Predicting Indoor Location

Hardik Shahu

The Main Goal

The primary goal of this project is to find out if we can predict one’s location based on Wi-Fi Signals. In order to do that, we will build a few models and see which one performs the best. To be more specific, we will compare their accuracy and kappa and pick the one that performs well, assuming that we are able to predict. Additionally, we will also give any future recommendations if we have any by the end.

The Data

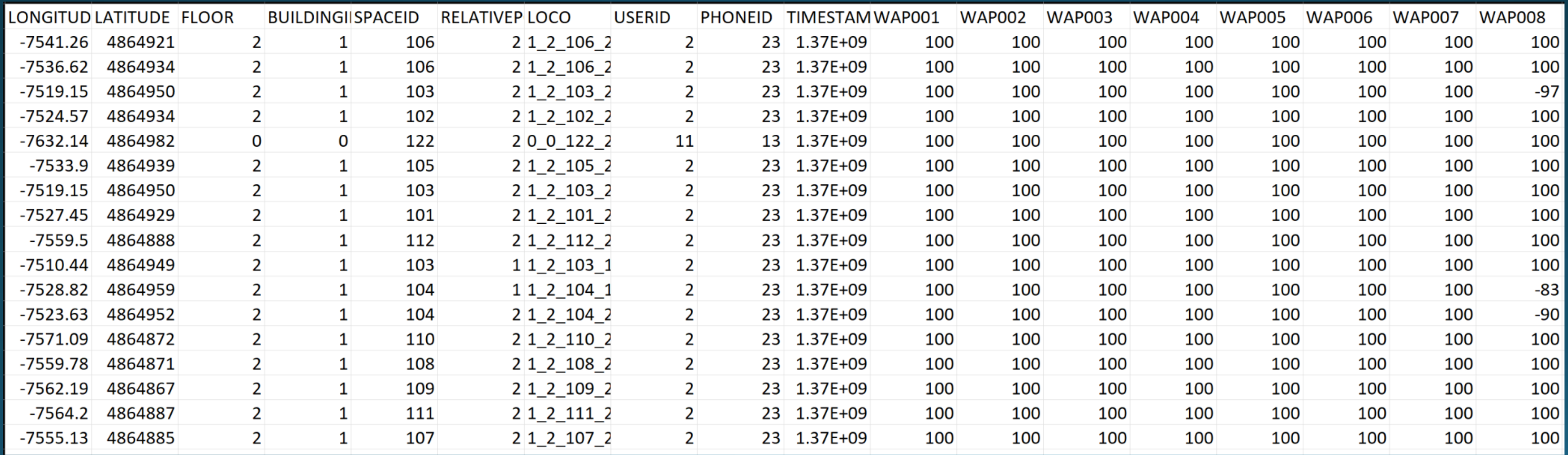
For this project, we were given the data. The data had ~20k entries with attributes that included longitude, latitude, floor building ID, space ID, relative position, user ID, phone ID, timestamp and 500 Wi-Fi access points (WAPs).

For our purpose, the relevant attributes were building ID, floor, space ID, relative position and the 500 WAPs.

All the WAPs will be our independent variables while the other four will be our dependent variables.

Prepping the Data

This time around, we didn’t have to modify the data too much. The most significant change we did was concatenating our 4 dependent variables into one new attribute (we titled it “LOCO”). Another minor modification was that we had to change our data types to be factors. One important thing to note is that our dependent variable currently has ~900 factors.



LONGITUD	LATITUDE	FLOOR	BUILDINGII	SPACEID	RELATIVEP	LOCO	USERID	PHONEID	TIMESTAM	WAP001	WAP002	WAP003	WAP004	WAP005	WAP006	WAP007	WAP008
-7541.26	4864921	2	1	106	2	1_2_106_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7536.62	4864934	2	1	106	2	1_2_106_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7519.15	4864950	2	1	103	2	1_2_103_2	2	23	1.37E+09	100	100	100	100	100	100	100	-97
-7524.57	4864934	2	1	102	2	1_2_102_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7632.14	4864982	0	0	122	2	0_0_122_2	11	13	1.37E+09	100	100	100	100	100	100	100	100
-7533.9	4864939	2	1	105	2	1_2_105_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7519.15	4864950	2	1	103	2	1_2_103_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7527.45	4864929	2	1	101	2	1_2_101_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7559.5	4864888	2	1	112	2	1_2_112_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7510.44	4864949	2	1	103	1	1_2_103_1	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7528.82	4864959	2	1	104	1	1_2_104_1	2	23	1.37E+09	100	100	100	100	100	100	100	-83
-7523.63	4864952	2	1	104	2	1_2_104_2	2	23	1.37E+09	100	100	100	100	100	100	100	-90
-7571.09	4864872	2	1	110	2	1_2_110_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7559.78	4864871	2	1	108	2	1_2_108_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7562.19	4864867	2	1	109	2	1_2_109_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7564.2	4864887	2	1	111	2	1_2_111_2	2	23	1.37E+09	100	100	100	100	100	100	100	100
-7555.13	4864885	2	1	107	2	1_2_107_2	2	23	1.37E+09	100	100	100	100	100	100	100	100

