PROJECT REPORT

Estimating occupancy of rooms in a facility for effective resource management

Course: CS225 Spatial Computing

Category: Spatial Application

12.02.2021

Rohit Ramesh Kumashi

Subir Shirish Jadhav

Vineet Madhav Naique Dhaimodker

INTRODUCTION

Occupancy prediction is the determination of the presence of people in rooms of a building. Applications of occupancy prediction find use in efficient allocation of classrooms/staff rooms/conference halls and subsequent plans to save energy whenever possible. Our project tackles the question of whether persons occupy a room or not based on Spatio-temporal factors like CO2 concentration, room air humidity, room temperature, and luminosity

MILESTONES AND DELIVERABLE STATUS

I. Data preparing/cleaning

II. Choosing an appropriate model to predict the footprint of a building/room

III. Training, Evaluating, parameter tuning of models

IV. Evaluation Results and Conclusion

Timeline

week 4, 5: Deliverable I

week 6: Deliverable II

week 7: Deliverable III

week 8: Deliverable IV

Deliverable I:

- Combine the data of each room (51 rooms) to create one single dataset to get a holistic view of the data.
- Removed the outliers for the features Co2, humidity, light, temperature
- Cleaned the data set, removing NaN values and blank data cells. Got about 3.6 million data points, 6 columns after removing NaN values indexed on room numbers
- Under sampled the major category data to match the minor category data in order to overcome bias in training models

Deliverable II:

- Did the literature survey of the mentioned articles where we studied more about the problem at hand and similar problems and also various approaches that have been used to solve such class of problems
- We studied about approaches like Random Forest, Decision Trees, K nearest neighbors, etc and chose them as suitable solutions for our problem statement
- We chose following six Classification Algorithms:
  - Random Forest
  - Naive Bayes
  - K Nearest Neighbours
  - Decision Tree
  - Multilayer Perceptron Classifier
  - Logistic Regression

Deliverable III:

- We used spatio-temporal features like Co2, temperature, humidity, light for training our model
- The Target Feature is pir with categories as 0 if no occupancy detected and 1 if occupancy detected
- Split the data for training and testing the models with 70% for Training and 30% for Testing

Deliverable IV :

- Executed the above algorithms and used Evaluations Metrics as follows:
  - Accuracy
  - F1 score
  - Precision
  - Recall

- Compared all the six classifiers and concluded that Random forest and Decision Tree classifier had the best performance

TAXONOMY

Academic work

[1] Transfer Learning Approach for Occupancy Prediction in Smart Buildings

[2] A review of studies applying machine learning models to predict occupancy and window-opening behaviors in smart building

[3] Improved thermal comfort modeling for smart buildings: A data analytics study

[4] A comparison of machine learning algorithms for forecasting indoor temperature in smart buildings

[5] Occupancy determination based on time series of CO2 concentration, temperature, and relative humidity

[6] Occupant Behavior Prediction and Real-Time Correction-based Smart Building Energy Optimization

[7] IoT-based Occupants Counting with Smart Building State Variables

[8] Indoor temperature, relative humidity, and CO2 levels assessment in academic buildings with different heating, ventilation, and air conditioning systems

[9] Indoor Air Quality (IAQ) in Two schools, Measurements of Airborne Fungi, Carpet Allergens, CO2, Temperature, and Relative Humidity

	Application/Domain	Model/Algorithm	Features
[1]	Occupancy prediction	LSTM Random forest SVM Transfer learning	Temperature, relative humidity, CO2, motion sensor
[2]	Occupancy prediction Window opening behaviour prediction	Decision tree KNN Random forest logistic regression, etc	temperature, humidity, Wind speed, CO2, etc
[3]	Thermal comfort modeling	Neural Networks linear reg SV regression	HVAC data Temperature humidity outdoor irradiance illuminance
[4]	Indoor temperature prediction	SV regression RNN ELM	AC temperature AC humidity solar radiation
[5]	Occupancy prediction	KNN linear discriminant analysis	Temperature Humidity CO2
[6]	Occupant movement prediction Occupancy prediction	occupant prediction real-time occupant movement correction	movement using sensors
[7]	Occupancy prediction	KNN random forest multi-layer perceptron	Temperature CO2 Lighting ventilated state
[8]	Indoor Air Quality(IAQ) and thermal comfort levels assessment	Descriptive Statistical Analysis SPSS 14	HVAC data Temperature relative humidity CO2
[9]	Indoor Air Quality(IAQ) and thermal comfort levels assessment	Descriptive Analysis Fixed nested analysis Random nested analysis Durbin-Watson Test	HVAC data temperature relative humidity CO2

