TITLE OF THE PROJECT REPORT

A project report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

In

Department Of Computer Science and Engineering

By
Subhankar Halder (15101104044)
Sraboni Mondal (15101104021)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
JALPAIGURI GOVERNMENT ENGINEERING COLLEGE
JALPAIGURI, WEST BENGAL-735102.

FEBRUARY 2017

BONAFIDE CERTIFICATE

This is to certify that the project titled **PREDICT THE ENERGY EFFICIENCY** is a bonafide record of the work done by

Subhankar Halder (15101104044)
Sraboni Mondal (15101104021)

in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Department of Computer Science and Engineering** of the **JALPAIGURI GOVERNMENT ENGINEERING COLLEGE** during the year 2016-2017.

NAME	**DR. DIPAK KUMAR KOLE**
Guide	Head Of Department

Project Viva-voce held on ______________________________

Internal Examiner **External Examiner**

ABSTRACT

In this project, we present a python code that can predict the energy efficiency (Heating load and Cooling Load) of buildings. In order to select a prediction method, various regression methods were explored and compared. Boosted Decision Tree Regression method was chosen as our model due to its ability to learn non-linearity in data.

We performed energy analysis using 12 different building shapes simulated in Ecotect. The buildings differ with respect to the parameters such as Relative Compactness, Glazing Area and Glazing Area Distribution amongst other parameters. We simulate various settings as functions of the afore-mentioned characteristics to obtain 768 building shapes. The dataset comprises 768 samples and 8 features, aiming to predict two real valued responses.It is used as a Regression Problem. It can also be used as a multi-class classification problem if the response is rounded to the nearest integer.

The dataset contains eight attributes (or features) and two responses (or outcomes). The aim is to use the eight features to predict each of the two responses.

A Machine Learning algorithm is used to predict the two responses (or outcomes) by getting the input of the other 8 attributes (or features).

Keywords:- Boosted Decision Tree Regression: Relative Compactness; Glazing Area; Glazing Area Distribution; Regression; Machine Learning;

Acknowledgments

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organisations. I would like to extend my sincere thanks to all of them.

Foremost, we would like to express our sincere gratitude to our Head of the Department Dr. Dipak Kumar Kole for the continuous support, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of writing of this project report.

My sincere thanks also goes to my seniors Somnath Rakshit and Suvojit Manna for offering me the opportunity to work on such diverse exciting projects. Last but not the least, I would like to thank my family: my parents, for supporting me throughout my life. I am highly indebted to them for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

My thanks and appreciations also go to my year mates in developing the project and people who have willingly helped me out with their abilities.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 General.

The given problem falls under the domain of Machine learning.

Machine learning is the subfield of computer science that gives computers the ability to learn without being explicitly programmed. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms is infeasible; example applications include spam filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), search engines and computer vision.

In this Project, we use Energy efficiency regression dataset also available at
https://archive.ics.uci.edu/ml/machine-learning-databases/00242/ENB2012_data.xlsx

Variables Information:

1.1 Relative Compactness

1.2 Surface Area - m^2

1.3 Wall Area - m^2

1.4 Roof Area - m^2

1.5 Overall Height - m

1.6 Orientation - 2: North, 3: East, 4: South, 5: West

1.7 Glazing Area - 0%, 10%, 25%, 40% (of floor area)

1.8 Glazing Area Distribution (Variance) - 1: Uniform, 2: North, 3: East, 4: South, 5:West

1.9 Heating Load - kWh/m^2

1.10 Cooling Load – kWh/m^2

Machine learning is the study of algorithms that can learn from data and make predictions, by building a model from example inputs rather than following static instructions. These algorithms are typically classified into three categories:

1.1.1Supervised Learning

In this type of learning, the system is presented with example inputs and outputs, with the aim of producing a function that maps the inputs to outputs. Regression and classification problems are the two main classes of supervised learning.

i. **Regression**

In Regression, which is also a supervised problem, the outputs are continuous rather than discrete.

In general, Linear Regression is used but due to the nonlinearity in the Values in dataset, Boosted Decision Tree Regression is used.

Boosted Decision Tree Regression: - This module can be used to create an ensemble of regression trees using boosting. *Boosting* means that each tree is dependent on prior trees, and learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage.

ii. **Classification**

In Classification, inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".

1.1.2Unsupervised Learning

It is concerned with leaving the system to find a structure based on the inputs, hopefully finding hidden patterns. Examples include density estimation, dimensionality reduction and clustering.

i. **Density Estimation**

It finds the distribution of inputs in some space.

ii. **Dimensionality Reduction**

It simplifies inputs by mapping them into a lower-dimensional space. Topic modelling is a related problem, where a program is given a list of human language documents and is tasked to find out which documents cover similar topics.

iii. **Clustering**

In Clustering, a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

1.1.3 Reinforcement Learning

In Reinforcement learning is the study of how a system can learn and optimise its actions in an environment to maximise its rewards, such as training a robot to navigate a maze.

CHAPTER 2 LITERATURE REVIEW

2.1 About the Topic

There has been a considerable body of research on the topic of energy performance of buildings (EPB) recently due to growing concerns about energy waste and its perennial adverse impact on the environment [1], [2]. Moreover, buildings in European countries are legally bound to conform to appropriate minimum requirements regarding energy efficiency following the European Directive 2002/91/EC [1]. Reports suggest that building energy consumption has steadily increased over the past decades worldwide [3], [4], and heating, ventilation and air conditioning (HVAC), which have a catalytic role in regulating the indoor climate [5], account for most of the energy use in the buildings [6]. Therefore, one way to alleviate the ever-increasing demand for additional energy supply is to have more energy-efficient building designs with improved energy conservation properties. When it comes to efficient building design, the computation of the heating load (HL) and the cooling load (CL) is required to determine the specifications of the heating and cooling equipment needed to maintain comfortable indoor air conditions. In order to estimate the required cooling and heating capacities, architects and building designers need information about the characteristics of the building and of the conditioned space (for example occupancy and activity level), the climate, and the intended use (residential buildings have generally different requirements compared to industrial buildings). Building energy simulation tools are currently widely used to analyse or forecast building energy consumption, in order to facilitate the design and operation of energy efficient buildings since practice has shown that the results of the simulations can often accurately reflect actual measurements [7]. Simulation tools are used extensively across diverse disciplines because they enable experimentation with parameters that would otherwise be infeasible, or at least very difficult to control in practice [8]. In the context of building energy design for example, simulations could facilitate the comparison of identical buildings where only a single parameter is modified across a range of possible values to investigate its effects on some observed quantity of interest. For an overview and comparison of building simulation tools we refer to Yezioro et al. [7] and to Crawley et al. [9]. Using advanced dedicated building energy simulation software may provide reliable solutions to estimate the impact of building design alternatives; however this process can be very time-consuming and requires userexpertise in a particular program. Moreover, the accuracy of the estimated results may vary across different building simulation software [7]. Hence, in practice many researchers rely on machine learning tools to study the effect of various building parameters (e.g. compactness) on some variables of interest (e.g. energy) because this is easier and faster if a database of the required ranges of variables is available [10], [11], [2]. Using statistical and machine learning concepts has the distinct advantage that distilled expertise from other disciplines is brought in the EPB domain, and by using these techniques it is extremely fast to obtain answers by varying some building design parameters once a model has been adequately trained. Moreover, statistical analysis can enhance our understanding offering quantitative expressions of the factors that affect the quantity (or quantities) of interest that the building designer or architect may wish to focus on. Therefore, the integration of machine learning in EPB has sparked enormous interest lately. Various machine learning techniques such as polynomial regression [11], support vector machines (SVM) [10], [12] artificial neural networks (ANN) [13], [14], and decision trees [2] have been explored to predict various quantities of interest in the context of EPB. Machine learning tools have also been explicitly used in predicting HL and CL. Catalina et al.

