CSC 859 AI Ethics and Explainability

San Francisco State University Fall 2022

Team project: Development and Ethics Audit of AI Application

Team #1

Theodore McCullough <tmccullough@sfsu.edu>

Ishank Aggarwal <iaggarwal@sfsu.edu>

Alex Rabanes <arabanes@sfsu.edu>

Date: 12/12/22

**I. Executive Summary**

Location Detection in a multi-room structure is performed using a Random Forest Machine Learning Model trained on a Wireless Indoor Localization Data Set to make location predictions relating to persons in this structure. Relative to other ML approaches, the RF-ML Model has better Accuracy, but, as will be discussed in this report, with the benefit of high confidence. In addition to Accuracy, the RF-ML Model lends itself to superior explainability and can meet the ethical, legal, and business challenges ultimately resulting in a RF-ML Product that we can take to market.

20

40

60

100

80

0

RF-ML

FPSOGSA-NN

NAÏVE BAYES

GSA-NN

PSOGSA-NN

PSO-NN

SVM

64.66%

83.28%

77.53%

95.16%

90.47%

92.68%

100%

**FIG. 1: Graph representing the Accuracy advantages of RF-ML Model as compared to other ML models used to perform Location Detection. Accuracy data taken from the article by Rohra et al.[[1]](#endnote-1) which used models trained on a smaller three (3) class data set, as opposed to the larger four (4) class data set used herein. As shown, RF-ML Model attains 100% accuracy, but (as will be discussed in detail in this report) with better explainability than the Neural Networks (NN) models used by Rohra et al.**

**II. Problem Description and Case Study Goals**

The detection and prediction of a person’s location in a multi-room structure (“Location Detection”) can be important, if not critical, for certain applications.[[2]](#endnote-2) For example, a person’s location is important for marketing purposes[[3]](#endnote-3), but critical for law enforcement[[4]](#endnote-4). This problem of Location Detection presents a number of technical, ethical and legal, and business challenges and, in certain cases, risk. That said, if these challenges and risk can be appropriately addressed, the value proposition[[5]](#endnote-5) for a Location Detection solution is immense.

Starting with the technical challenges, an initial question to consider is what are the technical means to detect a person’s location?; while a second question is how would one evaluate the output from this technical means? For the purposes of this report, we are using data in the form of signal strength values generated by Wi-Fi Access Points, and received by a mobile device, as the means to detect a person’s location in a multi-room structure. The benefits of using Wi-Fi Access Points for such an application are well known in the industry.[[6]](#endnote-6)

While the benefits of using of Wi-Fi Access Points are well known, it is the second question of evaluating this output that presents challenges. Specifically, methods, and Machine Learning (ML) methods in particular, to evaluate and engage in Location Detection based upon the signal data generated by these Wi-Fi Access Points can vary.[[7]](#endnote-7) However, one common problem associated with such an evaluation is that of local minima.[[8]](#endnote-8) As applied in the context of Location Detection, the problem of local minima arises where a person is predicted to be in one room versus another, and the prediction fails to take into account data showing that the person might be in a room different from the predicted room due to the ML being effectively locked into a local minima[[9]](#endnote-9). To resolve this problem of local minima in the Location Detection space, and as will be discussed below in more detail, we will use and evaluate Random Forest ML (RF-ML) for Location Detection. One advantage of the RF-ML is that it avoids the problem of local minima through the random selection of sample data.

In addition to technical challenges, Location Detection presents a number of ethical, legal, and business challenges and, in certain cases, risk. As will be discussed in more detail in this report, not only would any lack of explainability related to RF-ML present ethical challenges to selling a Location Detection solution, but the fact that Location Detection tracks the behavior of people in multi-room structures (e.g., track people in their home) presents additional ethical challenges. Legal challenges in the form of obtaining consent to use what is Personal Data[[10]](#endnote-10) are also addressed in this report. Finally, as to business challenges, given the personal nature of the data used to train and take to market an RF-ML model for Location Detection, addition cost may need to be incurred to limit the ethical and legal risks presented by such a Location Detection Solution.

**III. Data Description and Audit**

For the purposes of training and testing this RF-ML based Location Detection solution (a “RF-ML Model”), we utilized a tabular Wireless Indoor Localization Data Set[[11]](#endnote-11) (“Data Set”) consisting of data in the form of the signal strengths of seven (7) Wi-Fi access points visible to a mobile device. Specifically, this Data Set is comprised of the following:

* 2000 samples
* Seven (7) features per sample, each feature representing the signal strength of a Wi-Fi access point appearing on a mobile device
* Four (4) possible classes (1, 2, 3, 4) for each sample, the classes corresponding to one of four (4) rooms in a structure

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Access Point 1 | Access Point 2 | Access Point 3 | Access Point 4 | Access Point 5 | Access Point 6 | Access Point 7 | Class |
| -64 | -56 | -61 | -66 | -71 | -82 | -81 | 1 |

**FIG. 2: Table representing a single sample and the feature titles and class value associated with this sample. To the Data Set, column headers were added (i.e., “Access Point 1”, “Access Point 2”, “Access Point 3”, “Access Point 4”, “Access Point 5”, “Access Point 6”, “Access Point 7”).**

As described in FIG. 1, the Data Set is numeric (as opposed to categorical in nature), has no missing data or noise, and each class comprises twenty-five percent (25%) of the sample total[[12]](#endnote-12) such that no one class comprises less than ten percent (10%) of the sample total. Further, there are more than ten times (10x) the number of samples, than features. Of note, to the extent there is high variance between feature values this could result in a dominance problem where this high variance could effectively skew the rest of the data in the Data Set. While not performed as a part of the data processing conducted herein, Mean Centering, setting the Standard Deviation of the mean to “1”, or other techniques could be applied to address this dominance problem.

**IV. Application of Random Forest ML Method to Data Set**

**Advantages of RF ML Generally**

RF-ML is a type of ensemble learning utilizing a plurality of decision trees[[13]](#endnote-13) to perform classification tasks on tabular data. One advantage[[14]](#endnote-14) of RF ML is that, relative to other ML models[[15]](#endnote-15), it is explainable and transparent, as opposed to opaque, in nature. The explainable and transparent nature of our RF-ML Model is explored in the following sections of this report.

**Description of RF-ML Model Set Up**

We analyzed the Data Set using an RF-ML Model with one thousand (1000) trees, and with the optimal number of features to examine at each split as six (6).[[16]](#endnote-16) Unless otherwise stated, we used the implementation and default parameters of Sci-Kit Learn as implemented in a Jupyter Notebook IDE. These default parameters included the default cut off value of .5[[17]](#endnote-17). We divided the Data Set into train (1600 samples) (“Train Instance”) and test (400 samples) (“Test Instance”) instances. Further, Cross Validation was not used as inherent to RF-ML is the random dividing of the test instances into multiple subsets that approximates the benefits of Cross Validation.

A picture containing text

Description automatically generated

**FIG. 3: Image of three (3) tree forest of an RF-ML Model generated from the training stances. This forest has approximately one-hundred and ninety-seven (197) nodes generated from the test instances. Further, and as more fully described in FIGs 3 & 4, parent nodes have an “Access Point” feature identifier and leaf nodes do not. As noted above, the actual RF-ML Model used throughout this report has one thousand (1000) trees and would not be easily used to show the actual structure of the RF-ML Model, hence FIGs. 3-5 show this structure albeit on a smaller scale.**

Access Point 1 <= 54.5

gini = 0.75

samples = 1006

values = [399, 409, 412, 380]

class= Room 3

**FIG. 4: Image of the root, parent node of the individual tree from FIG. 2. The “Access Point” identifier denotes the probability that the identified feature is associated with a sample (a high value meaning a greater probability), the “gini” value denotes the probability of misclassification (with “0” meaning a low chance of classification, and “1” meaning a high chance), the “samples” value denoting the samples utilized to make a classification decision (here 1006 references the size of the training instances), “values” set denoting the distribution of samples associated with a particular class, and “class” being the likely class given this distribution of the samples.**

gini = 0.0

samples = 1

values = [0, 0 , 0, 1]

class= Room 4

**FIG. 5: Image of a leaf node of Tree 1 from FIG. 2. This leaf node is a child of a parent node for Access Point 6. Of note, as this is a leaf node, the “gini” value is “0” (no likelihood of misclassification), and “values” set has only one value denoting a single class. Of note, even though the “gini” value might be “0”, there is still the possibility of false positives and false negatives. As will be discussed in the next few sections of this report, there are various confidence related metrics (e.g., Confusion Matrices, Recall, Precision, F1 Score, OOB Score, and RFEX) that can be used to develop an intuition of the level of confidence that one should have in the “0” value generated by the “gini” function.**

**Analysis of the RF-ML Model**

While the stated Accuracy of our RF-ML Model is one-hundred percent (100%), Accuracy alone is not enough due to the twin problems of false positives and false negatives. Specially, a prediction made by our RF-ML Model may be accurate, but the classification itself incorrect. As compared to the actual observed data (i.e., the Ground Truth[[18]](#endnote-18)) a true prediction could in fact be false (a false positive), and a false prediction could in fact be true (a false negative). To address the problems of false positive and false negative, we utilized a number of confidence related metrics to evaluate the Accuracy of our RF-ML Model.

