Occupancy Detection

Data Mining II
a.y. 2019-2020
S. Citraro, R. Guidotti, D. Pedreschi

Marta Fioravanti - 603574
Gianmarco Franchini - 607674
Francesco Santucciu - 599665

UNIVERSITÀ DI PISA

Data Understanding

Introduction

The data (originally organised in three datasets) describe several characteristics of a room with the objective of learning a model for room occupancy (i.e., whether the room is occupied by people or not). In the following section, we present a joint analysis of the three datasets.

Features

The data is described by 5 numeric features and a timestamp. The target variable, *Occupancy*, is also provided and is based on photographic detection of human presence in the room[1]. The total number of observations is 17895, with no missing values. A detailed analysis of each feature follows in which its semantic and predictive aspects are presented.

Date

This is a timestamp recording the date and time at which the observation was registered. The format is yyyy-mm-dd hh:mm:ss. This information will be crucial in discovering and factoring out attribute variations due to the time of day or season rather than to occupancy.

Temperature

This feature measures the temperature of the room in Celsius degrees. It is potentially a good predictor of human presence. However, its predictive capacity is impaired by:

- the number of people in the room: a single person by themselves might lead to an inappreciable increase in temperature;
- in-day and seasonal variations.

A correct usage of this variable for present purposes will therefore require correcting all variability not due to human presence.

Humidity

This feature measures the relative humidity[2] of the room. We remain agnostic as to its predictive potential, as none of us possesses any knowledge to this regard.

Light

This feature measures the light in the room in lux units. It is expected to be the best predictor of occupancy, barring the presence of a natural source of light, which would potentially skew the measurements and subject this variable to seasonal variations.

CO_2

This feature measures the concentration of CO_2 in the room in ppm (parts-per-million). Simply put, this is the ratio of CO_2 units to other aerial gas units (whichever the chosen unit). It is expected to be a strong predictor of room

1. Luis M. Candanedo, Véronique Feldheim 2016,'Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models', *Energy and Buildings*, vol. 112, p. 30.
2. For further details, see RelativeHumidity.

occupancy, given that normal respiratory processes in humans envisage the emission of CO2 as a byproduct.

HumidityRatio

This feature is derived from Temperature and Humidity[3]. It is measured in kgwater-vapor/kg-air. The same considerations as for Humidity apply.

Statistics

Basic statistics were computed on all numeric features to get a glimpse of their distribution. Subsequently, disaggregated statistics (i.e., basic statistics broken down by class label) were computed to get a sense of the difference between occupied and empty rooms. Finally, the disaggregated distribution of each numeric feature with respect to Occupancy was plotted, to get a sense of the predictive power of each feature. The results are shown below.

Basic statistics

	Temperature	Humidity	Light	CO2	HumidityRatio
mean	20.83	27.99	121.45	686.48	0.004258
std	1.04	5.17	202.21	313.67	0.000784
median	20.62	27.79	0.00	564.00	0.004343
min	19.00	16.74	0.00	412.75	0.002674
25%	20.10	24.89	0.00	458.75	0.003767
75%	21.50	31.86	217.92	792.00	0.004860
max	24.39	39.50	1581.00	2076.50	0.006476

The most striking fact certainly concerns the distribution of Light: the deviation between its mean and median is indeed telling of a skewed distribution, which will be further explored in the following sections. The measured statistics for Temperature, Humidity and HumidityRatio, on the other hand, hint at a normal distribution. CO2 seems to be caught in the middle, with a mean significantly higher than its median, but not as much as for Light.
The Pearson coefficients for linear correlations are as follows:

	Temperature	Humidity	Light	CO2	HumidityRatio
Temperature	-	-0.19	0.67	0.39	0.15
Humidity	-0.19	-	-0.06	0.28	0.94
Light	0.67	-0.06	-	0.39	0.18
CO2	0.39	0.28	0.39	-	0.43
HumidityRatio	0.15	0.94	0.18	0.4	-

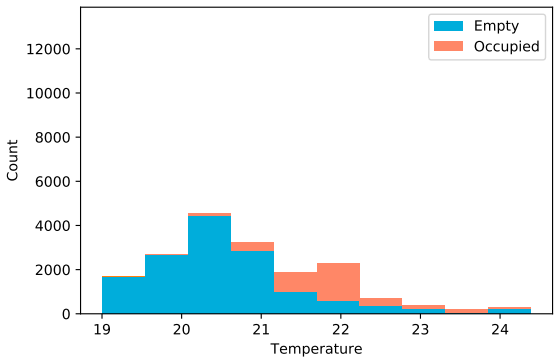
Predictably, Humidity and HumidityRatio present a very high Pearson coefficient (ca 0.94), which makes them redundant. However, given the low dimensionality of the dataset, we decided to keep them both in at least one of the Attribute Sets[4] that we later test.
As concerns the class, the data is fairly imbalanced, with only 3779 records belonging to class 1 (21.12% of the whole dataset).

3. See the dataset's repo, under Attribute Information.
4. From here onwards AS.

Disaggregated statistics and distributions

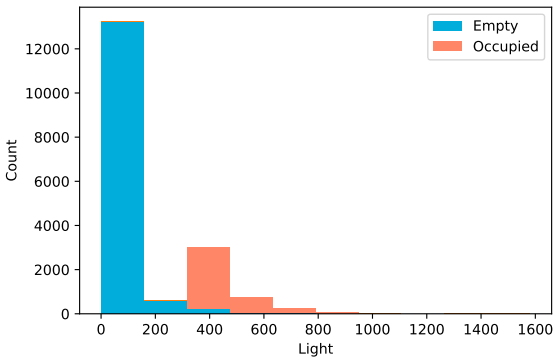
For brevity's sake, only a few statistics are presented and only for those attributes where the variation between occupied and empty rooms was evident.

Temperature



	mean	median	std
empty	20.54	20.39	0.92
occupied	21.86	21.80	0.74

Light



	mean	median	std
empty	26.18	0.00	81.41
occupied	477.43	449.5	90.88

Light is clearly the best predictor of Occupancy: almost all values lower than a certain threshold, tentatively 300, belong to class 0, whilst the opposite is true for higher values. Furthermore, both the mean and the median are significantly far apart for the two classes, with the standard deviation being sufficiently low to make this difference reliable across observations.

The situation is slightly less clear-cut with Temperature, where class 1 is almost absent below a threshold of approximately 20.7, but class 0 is present at all values (although preponderant, as was expected, at lower temperatures). Its table tells a similar story: although there is an appreciable difference between the mean and the median of the two classes, the standard deviation is too high for it to be significant.

Data Preprocessing

Dataset transformation

The original data came in 3 separate datasets. Since this division did not conform to our needs, we decided to join them in a single dataframe which we then split in training and test, accounting for 2/3 and 1/3 of the data respectively[5]. The data transformations described in the following sections were all based on the analysis of the new training.

Feature engineering

The original timestamp was parsed to obtain 10 separate time attributes:

-Date (dd-mm-yy)
-Time (hh:mm:ss)
-Year
-Month (in numbers: 1, 2...)
-Day (in numbers: 1, 2...)
-WeekDay (in words: Monday, Tuesday...)
-Hour
-Minute
-Second.

WorkingHours

We looked for a time-related pattern in room occupancy. A clear one emerged: the room was always empty on Saturdays and Sundays and from 6 p.m. to 7 a.m. approximately on all other days. A new binary variable was thus created to account for this fact, which was put down to the room occupants following a work schedule: the new variable was called WorkingHours and takes the value of 0 on Sundays and Saturdays and between 6 p.m. and 7 a.m. on all other days, 1 in all other cases.
The SMC (Simple Matching Coefficient) between WorkingHours and Occupancy for the training was 0.92 approximately, indicating a high correlation between the two.

Delta attributes

A more ambitious attempt at managing the time dimension was made with what we called the Delta attributes. From each numeric attribute except HumidityRatio (due to its high correlation with Humidity) a new attribute was extracted called "*Delta + name_of_the_original_attribute*".
For each attribute, the Delta attribute is measured as the difference between the value of the record and the median of the values of all class-0 records timed at the same hour. The idea is to compare the value occurring in the record with an empty-room scenario; only records from the same hour are chosen to neutralise the in-day variation of the original attributes. With this in mind, the median was chosen because robust to noise, as opposed to the mean.
The expectation is that the higher the delta, the more likely the room is to be occupied. Naturally, this does not neutralise all seasonality-related effects (as that

5. The task on time series then required a different split, given that this one was date-indifferent.
6. For example, the temperature will be averagely higher in July than in February.

would also involve taking into account the between-day variation[6], but goes some way towards doing so.