[11] used polynomial regression (including up to quadratic terms) to predict monthly heating demand for residential buildings. They used as inputs for the regression model the building shape factor, the envelope U-value, the window-to-floor area ratio, the building time constant, and climate. Wan et al. [15] studied the impact of climate change on HL and CL for office buildings in China. Schiavon et al. [16] focused on the influence of raised floor, structure type, window-to-wall ratio and the presence of carpet to determine CL for different zones, and reported that orientation and the presence of carpet are the most important predictors. Li et al. [12] forecast hourly building CL based mainly on preceding environmental parameters. Of particular interest to this study, HL and CL have been associated with variables such as relative compactness (RC) [17], climate [15], surface area, wall area, and roof area [16], [17], orientation [16], [17], and glazing [17]. The rationale for studying these variables is that designers and engineers have found that they are correlated with energy performance, and HL and CL in particular. Many studies in the general research area of EPB have made rigid simplifying mathematical assumptions relying on linear correlations and classical least squares regression techniques, tools which are known to be ill-suited for many complicated applications where normality assumptions do not hold. Other studies have used complicated machine learning tools, but have failed to rigorously examine the available data (data mining), for example to report which variables are the most important for the particular problem addressed, thus failing to leverage on important information that can be inferred when using statistical tools. In this study, we investigate the effect of eight input variables: (RC), surface area, wall area, roof area, overall height, orientation, glazing area, and glazing area distribution, to determine the output variables HL and CL of residential buildings. Those eight variables have been frequently used in the EPB literature to study energy-related topics in buildings, and this study builds on the work of Pessenlehner and Mahdavi [17] who used those particular eight variables to investigate their effect on HL. We statistically formally explore the data, provide meticulous statistical analysis to gain important insight of the underlying properties of input and output variables, and use robust classical regression and state of the art nonlinear and nonparametric statistical machine learning tools (random forests) to map the input variables to HL and CL.

2.2 About the Datasets

Taking the elementary cube (3.5 × 3.5 × 3.5) we generated 12 building forms where each building form is composed of 18 elements (elementary cubes). The simulated buildings were generated using Ecotect. All the buildings have the same volume, which is 771.75 m3, but different surface areas and dimensions. The materials used for each of the 18 elements are the same for all building forms. The selection was made by the newest and most common materials in the building construction industry and by the lowest U-value. Specifically, we used the following building characteristics (the associated U-values appear in parenthesis): walls (1.780), floors (0.860), roofs (0.500), windows (2.260). The simulation assumes that the buildings are in Athens, Greece, residential with seven persons, and sedentary activity (70W). The internal design conditions were set as follows: clothing: 0.6 clo, humidity: 60%, air speed: 0.30 m/s, lighting level: 300 Lux. The internal gains were set to sensible (5) and latent (2 W/m^2), while the infiltration rate was set to 0.5 for air change rate with wind sensitivity 0.25 air changer per hour. For the thermal properties we used mixed mode with 95% efficiency, thermostat range 19 ºC – 24 ºC, with 15-20 hours of operation on weekdays and 10-20 hours on weekends. We used three types of glazing areas, which are expressed as percentages of the floor area: 10%, 25%, and 40%. Furthermore, five different distribution scenarios for each glazing area were simulated: 1) uniform: with 25% glazing on each side, 2) north: 55% on the north side and 15% on each of the other sides, 3) east: 55% on the east side and 15% on each of the other sides, 4) south: 55% on the south side and 15% on each of the other sides, and 5) west: 55% on the west side and 15% on each of the other sides. In addition, we obtained samples with no glazing areas. Finally, all shapes were rotated to face the four cardinal points. Thus, considering twelve building forms and three glazing area variations with five glazing area distributions each, for four orientations, we obtained $12 \times 3 \times 5 \times 4 = 720$ building samples. In addition, we considered twelve building forms for the four orientations without glazing. Therefore, in total we studied $12 \times 3 \times 5 \times 4 + 12 * 4 = 768$ buildings. Each of the 768 simulated buildings can be characterized by eight building parameters (to conform to standard mathematical notation and facilitate the analysis in this work, henceforth these building parameters will be called input variables and will be represented with X) which we are interested in exploring further. Also, for each of the 768 buildings we recorded HL (Heating Load) and CL (Cooling Load) (henceforth these parameters will be called output variables and will be represented with y). Table 1 summarizes the input variables and the output variables in this study, introduces the mathematical representation for each variable, and indicates the number of possible values. [18]

TABLE 2.1

Mathematical representation of the input and output variables to facilitate the presentation of the subsequent analysis and results

Mathematical Representation	Input or output variable	Number of possible values
X1	Relative Compactness	12
X2	Surface Area	12
X3	Wall Area	7
X4	Roof Area	4
X5	Overall Height	2
X6	Orientation	4
X7	Glazing Area	4
X8	Glazing Area Distribution	6
y1	Heating Load	586
y2	Cooling Load	636

Following the classical mathematical convention, we use X to denote input variables and y to denote output variables. Although 768 different buildings were simulated, in some cases the output variables of different buildings might coincide. The graph along with the probability densities for each variable are shown in figures below.