Sampling the Data (Optional)

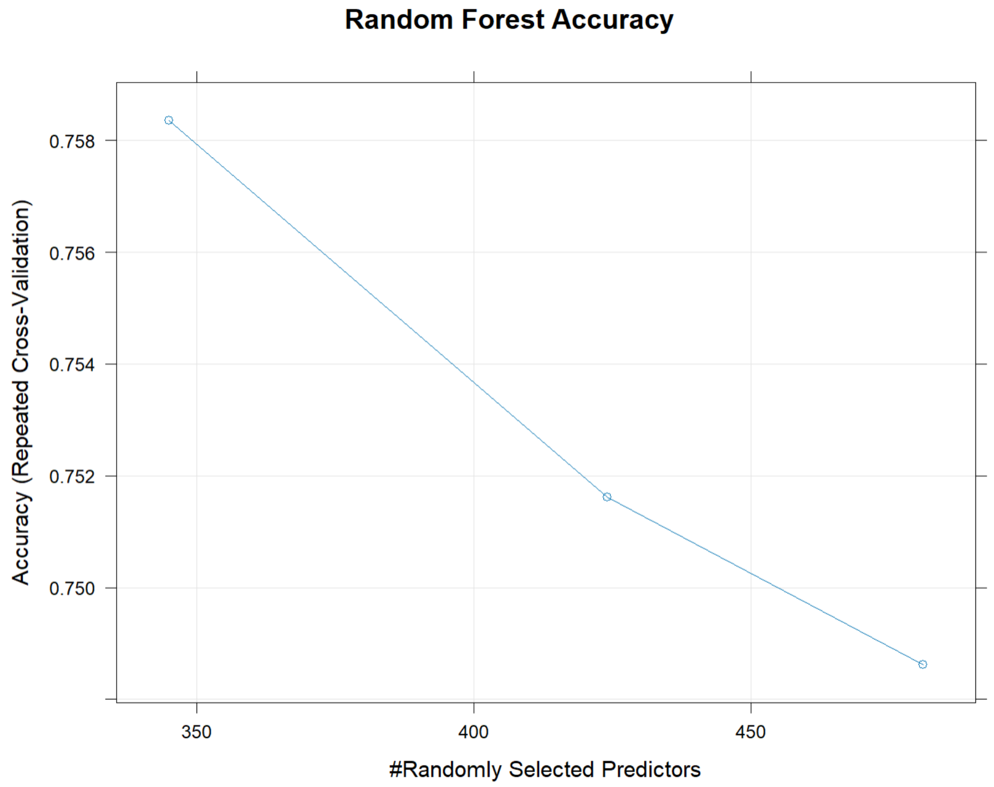
If we had a very powerful machine, we could theoretically move on to the next step. However, due to time constraints and weaker hardware, we must do this step.

There are a couple of ways to do this: to randomly pick a sample or to limit it to only one building. We choose the latter and the main reason being, the ratio of dependent factors to sample size. In order words, had we randomly picked a sample, we would still likely end up with the same number of factors, but with less data. This will make our predictions worse by nature. Therefore, if we reduce data to one building, we will also be decreasing our factors. In this case, we went from ~20k to ~5k entries and ~900 to ~200 factors.

Model 1: Random Forest

The first model we tried was Random Forest Classifier. The reason we choose this was that this model seemed to work very well for our past projects. Therefore, we had to try it out.

