Master's Thesis 2021

Occupancy Estimation Using Machine learning Algorithms

Yosief H. Abraham

Building Occupancy Estimation Using machine learning algorithms

By: Yosief Hailemariam Abraham

Email: yh.abraham@stud.uis.no

Date: June /15/2021

Master Thesis work carried out at Veni As in collaboration with the

University of Stavanger

Supervisors

Pål Bårdsen, palb@veni.no
Rong Chunming, chunming.rong@uis.no
Olav Løining , ol@veni.no
Bjørn Lunde Kristensen, blk@veni.no

Abstract

Building occupancy recently has drawn the attention of many researchers. With the advance of new technologies in AI and IoT, it has become possible to further optimize building energy consumption without compromising comfort of the occupants. In this thesis project, occupancy is estimated by training models on data collected from the building called Arkivenshus in Stavanger. The data collected includes measurements of electricity consumption, ventilation, hot and cold-water consumption and PIR sensors (Passive infra-red sensors). The models that are trained are classification algorithms such as KNN, decision tree, random forest, and support vector machine. Data from the building is collected over two months period where data points are collected every 15min.

Occupancy detection solutions that employ cameras, WIFI activities etc can be used to detect occupancy in buildings, however these solutions can be intrusive, costly and computationally expensive. Moreover, PIR sensors which are used for activation of lighting systems detect occupancy, they however cannot be directly related to the count of number of people. To estimate the number of people inside building I have labelled the data in five categories, where 1 represents counts less than 5, 2 represents between 5 and 25,3 represents between 25 and 50, 4 represents between 50 and 75 and for counts greater than 75 they are represented by class 5. Due to the pandemic I was not able to register number of people inside the building more than 80, which presumably has an impact on the efficiency of my model.

The performance of the models are compared using various metrices, Since the data is nor balanced and I have divided the target into five classes, looking only the accuracy of a model is a bit misleading in selecting the best model. Considering accuracy, confusion matrix and learning curves of each model the best performing model is found to be SVM (Support vector machine).

Acknowledgements

I would like to extend my heart felt appreciation to Pål Bårdsen at Veni AS and Rong Chunming at the University of Stavanger for their advice and guidance throughout this project. I would also like to thank Olav Løining and Bjørn Lunde Kristensen for providing the needed data and giving guidance during the initial stage of the project.

Contents

Chapter 1

Introduction

Buildings have been using BMS (Building management systems) for more than 50 years to optimize energy consumption. BMS is essentially a computer-based control system that monitors and manages a building's mechanical and electrical equipment, including ventilation, lighting, power, fire and security systems [1]. With the advent AI and IoT technologies operational efficiency of buildings can be increased or buildings can be made even more smarter.

Today, a smart building management system can lock down entrance doorways during off hours, reopening them during operating hours. With AI, potentially, the system could automatically open doors for entry during off hours if a fire or other emergency were detected in a portion of a building. A sophisticated AI system connected to motion sensors would even direct firefighters or police to where victims or perpetrators were located in a facility [2].

In a more everyday example, a smart building temperature control system today will know to turn on air conditioning at a certain time in advance of the building's opening, to allow tenants to arrive at work with a comfortable temperature. With AI, a system could potentially monitor real-time weather forecast conditions with added information from sensors indicating if it will be cloudy on a large glass building face, thus lowering the need for AC [2]

The central part in todays AI and IoT technologies is the ability to analyse and collect data. The huge amount of information collected from digital devices, provides insights about the operations, use and condition of everything from the building's infrastructure, physical environment, climate, water and energy usage, to an occupant's experience and satisfaction [3].

Occupant detection is one of the interesting applications that smart buildings offer. There many types of solutions, including PIRs, surveillance cameras and harnessing WiFi's channels.

However, many of these proposed solutions are limited either by their reliance on privacy-infringing hardware such as cameras, or require the user to carry a device, or they cannot identify populations beyond a limited group of people [4].

1.1 Problem Definition

The aim of the thesis is to estimate the number of people inside the building by using the already installed devices which give measurements on how the building is behaving. Mainly I am using the energy consumption of the building to set my features which then is used to train models for estimating number of people inside the building.

It is possible to benefit from detecting the number of people inside a building in many ways. Since the building which is chosen for study in this thesis is a public building where the building is used to for archiving documents, testing centre and library, during the week people visit the building for many different purposes. One possible use of having the information or estimation of people inside the building could be service planning, for example there is a canteen inside the building and having the information of people inside the building at hand the canteen can plan amount of meals to prepare.

The project is preformed on a building in the city of Stavanger, the building is known as Arkivenshus. All the features and the targets to train my model are collected from this building.

1.2 Related Work

The recent advance in IoT technologies has enabled modern day buildings integrate variety of sensors to collect information about the context of a given building. Buildings today accommodate sensors that measure temperature, motion sensors, humidity sensors and sensors that measure amount of CO2 inside building, moreover, detecting human activity inside a building is also conducted using passive infrared (PIR) sensors, video-cameras, infra-

red cameras, light beams installed in door frames and device-free localization (based on radio signals). All this measurement can be used to predict occupancy inside building [5-9].

To detect human presence inside a building several techniques are employed. As mentioned in the previous paragraph passive infrared (PIR) sensors are among types of sensors which are used to detect movement of people inside a building. They are normally used to activate lighting systems and security alarms. However, they can be inaccurate due to their inability to correctly identify the exact number of people passing by through a location when the number of people is large. Moreover, using PIRs for the purpose of counting people may need a new installation in specific sites which increases the cost. In [10-11] attempt is made to build people counters using PIRs by extracting motion patterns from the raw sensor data with an infinite hidden Markov model (iHMM) and using those patterns to infer the number of occupants using basic statistical regression methods. This system is well-suited to the adaptive setting on active deployment whereby the iHMM readily finds new motion patterns in the signal as new data arrives. However, there is a challenge in obtaining accurate estimation, since occupants can easily block other occupants from the field of view. Moreover, range motion is that can be differentiated with types of sensors is limited. For example, if two or more people are sufficiently active and close to the sensor to generate more than the maximum range of detectable motion, the sensor would be unable to detect the motion patterns of the rest of the occupants. That is, the occupants occlude each other not only by physically hindering the field of view of the sensor, but also by exceeding the maximum range of motion that the PIR can measure. Among other techniques used to detect occupancy accurately is CO2 concentration measurements which are available in many modern buildings. These measurements are very good indicators of human occupancy in a building.

Buildings can benefit from monitoring the level of CO2 concentration indoors. For example, ventilation units can be activated in a room when the CO2 level exceeds a certain threshold, it is also possible to adjust temperature of rooms if occupancy is detected using the CO2 concentration. By doing so HVAC control strategies benefit from robust occupant presence estimation [12-13].In addition sensors which measure CO2 concentration are installed in modern day buildings at relatively cheaper prices. Hence, CO2-based occupant detection is an inexpensive way to integrate demand-based control algorithms into building automation systems.

Although CO2 concentration measurements are excellent indicators of human presence in buildings, they may be affected negatively by many other factors such as the opening of windows and doors, the delay time between the occupation pattern and the CO2 concentration which have a strong influence on the CO2 concentration and their effects may affect the prediction process negatively[14].

By installing a surveillance camera in a building or a room it is also possible to detect occupancy. This method has both the ability to count number of people and detect the specific location a person is residing.

In [15] a multi-camera network is used to detect and track occupants in a large building between multiple camera views. The work from [16] developed an occupancy tracking and identification system which implemented various image processing techniques to infrared depth frame images recorded with a ceiling-mounted Kinect camera to anonymously detect and track persons entering or leaving a room and, consequently, counting the number of occupants in the room. The method demonstrated an accuracy of more than 98% in a stress test, 100% in a range of relevant function test and 99% in a three weeklong room occupancy test. Image based methods of occupancy detection are the most accurate, however they have limitation of privacy concerns and are expensive to install.

Wi-Fi activities are among the methods which are used to detect occupancy. One of the main advantages of using Wi-Fi data is its availability at no additional cost to building managers and operators. Reports can be generated using the Wi-Fi network administration system in a relatively short time, thus eliminating the need to invest in personnel, equipment[17].However, new smart phone models have the battery-saving function that will switch off Wi-Fi communication in the idle mode, causing significant detection uncertainties

Chapter 2

Data

2.1 Data Collection

In this project the data is collected from the various meters installed to in the Building. The Building's name is Arkivenshus (Archive house), which mainly serves store paper archives from the region and from all over the country. The actual number of people are counted by going physically to the building and counting numbers of people coming in and out of the building during working hours(7am-5pm).

The data collected from the building are power and energy consumption measurement of different types, air volume measurements, PIR sensor measurement. As features I have mainly used the power measurement with some feature engineering.

2.2 Features

I have used 7 features to train my model, one of the features 'light_load' is not directly found from the measurements provided, but was extracted from the main meter measurement the following measurements

- Domestic hot-water power measurement
- Ventilation System 5 power measurement

- Ventilation system 6 power measurement

- Meter for electrical heater 15kw power measurement

- Meter for outdoor power measurement

- Meter for kitchen measurement

- Meter technical installation in the cellar power measurement

The other features are as follows

- Ventilation system 5 water heating power measurement

- Ventilation system 6 water heating archives power measurement

- Ventilation water cooling 001 power measurement

- Ventilation water cooling 004 power measurement

- Process cooling power measurement

- Meter for outdoor power measurement

Chapter 3

Theory

3.1 Machine Learning

Machine learning (ML) is a field of study in computer science that uses algorithms to train a computer using a collected data and through past experience. By learning from a data computer detect useful patterns which would be helpful to arrive in a conclusion. The machine learning process is the most appropriate alternative in cases where it is not possible to directly write programs to solve problems, i.e., when the solution is not a known prior, but can only be developed using data or experience. Moreover, machine learning is used when a given data is huge and intricate that a human expertise can't deal with. Machine learning has been traditionally used in the domains such as speech or face recognizing, language processing, spam filtering, etc. In the context of buildings, fundamental problems such as predicting occupant's behaviour and preferences,

forecasting energy demand and peak periods, etc., are difficult to be solved with traditional programming but potential solutions can only be learned from data[18].Now a days, ML application are being used in a range of fields where it has attracted the attention of researchers and scientists in several disciplines.

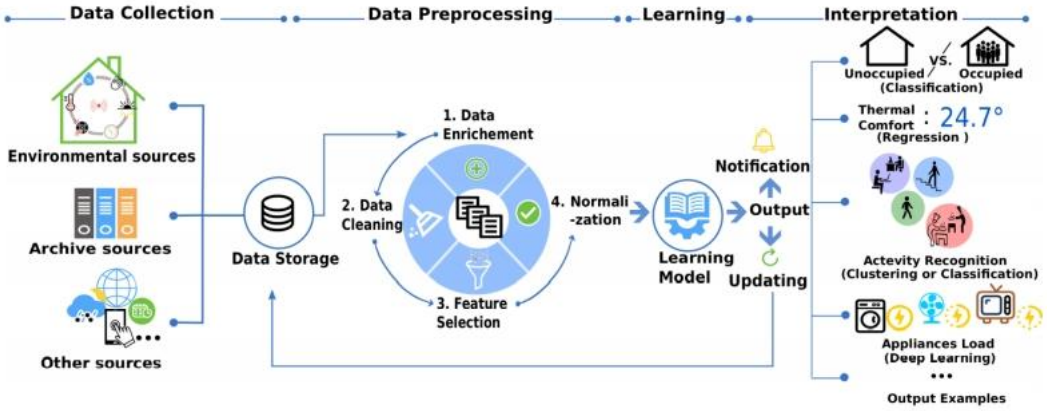

Fig. 1 Machine learning process [18]

In the above figure, the ML process is explained in a step by step fashion. Its steps are comprehensively divided in four.

A) Data collection: - Data is collected from different sources such as from sensors, archives, measurement meters etc. In our case the data is provided as csv files, the files are imported to pandas for further processing.

B) Data Pre-processing: In this stage data is pre-processed in such a way that it is made ready for the next step by detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. This step includes

i. Data enrichment: - adding statistical data such as the mean value of the samples, standard deviation, etc.

ii. Data cleaning: - Filling missing data, dropping data that includes a lot of missing values etc

iii. Feature Extraction: - Selection of the appropriate features from all the data, which depends on the task used in the learning step. PCA (Principal Component Analysis) is one of the most used methods for feature selection and dimensionality reduction.

iv. Data standardization or normalization:- Data normalization refers to shifting the values of your data so they fall between 0 and 1. Data standardization, in this context, is used as a scaling technique to establish the mean and the standard deviation at 0 and 1, respectively.

C) Learning step: the ML techniques are used to learn functions and models.

D) Interpretation step: a step where what has been learned in the previous step is interpreted based on the type of problem one is trying to solve.

3.2 Machine Learning algorithms

There are various machine learning algorithm used for different purposes, the selection of the right machine learning algorithm depends on what our problem is trying to achieve. Classification and regression are two major prediction problems that ML models are designed to handle. Classification is the process of categorizing data points into multiple categories. Regression aims to predict a data value based on a function defined by the available data points [25].

The Four classification models, K-nearest neighbours (KNN), Random Forest, Decision Tree and Support Vector Machine were used for comparison in this project, a brief description of the algorithms is discussed in the following section

3.2.1 K-Nearest Neighbours

KNN (K-Nearest Neighbours) is one of the machine learning algorithms used for classification or regression problems. It uses similarity scores to predict targets, by calculating the distance between two points the algorithm determines similarity. KNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using KNN algorithm [26].

3.2.2 Decision Tree

Decision tree is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a series of binary questions to classify data. Tree models where the target variable can take a discrete set of values are called classification trees. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

Decision Tree algorithm starts with the full data set called the root. The data set is broken down into two subsets by asking the binary question of the feature with the least Gini index. The model keeps splitting the data set into smaller subsets until all data points in the subset are labelled with the same classification, called a leaf. The flow from the root to the leaves can be treated as a classification rule. All future data samples that follow the same rule can be classified into the same class, the class of the leaf [27].