Figure 2.1 (Relative Compactness)

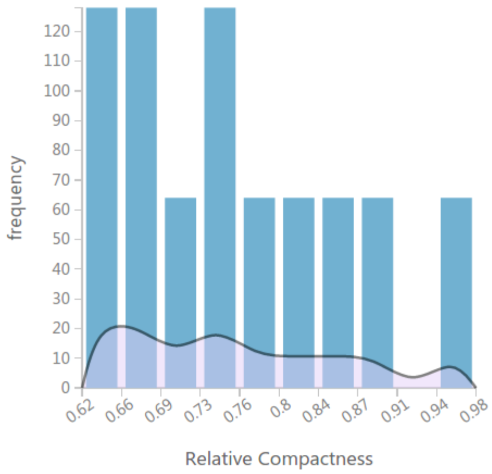


Figure 2.2 (Surface Area)

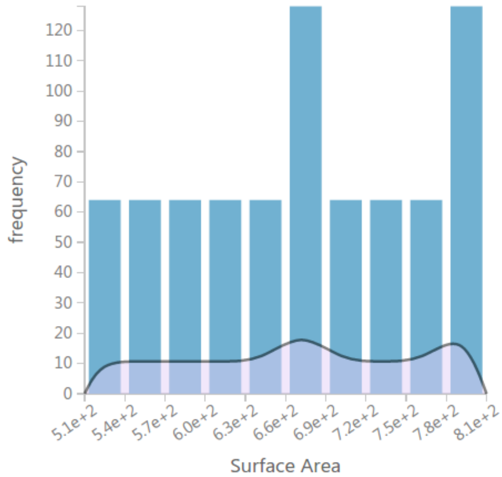


Figure 2.3 (Wall Area)

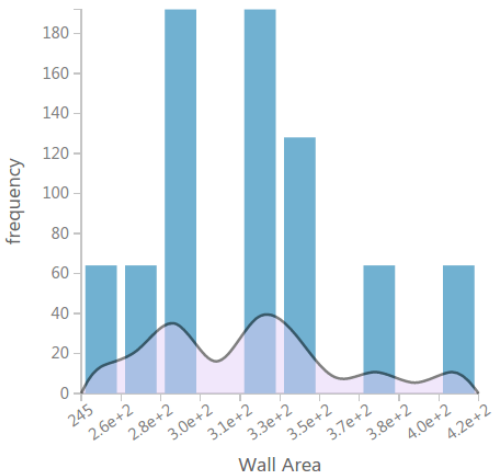


Figure 2.4 (Roof Area)

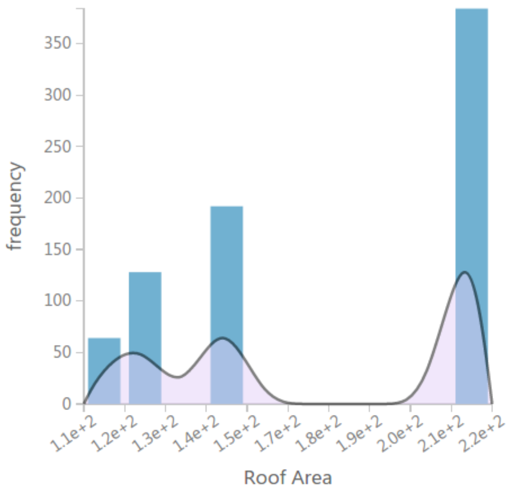


Figure 2.5 (Overall Height)

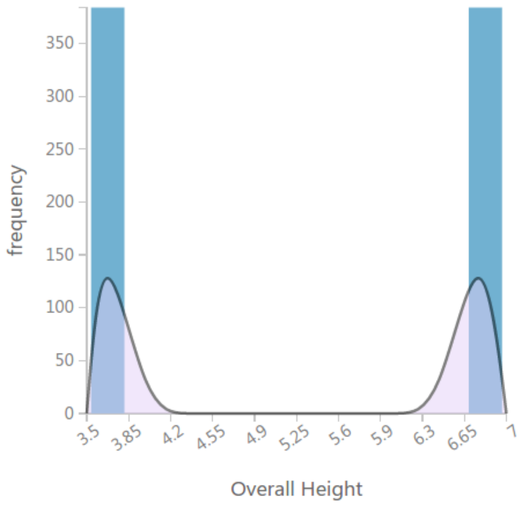


Figure 2.6 (Orientation)

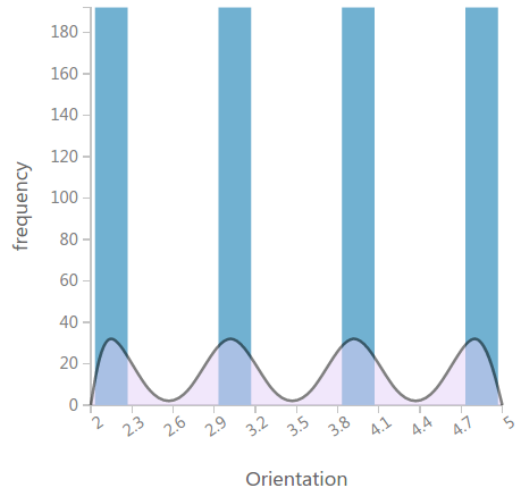


Figure 2.7 (Glazing Area Distribution)

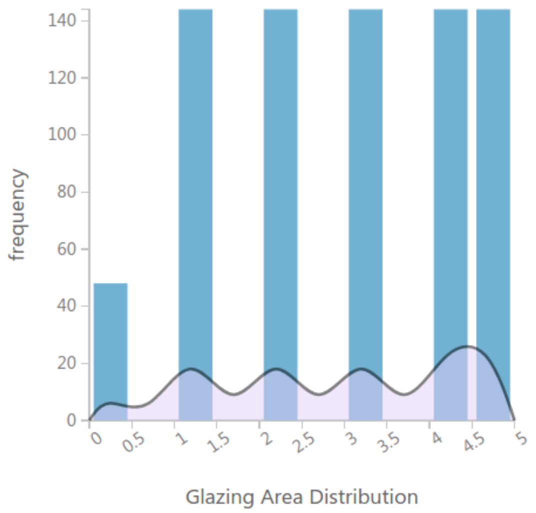


Figure 2.8 (Glazing Area)

