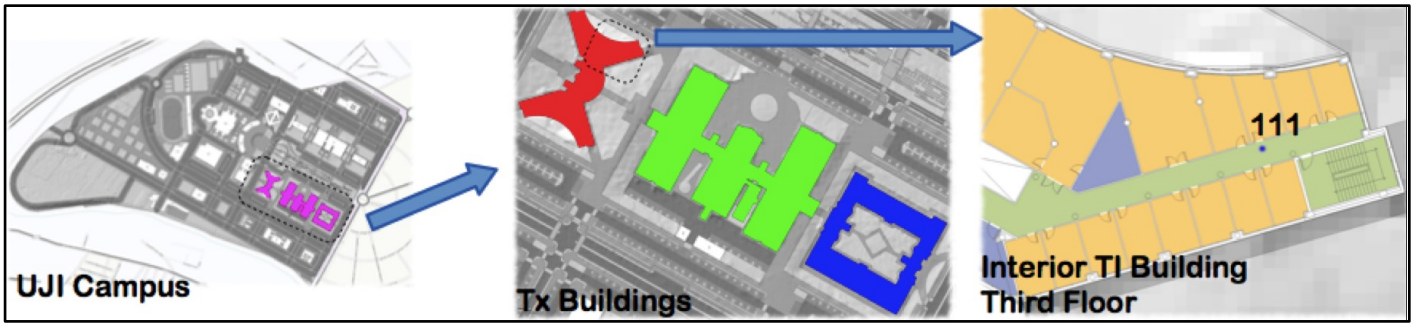
Wi-Fi Locationing Project Report
Greg Hepguler

Many applications need to know the localization of a user in the world to provide their services. Automatic user localization consists of estimating the position of a person by using an electronic device, usually a mobile phone. Outdoor localization problem can be solved very accurately as a result of the inclusion of GPS sensors into the mobile devices. However, indoor localization is a different issue mainly because of the loss of GPS signal in indoor environments.

Indoor Positioning System aims at locating objects inside buildings wirelessly, and have huge benefit for indoor location-aware mobile applications. In this project, we intend to use the fingerprint of Web Access Points (WAPs) as features to predict the position of a mobile device holder. The fingerprint of WAP is the Received Signal Strength Indicator (RSSI). We locate the building and floor level of a mobile device via machine learning methods using Wi-Fi fingerprint. We explore the data size, data features, classification models combined with parameter selection to accomplish our goal.

Our business objective is to develop a system that will help people navigate large, complex, unfamiliar interior spaces, such as buildings in large industrial campuses, business complex, commercial property or shopping malls without getting lost. We Investigate feasibility of using Wi-Fi fingerprinting to benchmark activity at The Universitat Jaume I (UJI) so that we can compare that activity to other locations to assess location-based marketing. We select a relevant dataset to perform data analytics, representative of required location-based solutions for malls, convention centers or a large business complex. We recommend the best model that produces best metrics for dependable results.

The Universitat Jaume I (UJI) is a public university in the northern part of the Valencian Community, a region on the European Mediterranean coast located among the cities of Valencia, Barcelona and Madrid. Established in 1991, the UJI has 15,000 students in its integrated, modern, functional and sustainable campus. The development of the UJI has been marked by advances in information and communication technologies.