Feature selection

Various AS's were created. In the end, we used only two:

- AS1 keeps the original attributes plus WorkingHours: {*Temperature, Humidity, Light, CO2, HumidityRatio, WorkingHours*};
- AS2 discards the original attributes in favour of the Delta attributes plus WorkingHours: {*Delta Temperature, Delta Humidity, Delta Light, Delta CO2, WorkingHours*}.

Final considerations

The decision was made not to standardise the numeric attributes beforehand not to impair the interpretability of those classifiers that do not require standardisation (e.g. decision trees). It goes without saying that standardisation was performed for all those classifiers that do require it. The same reasoning applied to logarithmising Light and CO2.

Basic Classifiers and Regression

Training the classifiers

All classifiers, here and in subsequent sections, were tuned with either grid or randomised searches unless otherwise specified.
As replicability is not the main concern in this context, we eventually resolved not to list all the parameter values we iterated over for reasons of space. In a similar vein, for DT we only present the results of an average or otherwise representative model for the best-performing models of each k-fold cv[9]. All omitted information is of course available upon request.

Decision trees

For decision trees (DT), different *k* values for cross-validation were also explored. In the latter case, the best-performing models for each *k* were then selected[7].

k	criterion	max depth	min samples leaf	min samples split
3	gini	1 - 4	1	2 - 22
5	gini	3	1	2
10	gini	5 - 3	26 - 1	67 - 17
20	gini	1 - 4	1 - 11	2
Best-performing models for Decision Trees				

Logistic regression

For logistic regression (LR), we tested all possible attribute combinations and we performed a grid search on the regularization strength, obtaining 0.001 as optimal value for both AS's. We also tested different solver and penalties, but at the end we came up with the defaults (*lbfgs* and *L2*) because there weren't significant differences in performance. For both AS's it emerged that (Delta) Light and WorkingHours originated the most accurate classifier: probably, WorkingHours makes the classifier less sensitive to anomalies in Light. Therefore, we only report the results for this specific combination.

KNN

For KNN, the best number of neighbours turned out to be 10, the best metric Manhattan and the best weight 'uniform' in both AS's.

Naïve bayes

For Naive Bayes, we dropped HumidityRatio because of its correlation with Humidity. In this case, we did not employ a grid or randomised search as this was not applicable. However, we carried out 5-fold cross-validation and we experimented both with Gaussian and Categorical NB[8], which we respectively abbreviate as GNB and CNB. We report the results for both.

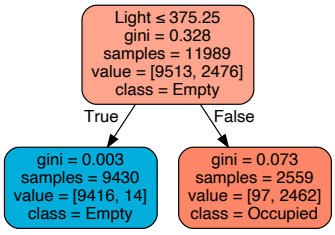
7. Where two values are present, the first refers to AS1 and the second to AS2. Otherwise, the same value was selected for both attribute sets.
8. For the latter, we discretized the attributes using Sturges' rule
9. The metrics used to select the best model was the F1 measure relative to class 1; where there was a tie between two models, the F1 measure relative to class-0 was used to break it.

AS1 results

	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
DT	0.99	1.00	0.99	0.99	0.97	0.99	0.98
LR	0.99	1.00	0.99	0.99	0.95	0.99	0.97
KNN	0.99	1.00	0.99	0.99	0.97	1.00	0.98
GNB	0.97	0.99	0.97	0.98	0.91	0.98	0.94
CNB	0.98	0.99	0.98	0.99	0.95	0.98	0.96
AS1 - Basic Classifiers Results							

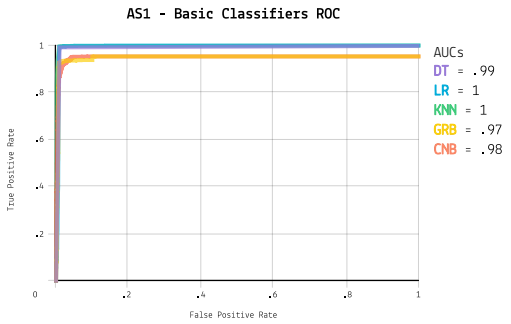
As the tables clearly show, the results are excellent both for class-0 and for class 1, and this despite the somewhat imbalanced distribution of the dataset (see Basic Statistics). The lowest value for all classifiers is class-1 precision: evidently, some class-0 records have similar enough values to class-1 records that they are misclassified. It is quite possible that the misclassified records are simply outliers the incorporation of which in our classifying model is of no interest. A possible explanation is that for those observations someone stayed in the room for a short time after the light had been switched off, or was alone, so that the variation in CO_2 and/or Temperature were unappreciable. It is also possible, of course, to think of a mistake in class labeling.

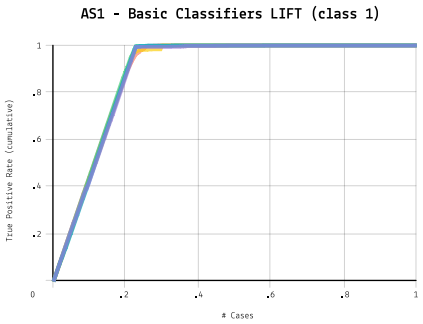
Another fact which is worth commenting upon is that NB is the worst-performing classifier, especially for class-1: as is visible from multilinear regression this is due to the fact that the attributes are not completely independent of one another.



For DT, a single attribute is sufficient to efficiently split the data, as we see in figure.

For brevity, we only report the ROC and Lift plots for AS1, since, as we see in AS2 results, the performances don't differ so much.





AS2 results

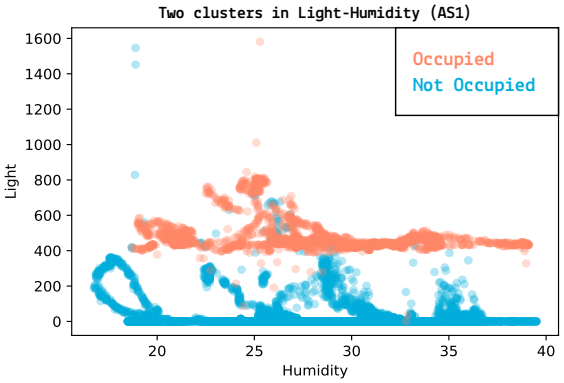
With respect to AS1, DT and KNN are basically unchanged. Conversely, performances for LR decrease whilst those of NB increase. This seems to suggest that there is less dependence between the attributes in AS2, which on the one hand facilitates NB and on the other hinders LR.

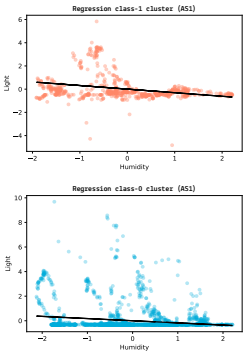
	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
DT	Same as AS1						
LR	0.98	0.99	0.99	0.99	0.96	0.96	0.96
KNN	0.99	1.00	0.99	1.00	0.97	1.00	0.98
GNB	0.98	1.00	0.98	0.99	0.94	1.00	0.97
CNB	0.98	1.00	0.99	0.99	0.96	0.98	0.97
AS2 - Basic Classifiers Results							

Regression

Linear regression

Light attribute demonstrated to have a very important role in the dataset. We tried to see if an attribute like Humidity that apparently has nothing to do with it was in a certain way related to Light, because this kind dependency would be an interesting information about the behaviour of the data. As in the previous sections, we performed the study on the two datasets, AS1 and AS2. Since the correlation between the two attributes in the whole dataset was poor (see Basic statistics), we tried to perform linear regression separately on each class, in order to see if there might be some hidden dependency. The regression was always done scaling the data.





