**MACHINE LEARNING FOUNDATION**

**CAPSTONE PROJECT REPORT**

**DETECTION OF ROOM OCCUPANCY AS A BINARY CLASSIFICATION**

**Name: Sandun Jayawardhana**

**Reg No: 241**

**Contents**

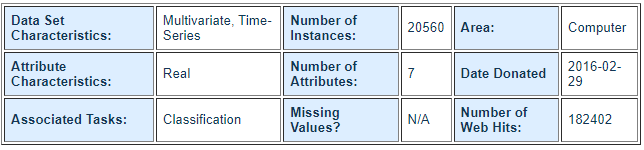
# Dataset

Name: Occupancy Detection Dataset

Source: Luis Candanedo, luismiguel.candanedoibarra '@' umons.ac.be, UMONS.

Link: <https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection>+

The dataset contains experimental data obtained in one minute frequency to determine occupancy of a room. Data columns include temperature, humidity, light, CO2 and humidity ratio.



The dataset is divided into 03 sets.

1. Training set
2. Testing set 1
3. Testing set 2

The above divided sets can be used for the machine learning model training and testing. Hence, train test split was not done in the project.

## 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

The Occupancy is the target column which need to be predicted using the machine learning algorithm.

# Methodology

The Google Colab environment was used for the development. Below libraries were used.

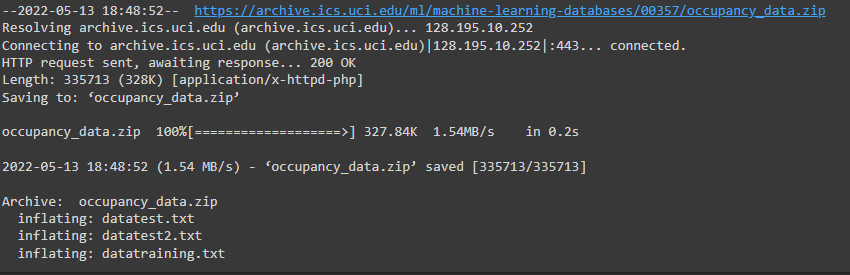
1. Pandas
2. Scipy
3. Sklearn
4. Matplotlib
5. Seaborn
6. Numpy

The below steps were followed for building the machine learning model.

1. Loading dataset
2. Data observation
3. Data cleaning
4. Feature selection
5. Model building
6. Model evaluation
7. Model tuning

## Loading dataset

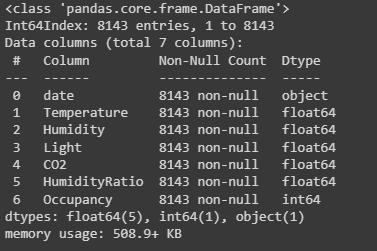
Dataset was loaded directly from web site to the Colab instance using Linux ‘wget’ command. Then it was unzipped and loaded as a Pandas dataframe.



## Data observation

The dataset was observed as a sample. Also, it was confirmed that the whole training dataset is loaded successfully. The dimensionality of the dataset was (8143,7).

The loaded dataset columns have below datatypes.

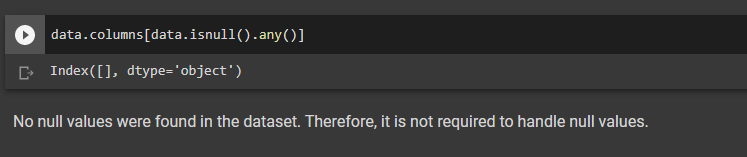


## Data cleaning

Data cleaning is a major and an important step of the machine learning model building procedure. Below 02 tasks were for data cleansing.

