Introducing a Machine Learning Algorithm in the Enterprise Data Centre

Recommendation Report

Prepared for the University of Victoria by:

Executive Summary

This report explores ways to reduce electricity consumption in the University of Victoria's Enterprise Data Centre. Specifically, it recommends implementing a machine learning algorithm designed to study the environmental and power usage data collected by the building, and make changes where the system could be more efficient. Since the data centre is one of the largest consumers of electricity on campus, even a moderate improvement in efficiency would result in a large amount of power saved. To investigate the topic we did online research to determine the best fit algorithm for a data centre, used an experiment conducted by Google to assess the potential of the algorithm, and coordinated with the energy manager David Adams to estimate the total power and economic savings. Afterwards, we discovered that the neural network machine learning algorithm is the best solution for the university's Data Center. We estimated that, after spending an extended period of time studying data that is already collected by the data centre, the building would save 10-15% of its total power consumption. In light of these findings and the relative small amount of risk involved, this report highly recommends implementation of a neural network in EDC2.

Table of Contents

Table of Figures

1. Client Background

This report presents an investigation into the feasibility of introducing a machine learning algorithm at the University of Victoria's Enterprise Data Centre (EDC2) in order to optimize and reduce its energy consumption. Based on our research, the report offers a recommended course of action on whether it would be viable to implement a particular algorithm, called a neural network, as a practical option for optimizing the Data Center operations.

According to the provided request for proposal, one of the goals of the University of Victoria's Sustainability Action Plan for Campus Operations is to "reduce campus electricity consumption intensity by 8% by 2019, relative to 2010 as the baseline year" [1]. The Problem Statement mentions that the Enterprise Data Centre is one of the largest energy consumers on campus, spending almost 4000 *MWh* of electricity in 2015 [2]. This report, therefore, aims at finding a solution to optimize power consumption at EDC2.

A Data Center, such as EDC2, is a facility that houses computers able to process large amount of data for various purposes. These computers generate large amounts of heat and require corresponding cooling systems in order to operate properly. Non-computational systems like cooling offer a promising target for reducing power consumption.

PROBLEM DEFINTITION: EDC2 incorporates various sensors to collect environmental and indoor data for monitoring and adjusting all the systems. Data collected from the past, however, is not currently utilized for statistical analysis to produce more efficient performance; no program is currently in place to enable data analysis that could be incorporated into optimizing efficiency. Because available data is not used for proper optimization, EDC2 operates under

potentially numerous inefficiencies and continues to consume more energy and cost more money than necessary.

To resolve this problem, our goal was to examine the feasibility of implementing a machine learning algorithm to manage non-computational systems. Implementing such an algorithm would meet a number of practical objectives. Machine learning allows computers to adjust to new data "without being explicitly programmed" [3]. According to one of the Google's blogs, every Data Center is unique in its implementation with very complex equipment interactions and slow adaptation to both internal and external changes; and machine learning can address these complications [4]. The only constraint imposed on our potential design solution is that any costs incurred should be recoverable through achieved energy savings within a five-year period.

As we envisioned it, potential benefits of this design solution would include maximizing power efficiency at EDC2 and reducing overall electricity consumption on campus, saving up to $50,000 a year, recouping the investment within 5 years, which is the requirement for this project.

2. Methods

To investigate the feasibility of applying a machine learning algorithm to the Data Center, we researched the various types of learning algorithms; coordinated with the energy manager, David Adams; and conducted a case study of Google's implementation of a machine learning algorithm. The methods are outlined in detail below.

2.1 Research

To study and determine the best applicable machine learning algorithm, each team member researched a different type of algorithm. After we filtered out several insignificant ones, we presented our summary document in our second weekly meeting for further study.

2.2 Email Correspondence

By email correspondence David Adams provided the current energy consumption of EDC2 and the monthly electricity expenditure report for 2015. In our third weekly meeting, we used this information to calculate the potential savings and benefits of implementing a machine learning algorithm.

2.3 Case Study

In the second week, after preliminary research was done, the decision was made to study Google's machine learning algorithm to examine how much energy machine learning algorithms can save. The Google algorithm was chosen because the nature of the project is similar to that of EDC2.