DATASET

The dataset used is collected from 255 time series sensors, installed on 51 rooms of the four floors of the SDH Hall at the University of California, Berkeley. Each room includes 5 types of measurements:

- CO2 concentration
- room air humidity
- room temperature
- Luminosity
- PIR motion sensor data

The passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view, which measures the occupancy in a room. These readings were collected from 08/23/2013 to 08/31/2013 i.e over the course of a week. All the sensors were sampled once every 5 seconds except for the PIR sensor which was sampled once over every 10 seconds. The data contains timestamps in Unix Epoch Time and the readings from the sensors. More info about the geospatial positioning of the rooms of the building can be found at the below link.

(https://citris-uc.org/about/sutardja-dai-hall/about-facilities/floorplans/)

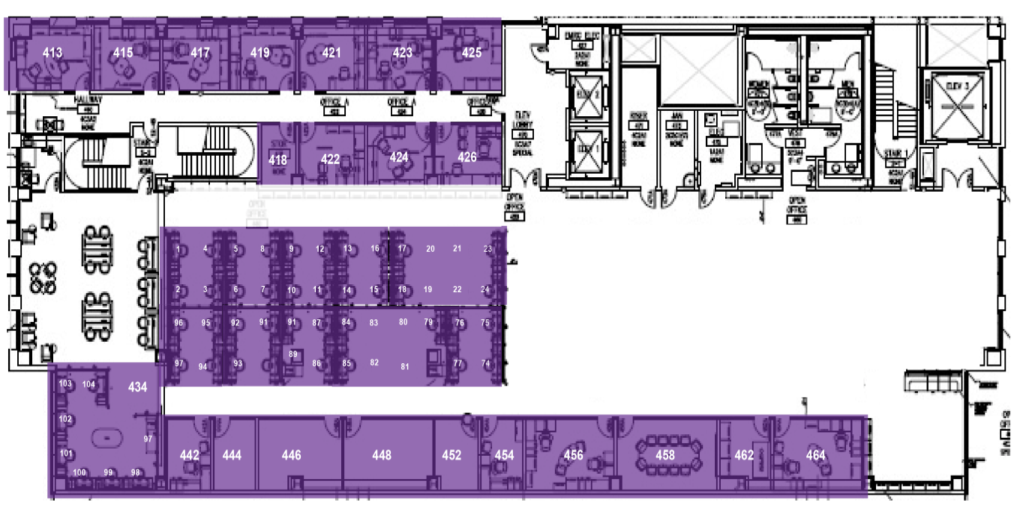


Figure 1. Floor Plan of the 4th Floor

IMPLEMENTATION

Data preparing and cleaning

The dataset consisted of data for 5 features of 51 rooms, each of the feature data was present in a separate csv file and each room had its own folder with 5 files. We combined the readings of each feature of a room and then combined the data for all rooms into a pandas dataframe. This includes data for all features and all rooms as shown below in figure 2.

	Unixtime	co2	humidity	light	pir	temperature
count	3.593902e+06	3.593902e+06	3.593902e+06	3.593902e+06	3.593902e+06	3.593902e+06
mean	1.377483e+09	3.956685e+02	5.544318e+01	8.639833e+01	7.141625e-02	2.399407e+01
std	1.099775e+05	9.347890e+01	4.383837e+00	3.230336e+02	2.575189e-01	2.199300e+01
min	1.377293e+09	8.000000e+00	-4.000000e+00	0.000000e+00	0.000000e+00	-4.010000e+01
25%	1.377388e+09	3.450000e+02	5.262000e+01	3.000000e+00	0.000000e+00	2.241000e+01
50%	1.377480e+09	3.990000e+02	5.536000e+01	4.000000e+00	0.000000e+00	2.310000e+01
75%	1.377572e+09	4.480000e+02	5.824000e+01	2.700000e+01	0.000000e+00	2.373000e+01
max	1.377761e+09	1.315000e+03	7.135000e+01	2.289500e+04	1.000000e+00	5.792700e+02