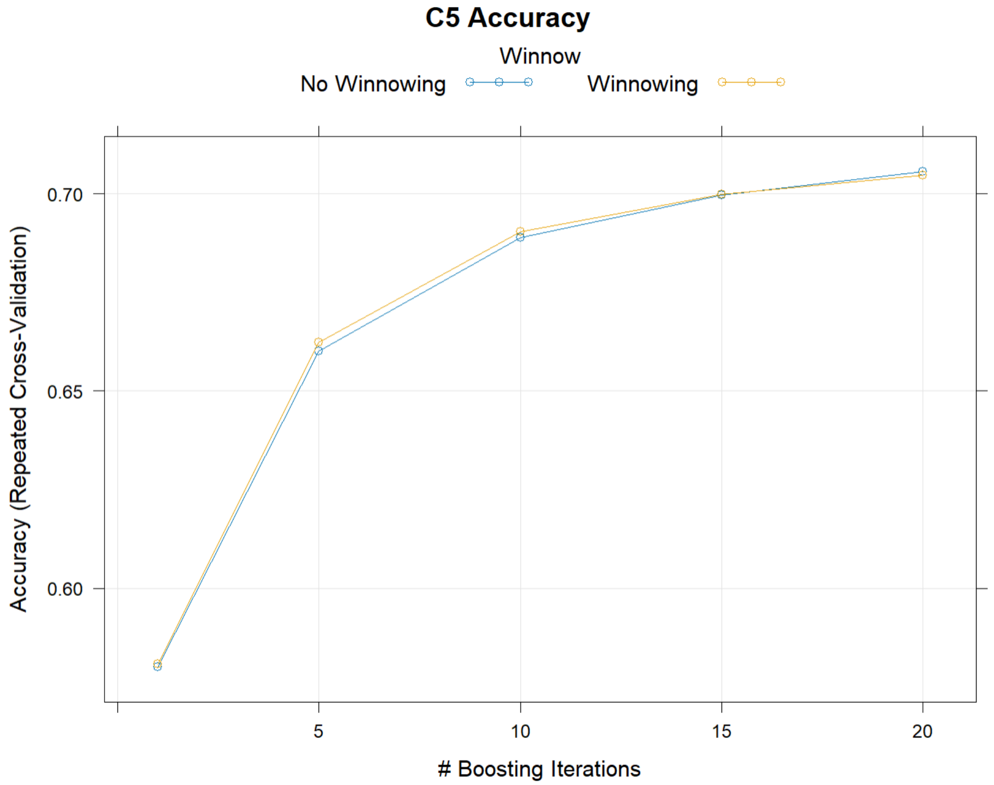
As expected, it performed decently well. The figure on the right shows the accuracy for a given number of randomly selected predictors. Around the 350 mark, it performs the best, however, the other ones aren't that far off; the difference is in the thousandth decimal place (which is about less than 1%).



Model 2: C5 Decision Tree

The second one we tried was C5 Classifier. This algorithm has many advantages as that it works very well out of the box for most situations compared to more sophisticated ones like NN or SVM.

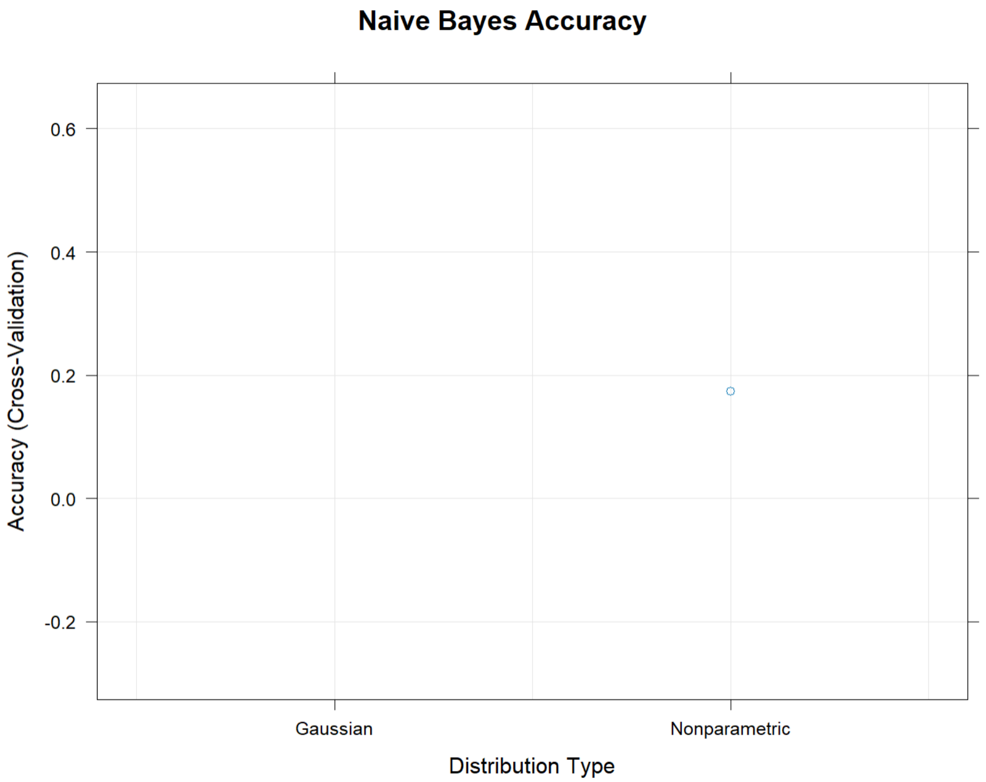
The results is shown on the right and as you can see, it performs decently well. Whether we are winnowing or not, the accuracy was relatively the same: ranging from ~0.55 to ~0.71. A pattern we can see is that that the greater number of boosting Iterations we use, the better accuracy it seems to have with diminishing returns.