3. Results

3.1 EDC2's total power consumption

David Adams' "Uvic Monthly Electricity Consumption for 2015 by Building" [5] declares a yearly electricity consumption of 4.2 million kilowatt-hours for EDC2. The monthly breakdown can be seen below in Figure 3.1.1. This Figure suggests that the current way that EDC2 manages its electricity does not have much change in power consumption throughout the

four seasons. A machine learning algorithm would study the data through each season and determine where, by changing protocol, the system could become more efficient.

Table 3.1.1: Monthly Electricity Consumption for EDC2 in kilowatt-hours [5]

Month in 2015	Power consumed in kWh	Month in 2015	Power consumed in kWh
January	325,200	July	324,000
February	336,000	August	336,000
March	304,800	September	315,800
April	331,200	October	314,400
May	326,400	November	337,200
June	302,400	December	337,200

3.2 EDC2's power usage efficiency

A data centre's power usage efficiency (PUE) is defined as the ratio between the building's total power consumption and IT equipment power consumption. Thus, the formula for a data centre's PUE is $\frac{Total\ power}{IT\ equipment\ power}$. The ideal PUE is one that is close to 1. This would indicate that the IT equipment is the only consumer of electricity in the building. Section 10.9 of the Integrated Energy Masterplan discloses that in 2011 the power usage efficiency of EDC2 was 2 [6]. A PUE of 2 indicates that the non-computational systems in EDC2 use as much power as the IT equipment.

In his article "Data Center energy metric: Power Usage Effectiveness (PUE)," Donald Beatty reports the US Environmental Protection Agency's (EPA) prediction for average data centre PUE in 2007-2011 (see Appendix 3.1.2). The EPA predicted that by 2011 the average data centre's PUE would be 1.9 and best practice systems would have a PUE of 1.5 or lower.

Assuming EDC2 has maintained the average trends, the EPA predicts that some systems could achieve a PUE as low as 1.2. As such, there is certainly room for improvement.

3.3 The machine learning algorithm specification

According to the Microsoft Machine Learning Cheat Sheet (Appendix 3.2.1), a supervised, regression neural network has been identified as the best fit algorithm for EDC2. In order to predict the power usage efficiency, which is a continuous and singular value with linear independent input variables, the flow chart recommends to use a neural network.

A neural network is a self-learning computation model inspired by how human brains solve problems. They are especially powerful in recognizing patterns and relationships within data. A supervised, regression neural network will independently discover a mathematical model which accurately predicts how one or more output values will be influenced by input data.

A neural network is composed of three or more layers used to manipulate the input data: an input layer, a number of hidden layers, and an output layer with one or more output variables. Referring to Figure 3.3.1, each layer contains a series of circles, which can be referred to as nodes or neurons. The red nodes are the input layer, the blue nodes are the hidden layer, and the yellow node is the output layer. Each layer of nodes are connected by arrows to each node in the next sequential layer. The arrows create pathways to the next node, and are commonly called edges or axions.

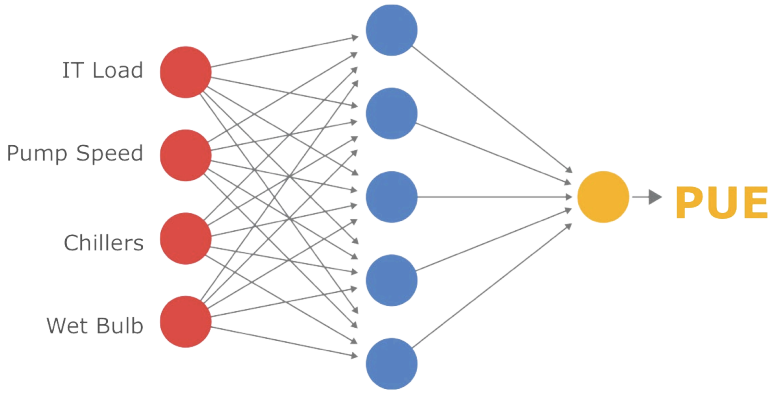


Figure 3.3.1: A simplified model of Google's Data Center neural network [7]

Once data is provided to the constructed neural network model, the supervised learning algorithm repeatedly performs the following steps:

1. *Forward Propagation*

Beginning as a randomized value, a weight is applied to each edge. The input variables are multiplied by the weight, and passed through the edge. Each node in the hidden layer has multiple incoming edges. The values passed through each incoming edge are summed together and an activation function is applied at each node. Using a nonlinear activation function allows the neural network to discover nonlinear relationships between the input values. The process of forward propagation is repeated through each hidden layer, until the final output layer. The values provided in the output layers are estimations of the output value [9].