3.2.2 Random Forest

Instead of one decision tree, random forests have multiple decision trees, which were constructed from randomly chosen subsets of data. Each tree is made with different input features, so that each decision tree can lead to a different decision for classification. Instead of relying on a decision from a single decision tree, the final decision is made based on the majority vote of different trees in the forest. Because Random Forests intentionally limit the number of features to train within each single tree model, it has less over-fitting problems than the conventional decision tree. However, Random Forests may require more input features and data for training in order to develop an accurate classification model [28].

3.2.3 Support Vector Machine

Support Vector Machine (SVM) is another machine learning algorithm based on minimizing the risk. The main objective of SVM is to construct a boundary that best separates a dataset into different classes. Support vectors are the data points nearest to the boundary, and these data points are considered as the most critical elements to determine the boundary. SVM sets the boundary in a way that the distance from the boundary to each class is maximized, so that future data can be classified with more confidence [29].

Chapter 4

Method

4.1 Tools and Technologies

4.1.1 Scikit-learn

Scikit-learn (formerly scikit learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy [19].

In this project in all the machine learning model scikit-learn has been used. More-over the model evaluation and data splitting are performed using the methods(function) that scikit-learn provides.

4.1.2 Pandas

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal. For data scientists, working with data is typically divided into multiple stages: munging and

cleaning data, analysing / modelling it, then organizing the results of the analysis into a form suitable for plotting or tabular display. pandas is the ideal tool for all of these tasks [20].

I have imported my data to Pandas data frames to work on the data and perform the needed analysis on Jupiter notebook. The data is originally provided in csv files.

4.1.2 NumPy

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

In this project NumPy arrays are used as an input the models that I trained, all the machine learning models accept their input in a NumPy array format and not in a pandas data frame format [21].

4.1.3 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib produces publication-quality figures in a variety of hard copy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface tool kits.

Matplotlib library is used in almost all of the visualization plots in this project. Mostly NumPy arrays are given as inputs to plot, but at times Pandas data frames could also be used [22].

4.2 Pre-processing and visualization

4.2.1 Pre-processing

After importing the data into Pandas data frame, missing values filled by using the bfill or ffill methods from pandas. Moreover , the energy consumption measurements are provided in a cumulative manner, in order to find the energy consumption at every 15min I used the shift method to create another column shifted by 15min from the original and subtracted it from the original , which gives the energy consumption at every 15min.From the new column it is possible to find the power measurement by multiplying every values by 4.

The temperature data is downloaded from yr.no, since the data is provided in 1hr time step, I had to do some coding in order to make it for every 15min, so I can have the same time steps as the energy measurements.

4.2.2 targets

The targets are the number of people residing inside the building. This data was collected by going physically into the building in the morning hours from 7am to 9:30am and 2:30pm-5pm for a week. In the second week which was after Easter 2021, I have tried to collect data, however, there was not much difference with what was collected previously. Hence the targets were mapped to the features for the two months period data provided.

4.2.3 visualization

After importing the various measurement into Pandas data frame a range of plots of the measurement are done to visualize our data. In the following section the plots are presented to show how various measurements relate with each other.

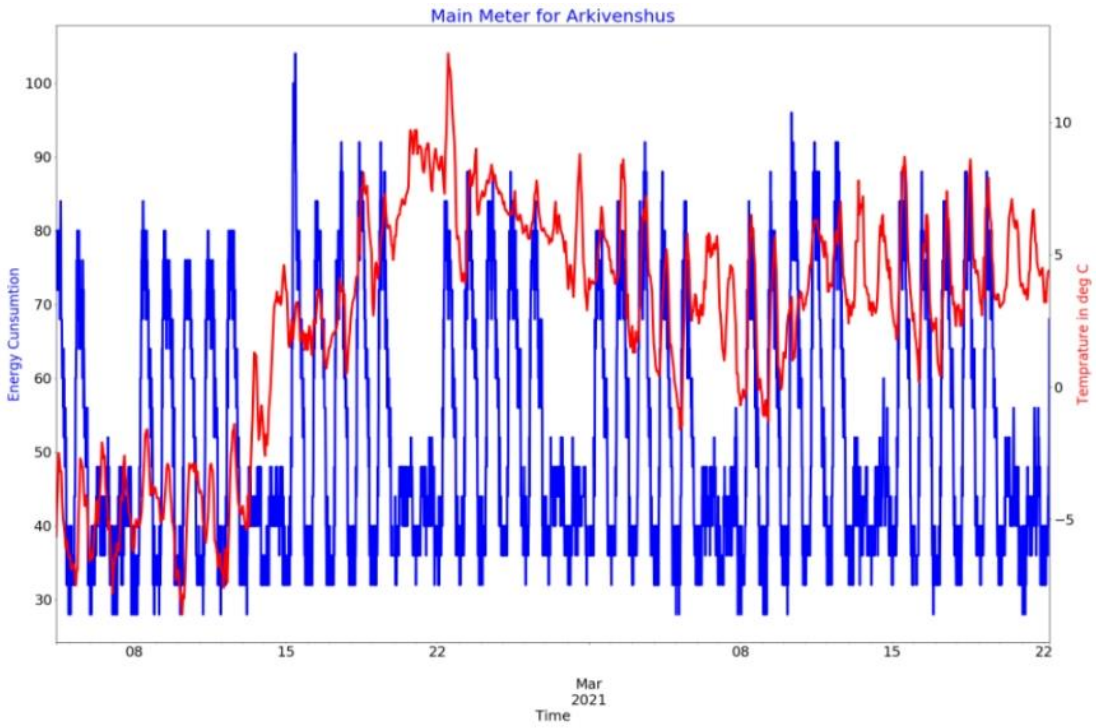

Fig. 2 Plot main meter Measurement Vs outside temperature

In The previous figure(Fig.2) the measurement of the main meter and the outdoor temperature over a time period of 04,February 2021 to 22, March 2021.As we can see in the plot the energy consumption is relatively lower during weekend than is during week days. To see closely how the energy consumption varies with temperature a plot for a shorter period of time is presented in the next figure (Fig.3). The plot of the two measurements does''t seem to have much correlation between each other specially during night time when presumably people are not inside the building the energy consumption is at a certain minimum level while the outdoor temperature does''t seem to follow the same trajectory the energy consumption follows.

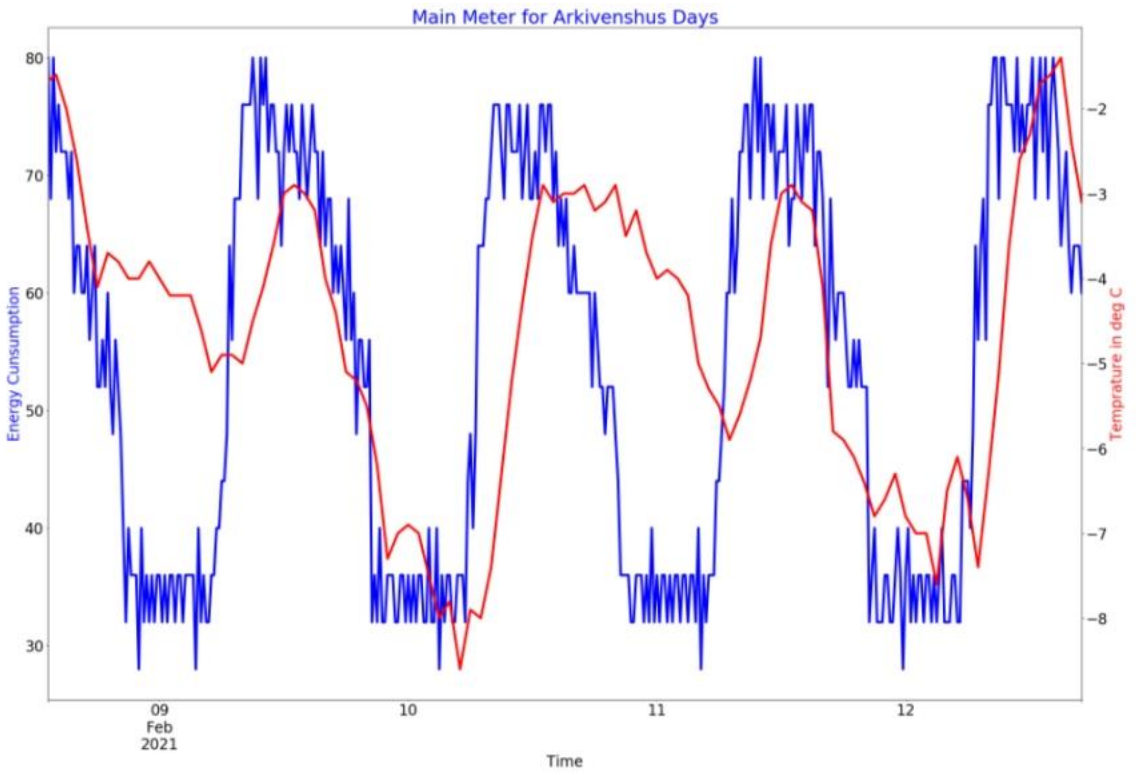

Fig.3 Main meter Vs. Temperature Day time

The above figure shows the plot of the main meter measurement and outdoor temperature over the period of 08, February 2021 and 12, February 2021. This plot is made to show how the profile of the energy consumption of the building varies against outside (Air) temperature over weekdays. As we can see, the two plots don't strictly follow each other.

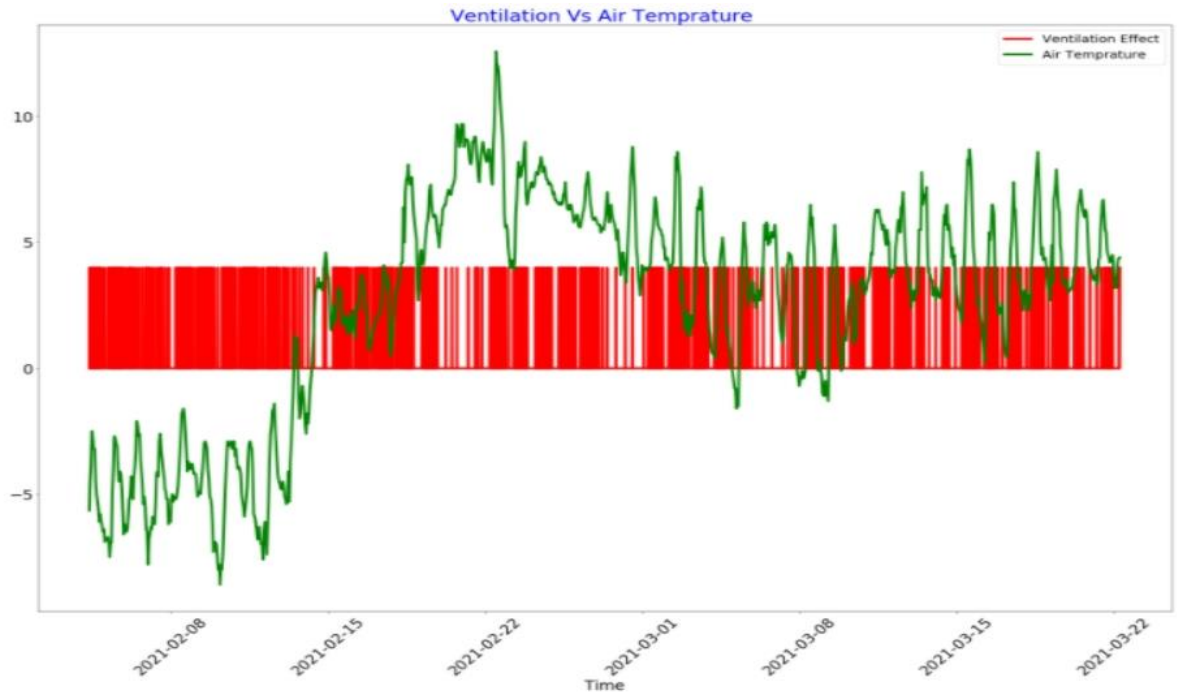

Fig.4 Ventilation Effect Vs. Outside Temperature

In Fig.4 the plot shows the outdoor temperature and the power consumption of ventilation system 6 in the building. The ventilation system has a constant maximum power consumption and constant minimum power consumption. From the plot we can observe that there is no correlation between the two measurements. Th plot show for the period 04 February 2021 to 22 March 2021. In the plot we see that the energy consumption of the ventilation system 6 in the building have a very uniform profile though out the time period.