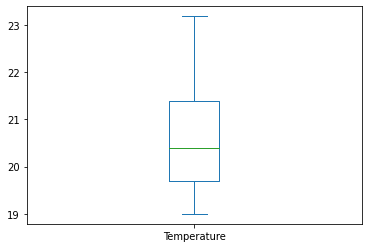
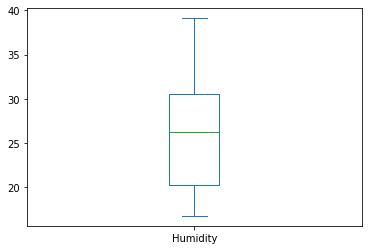
1. Treating null / Missing values

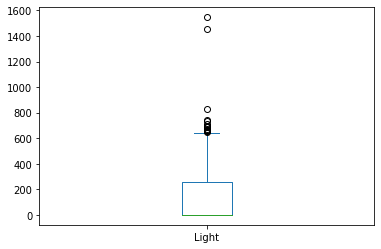
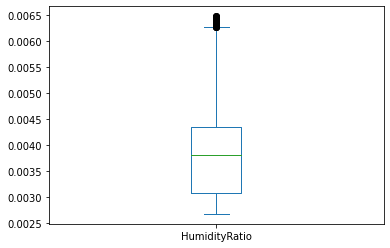
The selected dataset does not have null or missing values. Therefore, it was not required to handle or drop any data rows in this stage.



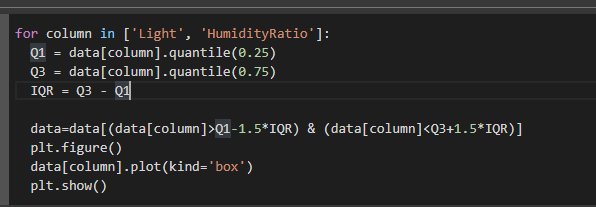
1. Treating outliers

Box plots were used to identify outliers of all the feature columns. Few bar plots are shown below.

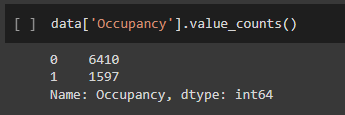
 

It can be observed that ‘Light’ and ‘Humidity Ratio’ has outliers. Those outliers were filtered using quantile-based filtering.

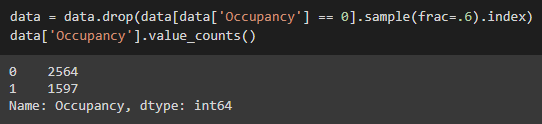


## Balancing the dataset

The dataset was observed for imbalance as the next step. It is observed that the value counts of target column are imbalanced as follows.



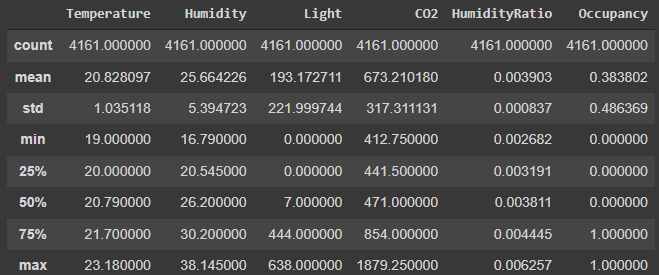
The dataset was balanced using down sampling of value 0 counting which is the majority class.



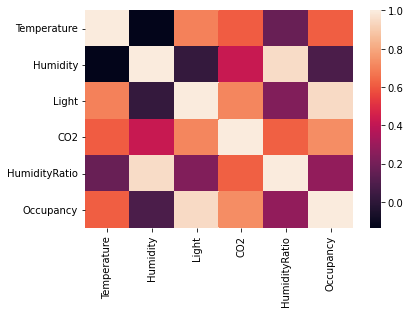
## Feature selection

There are 07 columns in the dataset. The target column is set as ‘Occupancy’. Out of the remaining 06 columns, index column is the datetime which was not considered as a feature. Therefore, 05 columns are available for feature selection.

As the first step, dataset was described using data.describe() function.



It is observed that all the columns have acceptable variance for considering them to the next step. As the next step, correlation matrix was observed.



