A comparison of machine learning models for occupancy detection in buildings

Alexander Marinov, Zeppelin Vanbarriger

***Abstract*—Applications of machine learning include occupancy detection of buildings. Machine learning models such as Random Forests, Support Vector Machines, and Logistic Regression are trained and compared on environmental data. The best performing model was a Logistic Regression model which trained on Light and Temperature data, yielding** 99.29% **on the test data.**

I. Introduction

Occupancy detection using building environmental data can prove fruitful for the reduction in energy consumption. Humidity, luminescence, and CO2 sensors provide real-time data which HVAC and lighting control systems can use to determine occupancy. This necessitates the development of predictive classifiers which use said environmental data to determine occupancy.

This group investigated the efficacy of various predictive classifiers, namely Random Forest Classifiers (RF), Support Vector Machines (SVM), and Logistic Regression. Such classification models are readily available with the programming language Python and its libraries. This work uses datasets, created in [1], consisting of humidity, luminescence, and C02 sensor readings to train and test RF, SVM, and Logistic Regression models for efficacy.

A. Literature Review

The work by Candanedo et al served as a guide for our analysis. They goal of their paper was to compare models whose efficacy had not been explored - namely Random Forest (RF), Gradient Boosting Machines (GBM), Linear Discriminant Analysis (LDA) and Classification and Regression Trees (CART). The datasets which they tested these models on were generated by the authors. Various combinations of features (Light, CO2, Temperature, Humidity, Date) were tested for each model. The best results were reported with CART on Light, Temperature, and date information, yielding 99.32% accuracy.

II. Exploratory Analysis

Before deciding on which models to use for this predictive task, we performed exploratory analysis on the data for the purpose of feature engineering and selection. We began by transforming the date parameter from the dataset into a categorical binary encoding representing work hours during weekdays with a 1 and non-work hours and weekends with a 0, since we believe that is the most important information from that feature for the task of predicting occupancy in an office. We then computed the correlation matrix for the dataset, which is summarized with color coding in Figure 1.

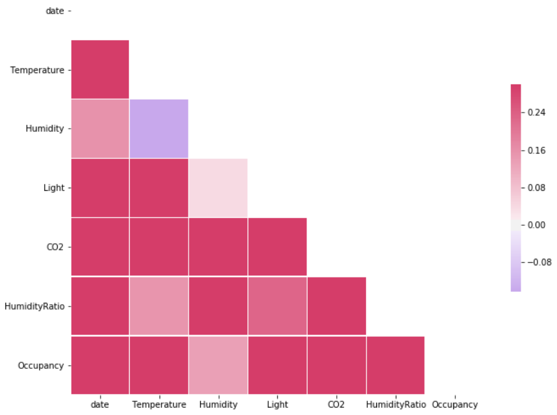

Fig. 1.

From this initial analysis we noticed that all features besides humidity are somewhat correlated to the occupancy, however, for more careful feature selection we plotted the pair plots between all variables for further analysis. The plots can be seen in Figure 2. We noticed a significant similarity between the distribution of occupancy and the distributions of date, light, and CO2, and thus we decided to experiment with combinations of these parameters when training our model.

III. Methodology

*A. **Random Forest (RF) Classifier (Baseline)***

Random Forests are an ensemble Decision Tree model which use bagging to generate many decision trees for classification. Each tree is grown by sampling (with replacement between tree generation) the same number of data variables and then randomly selecting a small subset of available features to determine the split. Generally, these trees are not pruned. During classification, each tree in the forest ”votes” and then these votes are aggregated.

RF are robust to over fitting, especially compared to traditional decision tress. They work well with large amounts of variables and are especially suited to multi-class

