Pat Junghenn
Thomas Keeley

Occupancy Detection:
Enhancing Energy Efficiency Through Machine Learning

Intro

Climate change is one of the heaviest debated topics in today's global discourse. The degree to which humankind contributes to the current trends through energy consumption and emission production is understood and discussed on widely varying levels. However, the constant drive for innovation and efficiency in any matter should be a common virtue for all. In terms of energy consumption and emissions, a large portion of the global share comes from buildings. Buildings, in particular non-residential, present a significant level of wasteful energy consumption which has led to stricter regulations and higher efficiency in recent years. At the base of this conservation effort is the understanding of building occupancy.

Occupancy detection is the active effort of preserving energy by varying the levels of energy consumption based on whether the space is occupied or not. There are multiple ways in which occupancy detection techniques are being applied to measure human presence. A few methods include camera images and video for presence detection, infra-red sensors to detect heat energy from the human body, and measurements of radio frequencies. This project, however, focuses on the measurement of indoor climate conditions. By taking samples of environmental conditions such as C02, temperature, humidity and light, patterns and behavior can be learned to enhance the efficiency of a buildings climate system by predicting human presence. This project examines the validity of this type of sensor data in modelling occupancy detection in non-residential buildings.

Data and Techniques

Data was acquired through the University of California-Irvine (UCI) Machine Learning Repository. This data is provided in three separate files comprising approximately 20,000 total samples over a 17-day period (02/02/2015 - 02/18/2015). Sensor readings were taken about once a minute to collect six potential features: temperature, humidity, humidity ratio, light, C02 and date/time. Additionally, an occupancy label is produced by collecting timestamped images

and annotating the human presence. The dataset is presented in a very clean form in that there is very little pre-processing that needs to be performed to be integrated into the modeling process.

This project will utilize a variety of modelling techniques to explore how well these sensor readings can learn from the labeled occupancy and ultimately be integrated into a building efficiency plan. First, the data will go through feature engineering to help enhance the robustness of the model. Additionally, a quick portion of exploratory data analysis (EDA) will be conducted to explore potential trends in the data and construct the overall curiosity surrounding this data. Next, the data will be compiled, split, and further processed for normalization. Lastly models will be produced and analyzed in multiple forms (Decision Tree, Random Forest, SVM, and Logistic Regression).

Feature Engineering and EDA

As mentioned before, this dataset did not require any major pre-processing. The three individual datasets were combined to achieve the desired split in preparation for model building. The date/time feature was engineered to produce a binary feature reflecting what was conceived to be typical work hours (0600-1800). This was done with the thought of potentially adding value to the models as well as an individual addition to the dataset.

Examining the pairplot of all features displays skewness in both light and C02. This was further validated by calculating the skewness to both be over 1. Additionally, there may be a risk of multicollinearity between humidity and humidity ratio as there appears to be a strong correlation between these two variables (Figure 1). There is an approximate 80/20 split for the target occupancy variable which is sufficient. All other features appear to be distributed well and not showing any concerning trends.

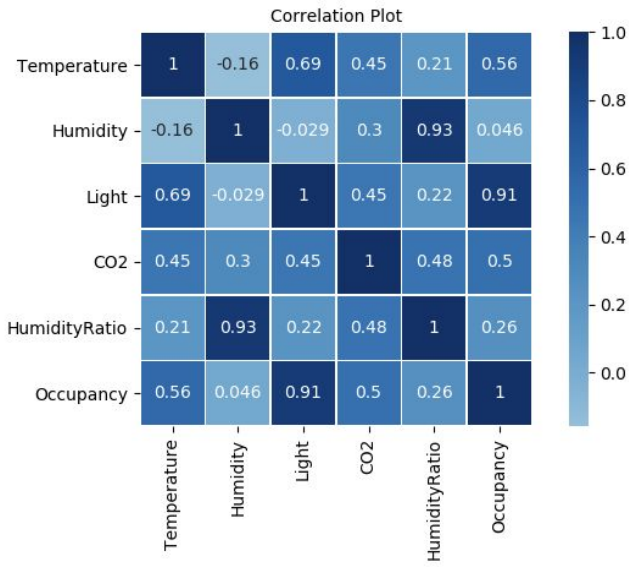

Figure 1

Pre-Processing

To help minimize the potential effect of skewness, the data is fed through a normalizer and scaler to bring the features to the same scale. This process improves the skewness factor for each feature except for temperature where it got slightly worse. As aforementioned, the features for the modelling process will be temperature, light, C02, humidity ratio, and work hours where the label will be occupancy. The data was then split into train/test datasets at a 70/30 split. These train and test datasets are used throughout the modelling process moving forward.