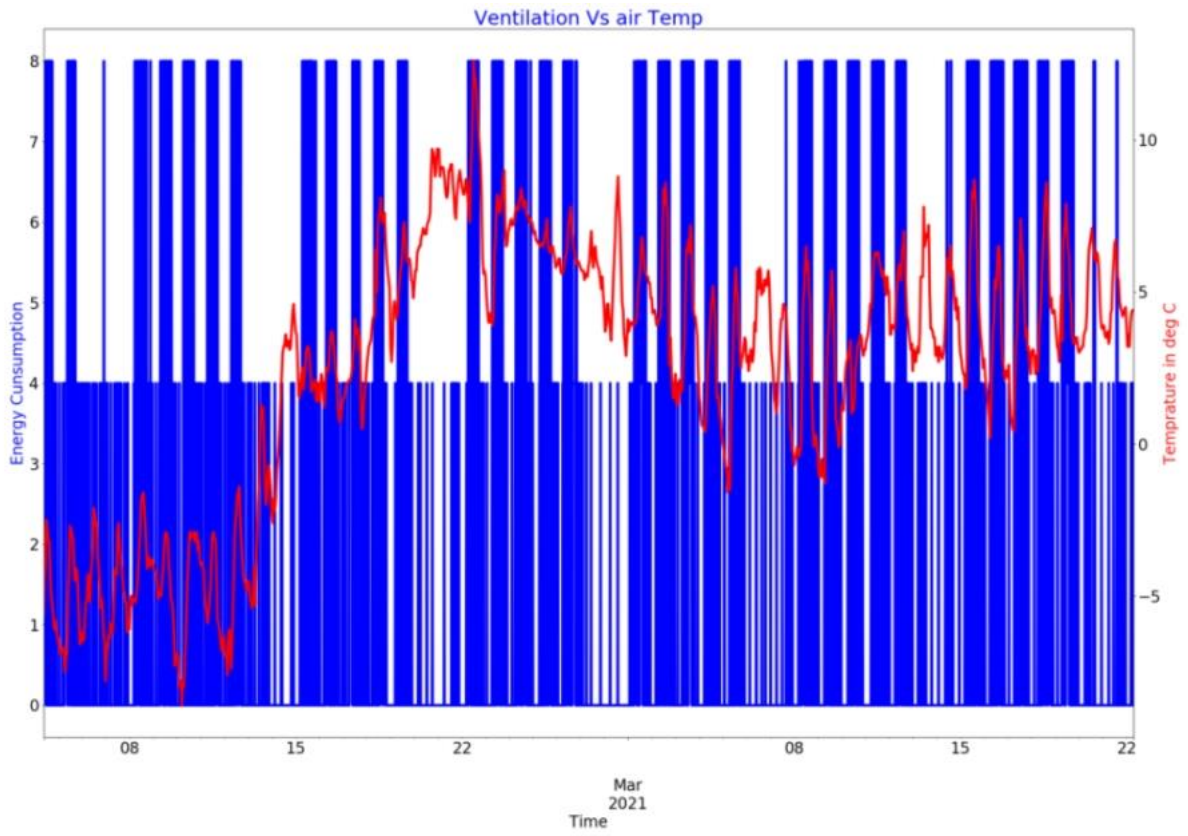

Fig.5 Ventilation in system 5 and 6 Vs. Outside Temperature

In Fig.5 the plot shows the power consumption of both ventilation systems 5 and 6. As we can see in the plot for energy consumption has a uniform profile while the temperature varies over time. Moreover, the ventilation system 6 has a longer life time than the ventilation system 5 where we can conclude the system 6 is on most of the time to ventilate the building where as system 5 on runs in the weekdays and during the day only.

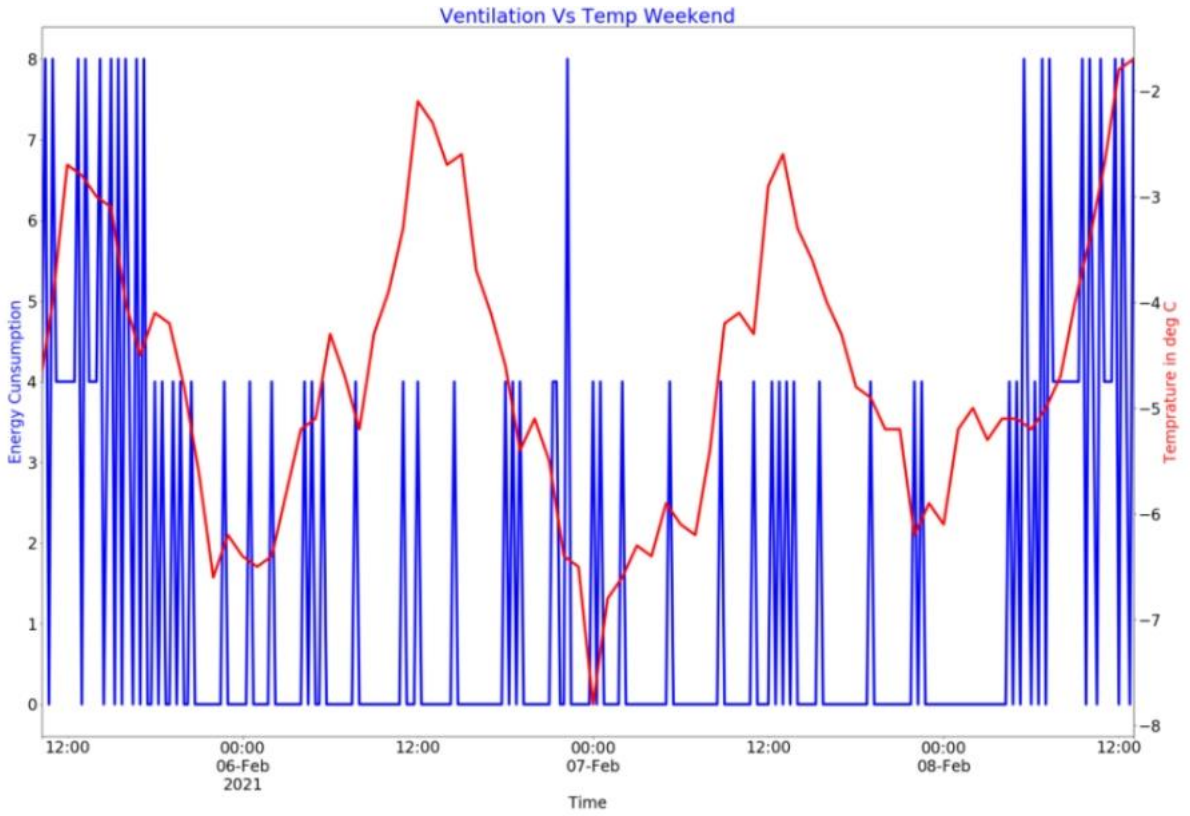

Fig.6 Weekend plot for Fig.5

In Fig.6 the plot shows a closer look in to how the profile of ventilation systems and the outdoor temperature over a weekend varies. As we can see from the plot the ventilation system has a uniform profile while the temperature varies.

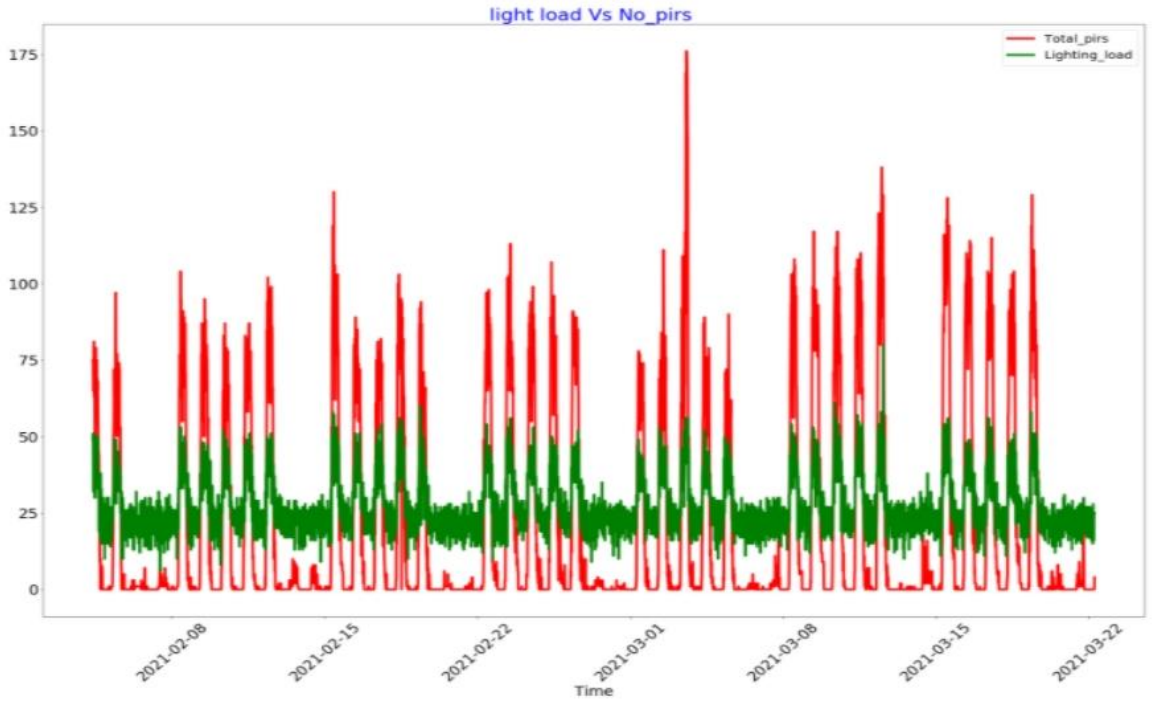

Fig.7 Electricity consumption (lighting and power outlets) Vs. PIRs count

In Fig.7 the plot shows the PIRs activation and the power consumption of the lights and power outlets inside the building. As we can see from the plot it is clear that the two measurements has a strong correlation with each other.

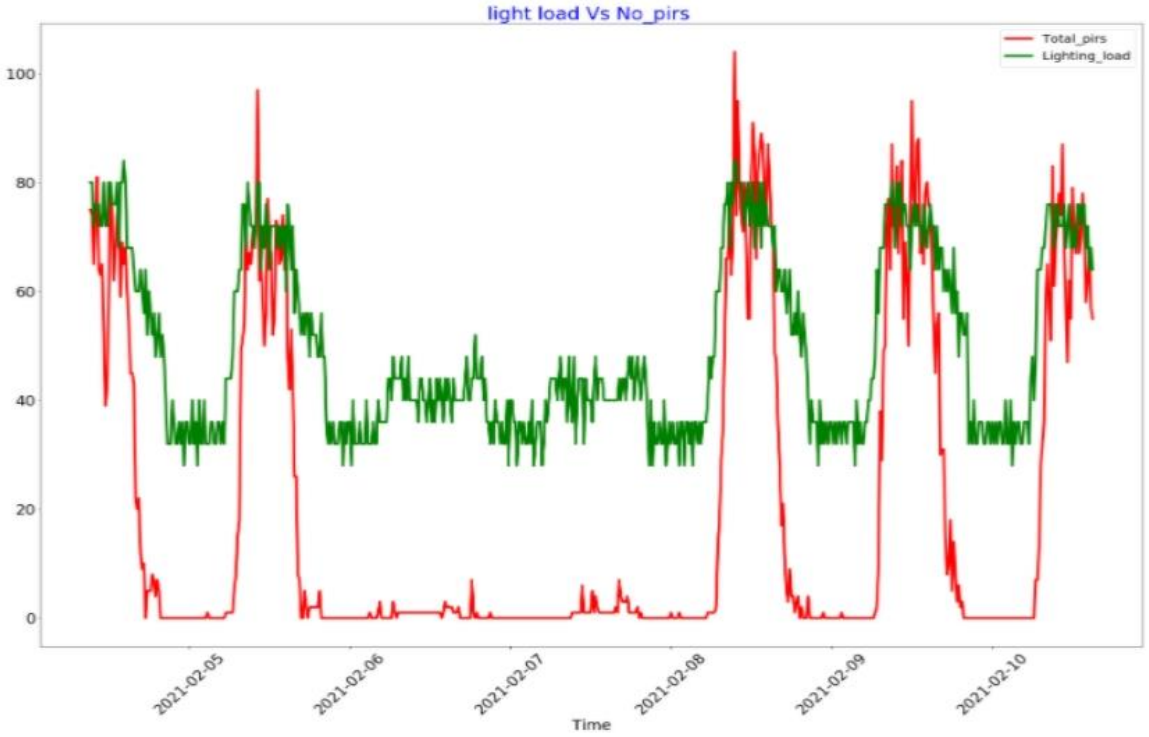

Fig.8 Weekend plot for the previous plot

In Fig.8 the plot shows the PIRs activations and the power consumption of lights and power outlets over a specific weekend. In this plot we have a closer look on how the weekend profile of the two plots looks like. The power consumption is at its lowest value with some fluctuations, it shows even in the weekend the power consumption is a bit higher in the day to time than in the night time.

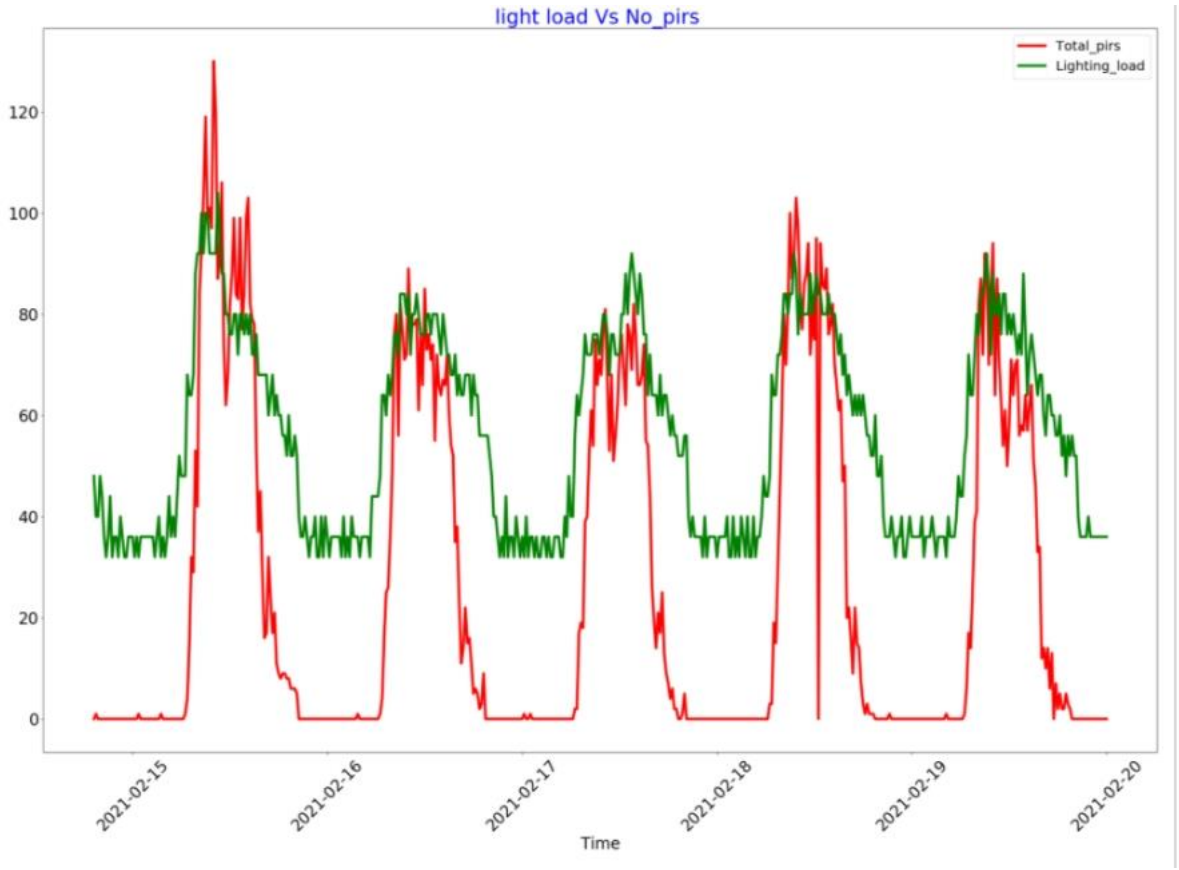

Fig.9 Weekdays plot for Fig.7

In Fig.9 the plot shows the PIRs activations and the power consumption of lights and power outlets over a specific weekday. We see that both plots strictly follow each other, the power consumption is at its pick during the day, and at its lowest point during the night. The same goes for the PIRs, The PIRs counts movement of people around the building.