Figure 2: Descriptive Statistics about the attributes of the dataset

We cleaned the dataset by removing NaN values and blank data cells and we got about 3.6 million data points, 6 columns after removing NaN values indexed on room numbers. Further, we plotted box plots for each feature to check for outliers that may affect the efficiency of our classifier. We then filtered out the features to keep only values that made sense.

We used the following filters:

- Co2< 1000
- Humidity>30
- Light<1000
- Temperature<100

In figures 3 and 4 we can see the before and after box plots for our features. We can see that we were successfully able to discard outliers. In our dataset, we had almost 90% of the data for the not-occupied class. This could have given an incorrect interpretation of the problem. Hence, we randomly sampled the unoccupied class data to make the number of observations of both the classes to be the same and get a balanced class dataset as shown in figure 5. Finally, we plotted a Correlation matrix to get an idea about the dependence of features. The correlation matrix is shown in figure 6.

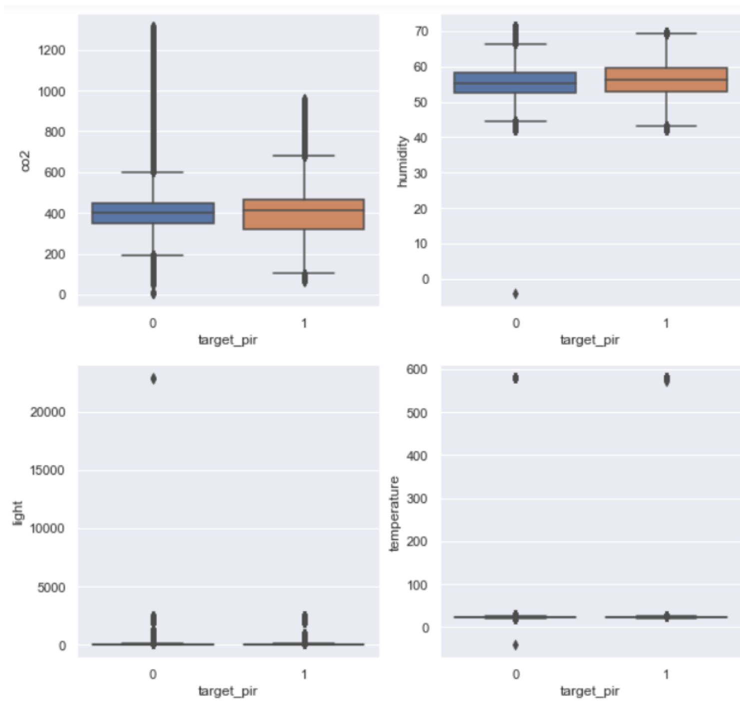


Figure 3. Box plot of features before filtering

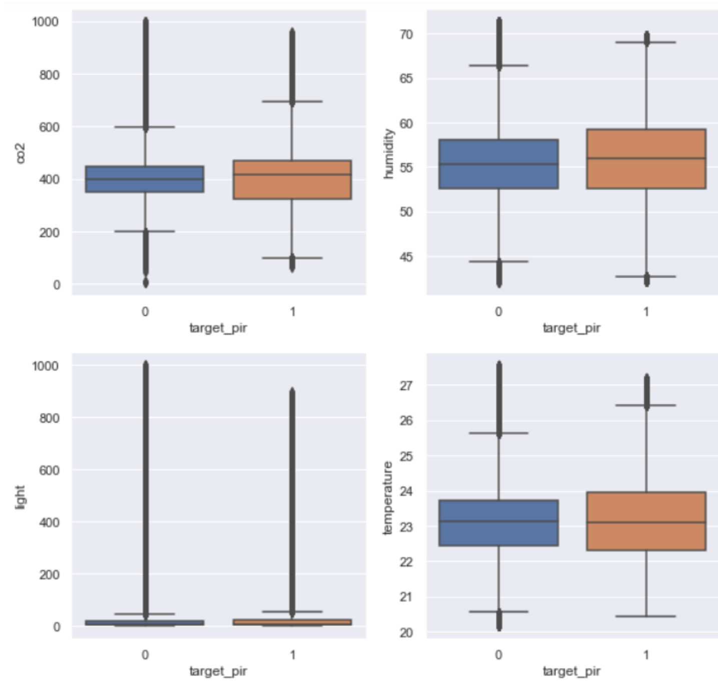


Figure 4. Box plot of features after filtering

```
X_equalized.describe()
```

	co2	humidity	temperature	light
