University of the West of Scotland
- REQ001061 -
Data Scientist - KTP Associate

KeyFM Dataset Analysis Report

Date: 18 January 2021

Name: Alok Kumar Sahu
Email: alok.kr.sahu@outlook.com
Personal Website: www.alokkrsahu.com
Contact: +44-77419-46965

Table of Contents

Table of Figures

1. Background

This report provides the analysis for KeyFM's facility management dataset for the year 2020 from the January to December month. The dataset contains time series information about a facility's space occupancy performance and associated environmental factors like humidity, temperature, noise, CO2 levels, Lux etc. The report contains in-depth analysis of the dataset and explains the relationship among the parameters and their influence on each other if present. Below figure shows a sample data from the dataset.

PARAMETERS	VALUES
DAY	21
MONTH	2
HR	16
MINUTEZONE	21-30
MONTHNAME	February
SPACE NAME	Lomond
AVERAGE OF MAXOCCUPANTS	6
AVERAGE OF HUMID	19
AVERAGE OF NOISE	53.11
AVERAGE OF CO2	8194
AVERAGE OF PRESENCE	100
AVERAGE OF OCCUPANCY	2
AVERAGE OF OCCUPANCYPERCENTAGE	33
AVERAGE OF TEMPERATURE_AJUSTED	26.91111
AVERAGE OF VOC	1187
AVERAGE OF LUX	102
AVERAGE OF PRESSURE	1005

Figure 1: Sample data from dataset February month

A new feature is created i.e., weekday using the Day, Month and Year to perform further analysis. Further the analysis, a predictive model is developed to predict the "Average of OccupancyPercentage" parameter using the parameters mentioned in Figure 1. The modelling and analysis is performed using the Python programming language version 3.8.5. The code used to perform the analysis, create visualization and develop the predictive model is provided in the appendices.

2. Presence and Space Name Performance Analysis

This section provides analysis of spaces against the observed occupancy and provides time series analysis of the space performance.

2.1 Space Name VS Presence

The below figure shows the occupancy performance across various space areas. From table 1 and Figure 2 it is seen that the maximum occupants were observed on Common Area B, while Common Area C is most frequently used space. The kitchen space remained the least utilized space in the year

2020, although it recorded second max occupancy. It is possibly due to the impact of Covid-19, as shared spaces were less frequently used during the pandemic of 2020.

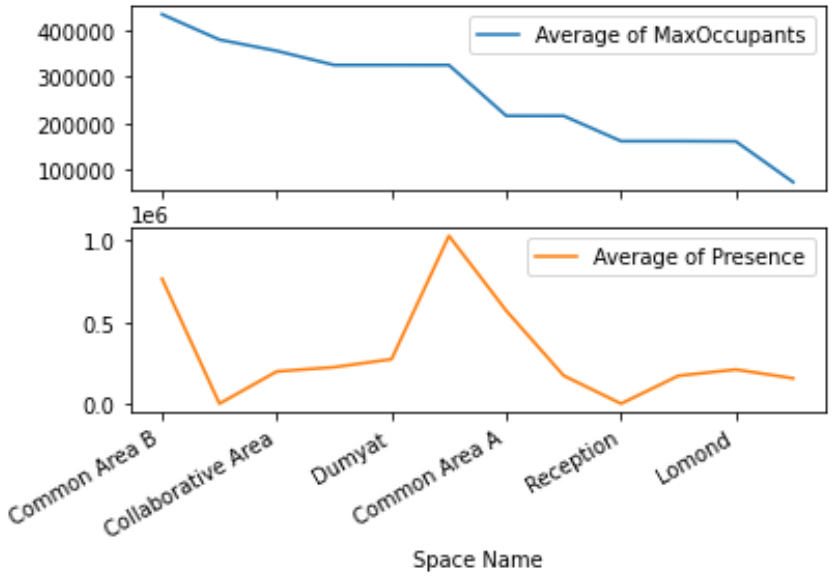


Figure 2: Space Name vs Presence and Max Occupants

AVERAGE OF PRESENCE	AVERAGE OF MAX OCCUPANTS
COMMON AREA C	COMMON AREA B
COMMON AREA B	KITCHEN
COMMON AREA A	COLLABORATIVE AREA
DUMYAT	NEVIS
NEVIS	DUMYAT
LOMOND	COMMON AREA C
COLLABORATIVE AREA	COMMON AREA A
SUILVEN	SUILVEN
MACDUI	RECEPTION
LIBRARY/RESEARCH	MACDUI
KITCHEN	LOMOND
RECEPTION	LIBRARY/RESEARCH

Table 1: Space Name by Average of Presence and Average of Max Occupants (Descending order)

2.2 Presence by Hour

The graph shown in figure 3 shows the average presence across spaces on hourly basis. A steep inclination in curve is observed during the time interval between 8:00 to 11:00. This time interval is usually the opening times of offices. From Figure 3: Presence on Hourly Basis and Figure 4: Space performance - Hourly basis, it is observed that, there is a uniform and temporary decrease in average presence across all spaces between 11:00 to 14:00. However, the graph recovers at around 15:00 to near original value. This absence noticed between 11:00 and 14:00 can be inferred as the lunch break time of the day. After 15:00, the presence gradually declines across all spaces and reaches minimum value after 22:00.

