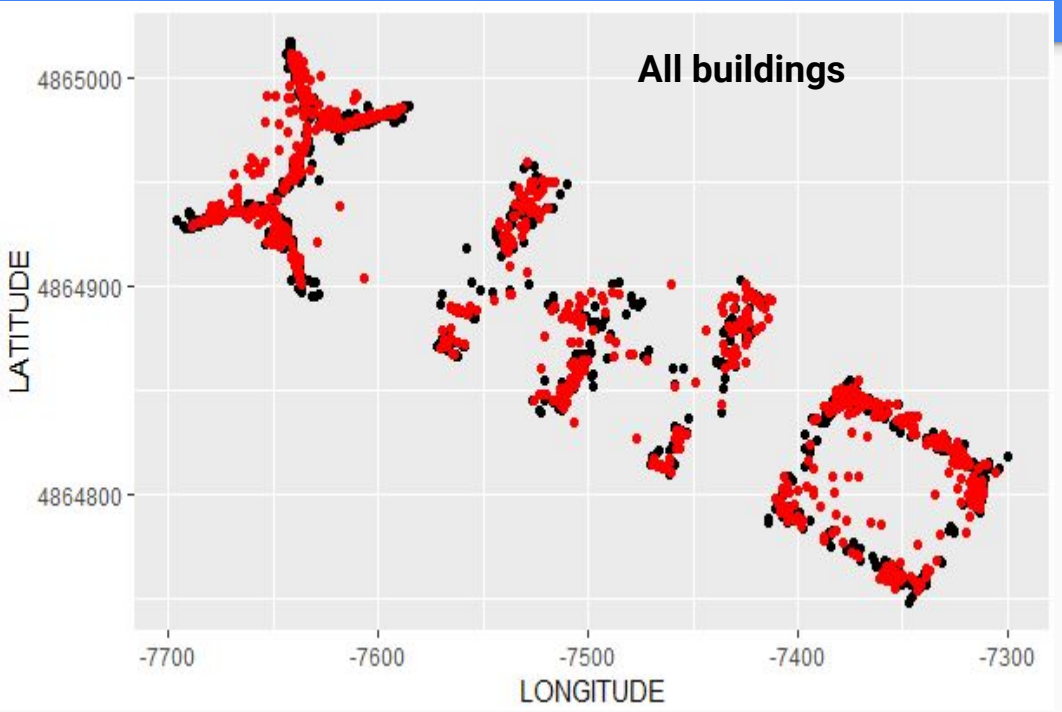
LATITUDE WITH WAPS AND LONGITUDE

Test set (15%):
RMSE: 5.04
R-squared: 0.99
MAE: 2.51m

Validation set (15%):
RMSE: 15.40
R-squared: 0.952
MAE: 8.21m

MAE: -0.13


BUILDING 0
FLOOR 0
4865010
4864980
4864950
4864920
4864890
LATITUDE
-7700
-7650
-7600
-7550
LONGITUDE


All buildings
4865000
4864900
4864800
LATITUDE
-7700
-7600
-7500
-7400
-7300
LONGITUDE