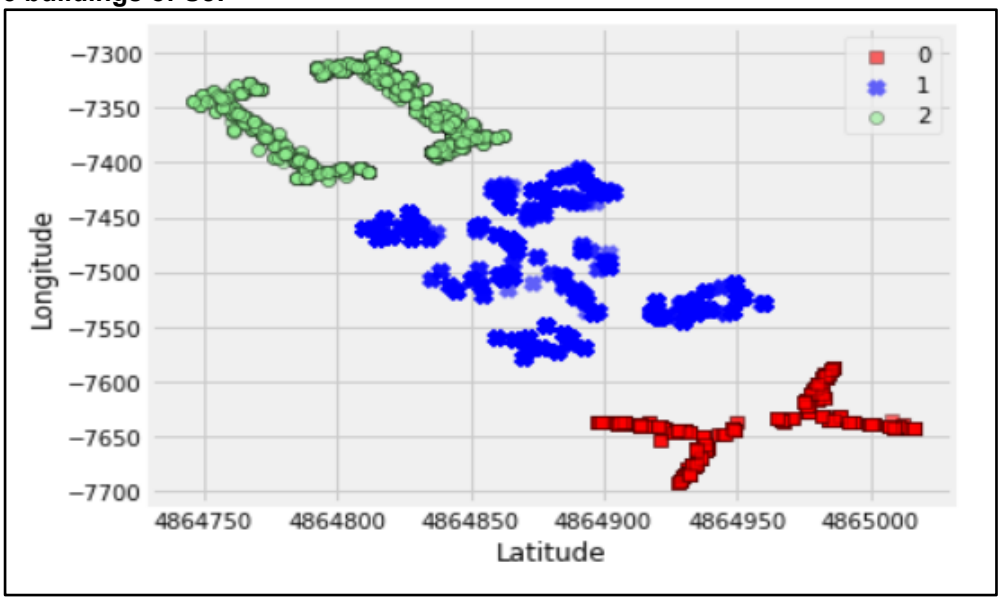


Our data analytics objective is to develop a Wi-Fi Fingerprinting model to locate people indoors in 3 large buildings in the campus of UJI, and evaluate the data analytics models by predicting a person's location. We use Wi-Fi fingerprint characterized by WAPs and the corresponding RSSI, compare models produced by three different algorithms, determine the algorithm that is best suited for this data, based on specific metrics as a justification of why it is the preferred choice.

To explore our solution, we choose UJI IndoorLoc database as our dataset, and build classification models based on K-Nearest Neighbor kNN, Support Vector machine SVM, and Random Forest RF. UJIIndoorLoc database is a common database in trying to solve the indoor localization problems using a Wi-Fi fingerprint-based indoor localization.

The UJIIndoorLoc database covers 3 buildings of UJI with 4 or more floors and almost 110,000 m^2. It can be used for classification, e.g. actual building and floor identification, or regression, e.g. actual longitude and latitude estimation. It was created in 2013 by means of more than 18 different users and 24 Android devices. The database consists of 19937 training/reference records (trainingData.csv file) and 1111 validation/test records (validationData.csv file). The dataset has 529 attributes, contains the Wi-Fi fingerprint, the coordinates where it was taken, building identification, floor & space identification and other useful information.

3 buildings of UJI



After defining the business objectives and translating those objectives into data analytics goals, we complete Exploratory Data Analysis, EDA by fully exploring the historical data at hand.

Each WiFi fingerprint is characterized by the detected Wireless Access Points (WAPs) and the corresponding Received Signal Strength Intensity (RSSI). The intensity values are represented as negative integer values ranging from -104dBm (extremely poor signal) to 0dbM. The positive value 100 is used to denote when a WAP was not detected. During the database creation, 520 different WAPs were detected. Thus, the WiFi fingerprint is composed of 520 intensity values.

Description of the Indoor Location Dataset

- Attribute 001 (WAP001) through Attribute 520 (WAP520): Intensity value for WAP001 through WAP520. Negative integer values from -104 to 0. The value +100 is used when WAP001 was not detected.
- Attribute 521, LONGITUDE : Negative real values from -7695.93875493 to -7299.78651673.
- Attribute 522, LATITUDE : Positive real values from 4864745.7450159714 to 4865017.3646842018.
- Attribute 523, FLOOR : Floors inside the building. Integer values from 0 to 4.
- Attribute 524, BUILDINGID : ID to identify the building. Integer values from 0 to 2.
- Attribute 525, SPACEID : Internal ID number to identify the Space, indicating office, corridor or classroom.
- Attribute 526, RELATIVEPOSITION : Relative position with respect to the Space, 1- Inside, 2- Outside.
- Attribute 527, USERID : User identifier. Categorical 3-digit integer values between 158 and 186.
- Attribute 528, PHONEID : Android device identifier. Categorical integer values.
- Attribute 529, TIMESTAMP : UNIX Time when the capture was taken. Integer value.

