Principal component Analysis in occupancy prediction

Omar Bouhamed[1]

Student ID: 40186319

[1]Concordia University, Montreal, QC, Canada, omar.bouhamed@mail.concordia.ca

***Abstract*—Over the last few years, we have truly entered the era of artificial intelligence and in particular machine learning in smart building applications in general and occupancy prediction in particular. Accurate and timely occupancy prediction has the potential to improve the efficiency of energy management systems in smart buildings In this context, we have applied Principal Component Analysis on data collected from an office room's sensors to get some statistical insights about the occupants' behaviors, and therefore accurately predict their presence. A study was carried out to investigate the performance of a well-known machine learning model, namely, support vector machines (SVM) to predict occupancy.**

***Index Terms*—PCA, Classification, Occupancy prediction, Support-vector machines.**

I. Introduction

According to recent statistics [1], buildings account for some 40% of the total energy usage in the world. Occupants' behavior influences significantly that energy usage. In order to reduce energy consumption, using artificial intelligence and smart equipment, it is possible to control Heating, ventilation, and air conditioning (HVAC) and lighting devices remotely. Recent research works have shown that it is possible to save HVAC energy, by automating its control, in a given building via occupancy detection (i.e. detecting the presence or absence of occupants inside the building), or prediction, When provided accurate occupancy models, demand-driven control can utilize such information to coordinate real-time HVAC usage, reducing energy use and maintaining indoor thermal comfort in buildings.

To this end, in this project, a study was conducted to accurately predict the occupancy in an office room using data from light, temperature, humidity, and CO2 sensors [2]. The goal of this paper is to analyze the changes captured by the sensors, try to predict the presence of occupants, and subsequently control the HVAC systems accordingly.

The rest of the paper is organized as follows: Section II presents the used dataset. Section III applies the concept of PCA and the steps followed to proceed with the data. Section IV discusses the classification problem and the results of the different conducted experiments. Finally, a conclusion is drawn in Section V.

II. Dimensionality reduction

Nowadays, the world's most valuable resource is no longer oil, but data. With the explosive growth in the use of smart devices, such as phones and sensors, a huge amount of data became available that opened doors to new horizons, businesses. However, the availability of such sophisticated data resulted in the appearance of new challenges, namely, data complexity. Despite the efforts made to upgrade the computational resources, data complexity was still a major concern for scientists. A huge amount of high-dimensional input (features) can significantly increase processing complexity and compromise the performance of a machine learning algorithm. However, many features may present the same information, i.e redundancy, which can negatively impact the performance of the machine learning algorithms. Therefore, to resolve this issue, various solutions were proposed such as feature selection and dimensionality-reduction techniques. Both concepts are often grouped together, however, while both methods are used to reduce the number of features, their theories are quite different. The feature selection is simply selecting and excluding features without changing them, as for the dimensionality reduction, it transforms features into a lower dimension. Over the years, numerous dimensionality reduction technique was proposed like Linear discriminant analysis (LDA), Neural autoencoder, and Principal component analysis (PCA) [3].

In this study, we will focus on PCA, the bedrock dimensionality-reduction technique that is often used to transform large data sets into smaller ones while keeping the most relevant information. Obviously, reducing the number of variables in a data set reduces precision, but the trick of dimensionality reduction is to substitute a little accuracy for simplicity. Since smaller data sets are easier to explore and visualize, without the need to deal with extraneous variables, machine learning algorithms can become faster and lighter.

III. Principal component analysis

In this section, a brief description of the used dataset is discussed, and then a detailed presentation of PCA implementation is presented.

A. Dataset description

In this paper, the used data set was found in the Machine Learning Repository of the University of California, Irvine.

The data set provides experimental data used for binary classification (room occupancy of an office room) from Temperature, Humidity, Light, and CO2. The data contains 8144 instances composed of 5 features along with the date and 1 class variable (occupied or not), which can be summarized as follows [2]:

- **Temperature**: Temperature in degree Celcius.
- **Humidity**: Relative humidity in percentage.
- **Light**: Illuminance measurement in unit Lux.
- **CO2**: CO2 in parts per million (ppm).
- **HumidityRatio**: Derived quantity from temperature and relative humidity, in kgwater-vapor/kg-air.
- **Occupancy**: Occupied or not, 1 for occupied and 0 for not occupied.

