Business Question:

Is it feasible to use wifi fingerprinting to determine a person's location in indoor spaces?

Background:

The purpose of this project is to determine how well wifi fingerprinting can be used to help people navigate an indoor space. We have been provided data that consists of recorded wifi fingerprinting at specific longitude and latitude points in three buildings at the Jaume I University in Spain. These buildings have multiple floors and different kinds of rooms (offices, labs). The data further specified whether the location was in or outside a corridor.

Scope:

This is a vast project and we have time constraints. To simplify this project and focus on the business question (Is this feasible?), we selected a sample of the dataset, specifically, building 0, floor 0. We will examine whether the data from wifi fingerprinting can tell us a person's location, by latitude and longitude. The reason we are looking at one building only and one floor only is because in a real-life situation, if someone wanted to know what building they were in through their phone, they could do so using already well-established outdoor GPS technology. Also, floor level is usually intuitive. In buildings, elevators have signs that signal what floor you are on. Sometimes, you can visually see what floor you are on. If time allows, in the future, we can expand the scope of this project to determine location, not only by latitude and longitude, but by floor level and building number.

We will build several machine learning models to identify location based on attributes in the dataset. Based on the results of our model, we will make a recommendation on whether this technology is feasible for identifying location indoors.

Data:

14953 instances, 529 variables
Variables: 520 Wireless Access Points, Building ID, Floor ID, SpaceID, Relative Position, UserID, PhoneID, TimeStamp, Longitude, Latitude

1. https://www.mist.com/documentation/rssi-values-good-bad-signal-strength/ , https://www.netspotapp.com/what-is-rssi-level.html
2. https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/
3. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Dimensionality Reduction:

These are a lot of variables. We used several techniques to reduce the dimension of this dataset. First, we dropped several variables that would not be able to be used in real life situations. We removed User ID, Phone ID, Space ID, Relative Position, and TimeStamp.

After reviewing the documentation, we learned that SpaceID and Relative Position would not be available in a real-life scenario. Therefore, we removed it as to test feasibility on on

We are not using these as predictor variables. The purpose of this project is to see if identifying user location based on attributes for a shopping mall is feasible.

In real life situations, UserID and phoneID will always change. We are looking for feasibility on large campuses like shopping malls. In large places, we may have numerous users, especially in public places like shopping malls. Not only will these ids change, researching the relationship between UserID, Location, and perhaps even TimeStamp would be indicative of a behavioral pattern project and not simply for identifying location. If the scope of this project was for a corporate campus, with permanent employees, we may be able to delve into behavioral patterns to identify location by employee. However, this is not in our scope as we are simply looking at wifi fingerprinting and location. Therefore, we will remove UserID,. Similarly, we could go into another study to see how the type of phone would affect the strength of wifi fingerprinting, but that is out of the scope of this project. Therefore, we will remove PhoneID. We want to be able to identify location independent of who the user is or what time of phone they are using.

Similarly, we want to be able to identify location independent of time. While there may be a logical relationship that exists inherently between time and location (lunch hours may have more people in the cafeteria [behavioral patterns]), time as a variable is out of the scope of this project. To effectively study time, we would need to observe it on data from a controlled experiment. Based on the documentation, the data wasn't recorded with the intention of using time as predictor variable, but simply as a timestamp, therefore we dropped it.

Per the documentation for the dataset,spaceID and relativePosition, would not be available in real-life scenarios. Since we are looking for feasibility in real-life scenarios, we will not use these two variables either.

Merge variables: In addition, we merged latitude and longitude into one target variable.

1. https://www.mist.com/documentation/rssi-values-good-bad-signal-strength/ , https://www.netspotapp.com/what-is-rssi-level.html
2. https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/
3. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Filter dataset: We filtered our data to only include instances where the location was in building 0 and floor 0.

Remove zero variance predictors: There were some WAPs that have constant values and no change. Some WAPs, for example, were for building 2, and obviously, reported no RSSI for building 0, which is useless for our model.

Remove low signal predictors: Per our research (see 1), RSSI signals starting at -70 dbm and lower are weak and correspond with low quality signal connections. We removed and WAPs that had only weak signals, meaning it has only RSSI levels at or below -70 dbm.

After these techniques, we reduced our variables from 529 to 38.

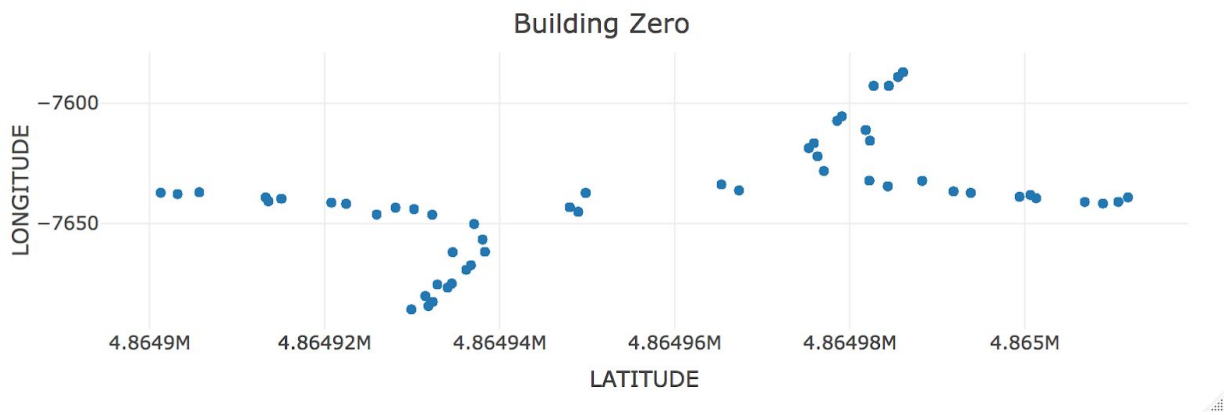
Train and Select Model:

Which models do we use? We start this approach by learning what type of variable our target is. Our goal is to identify location, which is specifically the combination of latitude and longitude values. Latitude and Longitude are continuous variables, however, in this case, our range is not continuous. The latitude and longitude values are limited to the size and location of the building and furthermore, the specific data points recorded within that. There are only 54 unique combinations of latitude and longitude for building 0, floor 0. Because of this, we approach this as a classification problem, with 54 categories.

We considered several known machine learning algorithms for classification[2], including decision tree based models like Random Forest, c5.0, and Gradient Boosting Trees.

A simple linear SVM model would work well, if our dataset was linearly separable. We have 54 different categories that would need to be seperated with multiple angles. A polynomial SVM model seemed more appropriate, considering the shape of building 0 (see plot below), however, it is computationally expensive, even with sampling of the data, and it resulted in an accuracy rate of merely 53%. The simple linear SVM model resulted in an accuracy of just 54%. We also tried kNN as it seems natural to classification problems on maps[3] , but unfortunately, it also resulted in a low accuracy rate of 57%.

1. https://www.mist.com/documentation/rssi-values-good-bad-signal-strength/ , https://www.netspotapp.com/what-is-rssi-level.html
2. https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/
3. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/



The decision tree models performed the best, with Random Forest performing the best, followed by a close tie between C5.0 and Gradient Boosted Trees (which was computationally expensive). Random Forest was the quickest and had the highest accuracy rate (83%) out of all our models.

```
Models: svMLinear, svmPolynomial, kNN, C50, gradientBoostMachine, randomForest
Number of resamples: 10

Accuracy
                          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
svMLinear            0.4457831 0.4603619 0.5031635 0.4941268 0.5160188 0.5500000    0
svmPolynomial        0.3974359 0.5118459 0.5216440 0.5189246 0.5443014 0.6049383    0
kNN                  0.4268293 0.4815733 0.5272554 0.5244267 0.5577201 0.6052632    0
C50                  0.6578947 0.7387048 0.7729420 0.7714172 0.8133677 0.8375000    0
gradientBoostMachine 0.7045455 0.7327744 0.7729600 0.7624427 0.7862025 0.8181818    0
randomForest         0.7901235 0.8119919 0.8394817 0.8378592 0.8677792 0.8734177    0

Kappa
                          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
svMLinear            0.4344542 0.4493760 0.4925190 0.4833334 0.5054419 0.5405966    0
svmPolynomial        0.3841760 0.5016610 0.5107822 0.5084807 0.5345261 0.5965131    0
kNN                  0.4156179 0.4708601 0.5171065 0.5145682 0.5488302 0.5968170    0
C50                  0.6503274 0.7330435 0.7679486 0.7664700 0.8093923 0.8339454    0
gradientBoostMachine 0.6986698 0.7267978 0.7679992 0.7572804 0.7817510 0.8141499    0
randomForest         0.7855140 0.8080224 0.8359743 0.8343240 0.8647574 0.8705555    0
```


1. https://www.mist.com/documentation/rssi-values-good-bad-signal-strength/ , https://www.netspotapp.com/what-is-rssi-level.html
2. https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/
3. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Recommendations:

While our random forest model was accurate most of the time (~83%), what happens the other 17% of the time? How off was our model? There are cases where we need very high accuracy and there are cases where it's ok to be a bit off. For the purpose of this project, we recommend that the client pursue other technology for identifying location indoors. The reason being the accuracy rate. The following list shows the area, in meters squared, that the model was off. Perhaps, a user might be forgiving up to around 3 meters in error. However, over half of the incorrect predictions were incorrectly predicted to be over 8 meters squared away from the actual location. Depending on who the user is, these errors can make all the difference in whether this kind of application is considered reliable or not, as it could indicate to the user that they are in a room that they are actually not in.

Distance Predictions are off from Actual Values ((in meters squared)

```
[1]   0.2065879   0.8001706   0.8659967   0.8864866   0.9187557   0.9187557
1.1013802
 [8]   1.2334686   1.2923918   1.2923918   1.2923918   1.4488265   1.9035318
1.9035318
[15]   2.0415561   2.0967828   2.8117302   2.8117302   2.8117302   8.8880880
9.0805415
[22]  10.0353149  10.4079902  12.3987774  14.6271616  14.6271616  15.3946295
15.6515850
[29]  15.6515850  16.3722582  23.5157353  29.1993041  30.7279032  30.9294855
34.1009078
[36]  34.1009078  38.6492872  44.8788157 121.7521110 121.7521110 121.7521110
176.8330607
[43] 194.0056457
```


1. https://www.mist.com/documentation/rssi-values-good-bad-signal-strength/ , https://www.netspotapp.com/what-is-rssi-level.html
2. https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/
3. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/