count	475696.000000	475696.000000	475696.000000	475696.00000
mean	398.683531	55.516553	23.186903	39.20515
std	110.370473	4.527482	1.193588	88.51914
min	65.000000	42.080000	20.180000	0.00000
25%	342.000000	52.560000	22.390000	3.00000
50%	408.000000	55.580000	23.100000	5.00000
75%	458.000000	58.610000	23.840000	23.00000
max	999.000000	71.320000	27.550000	989.00000

```
[92] y_equalized.describe()

count    475696.000000
mean          0.500000
std           0.500001
min           0.000000
25%           0.000000
50%           0.500000
75%           1.000000
max           1.000000
Name: target_pir, dtype: float64
```

Figure 5. Descriptive Statistics about the attributes of the Balanced class dataset

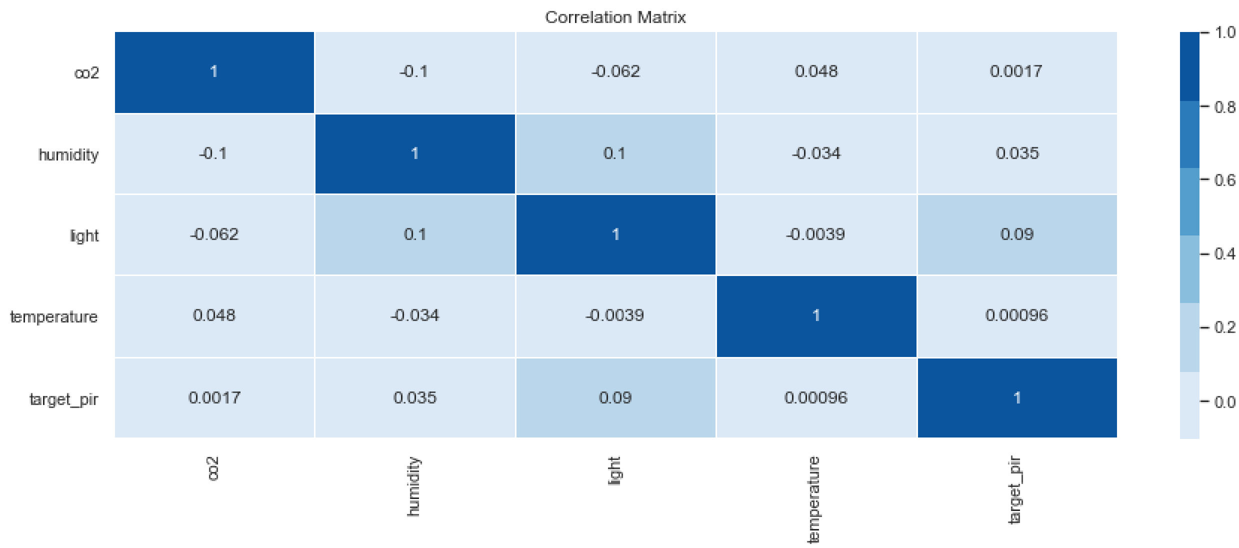


Figure 6. Correlation Matrix

The following pairs of attributes are weakly negatively correlated:-

- Humidity and CO2
- Light and CO2
- Temperature and Humidity
- Temperature and Light

The following pairs of attributes are weakly positively correlated:-

- Temperature and CO2
- Target_pir and CO2
- Light and Humidity
- Target_pir and humidity
- Target_pir and light
- Target_pir and temperature

We can see that all the correlations are weak which implies that each of the features are significantly contributing in the classification and we need to take all the features to get the most accurate model.

