Wireless Indoor Localization

Machine Learning Engineer

Nanodegree

Capstone project

Amr Ramadan

February 2 , 2019

I. Definition

Problem Overview

Localization is heterogeneous part of the field wireless communication networks that play a vital role in modern life. It's a technique to determine the position of an object or a person. Indoor localization system is a system that attempts to find the accurate position of the person and object inside a building , mall , rooms… etc. The localization systems try to identify the position of moving devices with the help of navigation, tracking, monitoring. Indoor environments are complex, because of signal inference and reflection inside building, it's highly depends on the environment such as position of object and behavior of person, also indoor communication like is unreliable [1].

Problem Statement

Predict the room location by observing signal strengths of seven WIFI signals.

To classify locations, we need to use machine learning model that can predict the location based on the 7 WIFI signals[2].

Machine learning algorithms used to solve this problem:

- **Support Vector Machine:** is an algorithm that outputs an optimal hyperplane that separate the plotted datapoints[3].
- **Random Forest:** is an ensemble classifier that creates a set of decision trees from randomly selected subset of training set, it then aggregates the votes from different decision trees to decide the final class of the test object[4].
- **Adaboost:** like Random Forest classifier is another ensemble classifier that combines weak classifier algorithm to form strong classifier[5].

Metrics

There are two main performance measure that are used for this dataset:

- Testing accuracy
- Type I error (false positive)

Testing accuracy will ensure that future observation will be classified correctly, and type I error in our case is classifying wrong location as found (Correct) location. The chosen model is the model with the highest test accuracy and with the least type I error .

TP : true positive : the location is correct, and the model classified it as correct.

TN : true negative : the location is wrong , and the model classified it as wrong.

FP: false positive : the location is correct, but the model classified it wrong .

FN : false negative : the location is wrong , but the model classified it as correct.

- **Accuracy** : is the most widely used performance metric in binary classification problems.

  $$Accuracy = \frac{\text{Nmuber of Correct Classifications}}{Total\ number\ of\ classifications\ made} = \frac{\text{TP+TN}}{TP+TN+FP+FN}$$

- **Type I error** : it’s the number of data points classified as positive while they are negative in reality

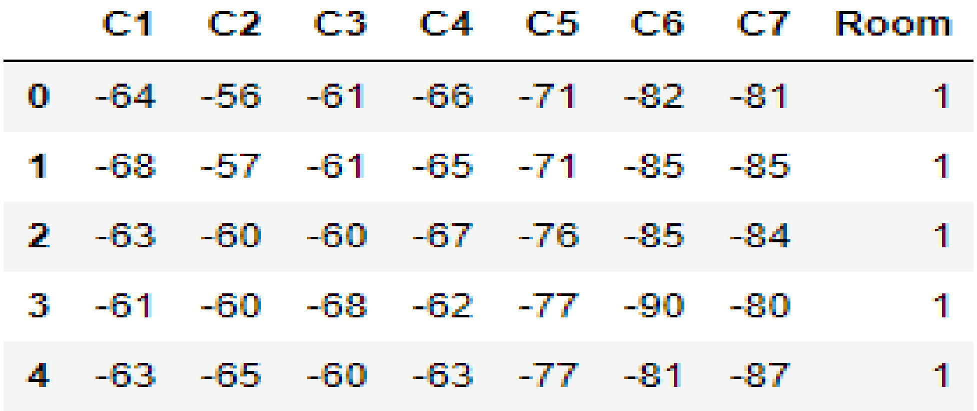
  $$Type\ I\ error = Number\ fo\ False\ Positives$$

II. Analysis

Data Exploration

The dataset used for this project is “Wireless Indoor Localization” from “ UCL machine learning Repository” it was published in December 2017 , it has 2000 rows and 8 columns

- First 5 rows in the datasets :