Modelling

Logistic Regression

The first model created for this analysis is a Logistic Regression algorithm. Logistic Regression will serve as a good starting point in the modelling process as it is a simpler approach to validating the features in predicting the occupancy label. Logistic Regression is appropriate for this data as it is designed to predict a categorical label. This algorithm aims to predict the probability of a label belonging to a particular class by plotting the dependent variable relative to each independent variable.

The overall accuracy of the Logistic Regression model in this study produced a near perfect score of 98.9%. This highly successful rate is a great start to the modelling process. When

examining the classification report and confusion matrix (Figure 2) there is a slight imbalance between precision and recall. The model performs quite well in correctly predicting positive occupancy but slightly overpredicts and displays a relatively high level of false positives labels. The imbalance in precision and recall comes at the expense of precision in this model.

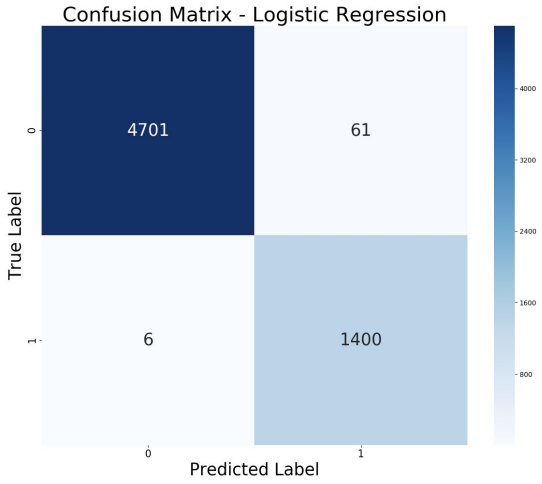

Figure 2

Decision Tree

The decision tree was built and optimized for attribute selection measures using Gini. Gini was chosen for this dataset because a majority of the variables are not binary split but continuous. The data appeared to be well distributed and would assume balanced probabilities. The overall accuracy of the model is tremendous at approximately 98.8%. The classification report (Figure 2) reveals that the model is slightly better at predicting non-occupancy over occupancy but the F-1 score disparity is only between 97% and 99%. The overall predictive power of this model is nearly perfect though the number of false negatives could be reduced to produce a more balanced result.

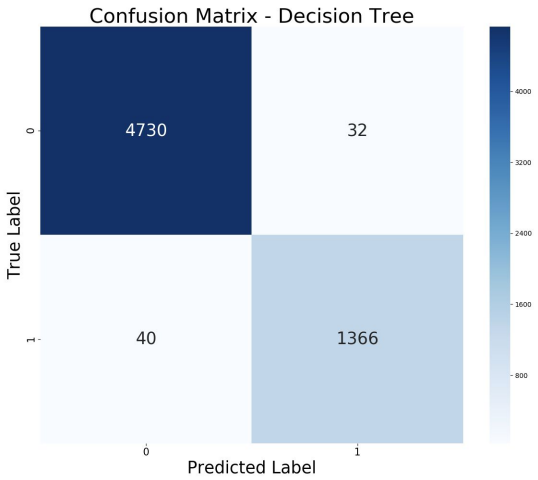

Figure 3

Based on the decision tree graphical output, the initial split feature in this model was light. This shouldn't come as too much of a surprise as many light features in buildings have another type of occupancy detection built into the system in the form of motion sensors. The next split was at temperature and the tree continues to split extensively from there. This is further explored when examining the variable importance in Figure 3. Light is overwhelmingly favored relative to the other variables. Using these results and graphical output, viewers can begin to understand the balances of these features in detecting occupancy and where these continuous features tend to be split in their values.