B. Dataset Exploration

First, in order to figure out if our data is balanced a count plot was drawn in Fig. 1. Based on this plot, most of the data samples were counted as not occupied; around 75% not occupied and the rest as occupied, hence, the data is not balanced.

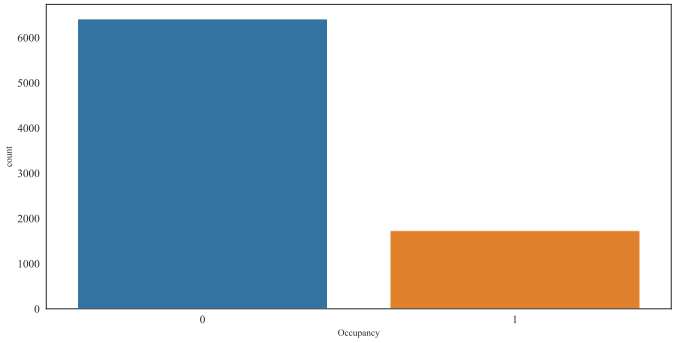

Fig. 1: Count plot.

Since features had different measurement values, the data were normalized to avoid biased decisions in the PCA method and the classification phase. Fig. 2, the box plots show the variation in the measurement of the sensor for both classes. it is noticeable that all the factors increase when existence is detected. This could be also noticed in Fig. 3, the pair plot shows a high elevation in the blue diagrams, occupancy, compared to the orange ones, with no occupancy.

The covariance matrix, shown in Fig. 4, demonstrates that there is a strong correlation between features, especially humidity and humidityRatio features, which is understandable since the latter is just another representation of the former. With such high covariance, a multicollinearity problem may arise. Therefore, to address this problem, we used PCA to obtain new uncorrelated features.

C. Principal component Analysis

PCA is the procedure used, primarily for dimensionality reduction of a given dataset. It is one of the most effective and

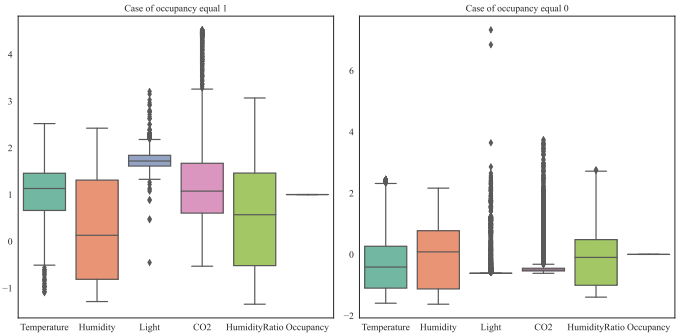

Fig. 2: Box plots of normalized data.

