Building a machine learning-based indoor localization system using the RSSI readings from ibeacons

<insert contact details here>

<insert the date here>

Abstract 2

Introduction 2

Methodology 3

Results 10

Discussion 11

Conclusion 12

References 12

Abstract

Finding accurate coordinates of the person located in the construction is a great assist in many useful use cases. It would be useful in tracking customers in supermarkets to know some patterns about their attitudes towards the different products that locate in the different locations, also it could be used in tracking users in some places that have forbidden areas to make sure they will not go to this places, in the same manner making the customers, students, know where they are located on the buildings will help them to easily go to their destinations. In this paper, we investigate how the Received Signal Strength Indicator (RSSI) readings from the ibeacons used in locating the person's location (coordinates). But, there is a problem the RSSI reading from the ibeacons not enough to accurately define the location of the receiver device (mobile in our case) as when we use the mobile we got a different RSSI reading from each ibeacon then the location recognized based on these readings as the lower the RSSI reading the closer the mobile device to this ibeacon but still we cannot accurately know the coordinates of the person location, so, we have decided to use machine learning classifier to train them on some labeled data with the locations defined. Then, use the trained classifier to predict the accurate coordinates of the person based on those RSSI readings from the ibeacons devices in the place.

Keywords: RSSI, coordinates, predict, classifier

Introduction

In this paper, we investigate how machine learning can be utilized with the data coming from ibeacons to accurately identify the coordinates of the persons within inbounded buildings. So, we have used different machine learning algorithms to be used in fitting the RSSI readings from a 13 ibeacons distributed in a Waldo library that located in the western Michigan university the RSSI recorded using iPhone s6 mobile that works as the receiver for the ibeacons signals and locations and the RSSI readings manually recorded by the experts. So, In our paper, we need to build a working solution that is able to automatically detect the coordinates of the location based on the readings of the RSSI of all the ibeacons in the place. So, the main goal of our research is finding the accurate coordinates of the person's location using only the ibeacons that work as a transmitter of the signals and the iPhone s6 that works as receiver of the signal. In our case, the iPhone receives 13 signals from 13 different ibeacons and RSSI value calculated for the 13 ibeacons this RSSI ranges from -200 which means this ibeacon is out of range and -60 this means that location is the closest to this ibeacon. We will not be able to directly find the location using the RSSI as for example if all the ibeacons are out of the range except one ibeacon we could not decide what is the location as there are many points that could be the closest to this specific ibeacon. The role of machine learning here is finding what is the accurate coordinates of the location that is closest to the specific ibeacon using the patterns that would be found in training the machine learning classifiers on the generated dataset. The dataset consists of 14 features and one label (the location coordinates). The 14 features are:

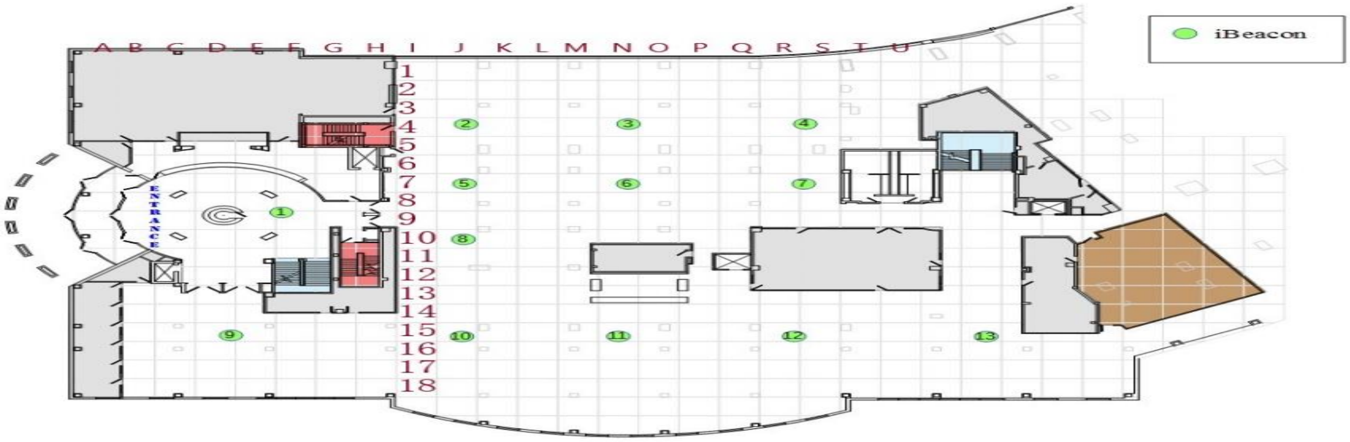
1- b3001: The RSSI reading for ibeacon number 1 at a given time for a specific person
2- b3002: The RSSI reading for ibeacon number 2 at a given time for a specific person
3- b3003: The RSSI reading for ibeacon number 3 at a given time for a specific person
4- b3004: The RSSI reading for ibeacon number 4 at a given time for a specific person
5- b3005: The RSSI reading for ibeacon number 5 at a given time for a specific person
6- b3006: The RSSI reading for ibeacon number 6 at a given time for a specific person
7- b3007: The RSSI reading for ibeacon number 7 at a given time for a specific person
8- b3008: The RSSI reading for ibeacon number 8 at a given time for a specific person
9- b3009: The RSSI reading for ibeacon number 9 at a given time for a specific person

10- b3010: The RSSI reading for ibeacon number 10 at a given time for a specific person
11- b3011: The RSSI reading for ibeacon number 11 at a given time for a specific person
12- b3012: The RSSI reading for ibeacon number 12 at a given time for a specific person
13- b3013: The RSSI reading for ibeacon number 13 at a given time for a specific person
14- date: The DateTime that defines the data and time that RSSI reading has taken for a specific person.
15- location: The coordinates that the person was located on while taking the RSSI reading at a specific time. (M. Mohammadi, 2018)
Hint: The person located in the coordinates of the location recorded for only 3 seconds.