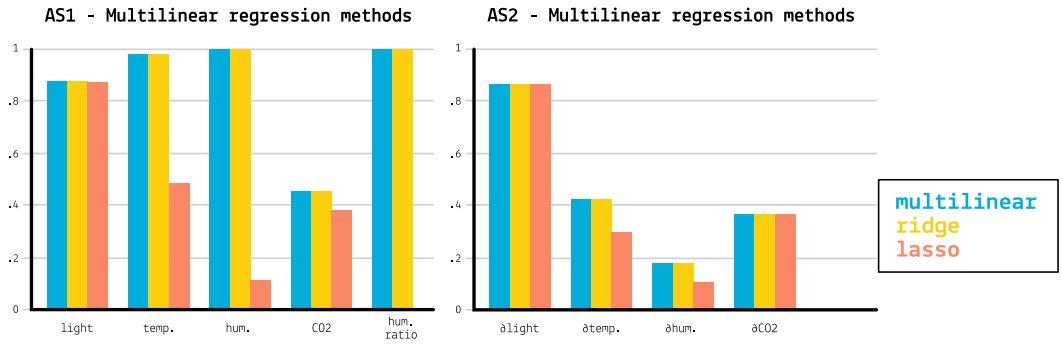
	R2	MSE	MAE
Occupied	-26.514	0.979	0.384
Empty	-45.726	0.963	0.758
AS1 linear regression performances			

	R2	MSE	MAE
Occupied	-31.730	0.996	0.523
Empty	-8.398	0.951	0.613
AS2 linear regression performances			

Despite this further processing, the R2 became negative, demonstrating the uncorrelation between the two attributes: in effect, if we look at the plots, we can see behaviours perpendicular to the regression line, meaning that the model cannot fit the data. We can therefore confirm the absence of a linear correlation between the two variables.

Multilinear regression

Since the total attributes in the dataset are quite few, we performed a multilinear regression finding the R2 for all countinuous attributes given the others. This task was performed via standard multilinear regression, ridge and lasso.
As we see, the performance of the models are definitely better in AS1, when the absolute values are available. Regression of Light has in both cases good results but we can see that in AS1 the values of Temperature, Humidity and Humidity_Ratio are very well discernible by the other values; the fact that their R2 is higher than the one



of Light confirms that this last attribute has a primary importance in the dataset. Another hypothesis was that Humidity Ratio also is important, but removing it from the analysis we didn't obtained significant changes.
The fact that Lasso has so a poor performance is probably due to the fact that it is trying to do too much economy in predictors.

	R2	MSE	MAE
Multilinear	0.862	0.138	0.195705
Ridge	0.862932	0.138914	0.195712
Lasso	0.86285	3879	32.519
AS1 regression performances			

	R2	MSE	MAE
Multilinear	0.874	0.126	0.192
Ridge	0.874	0.126	0.192
Lasso	0.871	5256	39.719
AS2 regression performances			

Dimensionality reduction

Variance threshold, Univariate Feature Selection (with ANOVA F1-value), Recursive Feature Elimination

For reasons of space, we do not dwell on details concerning the implementation of these three methods and only present the most interesting results and conclusions.
The variance threshold method confirmed its limits: although it certainly makes sense to eliminate features with zero variance, in all other cases variance per se is not indicative of predictive power. Indeed, Light and Delta Light were the first attributes to be ruled out (at thresholds 0.036 and 0.015 respectively[10]) in spite of being the best predictors of occupancy. This lowered performance scores (F1 for class 1 decreased to 0.96 for AS1 and 0.94 for AS2). Interestingly, AS's {*Humidity, WorkingHours*} (from AS1 with threshold 0.05) and {*DeltaHumidity, WorkingHours*} (from AS2 with threshold 0.035) performed quite differently for decision trees.

	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
AS1 {hum., work_h}	0.94	0.96	0.96	0.96	0.87	0.87	0.87
AS2 {delta_hum., work_h}	0.97	0.99	0.98	0.98	0.92	0.95	0.94
Results given by decision trees optimised through a randomised search.							

This seems to point to the superiority of Delta Humidity over Humidity, given that WorkingHours is the same in the two AS's. Evidently, neutralising the in-day variations of Humidity allowed those variations due to human presence to become more easily detectable.
Another interesting fact was that when only WorkingHours was used, the recall for class 1 peaked (0.98) for decision trees, despite all other scores being at a minimum. This means that whilst it sometimes happens that the room be empty during what we roughly established to be the working hours (precision for class 1 was 0.74), it is almost never occupied at non-working hours.
UFS selected Light/DeltaLight and WorkingHours, whilst RFE selected Light/DeltaLight. They both performed comparably to classifiers run on the whole dataset (for AS1, the decision tree scores exactly the same). This is quite natural and needs no further comments, given that for decision trees Light by itself was enough to achieve optimal results (see results of DT on AS1).

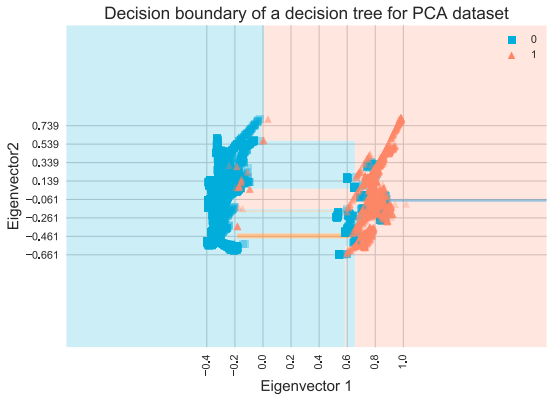
10. All attributes were scaled beforehand.

PCA

Prior to running PCA, min-max scaling was applied to numeric features. For decision trees[11], results are slightly worse than those obtained on the original AS's, even when only Light/DeltaLight was used.

	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
AS1	0.98	0.99	0.96	0.99	0.95	0.95	0.95
AS2	0.98	0.99	0.98	0.98	0.93	0.96	0.95

Decision trees scores on reduced AS1 and AS2.



For AS1, the visualisation of the decision boundary sheds some light onto why the performances slightly decrease. Indeed, the boundary in figure consists of a rather bizarre line that probably overfits the training.

Imbalanced Dataset

Imbalancing the data

Both the training and the test sets were randomly imbalanced so as to obtain a class composition of 96 vs. 4 % (class-0 and class-1 respectively).
Various techniques were used in order to try to rebalance the dataset: random undersampling, Condensed Nearest Neighbours, random oversampling, SMOTE oversampling, class weights and meta-cost sensitive classifiers. For reasons of space, we do not dwell on details concerning the implementation of these methods and only present the most interesting results and conclusions concerning AS1.

Conclusions

Even when the dataset was highly imbalanced (so prior to any rebalancing), the performance was reasonably good.

Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
0.99	1.00	0.99	1.00	0.84	0.98	0.90

Table showing the average performance of decision trees on the imbalanced dataset.

As suggested before (see Basic classifiers results in AS1), this points to clearly separated classes, which are easy to tell apart even when only a few records are available for one of the two[12]. A class-1 trend that was already visible for the original dataset is a relatively low precision coupled with a very high recall. This means that whilst it is very hard to mistake an occupied room for an empty one, the opposite is fairly frequent. This might suggest an in-class division within label-0, with a group of records easily identifiable (the majority) and a smaller, but non-negligible, cluster of observations more easily misclassified. A possible explanation for this minority

11. In this case, no grid or randomised search was run to tune the trees.
12. Also see Sun Y., Wong A., Kamel M. 2009,'Classification of imbalanced data: a review', *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687-719 and particularly 690-691.

group is that it captures the cases where no one was in the room for a very limited amount of time in between two busy times (e.g., for a quick break): the light might have stayed switched on and the other values registered no appreciable variation. As concerns rebalancing methods, oversampling performed generally better than undersampling, with random oversampling being the only one to outperform the classifiers on the imbalanced dataset (see previous table).

	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
Random und.	0.98	1.00	0.98	0.99	0.69	0.99	0.81
CNN	0.99	1.00	0.99	1.00	0.82	0.99	0.90
Random over.	0.99	1.00	0.99	1.00	0.86	0.98	0.92
SMOTE	0.99	1.00	0.99	1.00	0.82	0.99	0.90

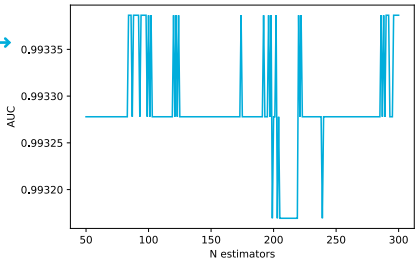
Table showing the average performance of decision trees on the re-balanced dataset. For CNN, we did not average over the performance recorded for a 5-iteration randomised search as it was starkly lower than the others and therefore the clear result of poor parameter tuning.

The impression is that undersampling (especially when random) greatly reduced the identification capability of the classifier on class-0, causing some 0-records to be mistakenly classified as 1 and thus reducing the precision for class-1. That random oversampling perform better than SMOTE was unexpected and might be caused by the fact that the synthetic records generated by SMOTE increased the frequence of a class-1 subconcept that class-0 records were more likely to be misclassified in.