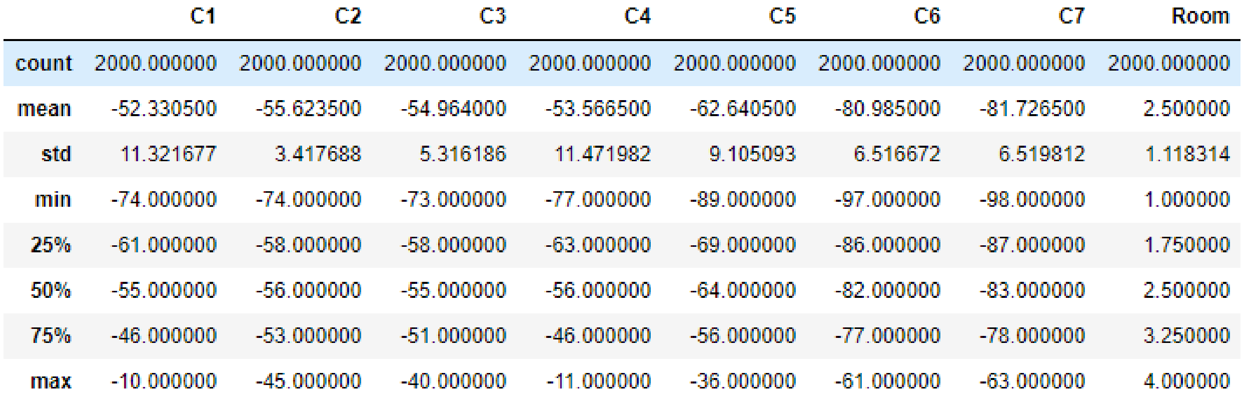


	C1	C2	C3	C4	C5	C6	C7	Room
0	-64	-56	-61	-66	-71	-82	-81	1
1	-68	-57	-61	-65	-71	-85	-85	1
2	-63	-60	-60	-67	-76	-85	-84	1
3	-61	-60	-68	-62	-77	-90	-80	1
4	-63	-65	-60	-63	-77	-81	-87	1

Figure 1 : First 5 Rows in the dataset

- Each row represents the WIFI’s signal strengths that refers to room number.
- All columns contain numerical values and our target value is the last column (room).
- We have four locations/rooms ,that each room is detected 500 times with the seven WIFI signals
- That data not have any null information.

- there is descriptive statistics that describe the features of the dataset :



	C1	C2	C3	C4	C5	C6	C7	Room
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	-52.330500	-55.623500	-54.964000	-53.566500	-62.640500	-80.985000	-81.726500	2.500000
std	11.321677	3.417688	5.316186	11.471982	9.105093	6.516672	6.519812	1.118314
min	-74.000000	-74.000000	-73.000000	-77.000000	-89.000000	-97.000000	-98.000000	1.000000
25%	-61.000000	-58.000000	-58.000000	-63.000000	-69.000000	-86.000000	-87.000000	1.750000
50%	-55.000000	-56.000000	-55.000000	-56.000000	-64.000000	-82.000000	-83.000000	2.500000
75%	-46.000000	-53.000000	-51.000000	-46.000000	-56.000000	-77.000000	-78.000000	3.250000
max	-10.000000	-45.000000	-40.000000	-11.000000	-36.000000	-61.000000	-63.000000	4.000000

Exploratory Visualization

To easy understand the relationship between features and detect outliers , a scatter matrix and pair plot were used because it simple and gives as a natural visualization about data.