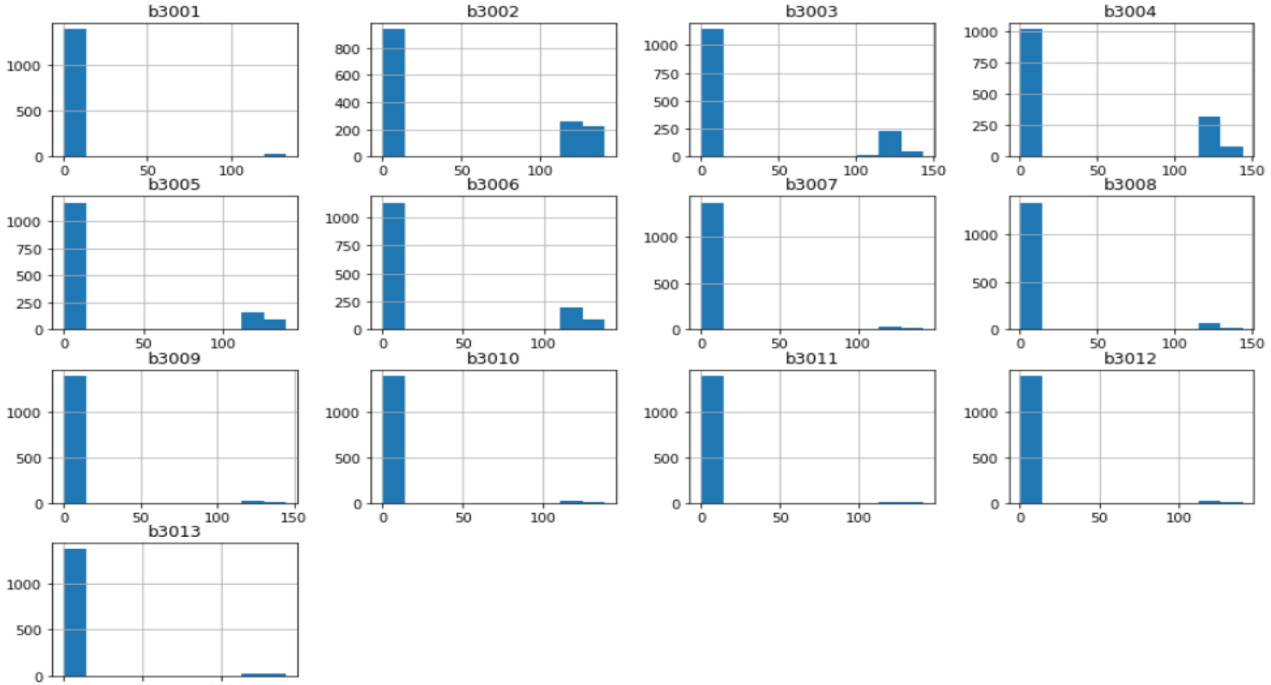
Finally, we have constructed a plan for cleaning the data.
The data is clean data but we need to do some preprocessing for the data. First, we need to ensure the data is clean from the missing, also assert the data doesn't have any out of the range values as the range of RSSI readings for each ibeacons from -50 to -200 so, we assert that there is no value below -200 or over -50. Then, we split the label (location) into two coordinates x and y to be easily visualized, then we add 200 to all the RSSI readings for the whole 13 ibeacons as they have the same range and won't impact anything also working on positive numbers for simplicity and visualization would be much better than working on negative numbers, then we oversample the data to make it balanced and avoid the bias that can be resulted due to imbalance problem, then we would split the data into two partitions; training data and testing data to make it ready for the modeling phase.

Methodology

We would start with the analysis methods and hypothesis of the data and their validity then we will go to explaining the methods that applied to the data to make it ready for the modeling phase.
1- The analysis: we have conducted both types of analysis of the univariate analysis that conducted one each feature individually and the multivariate analysis.
a - The univariate analysis:
As seen in the below figure the histogram for all the RSSI readings for all the 13 variables that each one represents one ibeacon. As noticed in the 13 sub-figures below the beacons b3002, b3003, b3004, b3005, b3006 have the highest RSSI reading values and this seems somehow logical as the position of these beacons in the middle of the map as seen in Figure 2 make it cover a bigger area and good a higher range of readings. In the same manner, we can notice that the beacons b3001, b3007, b3009, b3009, b3010, b3011, b3012, b3012 have very low RSSI readings and this also due to the small range covered by these beacons.