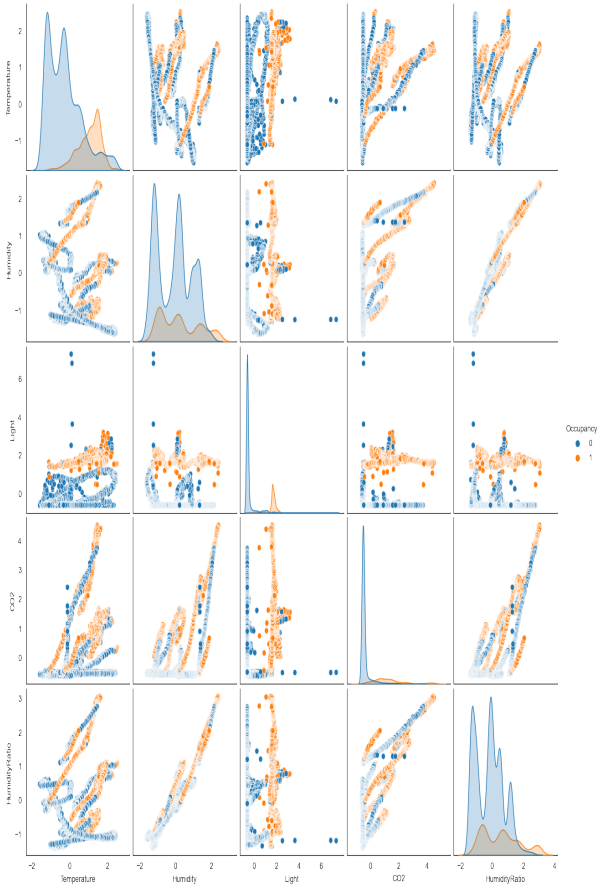

Fig. 3: pair plots.

reliable methods for reducing data dimensions [4]. Ultimately, PCA goal is to reduce the dimensions of a n-dimensional dataset by projecting it into a r-dimensional subspace (where r $\leq$ n) to maximize computational efficiency while preserving the majority of the information. PCA transforms strongly correlated variables into new uncorrelated ones (principal components), i.e dropping duplicate and redundant information.

Let $\mathcal{X}$ be the data matrix formed by the occupancy dataset with a size $n \times p$. the PCA algorithm consists of four main

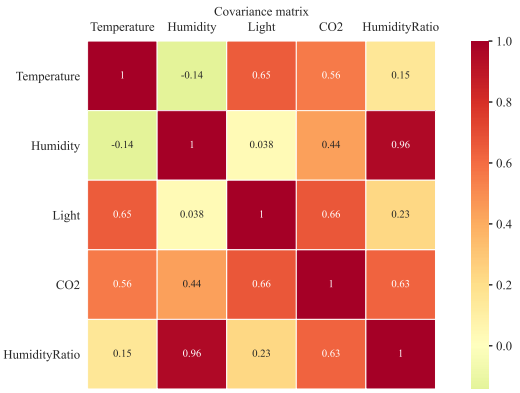

Fig. 4: Covariance matrix.

steps [5]:

- Step 1: Standardization. The objective of this step is to make the features (data Columns) contribute equally in the analysis. standardization is made by calculating the centered data matrix $\mathcal{Y} = H\mathcal{X}$ by subtracting off-column means.
- Step 2: Computation of the covariance matrix . The objective of this step is to identify any relationship between the features. The $p \times p$ covariance matrix S of the centered data matrix is computed as follows:

$$S = \frac{1}{n-1} Y^{\intercal} Y$$

As shown in Fig. 4, S serves as a perfect way to identify the correlated features and it is noticeable that our data variables are strongly correlated. Hence, the employment of the principal component analysis is suitable for this situation.

- Step 3: Computation of the eigenvalues and eigenvectors of the covariance matrix to identify the principal components. The eigenvectors are the new representation of our data, precisely a linear combination of the initial features. Each eigenvector is associated with an eigenvalue which can be interpreted as the importance or the weight of the vector (the higher the eigenvalue the more the vector contains information about the initial data). The eigenvectors and eigenvalues of S are computed using eigendecomposition:

$$S = A\Lambda A^{\intercal} = \sum_{i=1}^{p} \lambda_i a_i^{\intercal} a_i.$$

The eigenvectors of the occupancy dataset are given as follows:

$$A = \begin{pmatrix} 0.34 & 0.53 & -0.71 & 0.22 & -0.18 \\ 0.39 & -0.57 & 0 & 0.22 & -0.67 \\ 0.41 & 0.44 & 0.66 & 0.43 & 0 \\ 0.55 & 0.12 & 0.11 & -0.81 & 0 \\ 0.50 & -0.41 & -0.18 & 0.20 & 0.70 \end{pmatrix}$$

The following vector indicates eigenvalues:

$$\lambda = \begin{pmatrix} 2.73 \\ 1.69 \\ 3.48 \\ 2.14 \\ 8.09 \end{pmatrix}$$

- Step 4: Computation of the transformed data matrix $\mathcal{Z} = \mathcal{Y}A$, where $\mathcal{Z}$ is a matrix of size $n \times p$, its columns correspond to the principal components (PCs) scores, and the variances of its columns are the eigenvalues. The first and second principal components are computed as follows:
  1) $Z_1 = 0.34X_1 + 0.39X_2 + 0.41X_3 + 0.55X_4 + 0.50X_5$.
  2) $Z_2 = 0.53X_1 - 0.57X_2 + 0.44X_3 - 0.41X_5$.