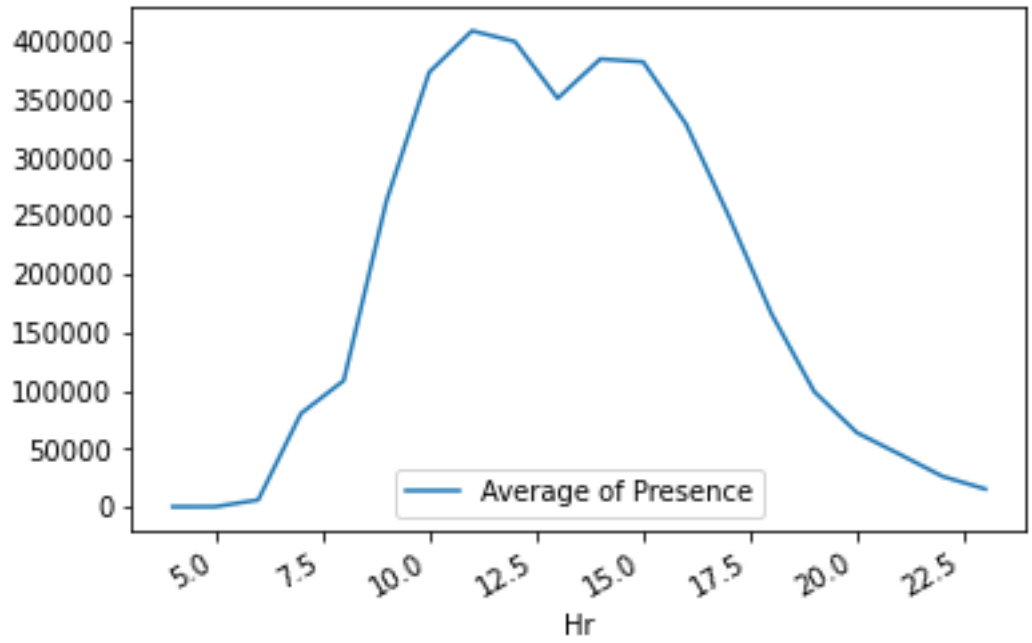


Figure 3: Presence on Hourly Basis

The absence interval observed between 11:00 and 14:00 can used to optimize the resource utilized across spaces by fine tuning the equipment's.

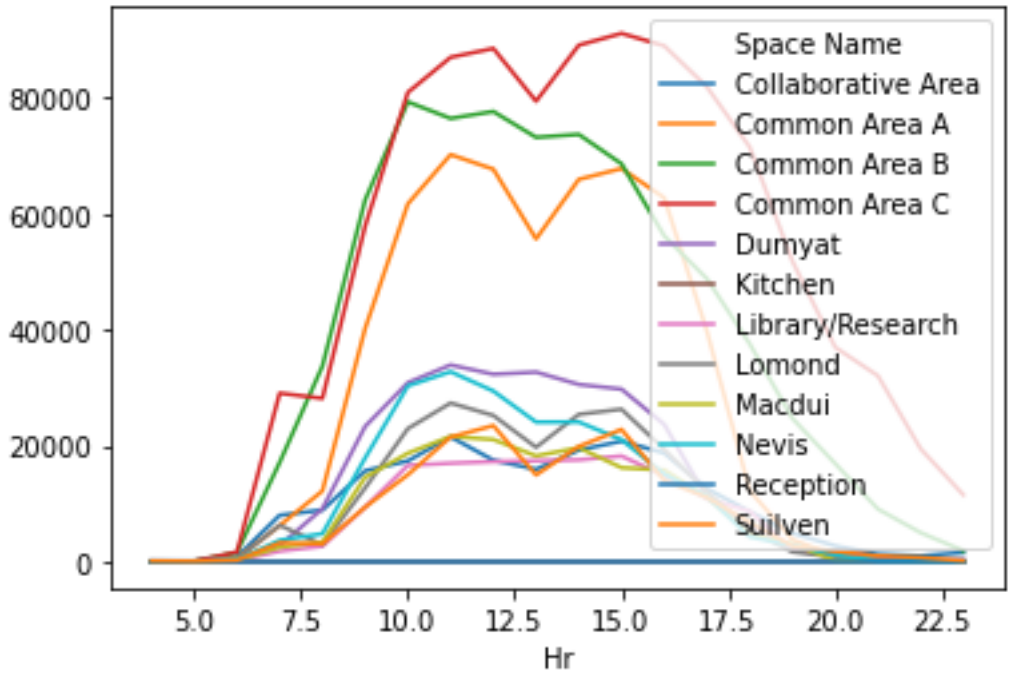


Figure 4: Space performance - Hourly basis

2.3 Presence on Monthly Basis

The below graph shows average presence recorded across all spaces on monthly basis. A sudden decline in the March (3) and April (4) month can be observed from the Figure 5: Month VS Average of Presence. This is the nationwide lockdown period observed globally in the year 2020 due to the novel Covid-19 outbreak. Hence the decline in presence during this period is interpretable as due to the pandemic and lockdown restrictions imposed across the country.

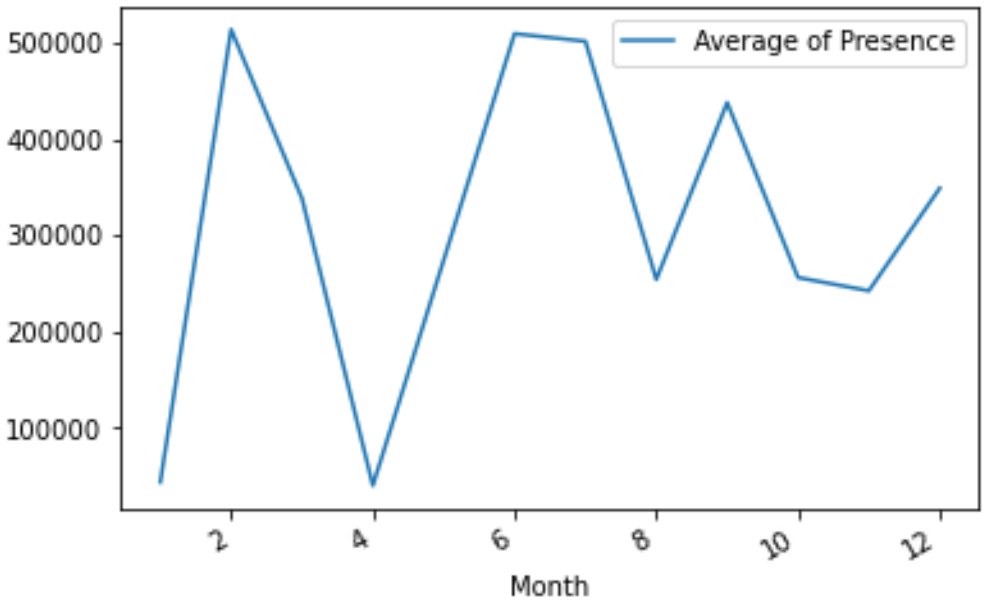


Figure 5: Month VS Average of Presence

2.4 Presence on weekday basis

The Figure 6: Weekday vs Average of Presence, shows the presence across all spaces against the weekdays. The maximum presence is observed on Wednesdays and the weekends marks near absolute absence across the facility.