4.2.4 Correlation graphs

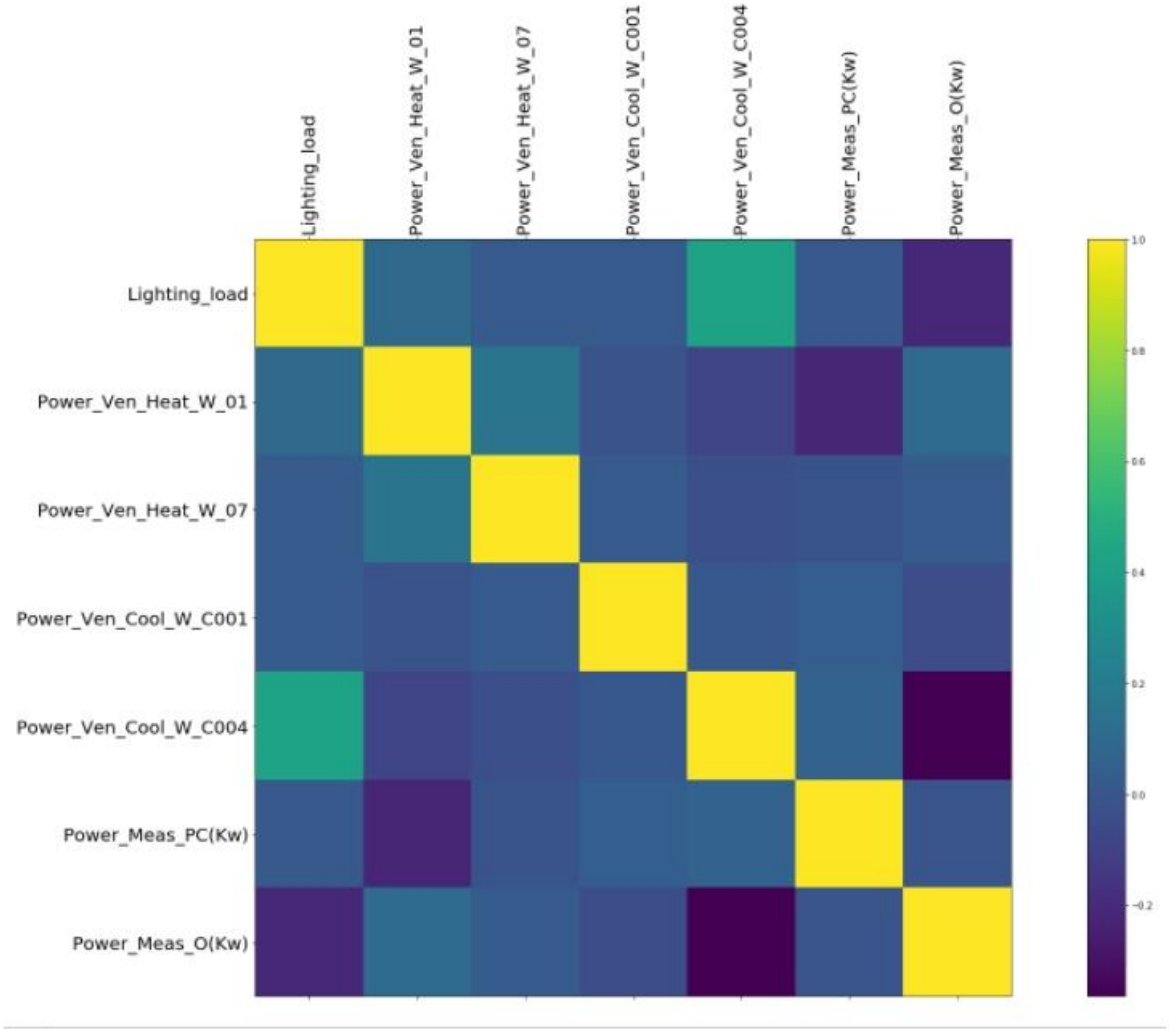

Fig. 10 covariance matrix for data features

In Fig.10 a correlation graph is plotted to show the relationship between the features. From the plot we see that the light_load and power_ven_cool_w_004 features are the most strongly correlated ones among all the features. For more detailed relationship between the features we can look at Table. 1

Lighting_load	Power_Ven_Heat_W_01	Power_Ven_Heat_W_07	Power_Ven_Cool_W_C001	Power_Ven_Cool_W_C004	Power_Meas_PC(Kw)	Power_Meas_O(Kw)
1.000000	0.105452	0.027844	0.029661	0.428930	0.016554	-0.209246
0.105452	1.000000	0.162097	-0.017074	-0.080038	-0.216572	0.114662
0.027844	0.162097	1.000000	0.028009	-0.030495	-0.008016	0.031635
0.029661	-0.017074	0.028009	1.000000	0.012037	0.041571	-0.039722
0.428930	-0.080038	-0.030495	0.012037	1.000000	0.066543	-0.364218
0.016554	-0.216572	-0.008016	0.041571	0.066543	1.000000	-0.003175
-0.209246	0.114662	0.031635	-0.039722	-0.364218	-0.003175	1.000000

Table.1 Corelation table of data features

In Fig.11 we see the detailed correlation between each feature, we observe that some of the correlation coefficients are negative, which shows that features are negatively correlated.

4.3.PCA (principal component analysis)

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because

smaller data sets are easier to explore and visualize and make analysing data much easier and faster for machine learning algorithms without extraneous variables to process [23].

4.3.1 Standardization

In order to do PCA the first step we do is standardize our dataset, so that each feature contributes equally to the analysis. More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem [24].

4.3.2 Covariance Matrix computation

By computing the covariance matrix we learn how each variable in the data set varies from the mean, or to see the relationship of variables between each other. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix. In Fig.11 we see the covariance matrix for our data, each value in the covariance matrix tells us how strongly variables are correlated with each other, more over the signs explain the direction of increase or decrease between two variables, that is to say if the sign is positive the two variables increase or decrease together and if the sign is negative then as one of the variables increases the other variable decreases and vice versa[30].

4.3.3 EIGENVECTORS AND EIGENVALUES Computation from Covariance Matrix

In order to determine the principal component of the data we need to compute Eigenvectors and eigenvalues. In this project I don't need to compute eigenvectors and eigenvalues python takes care of the previous two steps to determine the principal components of the data.

4.3.4 Explained Variance

Principal component analysis transforms variables of a data set in to new set of variables, the new variables represent the original data without losing much information, in such a way that it is possible to restore the original data from the transformed variables[31].In addition, the total variance remains unchanged but is redistributed among the new variables. The first PC (principal component) explains the most variance and then the second etc. In a more general sense, the first k principal components explain the most variance. The principal component analysis performed in our data set shows that the first two principal components explain

65.6% of the variance. in the following table we see how the variance is distributed in the 7 components and the ration explained variance by each component.

	explained_variance	explained_variance_ratio
0	0.058554	0.353720
1	0.050170	0.303075
2	0.019755	0.119340
3	0.011508	0.069520
4	0.010540	0.063669
5	0.009668	0.058406
6	0.005342	0.032269

Table. 2 Explained variance by each Principal component

In Fig.11 , we see the explained variance by each component on the horizontal axis and the percentage explained variance on the vertical axis, moreover it is explained in the graph that we need to keep six principal components to have 95% of the variance explained.

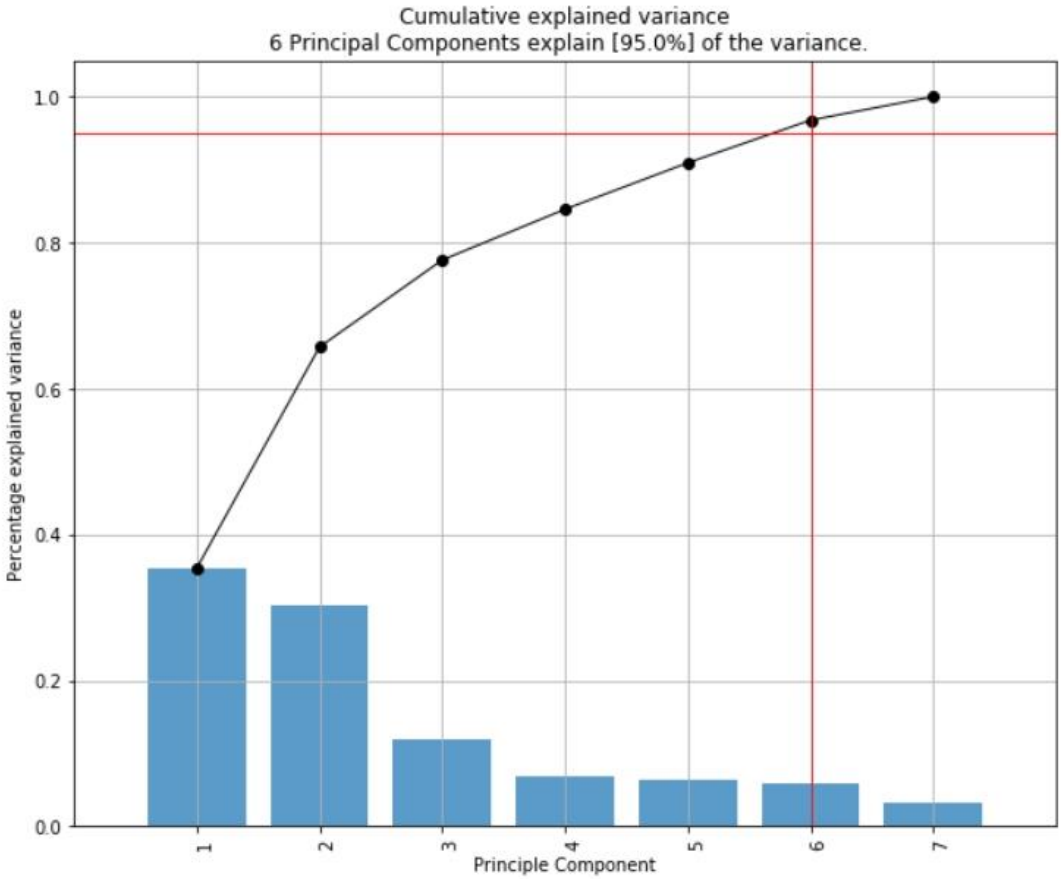

Fig. 11 Plot for explained variance for each Principal Component

4.3.4 Biplot

A biplot overlays a score plot and a loading plot in a single graph. As it is shown in Fig.13. Points are the projected observations; vectors are the projected variables. If the data are well-approximated by the first two principal components, a biplot enables us to visualize high-dimensional data by using a two-dimensional graph [32]. In the Fig. 12 we observe that var1 and var5 has a lot of positive influence in PC1 and little influence on PC2, var7 has a negative influence on PC1 but does’ˈt seem to have an influence on PC2,var4 seems to have a lot of negative influence in PC2.

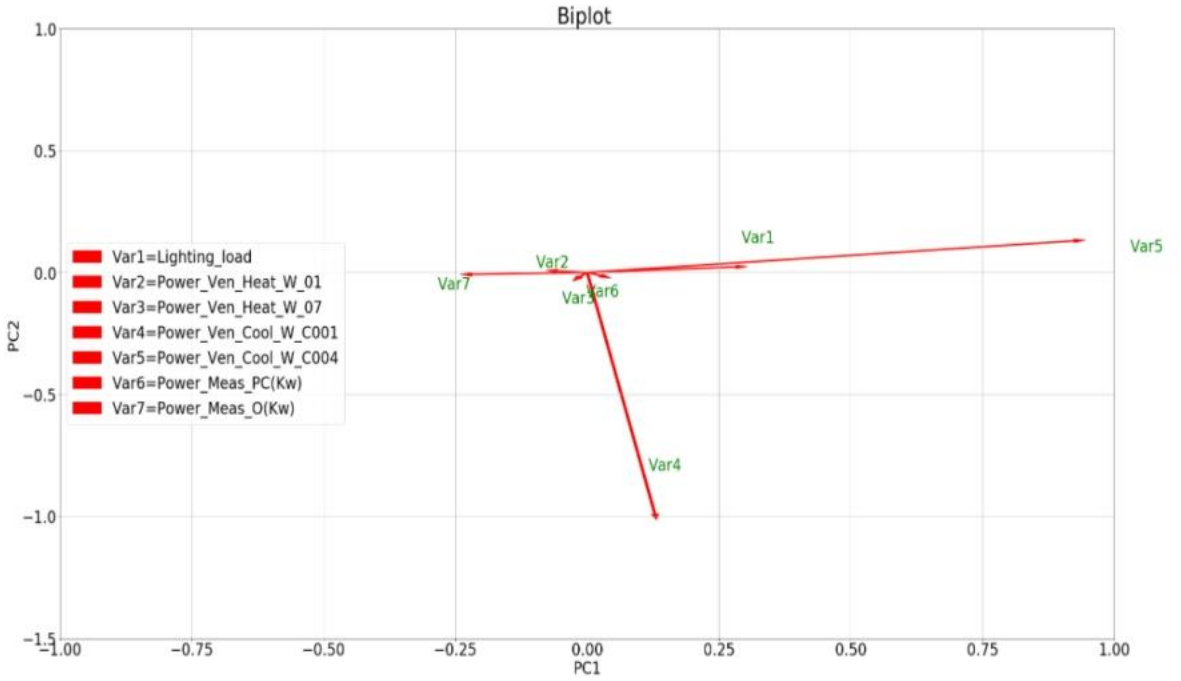

Fig.12 Biplot on the various data features

Chapter 5

Model Selection and Evaluation

5.1 Model Selection

In machine learning there are a range of learning models to choose from on a given prediction or classification model. However, any given model has several limitations depending on the data distribution. None of them can be entirely accurate since they are just estimations. These limitations are popularly known by the name of bias and variance.