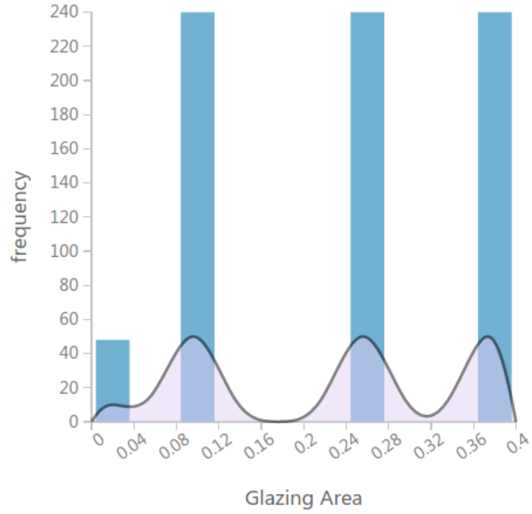


Figure 2.9 (**Heating Load)**

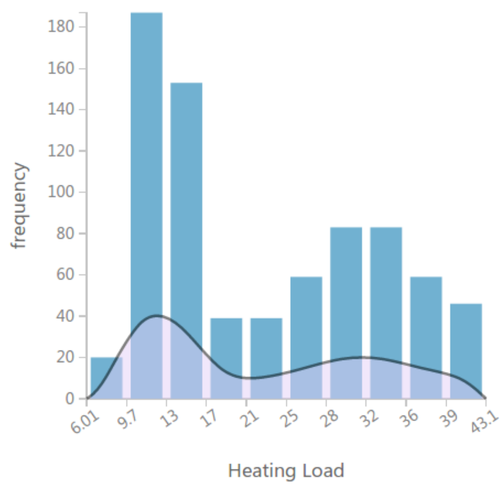
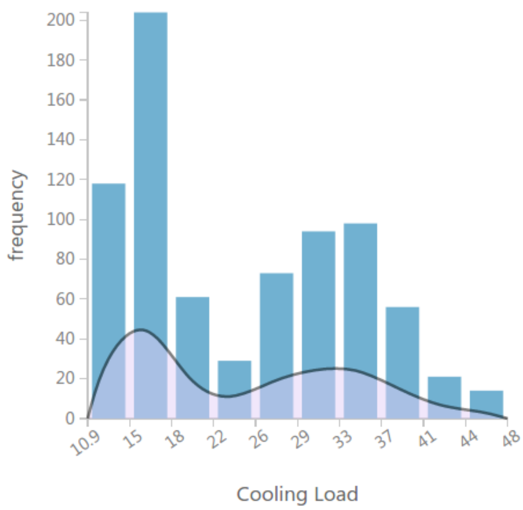


Figure 2.10 (**Cooling Load)**



CHAPTER 3 Methodology

3.1 Materials.

We will be using **Microsoft Azure Machine Learning Studio** to Predict the Energy Efficiency from a given Dataset.

In this project, we needed to predict two things: -

1. Heating Load
2. Cooling Load

Microsoft Azure Machine Learning Studio is a collaborative, drag-and-drop tool one can use to build, test, and deploy predictive analytics solutions on your data. Machine Learning Studio publishes models as web services that can easily be consumed by custom apps or BI tools such as Excel.

Azure Machine Learning Studio gives you an interactive, visual workspace to easily build, test, and iterate on a predictive analysis model. You drag-and-drop *datasets* and analysis *modules* onto an interactive canvas, connecting them together to form an *experiment*, which you run in Machine Learning Studio. To iterate on your model design, you edit the experiment, save a copy if desired, and run it again. When you're ready, you can convert your *training experiment* to a *predictive experiment*, and then publish it as a *web service* so that your model can be accessed by others.

As stated above that this project generates two outputs. Because of which two differents models are to be trained and a third experiment is to be created where the final code is generated.

3.2 Heating Load Model.

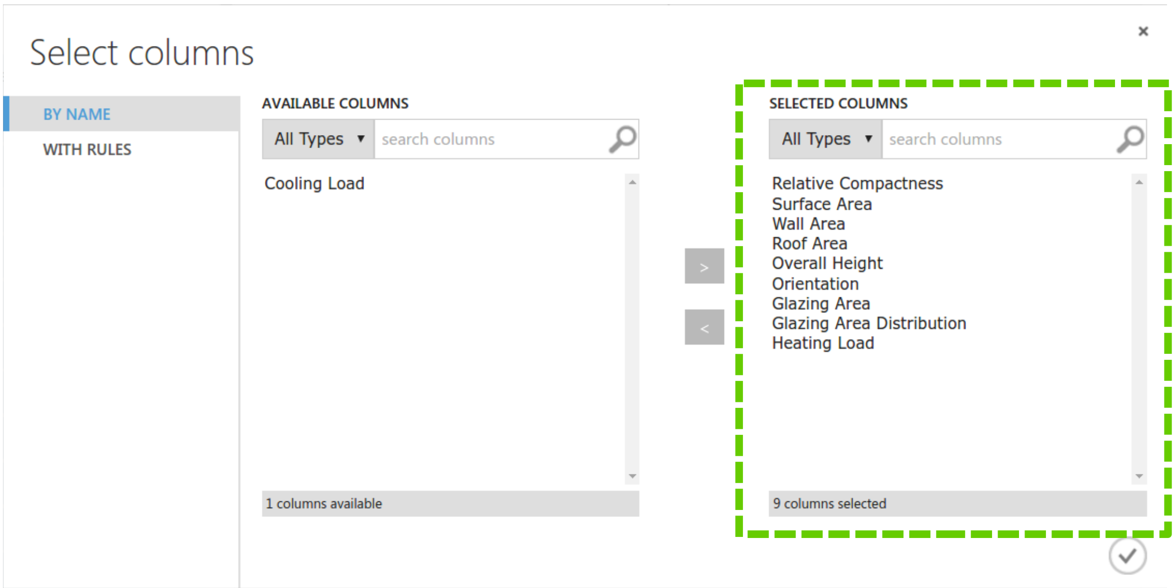
Steps for the first Trained Model:

1) First, we need to go to **Experiments** and create a New Experiment and choose the **Blank Experiment** Template. The experiment is presented as a Blank Canvas in the centre and has Components to choose from on the left side and Properties on the right side. We need to give the name of the experiment, at first. As this training model should contain the trained data for the Heating Load as output so let us name it 'Heat Load'.

2) To import data, we need to select one of the many sample datasets from **Saved Datasets Sample Data**. In our case, we will be using **'Energy Efficiency Regression Dataset'**. In it, a lot of columns of different data information's are given such as glazing area, the glazing area distribution, and orientation.