TRAINING AND EVALUATION OF MODELS

Based on the studies that we read about in the related work, we chose the following models for our classification task:

- Decision Tree Classifier
- Random Forest Classifier
- K-nearest Neighbours Classifier
- Multi-Layer Perceptron
- Naive Bayes Classifier
- Logistic Regression

In order to check how good our classifier performs, we need to test it on some data, this is where the Train-Test split comes into the picture. Train-Test split is a technique for evaluating the performance of a machine learning algorithm where we split the dataset into training data and testing data. The training data is used to train our model and the testing data can be used to check how well our model performs. The Train-Test split used for our dataset was 70-30, where 70% was training data and 30% was the testing data.

For evaluating the model, we use Accuracy, Precision, Recall and F1 score

Accuracy is the ratio of correctly predicted observation to the total observations

Accuracy = (TP+TN)/(TP+FP+FN+TN)

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations

Precision = TP/(TP+FP)

Recall is the ratio of correctly predicted positive observations to all observations in actual class - yes

Recall = TP/(TP+FN)

F1 score is the weighted average of Precision and Recall

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

RESULTS

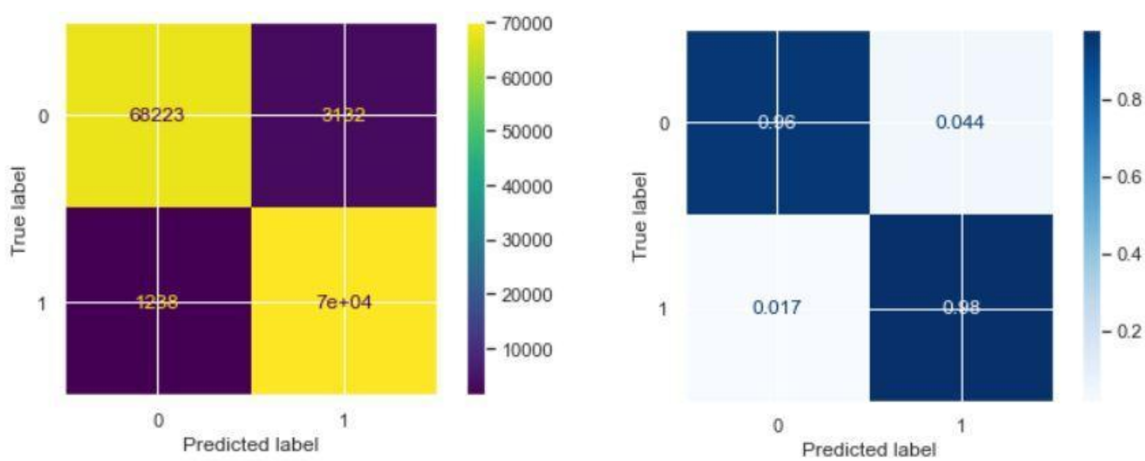
Table 1 summarizes the performance of all the classifiers we have used. The Classifiers have been sorted in descending order of accuracy.

Sr No	Model	Accuracy	f1_score	precision	recall
1	Random Forest	0.969	0.969	0.957	0.982
2	Decision Tree	0.956	0.956	0.957	0.955
3	KNN	0.864	0.870	0.836	0.906
4	MLP	0.628	0.551	0.694	0.456
5	Naive Bayes	0.594	0.509	0.643	0.421
6	Logistic Regression	0.561	0.585	0.554	0.618

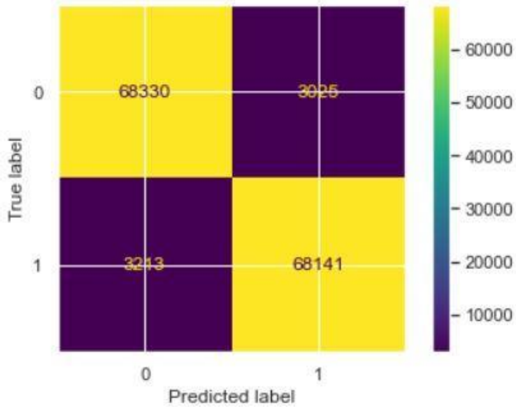
Table 1. Evaluation metrics for classifiers