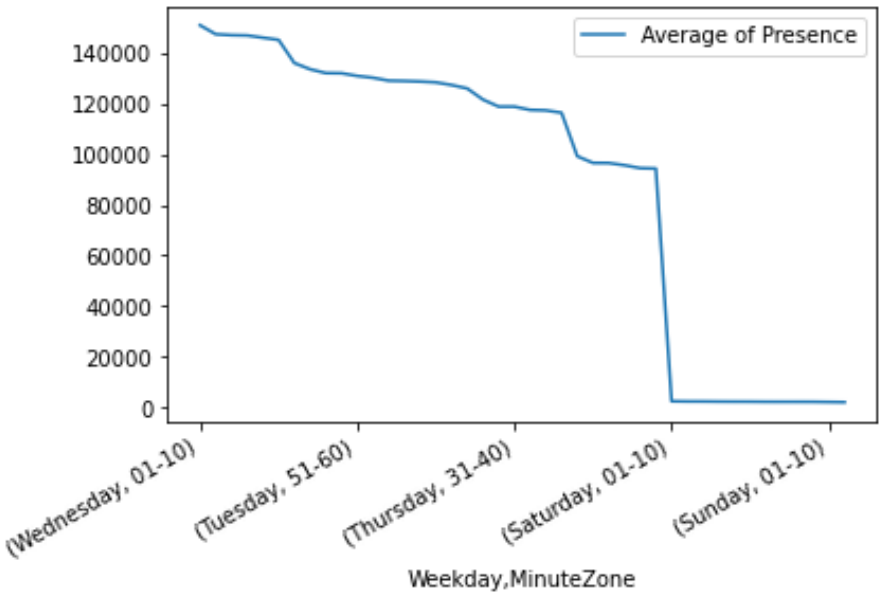


Figure 6: Weekday vs Average of Presence

2.5 Presence on day of the month

From the Figure 7: Day of the Month VS Average of Presence, it is observed that the second week of the month observes the maximum presence and the average presence decline on the fourth week of the month. A periodic increase and decrease in the average number of presences is observed on the graph roughly after every 2 to 3 days. A steep reduction in presence is observed on 5th, 15th, and around 30th of the month.

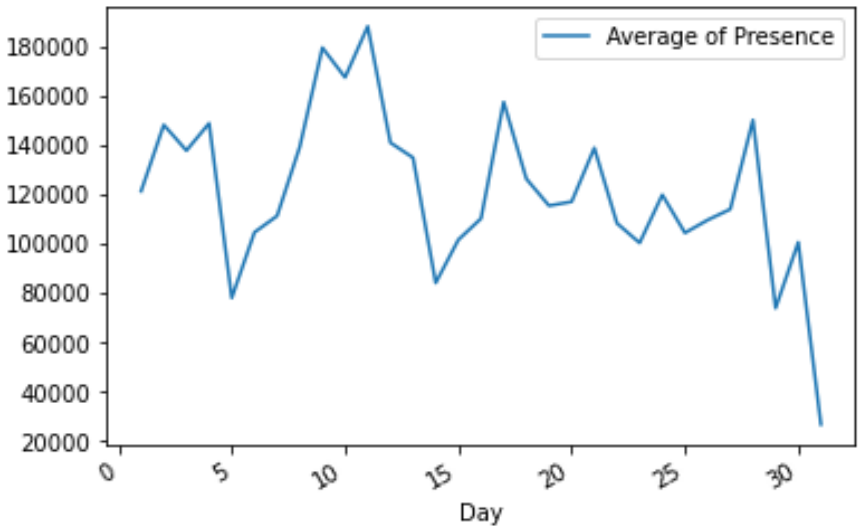


Figure 7: Day of the Month VS Average of Presence

3. Space Name analysis against other parameters

3.1 Space Name VS Lux (Light Levels)

The below Figure 8: Space Name VS Lux (Light Levels), shows the light levels across all spaces. The light levels are observed maximum on Common Area A and Dumyat while the light levels are observed lowest in Common Area B, Common Area C and Collaborative Area despite having high number of presences as shown in Table 1: Space Name by Average of Presence and Average of Max Occupants (Descending order). These low light levels and high presence can be due to the incoming ambient light in the Common Area B and C.