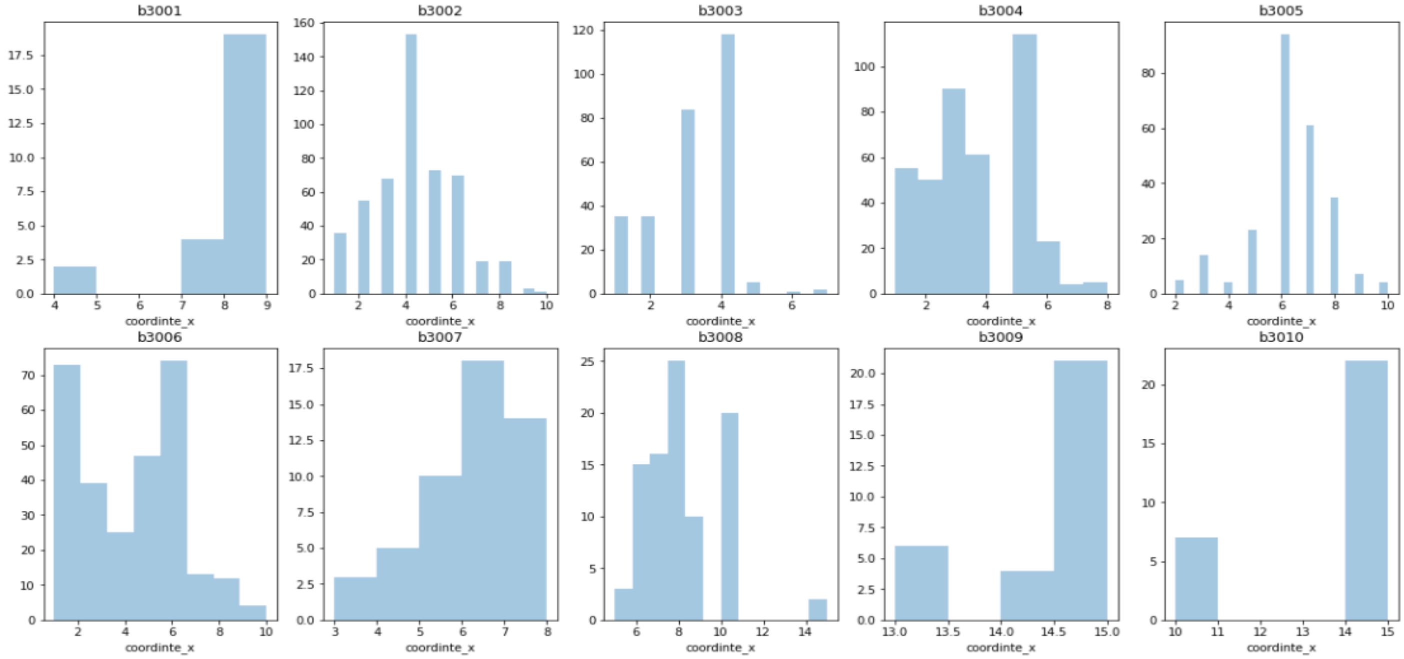


"Figure 1:The map for the waldo library"

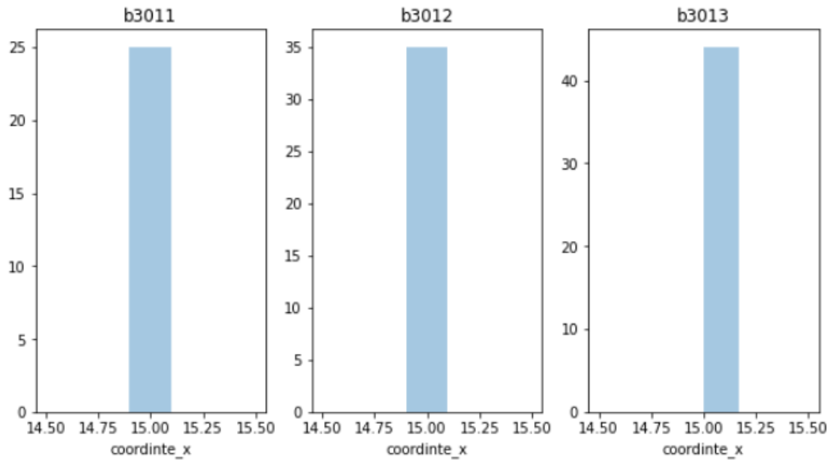


“Figure 2: The histogram of the distribution of the numerical ibeacons measures”

In the same manner, as noticed in the figures Figure 3 and Figure 4 the ibeacons; b3002, b3003, b3004, b3005, b3006, and b3007 their strong RSSI readings are centered around the 2, 4, 6 and 8 x-axis coordinate values while the strong RSSI readings for the ibeacons b3001, b3009, b3010, b3011,b3012, b3013 are centered around the x-axis coordinate 14 and 15. That also due to the distribution of the ibeacons in the Waldo Library.

