Project 4

Anton Stråhle & Max Sjödin

28 december 2020

Introduction

This is a project for the course MT7038 which was given during the fall of 2020. In the project we'll examine some basic methods for binary classification on a granular level and then proceed with a more in depth analysis and discussion for those that seem to perform best. The methods that we will initially test are SVM, KNN and Decision Trees.

Data

We picked the `Occupancy Detection` data set as we wanted to work with a binary classification problem as this would allow us to apply most of the methodologies discussed throughout the course. As there are quite a few different binary data sets at UCI we specifically chose our data set as it had a sizeable number of instances as well as few, but intuitivley explanatory, features.

Attributes

The data `Occupancy Detection` data set includes snapshots of a specific room every minute throughout the course of a few weeks. The aim is to classify the current `Occupancy` of the room using the five features, `Temperature`, `CO2`, `Humidity`, `HumidityRatio` and `Light`, which are observed each minute. The first three features are quite self-explanatory but the two final ones could use some clarification. The `Light` in the room is the light intensity, measured in Lux whilst the `HumidityRatio` is vaguely described as a "derived quantity from temperature and relative humidity, in kgwater-vapor/kg-air".

Exploration

From a quick overview we see that the data set is quite unbalanced.

Table 1: Unoccupied

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
Temperature	1	15810	20.585	0.895	20.500	20.488	0.741	19.000	24.390	5.390	1.325	2.757	0.007
Humidity	2	15810	27.530	5.119	27.150	27.556	5.411	16.745	39.500	22.755	-0.030	-0.760	0.041
Light	3	15810	25.238	81.824	0.000	2.966	0.000	0.000	1546.333	1546.333	4.667	31.278	0.651
CO2	4	15810	604.997	253.027	511.000	545.863	102.299	412.750	2076.500	1663.750	2.442	5.839	2.012
HumidityRatio	5	15810	0.004	0.001	0.004	0.004	0.001	0.003	0.006	0.004	-0.151	-0.767	0.000
Occupancy*	6	15810	1.000	0.000	1.000	1.000	0.000	1.000	1.000	0.000	NaN	NaN	0.000