**Accuracy**

The starting point for evaluating the Accuracy of any ML model is understanding what constitutes Accuracy. Accuracy is defined as:

Total correctly classified samples (i.e., TP + TN)/total samples (i.e., TP + TN + FP + FN), where “TP” is the True Positive value, “TN” is the True Negative value, “FP” is the False Positive value, and “FN” is the False Negative Value.[[19]](#endnote-19)

As described in FIG. 1, and as provided below, the Test Instances were used by the RF-ML Model to calculate Accuracy.

Accuracy is:

1.0

**FIG. 6 is an Accuracy prediction generated using the RF-ML Model and data in the form of the Test Instances. As shown here, and in FIG. 1, the stated Accuracy of our RF-ML Model is one hundred percent (100%)**

**Code Snippet for Accuracy in the RF-ML Model**

#Input: Feature Test Set (x\_test), Label Test Set (y\_test), Trained Random Forest Object (clf)

#Processing: Creates an array (y\_pred) of predicted labels from x\_train, provides y\_pred and y\_train to accuracy\_score function

#Output: Prints Accuracy score for Trained Random Forest Object

def Print\_Accuracy(x\_test, y\_test, clf):

length = len(x\_test)

y\_pred = []

for index in range(0, length):

y\_pred.append(clf.predict(x\_test[[index]]))

accuracy = accuracy\_score(y\_test, y\_pred, normalize=True)

print("Accuracy is:")

print(accuracy)

**Confusion Matrix**

While at first blush the Accuracy score of 1.0 in FIG. 6 may seem dispositive of the effectiveness of our RF-ML Model, Accuracy alone is not enough. Specifically, Accuracy as a measure does not take into account the FP and FN value relative to the total samples as one could have a high Accuracy score, yet high FP or FN values. As will be discussed in more detail below, other metrics such as Confusion Matrices, Precision, Recall, and F1 score can be used to more fully evaluate the FP and FN values relative to the total samples and in doing so provide a better sense of the confidence in the RF-ML Model.

Graphical user interface, application, Teams

Description automatically generated

**FIG. 7 is a Confusion Matrix generated from the Test Instances. As shown, there are four (4) instances of TP with values of 99, 94, 94, and 106 corresponding to Room 1, Room 2, Room 3, and Room 4 respectively. Further, there is one (1) instance of False Negative (FN) for Room 1, and two (2) instances of FN for Room 2. As to False Positive (FP) there are three (3) instances for Room 3, and a single (1) instance of FP for Room 4. The sum of value represented in this matrix is 400 corresponding the number of test instances. As the FP and FN rates for the RF-ML Model is under three percent (3%)[[20]](#endnote-20), this suggests we should high confidence in the classifications made by the RF-ML Model for the given application (i.e., the prediction of a person’s location in room in a structure). By contrast, a 3% FP or FN rate for a ML model that detects cancer would be unacceptable.**

**Code Snippet for Confusion Matrix**

#Input: Random Forest Object (clf), Feature Test Set (x\_test), Label Test Set (y\_test)

#Processing: Creates labeled Confusion Matrix for Random Forest Object

#Output: PLots Confusion Matrix for Random Forest Object

def Print\_Confusion\_Matrix(clf, x\_test, y\_test):

length = len(x\_test)

y\_pred = []

for index in range(0, length):

y\_pred.append(clf.predict(x\_test[[index]]))

cm = confusion\_matrix(y\_test, y\_pred, labels=clf.classes\_, normalize = None)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm,display\_labels=["Room 1","Room 2", "Room 3", "Room 4"])

disp.plot(cmap=plt.cm.Blues)

plt.show()

**Precision, Recall, and F1 Metrics**

Where the Confusion Matrix of FIG. 7 provides a graphical representation of the various TP, TN, FP, FN values, the concepts of Precision, Recall, and F1 scores place numerical values on the ratios between these values in order to better understand the confidence that one might have in the Accuracy of our RF-ML Model. As shown below, these scores are 1.0, suggesting high confidence in the Accuracy of our model.

Precision is defined as:

The number of correctly classified positive results divided by the total number of positive results predicted by the classifier. Precision = TP/(TP + FP)[[21]](#endnote-21)

Recall is defined as:

The number of correctly classified positive results divided by the number of all true positive samples. Recall = TP/(TP + FN), where TP is the True Positive value, and FN is the False Negative value.[[22]](#endnote-22)

F1 Score is defined as:

A single number combining recall and precision. F1 = 2 \* (recall\*precision)/(recall + precision)[[23]](#endnote-23)

precision recall f1-score support

Room 1 1.00000 1.00000 1.00000 400

Room 2 1.00000 1.00000 1.00000 404

Room 3 1.00000 1.00000 1.00000 403

Room 4 1.00000 1.00000 1.00000 393

accuracy 1.00000 1600

macro avg 1.00000 1.00000 1.00000 1600

weighted avg 1.00000 1.00000 1.00000 1600

**FIG 8 is a report generated by applying a Precision function to the RF-ML Model, the function generating a report for Precision, Recall, and F1 score from the test instances.**

**Code Snippet for the Report Function Use to Generate Precision, Recall and F1 Score**

#Input: Feature Test Set (x\_test), Label Test Set (y\_test), Trained Random Forest Object (clf)

#Processing: Creates an array (y\_pred) of predicted labels from x\_train, provides y\_pred and y\_train to classfication\_report function

#Output: Prints classification report with F1, Recall, and Accurcy Score

def Print\_Report(x\_test, y\_test, clf):

length = len(x\_test)

y\_pred = []

for index in range(0, length):

y\_pred.append(clf.predict(x\_test[[index]]))

target\_names = ["Room 1", "Room 2", "Room 3", "Room 4"]

print(classification\_report(y\_test, y\_pred, target\_names=target\_names,

sample\_weight=None, digits = 5))

**OOB Accuracy**

A further RF-ML specific metric to evaluate confidence in our RF-ML Model is Out Of Bag (“OOB”) Accuracy. By way of background, in RF-ML there are samples that are “In the Bag” or OOB. In the Bag samples are those being used by a tree in the RF-ML Model forest[[24]](#endnote-24) to make a prediction, while the OOB samples are those not being used by this tree. OOB Accuracy determines the accuracy of the RF-ML Model as it relates to those samples not being used by a particular tree, and is akin to an F1 score for RF-ML models.

F1 score generally:

1.0

OOB score generally:

0.98125

**FIG. 9 is the F1 Score and OOB Accuracy for the RF-ML Model predictions based upon the Test Instances. As with the previously discussed with respect to Recall, Precision, and F1 Scores metrics, the OOB Accuracy Score is high (98% ) suggesting high confidence in the Accuracy of the RF-ML Model.**

**Code Snippet for OOB Accuracy**

#Input: Trained Random Forest Object (clf), Feature Test Set (x\_test), Label Test Set (y\_test)

#Processing: Creates an array (y\_pred) of predicted labels from x\_train, provides y\_pred and y\_train to F1 function, generates OOB accuracy score for clf

#Output: Prints f1 score and OOB accuracy scores generally for clf

def Print\_OOB\_F1(x\_test, y\_test, clf):

length = len(x\_test)

y\_pred = []

for index in range(0, length):

y\_pred.append(clf.predict(x\_test[[index]]))

f1 = f1\_score(y\_test, y\_pred, labels= [1,2,3,4], average=None)

f1\_add = f1[0] + f1[1]

f1\_average = f1\_add/2

print("F1 score generally:")

print(f1\_average)

print("OOB score generally:")

print(clf.oob\_score\_)

**ROC/AUC Analysis**

With respect to optimizing the RF-ML Model, Receiver Operating Characteristics (ROC) (and an associated Area Under the Curve (AUC) analysis)[[25]](#endnote-25) may be used to understand the TP rate (or sensitivity) values relative to the FP rate, and in doing so enable one to adjust the RF-ML Model to meet business or legal goals. For example, while a perfect ML model may have a TP rate of “1.0” (i.e., 100%) and a FP rate of “0”, such a 1.0 TP rate may not be necessary for all application of the RF-ML model, and an RF-ML model with a reduced overall rate may be preferable especially where the risk presented by the model low. This preference may be based upon such things savings on- computing costs, time or other factors.