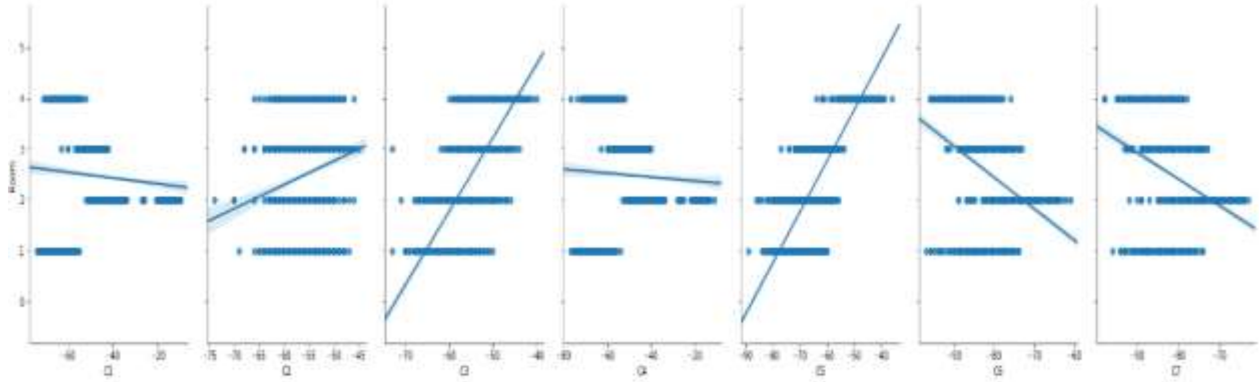


Figure 2 : WIFI’s Shapes

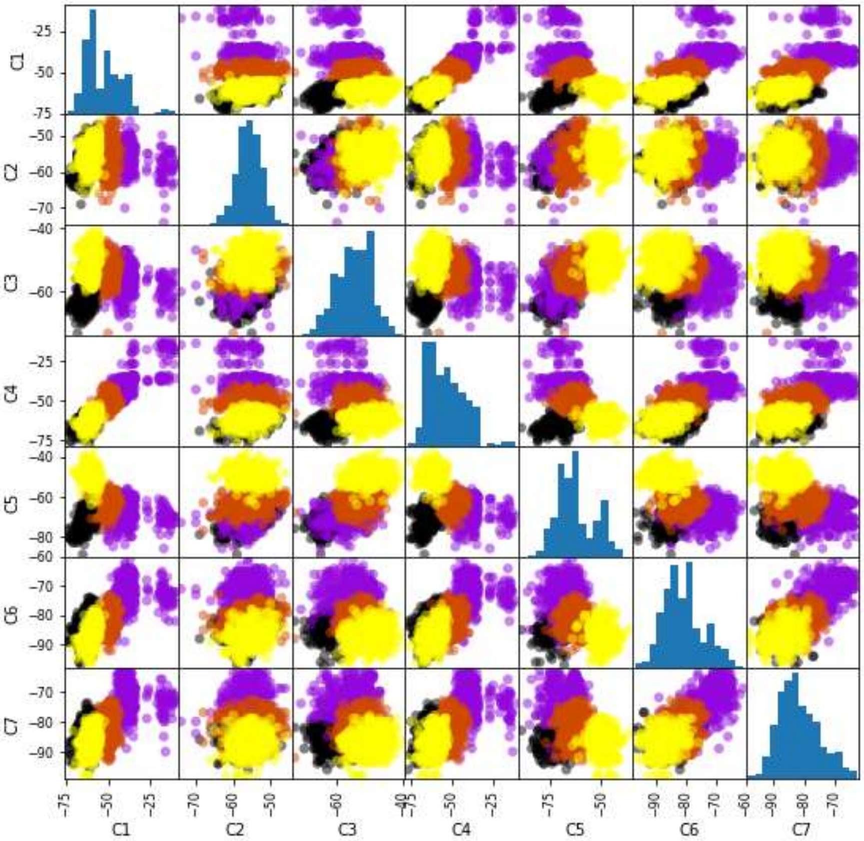


Figure 3 : WIFI’s scatter matrix

According to the Figure 3 scatter matrix above, it seems that we have a small number of outliers that won't affect the classification algorithms, also it seems that all features have a huge relationship that means that we can’t remove any WIFI signal from the Datasets.

Algorithms and Techniques

Wireless Indoor Localization Classification problem can be solved with supervised machine learning models, since all the data points are already labeled, The Three algorithms that I used to solve this problem are: “Adaboost , Support Vector Machine , Random Forest “

‘Adaboost’: is a supervised ensemble classifier that’s usually used fir binary classification, it uses the boosting ensemble method , in fact is was the first proposed method to use boosting. Boosting is a general ensemble method that creates a strong classifier from a number of weak classifiers. It focuses on classification problems and aims to convert a set of weak classifiers into a strong one. This done by building a model from the raining data, then creating a second model that attempts to correct the errors from the first model , Models are added until the training set is predicted perfectly, or maximum number of models are added.