In order to verify that the data was successfully into uncorrelated ones, the covariance matrix of the new components was drawn in Fig 5.

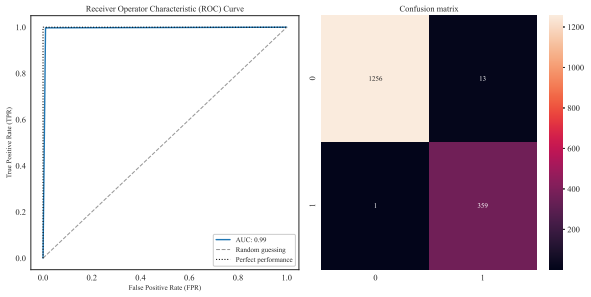

Fig. 5: PCA covariance matrix.

The two-dimensional scatter plot of PC1 and PC2 score is given in Fig. 6. This plot is a graphical display of observations and could be a very good approximation to the original scatter plot in five-dimensional space [6]. Fig. 7 is a scatter plot of PC1 coefficient as a function of PC2 coefficient. This plot is useful to interpret the information carried out by the PCs. For instance, as shown in Fig. 7, X_4 (Co2) has the most contribution within PC1, followed by X_5 (HumidityRatio). As for PC2, X_2 (Humidity) has the most contribution along with X_1 (Temperature).

Now that the data columns were transformed into new uncorrelated ones, it is time to decide which PCs to keep and which to discard. PCA always tries to put the maximum possible information in the first component, then the maximum remaining info into the second, and so on. So, the final task is to determine which components to keep, to reduce the data matrix size from $n \times p$ to $n \times r$, where r is the number of kept components. There are various methods to determine r,

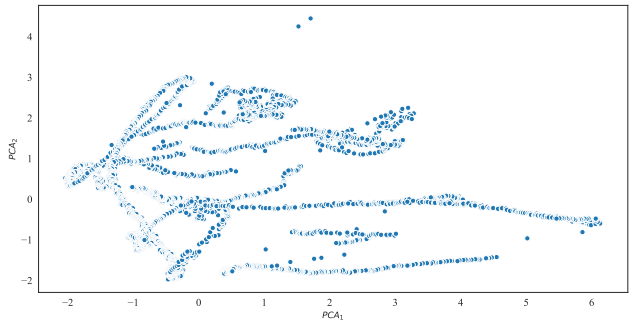

Fig. 6: Scatter plot of PC1 vs. PC2 score.

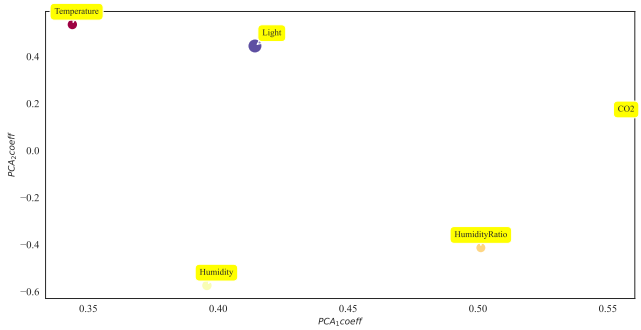

Fig. 7: Scatter plot of PC1 vs. PC2 coefficients.

but the most known one is the scree plot and the Pareto plot shown in Fig 8 and Fig 9, respectively.