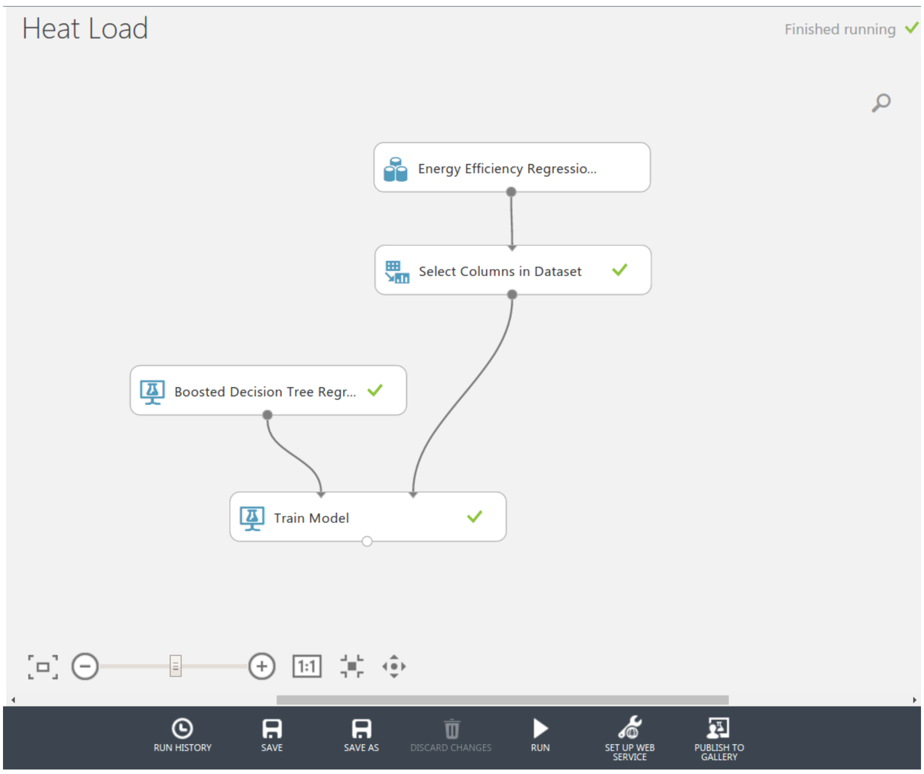
3) Then we need to select the **Select Columns in Dataset** and connect the input to the output of the **Energy Efficiency Regression Dataset**

The following columns were selected: -



4) Select **Boosted Decision Tree Regression** from the left panel.
5) Select **Train Model** from the left panel and connect the one input of it with the output of *Boosted Decision Tree Regression* and the other with the output of the *Select Columns in Dataset*.

The final model should look like this: -



6) Click on **Run.**

7) Right click on the *Train Model* and select *Trained Model → Save as Trained Model.*

That's it our first Trained model is ready for use. Now the second one is to be created i.e for Cooling Load

3.3 Cooling Load Model.

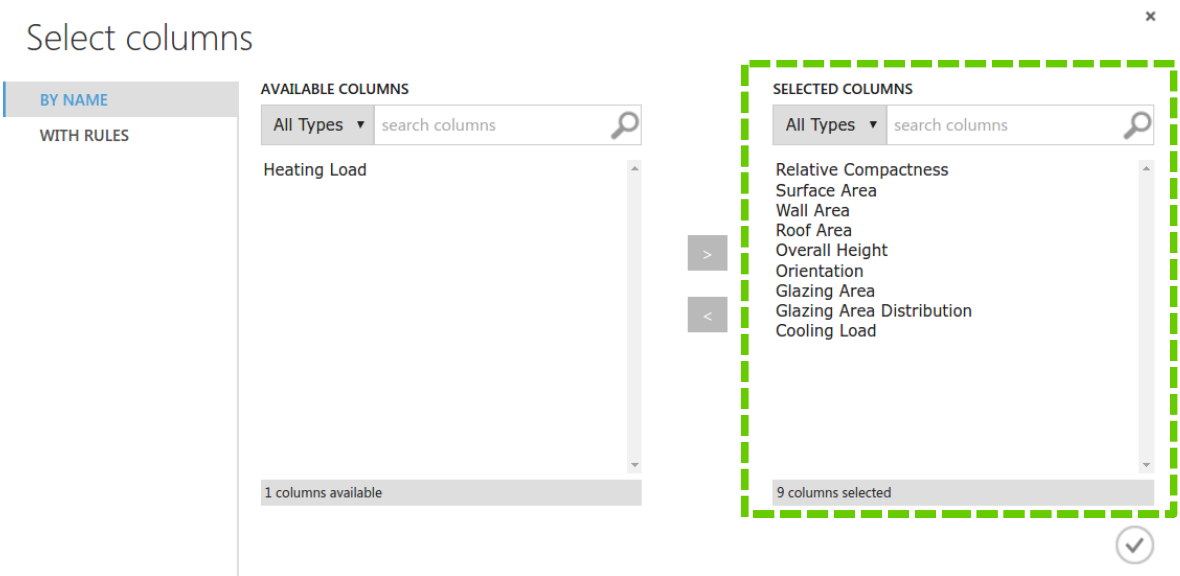
Steps to create the second Trained Model:

1) First, we need to go to **Experiments** and create a New Experiment and choose the **Blank Experiment** Template. The experiment is presented as a Blank Canvas in the centre and has Components to choose from on the left side and Properties on the right side. We need to give the name of the experiment, at first. As this training model should contain the trained data for the Cooling Load as output so let us name it 'Cool Load'.

2) To import data, we need to select one of the many sample datasets from **Saved Datasets Sample Data**. In our case, we will be using **'Energy Efficiency Regression Dataset'**. In it, a lot of columns of different data information's are given such as glazing area, the glazing area distribution, and orientation.

3) Then we need to select the **Select Columns in Dataset** and connect the input to the output of the **Energy Efficiency Regression Datase**t