One-vs-One ROC AUC scores:

0.999296 (macro),

0.999319 (weighted by prevalence)

One-vs-Rest ROC AUC scores:

0.999326 (macro),

0.999349 (weighted by prevalence)

**FIG. 10 represent ROC and AUC score for the RF-ML Model based upon a One-vs-One scheme that compares every unique pairwise combination of classes, and a One-vs-Rest scheme compares each class against all the others (assumed as one). Depending on the application of our RF-ML Model, a lower score may be acceptable, if the application is for, for example, marketing purposes as opposed to say monitoring one’s location in a structure as part of a criminal sanction (e.g., home detention).**

**Code Snippet Used to Generate ROC/AUC Analysis**

#Input: Label Test Set (y\_test), Feature Test Set (x\_test), Trained Random Forest Object (clf)

#Processing: Uses input to determine true positive rate over false positive rate at different classification thresholds

#Output: Represents ROC values

def Print\_ROC(y\_test, x\_test, clf):

y\_prob = clf.predict\_proba(x\_test)

macro\_roc\_auc\_ovo = roc\_auc\_score(y\_test, y\_prob, multi\_class="ovo", average="macro")

weighted\_roc\_auc\_ovo = roc\_auc\_score(

y\_test, y\_prob, multi\_class="ovo", average="weighted"

)

macro\_roc\_auc\_ovr = roc\_auc\_score(y\_test, y\_prob, multi\_class="ovr", average="macro")

weighted\_roc\_auc\_ovr = roc\_auc\_score(

y\_test, y\_prob, multi\_class="ovr", average="weighted"

)

print(

"One-vs-One ROC AUC scores:\n{:.6f} (macro),\n{:.6f} "

"(weighted by prevalence)".format(macro\_roc\_auc\_ovo, weighted\_roc\_auc\_ovo)

)

print(

"One-vs-Rest ROC AUC scores:\n{:.6f} (macro),\n{:.6f} "

"(weighted by prevalence)".format(macro\_roc\_auc\_ovr, weighted\_roc\_auc\_ovr)

)

**Feature Importance (Mean Decrease in Impurity (MDI) Analysis)**

A further metric to optimize the RF-ML Model is to use an MDI Analysis to rank the features of the Data State so as to determine the most important features.[[26]](#endnote-26) This is valuable as it allows us to optimize the training and testing of the RF-ML Model to specific features as opposed to all feature, this saving time and money in the training and testing of the RF-ML Model. For example, as depicted in FIG. 11, Access Points 1, & 5 are the most important futures utilized by the RF-ML Model in making predictions, and hence if efficiency is central in the building of a Location Detection Solution implementing our RF-ML Model we may want to explore taking to market a solution that only looks at data from this these Access Points 1 & 5 for a given multi-room structure.

A picture containing graphical user interface

Description automatically generated

Access Point 1 0.586357

Access Point 5 0.358906

Access Point 4 0.023625

Access Point 6 0.013200

Access Point 3 0.010313

Access Point 2 0.004502

Access Point 7 0.003097

**FIG. 11 is a bar graph and associated MDI ranking data for the RF-ML Model making predictions utilizing the test instances. As shown above, Access Points 1 & 5 together constitute approximately 94% of the features utilized to make a prediction by the RF-ML Model.**

**Code Snippet for MDI Analysis**

#Input: Trained Random Forest model(clf)

#Processing: Loads feature names, selects top feature types used by clf

#Output: Bar graph of top feature types for clf, and data for top features types

def Print\_MDI(clf):

feature\_names = ["Access Point 1","Access Point 2","Access Point 3","Access Point 4","Access Point 5","Access Point 6","Access Point 7"]

mdi\_importances = pd.Series(clf[-1].feature\_importances\_, index=feature\_names).sort\_values(ascending=True)

ax = mdi\_importances.plot.barh()

ax.set\_title("Random Forest Feature Importances (MDI)")

ax.figure.tight\_layout()

mdi\_importances.sort\_values(ascending=False, inplace=True)

print(mdi\_importances)

**Top 1, 2, 3 Features and Associated Confusion Matrices, Precision, Recall, and F1 Metrics**

Like the Confusion Matrices, Precision, Recall, and F1 Metrics provided generally for the RF-ML Model, in this section we explore these values for the Top 3 features as identified through the MDI analysis (i.e., Access Point 1, Access Point 5, and Access Point 4). Through this exploration we will be provided a better sense of how these features reflect the above provided general Precision, Recall, and F1 Metrics.

Confusion Matrix:

[[108 0 0 0]

[ 0 96 3 0]

[ 1 0 92 1]

[ 0 0 1 98]]

Classification Report:

precision recall f1-score support

1 0.99 1.00 1.00 108

2 1.00 0.97 0.98 99

3 0.96 0.98 0.97 94

4 0.99 0.99 0.99 99

accuracy 0.98 400

macro avg 0.98 0.98 0.98 400

weighted avg 0.99 0.98 0.99 400

Accuracy is:

0.985

OOB Accuracy is:

0.975

**FIG. 12 Top 1 (Access Point 1) feature and associated Confusion Matrix, Precision, Recall, F1 Score, OOB Accuracy, and Accuracy. With respect to the Confusion Matrix, there are TP values in the form of 108 (Room 1), 96 (Room 2), 92 (Room 3), and 98 (Room 4), FP values of 3 (Room 2), and 1 (Room 3), and FN values of 1 (Room 3) and 1 (Room 4). The sum of value represented in this matrix is 400 corresponding the number of test instances. By contrast the Confusion Matrix in FIG. 7 has TP values of 99 (Room 1), 94 (Room 2), 94 (Room 3), and 106 (Room 4). Further, there is one (1) instance of False Negative (FN) for Room 1, and two (2) instances of FN for Room 2. As to False Positive (FP) there are three (3) instances for Room 3, and a single (1) instance of FP for Room 4. Of note, even where only data generated by Access Point 1 is used, the FP and FN negative rates are low (e.g., 1/92 (or 1%) as the FN and FP rate) for the given application of location detection in a structure. This low rate suggests high confidence in using Access Point 1 data alone.**

Confusion Matrix:

[[100 0 0 0]

[ 0 94 2 0]

[ 0 3 94 0]

[ 0 0 0 107]]

Classification Report:

precision recall f1-score support

1 1.00 1.00 1.00 100

2 0.97 0.98 0.97 96

3 0.98 0.97 0.97 97

4 1.00 1.00 1.00 107

accuracy 0.99 400

macro avg 0.99 0.99 0.99 400

weighted avg 0.99 0.99 0.99 400

Accuracy is:

0.9875

OOB Accuracy is:

0.97875

**FIG. 13 Top 2 (Access Point 1, & 5) MDI identified feature and associated Confusion Matrix, Precision, Recall, F1 Score, Accuracy OOB Accuracy Scores. With respect to the Confusion Matrix, there are TP values in the form of 100 (Room 1), 94 (Room 2), 94 (Room 3), and 107 (Room 4), FP values of 2 (Room 2), and FN values of 3 (Room 3). The sum of value represented in this matrix is 400 corresponding the number of test instances. By contrast the Confusion Matrix in FIG. 7 has TP values of 99 (Room 1), 94 (Room 2), 94 (Room 3), and 106 (Room 4). Further, there is one (1) instance of False Negative (FN) for Room 1, and two (2) instances of FN for Room 2. As to False Positive (FP) there are three (3) instances for Room 3, and a single (1) instance of FP for Room 4. Like FIG. 12 showing the FP and FN rates for using data generated by Access Point 1, here the FP and FN rates are low (2/94 or 2% for FN) when using data generated by both Access Point 1 and Access Point 5. This again suggests high confidence in Accuracy.**

Confusion Matrix:

[[111 0 1 0]

[ 0 99 1 0]

[ 3 0 91 0]

[ 2 0 0 92]]

Classification Report:

precision recall f1-score support

1 0.96 0.99 0.97 112

2 1.00 0.99 0.99 100

3 0.98 0.97 0.97 94

4 1.00 0.98 0.99 94

accuracy 0.98 400

macro avg 0.98 0.98 0.98 400

weighted avg 0.98 0.98 0.98 400

Accuracy is:

0.9825

OOB Accuracy is:

0.981875

**FIG. 14 Top 3 (Access Points 1, 5, &4) features and associated Confusion Matrix, Precision, Recall, F1 Score, Accuracy, and OOB Accuracy Scores. With respect to the Confusion Matrix, there are TP values in the form of 111 (Room 1), 99 (Room 2), 91 (Room 3), and 92 (Room 4), FP values of 1 (Room 1), and 1 (Room 2), and FN values of 3 (Room 3) and 2 (Room 4). The sum of value represented in this matrix is 400 corresponding the number of test instances. By contrast the Confusion Matrix in FIG. 7 has TP values of 99 (Room 1), 94 (Room 2), 94 (Room 3), and 106 (Room 4). Further, there is one (1) instance of False Negative (FN) for Room 1, and two (2) instances of FN for Room 2. As to False Positive (FP) there are three (3) instances for Room 3, and a single (1) instance of FP for Room 4. Like FIGs. 13 and 14 the FP and FN rates are again low (e.g., 3/91 or 3% of FP) relative to the given application of location detection in a structure, thus suggesting high confidence in the Accuracy value.**

**Code Snippet for Top 1 Feature (Access Point 1)[[27]](#endnote-27)**

#Input: None

#Processing: Trains RF-ML Model

#Output: Generate Confusion Matrix, report for RF-ML Model (Precision, Recall, F1), Accuracy, and OOB Accuracy for #1 MDI Feature ("Access Point 1")

def Print\_Top1Feature():

n\_features = int(input("Select number of features to sample: "))

dataset = pd.read\_csv('WiFi Localization Data.csv')

labels = ["Access Point 2","Access Point 3","Access Point 4","Access Point 5","Access Point 6","Access Point 7"]

dataset.drop(columns=labels,index=None)

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 7].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=12)

classifier = RandomForestClassifier(n\_estimators=100, random\_state=25, oob\_score=True, max\_features=n\_features)

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

print(confusion\_matrix(y\_test,y\_pred))

print(classification\_report(y\_test,y\_pred))

print ("Accuracy is:")

print(accuracy\_score(y\_test, y\_pred))

print ("OOB Accuracy is:")

print(classifier.oob\_score\_)

**Random Forest Explainability (RFEX) Analysis**

Like OOB Accuracy, another measure that is specific to RF-ML Models is the Random Forest Sample Explainer (“RFEX”)[[28]](#endnote-28). RFEX is used for situations when for example: a) one wants to understand why and how the trained RF classified a specific sample for which they know the Ground Truth, critical in forming their trust in RF classification; and b) for ML quality control (e.g., editing of training data where they want to identify and possibly delete outlier training samples or features of marginal quality).[[29]](#endnote-29) In addition to determining the average and standard deviation for features, RFEX calculates a Cohen Distance to determine the separation between features. The Cohen Distance is defined as:

Cohen Distance = ABS(AV+ - AV-)/SDmax.

ABS is a function that calculates an absolute value, while AV+ is the average of feature values for the positive class; AV- is the average of feature values for the negative class. SDmax is the larger standard deviation of the two feature value populations.[[30]](#endnote-30) Of note, a Cohen Distance of less than 0.2 denotes a small separation, from 0.2 to 0.5 indicates a medium, from 0.5 to 0.8 large, and above it a very large separation.[[31]](#endnote-31) While RFEX is typically used for binary classification problems, here we are using it for a four class problem.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | AP1 | AP2 | AP3 | AP4 | AP5 | AP6 | AP7 |
| 1 | AVG | -62.49 | -56.276 | -60.52 | -64.182 | -70.262 | -82.92 | -84.026 |
|  | SD | 3.300862 | 3.252227 | 3.684665 | 3.645196 | 4.615188 | 3.728459 | 3.944165 |
|  |  |  |  |  |  |  |  |  |
|  | Min | -74 | -69 | -73 | -77 | -89 | -97 | -96 |
|  | Max | -55 | -47 | -50 | -54 | -60 | -74 | -74 |
|  |  |  |  |  |  |  |  |  |
| 2 | AVG | -36.924 | -56.108 | -55.902 | -38.066 | -67.614 | -72.644 | -73.498 |
|  | SD | 8.72413 | 3.27686 | 4.177955 | 7.996846 | 5.255115 | 4.665275 | 4.795831 |
|  |  |  |  |  |  |  |  |  |
|  | Min | -52 | -74 | -71 | -53 | -86 | -89 | -92 |
|  | Max | -10 | -46 | -46 | -11 | -56 | -61 | -63 |
|  |  |  |  |  |  |  |  |  |
| 3 | AVG | -49.726 | -54.886 | -52.784 | -50.702 | -63.232 | -81.368 | -82.392 |
|  | SD | 2.755014 | 3.686215 | 3.11846 | 3.614876 | 3.566457 | 3.602017 | 4.222083 |
|  |  |  |  |  |  |  |  |  |
|  | Min | -63 | -68 | -73 | -63 | -77 | -92 | -93 |
|  | Max | -42 | -45 | -44 | -40 | -54 | -73 | -73 |
|  |  |  |  |  |  |  |  |  |
| 4 | AVG | -60.182 | -55.224 | -50.65 | -61.316 | -49.454 | -87.008 | -86.99 |
|  | SD | 3.011813 | 3.243897 | 4.151864 | 3.854069 | 3.591892 | 3.394552 | 3.54387 |
|  |  |  |  |  |  |  |  |  |
|  | Min | -71 | -66 | -60 | -77 | -64 | -96 | -98 |
|  | Max | -52 | -46 | -40 | -52 | -36 | -76 | -78 |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | Cohen 1 & 2 | -3.87617 | -0.05146 | -1.17237 | -4.20251 | -0.53543 | -2.43338 | -2.3978 |
|  | Cohen 1 & 3 | -4.19839 | -0.39988 | -2.26641 | -3.71343 | -1.70453 | -0.42337 | -0.39995 |
|  | Cohen 1 & 4 | -0.73046 | -0.32389 | -2.51451 | -0.76405 | -5.03178 | 1.14657 | 0.790534 |
|  | Cohen 2 & 3 | 1.978922 | -0.35039 | -0.8458 | 2.036253 | -0.97576 | 2.093245 | 1.968538 |
|  | Cohen 2 & 4 | 3.563812 | -0.27113 | -1.26101 | 3.703953 | -4.03466 | 3.520854 | 3.199753 |
|  | Cohen 3 & 4 | 3.622667 | 0.097347 | -0.5812 | 2.840712 | -3.84947 | 1.611511 | 1.179653 |

**FIG. 15 is a table reflecting the RFEX analysis for the RF-ML Model.**

**V. AI Ethics Audit and Evaluation**

The RF-ML Model as implemented in a Location Detection (“RF-ML Product”) solution present opportunities and risks for our company. Opportunity in the form of a potentially large value proposition associated with productizing this RF-ML Product, but with corresponding ethical and legal risks. Ethical risk, and failures, arising from this RF-ML Product can result in a lack of public trust [[32]](#endnote-32) that can impact the ability of an RF-ML Product to gain traction in the marketplace thus creating financial risk. Legal risk, and failures, can create additional financial risk in the form of monetary penalties.[[33]](#endnote-33) To navigate ethical risks, it is common in our industry to utilize frameworks that include checklists[[34]](#endnote-34), while legal risk can be address via adherence to legal requirements of the applicable legal regime such as the EU’s General Data Protection Regulation (“GDPR”)[[35]](#endnote-35) and draft Artificial Intelligence Act (“AIA”).[[36]](#endnote-36) For the purpose of this report, we will examine certain legal requirements, and propose an Audit Plan that emphasizes- “Explainability, Transparency and Data Provenance By Design” to mitigate these ethical and legal risks.

**GDPR and User Privacy**

While issues of bias arise with respect to certain ML models[[37]](#endnote-37), here the ethical and legal issue that arises is that of privacy in general, and in one’s home in particular. Specifically, the Location Detection solution will be using the RF-ML Model to predict one’s location in, amongst other places, their home. While there are a number of privacy legal regimes in the world, for the purposes of this report we will look at the GDPR as it has a long history and serves as a regulatory baseline for most privacy related matters world-wide. As applied to the use case contemplated herein (i.e., location detection of mobile devices associated with a person within structures including a home), minimally any processing of the Data Set will considered as the processing Personal Data[[38]](#endnote-38) within the meaning of the GDPR. As such, such processing must be lawful within the meaning of the GDPR such that it must be based upon one of consent, contractual necessity, or other legal basis under the GDPR.[[39]](#endnote-39) While at first blush these lawful processing criteria may seem onerous, our company has in place a robust compliance program such that identifying and documenting this legal basis to process person data should be fairly straight forward. However, in addition to GDPR compliance, the company will need to engage in substantial efforts to market the value of our RF-ML Product, and, more importantly, the measures we undertake to protect the security of the Personal Data generated and provided to the RF-ML Model to enable it to make predictions. One’s home is their sanctuary, and our RF-ML Product should not be seen as intruding on this sanctuary.

