# Machine Learning Engineer Nanodegree

## Capstone Proposal

Payam Mohajeri
18 December 2019

## Domain Background

Virtualisation of application environments and containerisation of applications is a trending topic these days. Different solutions are provided to virtualise an operating system or containerise an application; and implement a cluster of nodes for managing or maintaining applications. One of the common topics within these clustering solutions is about distributing the load based on the available infrastructure resources, for example processing power and memory. For distributing the load, a component needs to keep monitoring the environment and provide the information to a scheduler to maintain the environment and organise future application deployments. The scheduler has an important role for improving and enhancing the environment or applications stability. One of the key metrics that scheduler has to consider is CPU utilisation. This project is going to look into providing a model for predicting the CPU utilisation.

## Problem Statement

Improving the stability and performance of distributed applications a cross the globe is a challenging topic for enterprises. An application environment can get unstable when there is a heavy load on specific applications or nodes. Users often face high response time and low performance in case of any environment instability and this can have a huge impact on the branding or business of targeted enterprises and organisations. Today's competitive market requires organisations to provide stable and well preforming application environment to their customers, but providing such stability based on a limited visibility on utilisation of infrastructure resources is very challenging. On this project I'm focusing on improving the visibility of system metrics by providing a model for predicting the CPU utilisation based on the previous infrastructure monitoring records. Beside improving the stability, saving energy and costs are also other use cases on such model.

## Datasets and Inputs

Following provided dataset from Burn CPU Burn competition on Kaggle will be used:
https://www.kaggle.com/c/model-t4/data

The goal is to predict the load on the CPUs in a cluster of servers based on the applications behaviour running on these servers. There are two CPUs per each server and in total there are seven servers in a cluster.

“The dataset consists of a set of variables that were measured over about a one month period. Measurements were taken in one minute intervals and on each server. Measurements are usually the average or sum over that one minute interval. For instance the number of packets received, the average number of IO operations, etc.”

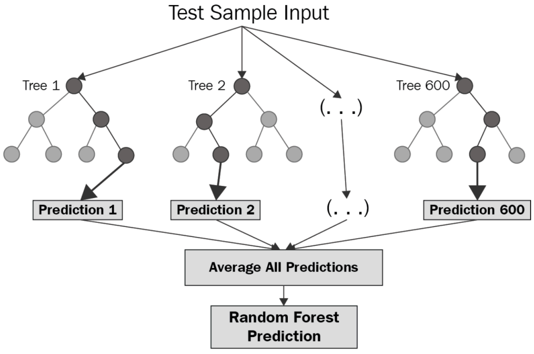
This dataset is from a real cluster that is used to control train traffic in a geographical area spanning several cities. These fields are available on provided CSV files:

- sample time: the date and time the data was sampled.
- m_id: the ID of the server the data was sampled at.
- appxxxx: data about specific application.
- pagexxx: data on memory usage of the server.
- syst_xxx: data on page fault rate, number of processes, etc.

- state_xxx: data on the state the system is in.
- io_xxx: data about general IO usage, (file IO, direct IO).
- tcp_xxx: data on incoming and outgoing TCP traffic.
- llxxx, ewxxx: data on incoming and outgoing network traffic.
- cpu_01_busy: the variable we are trying to predict.

## Solution Statement

The solution is to collect the system related metrics like the usage of memory, I/O, network traffic and train a model which can link the recorded data to the CPU utilisation and can provide the estimation in case the system behaviour is similar. This trained model for our prediction needs to be based on a supervised learning. Different regression algorithm can be used to train the model. As a first idea I would like to try on Random Forests and Extra Trees methods since the data set has so many different fields and it might be better to try ensemble learning methods.



Random Forest Structure

Random Forest Regressor:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Extra Trees Regressor:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html

## Benchmark Model

Based on the information provided on https://www.kaggle.com/c/model-t4/leaderboard the benchmark model is a Two Variable Random Forest model which have been evaluated and scored as 11.18079. For our case here, I think Two Variable model is a very basic model and I'm looking to get better results since there are more than 2 variables to be considered according to our datasets.

## Evaluation Metrics

Based on the information provided on https://www.kaggle.com/c/model-t4/data the evaluation metric is the Root Mean Square Error:

$$S = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{l}_i - l_i)^2}$$

* l hat is the predicted load and l the actual load.

“For every line in the  test set, submission files should contain two columns: Id and Prediction. Prediction is a real number in the range of 0 to 100 predicting how busy CPU 1 was in the given machine on the given date.”

## Project Design

I intend to use what I have learned so far on this project, I will go through the discussions on Kaggle and implement a model on Jupiter notebook while deploying it on AWS. I'm interested to submit the results on Kaggle and compare / improve them based on the available rankings as well.