The particular space (offices, labs, etc.) and the relative position (inside/outside the space) where the capture was taken have been recorded. Outside means that the capture was taken in front of the door of the space.

Independent variables or features are Attribute 001 (WAP001) through Attribute 520 (WAP520). The attributes to be predicted (dependent variables) can be Building ID, Floor ID and the location coordinates latitude & longitude. We will define a unique code as dependent variable representing a unique LOCATION.

We have a Training dataset and Test (Validation) dataset. In the EDA process, we view descriptive statistics of both datasets such as minimum, maximum, mean values of each attribute, check for missing or 'NaN' values, check for duplicate rows and check for out of range values. In this project, we could define WAP value +100 as 'NaN' and remove those columns that have only 'NaN' values from the dataset. We chose not to remove any columns with undetected signals because we want to keep the original dataset dimensions as we take random sample (generally 20%) of the dataset for modeling.

Training Dataset

1		*520*	*521*	*522*	*523*	*524*	*525*	*526*	*527*	*528*	*529*
WAP001	**...**	**WAP520**	**LONGITUDE**	**LATITUDE**	**FLOOR**	**BUILDINGID**	**SPACEID**	**RELATIVEPOSITION**	**USERID**	**PHONEID**	**TIMESTAMP**
100	...	100	-7541.2643	4.864921E+06	2	1	106	2	2	23	1371713733
100	...	100	-7536.6212	4.864934E+06	2	1	106	2	2	23	1371713691
100	...	100	-7519.1524	4.864950E+06	2	1	103	2	2	23	1371714095
100	...	100	-7524.5704	4.864934E+06	2	1	102	2	2	23	1371713807
100	...	100	-7632.1436	4.864982E+06	0	0	122	2	11	13	1369909710

Training Dataset Descriptive Statistics

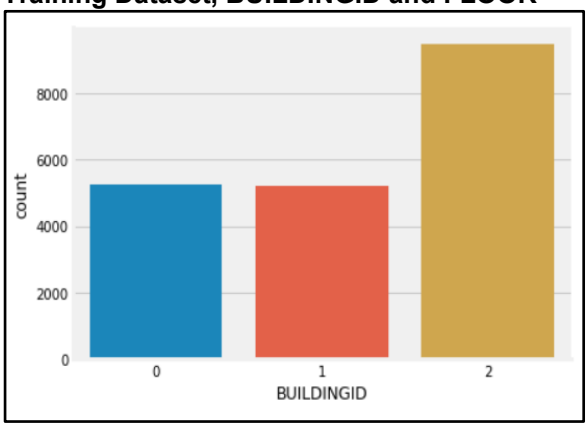
	WAP001	**...**	**WAP520**	**LONGITUDE**	**LATITUDE**	**FLOOR**	**BUILDINGID**	**SPACEID**	**RELATIVE POSITION**	**USERID**	**PHONEID**	**TIMESTAMP**
count	19937	...	19937	19937	1.99370E+04	19937	19937	19937	19937	19937	19937	1.99370E+04
mean	99.82364	...	100	-7464.27595	4.86487E+06	1.67458	1.21282	148.42995	1.833024	9.06801	13.02187	1.37142E+09
std	5.866842	...	0	123.40201	6.69332E+01	1.22308	0.833139	58.342106	0.372964	4.98872	5.36241	5.57205E+05
min	-97	...	100	-7691.3384	4.86475E+06	0	0	1	1	1	1	1.36991E+09
25%	100	...	100	-7594.737	4.86482E+06	1	0	110	2	5	8	1.37106E+09
50%	100	...	100	-7423.0609	4.86485E+06	2	1	129	2	11	13	1.37172E+09
75%	100	...	100	-7359.193	4.86493E+06	3	2	207	2	13	14	1.37172E+09
max	100	...	100	-7300.81899	4.86502E+06	4	2	254	2	18	24	1.37174E+09

