Report

Objectives

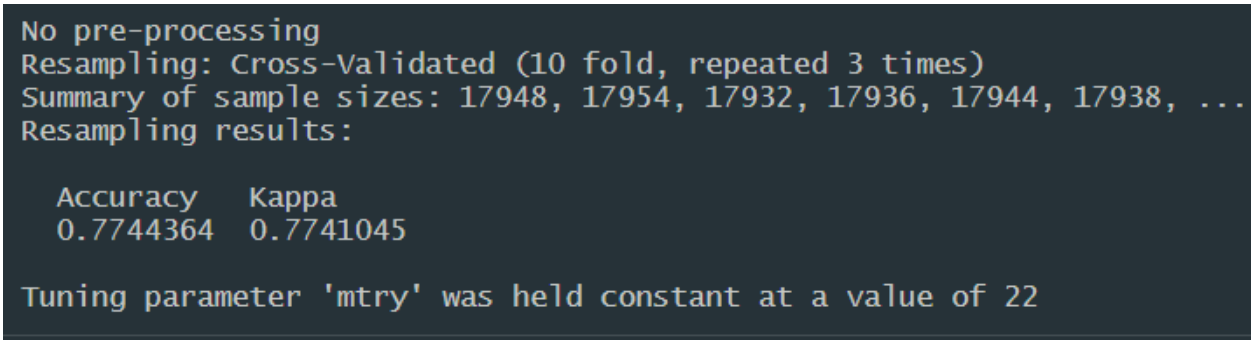
The main objective is to build a model that can correctly predict in-campus locations using Wi-Fi signal strength from various routers around the buildings. This indoors positioning proves to be tricky as the multiple devices interfere with each other and prevents geo-location using the usual GPS signals.

Data Cleaning and optimization

Initially I tried to run the raw data through the entire modelling algorithm of Random Forest, with the computing power I have, it was able to run over a period of 6hrs. However, this had to be improved for the other algorithms, therefore, I took advantage of R parallelization capabilities and used the *doParallel* library to utilize 7 cores out of the 8 available on my machine. Furthermore, I reduced the number of instances from 19,937 on each of the 529 features to only 5249 instances over the 529 features. This was achieved by only looking at one building (building 0). This combination allowed my computer to runt the random forest modeling algorithm in 5 minutes. With these new settings in place I ran 3 other algorithms: K-Nearest Neighbors (KNN), SVM Radial, and C50.

Results

Random Forest – All data results

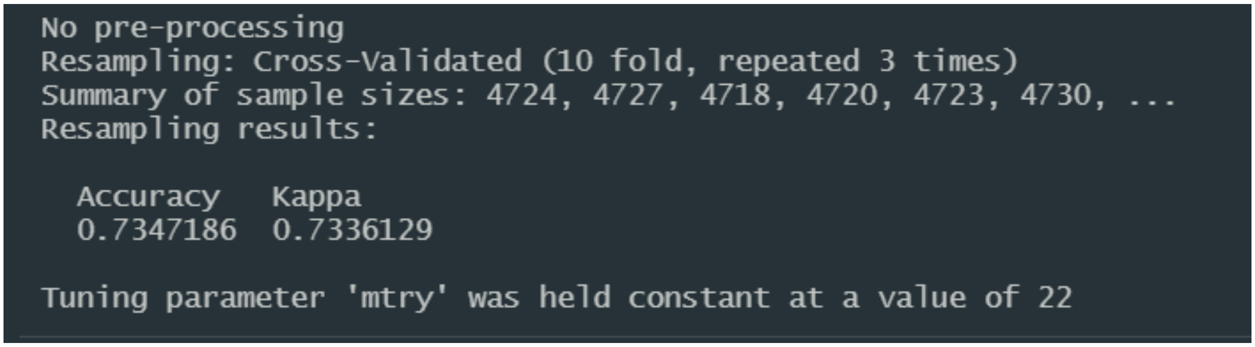


```
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 17948, 17954, 17932, 17936, 17944, 17938, ...
Resampling results:

  Accuracy   Kappa
  0.7744364  0.7741045

Tuning parameter 'mtry' was held constant at a value of 22
```