Advanced classifiers

Training the classifiers

For SVM and NN, all numeric attributes were standardised with z-score normalisation before training and testing the classifiers.
As to parameter tuning, we employed grid/randomised searches for all classifiers. For both linear and non-linear SVM we also experimented with the same *k*s for cross-validation as for DT; due to space contraints, we are unable to list all of the various best combinations we obtained. Anyhow, it is still worth mentioning that for linear SVM the best value of C[13] was always equal to 1, except for AS1 with k =3, for which it was 11; for non-linear SVM, instead, there was no preferred kernel for AS1, whilst all searches on AS2 returned *sigmoid* as the best kernel function.
For NN's, we also split the training set in training and validation (70% and 30%) to have an ulterior validation phase. For both AS's, the best penalisation parameter for the SNN (Single-layer NN) was 10^{-6}; for the MP (Multilayer Perceptron), the layer size was 500[14], , the activation function was relu, and the learning rate 0.001 with optimizer *Adam* and a value of momentum of 0.9; for DNN's, the best results were given by class weights equal to the inverse probability of the classes, a dropout of 0.1 and either batch gradient descent (AS1) or a minibatch equal to half the training set (AS2). The nets were built with a layer size of 128 for the first layer and 64 for the second[15], *relu* as the activation function for the hidden layers and *sigmoid* for the output one, and a learning rate of 0.001 with optimizer *Adam*.
For Ensemble classifiers, we first ran each of them with default parameters except for the numbers of estimators/iterations, which we determined with the aid of graphs like the one in figure; then, we found the best parameters including the numbers of estimators/iterations with a randomised search. Finally, for bagging and boosting, we tried different base classifiers from DT (NB, MP, SVM and RF), eventually selecting RF as the best performer. We report the results for all two/three model configurations.



AS1	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
SVM	0.99	0.99	0.99	0.99	0.97	0.98	0.97
SNN	0.94	0.93	0.99	0.96	0.97	0.75	0.85
MP	0.99	0.99	0.99	0.99	0.97	0.98	0.98
DNN	0.94	0.78(general)	1.00(general)	-	-	-	-
AS2	**Accuracy**	**Precision (0)**	**Recall (0)**	**F1 (0)**	**Precision (1)**	**Recall (1)**	**F1 (1)**
SVM	0.99	1.00	0.99	0.99	0.97	1.00	0.98
SNN	0.93	0.93	0.99	0.96	0.96	0.70	0.81
MP	0.99	1.00	0.99	0.99	0.97	0.99	0.98
DNN	0.97	0.87(general)	1.00(general)	-	-	-	-

Tables showing the results of SVM and NN on both AS's. For a given AS, the average performances of linear and non-linear SVM were identical.
Unfortunately we weren't able to set the metrics of DNNs to obtain class-specific performances.

13. Which was searched over the following range: [1, 1000) with steps of 10.
14. Contrary to the other parameters, the size of the hidden layers for all NN was not tuned with a grid or randomised search due to the high computational cost. Instead, we tried to get a ballpark estimate of the right size by trying out values far away from each other (like 64 and 1000 for DNN's) and seeing which gave the best results.
15. We also attempted to build a net with 3 hidden layers, but performances were much lower, probably because the gradient vanished/exploded.

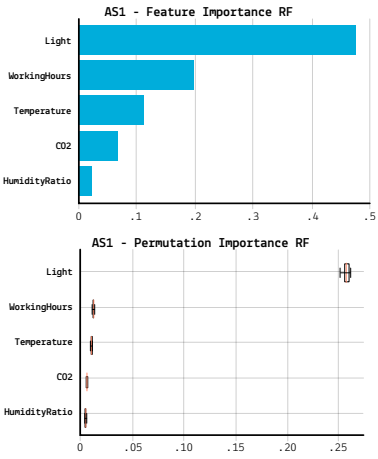
Results and conclusions: SVM and NNs

The single-layer network is clearly the worst performer: evidently, the simplicity of the input decomposition and processing was unsuitable for capturing the class division. For AS2, class-1 recall and class-0 precision are better than when using AS1 both for SVM and MP. An increase in performance when using AS2 may also be seen for the DNN. This suggests that AS2 aids the classifiers in recognising class-1 records. Interestingly, class-1 precision is still equal to or lower than 0.97, like for basic classifiers (see results of basic classifiers on AS1 for a possible explanation).
As to linear versus non-linear SVM, neither seems to outperform the other one. However, it is worth bearing in mind that the search over non-linear SVM was not exhaustive (we only ran randomised searches as opposed to grid ones for linear SVM), so that a better tuning of the parameters might be found. Overall, given the computational cost of building a non-linear SVM model, decision tree classifiers or linear SVM might be preferable.

The single-layer network is clearly the worst performer: evidently, the simplicity of the input decomposition and processing was unsuitable for capturing the class division. For AS2, class-1 recall and class-0 precision are better than when using AS1 both for SVM and MP. An increase in performance when using AS2 may also be seen for the DNN. This suggests that AS2 aids the classifiers in recognising class-1 records. Interestingly, class-1 precision is still equal to or lower than 0.97, like for basic classifiers (see section results of DT in AS1 for a possible explanation).
As to linear vs. non-linear SVM, neither seems to outperform the other one. However, it is worth bearing in mind that the search over non-linear SVM was not exhaustive (we only ran randomised searches as opposed to grid ones for linear SVM), so that a better tuning of the parameters might be found. Overall, given the computational cost of building a non-linear SVM model, decision tree classifiers or linear SVM might be preferable.

Results and conclusions: Ensemble Methods

Random Forest



The most important feature is *Light*, and this is demonstrated both by the permutation importance and by the feature importance plots. In AS2 we notice that Delta Light is even a more important feature.
Although the models obtained with a RandomisedSearch have a best AUC score, the ones obtained with default parameters are more sensitive and are as precise as the others.
As we can see, from the AS2 tables, how the default model has a better class-1 Precision in AS2 than the one gotten with a RandomisedSearch CV: this means this model makes fewer mistakes in classifying rooms as occupied.

AS1	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)	AUC
Default	0.99	1.00	1.00	1.00	0.98	0.99	0.99	0.993
Random Search	0.99	1.00	0.99	1.00	0.98	0.99	0.99	0.994
AS2	**Accuracy**	**Precision (0)**	**Recall (0)**	**F1 (0)**	**Precision (1)**	**Recall (1)**	**F1 (1)**	**AUC**
Default	0.99	1.00	0.99	1.00	0.98	1.00	0.99	0.995
Random Search	0.99	1.00	0.99	1.00	0.97	1.00	0.98	0.994

Table showing the performances of Random Forests in both ASs

Bagging

The default model is obtained with 100 estimators; the RandomisedSearch model has 132 estimators; the bagging+RF has 200. The RF used as internal estimators for bagging was the best RF obtained in the previous section.

AS1	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)	AUC
Default	0.99	1.00	1.00	1.00	0.98	0.99	0.99	0.992
Random Search	0.99	1.00	1.00	1.00	0.98	0.99	0.99	0.992
Bagging + RF	0.99	1.00	0.99	1.00	0.98	0.99	0.99	0.993
AS2	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)	AUC
Default	0.99	1.00	0.99	1.00	0.98	0.99	0.99	0.994
Random Search	0.99	1.00	0.99	1.00	0.98	0.99	0.98	0.99
Bagging + RF	0.99	1.00	0.99	1.00	0.98	0.99	0.99	0.994

Table showing the performances of Random Forests in both ASs

Boosting

AS1	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)	AUC
Default	0.99	1.00	0.99	1.00	0.98	0.99	0.99	0.993
Random Search	0.99	1.00	0.99	1.00	0.98	0.99	0.99	0.993
Boosting + RF	0.99	1.00	1.00	1.00	0.98	0.99	0.99	0.993
AS2	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)	AUC
Default	0.99	1.00	0.99	0.99	0.97	0.99	0.98	0.9897
Random Search	0.99	1.00	0.99	0.99	0.97	0.99	0.98	0.9893
Boosting + RF	0.99	1.00	0.99	1.00	0.98	1.00	0.99	0.9849

Table showing the performances of Boosting in both ASs

Conclusions

The AS2 is more appropriate than AS1 as we can see from the AUC_scores of the models; we can also see how the RF models itself could provide the best results in terms of performance.