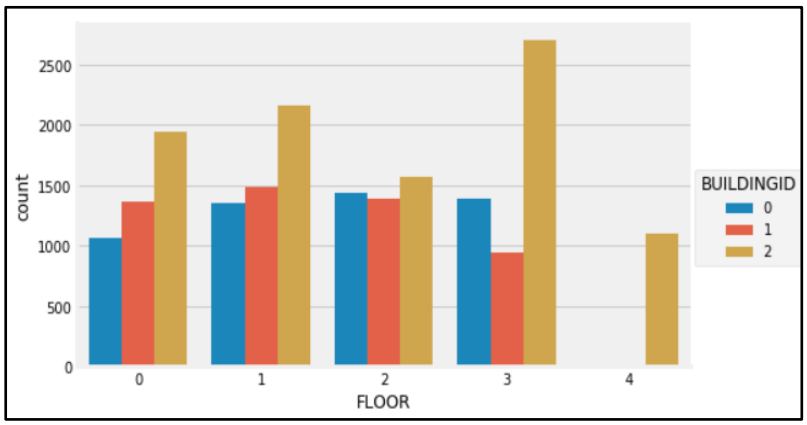
In summary:

- Dataset has 19937 observations of 529 variables
- Checked for NA values, none found
- USERID is an anonymized user height (cm)
- PHONEID is an Android device type that includes GT, Galaxy Nexus, HTC, LT
- Details for USERID and PHONEID are provided at UCI Machine Learning Repository
- UCI web address is: https://archive.ics.uci.edu/ml/datasets/ujiindoorloc

After preparing and reviewing the dataset, we visualize the data through histograms. The datasets reference 3 buildings. Buildings 0 and 1 have four floors whereas building 2 has 5 floors including the ground level. We have more data points, almost twice as much from building 2, which is the Science and Technology building.