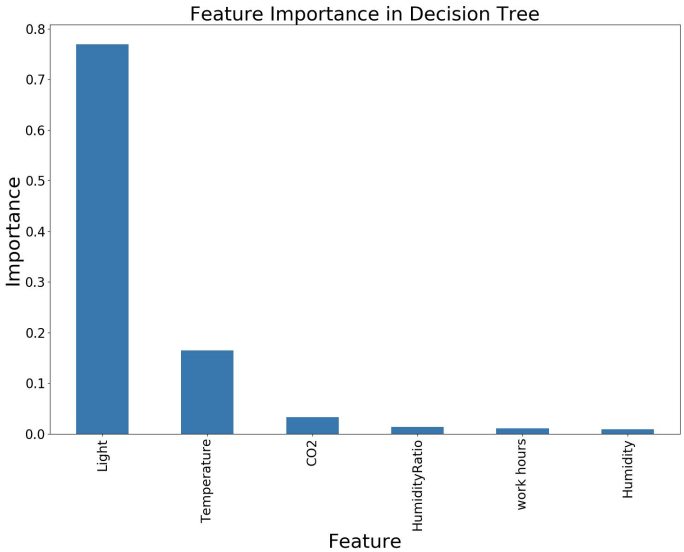

Figure 4

Random Forest

To build on the results of the Decision Tree model, a Random Forest Classifier was built with 100 estimators. The random forest algorithm is creating multiple decision tree models using random samples from the data. Some data points may be used multiple times in a process known as bootstrapping. The predictions are then made by averaging the predictions of each individual decision tree. The initial decision tree model in this project did not leave a lot of room for improvement but the Random Forest Classifier proved to enhance the results.

The results of the Random Forest model produced an overall accuracy rate of about 99.1%. Examining the classification report, the improvements were made in correctly predicting occupancy presence in the dataset. The Random Forest does not display a significant change in precision but does display an increase in recall. Though the model has the ability to decrease the false positives of the Decision Tree model, it slightly increases in false negatives (Figure 4). This will be the type of tradeoff that is considered within the context of the data and the implications that this would have on the specific label. In regard to occupancy, the decision may be made that the increase in awareness of positive occupancy may be worth the extra energy spent on falsely classifying a room as occupied.

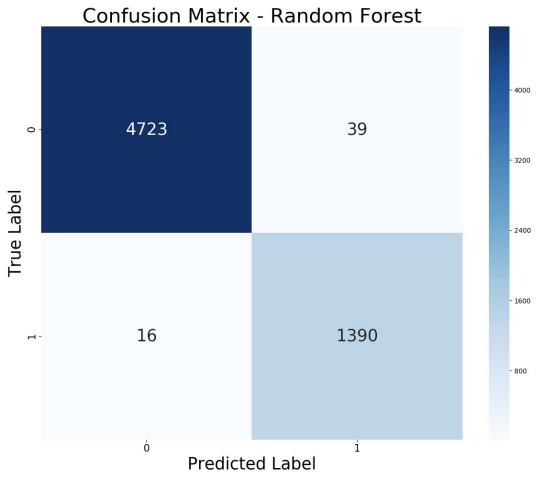

Figure 5

Support Vector Machine (SVM)

SVM is a slightly different approach to classifying labeled data in that it introduces a kernel function in defining the separability of the data. SVM discriminates the classifying variable by defining the optimal separating hyperplane. The SVM model was constructed using the polynomial kernel function. The polynomial function was chosen over the linear function as the data may not be highly linearly separable with the addition of the work hours feature. Additionally, the Gamma is set to 2 increase the model's ability to capture the complexity of the data. A lower Gamma value will constrain the model how deep the model will search for each training point's influence in constructing the hyperplane.