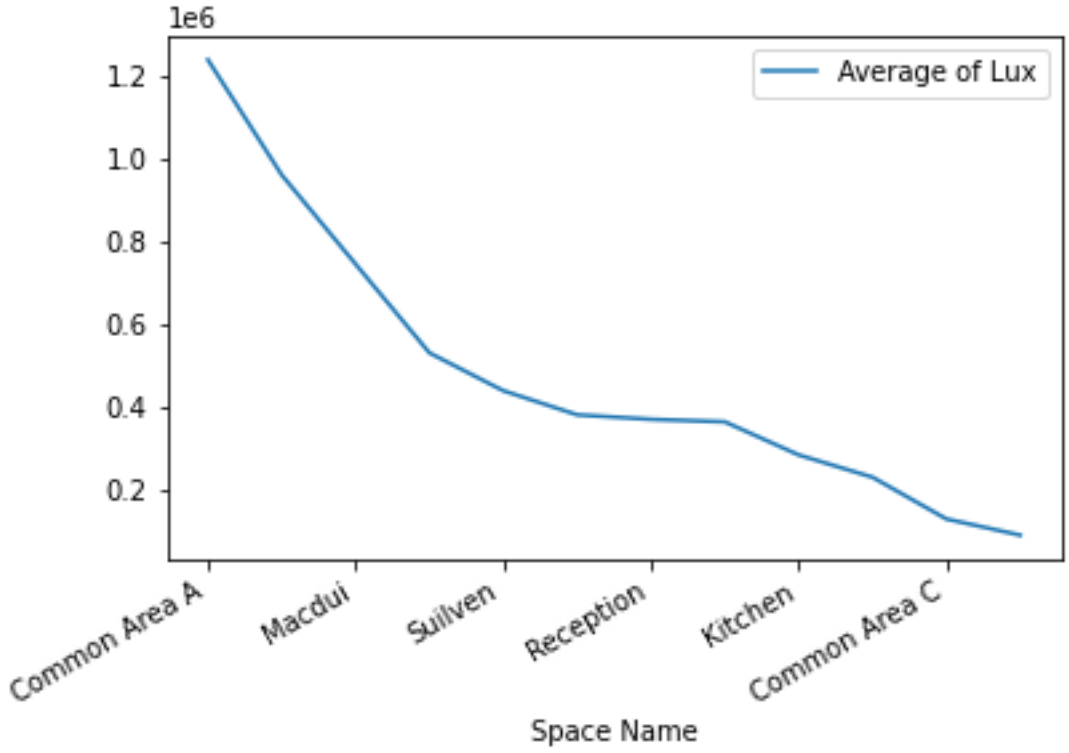


Figure 8: Space Name VS Lux (Light Levels)

3.2 Space Name VS VOC, CO2 and Noise levels

The Figure 9: Space Name VS CO2, Noise and VOC Levels shows good relationship among CO2, VOC and Noise parameters. CO2 and VOC parameters has high correlation and covariance. Hence it can be said that anyone (CO2 or VOC) is a representation of other. While CO2/VOC and Noise has high covariance and no correlation. CO2/VOC and Noise levels have got high correlation with the Average of Presence parameter as shown in Figure 12: Covariance Matrix of Parameters and Figure 13: Correlation Matrix of Parameters. As shown in Table 1: Space Name by Average of Presence and Average of Max Occupants (Descending order), it is observed that Average Presence is one of the least in Kitchen. It is the reason why the Noise levels and CO2 levels are observed least among all spaces in the Kitchen as shown in the Figure 9: Space Name VS CO2, Noise and VOC Levels.

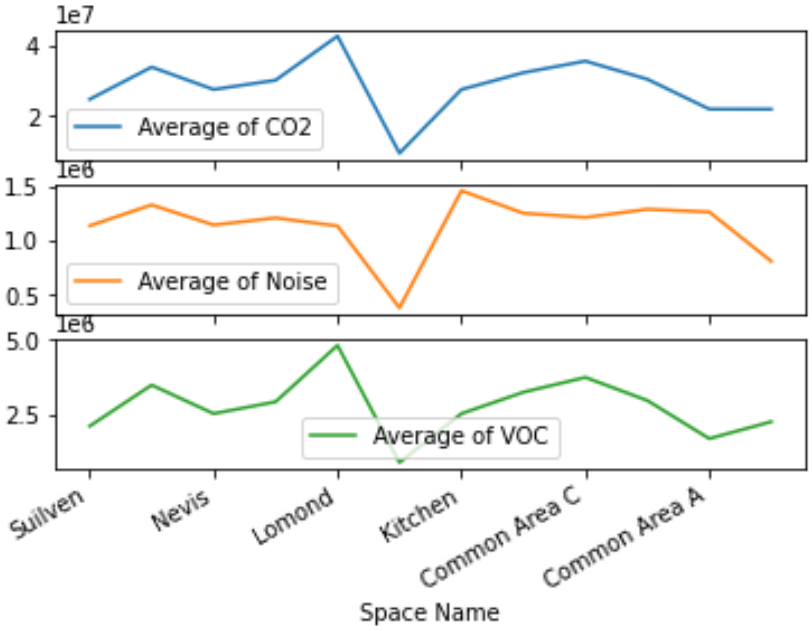


Figure 9: Space Name VS CO2, Noise and VOC Levels

3.3 Space Name VS Humidity Pressure and Temperature, compared on hourly basis

The below Figure 10: Space Name VS Humid, Pressure and Temperature shows the fluctuations in the Humid, Pressure and Temperature parameter observed across spaces on hourly basis through out the year. These parameters on observed to be lowest in the Library/Research followed by Collaborative area and highest on Lomond and Common Area C & A. From the figure below it can be said that the three parameters have high covariance.

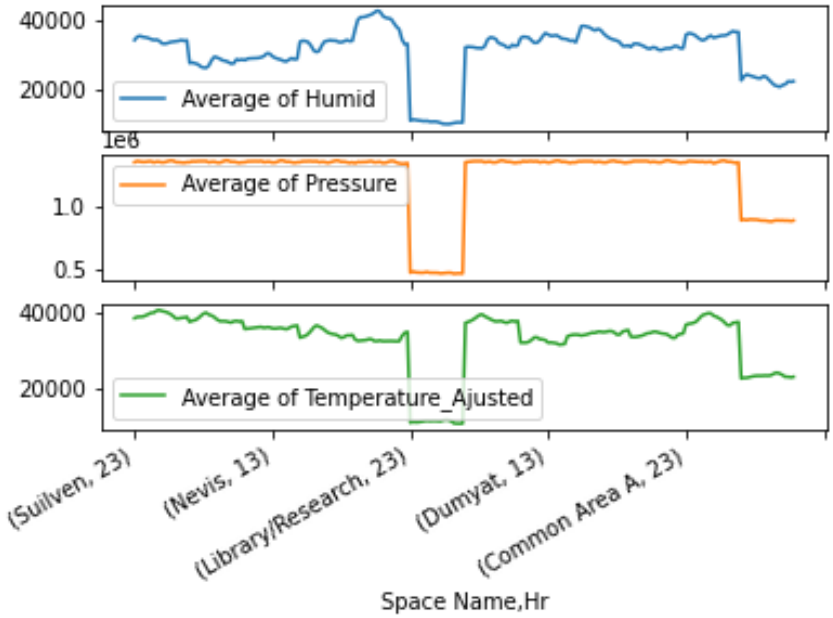


Figure 10: Space Name VS Humid, Pressure and Temperature

3.4 CO2 vs Average of Lux Level analysis

The Figure 11: CO2 level vs Lux Level - Hourly shows the variance in the level of CO2 and Lux observed on hourly basis throughout the year. It clearly observed that both the parameters CO2 and Average of Lux are inversely proportional i.e., with the decrease in light level CO2 increases. It is naturally normal to observe high levels of CO2 at night with decreased light levels and as can be seen in the data.