Random Forest - Building 0

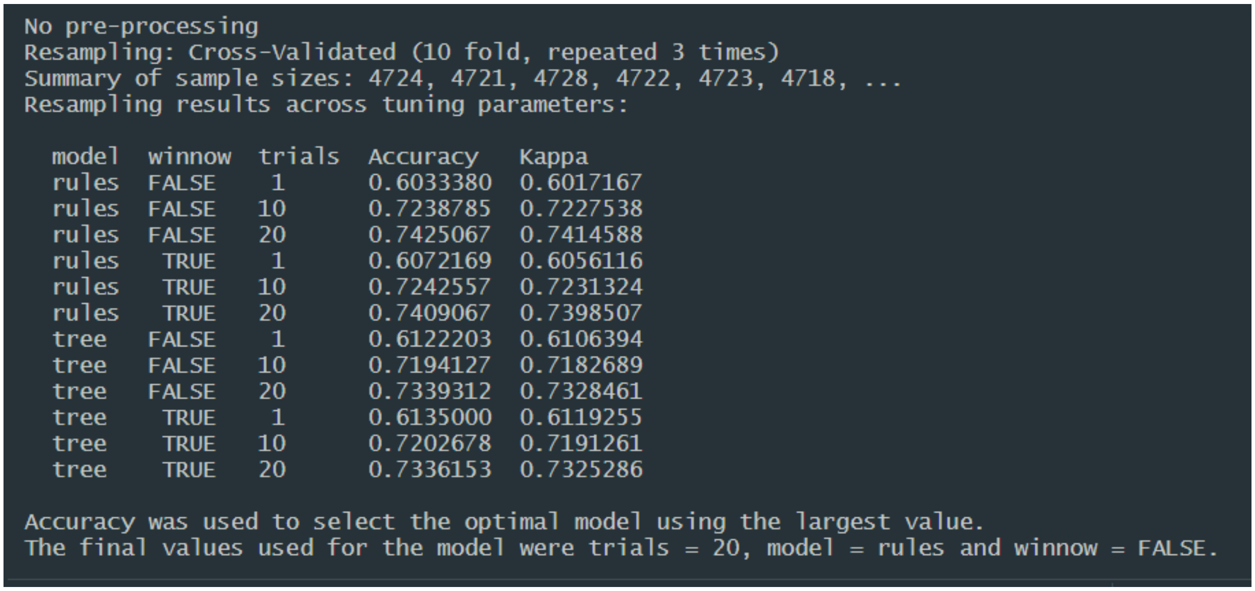


```
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 4724, 4727, 4718, 4720, 4723, 4730, ...
Resampling results:

  Accuracy   Kappa
  0.7347186  0.7336129

Tuning parameter 'mtry' was held constant at a value of 22
```