A highly biased model, therefore, is one which oversimplifies the relationship or mapping between the features and a target (e.g.: in Linear Regression, irrespective of data distribution, the model will always assume a linear relationship). A highly biased is also known as an under-fitted model.

A model with high variance tends to pay much attention on the training data and struggles to generalize on the data it hasn't seen before, hence the model performs well on the training data and comes with a lot of error on the test data.

Model selection can be done both across different models or on the same type of model which is configured using different hyper-parameters. For example, if we have a data set where we need to develop a classification or regression model then we fit different types of models and by evaluating each model we choose the best performing model as our final one.

5.1.1 Model Selection Technique

The ideal situation for during our model selection process is to have enough data where we can split the data set in three parts, training data set, validation data set and test data set. We begin by fitting the training data into the model, evaluate and select on the validation data set and we report the performance of the model on the test data set.

In cases where we have not sufficient data(is almost always the reality in regards to availability of data) we might be in need of more training data set which would mean fewer validation data set, this leads to relatively noisy estimate of predictive performance. Generally, we have two methods of model selection techniques, the probabilistic measurements and re-sampling method.

In the probabilistic measure the model is analytically scored using both the training data set and complexity of the model. Information Criteria (IC) is used to correct the bias of maximum likelihood by the addition of a penalty term to compensate for the over-fitting of more complex models [24]. The score that we get from IC tells us how good our model is, hence the lowest the score the best the model is. Moreover model with fewer parameters is believed to be less complex since it can generalize better on average.

Four commonly used probabilistic model selection measures include:

Akaike Information Criterion (AIC).

Bayesian Information Criterion (BIC).

Minimum Description Length (MDL).

Structural Risk Minimization (SRM).

Probabilistic measures are appropriate when using simpler linear models like linear regression or logistic regression where the calculating of model complexity penalty (e.g. in sample bias) is known and tractable-sampling method in model selection is the process of repeatedly

drawing a sample data from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model[33-34].

Some common re-sampling techniques are: -

- Random train/test split
- Cross validation (K-fold, LOOCV)
- Bootstrap

Out of the above-mentioned methods the most popular method is the cross-validation method which includes many subtypes.

The K-fold cross validation method splits the training data into K subsets or folds, one of the subsets is then used as a validation set while the remaining are used to fit the needed model. This process is repeated until every fold or subset is used as a validation set, finally the average of prediction error at each trial is taken to decide the performance of the model.

The leave one out cross validation is a special form of the K-fold cross validation where it takes one data sample as a validation set and the remaining training data samples as a training set. After multiple runs the error is estimated as the mean of the errors for each run. This method is much better, because it has far less bias, since more observations are used to fit the model. There is no randomness in the training/validation set splits. Therefore, we reduce the variability of the MSE [35-37].

5.2 Bias-Variance Trade-Off

In machine learning the primary task of a model is to learn from data and predict patterns that would help to map features to target, in other words a machine learning model develops a function that estimates the true relation between features and target. It is however not absolutely known the true relationship between the features and target. We use algorithms such as KNN, SVM , random forest etc to estimate the true relationship , the model choice

that we make influences the estimation that we are looking for, hence our model will biased by the choice we are making. Another interesting influencing part in developing our model is the training data we use; models may give us two different outputs for two different sets of training data. Hence, the amount difference in the output is called the variance.

Both bias and variance contribute to the model error hence we want to keep both low to have lowest error possible. By keeping the bias low, we will be able to develop a model that performs good on training and test data. In the same fashion we want to keep our variance low by building a model which is not overly complex. An overly complex model would perform well on the training data and would perform poorly on the test data. In practice, however, we need to accept a trade-off. We can't have both low bias and low variance, so we want to aim for something in the middle. Fig.13 shows the bias-variance trade-off where we see the error decreases for the bias curve as model complexity increases and error increases as model complexity increases. The optimum model complexity is where the total error is low [38].

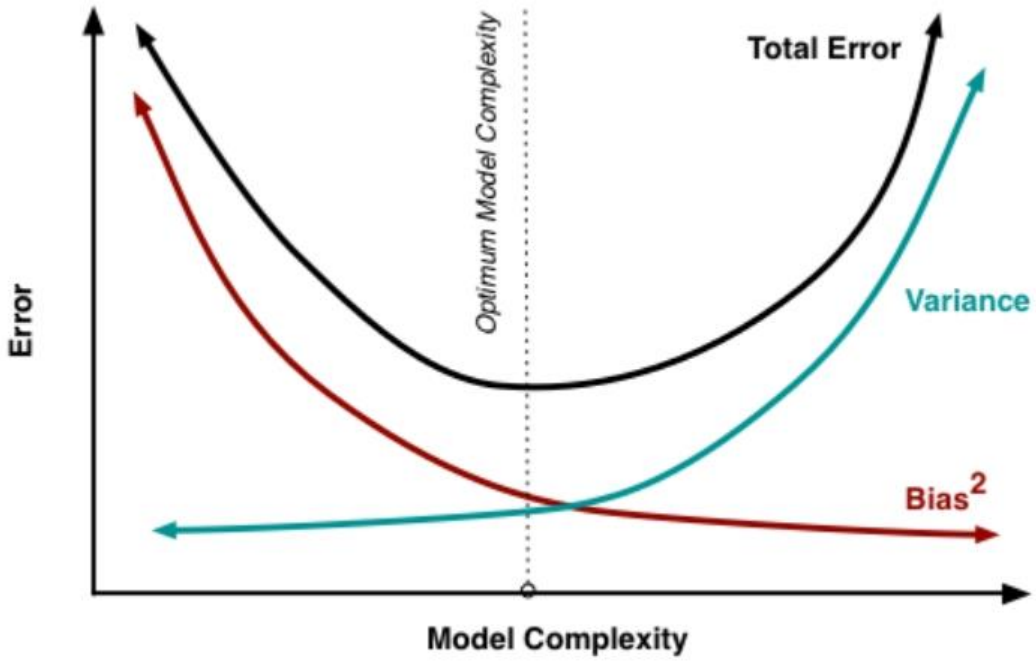

Fig. 13 The Bias-Variance trade-off [38]

5.3 Learning Curve

In machine learning a learning curve shows curves from training set and validation set, the curves show how well a model is learning on the training set and how well the model generalizes on the validation set. The shape of the learning curves can be used to learn about the behaviour of the model, by doing so it is possible to reconfigure the model in an attempt of improving it [39].

During our model development we may experience a model that is under-fit, over-fit or good fit.

'A plot of learning curves shows underfitting if:

The training loss remains flat regardless of training.

The training loss continues to decrease until the end of training.'[40]

'A plot of learning curves shows overfitting if:

The plot of training loss continues to decrease with experience.

The plot of validation loss decreases to a point and begins increasing again.'[40]

'A plot of learning curves shows a good fit if:

The plot of training loss decreases to a point of stability.

The plot of validation loss decreases to a point of stability and has a small gap with the training loss.'[40]

5.4 Confusion Matrix

The confusion matrix is a performance measurement for classification problems in machine learning. It is a table with four combinations of actual and predicted values [41].

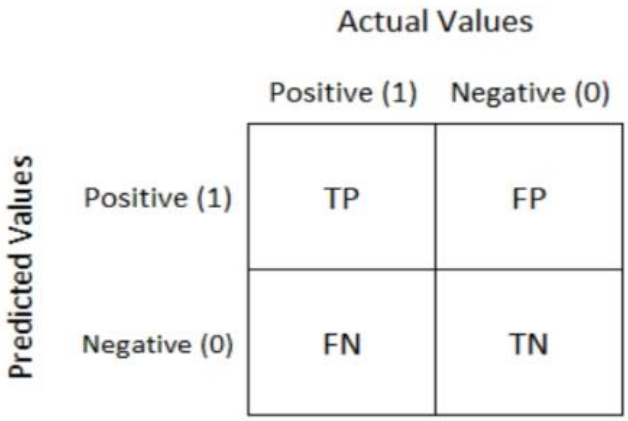

Fig. 14 Confusion matrix for binary class [41]

True Positives (TP): targets predicted are actually positive.

True Negatives (TN): targets predicted are actually negative.

False Positives (FP): Predicted targets as positives are not actually positives.

False Negatives (FN): predicted targets as negatives are actually not negatives.

In classification problems which contain more than two classes there is not negative or positive classes an example is given in the following figure.

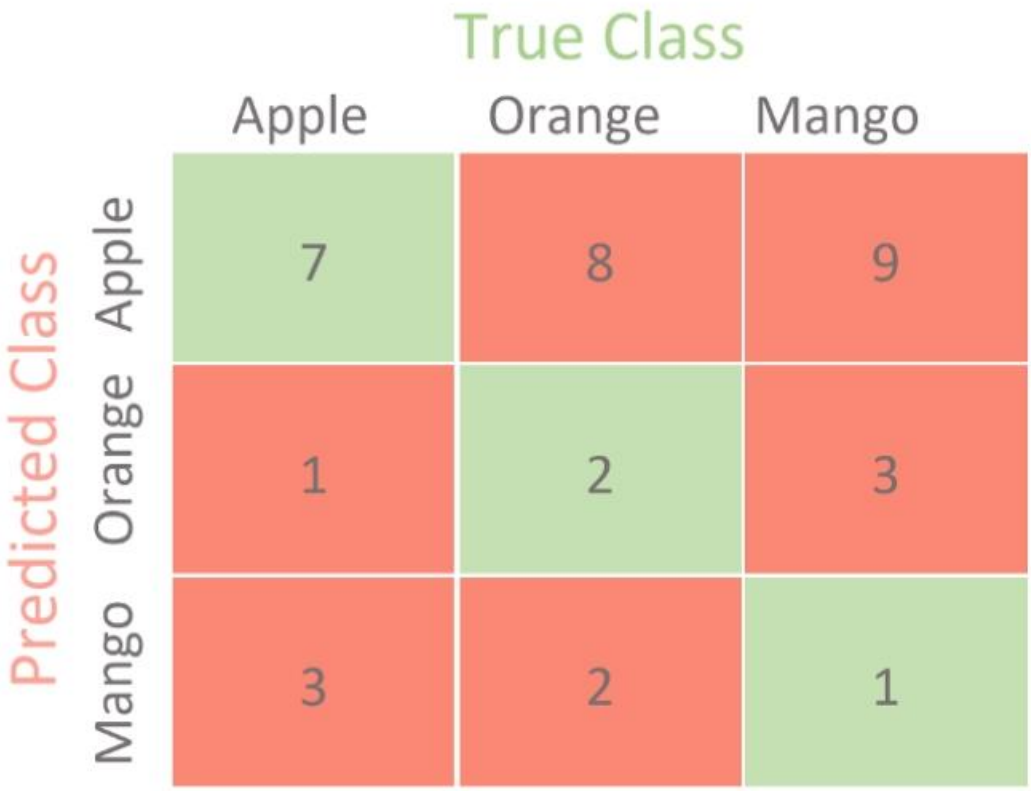

Fig. 15 Confusion matrix for Multi-class Classification [42]

It is possible to get the values of TP, TN, FP and FN for the previous confusion matrix as follows

- $TP = 7$
- $TN = (2+3+2+1) = 8$
- $FP = (8+9) = 17$
- $FN = (1+3) = 4$

From the confusion matrix presented above we can extract performance measures. Here are some of the most common performance measures you can use from the confusion matrix.

Accuracy: It gives you the overall accuracy of the model, meaning the fraction of the total samples that were correctly classified by the classifier [42].

$$\frac{TP+TN}{TP+TN+FP+FN}$$

Misclassification Rate: It tells you what fraction of predictions were incorrect. It is also known as Classification Error [42].

$$\frac{FP+FN}{FP+TN+FN+TP}$$

Precision: It tells you what fraction of predictions as a positive class were actually positive [38].

$$\frac{TP}{TP+FP}$$

Recall: It tells you what fraction of all positive samples were correctly predicted as positive by the classifier. It is also known as True Positive Rate (TPR), Sensitivity, Probability of Detection [42].

$$\frac{TP}{TP + FN}$$

Specificity: It tells you what fraction of all negative samples are correctly predicted as negative by the classifier. It is also known as True Negative Rate (TNR) [42].

$$\frac{TN}{TN + FP}$$

F1-score: It combines precision and recall into a single measure. Mathematically it's the harmonic mean of precision and recall [42].

$$\mathbf{F_1 - Score} = 2 * \frac{\mathbf{Precision} * \mathbf{Recall}}{\mathbf{Precision} * \mathbf{Recall}}$$

$$= \frac{\boldsymbol{2TP}}{\boldsymbol{2TP + FP + FN}}$$

The aim of our machine model is to get high precision, recall and accuracy. In building classification models, we are most probably going to need confusion matrix and related metrics to evaluate our model. Confusion matrices are not just useful in model evaluation but also model monitoring and model management.

The confusion matrix helps us see the abundance of false positives and false negatives. It is always part of model development in data science to test multiple models to find the most well performing model. Confusion matrix can help us do so, we can see not only how a model

is accurate but also but also see more granularly how a model does in sensitivity or specificity, as those might be more important factors than general accuracy itself.