‘Support Vector Machine‘: is a supervised machine learning algorithm which can be used for both classification in our case . The idea of SVM is trying to find a hyperplane that best divides a dataset into two classes. We can think of a hyperplane as line that linearly separates and classifiers a set of data. Choosing that right hyperplane that can separate the two classes is based on the distance between the hyperplane and the nearest data point from either set which is know as the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a great change of new data being classified correctly .

‘Random Forest’ : which is an ensemble classifier that creates many decision trees and form randomly selected subset of training set. The nominated output class is the result of aggregated votes viewed by taking the mode of the classes output by individual trees because a single decision tree may be prone to a noise, but aggregate of many decision trees reduce the effect of noise giving more accurate results which is why I closed this classifier along with it’s ability to fight overfitting (compared to basic decision trees by using bootstrap aggregation or bagging which reduces variance. The model was provided with ‘ random_state’ value to maintain consistency .

Benchmark

The chosen benchmark model for this project is simple “ Logistic Regression” classifier. Also, the reason for choosing this specific algorithm is because it is simple and straight forward and doesn’t require any hyper-parameters to specify beforehand, and it is so popular for binary classification problems that almost every machine learning engineer has used before. Also “ Logistic Regression “ the most commonly reported data science mothed used at work for all industries expect military and security where neural networks are used slightly more frequently according to “ Kaggle : The state of ML and Data Science 2017” .

III. Methodology

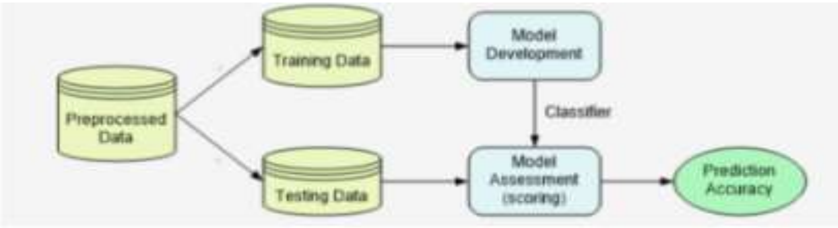
Data Preprocessing

This dataset approximately doesn't need preprocessing,

- just splitting the dataset into training and testing sets with ration of 25%.

Implementation

- The dataset is divided into 75% for training and 25% for testing data .
- Built a benchmark model which is a logistic regression model, it has been trained on all of the training set and it got a type I error of 17 that is even without cross validation.
- That main matric that I evaluate the model on, is the type I error .
- After I have implemented these three algorithms (Random forest, SVM , Adaboost) without any modification in the default's parameters.
- The optimal and best accuracy for the 3 algorithms was with the random state = 2 in the dataset division section.
- The last step was to choose the model with least type I error and further optimized using grid search cross validation.



Complication:

The implementation was smoothly because data was cleaned, and the problem was strait forward without any complications.

Refinement

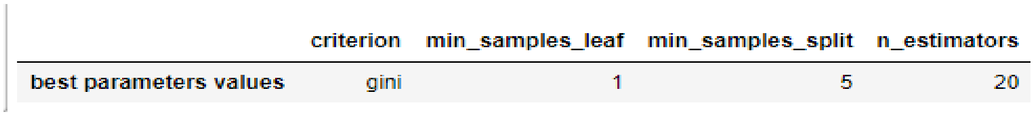
After training and evaluating different algorithms with their default parameters the model with the least Type I error was chosen for further turning and optimization[7].

Model	Type I error
Random Forest	10
SVM	135
Adaboost	23

The random forest classifier needs to be refining to improve its accuracy . hyper-parameter turning was used to find out if there is better collection of parameters that may passed to the model and enhance the result and to find the feature importance.

- ‘n_estimators’ : the number of trees in the forest.
- ‘criterion’ : the function to measure the quality of a split
- ‘min_samples_split’: the minimum number of samples required to split an internal node.

- Min_samples_leaf' : the minimum number of samples required to be at a leaf no.



	criterion	min_samples_leaf	min_samples_split	n_estimators
best parameters values	gini	1	5	20

Figure 4 : grid search best parameters.