Time Series Analysis

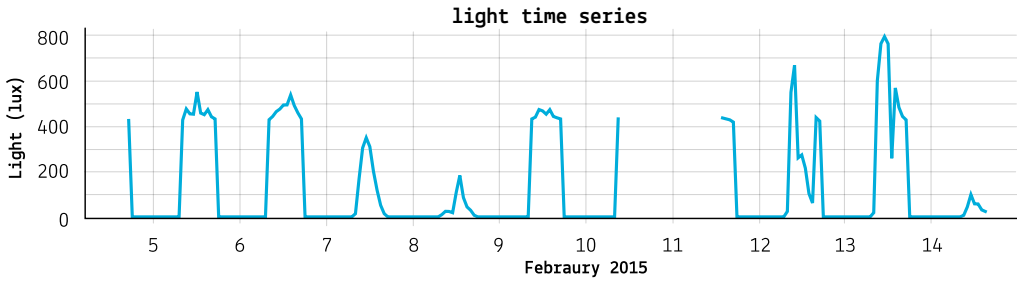
Data preprocessing

As was anticipated (see Dataset Transformation), the split we used previously could not be kept in this section.
Therefore, we performed a new split, with 70% of the records (the older ones) going into training and 30% (the more recent ones) being set aside for testing.

Univariate time series

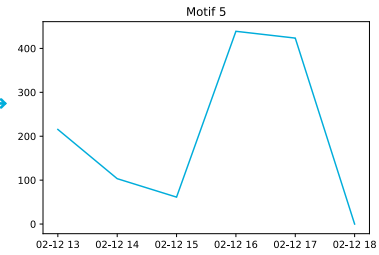
Time series extraction

We selected a feature to study across time (we report the results for Light) and eliminated all other features.
Subsequently, we resampled observations over hourly[16] windows, selecting the median[17] as synthetic indicator.



Motifs

Motifs and anomalies were mined with window size equal to 6.
We extracted the top 6 motifs and anomalies. As per usual, we only present the most interesting results.
Most of the detected motifs were uninteresting, as they only captured the descending slope of the daily cycle and the plateau at value 0 that ensues (i.e., evening and night). Only motif 5, which we've isolated in figure differs.

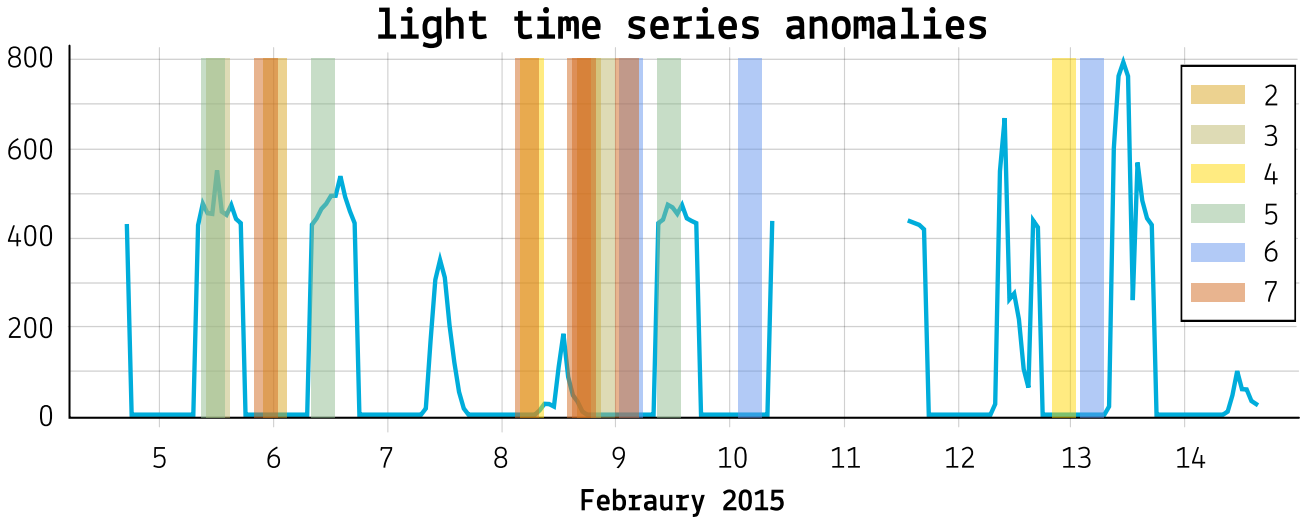


This motif captures something that happens on two consecutive days (the twelfth and the thirteenth), seemingly in the middle of the day: first a rather abrupt darkening of the room, followed by partial recovery and then another diminution of luminosity. It is plausible to assume that these periods coincide with a lunch break, during which most light sources in the room would have been switched off. The fact that the recovery is only partial would then be due to the natural decline in luminosity occurring over the second half of the day.

Anomalies

Some anomaly occurrences found by the algorithm (in the following figure) are puzzling, as they would rather be expected to figure amongst the motifs (e.g. the yellow and blue rectangles on the right).

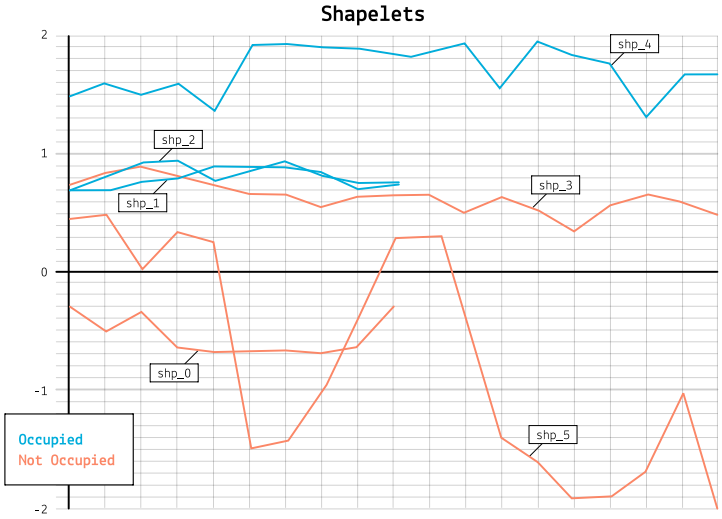
16. We chose hourly intervals because of the natural behaviour of light, which loops in daily cycles. It seemed like the fair compromise between obtaining meaningful information concerning the temporal evolution of the variable (which a larger window might have concealed) and not getting bogged down in too much detail (which a shorter window might have produced).
17. The median was preferred to the mean because of its greater robustness to noise.



While the yellow rectangle may be assumed to be featured as the closest neighbour of the actual anomaly (the other rectangle of the same colour), this is evidently not true for the blue ones.
Indeed, the only anomaly that might really point to an unusual pattern is the one in green. Especially the first occurrence of this anomaly describes a rather bizarre fluctuation in light, which looks like a variation on the surroundings of motif_5 (see figure motif_5): rather than there only being a partial decrease followed by a partial increase to the right of the daily light peak, the same pattern (partial decrease followed by increase) is mirrored on the left of the peak. Similarly to what was proposed for motif_5, we might surmise that a lunch break caused some sources of light to be obscured and thus removed a level component from the series for all the time passed between the two smaller peaks.

Shapelet discovery

Shapelet mining



The complete time series was split in subsequences labeled with a unique class; we eliminated the shortest amongst them, whilst the others were resampled with interpolation in order to make them of equal length[18]. The extracted shapelets perfectly split the two classes and are shown in figure.
As we see, sequences that tend to decrease and are located at lower values are associated with class-0, while the ones which describe vaguely convex lines at high values represent class 1. Behaviours like in shp_3 or shp_5, that exhibit a decrease in Light, could be associated to the end of the day (outside of working hours). Shp_0, on the other hand, might capture a brief period of emptiness between two occupied

18. Unfortunately, this means that we're unable to comment on the length of the extracted shapelets.

moments: the light is switched off (descending portion of the curve) when the the room is left and switched back on (ascending portion of the curve) when the someone reenter.

Time series classification

Univariate classification

We tried to classify the sequences using distance-based classifiers (DT and NN) reducing the dimensionality of the data with a PCA, a feature-based DT with parameters obtained from a grid search on the most common statistical attributes, and an LSTM. For this last model many preprocessing strategies architectures were tested: the data was transformed in 2 hours sequences and then smoothed; it came up that best results were obtained adding a SAX approximation to a smoothing of 15% and a structure with 5 LSTM layers (nodes : 64,32,16,8,5), a *l2* regularizer (3%), and 5 neural layers (nodes: 32,16,8,4,2).
For shortness we report only the best results for each classification strategy.