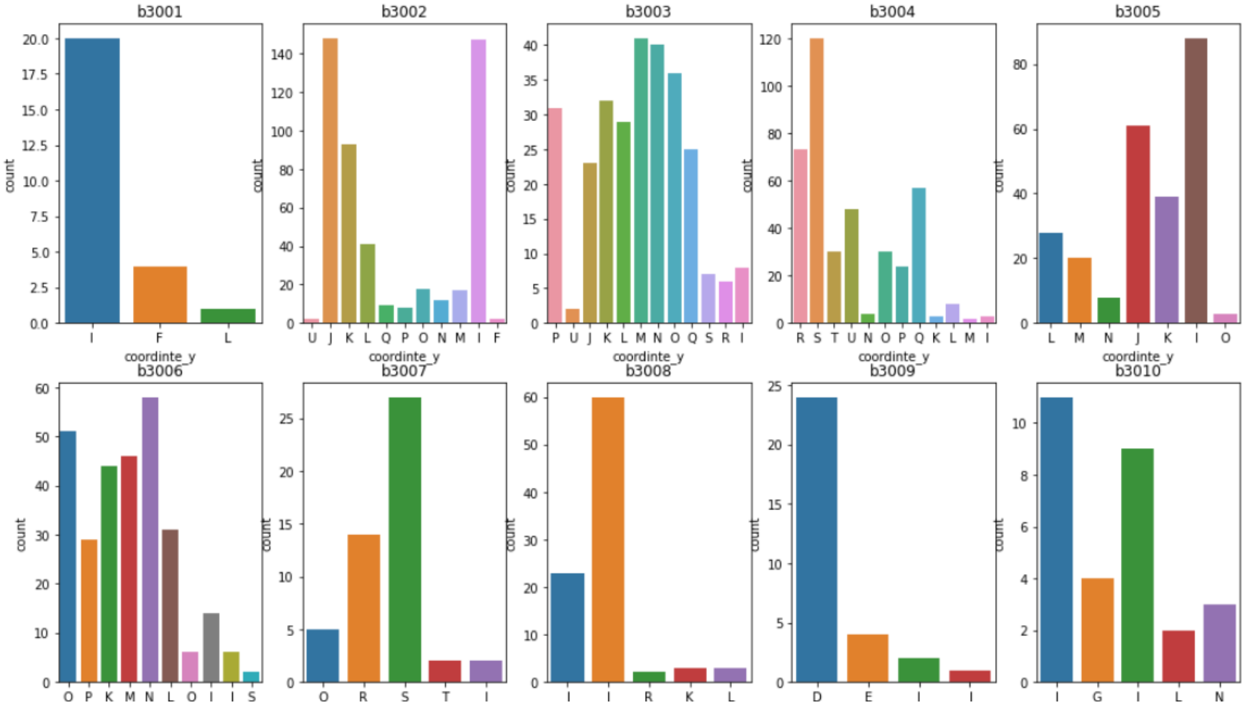


“Figure 3”

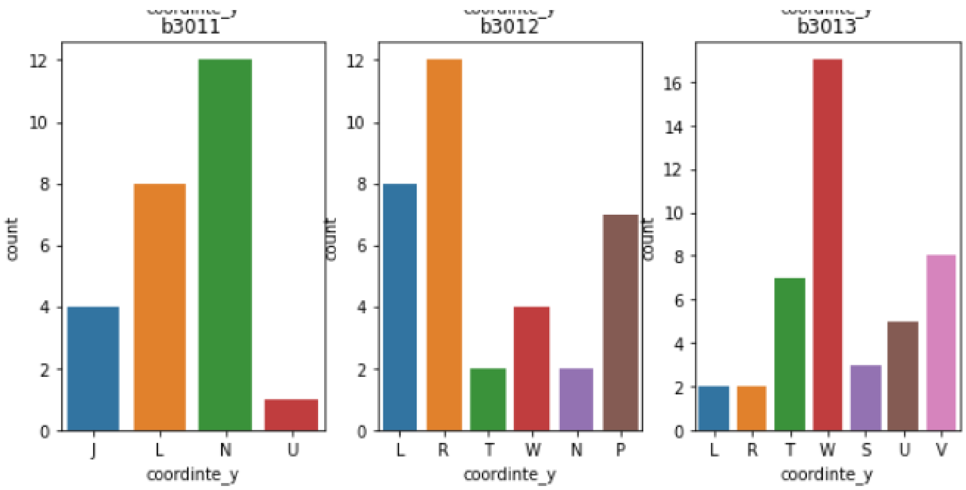


“Figure 4: The relationship between the coordinate_x and each variable”

According to the y-axis, the ibeacon b3001, have strong RSSI readings on the coordinate I, F and coordinate L, while the ibeacon b3002, b3003, b3004, b3005, b3006 have strong RSSI readings at y-axis coordinate I, J, K, L, M, N, O, P, Q, in the same way, we can notice that the ibeacon b3007 generate strong RSSI readings at the y-axis coordinates I, O, R, S, T, for the ibeacons b3012 and b3013 the strong RSSI reading occur at the y-axis coordinate L, R, T, W, S, U, V, while the ibeacon b300 has a strong RSSI reading at the y-axis coordinate K, L, I, R, according to ibeacon b3009 have a strong RSSI reading at y-axis coordinate D, E, I, For the ibeacons b3010 and ibeacons b3011 have a strong RSSI reading at y-axis coordinate I, J, G, L, N, U. All this indicates that the ibeacons distributed in the way that cover all the y-axis coordinates while it is limited for the x-axis coordinate that means to make the localization system more accurate either make the points of measure larger or apply the measurements in the whole points in the Library.