C50 - Building 0

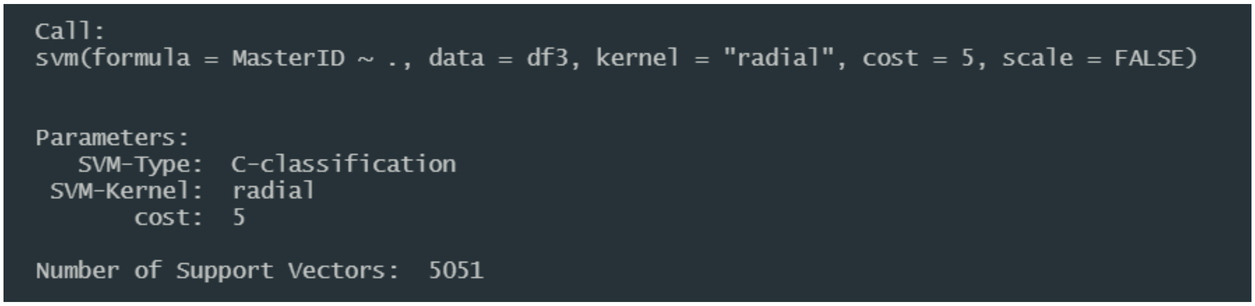


```
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 4724, 4721, 4728, 4722, 4723, 4718, ...
Resampling results across tuning parameters:

  model  winnow  trials  Accuracy   Kappa
  rules  FALSE    1      0.6033380  0.6017167
  rules  FALSE   10      0.7238785  0.7227538
  rules  FALSE   20      0.7425067  0.7414588
  rules   TRUE    1      0.6072169  0.6056116
  rules   TRUE   10      0.7242557  0.7231324
  rules   TRUE   20      0.7409067  0.7398507
  tree   FALSE    1      0.6122203  0.6106394
  tree   FALSE   10      0.7194127  0.7182689
  tree   FALSE   20      0.7339312  0.7328461
  tree    TRUE    1      0.6135000  0.6119255
  tree    TRUE   10      0.7202678  0.7191261
  tree    TRUE   20      0.7336153  0.7325286

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were trials = 20, model = rules and winnow = FALSE.
```

SVM Radial Fit - Building 0

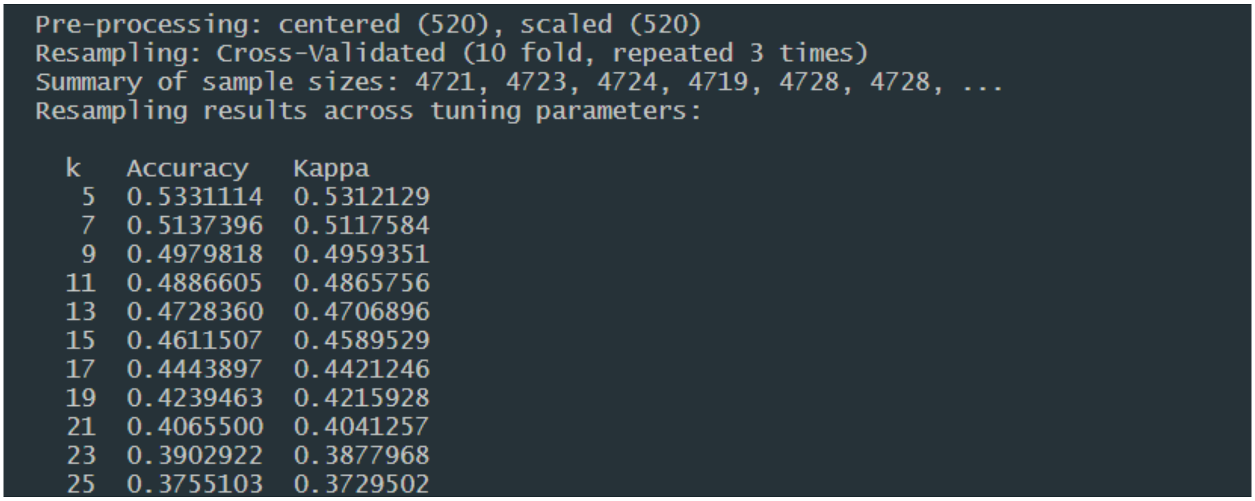


```
Call:
svm(formula = MasterID ~ ., data = df3, kernel = "radial", cost = 5, scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  5

Number of Support Vectors:  5051
```