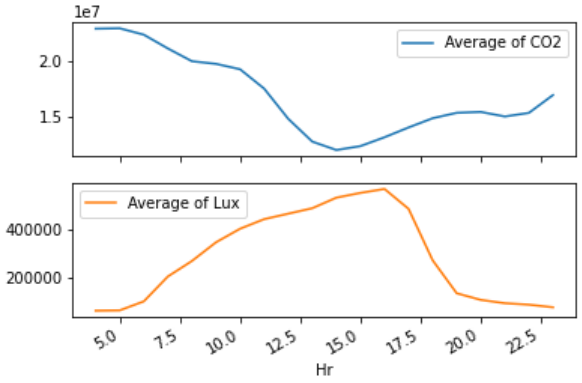


Figure 11: CO2 level vs Lux Level - Hourly

4. Covariance and Correlation matrix analysis among parameters

This section presents the covariance and correlation relationship and analysis among different parameters. The following subsection shows the covariance matrix, correlation matrix and pairwise plot among the parameters.

4.1 Pair wise covariance relationship among parameters

This section presents the pairwise covariance relationship of parameters as shown in the Figure 12: Covariance Matrix of Parameters

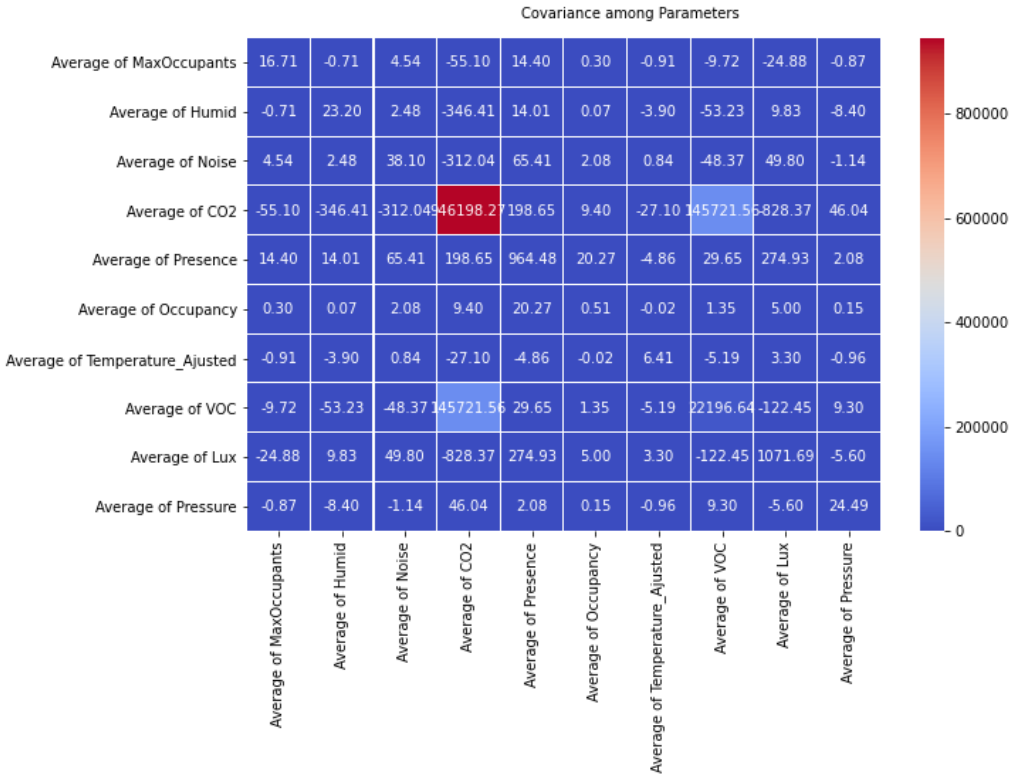


Figure 12: Covariance Matrix of Parameters

4.2 Pair wise correlation among parameters

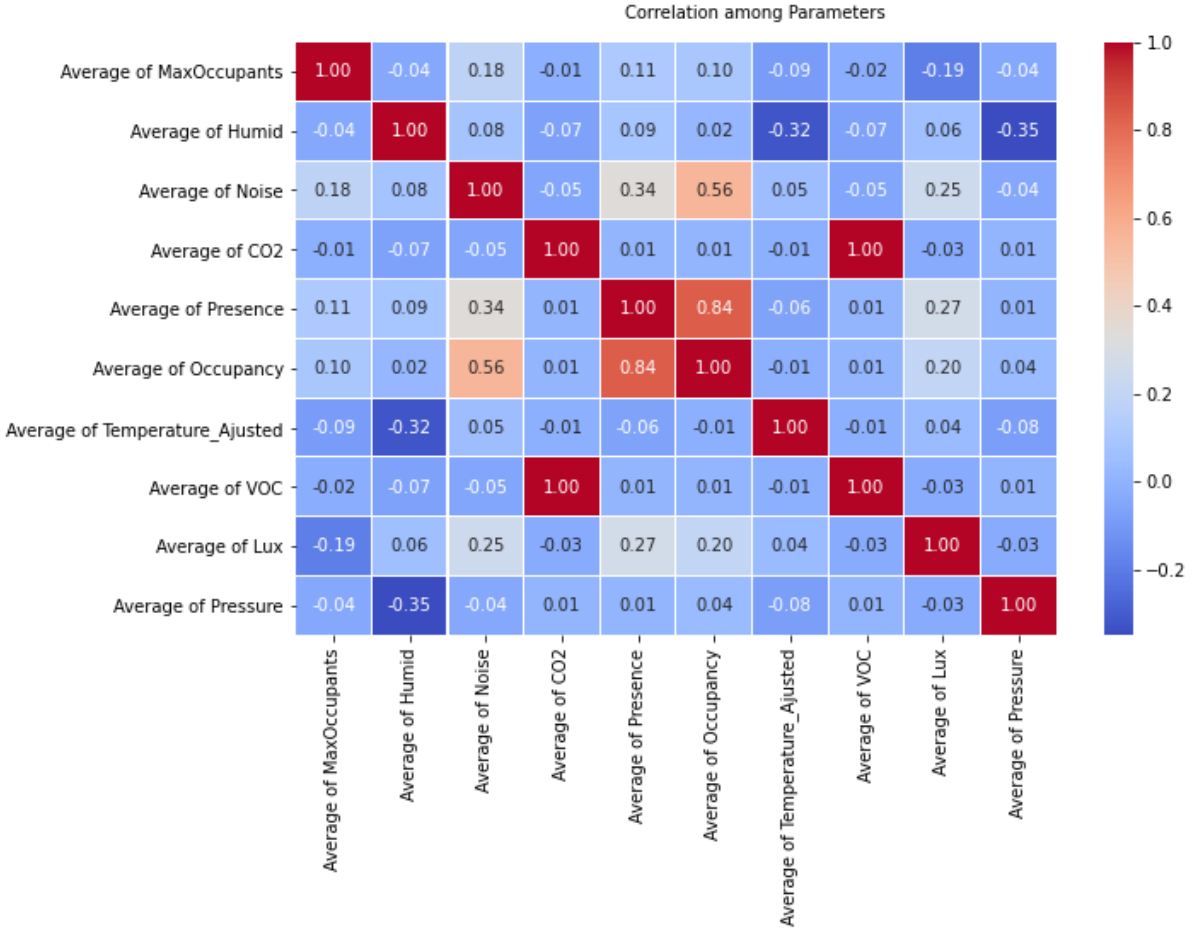
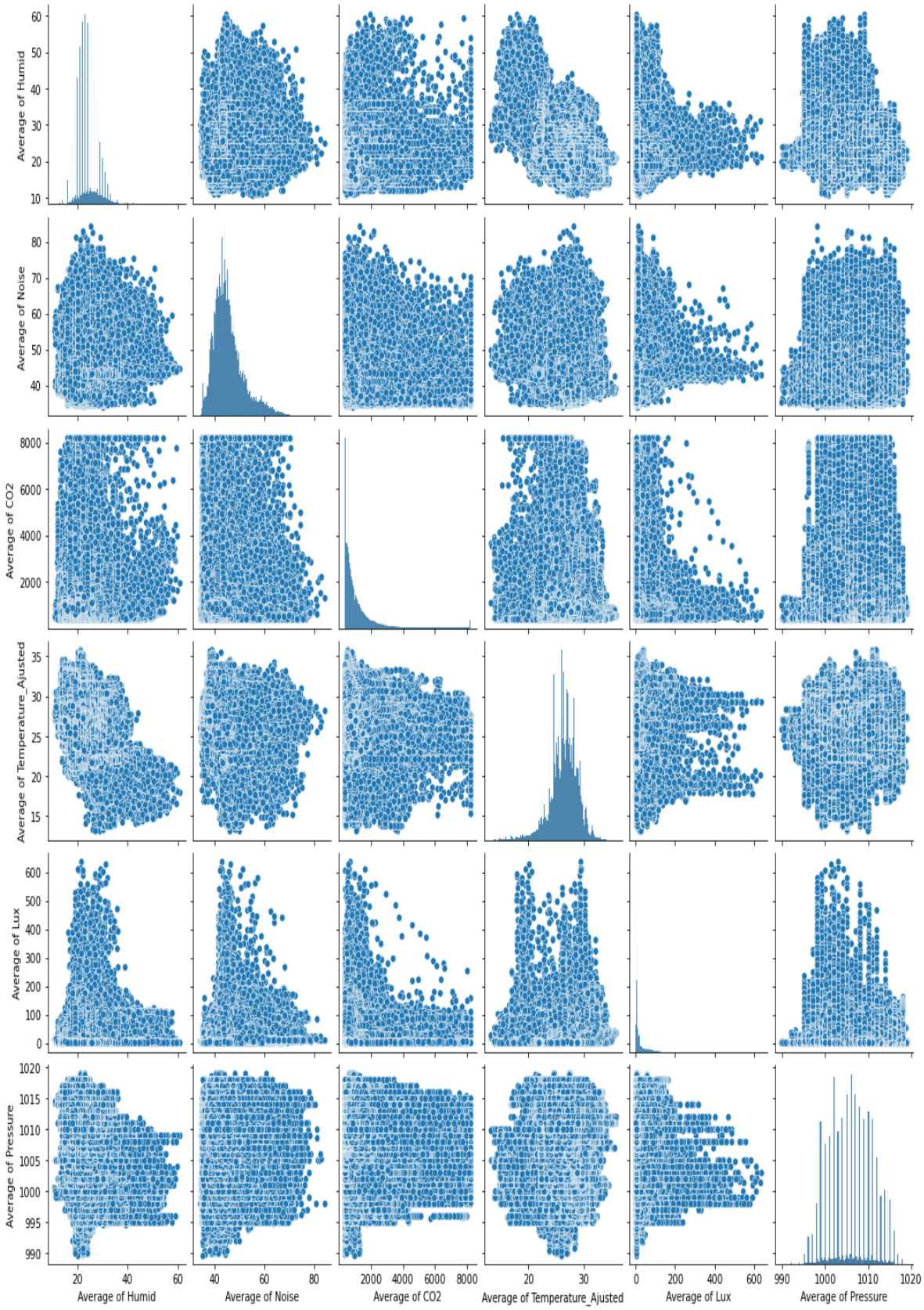


Figure 13: Correlation Matrix of Parameters

4.3 Pair plot of all parameters



The following conclusion can be deduced from the above pair plot, covariance and correlation matrices.

