# Topic:

Hi my name is Jiaqian. I am a first year graduate student. Today I would like to introduce my class project topic which is to predict process performance using ML

The prediction of OS process performance can bring several benefits, including:

Early detection of performance issues

Improved resource utilization

Better capacity planning

Reduced downtime

Improved user experience

Cost savings

From terminal the user is able to check the information for different processes.

PID: Shows task’s unique process ID.

USER: User name of owner of task.

PR: Stands for priority of the task.

NI: Represents a Nice Value of task. A negative nice value implies higher priority, and positive nice value means lower priority.

VIRT: Total virtual memory used by the task.

RES: Represents the amount of actual physical memory a process is consuming.

SHR: Represents the Shared Memory size (kb) used by a task.

S: The state the process is in.

%CPU: Represents the CPU usage.

%MEM: Shows the Memory usage of task.

TIME+: CPU Time, the same as ‘TIME,’ but reflecting more granularity through hundredths of a second.

COMMAND: The command that is being run.

apply Machine Learning (ML) techniques to learn the performance and behavior of processes in operating system

~~In our experimentation we modify the Linux Kernel scheduler to allow scheduling with customized time slices. The "Waikato Environment for Knowledge Analysis" (Weka), an open source machine-learning tool is used to find the most suitable ML method to characterize our programs. We experimentally fined that the C4.5 Decision Tree algorithm most effectively solved the problem. We fined that predictive scheduling could reduce TaT in the range of 1.4% to 5.8%. This was due to a reduction in the number of context switches needed to complete the process execution. We find our result interesting in the context that generally operating systems presently never make use of a program's previous execution history in their scheduling behavior.~~

# Approach:

We have approaches to make this project reach our goal.

Firstly we need large amount of data related to process. We should gather data related to various performance metrics and system configurations. Once we have gathered the data, we need to preprocess data and use machine learning algorithms to build and train a predictive model that can predict processor performance based on the input data. the data will be divided into trained data and test data. We can use regression algorithms such as linear regression, decision tree regression, or random forest regression to predict the performance. You can also use classification algorithms such as decision trees or neural networks to classify the performance as good or bad based on the input data.

My current progress

I have devided the project into couple steps. 1. Project design, 2. Kernel module writing in c lanague, 3 and 4. Collecting data on linux and macbook 5. Preprocess data 6. Build ml mordule for prediction 7. Last but not least create test cases to enfure the data we collect are accatuae.

As you see, I have marked some of steps done and will provide demo for kernel module and data collecting in the next few slides.

Besides, I have attacked my current repository structure. It may change slightly since I haven’t 100% complated this project on coding part. Most of code are categoried into kernel module filder, data collecting folder and ML python folder. For c lanague we need to compile the file to make it exexuable on linux. So I have created couple make file to make the work of typing command into terminal in an easier way to handle.

# Steps:

Here is the details about each steps

The first one is about kernel module. I have writtn a kernel module which help collects the data in linux system. The goal for this kernel modle is to log information related to processor. The kernel module will be installed in linux and when we dump the list of kernel modules, it will be shown in the list with other default kernel module for Linux system.

Click on the demo video. As we see, we exec the make file for compiling the proclog c file and it will generate couple files with proclog name but different file types. Proclog.ko is the one used to install in Linux system. When ko file is installed successfully, the msg of “The process logger module loaded” will be shown when we dump the kernel print by command `dmesg`. Meanwhile, A proc\_dir\_entry is created in folder /proc/ and it is a virtual file called log\_file. In the demo I have dumped the first 10 processes logged into this entry. This log\_file is unwritable by user but only written by kernel. As the linux environment keeps on, the file is updated constantly. As long as we type “make stop” which is a custom command line from make file I made for removing the kernel module, the log\_file entry will be removed. If it is not removed, next time we run the install kernel module again, this log\_file will be counted as registered we don’t want it become a useless entry point in the /proc/ folder because no user can use it any more and it is sitting there and not working by our purpose.

# What is Linear Regression?

Linear regression is a basic and commonly used type of predictive analysis.  The overall idea of regression is to examine two things: (1) does a set of predictor variables do a good job in predicting an outcome (dependent) variable?  (2) Which variables in particular are significant predictors of the outcome variable, and in what way do they–indicated by the magnitude and sign of the beta estimates–impact the outcome variable?  These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables.  The simplest form of the regression equation with one dependent and one independent variable is defined by the formula y = c + b\*x, where y = estimated dependent variable score, c = constant, b = regression coefficient, and x = score on the independent variable.

Naming the Variables.  There are many names for a regression’s dependent variable.  It may be called an outcome variable, criterion variable, endogenous variable, or regressand.  The independent variables can be called exogenous variables, predictor variables, or regressors.

Three major uses for regression analysis are (1) determining the strength of predictors, (2) forecasting an effect, and (3) trend forecasting.

A random forest regressor.

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max\_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

Some challenges and critical thinking I have for this project. First of all, the data comes from my linux system. Can they represent all other system for prediction?

Secondly, is the ML model make our life easier to predict the performane. Next, how to prove my model can provide accuate prediction. Some of those thoughts may be addressed in the process of accomplishing this project, some thouhts may be carried into next milestone.

Challenges and Future work

1. Kernel module creates a virtual file by `proc\_dir\_entry` kernel function and it is dangerous for kernel to write an actual file so that an additional c file is needed to export content of virtual file into actual file. —> Not efficient enough to collect linux data
2. However, the data in /proc/ is meaningful to collect so that I didn’t give up on building kernel module to log process in with custom inforation