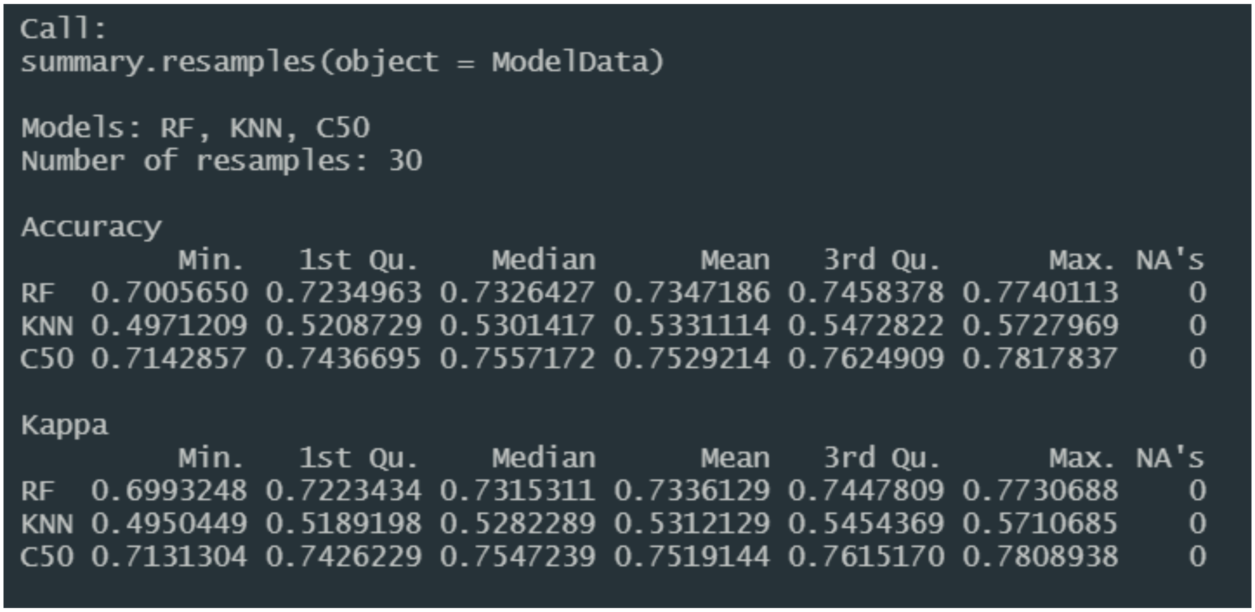
K-Nearest Neighbors (KNN) – Building 0



```
Pre-processing: centered (520), scaled (520)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 4721, 4723, 4724, 4719, 4728, 4728, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.5331114  0.5312129
   7  0.5137396  0.5117584
   9  0.4979818  0.4959351
  11  0.4886605  0.4865756
  13  0.4728360  0.4706896
  15  0.4611507  0.4589529
  17  0.4443897  0.4421246
  19  0.4239463  0.4215928
  21  0.4065500  0.4041257
  23  0.3902922  0.3877968
  25  0.3755103  0.3729502
```

<u>Summary of Results</u>



```
Call:
summary.resamples(object = ModelData)

Models: RF, KNN, C50
Number of resamples: 30

Accuracy
         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
RF  0.7005650 0.7234963 0.7326427 0.7347186 0.7458378 0.7740113    0
KNN 0.4971209 0.5208729 0.5301417 0.5331114 0.5472822 0.5727969    0
C50 0.7142857 0.7436695 0.7557172 0.7529214 0.7624909 0.7817837    0

Kappa
         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
RF  0.6993248 0.7223434 0.7315311 0.7336129 0.7447809 0.7730688    0
KNN 0.4950449 0.5189198 0.5282289 0.5312129 0.5454369 0.5710685    0
C50 0.7131304 0.7426229 0.7547239 0.7519144 0.7615170 0.7808938    0
```

Based on these results, the best algorithm to use is the C50 algorithm, however if run time is not an issue, the Random Forest prediction using the full range of data is the best choice with the highest Accuracy and Kappa scores.

<u>Recommendations</u>

Attempting other algorithms with using cloud computing services from amazon (aws) is an option that is worth mentioning. For this project I attempted to use these but proved to be more problematic and complicated than just cleaning and refining the data.

Taking principal components analysis (PCA) and applying it to this large amount of data can be a useful tool to further increase the time output for each model.