	Accuracy	Precision (0)	Recall (0)	F1 (0)	Precision (1)	Recall (1)	F1 (1)
DT distance based	0.88	0.75	1.00	1.00	0.83	0.86	0.91
DT feature based	0.88	0.75	1.00	1.00	0.83	0.86	0.91
LSTM	0.85	0.95	0.61	0.86	0.83	0.91	0.71

Table showing the performances of univariate classification of time series

Multivariate classification

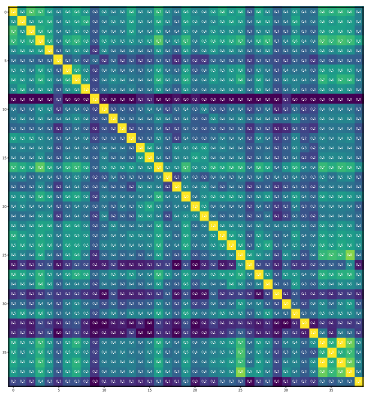
The same LSTM used before was also used to perform a classification taking into account all the original attributes, and, only increasing by 2% the regularizer we obtained a model that perfectly fit the data. However, the performance of the model should be tested on data from different months to understand if it is not only representing the behaviour of a specific period. Also this model could be used to improve the energy efficiency of the offices through smart app build from this kind of model, and because it is based on environmental attributes such as the quantity of light or the quantity of CO2 in the room, it respect the privacy laws.

Time series clustering

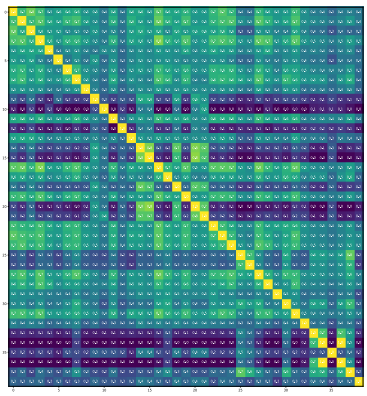
Preprocessing

The time series obtained for the shapelet discovery were used to do clustering and study the groups having information about occupancy class. We performed, offset translation, amplitude scaling and a very soft noise smoothing.
We obtained shape-based clusters using euclidean distance, manhattan, DTW; then we looked for feature-based ones using custom set of features like the measures of central tendency, the percentiles' values (10, 25, 50, 75, 90) and the interquantile range, but also with automatic methods like *tsfresh*, but these with bad results; finally, we used some compression methods and the only acceptable methods where PAA and SAX.
For each method, we took into account the silhouette in order to find the optimal k and we measured inertia. Then we took the best results and studied the final clusters. In order to choose the best shape-based distance metric, we plotted a similarity matrix between all the time series for each metric, and we observed how

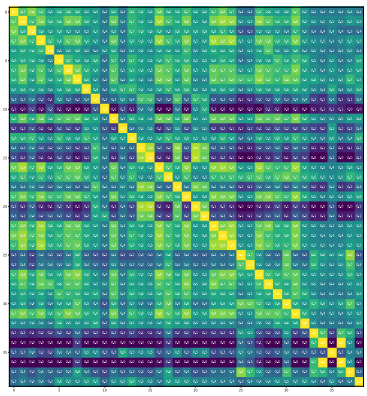
captured similarities



Manhattan



Euclidean



Dynamic Time Warping

Each cell represents the similarity between two time series.

DTW is the most sensitive: in fact, as we see, the yellow areas are more distributed.

Clusters observation

The most interesting set of clusters was obtained via DTW, because two groups over three contained only time series of a single occupancy class. This permitted to compare the features of the groups and make considerations about the classes.
As expected, the mean and median values for the time series of cluster 1 have mean and median strongly lower than the ones in the cluster 2. All the time series with non extreme values are instead contained in the cluster 0 and are probably responsible of the misclassifying errors.

	Mean	Median	Std	Min	Max
c0 (class-0)	256.91	206.68	228.89	0.0	674.85
c0 (class-1)	504.32	465.67	94.39	236.5	971.39
c1 (class-0)	84.71	13.00	111.46	0.0	497.0
c2 (class-1)	508.04	454.00	131.4	31.00	1550.97

Statistical description of the most interesting clustering (Light)

Time series forecasting

Preprocessing

In forecasting, we adopted two different strategies to tackle missing values. With SARIMA, we only used the data up to the first missing value to train the forecasting model and then forecast both the missing values and the last portion of the training, for which the real values were available, in order to get a rough estimate of the reliability of the predictions. Given the poor results we obtained with ARIMA, which may have been due to the scarcity of training data, with all other methods we estimated the missing values by interpolation[18] and then forecast the records contained in the test set.

Stationarity analysis

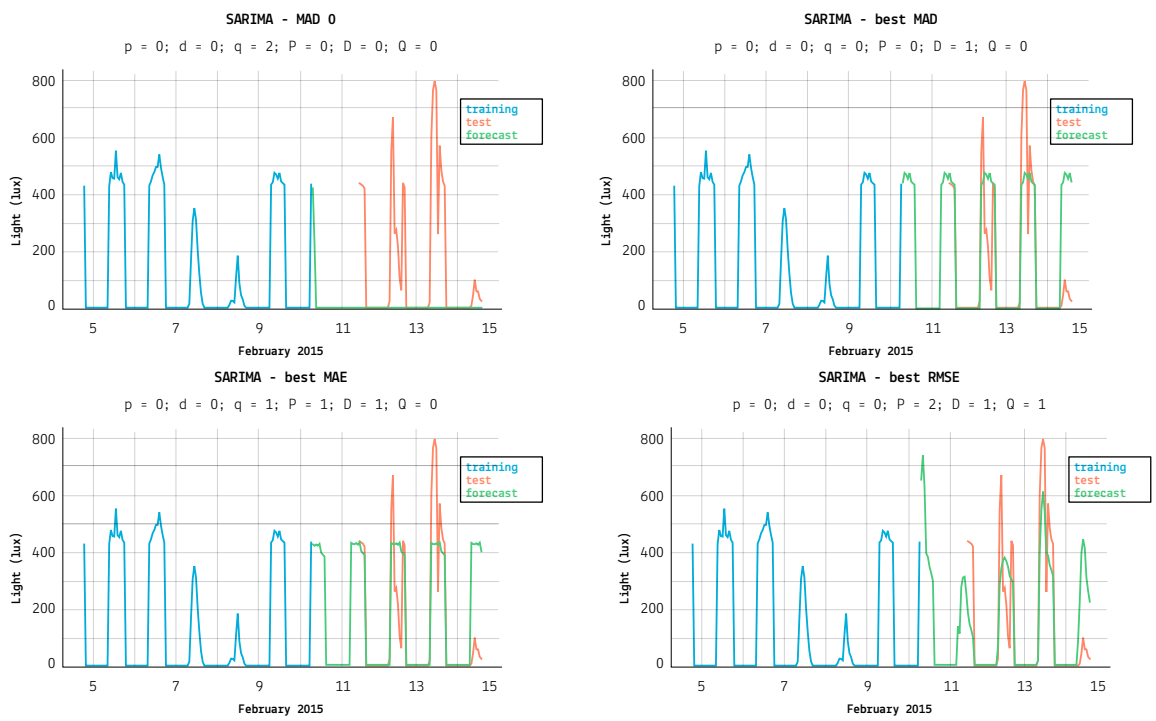
The Dickey-Fuller test statistic (-3.35) placed the confidence level of a stationarity verdict at 95%.This reasonably assured us that there was no trend to be eliminated. However, the series was obviously seasonal (see the time series plot where a daily cycle clearly stands out). Thus, we employed methods able to natively deal with seasonality.