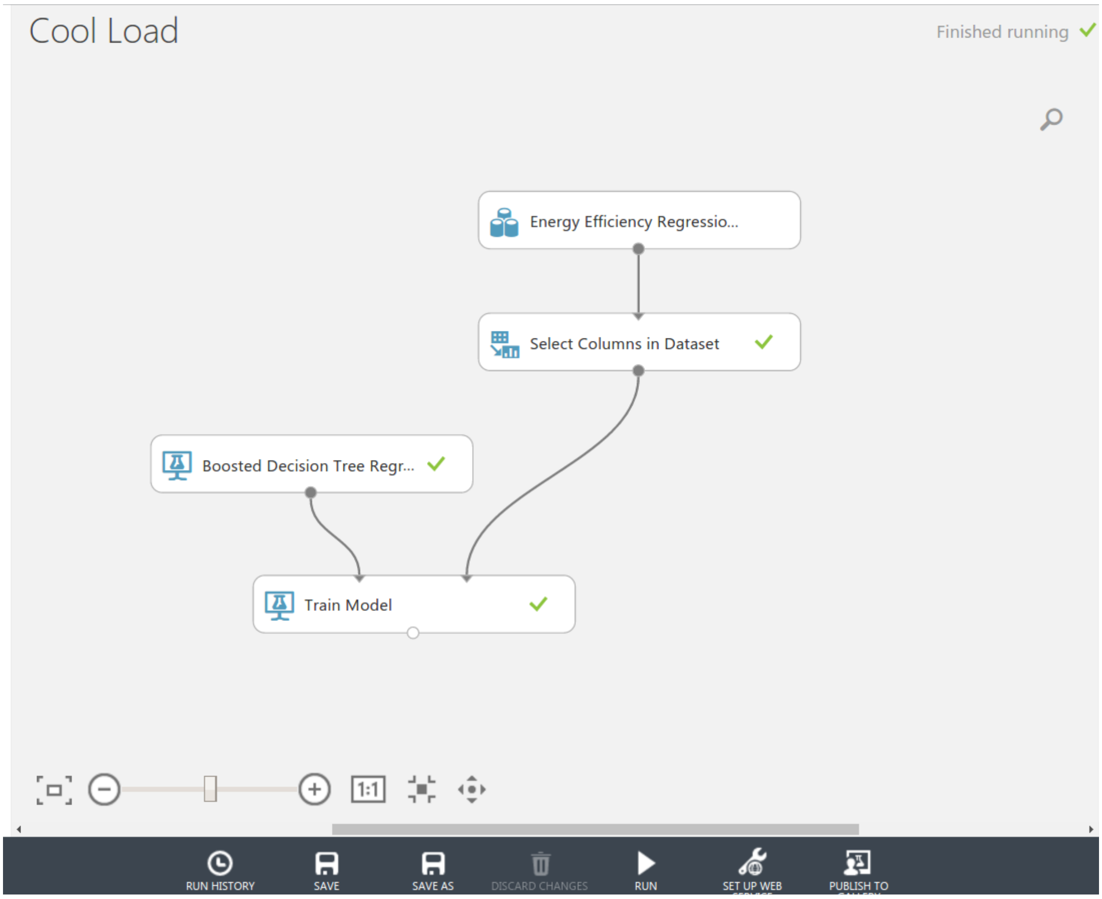
   The following columns are selected: -



4) Select **Boosted Decision Tree Regression** from the left panel.

5) Select **Train Model** from the left panel and connect the one input of it with the output of *Boosted Decision Tree Regression* and the other with the output of the *Select Columns in Dataset*.

The final model should look like this: -



6) Click on **Run.**

7) Right click on the *Train Model* and select *Trained Model* → *Save as Trained Model.*

Our Second Model is also completed.

3.4 Final Model.

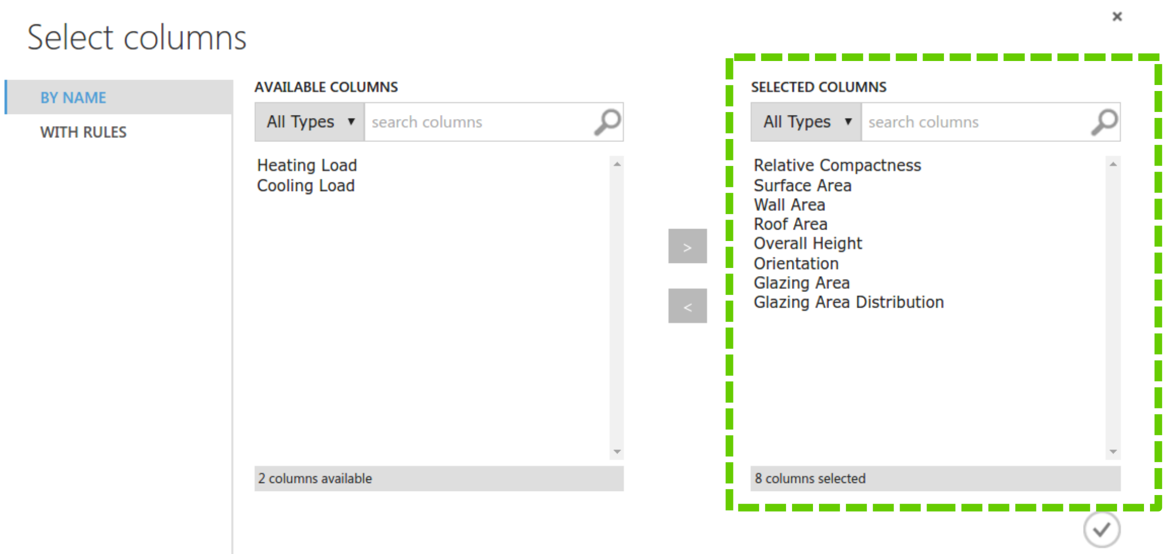
Steps to create the final Model: -

1) First, we need to go to **Experiments** and create a New Experiment and choose the **Blank Experiment** Template. The experiment is presented as a Blank Canvas in the centre and has Components to choose from on the left side and Properties on the right side. We need to give the name of the experiment, at first. Let us name it 'Final Experiment'

2) To import data, we need to select one of the many sample datasets from **Saved Datasets Sample Data**. In our case, we will be using **'Energy Efficiency Regression Dataset'**. In it, a lot of columns of different data information's are given such as glazing area, the glazing area distribution, and orientation.

3) Then we need to select the **Select Columns in Dataset** and connect the input to the output of the **Energy Efficiency Regression Dataset.**

The following columns should be selected: -



4) Now Select both the **Heat Load** model & **Cool Load** model (which we just created) from the left panel.