**AIA and the RF-ML Model**

In addition to financing risks arising from the failure to comply with GDPR[[40]](#endnote-40), financial risk can also arise from the failure to follow various legal requirements related to the selling of an RF-ML Product. While generally speaking, and from a regulatory perspective, the law in still catching up to AI as a technology[[41]](#endnote-41), in April of 2021 the European Union (“EU”) came out with a draft of the AIA[[42]](#endnote-42). Different from the hypothetical financial risks of an ethical misstep, the AIA outlines actual financial penalties for non-compliance with the AIA.[[43]](#endnote-43) To understand the AIA, and compliance with its terms, it is necessary to outline- i) Use Types under the AIA; and ii) Technical and Auditing Requirements under the AIA.

**Use Types**

As a threshold matter, the AIA’s compliance obligations fall on the party that places the AI product in the EU market.[[44]](#endnote-44) The AIA creates three categories of AI- i) Prohibited AI[[45]](#endnote-45); ii) High-Risk AI[[46]](#endnote-46); and iii) Low Risk AI[[47]](#endnote-47). For the purposes of this proposal we are going to focus on High-Risk AI. An AI product is considered High-Risk AI if it is used as a safety component of a product, or if it is covered by one of 19 specified pieces of EU single market harmonization legislation (e.g., aviation, cars, medical devices)[[48]](#endnote-48). Further, an AI product is high risk, if it is deployed in one of the following industry sectors[[49]](#endnote-49):

* Critical infrastructure where the AI system could put people’s life and health at risk;
* Educational and vocational settings where the AI system could determine access to education or professional training;
* Employment, worker management and self-employment;
* Essential private and public services, including access to financial services such as credit scoring systems;
* Law enforcement;
* Migration, asylum and border control, including verifying the authenticity of travel documents; or
* The administration of justice.

**Technical and Audit Requirements**

With respect to these High-Risk areas, the AIA requires that the following audits be performed[[50]](#endnote-50):

* Creating and maintaining a risk management system for the entire life cycle of the system;
* Testing the system to identify risks and determine appropriate mitigation measures, and to validate that the system runs consistently for the intended purpose, with tests made against prior metrics and validated against probabilistic thresholds;
* Establishing appropriate data governance controls, including the requirement that all training, validation, and testing datasets be complete, error-free, and representative;
* Detailed technical documentation, including around system architecture, algorithmic design, and model specifications;
* Automatic logging of events while the system is running, with the recording conforming to recognized standards;
* Designed with sufficient transparency to allow users to interpret the system’s output;
* Designed to maintain human oversight at all times and prevent or minimize risks to health and safety or fundamental rights, including an override or off-switch capability.

**Proposed AI Audit Plan for RF-ML Product**

Our proposed Audit Plan is as follows:

**Product Life Cycle**- Creating and maintaining a risk management system for the entire life cycle of the RF-ML Product

**Conception-**

Designed with sufficient transparency to allow users to interpret the system’s output

Designed to maintain human oversight at all times and prevent or minimize risks to health and safety or fundamental rights, including an override or off-switch capability

**Production-**

Testing the system to identify risks and determine appropriate mitigation measures, and to validate that the system runs consistently for the intended purpose, with tests made against prior metrics and validated against probabilistic thresholds

Establishing appropriate data governance controls, including the requirement that all training, validation, and testing datasets be complete, error-free, and representative

Detailed technical documentation, including around system architecture, algorithmic design, and model specifications

**Market-**

Automatic logging of events while the system is running, with the recording conforming to recognized standards

**FIG. 16 is a graphic of our Audit Plan reflecting an approach of Explainability, Transparency and Data Provenance By Design.**

**Product Life Cycle- Internal Management System**

Our Audit Plan contemplates an audit regime that covers the complete product life cycle for the RF-ML Product. To manage the entire product life cycle for this RF-ML Product, an internal management system would need to be developed (or purchased). Further, a team would need to be hired and trained to use such an internal management system. This team would not only include IT staff, but also business and legal team members. The cost of developing or procuring such a system, and hiring and training a team would need to be measured against the aforementioned ethical and financial risks presented by non-compliance.[[51]](#endnote-51)

**Conception-** **Explainability, Transparency and Data Provenance By Design**

As threshold matter, to implement most any of concepts outlined in the Audit Plan, an approach of “Explainability, Transparency and Data Provenance By Design” would need to taken by the company[[52]](#endnote-52) as many of the key point of this plan cannot be retroactively built into an RF-ML Product. Meaning starting with the conception, design phase of the RF-ML Product, the ability to meet the ethical and legal requirements must built into the design itself. For example, if users of the AI product are to be able to interpret its output, then the output must be human readable (or understandable). Further, if the product is to be overseen by humans at all times and include the ability to shut off, then there must be some sort of remote monitoring capability built into the RF-ML Product.

**Production- Extended Beta Periods**

As to Production, the RF-ML Product will need to be developed and tested over an extended beta period. Specifically, while, for example, the RF-ML Product may work well using testing data derived from a training set this data can carry with it assumptions about use cases and applications that may not be reflected in the actual use environment. To address this possibility, an extended beta testing period in a real world environment may need to be used.[[53]](#endnote-53) For example, the reported bias in certain facial recognition software may have been caught, if only the beta testing period had been extended and different faces presented to the software.[[54]](#endnote-54) While extending beta periods may delay monetization of the RF-ML Product, it will help us to launch products that "run consistently for the intended purpose,” and which use data sets that are ”complete, error-free, and representative.”[[55]](#endnote-55)

**Market- Wireless Connectively**

Regarding the taking to Market of our RF-ML Product, not only will we need an internal management system, but such a system will need the ability to remotely monitor the RF-ML Product performance to ensure it is “conforming to recognized standards”[[56]](#endnote-56). The most straight forward approach to remote monitoring is via a wireless mechanism, where the RF-ML Product does not reside on a server. If not currently part of our AI product roadmap, such an approach would not only ensure compliance with the AIA, but it would also allow us to obtain other metrics around our AI product that could help us optimize this product or otherwise add value for not only our company but the customer.[[57]](#endnote-57)

**Explainability**

Applying the concept of “Explainability, Transparency and Data Provenance By Design,” Explainability (or “XAI”) is AI in which humans can understand the decisions or predictions made by the AI.[[58]](#endnote-58) XAI can be contrasted with black box AI where even AI product designers cannot explain why an AI arrived at a specific decision.[[59]](#endnote-59) Legal requirements such as the AIA aside, the value of XAI is its ability to mitigate ethical risks. Specifically, if the performance results of our AI product cannot be explained, and if these results constitute a negative outcome for a person (e.g., someone does not receive a loan, or is denied employment), then our inability to explain how our RF-ML Product arrived such performance results could be deemed unfair by the public. As a result of this perceived unfairness, our AI product may not gain traction in the marketplace.

**Transparency**

Closely related to Explainability is the concept of Transparency.[[60]](#endnote-60) Generally speaking, it means providing explanations of algorithmic models and decisions that are comprehensible for the user and which can also be taken as a broader socio-technical and normative ideal of openness[[61]](#endnote-61). Where Explainability relates to how the output of the RF-ML Product created (or arrived at), Transparency relates to a description of the actual mechanisms used to arrive at this outcome. For example, whereas through Explainability a customer may learn what data is used to train and test our RF-ML Product, and how this training and testing translates into the actual real world use of the RF-ML Product, Transparency discloses how this data is actually processed by the AI product in at least a real work use. Like Explainability, a failure of our RF-ML Product to meet this Transparency requirement can result in ethical risks to the company.

**Data Provenance**

Data Provenance means the conducting of inquires related to the history of data sets used to train, test, and use our AI products[[62]](#endnote-62). These inquiries may relate to the gathering, labeling of data, and include questions such as- “Is the data complete or is it missing values?”, or “Is the data balanced across labels?” By instituting processes to manage Data Provenance, not only can mistakes made by our AI products be understood, but as with Explainability and Transparency, ethical risk can be reduced. For example, if one of our RF-ML Product were to misclassify[[63]](#endnote-63), having a process to manage Data Provenance would assist us in resolving this misclassification. Of note, however, due to the central, differentiating nature that data plays in training our AI products, we should only disclose the specifics (e.g., at the feature level) of our data where absolutely required to mitigate ethical or legal risk.