2. *Backward Propagation*

The predictions given from the output layer will be compared to the real expected value. The amount of error in the prediction is analyzed with a cost function. The process of backwards propagation transforms our predictions from being random to extremely accurate.

2.1 *Gradient Descent*

Gradient descent is used to minimize the error, without having to try every possible weight. The cost function can be represented as a function of the weights and activation functions used to predict the output value, and the expected output value. Using partial derivatives, the rate of change of the cost function is determined in respect to the each weight. If the partial derivative of the cost function evaluated at a certain weight reaches zero, a local minimum is found for that particular weight. To summarize, gradient descent, using calculus, determines which direction to adjust each weight, incrementally applies the appropriate adjustment, and discovers at what point the weight minimizes the error. Each iteration of backwards propagation will provide a more accurate prediction of the PUE [9].

After repeating forward propagation, gradient descent, and backward propagation the model will converge on an accurate prediction of a mathematical model of how output variable is influenced by the input variables.

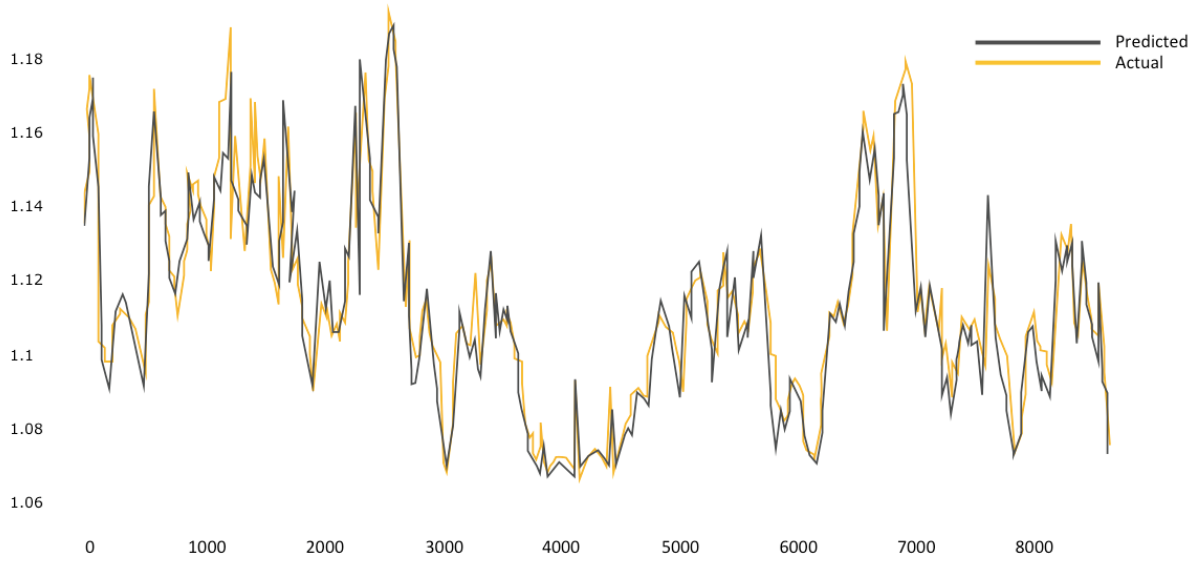


Figure 3.3.2: Google's algorithm predicting PUE with 99.6 percent accuracy [7]

Input variables

For the algorithm to run optimally, it is important for our input variables to be linearly independent from each other [8]. The list of input variables for this algorithm would be compiled from the following categories:

1. Energy drawn by the servers
2. Number of cooling equipment (pumps, towers, chillers, dry coolers etc.) running
3. Speeds of the water pumps
4. Temperature of water in cooling equipment
5. Outside environmental factors

The list is a simplified version of the input variables used in Google's neural network. (Appendix 3.3.2) The final list of input variables would be the common variables between both Data Centers as well as any unique equipment in EDC2. According to the *Integrated Energy Masterplan* in 2011 [6], EDC2 had chosen to reimplement their cooling systems with water cooled chillers, therefore, variables related to water cooling equipment would be applicable.

Output variable

The output variable is the power usage effectiveness (PUE). Our neural network will be able to predict accurately how the PUE will be affected by any changes in our equipment setpoints, outside environment, or spikes in server usage.