Training Dataset, BUILDINGID and FLOOR

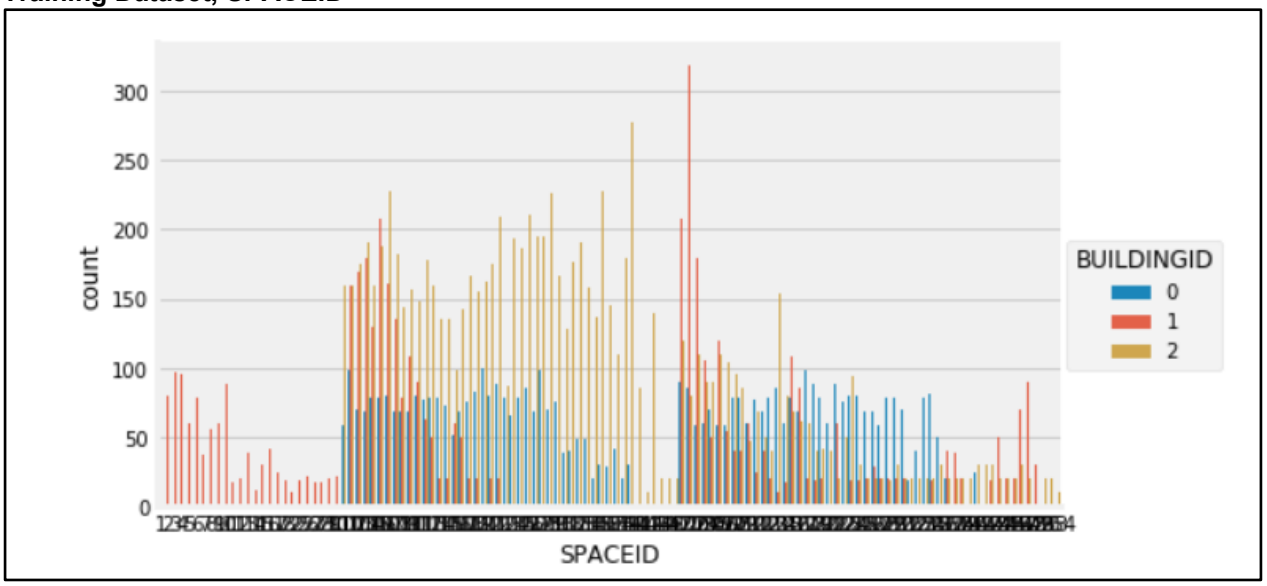




Data Types:

- BUILDINGID, FLOOR, SPACEID, USERID, PHONEID, RELATIVEPOSITION are defined as integer
- LONGITUDE, LATITUDE are defined as float
- TIMESTAMP is defined as POSIXct in UTC time zone

Training Dataset, SPACEID



We complete similar investigation of the test dataset. Test Dataset has 1111 observations of 529 variables. We confirm that this data set is consistent with the Training Dataset. SPACEID that identifies office, corridor or classroom was not populated in the Test Dataset.

Test (Validation) Dataset Descriptive Statistics

	WAP001	...	WAP520	LONGITUDE	LATITUDE	FLOOR	BUILDINGID	SPACEID	RELATIVEPOSITION	USERID	PHONEID	TIMESTAMP
count	1111	...	1111	1111	1111	1111	1111	1111	1111	1111	1111	1.11100E+03
mean	98.627363	...	99.843384	-7529.197448	4.86490E+06	1.57	0.76	0	0	0	11.92	1.38060E+09
std	16.127245	...	5.220261	120.209336	7.02728E+01	1.00	0.82	0	0	0	6.56	5.00322E+05
min	-94	...	-74	-7695.938755	4.86475E+06	0	0	0	0	0	0	1.37958E+09
25%	100	...	100	-7637.4238	4.86484E+06	1	0	0	0	0	9	1.38019E+09
50%	100	...	100	-7560.3763	4.86492E+06	1	1	0	0	0	13	1.38087E+09
75%	100	...	100	-7420.539659	4.86497E+06	2	1	0	0	0	15	1.38088E+09
max	100	...	100	-7299.786517	4.86502E+06	4	2	0	0	0	21	1.38125E+09

In EDA, the covariance is normalized using standard deviation to a score between -1 and 1, to make its magnitude interpretable, and the result provides a correlation matrix CM of the variables in a dataset. The correlation matrix shows the highly-correlated features, where correlation coefficient is greater than a pre-selected threshold. In this dataset we use 520 WAP signals from different persons at different locations as independent variables. We could not remove any features based on the CM.

Define LOCATION

Before we continue on to modeling, we must define our dependent variable LOCATION which represents BUILDING ID and FLOOR ID in addition to LATITUDE and LONGITUDE to specify a unique location. We defined LOCATION as:

LOCATION = (BUILDINGID+1)*10000 + (FLOOR+1)*1000 + integer(LATITUDE / (-LONGITUDE))

This equation provides a 5-digit number where the first digit is a Building identification, second digit is a Floor identification and the last 3 digits are an integer value given as the ratio of Latitude to Longitude at a particular location. For example:

LONGITUDE	LATITUDE	BUILDINGID	FLOOR	LAT/LONG Ratio	LOCATION
-7414.873	4864881.277	1	2	656	23656

We should note that it was not possible to include SPACEID to predict LOCATION because SPACEID in Test dataset has not been populated and has all zero values.

The concept of using the ratio of latitude to longitude originates from trigonometry, in formulas that are used for calculating angular distance as the difference in distance along a parallel of latitude corresponding to a difference in longitude and vice versa.

Modeling

We start the modeling process by running the chosen algorithms out of the box – using the default parameters, without any tuning. We chose 3 different machine learning algorithms for this classification problem:

- kNN, KNeighborsClassifier.
- Support Vector Machine, SVM C-Support Vector Classification, SVC.
- Random Forest Classifier, RFC.