According to the correlation matrix, it can be observed that ‘Temperature’, ‘Light’ and ‘CO2’ columns have higher correlation with ‘Occupancy’ column. ‘Humidity’ and ‘Humidity Ratio’ have lower correlation values.

However, it was decided to keep all the columns as features because of the low number of column count. The backward elimination will be used in the model tuning step.

## Model building

Since this is classification problem, 03 classification algorithms are used.

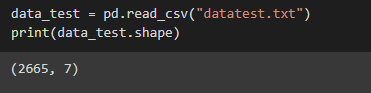
1. Random Forrest classifier
2. Decision Tree Classifier
3. K-Nearest Neighbour classifier

All the models were initiated and fit with training data as follows.

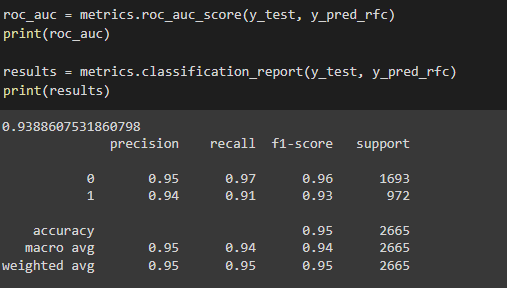


## Model evaluation

The dataset included a test dataset itself. It was loaded and features were extracted to be fed into the model.



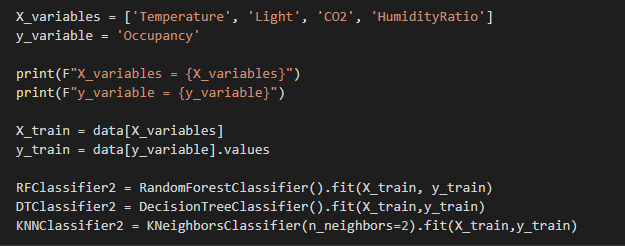
The built models were evaluated against the test data set using Sklearn metrics function. The results are shown below.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Area under ROC** | **Precision** | | **Recall** | | **f1-score** | |
|  |  | Class '0' | Class '1' | Class '0' | Class '1' | Class '0' | Class '1' |
| Random Forrest Classifier - 05 features | 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

The model was tuned by alternating the features set. Backward elimination was used for this, and ‘Humidity’ column was removed from the feature set.



## Model saving

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

The best model was saved for deployment using pickle. This model can be deployed standalone and exposed as via an API.

# Discussion

The machine learning problem which I have been trying to solve is to predict the room occupancy using the temperature, light, CO2 percentage and the humidity ratio. The model evaluation results show acceptable and good performance.

First, we will consider the raw models which were not tuned. The ROC values of all the models are higher than 0.85 which is a good indication. It is observed that all models have higher precision and recall values over both target variable classes. Hence, it can be assumed that the data preprocessing steps are properly done.

The Random Forrest classifier has outperformed other 02 models in terms of precision, recall and f1-score. Therefore, we can state that the model’s false positives and true negatives are lower. Also, the model is balanced with both majority and minority classes of the target variable having similar metric values. The dataset balancing should have caused this balanced performance.

Then we will compare the tuned models’ performance with initial models. It can be observed that the model metrics have slight improvements. The initial models had acceptable higher metrics therefore this is expected. We cannot expect drastic model improvements when the initial models already show good performance. The best model is selected as tuned Random Forrest Classifier. It can be used to accurately predict the room occupancy when temperature, light, CO2 percentage and humidity ratio are input.

Further improvements may be possible with new features derived from the present features. Also, data normalization can be done before feeding to the models.

# Conclusion

The machine learning model building pipeline was well followed during the project. It included dataset preparation, data preprocessing, model building and evaluation, model tuning and presenting the results. Each step has its importance to provide a better solution ultimately.

The results show that model can be deployed and used in production to give better predictions. It predicts both classes accurately. Hence, I think the outcome of the project is delivered.