**C4T3 REPORT INDOOR Wi-Fi FINGERPRINTING or INDOOR LOCATIONING**

For this project, we were asked to look at different algorithms for a “Wi-Fi fingerprinting” dataset to see if we could predict indoor location.  Our client is developing a system to be deployed on large industrial campuses, in shopping malls, et cetera to help people navigate a complex, unfamiliar interior space without getting lost. It is well known in the industry that GPS works somewhat reliably outdoors; the problem is that GPS generally doesn't work properly indoors.  Therefore, a look at different technology is necessary. So, we need to answer the following question:

Can this dataset be used to build a model to predict a person’s location in indoor spaces with sufficient accuracy, so that the model can be incorporated into a smart phone app.

We have been provided with a large database of Wi-Fi fingerprinting for a multi-building industrial campus. Outdoor localization does not face significant problems thanks to GPS sensors in the mobile devices. Since the GPS signals on mobile devices are often lost indoors, indoor localization is facing many problems. The application called CaptureLoc was used to capture all the attributes provided in the dataset. The information was captured by 18 different users with mobile devices in 3 different buildings.  Automatic user localization consists of estimating user position by using a mobile phone.  The following attributes were collected by 18 users in 3 different buildings with their mobile devices by using CaptureLoc:

1. WAPs - Wireless Wi-Fi Point from each localization in each capture. Each WAP represents the raw intensity level from a single Wi-Fi scan. There are 520 attributes of WAPs

2. Building ID - Wi-Fi signals were collected from 3 different buildings. The values of this attribute were from 0 to 2.

3. Floor – indicates on what floor of the building Wi-Fi capture was recorded.

4. Space ID - contains integer value and indicates room number inside the building such as office, labs, classroom, etc.

5. Relative Position – this attribute denotes if the capture of the location was made inside the room (value 1) or outside of the room, more specifically in front of the door (value 0).

6. User ID - contains integer values from 0 to 18, indicating that 18 different users were participating to collect Wi-Fi signals inside the 3 buildings with mobile devices.

7. Phone ID - an integer value representing the type of mobile device that was used to record each Wi-Fi capture.

8. Timestamp - represents the time in UNIX time format in which the capture was taken.

9. Longitude and Latitude - represents real-world coordinates of the floor and building.

For the purpose of our analysis, we got rid of the following 5 attributes: Relative Position, User ID, Phone ID, Timestamp, Longitude and Latitude. These attributes were not relevant in predicting location inside the building using Wi-Fi readings.  To build our predictive models we were working with the following attributes: WAP, Building ID, Floor, Space ID and Relative position.

Taking into consideration that our dataset is very large and RStudio only uses one core from the computer’s processor, we set up parallel processing, which means that we will use additional processor cores to speed up the processing time.

Furthermore, to reduce dimensionality of our data we subset the data to 3 buildings: Building 0, Building 1 and Building 2 Also, using nearZeroVar from caret package, we got rid of the features with no variance since they hold little to no information for model building purposes.

These three pre-processing steps indicated above, helped us to speed up processing time while running our algorithms.

Since the task of our project is to see if we can predict the location based on Wi-Fi readings, we combined FLOOR, SPACEID AND RELATIVEPOSITION in one variable for each building. After subsetting the data and combining in one variable, we converted it to a character format that is appropriate for classification models. With created new variables, we are looking to see with what accuracy our models can predict on what floor and  in what room users were located and whether users were located outside or inside the room.

We set up 3 different models for 3 different buildings and the results of Accuracy and Kappa scores can be found in the table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | C5.0 | | Random Forest | | KNN | |
|  | Accuracy | Kappa | Accuracy | Kappa | Accuracy | Kappa |
| Building 0 | 0.996 | 0.995 | 0.782 | 0.783 | 0.547 | 0.548 |
| Building 1 | 0.686 | 0.684 | 0.856 | 0.855 | 0.632 | 0.631 |
| Building 2 | 0.555 | 0.553 | 0.836 | 0.835 | 0.609 | 0.608 |

From the table above we see that for Building 1 the highest accuracy was obtained from the C5.0 model. Looking across the table, C5.0 is the only one that has such a high score. Random forest stands at .782 for building 0 and it means that out of 10 WIFI readings, in Building0 we can determine location 7 times correctly out of 10. KNN Accuracy for building zero stands at a much lower value of .547.

If we analyze the rest of the table, we can see that the Random Forest algorithm for building 1 has the highest accuracy among all the results, standing at .856. C5.0 and KNN are much lower for Building 1 if to compare with Random Forest.

Looking at the table above, we see that there is no consistency in the outcomes of running our models. We can tell that KNN has the lowest value of all the models, while C5.0 does not have consistency in the final outcomes. Random forest seems to have the highest Accuracy of all the models. While results for Random Forest for all three buildings are the highest ones, they are still far from perfect to use for predicting location using the current dataset. It will be up to the app developer team to decide if the produced results will be sufficient for the current dataset and models be used to build an app.