Further details

The more layers a neural network has the "deeper" and more accurate the model will learn. However, with each layer, the time the algorithm needs to train grows dramatically. The optimal number of layers should be determined in conjunction with technical staff at EDC2 and

the programming team on the project. Similarly, more research will be required to determine the number of neurons for each layer, the activation function, cost function, and other parameters.

3.4 The benefits of implementing a machine learning algorithm

A machine learning algorithm could contribute to lowering the PUE ratio by studying the energy data throughout the year and making energy protocol changes to fine tune the electricity consumption in the building. For example, Google, using their machine learning algorithm, was able to temporarily tweak their cooling systems when taking some servers offline for a few days. Normally, having to take servers offline would negatively impact the Data Centers efficiency, however, they were able to reduce the impact of this change with machine learning [7].

Google's machine learning experiment produced a decrease in total power consumption of 15% [8]. Similar results of a 10-15% reduction of the total 4.2 million kWh used by EDC2 would culminate in a reduction of 420,000-630,000 kWh annually. Figure 3.4.1 illustrates how dramatic the improvement was for Google's PUE.

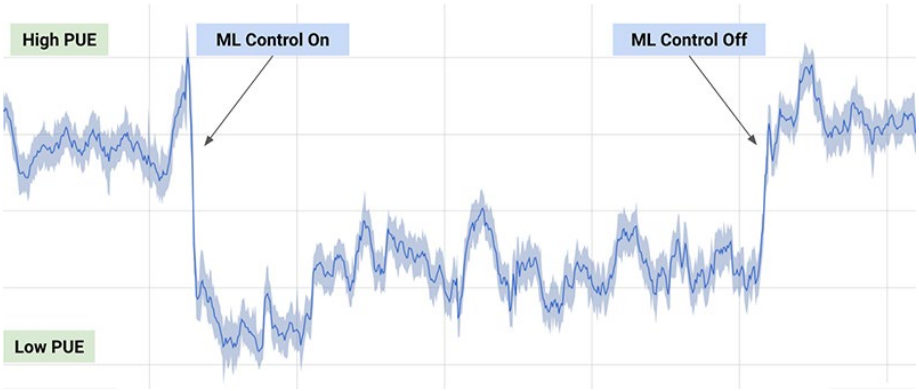


Figure 3.4.1 The effects of Google's machine learning algorithm on the PUE of their data centre [4]

In addition to the environmental benefits, reducing electricity consumption has an economic benefit for the university. At the rate of $0.08 per kilowatt-hour quoted by David

Adams [2], a 10-15% reduction of total power consumption in EDC2 would achieve a cost savings of $33,600 to $50,400 per year. Accounting for a six month development time costing $50,000 for an employee contract and a one year training period, the university can expect to recoup its investment by the end of two and a half years.

3.5 Implementation of the machine learning algorithm

While we found no open source machine learning algorithms available to simply plug into the data centre, there are numerous open source and free tools available to assist in constructing these programs. For example, Google's algorithm is proprietary, however they have created a machine learning API called TensorFlow that is open source and freely available. Other libraries and APIs are available for programming languages such as Python and R. Appendix 3.4.1 is a table comparing different open source and proprietary machine learning platforms.

Choosing between writing this algorithm in R or Python is a matter of personal preference. Both languages are commonly used in creating machine learning algorithms and provide numerous libraries to assist in doing so. While R is a language tailored specifically to machine learning and statistics, ultimately Python was chosen as it is much more commonly used by computer scientists. This will prove useful when interviewing potential software engineers since more programmers with machine learning experience are likely to be proficient with Python. Furthermore, Python is equipped with plenty of relevant tools such as NumPy and SciPy which are helpful in preprocessing the large amount of data to be analysed.

Perhaps the largest benefit of choosing Python is its compatibility with TensorFlow: the API developed by Google and used in their successful machine learning data centre project [10]. TensorFlow is a free,open source, machine learning development tool that automates much of

the more technical, complex realities of work with machine learning. Using TensorFlow would significantly simplify all of the procedures—forward propagation, backward propagation, gradient descent, and activation functions—detailed in the previous section.