LONGITUDE

KNN MODEL:
-CV split: 10, repetition: 2
-Neighbors: 9

LONGITUDE WITH WAPS

Test set (15%):
RMSE: 5.99
R-squared: 0.998
MAE: 3.08m

Validation set (15%):
RMSE: 14.65
R-squared: 0.985
MAE: 8.03m

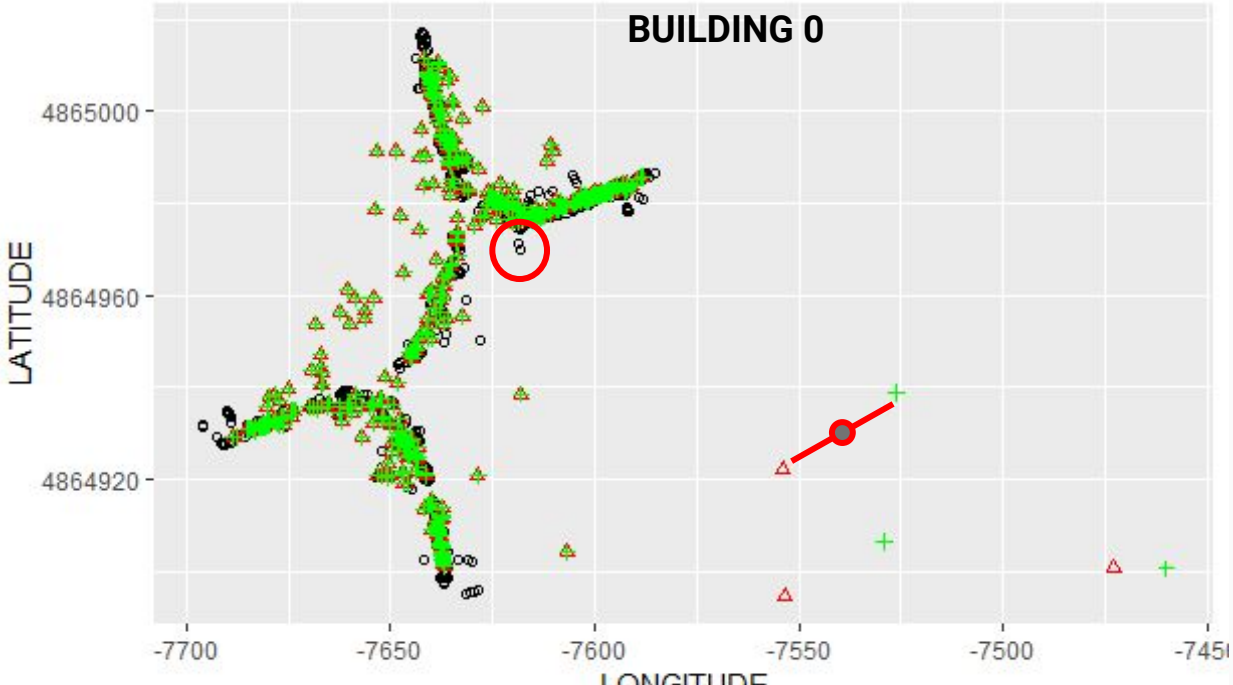
LONGITUDE WITH WAPS AND PREDICTED LATITUDE

Test set (15%):
RMSE: 6.01
R-squared: 0.998
MAE: 3.09m

Validation set (15%):
RMSE: 14.04
R-squared: 0.986
MAE: 7.93m

MAE: 0.10

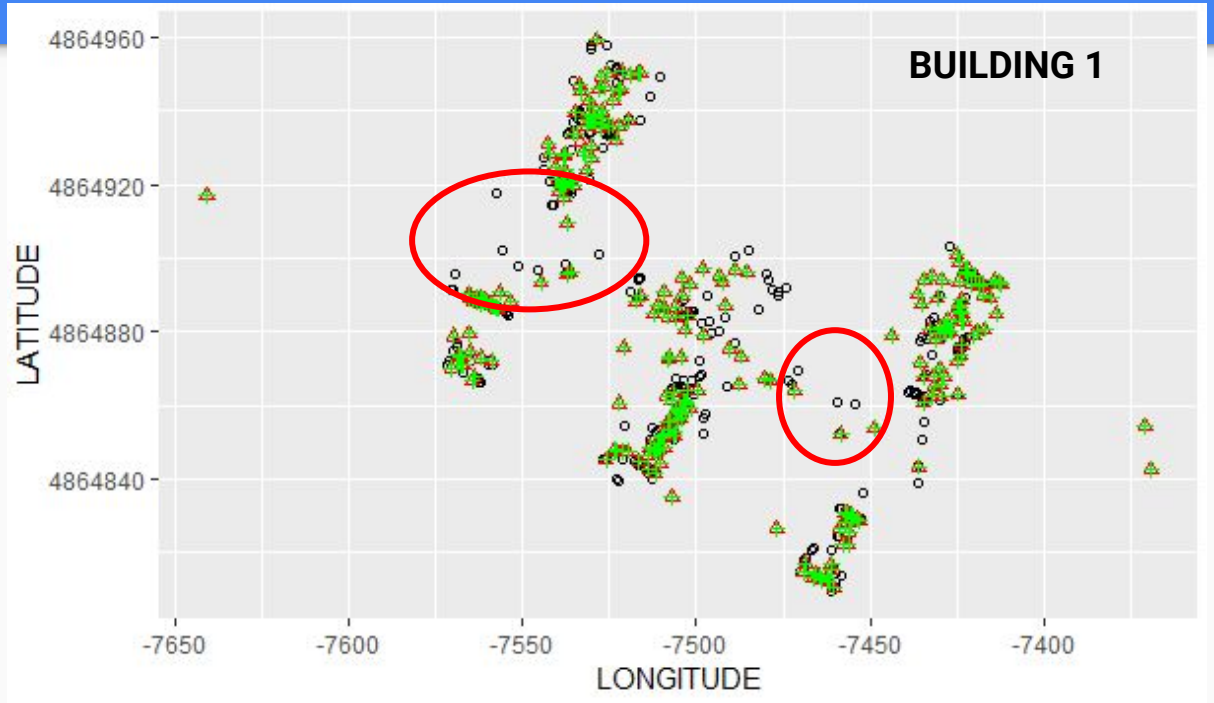
Predictions and Real values (BLACK)