SARIMA forecasting

We trained the SARIMA model with ad-hoc grid search (with no cross-validation) over all of its parameters except m/S, the period/season, which was fixed at 24. Overall, we tested 995 SARIMA models. We selected the three best respective ones as per the MAE, MAD and RMSE.

p	d	q	P	D	Q	R2	RMSE	MAE	MAD
0	0	1	1	1	0	-133.5	158.92	**82.77**	7.85*10-9
0	0	0	0	1	0	-144-29	166.25	83.7	**0.00**
0	0	0	2	1	1	-177.81	**146.14**	87.65	8.06*10-12
Table showing SARIMA performances									

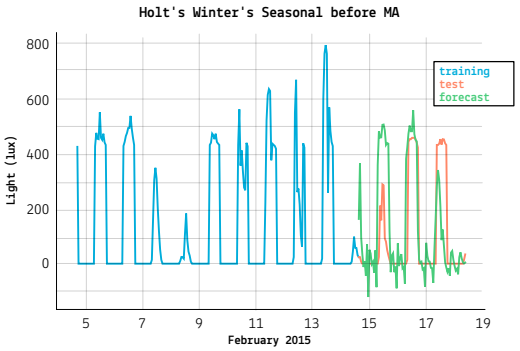
Evidently, none of the best models is a good fit (see the R2 score). In this regard, the MAD is misleading: because of the seasonal nature of the series, there's a recurring set of values (the zero ones) which are very easy to predict. The median error happens to capture the error of the prediction for one of these values. Therefore, it attains a minimum or in any case a very low value whenever this particular prediction is equal to 0 or very close it (which is often the case). Indeed, many other models that minimised it did a very poor job of capturing the structure of the series (see for example MAD = 0 plot, where MAD is 0, but the model is clearly unsuitable).



The difficulty of the forecasting models in predicting the correct values is also easily explained: based on the data in the training set, it seems impossible to guess the gradual increase in the peaks which occurs over days 13 and 14 and is the most visible deviation from the previously recorded cycle. Whether this variation is random (and therefore truly impossible to predict) or is part of a larger cycle (for example, a weekly one) could only be decided with the aid of a larger training set. Nonetheless, like Holt-Winters' seasonal method, the best SARIMA configurations do capture the correct cyclical period, aptly alternating zero and non-zero Light values.

Holt's Winter's seasonal method

As said before, preprocessing strategy chosen was the second (by interpolation). Since the first attempts' performances where quite poor (the R2 was 0.06), we wondered that it was caused by the noise in the test data. We then applied the Moving Average Smoothing and retook the test but obtained a decrease of performance (R2 became negative).

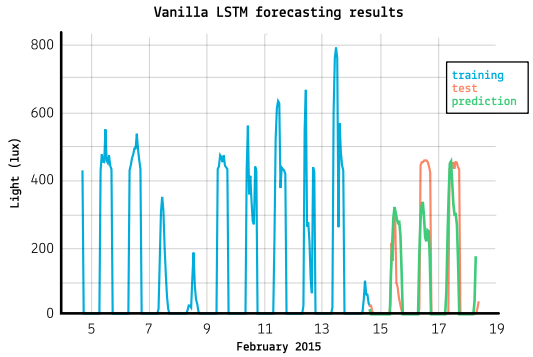


	R2	RMSE	MAE	MAD
Before MA	0.06	40.4	109.66	174.35
After MA	-178	34.82	98.61	154.96

Holt's Winter's forecasting performances

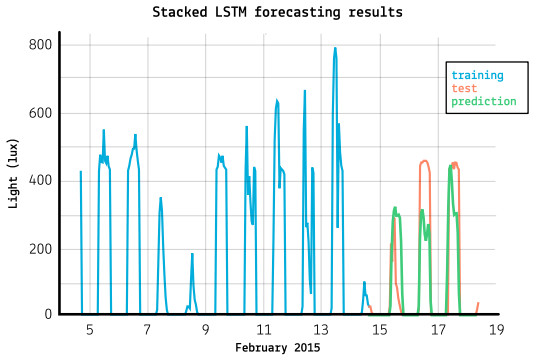
LSTM forecasting

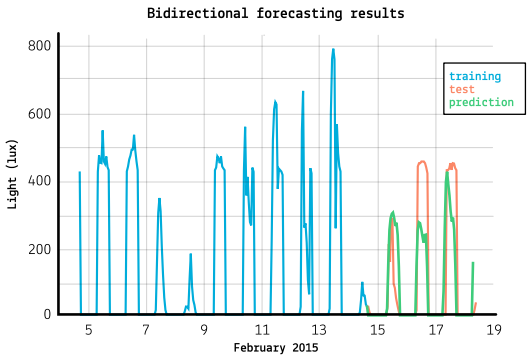
Another experiment we did was to train an LSTM and try to predict the behaviour of the Light attribute.
The training series data was transformed in a format that was suitable for the LSTM: with a sliding window we listed a series of subsequences, paired with the data that follow them. In this way the LSTM was made able to learn the succession of value given some input.
After many attempts we came with: a simple Vanilla LSTM (depth 300), a Stacked one (two LSTM, one deep 300, the other 100) and finally with a Bidirectional one (depth 300). All the models included a dropout of 0.3 and an early stopping with patience equal to 100. We suddenly noticed how the performance in the test step went better: this is due to the fact that a bigger training set helps to reach more accurate results.



	R2	MSE	MAE
Vanilla	0.6	0.059	0.135
Stacked	0.619	0.056	0.128
Bidirectional	0.585	0.062	0.140

LSTM forecasting performances





Sequential pattern mining

We decided to get a deeper view of the transition moments (from one class to another) in the time series.

Preprocessing

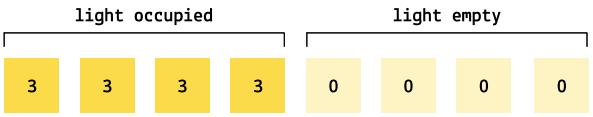
The dataset was prepared normalizing each attribute and approximating with SAX each column time series; the optimal number of symbols we found is 5. The new series were approximated to be a quarter of the original length. Obviously, the attribute Occupancy didn't need any of this preprocessing, since it is binary. We then created columns containing a human-readable version of the approximation for each attribute: we made a translation in words of the approximated value (e.g. Light was symbolized by 5 tokens, from light_0 for the lowest value to light_4 for the highest one).
Finally, two datasets were exported: one for the transitions from occupied to not occupied and the other from not occupied to occupied. The idea was to separately study the two types of event. The datasets consisted in many groups of observations, composed by 4 records of one class and 4 of another; this means that for each transition we have a span of 8 consecutive hours.

Analysis

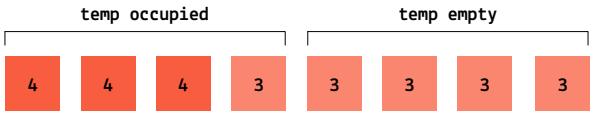
Both datasets were analyzed performing a GSP on each combination of attributes, starting from the couples containing Occupancy and another attributes and progressively increasing the number of variables to study (always including the class attribute). This is because analyzing all the attributes together, as we did at the beginning, didn't lead to interesting results.

From occupied to empty

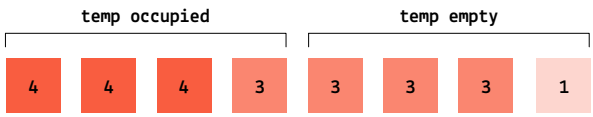
As expected we observe with 100% of support that light suddenly passes from an high value (3) to zero when the building is left.



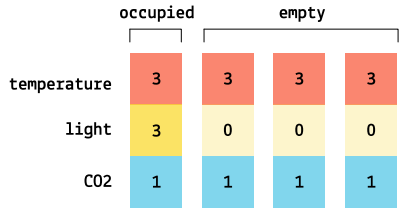
We however are interested in finding other patterns that could help us to get a better understanding of the system. With support 85.7% we observe the following pattern:



This makes more sense if we take a look also to the following, with support 71.4%:



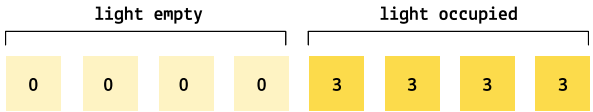
It's clear that the more the building is empty, the more the temperature falls. The fact that sometimes we register a decrease of temperature while the building is yet occupied is perhaps due to different working timetables or to the cleaning service after the working hours.
For what concerns the CO2, probably its index falls down quite slowly, because studying the attribute we see that only with 42.8% of support we observe a decrease when the building is left. The same can be said for Humidity Ratio, that with the same support shows a fall in the same circumstances. Humidity attribute, instead, doesn't seem to change at this scale, so we didn't find any interesting pattern.
With grouped attribute is evident that the stronger variables are Light, Temperature and CO2. For example, with support 57.1% we observe:



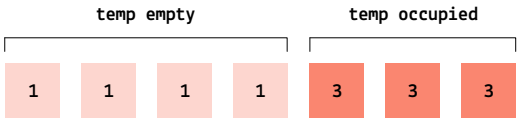
Clearly, the support of combined attributes falls, because the probability to encounter an anomaly grows up, but this pattern summarizes what we just said: the only attribute that suddenly changes to the class variation is Light, while the others take more time to change.