4. Conclusion

In summary, our plan is to implement a machine learning algorithm with the task of studying the power consumption data already collected by EDC2, attempting to optimize the efficiency, and ultimately reducing the overall power consumption by 10-15%. A machine learning algorithm is a program designed specifically to study a data set and make predictions of the desired output value. With this information, the computer can effectively make changes to the input data (in this case non-computational electricity consumers) to achieve a more desirable output value (the power usage efficiency). The algorithm works by taking the set of inputs and assigning each element to a node in an input layer. The input nodes are connected to nodes in a hidden layer by pathways associated with a weight. Using mathematical concepts in machine learning, the algorithm changes the input according to the weights in such a way that their sums are the desired output and stores the values in the hidden layer. Over time, the algorithm will learn how to produce the correct output by studying how the different inputs are related by changing the weights connecting the input to the output. Eventually, a mathematical model of the relationship between the power consuming systems and the efficiency of the power consumption in the building is constructed. This report estimates that a machine learning algorithm would have the potential to reduce electricity consumption in EDC2 by up to 630,000 kWh, saving up to $50,000 per year after a 30 month recoupment period.

5. Recommendation

5.1 Implementation

Considering that this project will meet the 5 year payback requirement, implementation is highly recommended. First and foremost the interested party should interview to find an experienced programmer in concepts of machine learning to implement the neural network at EDC2. This programmer is key to the success of the project, so the interviewer should be thorough.

Secondly, to implement the neural network, data about the operation of EDC2 from past years should be identified and accumulated and provided to the implementing programmer. David Adams stated after the progress report presentation that data should be available as far back as 5 years. Technical staff at EDC2 overseeing this project should discuss the input variables with the implementing programmer.

Once the neural network has reached completion, it should be left to complete its training time. One full year of training time would be best for this program. This would allow the neural network to experience all the seasons of the year. This detail is important since cooling is a huge contributor to power draw in Data Centers.

The final step should be letting the neural network designed for EDC2 make changes. This would be wise to do supervised by its original creator, and during a weekend or any other less busy day just in case it does not perform as expected.

5.2 Consequences of not acting on recommendations

If UVic does not implement a neural network to optimize their Data Center, they should consider making drastic improvements some other way. In 2011, a PUE of 2.0 was considered average, but trends are continuing to decrease . UVic is no longer competing to achieve the modern standards for efficiency. Using machine learning is becoming best practice for modern Data Centers. Neglecting to follow would be controversial to Uvic's sustainability goals.

5.3 Further research

First of all, training time was determined as a group, but has not been researched specifically in detail. This decision would be best determined by the implementing programmer due to their further expertise. Secondly, the workload of this project is undetermined. Other studies were not very open about programming teams or production time. This means that additional programmers or longer creation time might be necessary. Finally, it would be prudent to research the potential applicability of a neural network in other buildings on campus for climate control.

5.4 Benefits

This project is expected to pay itself back in 2.5 years ($50,000 investment in a software engineer for six months, one year training period, saving up to $50,000 per year) which falls within the necessary 5-year payback period. The algorithm should not require further maintenance or changes, meaning that it has effectively zero upkeep costs. Savings in energy consumption will only improve in subsequent years. This project is a smart economic investment for UVic and a big step towards achieving environmental sustainability, and becoming leaders for a greener future.

6. References

[1] D. Adams, "Request for Proposals", 2016.

[2] D. Adams, "ENGR 240 – Problem Statement", University of Victoria, 2016.

[3] M. Rouse. (2016, Feb). *Machine Learning* [Online]. Available: http://whatis.techtarget.com/definition/machine-learning

[4] R. Evans, J. Gao. (2016, Jul 20). *DeepMind AI reduces energy used for cooling Google Data Centers by 40%* [Online]. Available: https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/

[5] D. Adams, "UVic Monthly Energy Consumption for 2015 by Building", University of Victoria, 2016

[6] Office of Campus Planning and Sustainability, *Integrated Energy Masterplan*, University of Victoria, 2011

[7] J. Kava, "Better Data Centers through machine learning", *Official Google Blog*, 2014. [Online]. Available: https://googleblog.blogspot.ca/2014/05/better-data-centers-through-machine.html
[Accessed: 01- Dec- 2016].

[8] J. Gao, *Machine Learning Applications for Data Center Optimization*, Google, 2014.

[9] *Neural Networks Demystified*. Youtube: Welch Labs, 2014. Available: https://youtu.be/bxe2T-V8XRs?list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

[10] Louridas & Ebert. "Machine Learning" [Online] *IEEE Softw.* Vol. 83,. No. 5, pp.110-115. Sept.-Oct. 2016

[11] Beaty, Donald. "Data Center energy metric: Power Usage Effectiveness (PUE)" [Online] ASHRAE Journal. Vol. 55.1, p. 61. Jan 2013.