Chapter 6

Results and Discussion

6.1 KNN Confusion matrix

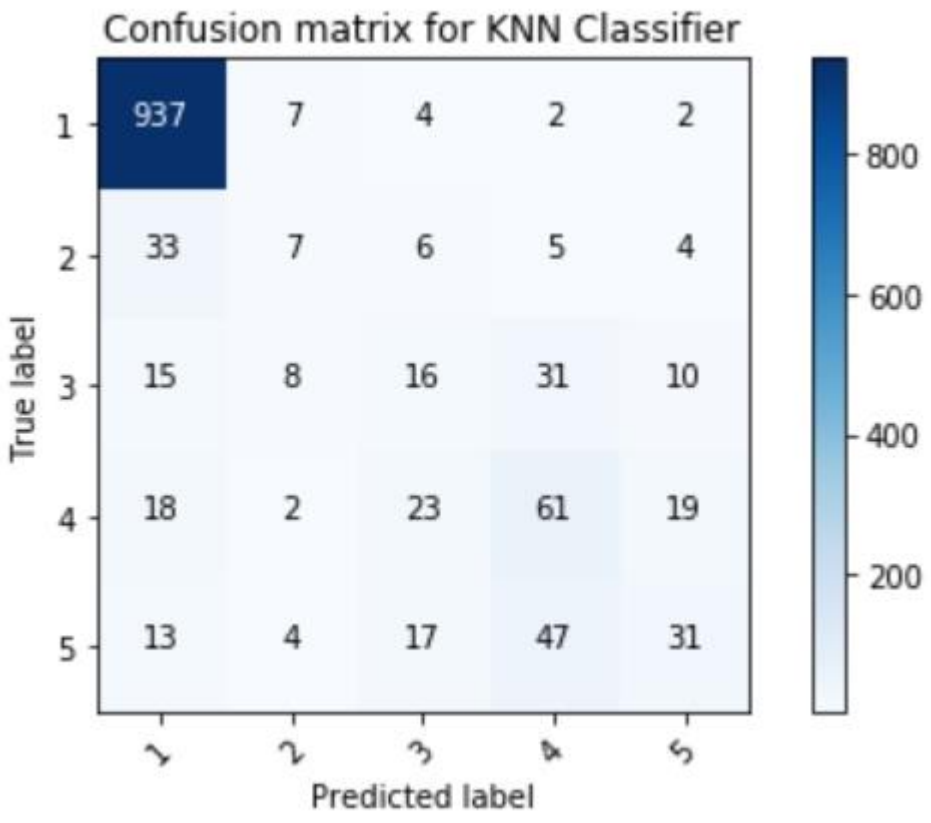

Fig. 16 Confusion Matrix for KNN classifier

From the above confusion matrix Fig. 16 we calculate the following evaluation metrics

Accuracy :- The accuracy for the KNN classifier is found to be 81.3%, The score is fairly good enough. However, to have a comprehensive idea of how our model is performing we need to check other metrics.

	precision	recall	f1-score	support
1	0.93	0.98	0.95	966
2	0.12	0.08	0.10	48
3	0.28	0.25	0.26	76
4	0.47	0.50	0.48	125
5	0.53	0.32	0.40	107
accuracy			0.81	1322
macro avg	0.46	0.43	0.44	1322
weighted avg	0.78	0.81	0.79	1322

Fig. 17 Confusion matrix report for (KNN=5)

6.1.1 Learning Curves for KNN

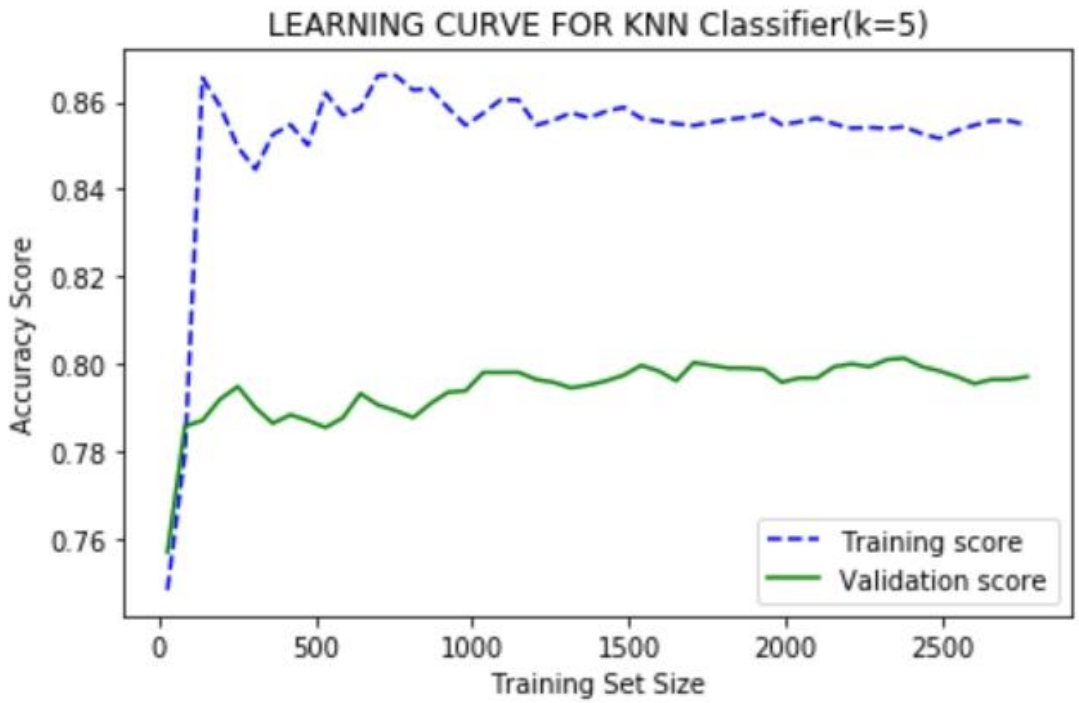

Fig. 18 learning curve for KNN classifier(k=5)

The learning curve shows that the model performs with some variance that is to say there is some over-fitting. Moreover, the gap between the two curves is big which means that we do have over-fitting case. As we can observe from the classification report our classes are not evenly divided. Our model has predicted very well for class1 where we see 0.93 precision values and 0.98 recall value. For all the remaining classes the precision and recall values happened to be very law. For class 1, the recall is higher than precision, which means that there are fewer false negatives than false positives.

Our model predicts fairly better for class 5 and class 4, in classes 2 and 3 the precision, recall and F1 score values are very low. Generally, it looks like our model would have predicted very well for a case we would classify our data as binary occupancy classification problem.

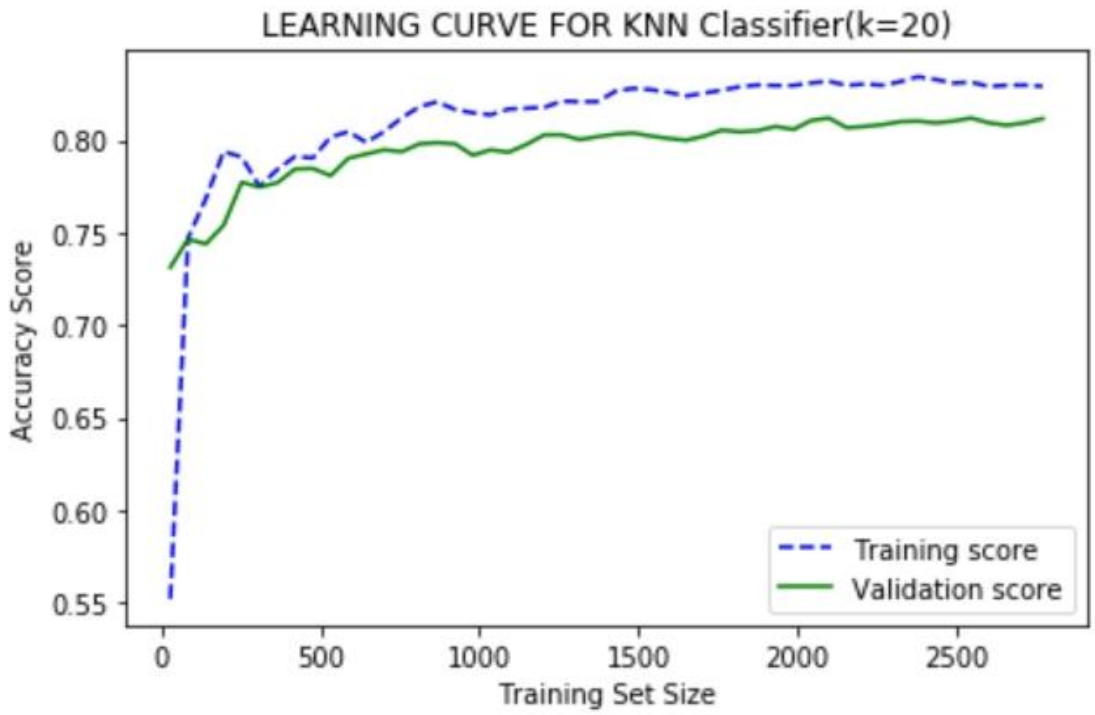

Fig. 19 learning curve for KNN classifier(k=20)

In Fig.19 I have increased the K parameter of the model to avoid the high variance that is observed in the previous model. In the new model, a good fit is observed, moreover the accuracy improved to 81%.

6.2 Decision Tree Confusion matrix

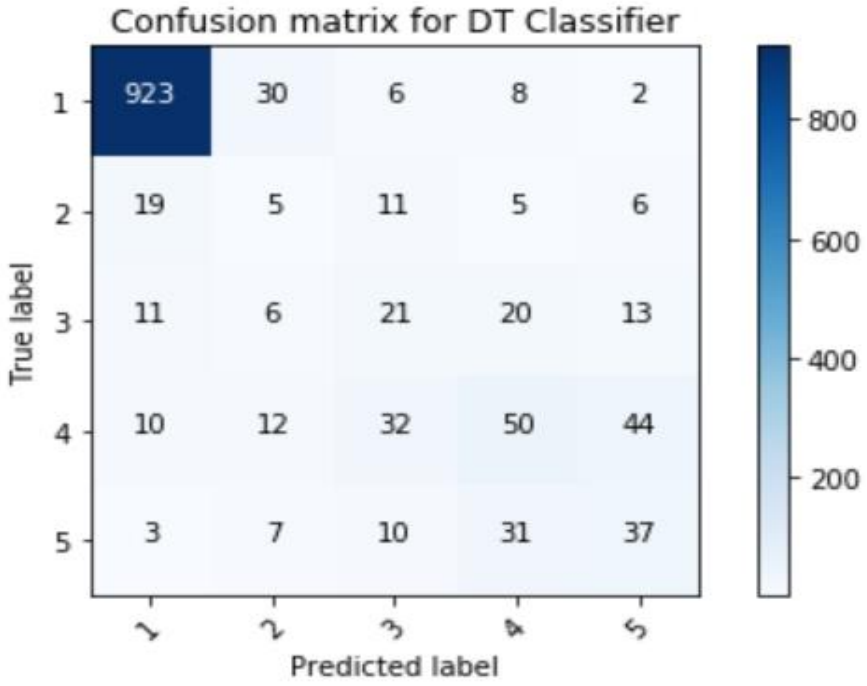

Fig.20 Confusion matrix for Decision Tree

Accuracy: - The accuracy for the Decision Tree classifier is found to be 78.13%, The score is lower than that for the KNN classifier.

	precision	recall	f1-score	support
1	0.95	0.96	0.95	960
2	0.04	0.04	0.04	48
3	0.22	0.22	0.22	76
4	0.44	0.42	0.43	142
5	0.35	0.34	0.35	96
accuracy			0.78	1322
macro avg	0.40	0.40	0.40	1322
weighted avg	0.78	0.78	0.78	1322

Fig. 21 Classification report for Decision Tree

6.2.1 Learning curves for Decision Tree

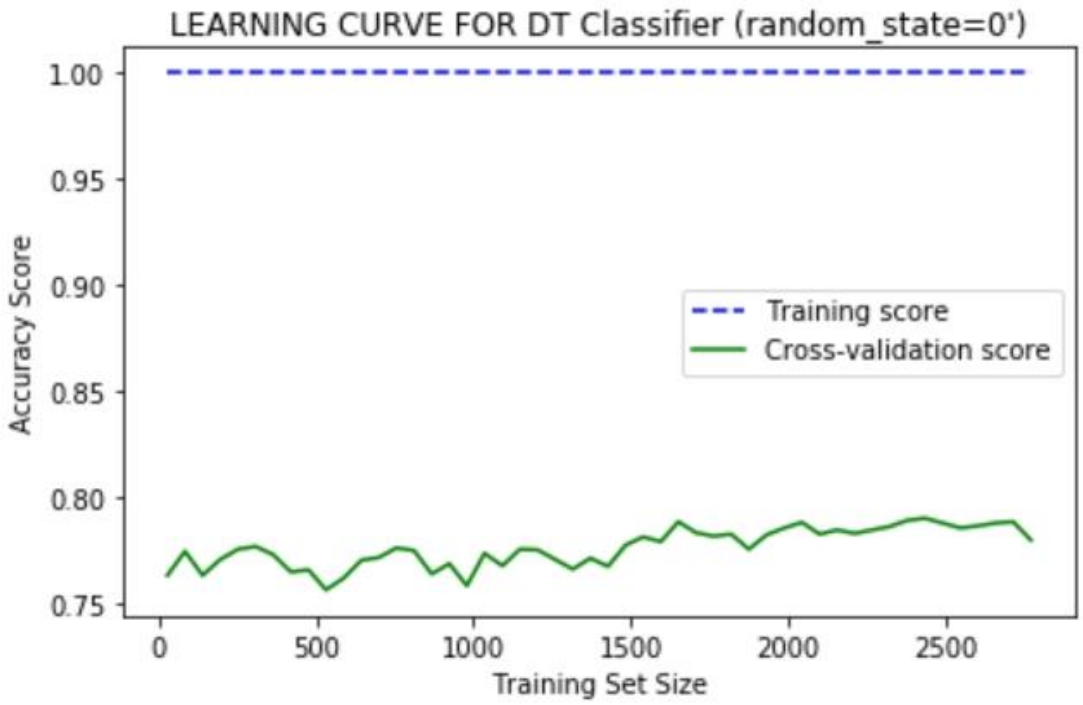

Fig. 22 learning curve for Decision Tree classifier(random_state=0)

In the decision tree model, the training data fits very well, and it shows no error. This is a sign of over-fitting. Moreover, there is huge gap between the training curve and validation curve which means the model has a very high variance.

