

# PREDICTION OF ROOM OCCUPANCY AS A BINARY CLASSIFICATION



# Dataset

Data Set Characteristics:	Multivariate, Time-Series	Number of Instances:	20560	Area:	Computer
Attribute Characteristics:	Real	Number of Attributes:	7	Date Donated	2016-02-29
Associated Tasks:	Classification	Missing Values?	N/A	Number of Web Hits:	182402

## Attributes

Date time year-month-day hour:minute:second

Temperature, in Celsius

Relative Humidity, %

Light, in Lux

CO2, in ppm

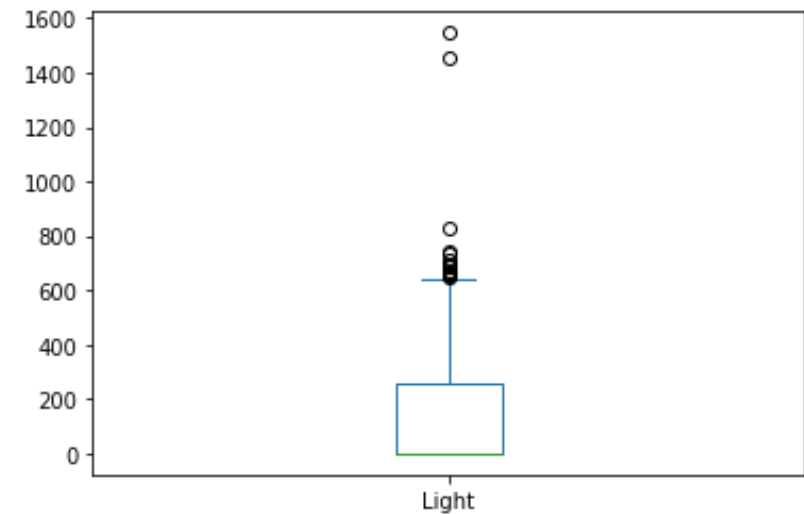
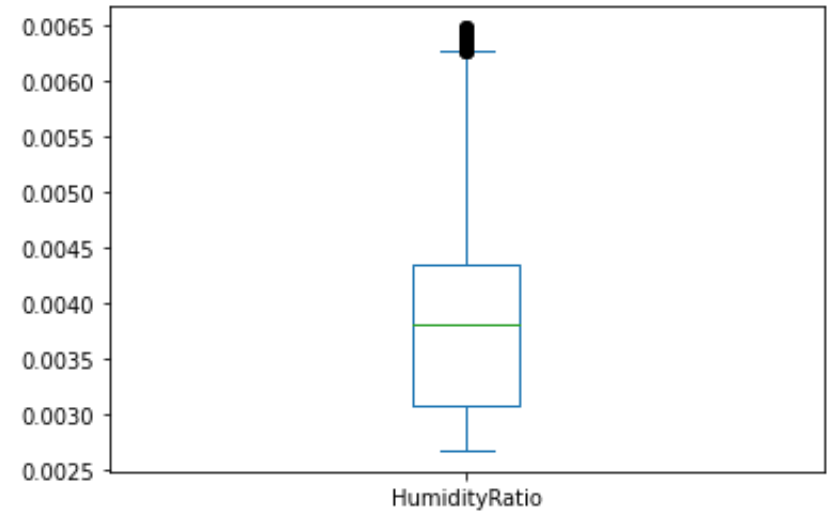
Humidity Ratio, Derived quantity from temperature and relative humidity, in kgwater-vapor/kg-air

Occupancy, 0 or 1, 0 for not occupied, 1 for occupied status

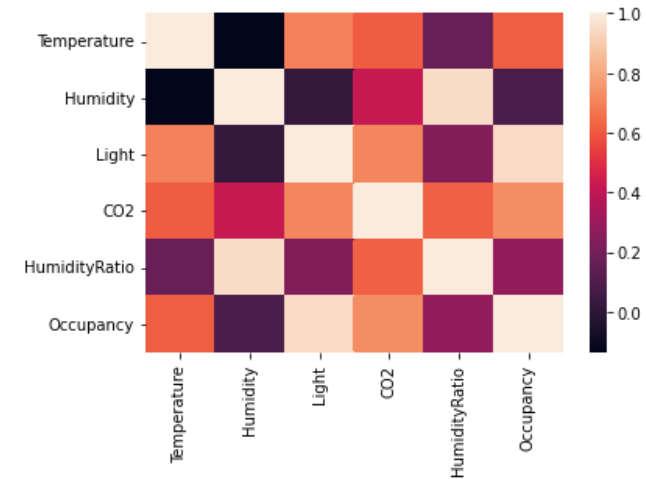
# Data Cleaning

---

- No null values found
- Outliers found and filtered out
- Imbalanced dataset was balanced
  - Majority class down sampling



# Feature Selection



- Correlation matrix
  - Higher correlation – Temperature, Light, CO2
  - All features used initially
  - Backward elimination for model tuning

	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
count	4161.000000	4161.000000	4161.000000	4161.000000	4161.000000	4161.000000
mean	20.828097	25.664226	193.172711	673.210180	0.003903	0.383802
std	1.035118	5.394723	221.999744	317.311131	0.000837	0.486369
min	19.000000	16.790000	0.000000	412.750000	0.002682	0.000000
25%	20.000000	20.545000	0.000000	441.500000	0.003191	0.000000
50%	20.790000	26.200000	7.000000	471.000000	0.003811	0.000000
75%	21.700000	30.200000	444.000000	854.000000	0.004445	1.000000
max	23.180000	38.145000	638.000000	1879.250000	0.006257	1.000000

# Model building

---

- Random forest classifier
- Decision tree classifier
- K-Nearest Neighbor classifier

## Random Forrest Classifier

```
[ ] from sklearn.ensemble import RandomForestClassifier

X_train = data[X_variables]
y_train = data[y_variable].values

RFClassifier1 = RandomForestClassifier().fit(X_train, y_train)
```

## Decision tree classifier

```
[ ] from sklearn.tree import DecisionTreeClassifier

DTClassifier1 = DecisionTreeClassifier().fit(X_train, y_train)
```

## K-Nearest Neighbour Classifier

```
[ ] from sklearn.neighbors import KNeighborsClassifier

KNNClassifier1 = KNeighborsClassifier(n_neighbors=2).fit(X_train,
```

# Model Evaluation

Evaluated against test dataset

Model		Area under ROC	Precision		Recall		f1-score	
			Class '0'	Class '1'	Class '0'	Class '1'	Class '0'	Class '1'
Random Classifier features	Forrest - 05	0.94	0.95	0.94	0.97	0.91	0.96	0.93
Decision Tree Classifier - 05 features		0.85	0.86	0.94	0.97	0.74	0.92	0.82
K-Nearest Neighbour Classifier - 05 features		0.93	0.94	0.95	0.97	0.9	0.96	0.9

# Model tuning

- Backward elimination
  - Lowest correlation – Humidity
- New metrics are shown

Model	Area under ROC	Precision		Recall		f1-score	
		Class '0'	Class '1'	Class '0'	Class '1'	Class '0'	Class '1'
Random Forrest Classifier - 04 features	0.94	0.95	0.94	0.97	0.92	0.96	0.93
Decision Tree Classifier -04 features	0.89	0.9	0.96	0.98	0.82	0.94	0.88
K-Nearest Neighbor Classifier - 04 features	0.93	0.94	0.94	0.93	0.94	0.94	0.94