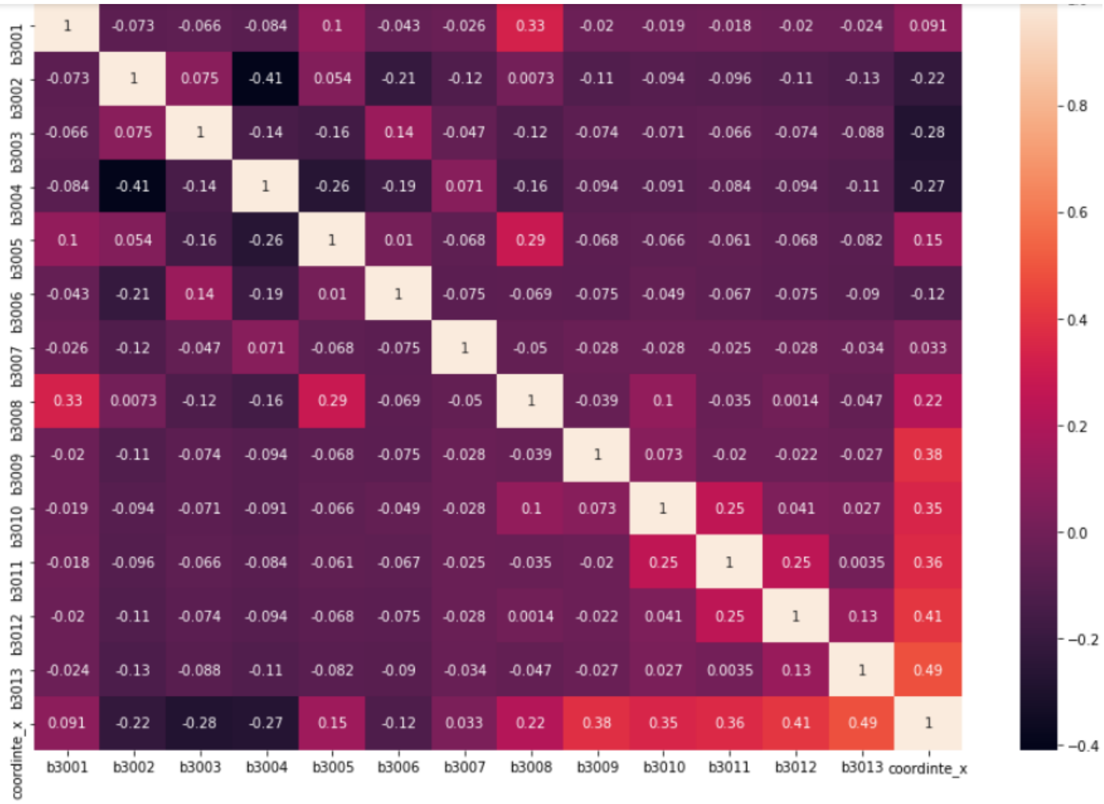
“Figure 5”



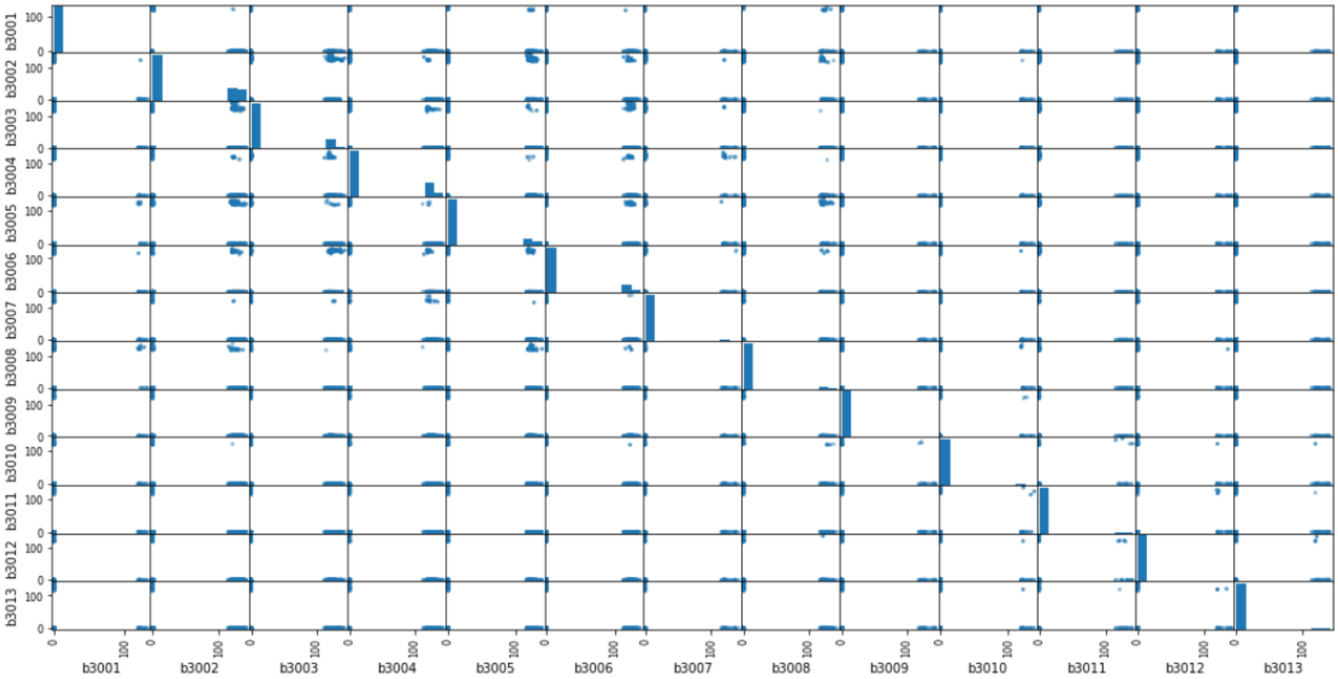
“Figure 6: The relationship between coordinate y and each variable”