[12] "Machine Learning Algorithm Cheat Sheet". *Microsoft Azure*. Microsoft., 2016. [Accessed: 02-Dec. 2016].

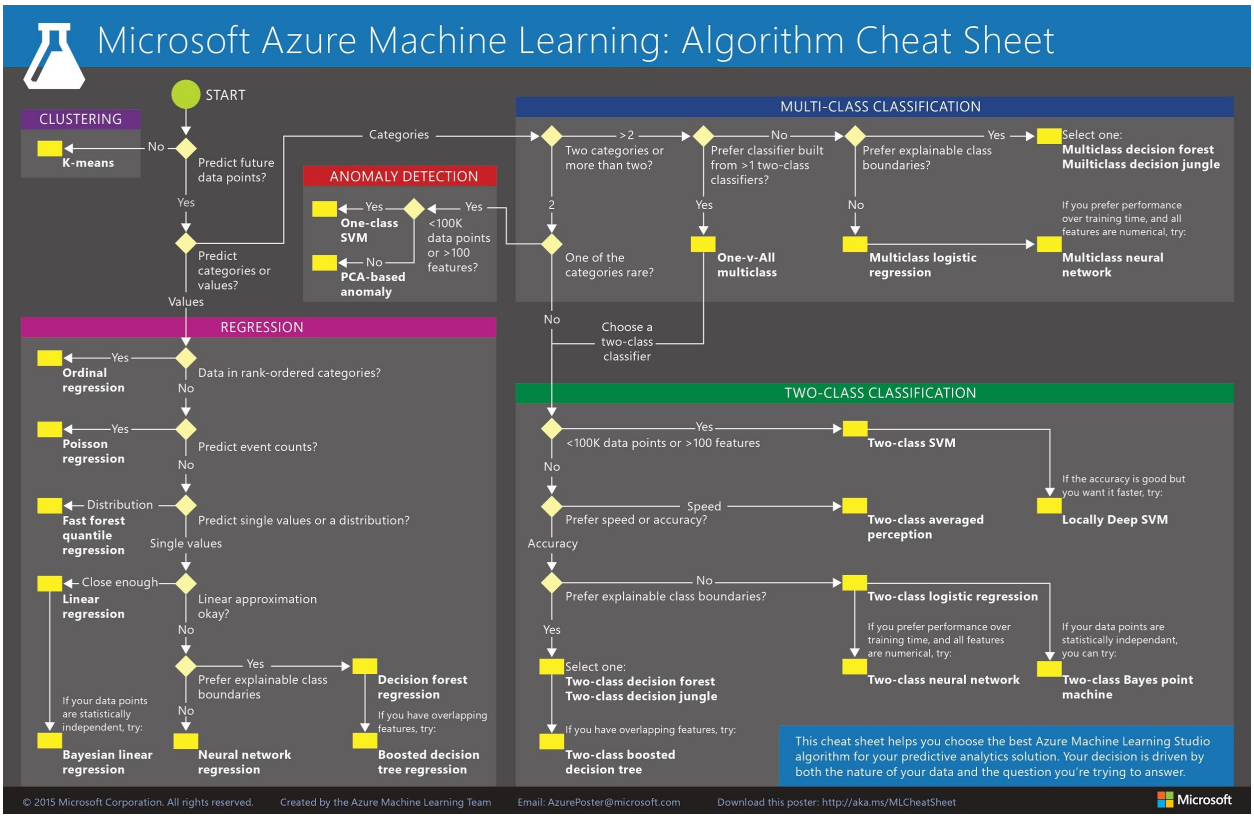
7. Glossary

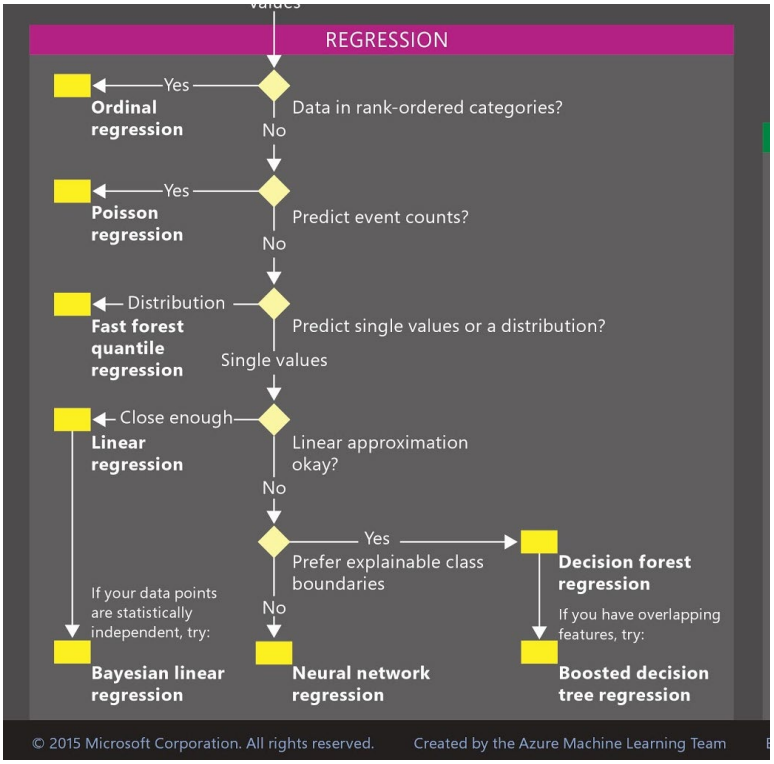
Activation function	An activation function is applied at a node and determines the output of the node, based on the inputs.
Algorithm	A set of instructions or computations to solve a certain problem, typically performed on a computer.
API	Set of tools and protocols used in assisting the construction of computer software.
Continuous	A continuous variable exists on the interval from negative infinity to infinity and can include any amount of decimal points
EDC2	UVic's Enterprise Data Centre 2 building
Gradient descent	An optimization algorithm which finds a local minimum of a function
IT equipment	All parts of the computational system. Excludes the cooling systems
Linearly independent	Variables are linearly independent if they cannot be expressed in a function with sums and constant coefficients.
Local minimum	Smallest value of a function within a certain range
Partial derivative	A partial derivative of a function in respect to one variable, while other variables are treated as constants. A derivative is a function which describes the rate of change of another function
Regression algorithm	When a continuous value is predicted
Supervised machine learning	A type of machine learning where training data includes a desired output

8. Appendices

Table 3.1.1 United States Environmental Protection Agency predictions for the trend of average data centre PUE [11]

Scenario	2007	2008	2009	2010	2011	Maximum Achievable
Current Trends	2.0	1.98	1,95	1.93	1.9	-
Improved Operation	1.94	1.88	1.82	1.76	1.7	1.7