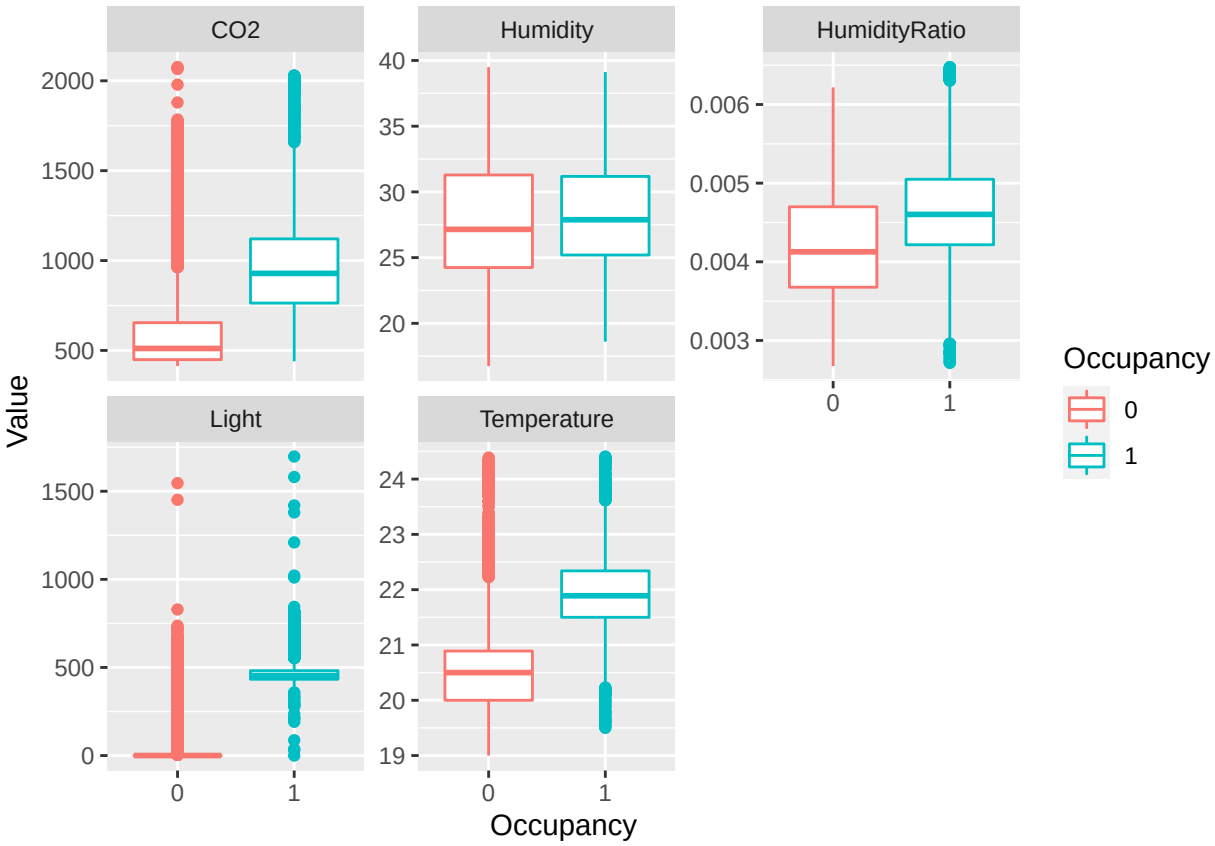
Table 2: Occupied

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
Temperature	1	4750	21.976	0.818	21.890	21.938	0.619	19.500	24.408	4.908	0.414	0.254	0.012
Humidity	2	4750	28.076	4.472	27.882	28.040	4.305	18.600	39.118	20.517	0.156	-0.323	0.065
Light	3	4750	481.967	94.704	454.000	461.597	31.135	0.000	1697.250	1697.250	2.953	17.353	1.374
CO2	4	4750	975.322	317.261	928.583	943.895	267.486	439.000	2028.500	1589.500	0.997	1.154	4.603
HumidityRatio	5	4750	0.005	0.001	0.005	0.005	0.001	0.003	0.006	0.004	-0.059	-0.062	0.000
Occupancy*	6	4750	2.000	0.000	2.000	2.000	0.000	2.000	2.000	0.000	NaN	NaN	0.000

We have about four times more unoccupied than occupied minutes. As the data is observed around the clock it is of course natural that the room is unoccupied during a majority of the day. Due to this quite severe inbalance we have to make sure that our training, validation and testing sets reflect this inherent property of the data. As such we decided to concatenate the three provided data sets (training, validation and testing) and split these up into balanced sets ourselves as the data providers seems to have just split the complete date by the timestamp which leads to a quite severe inbalance as a weekend is included (i.e. no Occupied observations for two days).

Beyond the poor splitting of the data into training, validation and testing we found not obvious inconsitensies in the data that needed addressing.

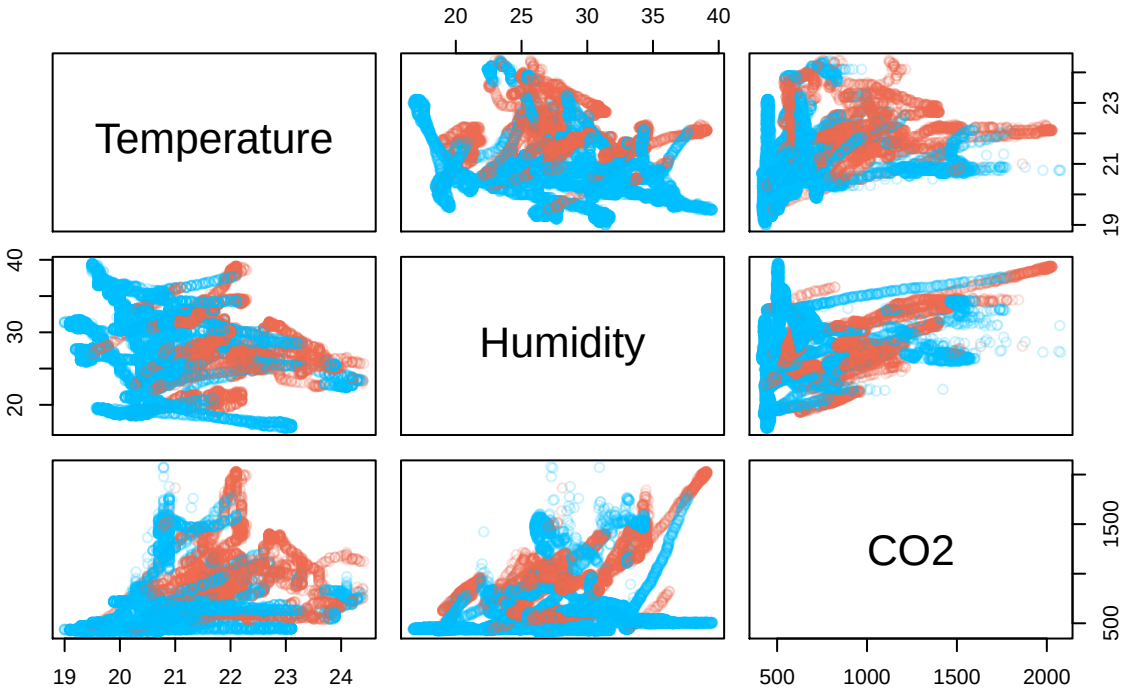
We also chose to standardize to allow for better performance of scale dependent classifiers, e.g. kNN or SVM. As some of the features include some very major deviations as can be seen in the figure below this choice to not standardize would surely impact the performance of these classifiers negativley.