**VI. Summary and Recommendations**

Our recommendation is to move forward with the productization of the RF-ML Product. This recommendation is based upon the following.

**RF-ML Performance**

The RF-ML Model trained and tested on the “Wireless Indoor Localization Data Set” performed better than other ML models at making location detection predictions.[[64]](#endnote-64) The general predictive Accuracy of the RF-ML Model is nearly 100%, while other ML models are less than this number.

**High Confidence in Accuracy Metric**

Using multiple criteria including Confusion Matrices, Precision, Recall, F1 Score, OOB Accuracy, and RFEX Scores it has been shown that the RF-ML Model’s Accuracy is not only high, but the FP, FN rates are low.[[65]](#endnote-65) Thus, high confidence in the Accuracy has been demonstrated.

**Ability to Mitigate Ethical, Legal and Business Risks**

Through leveraging existing compliance programs related to GDPR, and initiating a new compliance programs for AIA via our proposed Audit Plan, ethical, legal and business risks can be mitigated (or reduced).[[66]](#endnote-66) However, a compliance program and extended beta testing may need to be implemented to de-risk the sale of the RF-ML Product to the general public.

**VII. Summary of Team Member Contributions**

Theodore McCullough- Drafting of Project Report, slides, and limited coding

Ishank Aggarwal- Coding

Alex Rabanes- Review of Project Report and slides.

**VIII. Appendix**

#Theodore McCullough, Ishank Aggarwal, Alex Rabanes

#CSC 859. Fall 2022

#Team Project

#Team #1

import sklearn

import numpy as np

import csv

from sklearn import metrics

from sklearn.ensemble import RandomForestClassifier

from sklearn import tree

import joblib

import matplotlib.pyplot as plt

import itertools

from itertools import cycle

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import f1\_score

from sklearn.metrics import ConfusionMatrixDisplay

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import RocCurveDisplay

from sklearn.inspection import permutation\_importance

from sklearn.metrics import roc\_auc\_score

import pandas as pd

import sys

#Input: None

#Processing: Loads wifi data set, extracts‐ Features, and labels

#Output: Label train and test sets, Feature train and test set

def Process\_DataSet():

dataset = pd.read\_csv('WiFi Localization Data.csv')

X = dataset.iloc[:, :-1].values #Creates feature set

y = dataset.iloc[:, 7].values # Create label set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,test\_size=0.2, random\_state=100)

return [X,X\_train, X\_test, y\_train, y\_test]

#Input: None

#Processing: Helper function and called by Create\_Forest\_Objects function

#Output: Returns a Random Forest Tree Object

def Create\_Forest(forest\_size, n\_features):

return RandomForestClassifier(n\_estimators = forest\_size, criterion='gini',

max\_depth=None, min\_samples\_split=2,min\_samples\_leaf=1,

min\_weight\_fraction\_leaf=0.0,max\_features=n\_features, max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0,bootstrap=True,

oob\_score=True, n\_jobs=None,random\_state=None, verbose=0,

warm\_start=False, class\_weight=None,ccp\_alpha=0.0, max\_samples=None)

#Input: Feature Training Set, Label Training Set, Random Forest Object, Number of trees in forest

#Processing: Plots Random Forest Object, displays, and stores into .png file

#Output: Displayed Random Forest Object, and .png Forest file

def Print\_Forest(x\_train, y\_train, clf, forest\_size):

fn=x\_train

cn=y\_train

fig, axes = plt.subplots(nrows = 1,ncols = forest\_size,figsize = (10,2), dpi=1000)

if forest\_size == 1:

tree.plot\_tree(clf.estimators\_[0], max\_depth=None, feature\_names=["Access Point 1", "Access Point 2", "Access Point 3", "Access Point 4", "Access Point 5", "Access Point 6", "Access Point 7"], class\_names=["Room 1", "Room 2", "Room 3","Room 4"], label='all',

filled=False, impurity=True, node\_ids=False,proportion=False, rounded=False,

precision=3, ax=None, fontsize=None);

axes.set\_title('Tree: ' + str(1), fontsize = 11)

else:

for index in range(0, forest\_size):

tree.plot\_tree(clf.estimators\_[index], max\_depth=None,feature\_names=["Access Point 1", "Access Point 2", "Access Point 3", "Access Point 4", "Access Point 5", "Access Point 6", "Access Point 7"],

class\_names=["Room 1","Room 2", "Room 3", "Room 4"], label='all',filled=False, impurity=True, node\_ids=False,proportion=False, rounded=False,precision=3, ax=axes[index], fontsize=None)

axes[index].set\_title('Tree: ' + str(index + 1), fontsize = 11)

fig.savefig('rf\_trees.png')

#Input: Random Forest Object (clf), Feature Test Set (x\_test), Label Test Set (y\_test)

#Processing: Creates labeled Confusion Matrix for Random Forest Object

#Output: PLots Confusion Matrix for Random Forest Object

def Print\_Confusion\_Matrix(clf, x\_test, y\_test):

length = len(x\_test)

y\_pred = []

for index in range(0, length):

y\_pred.append(clf.predict(x\_test[[index]]))

cm = confusion\_matrix(y\_test, y\_pred, labels=clf.classes\_, normalize = None)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm,display\_labels=["Room 1","Room 2", "Room 3", "Room 4"])

disp.plot(cmap=plt.cm.Blues)

plt.show()

#Input: Trained Random Forest Object (clf), Feature Test Set (x\_test), Label Test Set (y\_test)

#Processing: Creates an array (y\_pred) of predicted labels from x\_train, provides y\_pred and y\_train to F1 function, generates OOB accuracy score for clf

#Output: Prints f1 score and OOB accuracy scores generally for clf

def Print\_OOB\_F1(x\_test, y\_test, clf):

length = len(x\_test)

y\_pred = []

for index in range(0, length):

y\_pred.append(clf.predict(x\_test[[index]]))

f1 = f1\_score(y\_test, y\_pred, labels= [1,2,3,4], average=None)

f1\_add = f1[0] + f1[1]

f1\_average = f1\_add/2

print("F1 score generally:")

print(f1\_average)

print("OOB score generally:")

print(clf.oob\_score\_)

#Input: Feature Test Set (x\_test), Label Test Set (y\_test), Trained Random Forest Object (clf)

#Processing: Creates an array (y\_pred) of predicted labels from x\_train, provides y\_pred and y\_train to accuracy\_score function

#Output: Prints Accuracy score for Trained Random Forest Object

def Print\_Accuracy(x\_test, y\_test, clf):

length = len(x\_test)

y\_pred = []

for index in range(0, length):

y\_pred.append(clf.predict(x\_test[[index]]))

accuracy = accuracy\_score(y\_test, y\_pred, normalize=True)

print("Accuracy is:")

print(accuracy)

#Input: Feature Test Set (x\_test), Label Test Set (y\_test), Trained Random Forest Object (clf)

#Processing: Creates an array (y\_pred) of predicted labels from x\_train, provides y\_pred and y\_train to classfication\_report function

#Output: Prints classification report with F1, Recall, and Accurcy Score

def Print\_Report(x\_test, y\_test, clf):

length = len(x\_test)

y\_pred = []

for index in range(0, length):

y\_pred.append(clf.predict(x\_test[[index]]))

target\_names = ["Room 1", "Room 2", "Room 3", "Room 4"]

print(classification\_report(y\_test, y\_pred, target\_names=target\_names,

sample\_weight=None, digits = 5))

#Input: Label Test Set (y\_test), Feature Test Set (x\_test), Trained Random Forest Object (clf)

#Processing: Uses input to determine true positive rate over false positive rate at different classification thresholds

#Output: Represents ROC values

def Print\_ROC(y\_test, x\_test, clf):

y\_prob = clf.predict\_proba(x\_test)

macro\_roc\_auc\_ovo = roc\_auc\_score(y\_test, y\_prob, multi\_class="ovo", average="macro")

weighted\_roc\_auc\_ovo = roc\_auc\_score(

y\_test, y\_prob, multi\_class="ovo", average="weighted"

)

macro\_roc\_auc\_ovr = roc\_auc\_score(y\_test, y\_prob, multi\_class="ovr", average="macro")

weighted\_roc\_auc\_ovr = roc\_auc\_score(

y\_test, y\_prob, multi\_class="ovr", average="weighted"

)

print(

"One-vs-One ROC AUC scores:\n{:.6f} (macro),\n{:.6f} "

"(weighted by prevalence)".format(macro\_roc\_auc\_ovo, weighted\_roc\_auc\_ovo)

)

print(

"One-vs-Rest ROC AUC scores:\n{:.6f} (macro),\n{:.6f} "