2- Multivariate Analysis

In this analysis, we would analyze the relationships between different pairs in the data. So, we have constructed a correlation matrix to investigate the correlation between all the pairs in the data. The Hypothesis is closer two ibeacons to each other the RSSI reading become relatively similar, and as can be noticed in the correlation matrix in Figure 7 and the scatter matrix in the below there is a relatively positive correlation between the RSSI reading of the ibeacon b3001 and the ibeacon b3008, also a relatively positive correlation between the RSSI reading of the ibeacon b3005 and the ibeacon b3008, also a relatively positive correlation between the RSSI reading of the ibeacon b3011 and the ibeacon b3012, this supports the validity of our hypothesis.

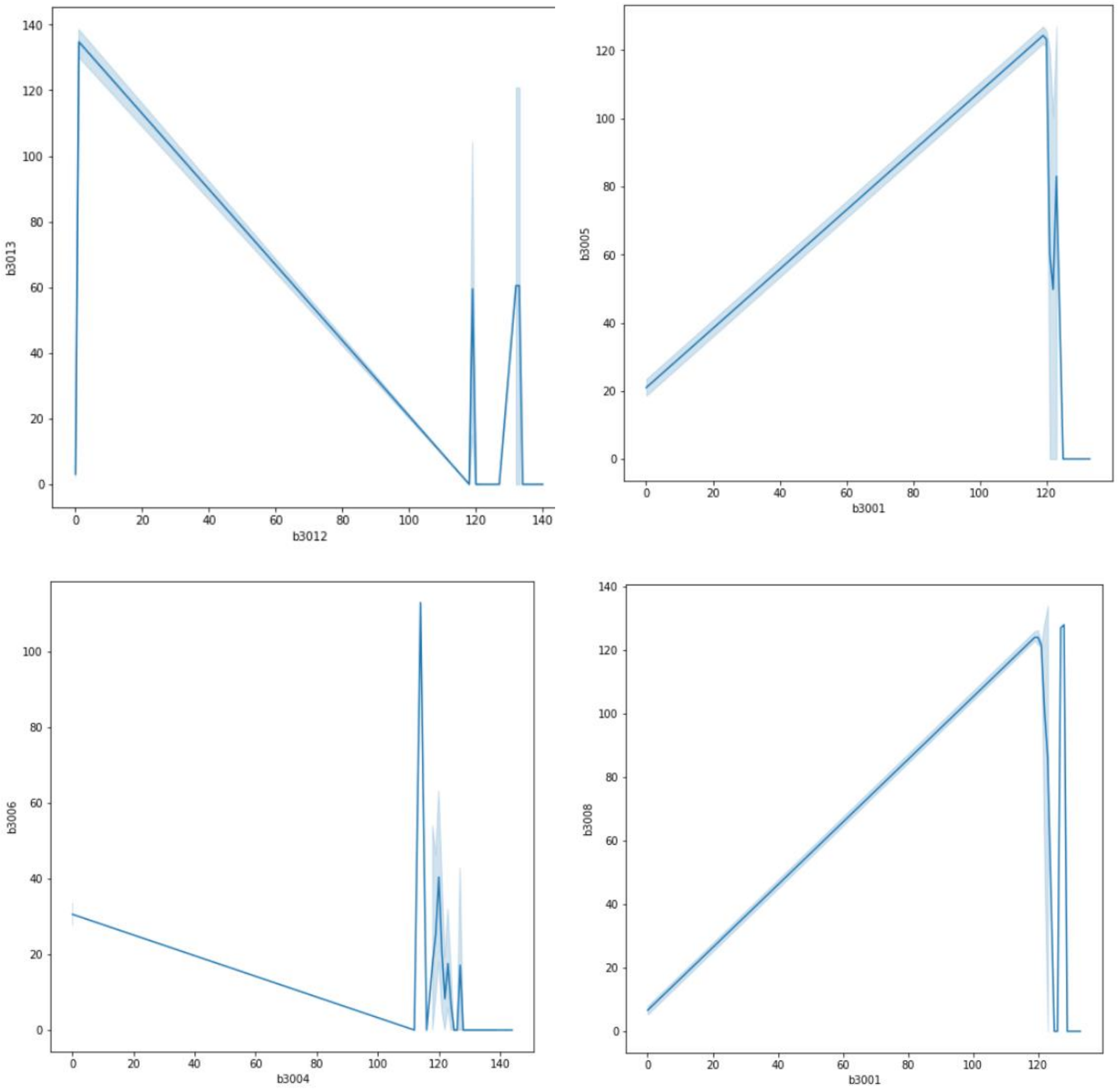


“Figure 7: The correlation matrix”