Except in class 1, in every other class the precision value is higher than the recall value. However, the values are very low where we can conclude our model is not good in predicting the classes apart from class 1. For class 5 and 4 we see relatively higher values that for class 2 and 3 but are lower when compared to the KNN classifier.

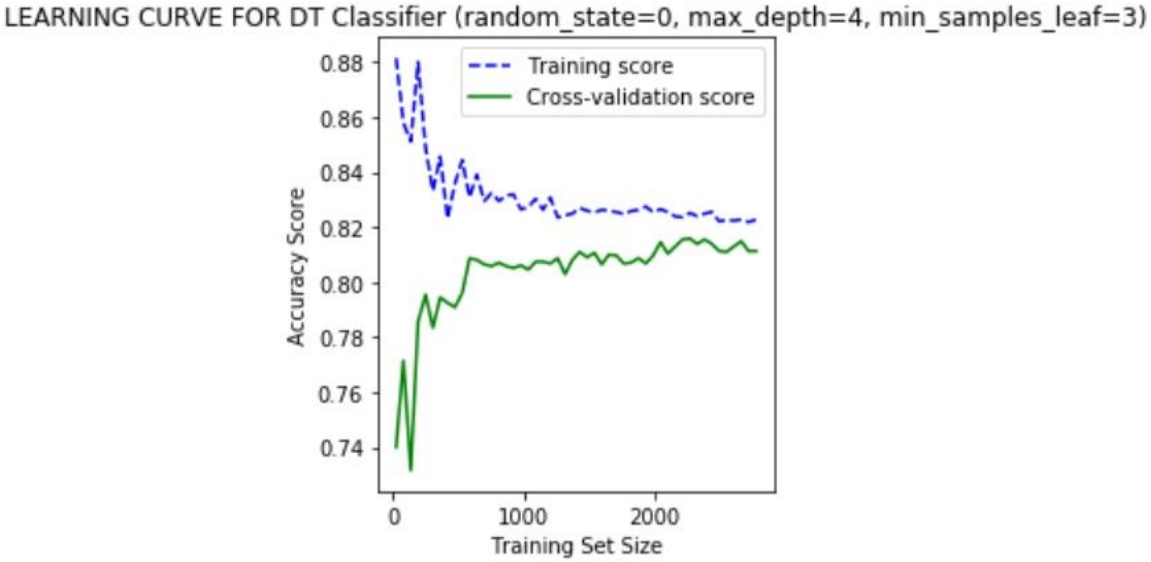

Fig. 23 learning curve for Decision Tree classifier (random_state=0, max_depth=4, min_samples_leaf=3)

Accuracy: By tweaking some parameters the model accuracy has improved from 78.13% to 80.86%.

From the learning curve, we see that we have managed to improve the model. The gap between the training curve and the validation curve is not very big nor is very narrow, which is a sign of a good-fit model.

6.3 Random Forest Confusion matrix

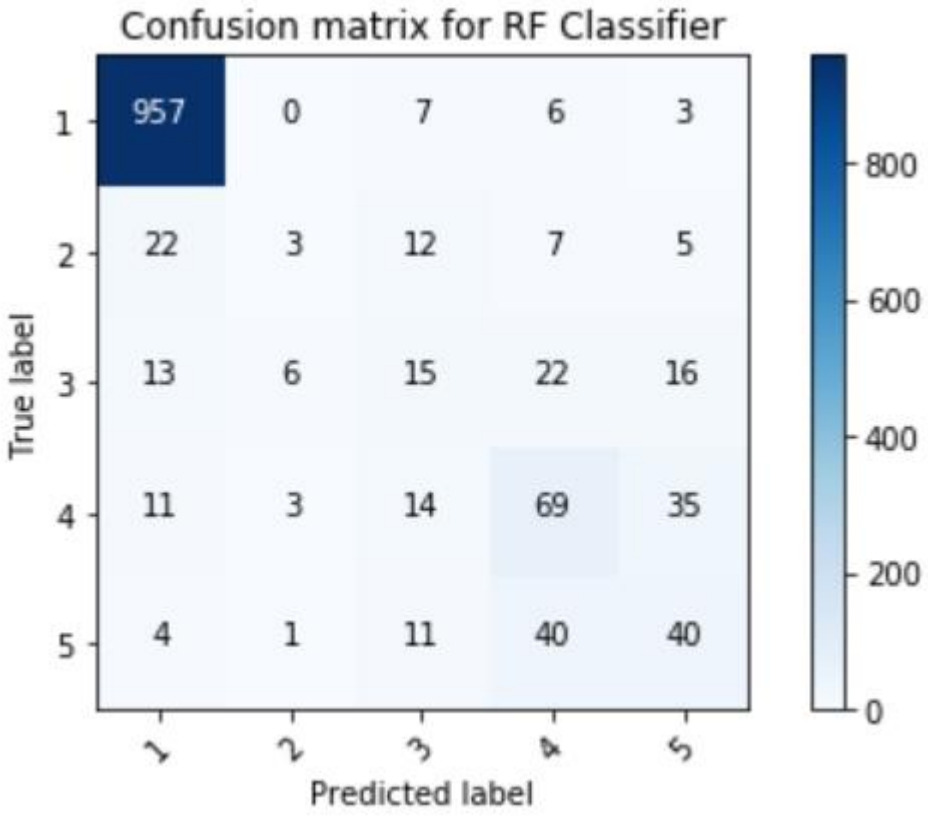

Fig. 24 Confusion Matrix for Random forest classifier

Accuracy:- The accuracy for the Random forest classifier(default parameters) is found to be 81.99%, The score is the highest of all the models tested so far.

	precision	recall	f1-score	support
1	0.95	0.98	0.97	973
2	0.23	0.06	0.10	49
3	0.25	0.21	0.23	72
4	0.48	0.52	0.50	132
5	0.40	0.42	0.41	96
accuracy			0.82	1322
macro avg	0.46	0.44	0.44	1322
weighted avg	0.80	0.82	0.81	1322

Fig. 25 Classification report for Random Forest (Default Model)

6.3.1 Learning curve for Random Forest

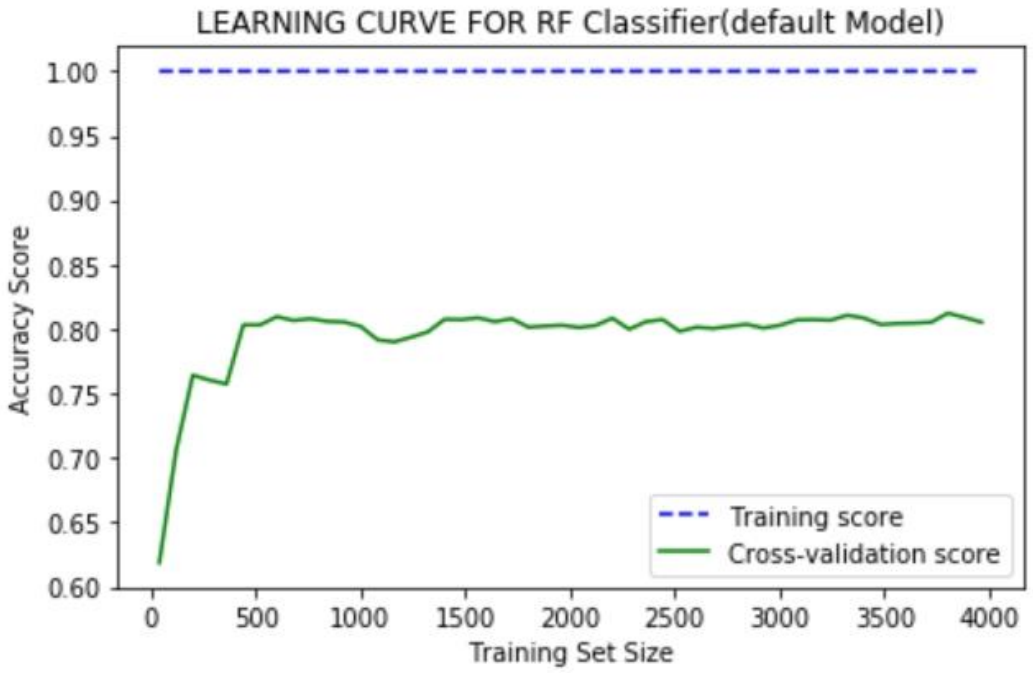

Fig. 26 learning curve for the random forest classifier using default hyper-parameters

The plot of the learning curve for the random forest (Fig.25) shows that the training set fits well and is with no error. The gap between the training curve and validation curve is huge indicating that our model suffers from high variance.

The recall values for class 1,4 and 5 are higher than the precision values. The precision values for class 2 and 3 are higher than the recall values. A higher recall values tells that the model is actually predicting True positives and less false negatives.

Although the accuracy seems to be higher than all the models I have tested, the model suffers from over-fitting which is an indication that our model is not the best.

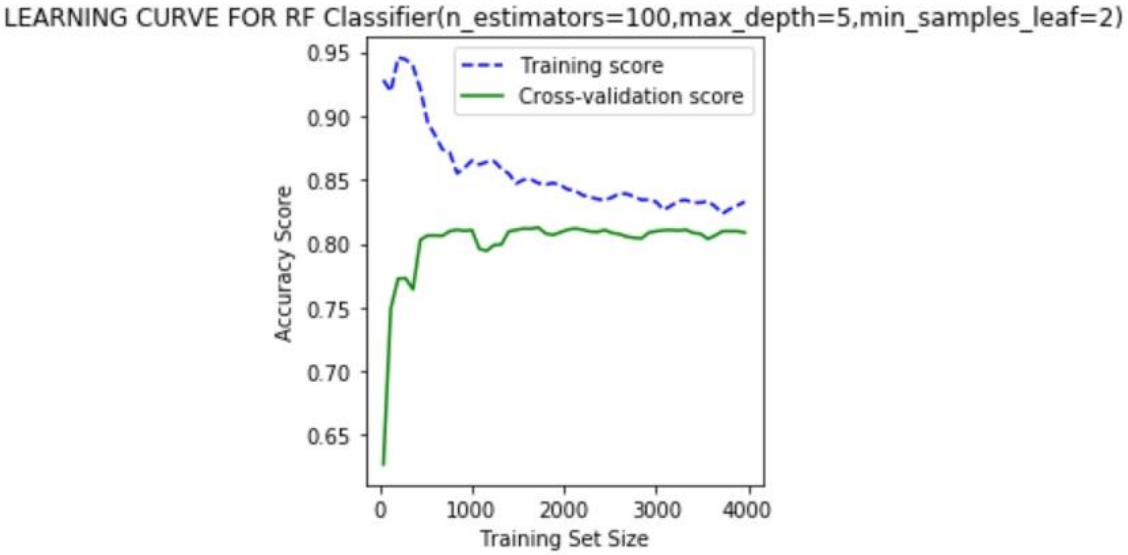

Fig. 27 RF classifier learning curve, good fit model

In order to fix the problem of over-fitting in the random forest model, I have tuned the following parameters.

n_estimator: to fix over-fitting we need to increase this parameter. If we decrease this parameter the random forest model becomes closer to decision tree. To plot the learning curve in Fig.26, I have used the default value 100.

max_depth: The default value for this parameter is none, in my model I have increased it to 5 so that the complexity of the model is reduced.

max_samples_leaf: This parameter also reduces the complexity of the model; hence I have set it to 2.

Accuracy: The accuracy of the RF model with some parameters being tuned to fix the over-fitting problem stays at 81%.

From Fig.26 we can conclude that with the chosen parameters the model has improved, we see the gap between the training curve and the validation curve is not very tight nor is very big which is a sign of good fit.

	precision	recall	f1-score	support
1	0.94	0.99	0.96	960
2	0.00	0.00	0.00	50
3	0.00	0.00	0.00	72
4	0.40	0.90	0.55	133
5	0.70	0.07	0.12	107
accuracy			0.81	1322
macro avg	0.41	0.39	0.33	1322
weighted avg	0.78	0.81	0.77	1322

Fig.28 RF classifier Confusion matrix report (good fitted model)

In Fig.26 the confusion matrix report shows that for classes 1 and 4 the recall has improved while the precision has only improved for class 5 only. From the report we observe that the model is unable to predict the positives for classes 2 and 3.