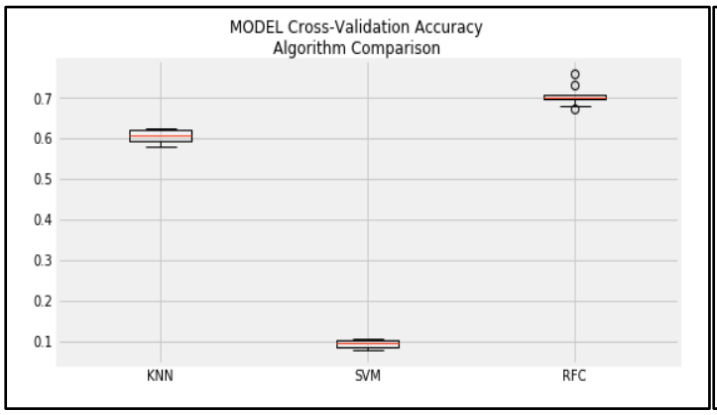
Default parameters used with each algorithm (out of the box) are listed in Table 1.

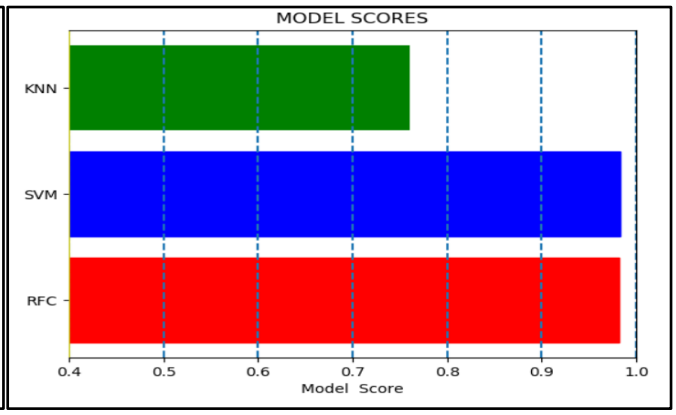
Table 1

Classifier	Run Type	Model Parameter 1	Model Parameter 2	Model CV Accuracy	Model Score	Prediction Accuracy	Prediction Kappa
kNN	Out of box	n_neighbors=5	weights=uniform	60.66	76.10	33.66	32.70
SVM	Out of box	C=1.0	kernel=rbf	9.55	98.37	2.52	0.00
RF	Out of box	n_estimators=10	max_features=auto	70.60	98.14	40.41	39.49

The choice of the "best model" depends on the performance metrics that we gather after running each model using the Training dataset. In this project, the initial set of performance metrics for the out-of-box evaluation include Model Cross Validation (CV) accuracy and Prediction Accuracy. We also observe Kappa while checking the accuracy.

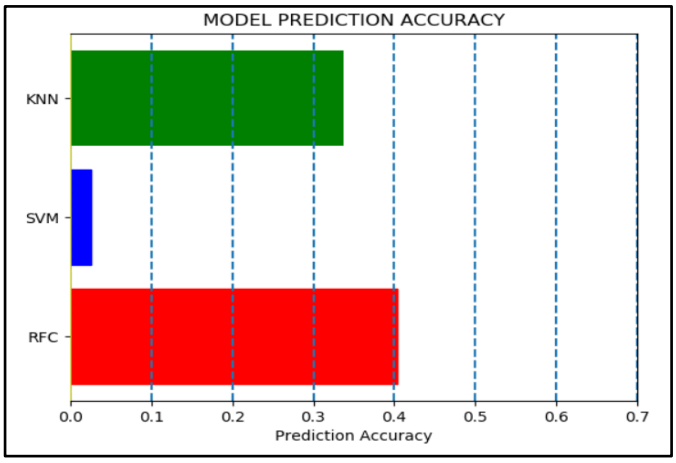
We used kNN, SVC and RFC *out of the box* on the ***Training Dataset*** to see the CV scores.