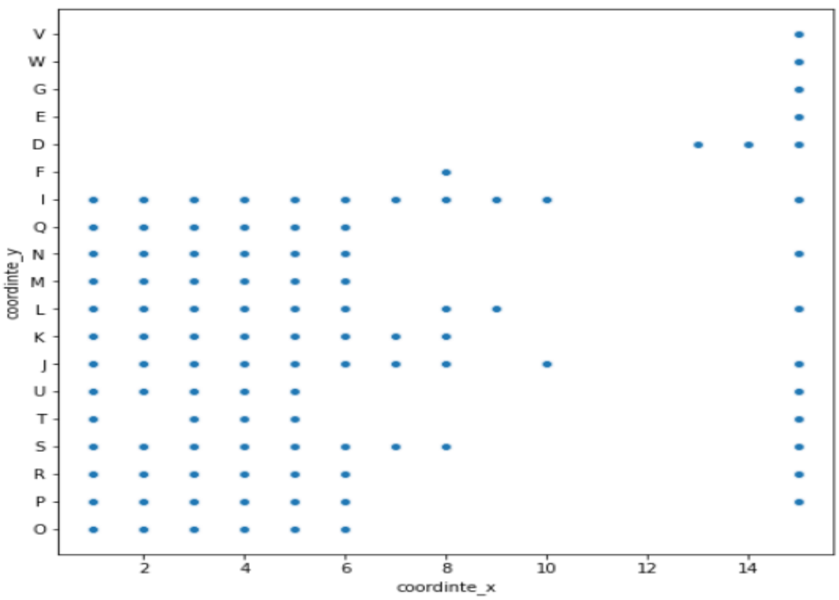
“Figure 8: The correlation scatter matrix”

For elaboration on the relationships between the different ibeacons readings, we have graphed a line plot for some pairs that we noticed some correlation between them. So, The second hypothesis is that there is a positive relationship between both the readings of the ibeacon b3001 and the readings of the ibeacon b3005. The hypothesis is that the lower RSSI readings for the ibeacon b3012 the lower the RSSI readings as they are aside each other But, as can be seen in the below line chart between the RSSI readings of the two ibeacons b3012 and b3013 there is a negative relationship between the two ibeacons readings the reading of ibeacon b3012 still out of the range for all the ibeacon b3013 readings below 130 once the reading of the ibeacon b3013 reach 130 the reading for the ibeacon 3012 start increase and the reading of the ibeacon 3013 start decreasing that means there is a negative relationship between the RSSI readings of those two ibeacons which led to invalidity of our hypothesis. And, as seen in the below line chart the reading of the two ibeacons b3001 and b3005 the readings of both ibeacons start increasing together with almost the same rates until reaching the reading 120 for the ibeacon b3001 the reading for the ibeacon b3005 be 0 (out of the range). The third hypothesis is that there is a strong positive relationship between both RSSI readings of the ibeacon b3001 and the readings of the ibeacon b30058 And, as seen in the below line chart the reading of the two ibeacons b3001 and b3008 the readings of both ibeacons start increasing together with almost the same rates until reaching the reading 120 for the ibeacon b3001 the reading for the ibeacon b3008 reach 0 which means this ibeacons to be out of coverage for this area. The fourth hypothesis is that there is a negative correlation between the RSSI reading from the ibeacon b3004 and the ibeacon b3006, As can be noticed the below line chart between both readings of the two ibeacon; b3004 and b3006 there is no clear relationships between the readings of the two RSSI reading of the two ibeacon that means the invalidity of our hypothesis.



“Figure 9: The relationships between the variables”

In the below figure Figure 10 we need to investigate the hypothesis that not all the points (coordinates) of the Waldo Library have been covered in the data. And, as seen in the below figure that the hypothesis is a true one as it shows only 151 coordinates out of the possible 21*18 = 378 which considered less than half of the possible coordinates. So, for making the classifier that we would build earlier we need to train it on the RSSI reading for the ibeacons in the whole possible coordinates of points in the Waldo Library.



“Figure 10:The point plot between the x and y coordinates”

2- Methods applied for cleaning and preparing the data

After making the analysis we started to ensure that the data is ready for the modeling phase so we have conducted a set of steps to achieve that:

A- First of all we have asserted that the data is clean from any missings or errors by running two test checks on the data. The first test check was checking all the columns of the data if there is any missing and we have found that the data is completely clean from any missings. The second test check was checking the range for the RSSI readings for each ibeacon and we found that there is no reading below -200 or higher than -50 which is the normal range of the RSSI reading for the ibeacons.

B- Adding 200 to all the RSSI readings for the 13 ibeacons as they all have the same range, so this won’t impact anything. We have successfully converted the RSSI readings of the 13 ibeacons into positive values for simplicity in analysis and to make it easier for interpretation.

C- We have dropped the date column as it won’t be useful regarding our application as it expresses the date in which the RSSI has been measured from the ibeacons at specific coordinates. So, it would be considered as redundant as it cannot impact in detecting the specific location.

D- As noticed we have 151 different coordinates (classes) in the location column that considered as the label, working on such a huge number of classes in machine learning as a thought process special if each class has a different number of instances as in our cause that called imbalance of the data which is a huge problem especially in classification applications like our application as imbalance impact the model generalization of the whole classes and makes it somehow biased toward the major classes. For avoiding this and also to increase the data samples we have decided to use oversample method that used to make the whole classes (location coordinates in our case) have the same number of instances as the major class as the K04 class in our application which is the major class with 34 instances we need to oversample all the other classes (coordinates) to match this 34 instances in the K04 major class. So, we have used a method called The Synthetic Minority Over-sampling or SMOTE for simplicity, this method is one of the best methods used for oversampling as it does not replicate samples of the minor classes to make them match the major class but it almost creates a new sample. SMOTE in our application used only one nearest neighbor for one of the minor samples and multiplied it by a random number in the range from 0 to 1 and continued repeating in this process until making the number of instances in the minor classes match the number of the major class. That made our data balanced and all the instances in all the classes are equal (Brownlee,2020).

