Many real world applications need to know the localization of a user in the world to provide their services. Therefore, automatic user localization has been a hot research topic in the last years. Automatic user localization consists of estimating the position of the user (latitude, longitude and altitude) by using an electronic device, usually a mobile phone. Outdoor localization problem can be solved very accurately thanks to the inclusion of GPS sensors into the mobile devices. However, indoor localization is still an open problem mainly due to the loss of GPS signal in indoor environments. Although, there are some indoor positioning technologies and methodologies, this dataset is focused on WLAN fingerprint-based ones (also known as WiFi Fingerprinting).

**Goal**

To evaluate the application of machine learning techniques to the problem of indoor locationing via wifi fingerprinting.

**Data Set Information**

* Data from three buildings of Universitat Jaume with 4 or more floors.
* More than 20 different users and 25 Android devices.
* 19937 training/reference records (trainingData.csv file)
* 1111 validation/test records (validationData.csv file).
* Attributes:
  + 520 intensity values
  + Latitude
  + Longitude
  + Floor
  + Building ID
  + Phone ID
  + User ID
  + Space ID
  + Relative Position
  + Timestamp

**Data Preparation**

* Removed the rows with missing/null data.
* Generated target Y label from distinct factor interactions of Floor and Building (FB) for both the files (training and validation data files)
* Stack all WAP (Wireless Access Points) in one column, to allow for transformation of values without looping over column indexes.
* Change the values of 100 RSSI to zero (no signal) and convert the remaining RSSI values to linear scale.
* Drop the zero variances columns in WAP features

**Known issues with the data:**

* Memory issues when trying to train the model using KNN and SVM when using more than 5000 instances.
* ROC-AUC curves do not work with the data set.

**Model Comparison**

**KNN Model**

|  |  |  |
| --- | --- | --- |
| K | Accuracy | Kappa |
| 5 | 0.9924874 | 0.9847971 |

**SVM Model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cost** | **Loss** | **Accuracy** | **Kappa** |
| **2.00** | **L1** | **0.9929998** | **0.9858851** |

**Random Forest Model**

|  |  |  |
| --- | --- | --- |
| **Mtry** | **Accuracy** | **Kappa** |
| **2** | 1.0000000 | 1.0000000 |

**C5.0 Model**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Winnow** | **Trials** | **Accuracy** | **Kappa** |
| **Rules** | **FALSE** | **10** | **0.9980198** | **0.9960792** |

**Rpart Model**

|  |  |  |
| --- | --- | --- |
| **Cp** | **Accuracy** | **Kappa** |
| **0.0000** | **0.9970099** | **0.9939875** |



**Recommendation of the algorithm to be best for this data and its justification**

|  |  |
| --- | --- |
| Model | Accuracy |
| KNN | 0.9924874 |
| SVM | 0.9929998 |
| RF | 1.0000000 |
| C50 | 0.9980198 |
| CART | 0.9970099 |



**Random Forest Model**

Reference

Prediction 0-0 2-1 2-2 3-2

       0-0   0   0   0   0

       2-1   0 352   0   0

       2-2   0   0   0   0

       3-2   0   0   0 445

 Accuracy (average) : 1

Each column holds the *reference* (or actual) data and within each row is the prediction. The diagonal represents instances where the observation correctly predicted the class of the item

Based on the above results, Random Forest Model worked very well with this data. There were no memory issues while training the model compared to KNN and SVM