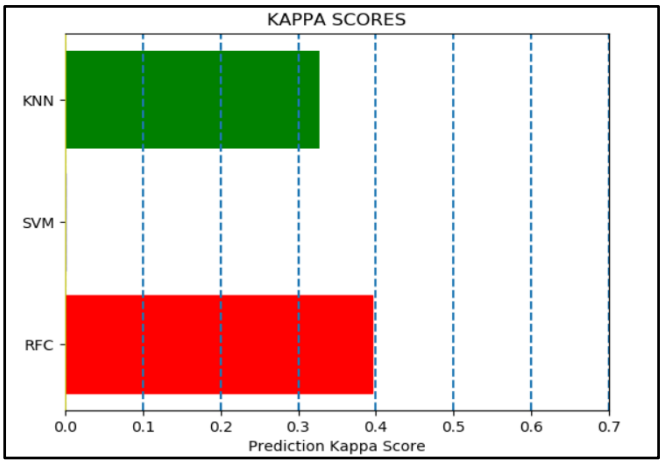




Cross Validation, CV estimates the performance of a machine learning algorithm by splitting the dataset into n parts where n might be equal to 5 or 10 splits. Each split of the dataset is called a fold. Each algorithm that we selected is trained on (n -1) folds with one fold held back. The algorithm is then tested on the held-back fold. This procedure is repeated so that each fold of the dataset is given a chance to be held back as the test set. CV provides n different performance scores which are summarized as mean CV-score and a standard deviation. This CV-score is a reliable estimate of the performance of the algorithm on new data because the algorithm is trained and evaluated multiple times on different set of data.

Prediction Accuracy is calculated as the proportion of predictions that represent the number of true positives and true negatives, divided by the total number of predictions. In other words, it measures correct predictions relative to total predictions. Kappa adjusts Accuracy by accounting for the possibility of correct prediction by chance alone.





Tuning the models

We tune all of the models by varying the most significant 2 parameters for each algorithm. These parameters for each algorithm are

- kNN : 1. K-value specified as n neighbors from 1 to 200 2. weights specified as uniform or distance
- SVM : 1. C-values ranging from 0.5 to 1.5 2. Kernel specified as rbf,
- RFC : 1. N estimators specified as n values from 2 to 200 2. Max_feature specified as auto or log2

Each algorithm has many other options for parameter specifications. Default values are accepted for those other parameters.

In order to find the best model, we tune each model using the Training dataset and search for the maximum CV-score per model. Models are tuned using a wide range of values specified for the first and second parameters. For example, kNN Classifier model runs 20 times using 10 different "n_neighbors" values and 4 different kernel types. RFC runs 24 times with associated parameters to get the best CV score and so forth. We recorded the mean CV score for each model.

We select the best model with the algorithm that yields the highest mean CV score. Those scores from tuned models are shown in Table 2.

Table 2

Classifier	Run Type	Model Parameter 1 for Best CV score	Model Parameter 2 for Best CV score	Best CV score
kNN	Tuning	n_neighbors=1	weights=uniform	0.6670
SVM	Tuning	C=1.5	kernel = rbf	0.1039
RF	Tuning	n_estimators=150	max_features=auto	0.7906

In this project, Random Forest Classifier with the parameters shown in Table 2 is selected as the best model. We use this RFC model in our predictions.

Prediction

Using RF Classifier and the Test (Validation) Dataset, we predict LOCATION based on the WAP signals listed on the Test dataset. Predicted LOCATION values are compared to the Actual LOCATION values in the Test dataset to see Prediction Accuracy and Kappa.

We specify kfold = 5 when we run cross validation using RFC in the prediction phase using the Test dataset. The Cross Validation Scores from each fold are:

[0.7656051 0.81146497 0.80254777 0.78471338 0.78853503]

Cross Validation Accuracy equals 79.06 % which is the average of the 5 CV sores.