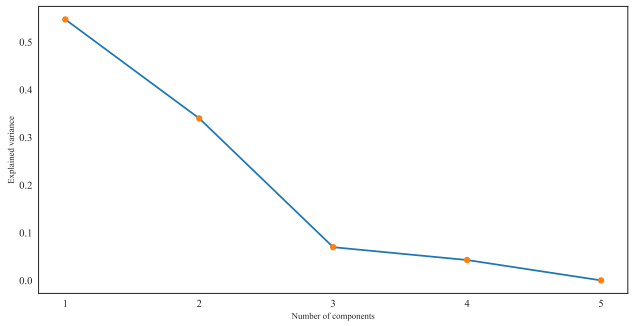

Fig. 8: Scree Plot.

The explained variance, shown in Fig 8, were computed by dividing each PC eigenvalue by the sum of eigenvalues, and it is given by:

$$l_i = \frac{\lambda_i}{\sum_{i=1}^{p} \lambda_i} 100\%, \forall j \in [1, p]$$

The rule is to retain a number of components that represents at least 80% of the data variance. As shown in Fig 9, more than 80% of the total variance can be explained by the first 2

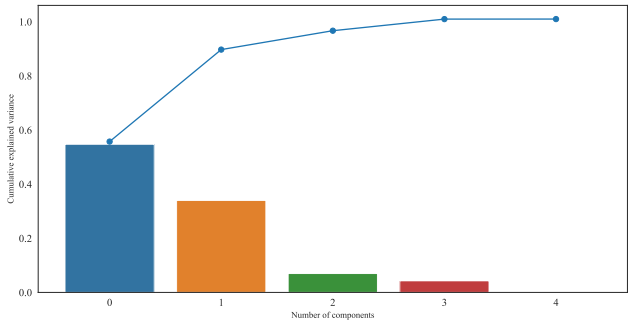

Fig. 9: Pareto plot.

PCs ($l_1 + l_2 = 54.73\% + 33.98\% = 88.71\%$). Hence, $r = 2$. In other words, the dataset can be reduced to 2-dimensions.

The Biplot, shown in Fig 10, displays how strongly each characteristic influences a principal component. The feature vectors are pinned at the origin of Pcs (PC1 = 0 and PC2 = 0) and their project values show how much they contribute on the respective PC [7]. It is noticeable that all the vectors contribute positively to the first PC. That corresponds to the vectors are directed to the right half of the plot. As for the second principal component, it has 3 positive coefficients and 2 negative coefficients. Another information can be extracted using this plot; Based on the angles between the vectors, the correlation between the features can be noticed:

- the closer two vectors are small angles, the more positively correlated are. For example Temperature and Light.
- In case the vectors are orthogonal, the two features are likely not correlated. For example Temperature and Humidity.

Each of the 8144 samples is represented in this plot by a point. The color of the data indicates whether it is occupied or not. As shown in Fig 10, the points with the lowest PC2 scores correspond to the unoccupied samples. We can see that the occupied class extends far to the top-right part following the vectors CO2, light, and Temperatures which shows that the increase in these factors plays a major role in concluding the presence of humans in the room/office.

IV. Classification

In this section, Support vector machine (SVM), a classification algorithm will be applied on the first two components, then on the original data [8]. Afterward, a comparative analysis with other known machine learning techniques will be carried out to validate the performance of SVM.

SVM hyper-parameters can be so tricky and highly dependable on the application and data to be tuned. In some cases, the two classes can be separated by simply drawing a straight line, also known as a decision boundary. Unfortunately, this does not work for most cases. Therefore, a pattern analysis approach, called the kernel method, comes into place. A widely used method in the SVM model to bridge linearity and

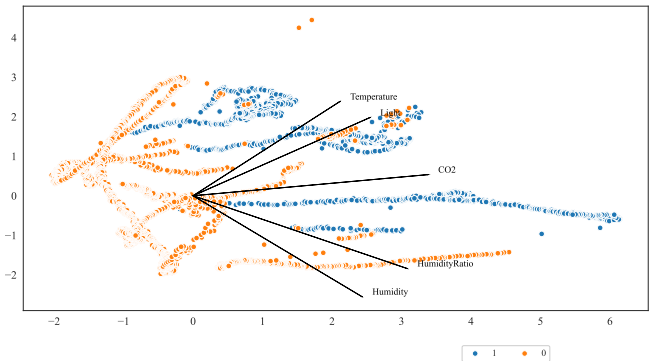

Fig. 10: 2D Biplot.

non-linearity. Simply explained, when the data is inseparable in the current dimension, another dimension is added. For instance, the data is mopped from 2d to 3d to find a gap between the different classes.