From empty to occupied

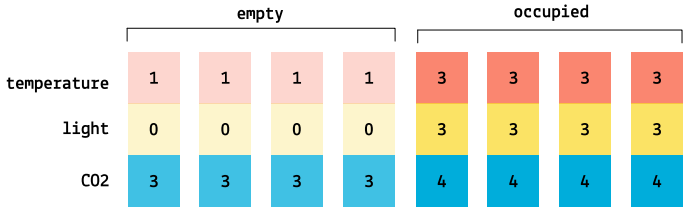
The dataset of transitions from empty to occupied confirms another time the importance of light information, because with a support of 100% we have an immediate variation from 0 to light_3 when someone arrives.



With a support of 42.8% we observe a variation in temperature from 1 to 3, as shown in the following pattern:



From this, we can notice that the temperature tend to increase quite rapidly when the building starts to be populated, while it decreases slowly when people left.
The variation of Humidity Ratio in this type of transition falls from the 42.8% of support observed in the previous section to 28.5%, while an increasing of CO2 is observed the 42.8% of times. Humidity seems not to be strongly influenced in a range of 4 hours by the change of state, as we see that the 28.5% of times remains unchanged. This explains why this variable is excluded for most of the attribute selection we computed.
For what concerns the combined attributes, the most frequent pattern always involve Light, Temperature and CO2, but now with some differences.



As we see, the passage between the two classes is quite evident, but we also must take into account that the support now is 28.5%.
In general, it seems that these transitions are more unstable than the other ones (each pattern found have less support), maybe because people enter the building in a sparse way.

Conclusions

The frequent sequential patterns confirm what has been found in the previous analysis. However, it's quite curious that difference in the characterization of the two transitions about which we can't do anything except making suppositions.
Another important aspect to consider is the velocity of the attribute variation from one class to another: if when the building is left the variables change quite slowly, the same cannot be said for the transition from class-0 to class-1. This could suggest that the start time is more rigid than the hour end of the working day.

Outliers and anomalies detection

Methodology

Because the data captures two distinct scenarios (empty vs. occupied room), it can be regarded as being generated from two very different distributions (see Disaggregated statistics). . Therefore, we separated the two classes and searched for the top 1% outliers within each class[19]. We used LOF, ABOD and KNN (average) with 20 neighbours.

Adopting tests of statistical significance[20], we compared each group of outliers against the population both in terms of attribute variations and of hourly distribution. The most semantically and statistically significant variations are discussed in the next section.

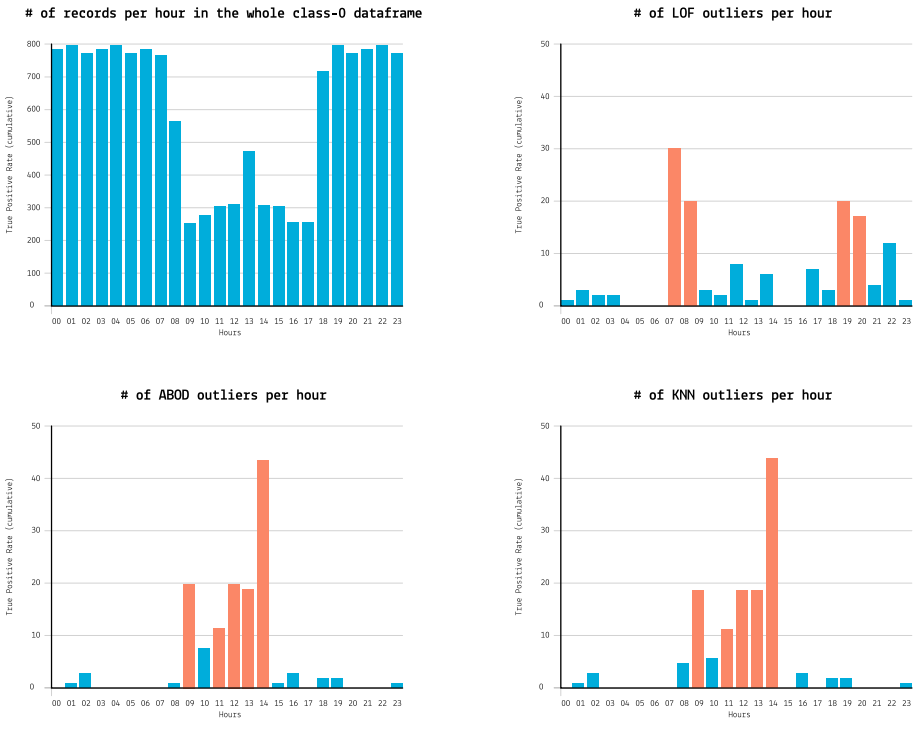
Outlier analysis

Class-0: cluster vs. population

Attribute variations were as follows (formula: outliers' mean - population mean).

	Temperature	Humidity	Light	Co2	HumidityRatio
LOF	0.26	-3.09	65.67	109.84	0.0004
ABOD	1.56	-1.63	354.12	263.93	0.00015
KNN	1.48	-1.68	412.78	237.43	0.00013
Attribute variations between outliers and population for class 0. The p-values were all smaller than 0.01 except for KNN-HumidityRatio (0.019)					

While the hourly composition is shown in the graphs below (hours for which the difference with the population achieved 5%-statistical significance are shown in red):



19. To be complete, we also searched for and analysed the top 1% outliers overall. The results were not only dubious for the aforementioned reason, but also difficult to interpret. For example, for ABOD's top outliers we obtained a Light mean of 435, as opposed to 121 in the whole dataset; however, the class composition was also starkly different (47-53 as opposed to 79-21 in the whole dataset), so that the average increase in Light could be either due to the higher percentage of class-1 records or to unusual values of the outlying class-0 records or, as an unstacked analysis suggested, to both.
20. We used t-tests to size attribute variations and z-tests to size variations in the hourly distribution.

From both points of view, it is apparent that KNN and ABOD find very similar outliers (the rate of overlap between the two is 76%), whilst LOF detects a different type of anomaly (74% of its outliers are exclusive to it).
All KNN and ABOD 0-outliers are detected at working hours. This explains the unusually high values for Temperature, Light and CO2: whilst there was no person in the room when the measurements were taken[21], either the hour of the day or the effects of recent human presence (or both) still affected these variables.
As to LOF, the time distribution shows that significant variations occur for transition hours (7-8 a.m. and 6-7 p.m.). This explains why attribute deviations from the population mean are not as steep as for KNN and ABOD, except for Humidity and HumidityRatio[22].
This clear-cut distinction between KNN and ABOD on the one hand and LOF on the other is certainly due to the fact that the former only take into consideration the position of a point's neighbours relative to the point itself, whereas the latter also considers the local density of each point's neighbour.

Class-1: cluster vs. population

For class 1, single attribute deviations from the population are not very informative: even when they are significantly different, they're too little to draw any sound conclusion from. However, this does not rule out the possibility that it is the combination of attribute values that is unusual.
As to the distribution in time, which we do not show for issues of space, the three groups have a good degree of overlap (measured at 65%), as opposed to what happened for class 1. KNN and ABOD still perform very similarly, with an 83% overlap degree. Overall, it would seem that class-1 outliers, much as their counterparts, tend to occur at hours at which people enter or leave the room (8-9 a.m., 1 p.m., 6 p.m.). The explanation, then, is akin to the one given in the previous section.

21. For which there might be various reasons, see Motifs and Anomalies for some speculation.
22. The significance for the differences in Humidity and HumidityRatio between LOF and KNN/ABOD outliers was measured with a t-test for samples with equal variances and is lower than 5%. Incidentally, that Humidity and HumidityRatio not vary in the same direction (see the table showing cluster vs population for class-0) is unusual of itself because of the high positive correlation between the two (see Basic statistics) and might help explain the outlying nature of these records

Conclusions

The dataset has proved to be suitable for variegate analysis. Due to the existing dependency between the attributes, methods like Naïve Bayes are less appropriate; on the other hands, we saw how with simple methods like KNN and Decision Trees (so even ignoring the continuous time variable) we can obtain appreciable results. Some of the complex methods like Ensemble bring the performance to the next level, while neural networks didn't make the difference in classification.
For what concerns the analysis of temporal data, we saw how a good forecasting needs very large amounts of data. However, studying the behaviour of the attributes in the temporal dimension gave us a deeper awareness about what takes place in the building.