Best Practices	1.9	1.8	1.7	1.6	1.5	1.3
State of the Art	1.89	1.78	1.67	1.56	1.45	1.2





Appendix 3.3.1: Machine Learning Algorithm Cheat Sheet by Microsoft [12]

Appendix 3.3.2: Table showing input variables for Google's Data Center neural network [7]

1. Total server IT load [kW]
2. Total Campus Core Network Room (CCNR) IT load [kW]
3. Total number of process water pumps (PWP) running
4. Mean PWP variable frequency drive (VFD) speed [%]
5. Total number of condenser water pumps (CWP) running
6. Mean CWP variable frequency drive (VFD) speed [%]
7. Total number of cooling towers running
8. Mean cooling tower leaving water temperature (LWT) setpoint [F]
9. Total number of chillers running
10. Total number of drycoolers running
11. Total number of chilled water injection pumps running

12. Mean chilled water injection pump setpoint temperature [F]
13. Mean heat exchanger approach temperature [F]
14. Outside air wet bulb (WB) temperature [F]
15. Outside air dry bulb (DB) temperature [F]
16. Outside air enthalpy [kJ/kg]
17. Outside air relative humidity (RH) [%]
18. Outdoor wind speed [mph]
19. Outdoor wind direction [deg]

Appendix 3.5.1: A table comparing machine learning programming platforms [10]

	Tool				
	Python	R	Spark	Matlab	TensorFlow
License	Open source	Open source	Open source	Proprietary	Open source
Distributed	No	No	Yes	No	No
Visualization	Yes	Yes	No	Yes	No
Neural nets	Yes	Yes	Multilayer perceptron classifier	Yes	Yes
Supported languages	Python	R	Scala, Java, Python, and R	Matlab	Python and C++
Variety of machine-learning models	High	High	Medium	High	Low
Suitability as a general-purpose tool	High	Medium	Medium	High	Low
Maturity	High	Very high	Medium	Very high	Low