To better perceive the impact of the hyper-parameters on the performance of our classifier, we trained an SVM model using random parameters at first, and then the Grid-search approach was used to find the optimal hyper-parameters. For the sake of better visualization, we used the first two components as input for our model. The experiment results are summarized in tab I. Clearly, the tuned SVM had a better result compared to the model with random parameters. The tuned SVM outperformed the random one in terms of accuracy, precision, and recall. Fig 11 and Fig 12 present the decision boundaries for both models. As expected, in contrast with the decision boundary for the untuned model, the decision boundary in Fig 12 wraps most of the class 1 points.

TABLE I: Experiment 1 results.

ML-classifier	Accuracy (%)	Precision	Recall
Random Par SVM	98.28	0.9586	0.9639
Tuned SVM	99.69	0.9863	1

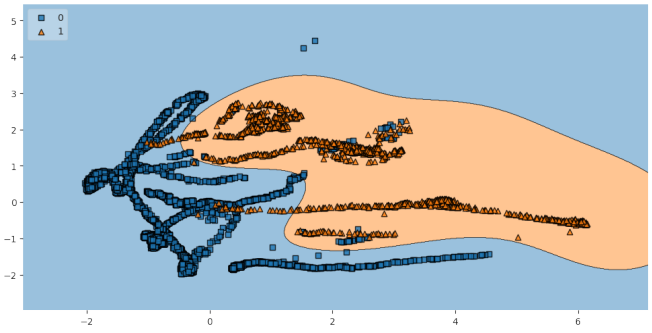

Fig. 11: Decision boundary for untuned SVM

The next step is to compare the performance of the tuned SVM model having the original data and just the first two components as input. To do so, we draw the roc curve and

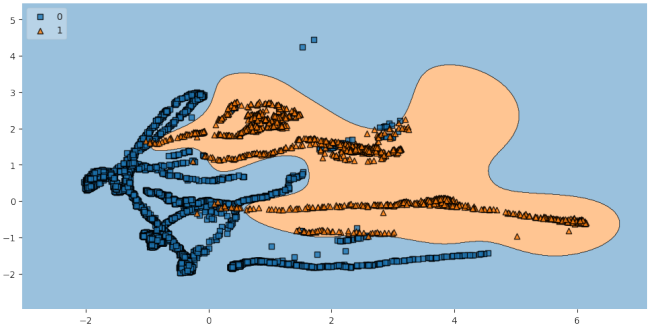

Fig. 12: Decision boundary for tuned SVM

the confusion matrix for both experiments in Fig 13 and Fig 14. As reflected in the confusion matrix, using just two-component, the performance of the model decreases. Using the original data the model made a total of 5 mistakes (false positive), however, while using just the two components, the number of mispredictions increased reaching a total of 28 mistakes (15 false positives + 13 false negatives). As expected, the SVM using the original data outperforms the SVM having just two columns. After all, PCA scarifies accuracy to win in terms of fitting time, a trade-off between computational efficiency and high accuracy. Unfortunately, for the case of a small dataset, like the one we are treating, the benefit brought in terms of complexity reduction is not that important, however, if we were using a dataset with thousands of samples, the difference would be more appealing.