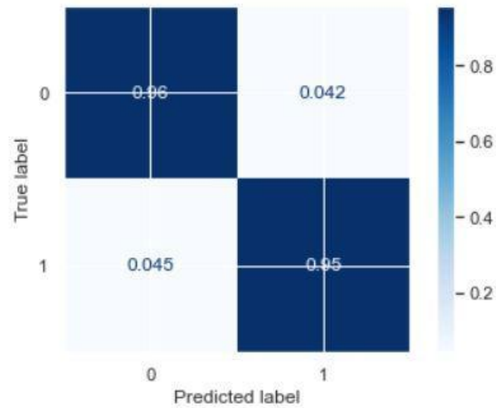
We have also plotted the confusion matrices for all the classifiers to get a better visualization of evaluation metrics. For each classifier, the confusion matrix on the left shows the number of observations belonging to true positive, false positive, true negative, and false negative classes. The confusion matrix on the right is the normalized matrix of the one on the left.

Random Forest

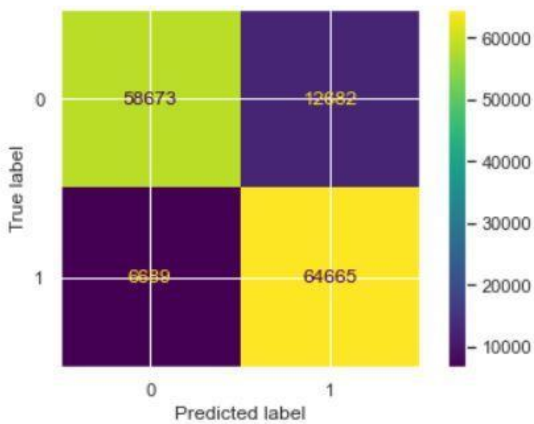


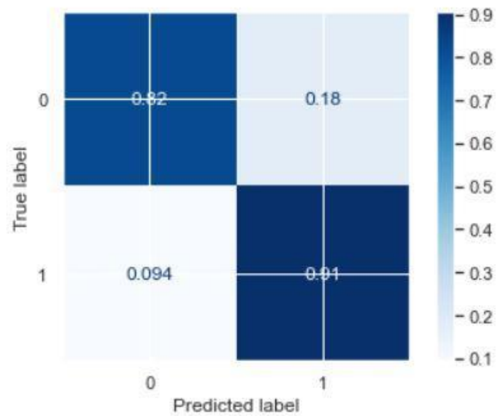
Decision Tree



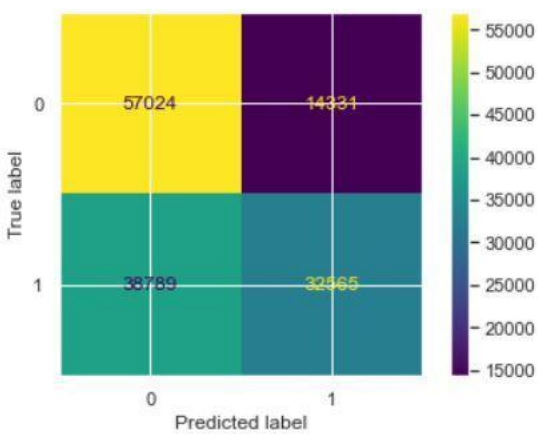


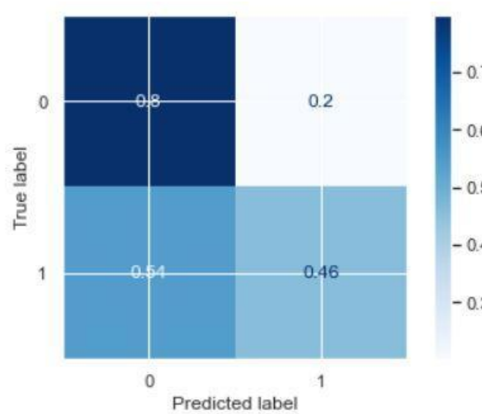
KNN



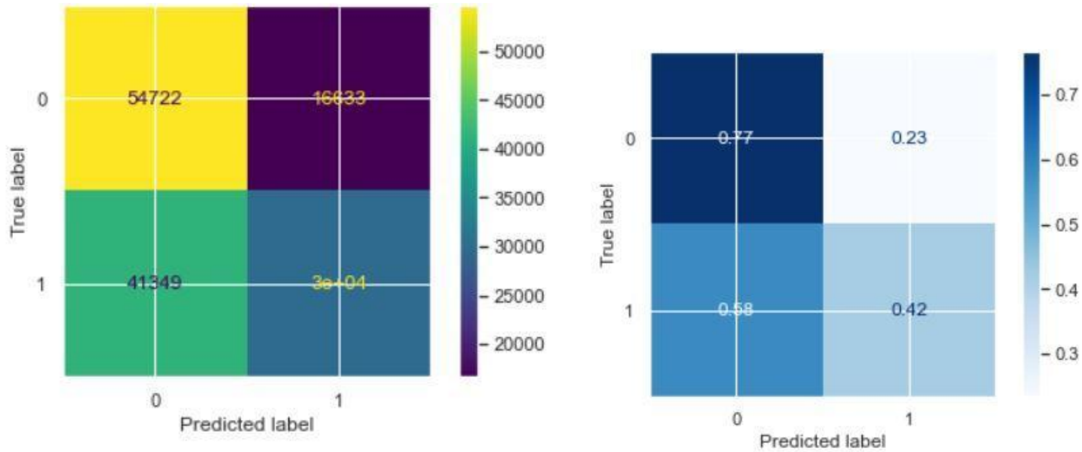


MLP

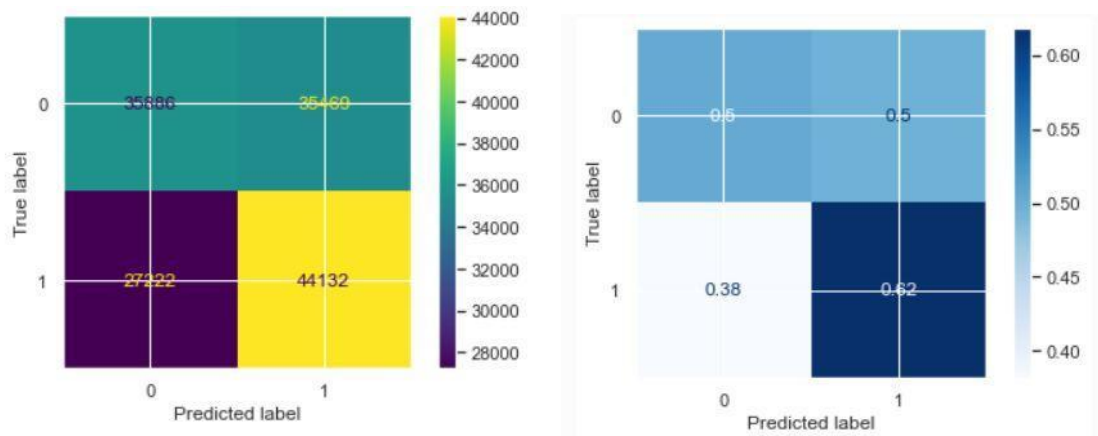




Naive Bayes



Logistic Regression



CONCLUSION

In this project we studied various approaches to solve the occupancy prediction problem such as Decision tree, KNN, Random forest, logistic regression, etc. We successfully processed spatial-temporal data for training on different models. We successfully trained Random Forest, Decision Tree, K-nearest Neighbours, Multilayer perceptron, Naive Bayes, and Logistic Regression classifiers for predicting occupancy. We can conclude that Random forest had the best performance among all the classifiers with an accuracy of 96%, followed by the Decision tree classifier with an accuracy of 95%. The K nearest neighbor classifier had the next highest accuracy of 86% implying that it was not as good as the RF and DT classifiers. On the other hand, MLP, Naive Bayes, and Logistic regression had poor performances and therefore are not suitable for our task. Therefore, we can choose any one of RF and DT as our classifiers for this problem.