First, the overall accuracy of the SVM model comes out to be about 98.9%. Initially it appears the model performed quite well but when examining the classification report, there is a dip in precision, particularly positive occupancy labels. Though the high level of overall accuracy was maintained, the SVM severely underpredicts the non-occupancy levels with a very large increase in false negatives. The model was able to maintain the overall accuracy level by decreasing the false positive rate, however. Ideally there will be a balance between false positives and false negatives while maintaining predictive accuracy. The SVM model in this case is highly sensitive to predicting positive occupancy but less so in predicting non-occupancy labels.

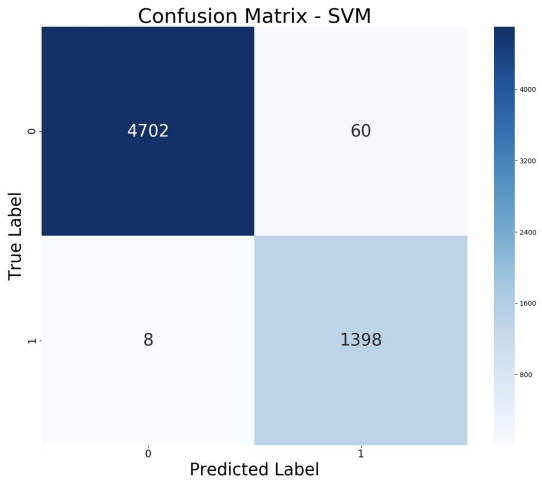

Figure 6

K Nearest Neighbors (KNN)

KNN is an algorithm that examines data points in the feature space and assumes that similar data are closer to each other in proximity. This algorithm takes the user defined number (k) of points that are closest in proximity and assumes the majority of those points with K being an odd number. This is a simple algorithm that can calculate distance using varying methods though euclidean distance is the most commonly used. The K-value requires the user to do a little bit of experimenting to find the optimal value. For this study, the optimal value for K was found to be five. The accuracy of this model returns the highest rate thus far of 99.1%. Additionally, there is a greater balance between precision and recall. There is greater sensitivity to positive labels though to this point, the KNN algorithm performs that greatest.

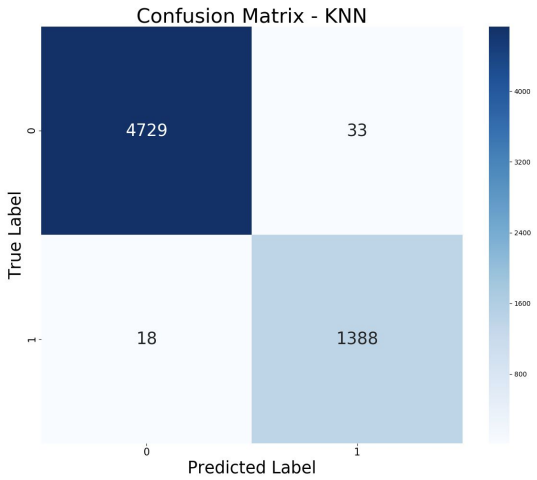

Figure 7

Ensemble

Ensemble learning is a method by which multiple models are trained and then combined to produce higher results. These individual models are often referred to as “weak learners” in that they perform better than a simple guess but may not be able to explain every unique behavior of a dataset. The main idea behind the ensemble method is using what is learned from each model to combine into a more accurate model. There is often a tradeoff for bias and variance within model development. By combining the features of each model, the complexity of the dataset may be better solved through the combination of these weak learners.

XGBoost

XGBoost is similar to the Decision Tree algorithm in that it is a collection of decisions that are considered weak learners. It is considered a gradient boosting method which uses the errors of one training set in training the new instances. XGBoost does not rely on making hard decisions for each node but rather assigns a positive or negative value to be used in averaging out the final decisions. XGBoost and similar algorithms require user controlled hyperparameter tuning to maximize the results of the model. For this, a grid search was conducted on the X and Y splits of the data to decide the best combination of max depth and max number of estimators. This method calculates the log loss for each parameter pairing and decides which provides the best results with the least amount of estimators. The ideal parameters were found to be a max depth of 12 and max number of estimators of 100. The results of the model return an accuracy of 99.1% and a slight dip in precision. This model performs very well with results similar to that of KNN. However, this method does not provide much improvement but will be considered along with all other methods.