Average of MaxOccupants: It has got positive correlation with Noise, Presence and has got high negative correlation with Average of Lux and CO2 levels.

Average of Humid: Humidity is found to have high negative covariance with CO2 and positive covariance with Lux Levels. It is negatively correlated with pressure and temperature.

Average of Noise: It is negatively correlated with CO2 and has got high positive correlation with Average of Presence, Lux levels.

Average of CO2: As seen before in Figure 11: CO2 level vs Lux Level - Hourly, Lux and CO2 are negatively correlated along with humidity and Noise. VOC has got highest covariance and correlation of 1 with CO2 i.e., both can be interpreted as representation of each other.

Average of Temperature_Adjusted: Temperature parameter is found to have negative covariance with humidity, CO2 and Presence.

Average of Pressure: Pressure increases with CO2 level and has got high correlation with CO2 and VOC. Humidity and pressure are inversely proportional and has got high negative covariance value.

5. Prediction Modelling

This section present details of a prediction model developed to predict the Average Occupant Percentage parameter. The model uses Random Forest Regressor to predict the Average Occupant Percentage parameter. Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. The basic steps involved for prediction in Random Forest are as below:

1. Pick at random k data points from the training set.
2. Build a decision tree associated to these k data points.
3. Choose the number N of trees you want to build and repeat steps 1 and 2.
4. For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

The dataset is split into train and test dataset with ratio of approximately 70 and 30 respectively. Feature selection is carried out using the feature score form SelectKBest function from Scikit-Learn library in Python. The feature score of the parameters are shown in the Figure 14: Feature Score of Parameters.

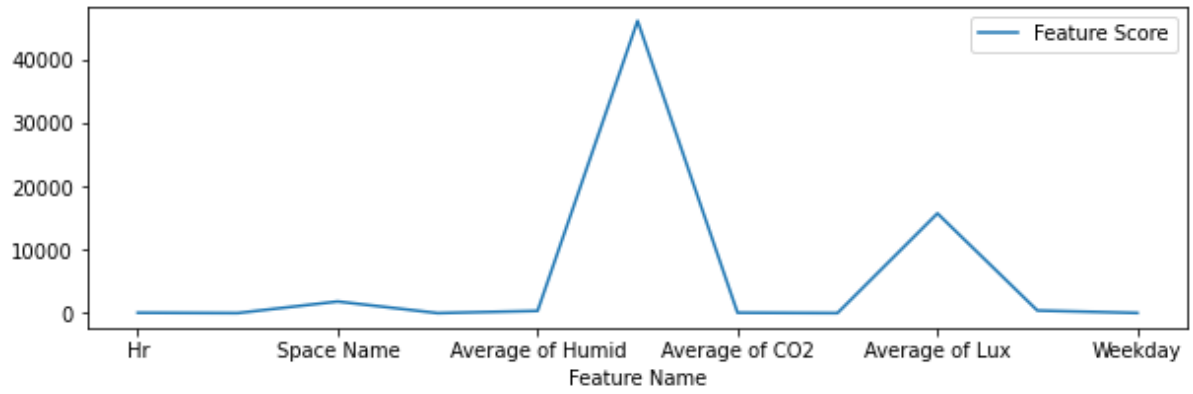


Figure 14: Feature Score of Parameters

The model is evaluated using the Mean Absolute Error metrics. The mean absolute error is a risk metric corresponding to the expected value of the absolute error loss. With max depth of 10 the models Mean Absolute Error was 1.4399.

6. Appendices

Data_extraction.py

```
from os import listdir

from os.path import isfile, join

import pandas as pd


myfiles = [f for f in listdir('./Data') if isfile(join('./Data', f))]


file_names  = list()

for each in myfiles:

    name =  each.split()

    file_names.append(name[-1])



data = dict()

all_data = pd.DataFrame(columns = list(pd.read_csv('./Data/Global Workplace ' + file_names[0])))

for fn in file_names:

    data.update({fn:pd.read_csv('./Data/Global Workplace ' + fn)})

    all_data = all_data.append(pd.DataFrame(data[fn]))
```

```
col_names = data[fn].columns

i = 1
space = dict()
space_name = data[fn]['Space Name'].unique()

for each in space_name:
    space.update({each:i})
    i += 1

i = 1
MinZone = dict()
MinuteZone = data[fn]['MinuteZone'].unique()
for each in MinuteZone:
    MinZone.update({each:i})
    i += 1

#ENCODING SPACE NAME, MINUTE ZONE
for each in file_names:
     data[each]['MinuteZone'] = data[name[-1]]['MinuteZone'].map(MinZone)
     data[each]['Space Name'] = data[name[-1]]['Space Name'].map(space)
#df['c'] = df.apply(lambda x: max([len(x) for x in [df['a'], df['b']]]))
all_data['Date'] =
pd.to_datetime((2020*10000+all_data.Month*100+all_data.Day).apply(str),format='%Y%m%d')
all_data['Weekday'] = all_data['Date'].dt.day_name()

i = 1
Weekday_num = dict()
Weekday = all_data['Weekday'].unique()
```

```
for each in Weekday:
    Weekday_num.update({each:i})
    i += 1
```