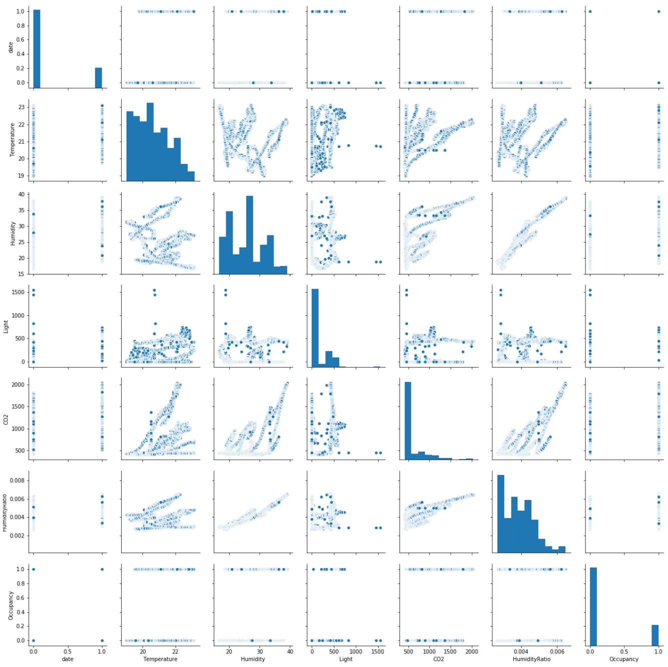

Fig. 2.

classification with unbalanced classes. Additionally, RF can illuminate correlation between features. Interestingly, the problem this group is investigating generally does not fit this criteria which suggests that this particular model is not suitable occupancy detection. Thus, this group includes RF only as a baseline model for comparison against other models, due to its inclusion in [1].

The model was trained on the data in *train.txt*. Hyperparameter selection was done on the validation set in *validation.txt*, where different selections of features (Light, Temperature, C02, Date) were compared against different forest sizes (100, 200, 500, 1000, 10000). Additionally, performance between feature selection methods in tree generation were compared ($\sqrt{p}$ vs $log_2(p)$). Outcome of these experiments are reported in the Results section.

B. *Support Vector Machine (SVM) Classifier*

Support Vector Machines (SVM) are a non-probabilistic binary linear classifier which learn the best possible maximal separation between data points. By learning the best possible class separation line, the model can accurately classify new data. Despite being inherently linear classifiers, SVMs can generalize to non-linear classification using kernel transforms which map the input data into a higher dimensional feature space in which the data is linearly separable.

Since our prediction task involves predicting a binary outcome depending on whether the office is occupied or not, we decided that the SVM model would be an appropriate candidate for a model. For SVMs, besides the features themselves, the most important hyperparameters to consider are the kernel and regularization constant. We chose to use an SVM with a linear kernel and a hinge loss function in case the data is not linearly separable, and experimented with various regularization constants to control the sensitivity of the learned margin.

We proceeded to train and validate the model experimenting with different combinations of parameters between Light, Temperature, Date and CO2 levels, and four different regularization constants of 1, 10, 100 and 1000. The model was trained on the data in *train.txt* and validated on the data in *validation.txt*. The final score obtained in the Results section was calculated on the *test.txt* portion of the data with the hyperparameters that obtained the best score on the validation set.

C. *Logistic Regression Classifier*

Logistic Regression is a probabilistic binary linear classifier which, when given some input data, yields a probability that that data belongs to a certain class. Results with a probability above some cutoff are then given a positive classification. The rest are given a negative result. This model is trained by evaluating an error function, which in the case of binary classification is Cross-Entropy Loss (or Log Loss). Gradient descent is then used to minimize this error function.

As in the SVM model, Logistic Regression was a natural choice for our occupancy detection problem due to its nature as a binary classifier. Such models are generally easy to train, tune, and evaluate. This type of model does well with multiple predictors and where there are at least 10 samples per possible feature value. These requirements are well met with our data set.

The logistic regression model was trained on the data in *train.txt* and validated on the data in *validation.txt*. Similar feature combinations were tested as in the above models, as well as regularization constants of 1, 10, 100, 1000. A cutoff value of 0.5 was used to determine classification, as it is implemented in the $sklearn$ Python library. The training and testing results are reported in the Results section.

IV. Results

A. *Random Forest (Baseline)*

The Random Forest model was trained on four of the feature configurations found in [1] as a comparison. Additionally, 5 different forest sizes were tested (100, 200, 500, 1000, 10000) as well as number of features p to consider for each cutoff ($\sqrt{p}$ vs $log_2(p)$). The best accuracy result for a given configuration (features used and forest size) is given in bold. The test set was evaluated with a model which used only Light as the input feature and ran for a forest of size 1000.

We found that the Random Forest Classifier performed best with just Light as the most optimal feature, with forest sizes of 200 and above performing the same. Additionally, choosing the method for feature selection (using $log2p$ instead of $sqrtp$) resulted in worse performance.