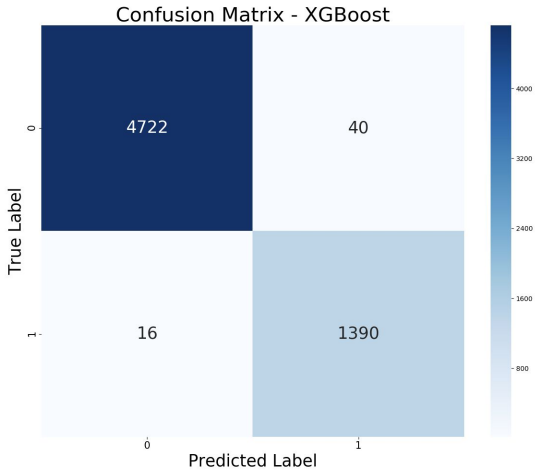

Figure 8

AdaBoost

AdaBoost is another ensemble method that runs a series of classifiers to maximize results. A base classifier is created and predictions are made. The misclassified data points are annotated and then used in the next classifier. This process is repeated until all of the classifiers are fitted to the model. The same hyperparameters that were derived from the grid search method are used for this classifier (max depth:12, max number of estimators:100). The results of the model

return the highest accuracy rates of all the previous models at 99.2%. This model also displays similar balance in recall and precision though the gap is slightly minimized while maintaining accuracy. AdaBoost is able to produce a Decision Tree algorithm that learns from the misclassification in previous instances. By learning from these prior errors, the data points are further modelled and classified to produce greater results.

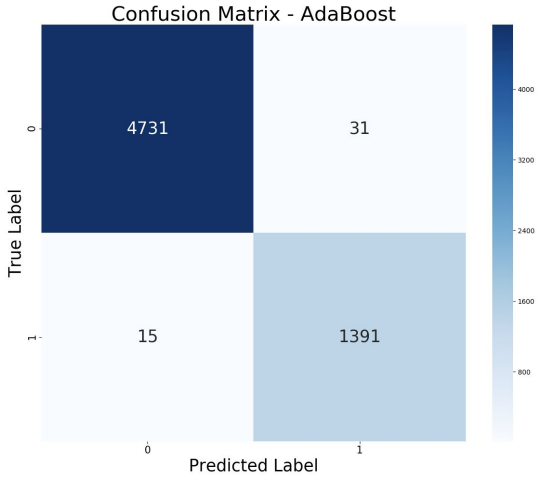

Figure 9

Blending

The next method of ensemble learning is known as blending. Blending combines the predictions from two more models. These predictions are made on a subset of the training dataset and then combined to form a single prediction model. The models will train on the same data and the predictions are then used as meta classifiers in the training data of a final model. In this case the final model will be a Logistic Regression. First the training data is split further to create a validation test set. Three new models (Decision Tree, Random Forest and SVM) are then trained on the validation set and test set. These models are then used as features in the Logistic Regression model.

This method does not greatly enhance the overall performance of our predictions as the accuracy is still lingering around 98.8%. There is a similar balance of precision and recall as seen in the SVM model. Where the false negative rate is minimized with this method, there still appears to be an imbalance in the model, though the predictive power remains. This blending method of ensemble learning will not be considered as an enhancement to the predictive power of occupancy detection.