It seems that default classifier parameters achieve better performance

IV. Results

Model Evaluation and Validation

All of the models have been evaluated and validated using the accuracy and type I error

The Best performing model in terms of the type I error is Random forest

The model can be considered robust because it is main performance rate (accuracy was measured with two methods, train_test_split and cross validation. using train_test_split method with 'RandomForest' output perfect scores with 98% accuracy, but this percentage cannot be trusted 100% because tuning the model with different random_States will some time change the percentage, and it depends heavily on how the data was splatted at first (especially when using validation technique , which ensures that all the data points will be treated as training or testing data point at some point of time, so by using this technique we trained the model several times and averaged the scores to be able to trust the final result and consider the model as robust after knowing its actual performance and output

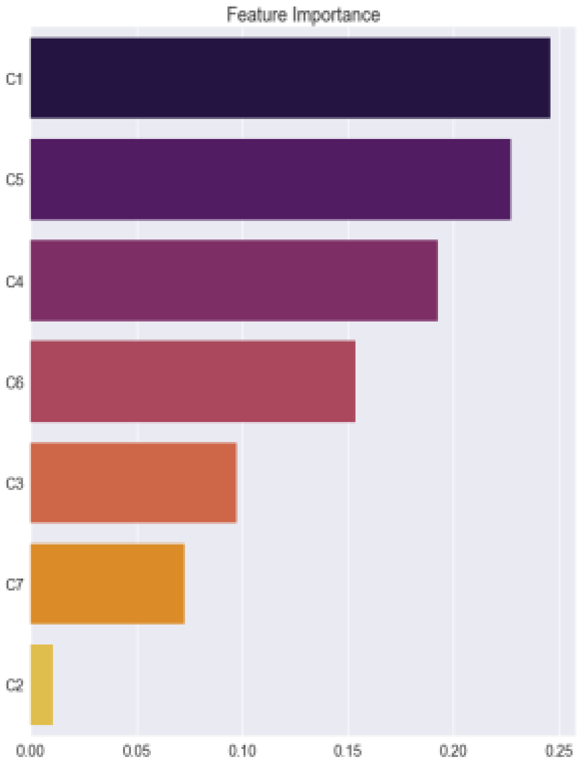
Justification

My benchmark model is a plain logistic regression model that has a type I error of 17and accuracy 96% , while it has a n accuracy lower than the proposed Random forest by 1.4 % .

V. Conclusion

Free-form Visualization

Plotting Random forest feature importance.



It seems that all features are important for Random forest.

Signals that are the strongest for the most common classes end up having the highest feature importance.

Reflection

The process used for this project can be summarized using the following steps:

- Getting the datasets
- Exploratory data analysis
- Visual data exploration
- Building benchmark model and evaluate it
- Building proposed model and evaluate it
- Model validation
- Model optimization and hyper-parameter turning

Our problem was about identifying room number using WIFI's signals, the problem was obtained from 'UCL Machine Learning Repository' It is a Numerical data with 8 columns and 2000 rows. The 'Room' Column was our target variable. First: we started with data exploration to find the relationship of each column and then we plotted every column against the target variable to see if we can predict the behavior of the model. Second, we created the benchmark model (LogisticRegressin) . and we build 3 other models (Random-forest , Adaboost , SVM) to make a comparison between al of them. LogisticRegressin scored very well in terms of accuracy, but Random Forest was the best model because it was scored the highest accuracy score compared to other algorithms used. Finally we implemented the hyper=parameter tuning technique to find the best parameters for our model, and to find the important features in our data.

Improvement

Improvements that will try do in future works:

- Grid search with more values on GPU accelerated machine.
- Combining all 3 models into a custom ensemble model. using model stacking.

References

[1]https://www.researchgate.net/publication/308842286_A_comparative_study_on_machine_learning_algorithms_for_indoor_positioning

[2]https://www.researchgate.net/publication/313954230_User_Localization_in_an_Indoor_Environment_Using_Fuzzy_Hybrid_of_Particle_Swarm_Optimization_Gravitational_Search_Algorithm_with_Neural_Networks

[3] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[4] https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

[5] https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c

[6] https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/

[7] https://en.wikipedia.org/wiki/Type_I_and_type_II_errors