TABLE I
Parameters: Light

Forest Size	Accuracy
100	95.53%
200	**95.68%**
500	**95.68%**
1000	**95.68%**
10000	**95.68%**

TABLE II
Parameters: Light, Temperature

Forest Size	Accuracy
100	93.62%
200	93.54%
500	94.71%
1000	95.46%
10000	94.82%

TABLE III
Parameters: Date, CO2

Forest Size	Accuracy
100	85.63%
200	86.00%
500	85.85%
1000	85.85%
10000	85.89%

TABLE IV
Parameters: Light, Temperature, Date, CO2

Forest Size	Accuracy
100	94.59%
200	95.27%
500	95.12%
1000	95.04%
10000	95.27%

TABLE V
Parameters: Light, Temperature, Date, CO2 w/ Log2

Forest Size	Accuracy
100	94.30%
200	94.67%
500	95.20%
1000	95.31%
10000	95.31%

B. *Support Vector Machine (SVM) Classifier*

The support vector machine model was ran on four different configurations of the parameters and with four different regularization constants as described in the methodology section. The results of these are outlined below with the best accuracy on the validation set in bold. The final accuracy on the test set is in the final results table at the bottom of the section.

We found the light and temperature parameters to produce the best results out of all the parameter configurations. An interesting result was that the date and CO2 parameters did not perform as well as the others as the model performed significantly worse when ran on just those two. The model performance also performed worse when considering all four parameters versus just the light and temperature.

TABLE VI
Parameters: Light

Regularization C	Accuracy
1	97.86%
10	63.64%
100	97.90%
1000	97.86%

TABLE VII
Parameters: Light, Temperature

Regularization C	Accuracy
1	71.63%
10	75.75%
100	**97.90%**
1000	97.86%

TABLE VIII
Parameters: Date, CO2

Regularization C	Accuracy
1	84.16%
10	85.48%
100	84.28%
1000	83.53%

TABLE IX
Parameters: Light, Temperature, Date, CO2

Regularization C	Accuracy
1	97.82%
10	97.86%
100	97.82%
1000	97.82%

C. *Logistic Regression model*

The initial runs on the logistic regression model are outlined below. The model was run on the same configurations of parameters as the SVM with a regularization constant of C=1. Results obtained from different regularization constants are omitted as we didn't discover a significant difference in the results.

TABLE X
Logistic regression with C=1

Parameters	Accuracy
Light	97.86%
Light, Temperature	**97.86%**
Date, CO2	91.56%
Light, Temperature, Date, CO2	97.82%

The final accuracy the models scored on the test set with the best hyperparameters selected from the validation set are outlined below. The best accuracy achieved is bolded.

TABLE XI
FINAL RESULTS ON TEST SET

RF (Baseline)	97.91%
SVM	98.54%
Logistic Regression	**99.29%**

V. Discussion

As reported in the Results section, Logistic Regression preformed most optimally, achieving $> 99\%$ accuracy on the testing set. However, the other methods- RF and SVM-performed almost as well, achieving 97.91% and 98.54% on the testing set, respectively. This suggests that occupancy detection using the given train and test data can be modeled very well with the given data. There are a number of reasons that this could be the case.

One such possible reason that was noticed during exploratory analysis and reflected in our hyperparameter tuning was the usefulness of Light as a predictor. Simply put, an office with the lights on is likely to be occupied. The predictive power of this feature was seen in hyperparameter tuning, where Light performed better than or almost equal to, across all models, Light combined with Temperature. Another possible cause of the high performance of our models is the dataset itself. This dataset was generated in a controlled fashion by the authors of [1], which could indicate some hidden bias. Using data collected from a non-controlled site could provide a more realistic training and testing set.

Since our models performed very well on the test set, there is little that we would change about our investigation. Logistic Regression and SVM models seem to be a natural choice for our problem due to their predictive power in binary classification.

Thus our investigation of this problem concludes that occupancy detection using features such as luminosity, temperature, CO2 levels, and humidity levels is feasible. More testing is required to fine tune real world models, most likely using real world data.

References

[1] Luis M. Candanedo, Vronique Feldheim. *Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models.* https://www.sciencedirect.com/science/article/pii/S0378778815304357.