As can be seen in both the figure and the tables above the feature `Light` seems to to quite well in describing the current occupancy of the room. This is quite evident as people rarely gather in a room with the lights turned off, and (hopefully) turn the lights off when they leave. This features solely dominates the others when it comes to classification and is, at least according to us, quite boring. As such we'll choose to excluded it in order to actually be able to perform a somewhat comprehensive analysis that doesn't just include the questions *Was the lights on or off?*.

As the `HumidityRatio` can be derived from `Humidity` and `Temperature` we exclude it from further analysis and examine the remaining three features.

Figure 1: Correlation between features



As we can see in the figure above the remaining features seem to behave quite nicely, although some could be considered to be somewhat correlated (e.g. `Humidity` and `CO2`). There seem to be some quite nice divisions of the occupancy status within the different features which should make adequate classification quite easy.

Methodology

As we're dealing with a binary classification problem there are some major routes that we decided to test out. Those being SVM, KNN and Decision Trees. Our idea is to examine all methods quite shallowly and then go a bit deeper for one or two that shows promise.

SVM

In Figure X above we noted that we seem to have non-linear data which should make the usage of certain SVM kernels useful in terms of classification.

Implementation

We examine three different kernels, a linear kernel, a radial kernel as well as a polynomial kernel. We use our aforementioned validation set in order to tune our cost C as well as the degree for our polynomial kernel. Using the package `e1071` and the function `svm` we attained the following results for the three scenarios.

Table 3: SVM Test Accuracies

Kernel	Cost	TestAccuracy
Linear	1e-01	0.831
Radial	1e+03	0.911
Pylynomial degree 3	1e+01	0.907

With a very coarse parameter search (grid search for the polynomial kernel) we achieved the results above. The linear kernel performs quite a bit worse than the radial and cubic polynomial kernels which further supports the idea that the data is in fact quite non-linear.

KNN

As noted previously the data seems to be quite non-linear and as KNN in a good non-linear classifier it should hopefully work well. Furthermore as KNN scales well with data we should hopefully be able to achieve quite good results given the size of our training set (approximatley 12000 observations).

Implementation

Using the package `Class` and the function `knn` we do a coarse search using our validation set for a good k.

The value of k which generates the best validation accuracy turns out to be 1. As the best kNN classifier, which performs extremley well with a test accuracy of 0.9440661, is a 1-NN this strongly indicates that our data is not very noisy at all, meaning that we should be able to achieve very high accuracies in one way or another.

Decions Trees

Using decision trees is a good idea since we have non-linear data and a simple binary classification problem.

Implementation

Constructing a simple decision tree resulted in a test accuracy of 0.9097763. This is can be further optimized by implementing bagging and boosting algorithms. Using the package `ipred` for bagging and the package `adabag` for boosting we attained the following results.

Table 4: Decision tree Test Accuracies

Method	Test.Error
Simple	0.910
Bagging	0.981
Boosting	0.964

When bagging trees we draw 1000 bootstrap samples and which resulted in the high test accuracy of 0.9807879 displayed in Table 4.

Analysis

Bibliography

Appendix