"(weighted by prevalence)".format(macro\_roc\_auc\_ovr, weighted\_roc\_auc\_ovr)

)

#Input: Trained Random Forest model(clf)

#Processing: Loads feature names, selects top feature types used by clf

#Output: Bar graph of top feature types for clf, and data for top features types

def Print\_MDI(clf):

feature\_names = ["Access Point 1","Access Point 2","Access Point 3","Access Point 4","Access Point 5","Access Point 6","Access Point 7"]

mdi\_importances = pd.Series(clf[-1].feature\_importances\_, index=feature\_names).sort\_values(ascending=True)

ax = mdi\_importances.plot.barh()

ax.set\_title("Random Forest Feature Importances (MDI)")

ax.figure.tight\_layout()

mdi\_importances.sort\_values(ascending=False, inplace=True)

print(mdi\_importances)

#Processes data, creates and trains Random Forest

forest\_size = int(input("Select number of estimators (trees) (enter 0 to end): "))

n\_features = int(input("Select number of features to sample: "))

if forest\_size == 0:

sys.exit("Bye Bye!!")

else:

clf = Create\_Forest(forest\_size, n\_features)

X, X\_train, x\_test, y\_train, y\_test = Process\_DataSet()

clf.fit(X\_train, y\_train)

#Input: None

#Processing: Trains RF-ML Model

#Output: Generate Confusion Matrix, report for RF-ML Model (Precision, Recall, F1), Accuracy, and OOB Accuracy for #1 MDI Feature ("Access Point 1")

def Print\_Top1Feature():

n\_features = int(input("Select number of features to sample: "))

dataset = pd.read\_csv('WiFi Localization Data.csv')

labels = ["Access Point 2","Access Point 3","Access Point 4","Access Point 5","Access Point 6","Access Point 7"]

dataset.drop(columns=labels,index=None)

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 7].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=12)

classifier = RandomForestClassifier(n\_estimators=100, random\_state=25, oob\_score=True, max\_features=n\_features)

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

print ("Confusion Matrix:\n")

print(confusion\_matrix(y\_test,y\_pred))

print ("\nClassification Report:\n")

print(classification\_report(y\_test,y\_pred))

print ("Accuracy is:")

print(accuracy\_score(y\_test, y\_pred))

print ("OOB Accuracy is:")

print(classifier.oob\_score\_)

#Input: None

#Processing: Trains RF-ML Model

#Output: Generate Confusion Matrix, report for RF-ML Model (Precision, Recall, F1), Accuracy, and OOB Accuracy for #1 & #2 MDI Features ("Access Point 1", "Access Point 5")

def Print\_Top2Features():

n\_features = int(input("Select number of features to sample: "))

dataset = pd.read\_csv('WiFi Localization Data.csv')

labels = ["Access Point 2","Access Point 3","Access Point 4","Access Point 6","Access Point 7"]

dataset.drop(columns=labels,index=None)

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 7].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=100)

classifier = RandomForestClassifier(n\_estimators=100, random\_state=25, oob\_score=True, max\_features=n\_features)

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

print ("Confusion Matrix:\n")

print(confusion\_matrix(y\_test,y\_pred))

print ("\nClassification Report:\n")

print(classification\_report(y\_test,y\_pred))

print ("Accuracy is:")

print(accuracy\_score(y\_test, y\_pred))

print ("OOB Accuracy is:")

print(classifier.oob\_score\_)

#Input: None

#Processing: Trains RF-ML Model

#Output: Generate Report Confusion Matrix, report for RF-ML Model (Precision, Recall, F1), Accuracy, and OOB Accuracy for #1, #2, & #3 MDI Features ("Access Point 1", "Access Point 5", "Access Point 4")

def Print\_Top3Features():

n\_features = int(input("Select number of features to sample: "))

dataset = pd.read\_csv('WiFi Localization Data.csv')

labels = ["Access Point 2","Access Point 3","Access Point 6","Access Point 7"]

dataset.drop(columns=labels,index=None)

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 7].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=20)

classifier = RandomForestClassifier(n\_estimators=100, random\_state=25, oob\_score=True, max\_features=n\_features)

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

print ("Confusion Matrix:\n")

print(confusion\_matrix(y\_test,y\_pred))

print ("\nClassification Report:\n")

print(classification\_report(y\_test,y\_pred))

print ("Accuracy is:")

print(accuracy\_score(y\_test, y\_pred))

print ("OOB Accuracy is:")

print(classifier.oob\_score\_)

while(True):

print()

print ("Select analysis type: ")

print ("1. Enter '1' to print Forest: ")

print ("2. Enter '2' to print Confusion Matrix: ")

print ("3. Enter '3' to print OOB & F1 score: ")

print ("4. Enter '4' to print Accuracy: ")

print ("5. Enter '5' to print Classification Report: ")

print ("6. Enter '6' to print ROC: ")

print ("7. Enter '7' to print MDI Graph: ")

print ("8. Enter '8' to print Classification Report for Top 1 Feature: ")

print ("9. Enter '9' to print Classification Report for Top 2 Features: ")

print ("10. Enter '10' to print Classification Report for Top 3 Features: ")

print ("11. Enter '11' to End Program: ")

menu\_selection = int(input("Select Input Options: "))

if menu\_selection == 1:

Print\_Forest(X\_train, y\_train, clf, forest\_size)

elif menu\_selection == 2:

Print\_Confusion\_Matrix(clf, x\_test, y\_test)

elif menu\_selection == 3:

Print\_OOB\_F1(X\_train, y\_train, clf)

elif menu\_selection == 4:

Print\_Accuracy(X\_train, y\_train, clf)

elif menu\_selection == 5:

Print\_Report(X\_train, y\_train, clf)

elif menu\_selection == 6:

Print\_ROC(y\_test, x\_test, clf)

elif menu\_selection == 7:

Print\_MDI(clf)

elif menu\_selection == 8:

Print\_Top1Feature()

elif menu\_selection == 9:

Print\_Top2Features()

elif menu\_selection == 10:

Print\_Top3Features()

elif menu\_selection == 11:

sys.exit("Bye Bye!!")

else:

print ("Entered wrong selection, try again")

print (" ")

**IX. References**

1. “User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm with Neural Networks,” Rohra et al. <<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>>