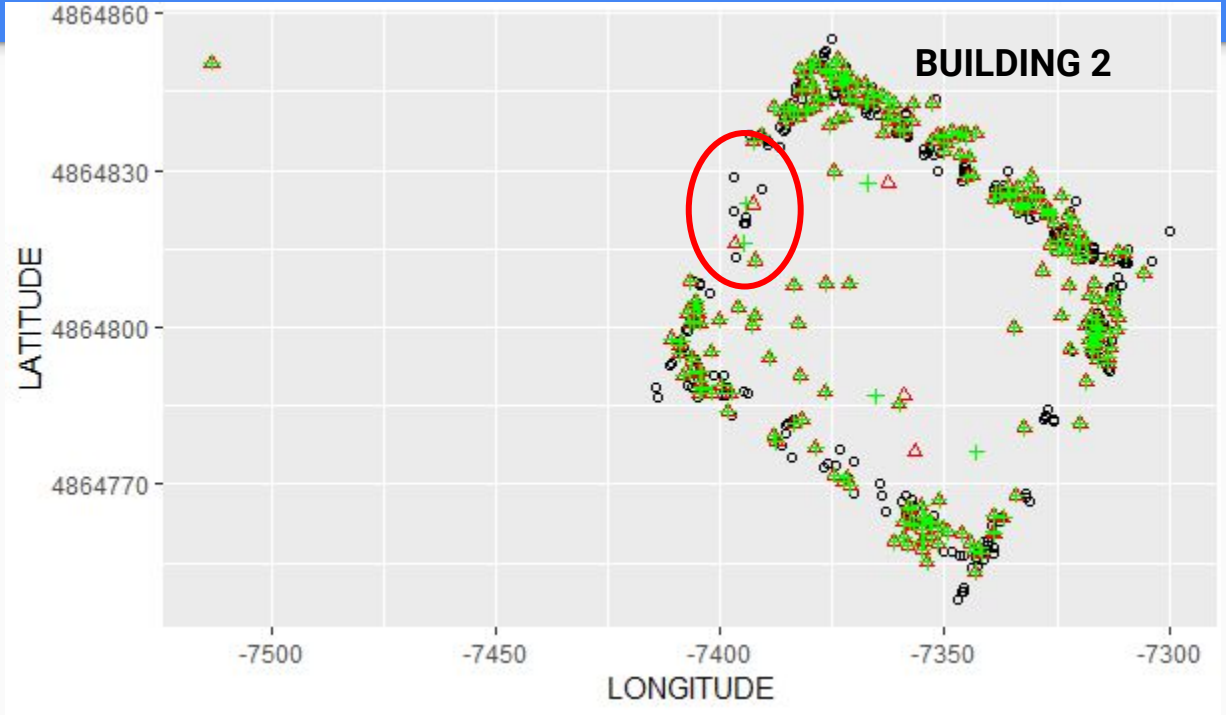
Getting the middle position doesn't work.