Model 3: Naive Bayes

For our third one, we tried Naive Bayes Classification. The main reason is that it is relatively light and easy to run when compared to more complex ones such as SVM or GBM. When you have a dataset our size, having a light model is desirable as then it will cut back training time.

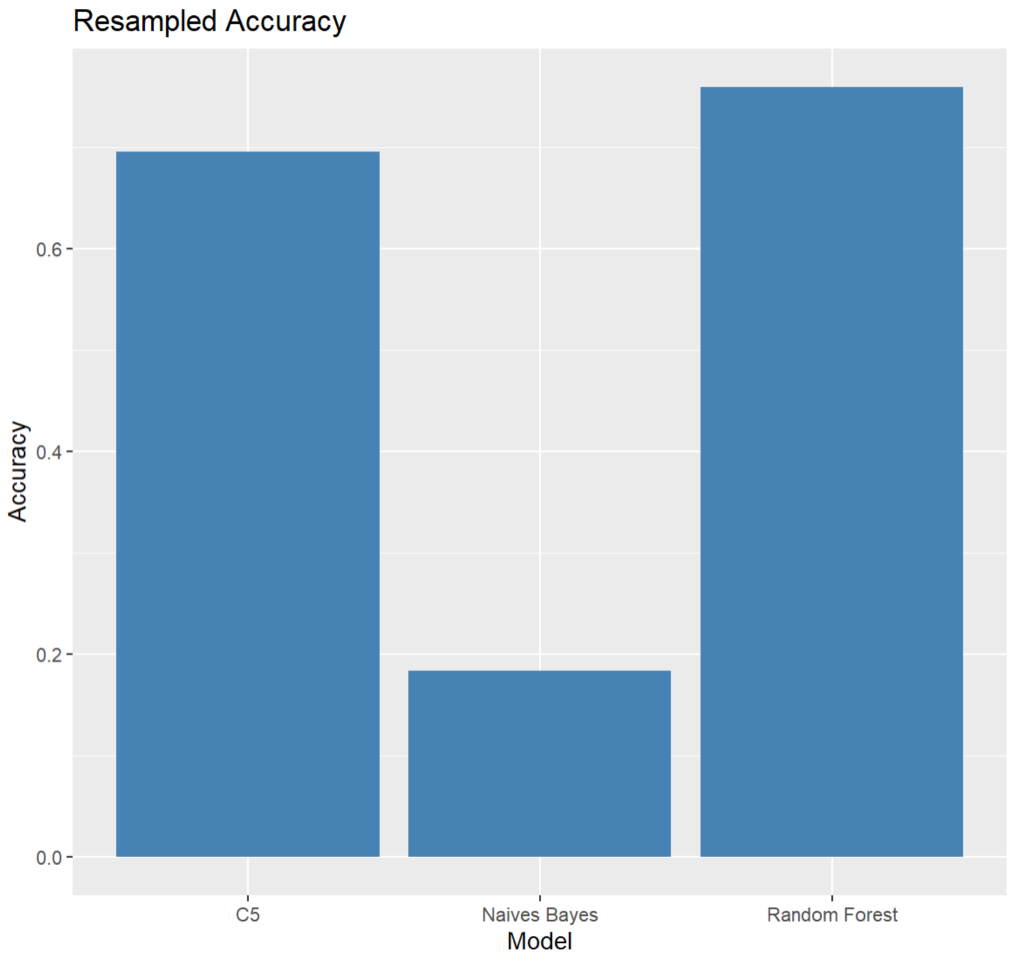
However, unfortunately, NB did not perform well at all. When the distribution type was Gaussian, it didn't work at all and for nonparametric we got an accuracy value of~0.20.



Comparing Accuracy

After we trained the three models, we ran a resample test and these were the accuracy results.