The performance metrics indicate that Prediction Accuracy is 50% with the corresponding Kappa score 49% indicating Moderate Agreement.

The results are appended into the Test Dataset file where we can observe Actual LOCATION and Predicted LOCATION along with all of the independent variables, i.e. the 520 WAP values.

WAP001	WAP002	WAP003	WAP004	...	LATITUDE	FLOOR	BUILDINGID	SPACEID	...	PHONEID	TIMESTAMP	LOCATION	PREDICTION
100	100	100	100	...	4.864890E+06	1	1	0	...	0	1380872703	22647	31656
100	100	100	100	...	4.864840E+06	4	2	0	...	13	1381155054	35658	35658
100	100	100	100	...	4.864847E+06	4	2	0	...	13	1381155095	35659	35660
100	100	100	100	...	4.864843E+06	4	2	0	...	13	1381155138	35660	35661
100	100	100	100	...	4.864922E+06	2	0	0	...	2	1380877774	13636	13636

In addition to Accuracy and Kappa, the prediction of LOCATION using Test dataset also provides a Classification Report listing Precision, Recall, F1 score and Support for each LOCATION prediction.

Classification Report

	precision	recall	f1-score	support
11632	0.67	0.4	0.5	5
11633	0.75	0.75	0.75	8
11634	0.6	0.6	0.6	5
11635	1	0.33	0.5	3
11636	0.55	0.72	0.63	29

.....

Precision is the number of True Positives TP divided by the number of True Positives TP and False Positives FP. It is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV). The precision is the ratio TP / (TP + FP). The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives FN. It is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate. The recall is the ratio TP / (TP + FN). Recall is intuitively the ability of the classifier to find all the positive samples.

		Predictions	
		no	yes
Actuals	no	TN	FP
	yes	FN	TP

	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

The F1 Score equals 2*((precision*recall)/(precision+recall)). It is also called the F Score or the F Measure. The F1 score conveys the balance between the precision and the recall. The F1 score can be interpreted as a weighted harmonic mean of the precision and recall, where it reaches its best value at 1 and worst score at 0.

The Support is the number of occurrences of each class. In this project, it is the number of occurrences related to the prediction of each LOCATION.

Conclusions and Recommendations

The prediction results are appended to the Test dataset in the output file: C5T4_CapstoneResults_GregHepguler.csv This file has the actual LOCATION value as we defined it prior to modeling and the predicted LOCATION value in the adjacent column.

We also create a LOCATION index file using the Training Dataset. This file is available for reference in order to resolve the predicted Latitude and Longitude from the last 3 digits of the 5-digit LOCATION indicator.

Random Forest classifier is the best model for training the model and predicting location. kNN classifier performed reasonably well, but SVM performed poorly in this classification project.

We developed a useful model that can help people navigate large, complex, unfamiliar interior spaces. We showed how this model worked with the dataset acquired at The Universitat Jaume I (UJI) campus. The UJIIndoorLoc database provided 520 RSSI fingerprints detected from 24 different Android phone versions and 18 users. At the time the data was acquired, the UJI campus was 110,000 m^2. This is an area larger than 20 football fields combined. We showed that our Wi-Fi fingerprinting model can locate 1 out of 2 people indoors in one of 3 buildings with 4 or 5 floors at some office, classroom or corridor.

Recommendation is to promote this Wi-Fi locationing system since many businesses depend on where people gather up, and location-based marketing can be essential to their business. The data acquisition for the indoor positioning systems must include permission of the person who owns the device.

Results and solution procedure could be improved by reducing the dimension of the problem, for example, predicting LOCATION by taking into account the WAP with the highest RSSI.

References

Jason Brownlee, Machine Learning Mastery With Python, Edition: v1.4, 2016.

Dan Li, Le Wang, Shiqi Wu: “Indoor Positioning System Using Wifi Fingerprint”, Stanford University, 2014.