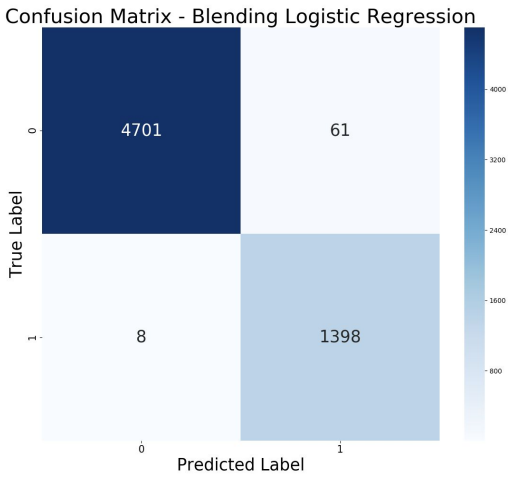

Figure 10

Hard Voting

The next method of ensemble learning used in this project is hard voting. This is simply the majority vote of a collection of models. For simplicity and assurance that a majority will be reached five models will be used: Logistic Regression, Decision Tree, Random Forest, SVM and KNN. Through this method, the estimators from each model are combined and then a majority vote of the classification is taken between the models. Based on this majority vote, the predictions are run again. The results display an increase in the overall accuracy to just over 99.1%. This method still displays a higher proportion of false positive rate relative to false negative but a closer balance between the two is achieved. Though this method did not greatly enhance the model's performance, it did slightly improve the overall results and bring a greater balance between precision and recall (Figure 7) compared to the blending ensemble.

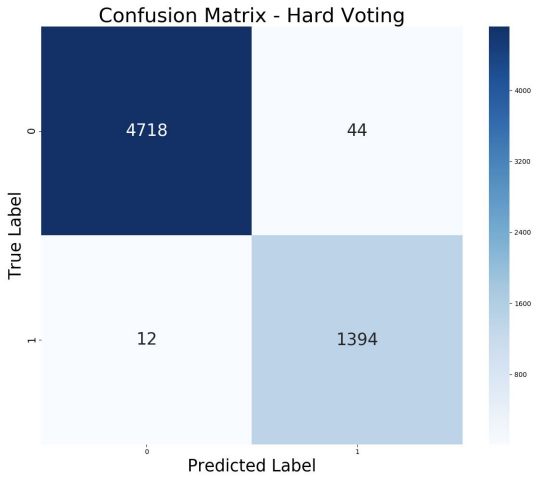

Figure 11

Conclusion

It is clear that there is high predictive power in these environmental sensor variables. The performance of these models is so high that multiple methods can be employed to achieve near perfect results. The impact of such modeling can be seen in the development of energy efficient systems that have the ability to constantly monitor the climate variables of a space to make adjustments to the energy output of climate controlling systems. Unsurprisingly, light readings were the highest contributing factor in predicting occupancy as it is a variable that is shaped and often controlled by humans.

The more interesting contributions come from variables that humans do not consciously contribute to such as C02 levels and humidity. Though when compared to light, these variables may seem to not have an impact on overall model performance. When taken out of the model, the overall accuracy of the models slightly decreased, and the balance of precision and recall was slightly skewed. The feature engineering of creating the work hours variable also proved to be slightly beneficial, especially in the decision tree model.

Overall, this dataset did not require an immense amount of pre-processing to produce actionable results. The models used in the study by themselves present quite respectable predictive power. The addition of ensemble learning did not appear to achieve our goal of boosting the already favorable results by a significant margin. However, in the end, the results of the AdaBoost ensemble may be held to the highest degree with the greatest overall accuracy.

This study proves the concept of using environment sensor readings as a proxy to human occupancy and thus may have a significant impact on increasing energy efficiency and subsequently limiting harmful emissions.

Logistic Regression		**0**	**1**		
	0	4701	61	**Accuracy**	0.989
	1	6	1400		
Decision Tree		**0**	**1**		
	0	4727	35	**Accuracy**	0.988
	1	40	1366		
Random Forest		**0**	**1**		
	0	4722	40	**Accuracy**	0.991
	1	15	1391		
SVM		**0**	**1**		
	0	4702	60	**Accuracy**	0.989
	1	8	1398		
KNN		**0**	**1**		
	0	4727	35	**Accuracy**	0.991
	1	19	1387		
XGBoost		**0**	**1**		
	0	4722	40	**Accuracy**	0.991
	1	16	1390		
AdaBoost		**0**	**1**		
	0	4731	31	**Accuracy**	0.993
	1	15	1391		
Blending		**0**	**1**		
	0	4701	59	**Accuracy**	0.989
	1	6	1400		
Hard Voting		**0**	**1**		
	0	4714	48	**Accuracy**	0.990
	1	11	1395		

Figure 12