HIGHER ERROR:
Low density of measurements in space.

Predictions and Real values (BLACK)



Predictions and Real values (BLACK)



Final insights

What things improve our model?

BIAS: Making freq. Of observations similar works well when predicting BUILDING and LONGITUDE/ LATITUDE (not FLOOR).

OVERFITTING:
Increasing number of neighbors to 3 different SPACE IDs works for predicting BUILDING and FLOOR.
For LONGITUDE/ LATITUDE, this works for just similar measurements (9 neighbors).

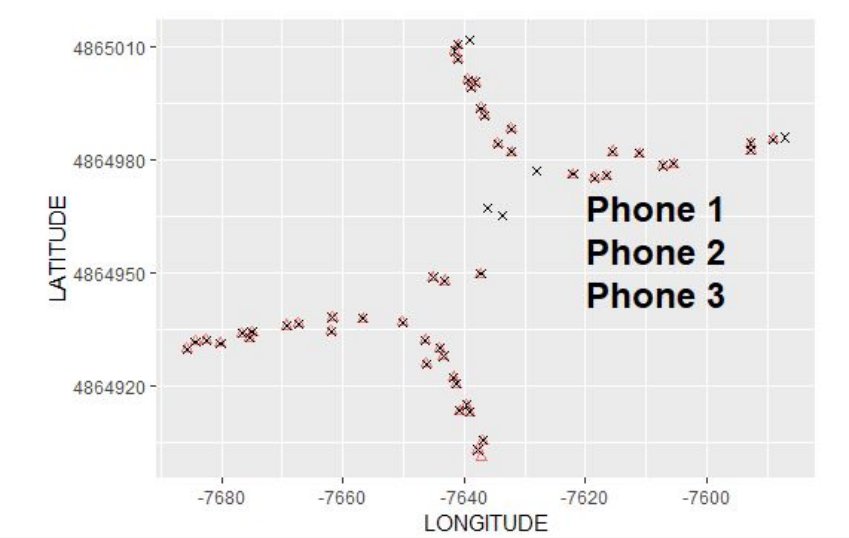
Predicting first LONGITUDE and then LATITUDE (or viceversa).

Next steps

PCA: After talking to other colleagues, it would seem that there's better ways to preprocess our data.

LOOKING WHERE OUR MODEL IS NOT PERFORMING WELL:
Understanding the nature of the points where our model is not working well for the test set. We can then make changes in our preprocessing to increase our chances of success.

BONUS!
User/ Phone behavior



Are phones working proportionally?

User 1: 0 0 2 6 0 8 2 → 0 0 .25 .75 0 1 .25 → 2.75
0 0 3 6 0 8 1 → 0 0 .38 .75 0 1 .13 → 2.25
Mean: 2.5

User 2: 0 0 5 12 0 16 2 → 0 0 .32 .75 0 1 .13 → 2.18
0 0 4 12 0 16 4 → 0 0 .25 .75 0 1 .25 → 2.75
Mean: 2.47

Users/ Phones 1 and 2 behave proportionally.

User 3: 0 0 0 3 0 4 0 → 0 0 0 .75 0 1 0 → 1.75
0 0 0 3 0 5 0 → 0 0 0 .60 0 1 0 → 1.6
Mean: 1.68

User/ Phone 3 is working differently, normalization by rows doesn't solve the phone issue.

Users/ Phones that don’t behave similarly

Differences in mean > 25%

7 users/ phones with big differences.
All floors of building 2.
Floor 3 of building 1.

General differences >85-95%

Insights

Nothing conclusive:

- **>20 cm difference in height.**
- **Android version (2 or 4) doesn’t affect.**
- **Timestamps affect (different furniture allocation, people…): phones GT-S6500 2.3.6 and HTC Wildfire S 2.3.5 (same user) have high and low differences depending on the building.**
- **Different phones inconclusive (big and small differences for different phones)**