6.4 Support Vector Machine Confusion matrix

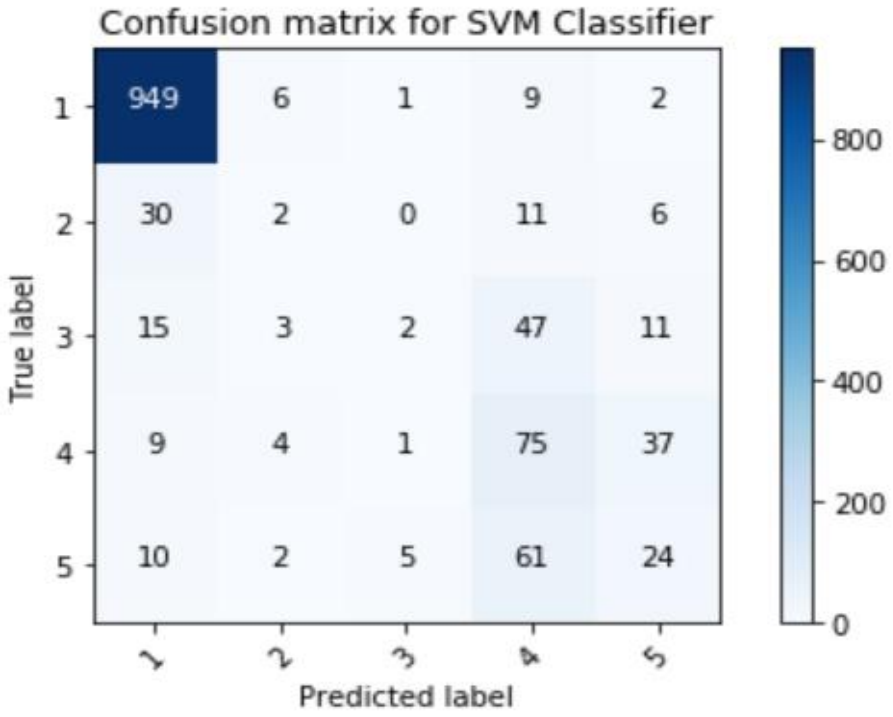

Fig. 29 Confusion matrix for SVM classifier (Default)

	precision	recall	f1-score	support
1	0.94	0.99	0.96	963
2	0.14	0.02	0.03	58
3	0.25	0.11	0.15	63
4	0.44	0.54	0.48	145
5	0.29	0.31	0.30	93
accuracy			0.80	1322
macro avg	0.41	0.39	0.39	1322
weighted avg	0.77	0.80	0.78	1322

Fig. 30 Confusion matrix report for SVM classifier (Default)

6.4.1 Learning curve for Support Vector Machine

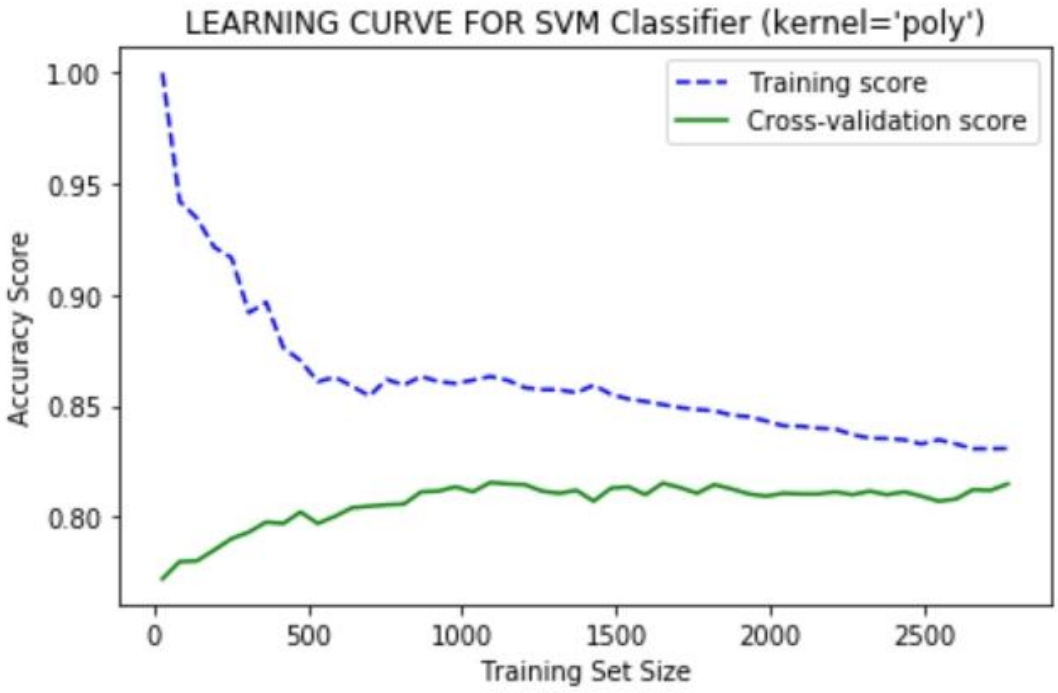

Fig.31 Learning curve for SVM classifier (Default)

Accuracy: The SVM classifier achieves accuracy 80.25%.

Recall value for class 1 is the highest of all the default models tested so far. It also has fairly high recall values for class 4, the remaining classes achieved recall value similar to the other default model.

The learning curve Fig. 31 shows some variance for the model. We see that the gap between the training data curve and the validation data curve is relatively fine, however there is room for improvement.

In Fig.32 the SVM model is improved by introducing C=0.1. C is the penalty parameter of the error term. It controls the trade-off between smooth decision boundary and classifying the training points correctly. By reducing the value of C it is possible to reduce variance. The gap between the training data curve and the validation data curve narrow than the previous plot, which is a sign of a good fit. Moreover, the accuracy has also improved a bit to 81.4%.

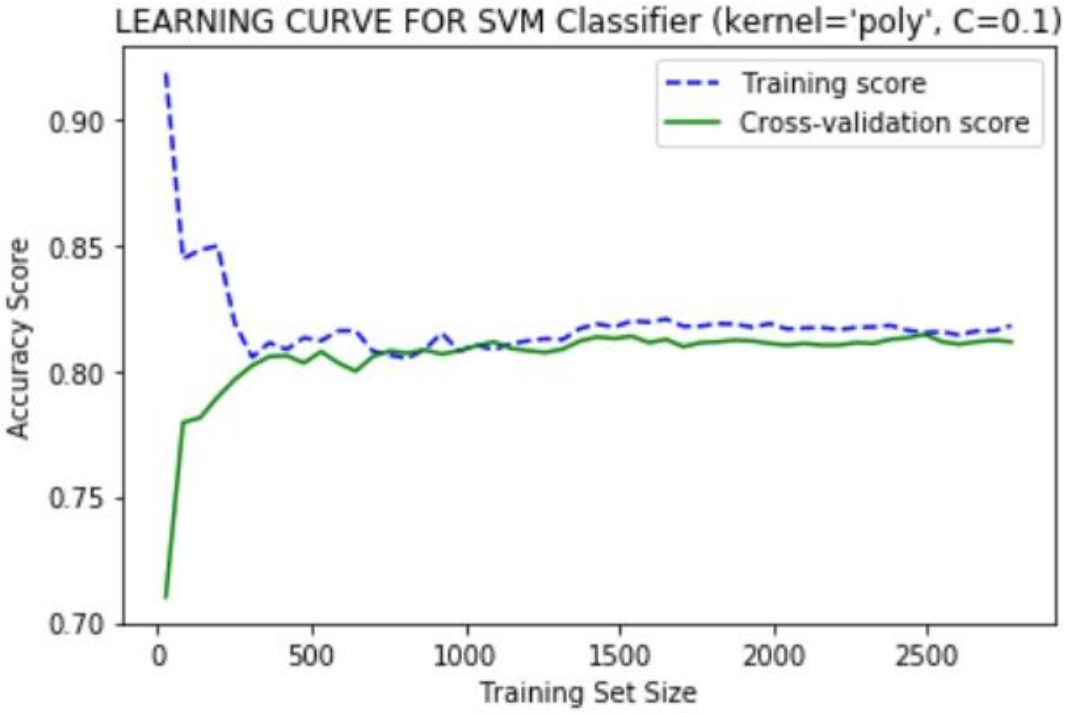

Fig. 32 Learning curve for SVM classifier(kernel='poly',C=0.1)

Chapter 7

Conclusion

In this thesis project, the aim was to estimate the number of occupants in the building (Arkivenshus) by using different features. Most of the features are energy consumption measurements, the number one feature contributing to the estimation of occupancy is electricity consumption in the building (light and power outlets) as it is shown in Fig.7, Fig.8 and Fig.9. Although there PIRs sensors in the building, the activation counts of these sensors doesn't give us the exact number of people who might be inside the building doing certain activities.

The occupancy data was collected by physical going to the building and counting number of people coming in and going out. Since the data was gathered during the pandemic period , where most of the people who use the building during working hours were working from home, hence the variation on the energy consumption may not be as is expected to explain how the energy consumption inside the building is related to the number of people inside the building. This phenomenon certainly has a negative impact on my analysis.

In this thesis I have tested four types of classification algorithms, KNN(k nearest neighbours), DT(Decision Tree), RF(Random Forest) and SVM(support vector machine).The accuracy in all the models tested lies between 78% - 82%. the highest value is found for the random forest (default) model, however this model suffers from high over-fitting, although I have tried to improve the over-fitting problem by introducing some parameters, the model still has a

problem in predicting classes 2 and 3.All the models are good in predicting class 1 which is rampant class in the data. However, except the SVM model all the models struggle to predict classes 2 and 3.

Out of the four models the SVM model with C=0.1 performs better considering the fact that it manages to predict classes 2 and 3, more over the explained variance and the accuracy score for this model is fairly okay when compared to the other models.

References

1- https://memoori.com/evolution-building-management-system-data-connectivity/

2- https://www.fierceelectronics.com/electronics/how-ai-will-turn-smart-buildings-into-smarty-pants

3- https://www.ibm.com/downloads/cas/2GYNP5R9

4- -https://uh-ir.tdl.org/bitstream/handle/10657/3299/KHALIL-DISSERTATION-2018.pdf?sequence=1&isAllowed=y

5- Ouf, M.M., Issa, M.H., Azzouz, A. and Sadick, A.M., 2017. Effectiveness of using WiFi technologies to detect and predict building occupancy. Sustainable buildings, 2, pp.1-10.

6- Dodier, R.H., Henze, G.P., Tiller, D.K. and Guo, X., 2006. Building occupancy detection through sensor belief networks. *Energy and buildings*, *38*(9), pp.1033-1043.

7- Yoshinaga, S., Shimada, A. and Taniguchi, R.I., 2010. Real-time people counting using blob descriptor. *Procedia-Social and Behavioral Sciences*, *2*(1), pp.143-152.

8- Ansanay-Alex, G., 2013, June. Estimating occupancy using indoor carbon dioxide concentrations only in an office building: a method and qualitative assessment. In REHVA World Congress on Energy efficient, smart and healthy buildings (CLIMA) (pp. 1-8).

9- Sankaranarayanan, A.C., Veeraraghavan, A. and Chellappa, R., 2008. Object detection, tracking and recognition for multiple smart cameras. Proceedings of the IEEE, 96(10), pp.1606-1624.

10- Calì, D., Matthes, P., Huchtemann, K., Streblow, R. and Müller, D., 2015. CO2 based occupancy detection algorithm: Experimental analysis and validation for office and residential buildings. Building and Environment, 86, pp.39-49.

11- Raykov, Y.P., Ozer, E., Dasika, G., Boukouvalas, A. and Little, M.A., 2016, September. Predicting room occupancy with a single passive infrared (PIR) sensor through behavior extraction. In Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing (pp. 1016-1027).

12- Mathews, E. and Poigné, A., 2009. Evaluation of a" smart" pedestrian counting system based on echo state networks. EURASIP Journal on Embedded Systems, 2009, pp.1-9.

13- Yun, J. and Lee, S.S., 2014. Human movement detection and identification using pyroelectric infrared sensors. Sensors, 14(5), pp.8057-8081.

14- Nienaber, F., Wolf, S., Wesseling, M., Cali, D., Müller, D. and Madsen, H., 2020. Validation, optimisation and comparison of carbon dioxide-based occupancy estimation algorithms. Indoor and Built Environment, 29(6), pp.820-834.

15- Franco, A. and Leccese, F., 2019. CO2 concentration and occupancy detection of educational buildings for energy efficiency purposes: an experimental analysis.

16- Wang, C., Jiang, J., Roth, T., Nguyen, C., Liu, Y. and Lee, H., 2021. Integrated sensor data processing for occupancy detection in residential buildings. Energy and Buildings, 237, p.110810.

17- Petersen, S., Pedersen, T.H., Nielsen, K.U. and Knudsen, M.D., 2016. Establishing an image-based ground truth for validation of sensor data-based room occupancy detection. Energy and Buildings, 130, pp.787-793.

18- Wang, J., Tse, N.C.F. and Chan, J.Y.C., 2019. Wi-Fi based occupancy detection in a complex indoor space under discontinuous wireless communication: A robust filtering based on event-triggered updating. Building and Environment, 151, pp.228-239.

19- Djenouri, D., Laidi, R., Djenouri, Y. and Balasingham, I., 2019. Machine learning for smart building applications: Review and taxonomy. ACM Computing Surveys (CSUR), 52(2), pp.1-36.

20- https://en.wikipedia.org/wiki/Scikit-learn

21- https://pypi.org/project/pandas/

22- https://pypi.org/project/numpy/

23- https://pypi.org/project/matplotlib/

24- https://builtin.com/data-science/step-step-explanation-principal-component-analysis

25- https://www.infoworld.com/article/3394399/machine-learning-algorithms-explained.html

26- https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

27- https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/

28- https://builtin.com/data-science/random-forest-algorithm

29- https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

30- https://datascienceplus.com/understanding-the-covariance-matrix/

31- https://ro-che.info/articles/2017-12-11-pca-explained-variance

32- https://bioturing.medium.com/how-to-read-pca-biplots-and-scree-plots-186246aae063

33- Biship, C.M., 2007. Pattern recognition and machine learning (information science and statistics).

34- https://machinelearningmastery.com/a-gentle-introduction-to-model-selection-for-machine-learning/

35- https://scikit-learn.org/stable/model_selection.html

36- https://neptune.ai/blog/the-ultimate-guide-to-evaluation-and-selection-of-models-in-machine-learning

37- Kohavi, R., 1995. *A study of cross-validation and bootstrap for accuracy estimation and model selection*. In Ijcai (Vol. 14, No. 2, pp. 1137-1145)

38- http://scott.fortmann-roe.com/docs/BiasVariance.html

39- https://www.dataquest.io/blog/learning-curves-machine-learning/

40- https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

41- https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

42- https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

Appendix

The data and the code used to perform the analysis are found in the following github link

https://github.com/yosiefha/Arkivenshus_analysis

all the csv files except the Occupant.csv are measurements from the building (Arkivenshus)

Occupant.csv file contains the number of people inside the building and their related classes.

The code used to analyse the data is on a Jupyter notebook file named as

Arkivens_hus_analysis.ipynb

Arkivenshus Logg Explanation.xlsx contains explanation about the measurements from the building.

Table.xlsx contains the weather data