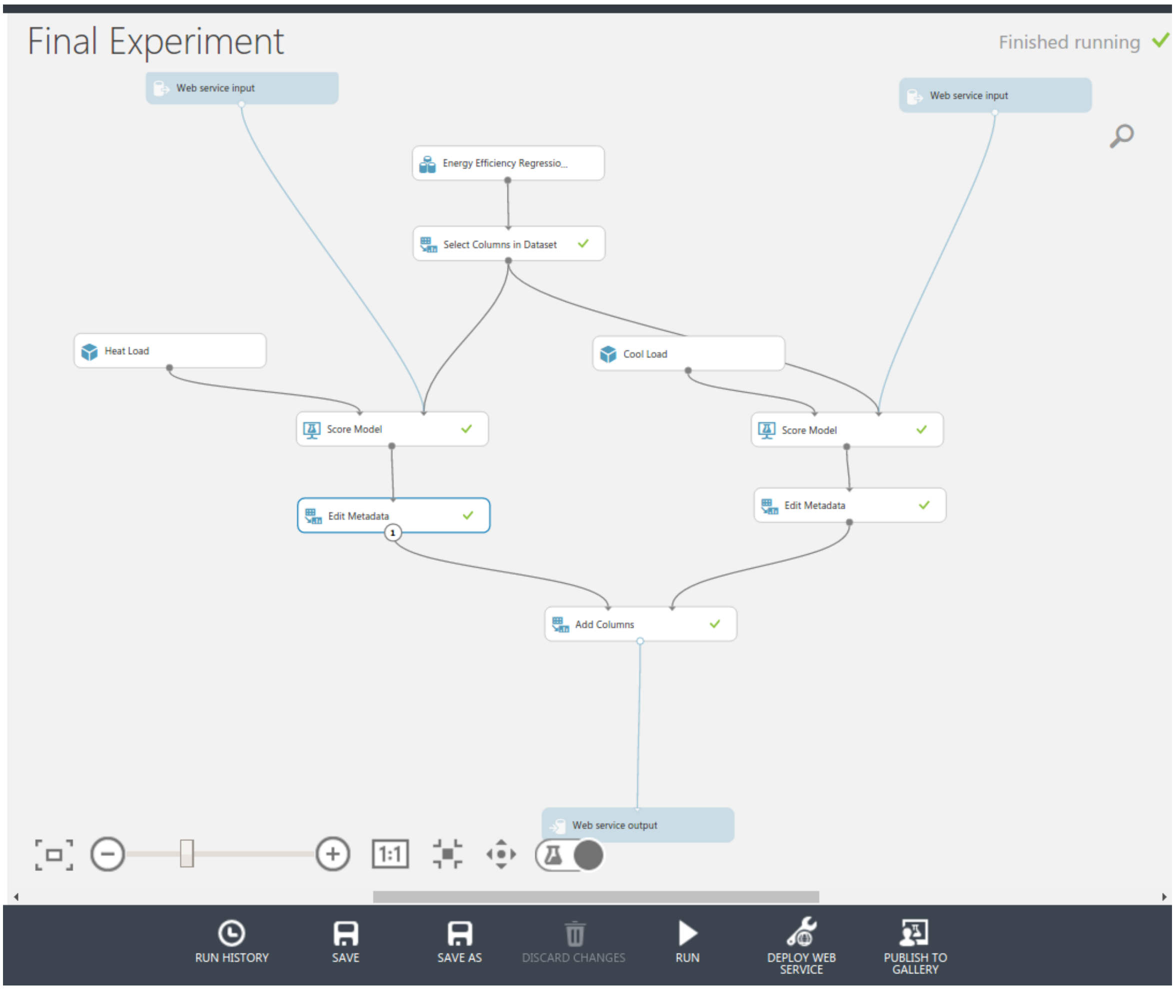
5) Select **Score Model** and duplicate it. Output of the *Heat Load* will be input for one of the *Score Model,* output of the *Cool Load* will be input for the other *Score Model.* The other input for the *Score Models* will be connected to the *Select Columns in Dataset.*

6) Select **Edit Metadata** from the left panel and duplicate it. Connect the input of each *Edit Metadata* with the output of their respective *Score Model* (Select all the columns in *Edit Metadata).*

7) Click on **Run** and the click on ***Set* up Web Services.**

8) Duplicate the **Web Services Input** and connect the output of each *Web Services Input* to the other input of *Score Model (i.e. which is connected to the Select Columns in Dataset).*

The final model should look like this: -



9) Click on **Run.**

10) Click on **Deploy Web Services.**

The Python Code and the API Key is now generated.

API key

e+u07brLqtluLi3UGnEEkHUhUzmspkKHl860TTLF2/Jp9BrAriGsjVcqJPzuXKb47qDKEHYdSZGtZFR8f1anIw==

3.5 Python Code.

After making the generated code user friendly, it is -

```
import urllib2
# If you are using Python 3+, import urllib instead of urllib2

import json

relative= raw_input("\nEnter the Relative Compactness: ")

surface= raw_input("\nEnter the Surface Area: ")

wall = raw_input("\nEnter the Wall Area: ")

roof = raw_input("\nEnter the Roof Area: ")

overall= raw_input("\nEnter the Overall Height: ")

orientation=raw_input("\nEnter Orientation: ")

glazing1= raw_input("\nEnter the Glazing Area: ")

glazing2= raw_input("\nEnter the Glazing Area Distribution: ")

data =  {

        "Inputs": {

                "input2":
                {
                    "ColumnNames": ["Relative Compactness", "Surface Area", "Wall Area", "Roof
Area", "Overall Height", "Orientation", "Glazing Area", "Glazing Area Distribution"],
                    "Values": [ [ relative, surface, wall, roof, overall, orientation, glazing1,
glazing2 ], ]
                },
                "input1":
                {
                    "ColumnNames": ["Relative Compactness", "Surface Area", "Wall Area", "Roof
Area", "Overall Height", "Orientation", "Glazing Area", "Glazing Area Distribution"],
                    "Values": [ [ "0", "0", "0", "0", "0", "0", "0", "0" ], ]
                },        },
            "GlobalParameters": {
}
    }
data["Inputs"]["input1"]=data["Inputs"]["input2"]

body = str.encode(json.dumps(data))
```

```
url =
'https://ussouthcentral.services.azureml.net/workspaces/d40075692ae645bf8c9108f6167c6b9
0/services/4f7e2a5c09b74bbd9a687cf3c6c8ab53/execute?api-version=2.0&details=true'

api_key =
'e+u07brLqtluLi3UGnEEkHUhUzmspkKHl860TTLF2/Jp9BrAriGsjVcqJPzuXKb47qDKEH
YdSZGtZFR8f1anIw=='
headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ api_key)}

req = urllib2.Request(url, body, headers)

try:
   response = urllib2.urlopen(req)

   # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
   # req = urllib.request.Request(url, body, headers)
   # response = urllib.request.urlopen(req)

   result = response.read()
   print(result)
except urllib2.HTTPError, error:
   print("The request failed with status code: " + str(error.code))

   # Print the headers - they include the requert ID and the timestamp, which are useful for
debugging the failure
   print(error.info())

   print(json.loads(error.read()))
```

4.1 General

To check the code, we gave the following input:

Relative Compactness	Surface Area	Wall Area	Roof Area	Overall Height	Orientation	Glazing Area	Glazing Area Distribution	Heating Load	Cooling Load
0.98	514.5	294	110.25	7	2	0	0	15.55	21.33

```
subhankar@subhankar:~/Desktop$ python Final\ Code.py
Enter the Relative Compactness: 0.98
Enter the Surface Area: 514.5
Enter the Wall Area: 294
Enter the Roof Area: 110.25
Enter the Overall Height: 7
Enter Orientation: 2
Enter the Glazing Area: 0
```

and the output was near about expected: -


```
{"Results":{"output1":{"type":"table","value":{"ColumnNames":["Relative Compactness","Surface Area","Wall Area","Roof Area","Overall Height","Orientation","Glazing Area","Glazing Area Distribution","Scored Labels","Relative Compactness (2)","Surface Area (2)","Wall Area (2)","Roof Area (2)","Overall Height (2)","Orientation (2)","Glazing Area (2)","Glazing Area Distribution (2)","Scored Labels (2)"],"ColumnTypes":["Double","Double","Double","Double","Double","Int32","Double","Int32","Double","Double","Double","Double","Double","Double","Int32","Double","Int32","Double"],"Values":[["0.98","514.5","294","110.25","7","2","0","0","16.0339088439941","0.98","514.5","294","110.25","7","2","0","0","21.6436157226563"]]}}}}
```


First rectangle box is the Heating Load (16.0339088439941) and the second Rectangle Box is the Cooling Load (21.6436157226563) which is approximately equal to the expected output.

CHAPTER 5 CONCLUSION

5.1 Conclusion

The project presents a successful approach to predict the energy efficiency that will occur by simply studying the dataset of energy efficiency rating and finding a relation between different categories of information on buildings by applying Machine Learning Algorithms (i.e. in this case Boosted Decision Tree Regression) and thus predicting the Heat Load by taking other information's as input.

The usage of Machine Learning to perform the project not only makes it more convenient and efficient but also decreases the complexity.
To improve the model performance further we could try to reduce the bias. To do we have to re-train the model with more training data. To get a model which could be used all year around, we have to include training data from a relevant subset of days throughout the. Another possible optimization is to include more places in the dataset. That would give the model more variables to use for prediction but then only one record to predict the value from.

5.2 Future Scope

The scope of Machine Learning is large. The variety of applications that Machine Learning supports includes search engines, image recognition, speech analysis, filtering tools, and robotics. The author of the article, Where Machine Learning Is Headed, predicts that in the coming year, the global community will witness a tremendous growth of smart apps, digital assistants, and main-stream use of Artificial Intelligence. Machine Learning will proliferate the mobile market and enter the territories of drones and self-driving cars. Gartner's Ian Bertram predicts that more domain-specific and Machine Learning-enabled technologies will emerge this year. Democratization of AI/machine learning will continue, according to Mark Koh. The demand for making algorithms more easily available will push vendors to offer many new Machine Learning tools. Though such canned products will be available in the market, the skills required to fine tune existing algorithms, tweak the data, and develop an advanced model will remain in demand.

References

[1] European Commission, Directive 2002/91/EC of the European Parliament and of the Council of 16th December 2002 on the energy performance of buildings, Official journal of the European Communities, L1/65 –L1/71, 04/01/2003.
[2] Z. Yu, F. Haghigrat, B.C.M. Fung, H. Yoshimo, A decision tree method for building energy demand modeling, Energy and Buildings, 42 (2010) 1637-1646
[3] L. Perez-Lombard, J. Ortiz, C. Pout, A review on buildings energy consumption information, Energy and Buildings 40 (3) (2008) 394–398.
[4] W.G. Cai, Y. Wu, Y. Zhong, H. Ren, China building energy consumption: situation, challenges and corresponding measures, Energy Policy 37 (6) (2009) 2054–2059.
[5] G. Platt, J. Li, R. Li, G. Poulton, G. James, J. Wall, Adaptive HVAC zone modelling for sustainable buildings, Energy and Buildings 42 (2010) 412-421
[6] R. Yao, B. Li and K. Steemers, Energy policy and standard for built environment in China, Renewable Energy 30 (2005) 1973–1988
[7] A. Yezioro, B. Dong, F. Leite, An applied artificial intelligence approach towards assessing building performance simulation tools, Energy and Buildings 40 (2008) 612-620
[8] A. Tsanas, J.Y. Goulermas, V. Vartela, D. Tsiapras, G. Theodorakis, A.C. Fisher and P. Sfirakis, The Windkessel model revisited: a qualitative analysis of the circulatory system, Medical Engineering and Physics, 31 (2009) 581-588
[9] D.B. Crawley, J.W. Hand, M. Kummert, B.T. Griffith, Contrasting the capabilities of building energy performance simulation programs, Building and Environment 43 (2008) 661-673
[10] B. Dong, C. Cao, S.E. Lee, Applying support vector machines to predict building energy consumption in tropical region, Energy and Buildings 37 (2005) 545-553
[11] T. Catalina, J. Virgone, E. Blanco, Development and validation of regression models to predict monthly heating demand for residential buildings, Energy and Buildings 40 (2008) 1825-1832
[12] Q. Li, Q. Meng, J. Cai, H. Yoshino, A. Mochida, Applying support vector machine to predict hourly cooling load in the building, Applied Energy86 (2009) 2249-2256 [13] J. Zhang, F. Haghighat, Development of artificial neural network based heat convection for thermal simulation of large rectangular crosssectional area earth-to-earth heat exchanges, Energy and Buildings 42 (4) (2010) 435–440.
[14] S.S.K. Kwok, R.K.K. Yuen, E.W.M. Lee, An intelligent approach to assessing the effect of building occupancy on building cooling load prediction, Building and Environment (2011), doi:10.1016/j.buildenv.2011.02.008

[15] K.K.W. Wan, D.H.W. Li, D. Liu, J.C. Lam, Future trends of building heating and cooling loads and energy consumption in different climates, Building and Environment 46 (2011) 223-234

[16] S. Schiavon, K.H. Lee, F. Bauman, T. Webster, Influence of raised floor on zone design cooling load in commercial buildings, Energy and Buildings 42 (2010) 1182-1191

[17] W. Pessenlehner, A. Mahdavi, A building morphology, transparency, and energy performance, Eighth international IBPSA conference proceedings, Eindhoven, Netherlands, (2003) 1025-1032

[18] : A. Tsanas, A. Xifara. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools', Energy and Buildings, Vol. 49, pp. 560-567

WEBSITES :

[1] https://azure.microsoft.com/en-us/services/machine-learning/

[2] https://www.coursera.org/learn/machine-learning

[3] https://en.wikipedia.org/wiki/Machine_learning

[4] https://rstudio-pubs-static.s3.amazonaws.com/244473_5d13955ea0fd4e5e9d376161b956e9dc.html

[5] https://archive.ics.uci.edu/ml/datasets/Energy+efficiency