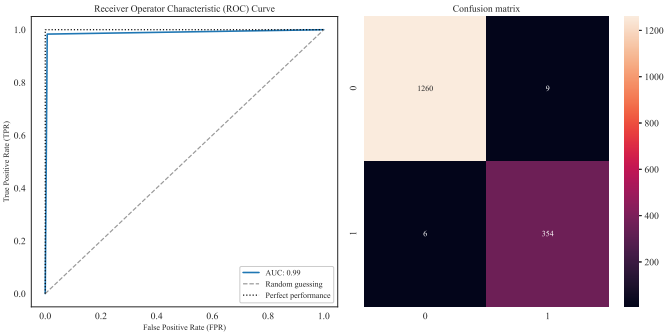

Fig. 13: Roc curve and confusion matrix using two components

With the purpose of evaluating the performance of the SVM algorithm, a comparison with known algorithms and methodologies used for classification problems was carried out (using the original data). The performances of the selected algorithms are presented in Table II. We can observe how the SVM model produces good results, if not the best (taking into consideration the accuracy, F1-score, and recall). The SVM model stands out in the Recall metric (in green). This metric is very crucial to the occupancy prediction since it reflects the capability of differentiating between occupied and unoccupied activities (minimum number of false negatives).

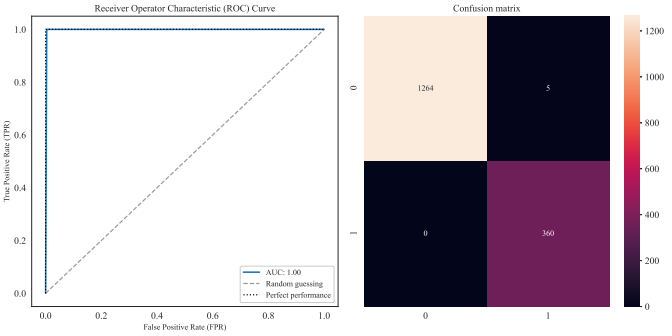

Fig. 14: Roc curve and confusion matrix using original data

TABLE II: SVM performance vs. different ML-techniques.

ML-classifier	Accuracy (%)	Precision	Recall
Logistic Regression	98.65	0.9643	0.9750
ANN	99.57	0.9836	0.9972
Naive Bayes	98.53	0.9468	0.9889
Decision Tree	99.57	0.9944	0.9861
SVM	99.69	0.9863	1

V. Conclusion

In this article, we used principal component analysis to solve the occupancy prediction problem and analyzed the effect of the dimensionality reduction technique on the efficiency of the support vector machine model. In the first section, PCA was applied to the occupancy dataset and it was discovered that the first two components covered 88.71 percent of the explained variance. In the second section, we briefly introduced the machine learning classifier, support vector machine, which has as objective to predict whether an office is empty or occupied. Afterward, we emphasized the importance of the hyper-parameters tuning of the model. Next, The initial dataset and the transformed data (after using PCA) were fed into the SVM classifier separately. An analysis of findings revealed that the classification accuracy declined with the decrease of the number of extracted features. Finally, a comparison was made between SVM and other known classifiers and it showed that the SVM model produced the best results in terms of accuracy, precision, and recall.

References

[1] Z. Chen, Q. Zhu, M. K. Masood, and Y. C. Soh. Environmental sensors-based occupancy estimation in buildings via ihmm-mlr. *IEEE Transactions on Industrial Informatics*, 13(5):2184–2193, 2017.

[2] Luis Candanedo Ibarra and Veronique Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models. *Energy and Buildings*, 112, 12 2015.

[3] A.M. Martinez and A.C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[4] Pushpak Dave and Jatin Agarwal. Notice of removal: Study and analysis of face recognition system using principal component analysis (pca). In *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, pages 1–4, 2015.

[5] A.BenHamza. Advanced statistical approaches to quality. In *unpublished*.

[6] Z. Jaadi. A Step-by-Step Explanation of Principal Component Analysis (PCA), 04 2021.

[7] B. Team. How to read PCA biplots and scree plots - BioTuring Team, 09 2018.

[8] Shanmukh Reddy Manne, Kiran Kumar Vupparaboina, Gowtham Chowdary Gudapati, Ram Anudeep Peddoju, Chandra Prakash Konkimalla, Abhilash Goud, Sarforaz Bin Bashar, Jay Chhablani, and Soumya Jana. Efficient screening of diseased eyes based on fundus autofluorescence images using support vector machine, 2021.