E- We have selected all the numerical features which express the measurement of the signals of the 13 ibeacons as we couldn't drop any one of them as dropping any of them would hinder us from getting the accurate estimation of some of the locations.

F- Splitting the data into two partitions 80% for training the machine learning model and 20% for testing the model. We have selected this ratio in splitting because we have a small number of samples which is almost 1500 samples, so, we need to provide the model with a decent number of samples to be trained on to be able to fit the data well. In the same way, 20% of the data for testing is a good amount to assess the performance of the model.

G- We have selected three machine learning models that come from different ideas and algorithms and used them for fitting the data with their default hyperparameters values as this is considered as the initial phase in finding the best model then after selecting the best model we would tune and interpret its hyperparameters. Those 3 models are decision trees which utilize the tree structure for finding the outcome (location), Randomforest uses a set of a predefined number of decision trees so it is called an ensemble-based algorithm and the K-nearest neighbors with k = 5. Then we have made predictions using the 3 models and evaluated them using the accuracy score and we have reported the results in the below result section.

Results

After constructing the 3 models discussed above with three different split ratios first one is splitting the data by 80% training and 20% testing, second one is 70% training and 30% testing, and the final one is splitting the data 50% training and 50% testing. We have used the models for prediction on the test data then evaluated them using the accuracy score and the results have been reported in the below table. So, as we can see in the results below Random forest with split 80% training and 20% testing is the best-performing model despite it is somehow not the accuracy that we aimed to but the nature of the data with more than 151 classes each class have below 50 instances so, this result can be acceptable but this could be improved with a tuning set of the Randomforest hyperparameters.

Model	Accuracy
Decision Tree with split (80% training and 20% testing)	0.571
Random Forest(80% training and 20% testing)	0.574
KNN(80% training and 20% testing)	0.494
Decision Tree with split (70% training and 30% testing)	0.556
Random Forest(70% training and 30% testing)	0.574
KNN(70% training and 30% testing)	0.484

Decision Tree with split (50% training and 50% testing)	0.554
Random Forest(50% training and 50% testing)	0.577
KNN(50% training and 50% testing)	0.451

“Table1: The comparison of models”

So, we have selected six hyperparameters to be tuned using a randomized search method that randomly select combinations of the hyperparameters then use them for fitting the model and keep do that for a specific number of iteration that defined while building the random search method then it gets the best hyperparameters that achieved the highest score. Those Hyperparameters are:

A- n_estimators: This indicates the number of decision trees would be ensembled. The values [100, 200, 300, 400,500, 600, 700, 800, 900, 1000, 1100, 1200,1300, 1400, 1500, 1600,1700, 1800, 1900,2000] would be tested for that hyperparameter.

B- max_depth: This indicates the maximum number of the levels in each ensembled tree. The values [10,20,30,40,50,60,70,80,90,100] would be tested for that hyperparameter.

C- min_samples_split: This indicates the minimum samples required for each node to split; we would test 20 values in the range from 2 to 100 as a node at least required 2 samples to split.

D- bootstrap: This indicates whether using the bootstrap method in selecting the training sample or not. Only two values would be tested for that hyperparameter either True or False.

E- min_samples_leaf: This indicates the minimum samples required for each leaf node to split; we would test 20 values in the range from 1 to 20 (Koehrsen,2018).

In the below table we compare the accuracy of the Randomforest before and after applying Randomized search tuning.

Model	Accuracy
Randomforest before tuning	0.571
Random Forest after tuning	0.575

“Table 2”

Discussion

As we can notice from the results in the two tables Table 1 and Table 2 Randomforest model is the most suitable for the data and especially after tuning it increased by almost 0.004 which is a relatively good value in terms of machine learning models mechanism of the improving. So, the tree structure seems to be the best fitting the data rather than the other types like SVM, logistic regression, Naive Bayes, etc. As the nature of data with more than 151 different classes, it trained on them so the hierarchical nature of the tree-based methods it could be adopted with this enormous number of classes in which a low number of instances rather than the other types of the model that could be work better in the case of the low number of classes.

Conclusion

Currently, we have a model that able to detect the coordinates of the locations based on the RSSI readings from different ibeacons with accuracy almost 58%, surely this accuracy is somehow low accuracy in terms of machine learning method but as seen in the code part the most mistakes done by the model was almost aside the correct coordinates which mean if the points were somehow bigger the accuracy surely would be increased as the number of classes would be decreased. Also, we need to mention that based on our application the accuracy of 58% is satisfactory as predication the exact locations are not the goal but the goal is a prediction of arbitrary accurate location and that what our model achieved.

References

Koehrsen, Will. (2018). Hyperparameter Tuning the Random Forest in Python. Retrieved from https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

Brownlee, Jason. (2020). SMOTE for Imbalanced Classification with Python. Retrieved from https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

M. Mohammadi and A. Al-Fuqaha, (2018).' Enabling Cognitive Smart Cities Using Big Data and Machine Learning: Approaches and Challenges,' IEEE Communications Magazine, vol. 56, no.