   (last visited 30 November 2022) [↑](#endnote-ref-1)
2. Id. [↑](#endnote-ref-2)
3. It would be valuable to a marketer to know the amount of time a person spent in a kitchen as this would allow the market to do more targeted marketing for certain goods and services. [↑](#endnote-ref-3)
4. See supra note 1, pg. 286 (“This This can be used for objectives like locating users in smart home systems, locating criminals in bounded regions, obtaining the count of users on an access point etc.”) [↑](#endnote-ref-4)
5. While not within the scope of this report, a follow on report might cover the hypothetical Total Addressable Market (TAM) for this Location Detection solution. [↑](#endnote-ref-5)
6. See supra note 1, pg. 286 (describing the merits of using Wi-Fi Access Points relative to other location detection means such as mobile devices plus GPS, or mobile devices plus Blue Tooth). [↑](#endnote-ref-6)
7. See supra note 1, pg. 293 (Evaluating the accuracy of predictions made by Machine Learning Models including PSO-Neural Networks, GSA-Neural Networks, PSOGSA-Neural Networks, FPSOGSA-Neural Networks, Support Vector Machines, and Naïve Bayes, the predictions based upon data in the form of signal strength values generated by Wi-Fi Access Points receiving signals from mobile devices). [↑](#endnote-ref-7)
8. The problem of local minima is well known in the ML field and arises when an ML model makes effectively a global prediction based upon a local data. Typically, the problem of local minima is solved through a “random move” that allows the ML model to make the prediction based upon additional data outside that of the local minima. (See generally, “Artificial Intelligence: A Modern Approach (2nd Edition)” by Russell and Norvig, pgs. 110-119 (describing the use of Simulated Annealing, Local Beam Search, & Genetic Algorithms to solve the problem of local minima). [↑](#endnote-ref-8)
9. See supra note 1, pg. 287 (“Considering these factors, we propose the FPSOGSA that overcomes the possibilities of trapping itself in the local minima and enhances the probability of a higher convergence rate.”) [↑](#endnote-ref-9)
10. See <<https://gdpr-info.eu/issues/personal-data/>> (last visited 30 November 2022) (defining “Personal Data” as data being subject to the EU’s General Data Protection Regulation (GDPR)). [↑](#endnote-ref-10)
11. See <<https://archive.ics.uci.edu/ml/datasets/Wireless+Indoor+Localization>> (last visited on 30 November 2022) [↑](#endnote-ref-11)
12. 500 samples classified as “1”, 500 samples classified as “2”, 500 samples classified as “3”, and 500 samples classified as “4”. [↑](#endnote-ref-12)
13. See < <https://en.wikipedia.org/wiki/Random_forest>> (last visited 1 December 2022) [↑](#endnote-ref-13)
14. As previously stated, another advantage of RF ML is its ability to avoid the problem of local minima. (See supra, pg. 2) [↑](#endnote-ref-14)
15. See “How the machine ‘thinks’: Understanding opacity in machine learning algorithms”, Jenna Burrell <<https://journals.sagepub.com/doi/full/10.1177/2053951715622512>> (last visited 1 December 2022) (“They are opaque in the sense that if one is a recipient of the output of the algorithm (the classification decision), rarely does one have any concrete sense of how or why a particular classification has been arrived at from inputs.”) [↑](#endnote-ref-15)
16. A “Split” is the decision to choose one child node v. another is a tree. We tested various split values of 2, 4, 6, and determine that 6 provided the best performance. [↑](#endnote-ref-16)
17. A “Cut Off” value is the threshold value based upon which a decision is made (e.g., if 50% (.5) or greater probability, then one node in the tree is chosen versus another. See <<https://www.graphpad.com/guides/prism/latest/curve-fitting/reg_mult_logistic_gof_classification.htm>> (last visted 11, December 2022) [↑](#endnote-ref-17)
18. See <<https://en.wikipedia.org/wiki/Ground_truth>> (last visited 11 December 2022). [↑](#endnote-ref-18)
19. See < <https://developers.google.com/machine-learning/crash-course/classification/accuracy>> (last visited 8 December 2022) [↑](#endnote-ref-19)
20. This 3% percent value is arrived at by identifying the number FN or FP values relative to the correctly classified values. So, for example, 3/94 = ~3%. [↑](#endnote-ref-20)
21. See < <https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/>> (last visited 8 December 2022). [↑](#endnote-ref-21)
22. See < <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>> (last visited 8 December 2022). [↑](#endnote-ref-22)
23. See < <https://en.wikipedia.org/wiki/F-score>> (last visited 9 December 2022) [↑](#endnote-ref-23)
24. See supra pg. 20 (FIG. 3). [↑](#endnote-ref-24)
25. See < <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>> (last visited 9 December 2022). [↑](#endnote-ref-25)
26. See <<https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance.html>> (last visited 9 December 2022). [↑](#endnote-ref-26)
27. In the interest of brevity code snippet for Top2 and Top 3 was omitted, but is provided in Appendix [?]. [↑](#endnote-ref-27)
28. See “Random Forest Model and Sample Explainer for non-experts in Machine Learning – Two Case Studies” D. Petkovic et al., pg.1. [↑](#endnote-ref-28)
29. Id. pg. 5. [↑](#endnote-ref-29)
30. Id. [↑](#endnote-ref-30)
31. Id. [↑](#endnote-ref-31)
32. “Ethics-Based Auditing of Automated Decision-Making Systems: Nature, Scope, and Limitations,” pg.1, Mokander et al. (<https://arxiv.org/abs/2110.10980#:~:text=Ethics-Based%20Auditing%20of%20Automated%20Decision-Making%20Systems%3A%20Nature%2C%20Scope%2C,and%20the%20natural%20environment%20are%20increasingly%20being%20automated>) (last visited 16 November 2022). [↑](#endnote-ref-32)
33. “Laying Down Harmonized Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts,” pg. 83, 2021/0106 (COD) (<https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence>) (last visited 16 November 2022). [↑](#endnote-ref-33)
34. See “Guardrails to Democratize AI: Creating an AI Checklist to streamline Idea to Production” (<https://towardsdatascience.com/the-ai-checklist-fe2d76907673>) (last visited on 17 November 2022) [↑](#endnote-ref-34)
35. See < <https://en.wikipedia.org/wiki/General_Data_Protection_Regulation>> (last visited 9 December 2022). [↑](#endnote-ref-35)
36. See supra note 29, pg. 1. [↑](#endnote-ref-36)
37. See infra note 47. [↑](#endnote-ref-37)
38. See < <https://gdpr-info.eu/art-4-gdpr/>> (“‘personal data’ means any information relating to an identified or identifiable natural person (‘data subject’); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person”). [↑](#endnote-ref-38)
39. See < <https://gdpr-info.eu/art-6-gdpr/>> (last visited 9 December 2022) [↑](#endnote-ref-39)
40. See < <https://gdpr.eu/fines/>> (last visited 9 December 2022). [↑](#endnote-ref-40)
41. See “Legislation Related to Artificial Intelligence“ (<https://www.ncsl.org/research/telecommunications-and-information-technology/2020-legislation-related-to-artificial-intelligence.aspx>) (last visited 17 November 2022) (describing law initiatives at the state level in the US). [↑](#endnote-ref-41)
42. See supra note 29, pg. 1. [↑](#endnote-ref-42)
43. Id. pg. 83. [↑](#endnote-ref-43)
44. “The Artificial Intelligence Act: A Quick Explainer,” B. Mueller (<https://datainnovation.org/2021/05/the-artificial-intelligence-act-a-quick-explainer/>) (last visited 16 November 2022). [↑](#endnote-ref-44)
45. See supra note 29, pg. 12. [↑](#endnote-ref-45)
46. Id. pgs. 27-29. [↑](#endnote-ref-46)
47. Id. [↑](#endnote-ref-47)
48. Id. [↑](#endnote-ref-48)
49. Id. [↑](#endnote-ref-49)
50. Id. [↑](#endnote-ref-50)
51. Failure to follow the AIA can result in substantial financial risk to our company. For example, non-compliance with prohibited uses and data governance obligations is punishable with a fine of up to €30M or 6 percent of the prior year’s worldwide annual revenue (whichever is higher), while for high-risk AI applications, the ceiling is €20M or 4 percent of revenue. The supply of incorrect, incomplete, or misleading information to national competent bodies responsible or administering compliance with the AIA is subject to a fine of up to €10M or 2 percent of revenue. See supra note 29. [↑](#endnote-ref-51)
52. Compare Privacy by Design as a concept in the EU’s General Data Protection Regulation (<https://en.wikipedia.org/wiki/Privacy_by_design>) (last visited 17 November 2022). [↑](#endnote-ref-52)
53. See supra note 5, pg. 2 (citing examples of bias in facial recognition software). [↑](#endnote-ref-53)
54. See (<https://www.codedbias.com/>) (last visited 17 November 2022) (web site for award winning film on bias facial recognition software). [↑](#endnote-ref-54)
55. See supra note 26. [↑](#endnote-ref-55)
56. These “recognized standards” basically means that the product is functioning according to its intended purpose. (See supra note 2, pg. 50.). [↑](#endnote-ref-56)
57. Of course, such data collection would need to comply with privacy legal regimes such as the General Data Protection Regulation (GDPR) (see <https://en.wikipedia.org/wiki/General_Data_Protection_Regulation>) (last visited 20 November 2022), or the California Consumer Privacy Act (CPRA) (see <https://en.wikipedia.org/wiki/California_Consumer_Privacy_Act>) (last visited 20 November 2022). [↑](#endnote-ref-57)
58. See (<https://en.wikipedia.org/wiki/Explainable_artificial_intelligence>) (last visited 17 November 2022). See also “High Level Experts Group On Artificial Intelligence: Ethical Guidelines for Trustworthy AI” (https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai) (last visited 16 November 2022), pg. 20 (Explainability concerns the ability to explain both the technical processes of an AI system and the related human decisions (e.g. application areas of a system”). [↑](#endnote-ref-58)
59. Id. [↑](#endnote-ref-59)
60. “High Level Experts Group On Artificial Intelligence: Ethical Guidelines for Trustworthy AI” (https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai) (last visited 16 November 2022), pg. 20 (“This [transparency] requirement is closely linked with the principle of explicability and encompasses transparency of elements relevant to an AI system: the data, the system and the business models.”). [↑](#endnote-ref-60)
61. See “Chapter 4: Should we know how AI works” (<https://ethics-of-ai.mooc.fi/chapter-4/2-what-is-transparency>) (last visited 17 November 2022). [↑](#endnote-ref-61)
62. Id. [↑](#endnote-ref-62)
63. For example, misclassify based upon skin color, or some other biological characteristic. [↑](#endnote-ref-63)
64. See supra pg. 1. [↑](#endnote-ref-64)
65. See supra pgs. 5-21. [↑](#endnote-ref-65)
66. See supra pgs. 22-26. [↑](#endnote-ref-66)