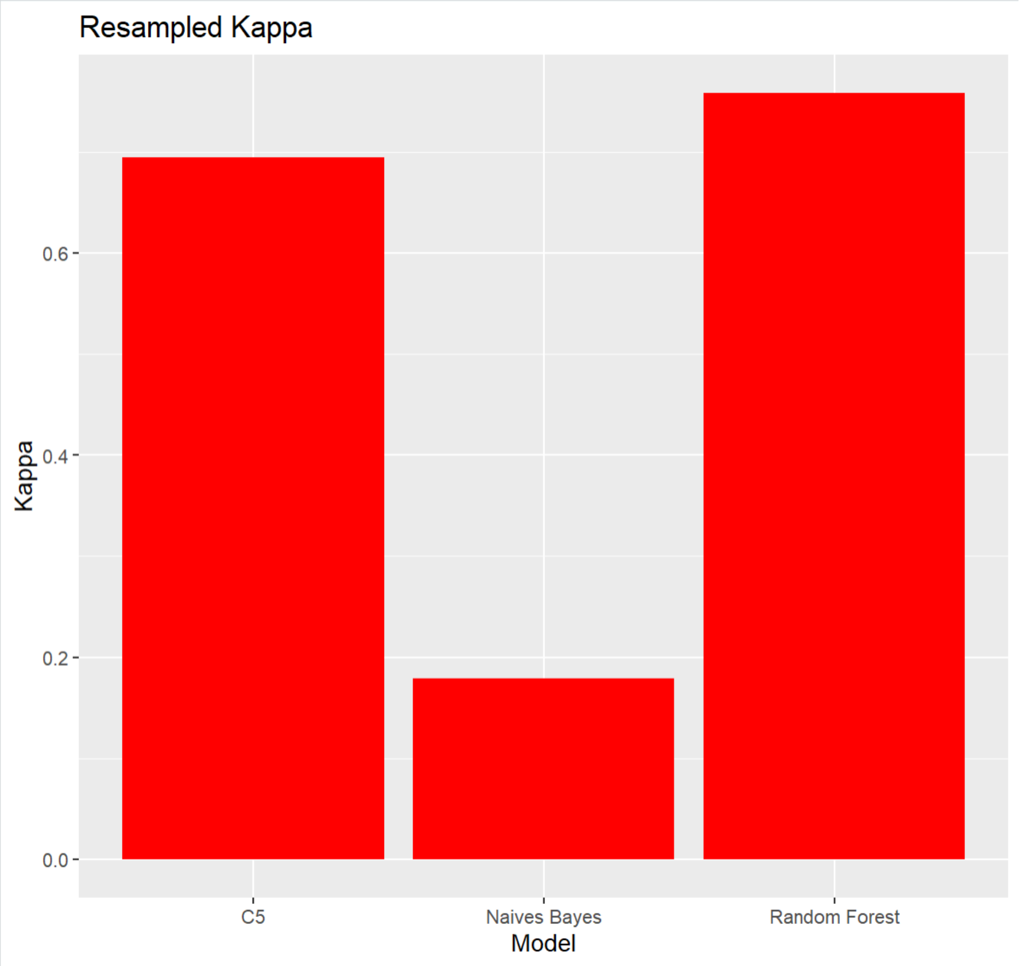
As you can see, Random Forest seems to have worked the best with having a value of 0.760



Comparing Kappa

When we ran the resample test, we also looked at the Kappa value as this is important too when measuring how well a model performs.

The results are not too different than the accuracy ones as the highest kappa, found in Random Forest again, was 0.759 which is only about 0.1% different it.



Therefore, we should use Random Forest to predict the location!

Important Notes to consider

- Ideally, we would want to train with all the data as that leads to best possible results, but that was out of our scope with the time constraint and our hardware.
- Generally, we desire an accuracy and kappa score of about 0.80 or higher to be confident in our predictions. However, when you consider the dataset's size and scope, we are satisfied with our best result.
- There was another model we attempted to train, Gradient Boosting Classifier (GBM). However, it would not train fast enough, so we had to abandon this model.

Future Recommendations

- If it’s possible, we believe that upgrading our computers to have more powerful components will enable us to do many things such as:
  - Train existing models much faster with the same sample.
  - Train using the entire dataset for better and more accurate results.
  - Allow us to use more powerful models that take longer to train but could yield better results.
- We should definitely keep considering Random Forest for future projects as it has worked well for this and our past projects!

Conclusion

In the end, we managed to find a way to predict the location by training the model Random Forest due to it performing the best of out the ones we test. Additionally, we also had some recommendations such as upgrading our computers and using Random Forest as one of the training models for future endeavors.