space_performance.py

```
import data_extraction as de
import matplotlib.pyplot as plt
import seaborn as sns

all_data = de.all_data
data = de.data
files = de.file_names
col = de.col_names

# ANALYSIS AND VISUALIZATION CREATED BY PANDAS PIVOT TABLE
all_data = all_data.sort_values(by = ['Month','Day','Hr'], )
all_data.pivot_table(index=['Weekday'], values=['Average of Presence'],
aggfunc='sum').sort_values(by='Average of Presence',ascending = False).plot(subplots=True)
all_data.pivot_table(index=['Month'], values=['Average of Presence'],
aggfunc='sum').plot(subplots=True)
all_data.pivot_table(index=['Space Name'], values=['Average of MaxOccupants','Average of
Presence'], aggfunc='sum').sort_values(by=['Average of Presence','Average of
MaxOccupants'],ascending = False).plot(subplots=True)
all_data.pivot_table(index=['Space Name'], values=['Average of Lux'],
aggfunc='sum').sort_values(by=['Space Name'],ascending = False).plot(subplots=True)
all_data.pivot_table(index=['Space Name'], values=['Average of Noise','Average of CO2','Average of
VOC'], aggfunc='sum').sort_values(by='Space Name',ascending = False).plot(subplots=True)
all_data.pivot_table(index=['Space Name','Hr'], values=['Average of Temperature_Ajusted','Average
of Humid','Average of Pressure'], aggfunc='sum').sort_values(by=['Space Name','Hr','Average of
Pressure','Average of Humid'],ascending = False).plot(subplots=True,kind='line')
all_data.pivot_table(index=['Hr'], values=['Average of Lux','Average of CO2'],
aggfunc='sum').plot(subplots=True)
all_data.pivot_table(index=['Hr'], values=['Average of Presence'], aggfunc='sum').plot(subplots=True)
```

```
all_data.pivot_table(index=['Hr'],columns='Space Name',values='Average of Presence',aggfunc='sum').plot()

all_data.pivot_table(index=['Hr'], values=['Average of Temperature_Ajusted'], aggfunc='sum').plot(subplots=True)

all_data.pivot_table(index=['Hr'], values=['Average of Lux'], aggfunc='sum').plot(subplots=True)

all_data.pivot_table(index=['Weekday','MinuteZone'], values=['Average of Presence'], aggfunc='sum').sort_values(by='Average of Presence',ascending = False).plot(subplots=True)

all_data[((all_data['Hr']>6) & (all_data['Hr']<20)) ].pivot_table(index=['Hr','MinuteZone'], values=['Average of Presence'], aggfunc='sum').sort_values(by=['Hr','MinuteZone','Average of Presence'],ascending = True).plot(subplots=True)

all_data.pivot_table(index=['Day',], values=['Average of Presence'], aggfunc='sum').sort_values(by=['Day','Average of Presence'],ascending = True).plot(subplots=True)


# CREATING COVARIANCE AND CORRELATION MATRIX

features = ['MinuteZone','Space Name','Average of MaxOccupants','Average of Humid','Average of Noise','Average of CO2','Average of Presence','Average of Occupancy','Average of Temperature_Ajusted','Average of VOC','Average of Lux','Average of Pressure']

feature_cov = all_data[features].cov()

feature_corr = all_data[features].corr()



# PLOTTING COVARIANCE MATRIX

f, ax = plt.subplots(figsize=(10, 6))

hm = sns.heatmap(round(feature_cov,2), annot=True, ax=ax, cmap="coolwarm",fmt='.2f',linewidths=.05)

f.subplots_adjust(top=0.93)

t= f.suptitle('Covariance among Parameters', fontsize=10)



# PLOTTING CORRELATION MATRIX

f, ax = plt.subplots(figsize=(10, 6))

hm = sns.heatmap(round(feature_corr,2), annot=True, ax=ax, cmap="coolwarm",fmt='.2f',linewidths=.05)

f.subplots_adjust(top=0.93)
```

```
t= f.suptitle('Correlation among Parameters', fontsize=10)

new_features = ['Average of Humid','Average of Noise','Average of CO2','Average of
Temperature_Ajusted','Average of Lux','Average of Pressure']


# CREATING PAIR PLOTS USING SEABORN

sns.pairplot(all_data[new_features])
```

Prediction_model_Random_Forest_Regressor.py

```
# -*- coding: utf-8 -*-

"""

Created on Mon Jan 18 17:48:23 2021


@author: alokk

"""


import data_extraction as de

from sklearn.ensemble import RandomForestRegressor

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_absolute_error

from sklearn.feature_selection import SelectKBest

from sklearn.feature_selection import f_regression


import pandas as pd


all_data = de.all_data

#Selected Features

features = ['Hr','MinuteZone','Space Name','Average of MaxOccupants','Average of Humid','Average
of Noise','Average of CO2','Average of Temperature_Ajusted','Average of Lux','Average of
Pressure','Weekday']


#Target Feature

label = ['Average of OccupancyPercentage']
```

```
regr = RandomForestRegressor(max_depth=10, random_state=0)

data = all_data[features]

data['Weekday'] = data['Weekday'].map(de.Weekday_num)
data['MinuteZone'] = data['MinuteZone'].map(de.MinZone)
data['Space Name'] = data['Space Name'].map(de.space)

data['Space Name'].fillna(value=data['Space Name'].median(), inplace=True)
data['Average of Lux'].fillna(value=data['Average of Lux'].mean(), inplace=True)
train = data
test = all_data[label]

test.fillna(value=test.mean(), inplace=True)

X_train, X_test, y_train, y_test = train_test_split(train, test, test_size=0.33, random_state=100)

def select_features(X_train, y_train, X_test):
        # configure to select all features
        fs = SelectKBest(score_func=f_regression, k='all')
        # learn relationship from training data
        fs.fit(X_train, y_train)
        # transform train input data
        X_train_fs = fs.transform(X_train)
        # transform test input data
        X_test_fs = fs.transform(X_test)
        return X_train_fs, X_test_fs, fs

X_train_fs, X_test_fs, fs = select_features(X_train, y_train, X_test)
# what are scores for the features
```

```
for i in range(len(fs.scores_)):
        print('Feature %s: %f' % (features[i], fs.scores_[i]))

# plot the scores
z = pd.DataFrame([features,list(fs.scores_)]).T
z.columns =['Feature Name', 'Feature Score']
z.plot('Feature Name', 'Feature Score',figsize=(10,3))



regr.fit(X_train, y_train)


y_pred = regr.predict(X_test)


print(mean_absolute_error(y_test, y_pred, multioutput='raw_values'))
```