Using crypto to make purchases securely depends on what you’re trying to buy. If you’re trying to make a payment in cryptocurrency, you’ll most likely need a cryptocurrency wallet, a software program that interacts with the blockchain and allows users to send and receive their stored cryptocurrency.

Here are some of the most common security features found in cryptocurrency wallets:

Private keys: long and complex codes that grant access to the assets stored in the wallet. Private keys should be kept confidential and not shared with anyone.

Two-factor authentication: Many cryptocurrency wallets offer this as an added layer of security. This involves requiring a second form of authentication, such as a text message or app-generated code, in addition to a password to access the wallet.

Biometric authentication: Some cryptocurrency wallets also offer biometric authentication, such as fingerprint or facial recognition, to add an additional layer of security.

Encryption: Some cryptocurrency wallets use good encryption to secure the private keys and other sensitive information stored in the wallet.

Proof of work and proof of stake are the two most widely used consensus mechanisms to verify transactions before adding them to a blockchain. The main difference between them is that for “Proof of work” the network requires a huge amount of processing power and for Proof of stake, on the other hand, it involves users staking or holding a certain amount of cryptocurrency as collateral to validate transactions and create new blocks on the blockchain.

As the cryptocurrency becomes popular, the use of cryptocurrency, however, varies on different countries. According to the U.S. Library of Congress, as of November 2021, nine countries have banned cryptocurrency completely. 42 countries have an implicit ban on the asset, generally by the means of not allowing financial institutions in the given country to take on crypto companies as clients.

# Topic:

To start the project, I have done some research and project design, aka determine what is the big picture of this project and draw a rough graph of how to accomplish my goal.

So here are my brief ideas of how to do it.

I plan to collect large amount of data related to process on both Linus and mac book. I need to gather data related to various performance metrics and system configurations. Once I have gathered the data, I need to preprocess data and make it ready to apply machine learning algorithms to build and train a predictive model that can predict processor performance. 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. I 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. So sounds straightforward.

Steps:

Third step,

I have chosen linear regression algorithm to build the ML model which has major uses of determining the strength of predictors, forecasting an effect, and trend forecasting. And I have chosen random forest regressor too. It 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. In the demo I have chosen each process kernel time as y and the other numeric columns as X. From both linear regression model and random forest regressor model, I have got over 96% accuracy between predicted value and actual values. I have applied Mean Squared Error (MSE) metric which is the most commonly used for evaluating a model. Also, I have answered the question I have raised from my project proposal.

3. Predictive Models

3.1. Linear Regression (LR) Models

Regression analysis is a statistical technique for investigating and modeling the relationship between variables. In this model, we have n observations y=y1,...,yn called the response variables and xi=xi,1,...,xi,p for i=1..n that are predictor or regressor variables. The simplest linear regression is of the form y=β0+β1x+ε. In this formula β represents the coefficients used in describing the response as a linear function of predictors plus a random error ε. In our input data set we have multiple predictor variables, causing the response y to be related to p regressor/predictor variables. The model then becomes y= β0+β1x+β2x+...+βpx+ε, where y and x are vectors of n numbers (observations). The fitting of a regression model to the observations is done by solving the p+1 β coefficients. The method of least squares error (LSE) is used. In this model, it is assumed that the error term ε has E(ε)=0, Var(ε)=σ2 , and that they are uncorrelated. The least-square equation is of the form [7]

Graphical user interface, text, application

Description automatically generated

In addition, I have also encountered some challenges for this project.

Firstly, there are Limited resources and various restriction on building Kernel module. It is not easy to debug on kernel module. The error msg does not provide sufficient information for developer to locate the error. And as I mentioned earlier, an additional c file is needed to export content of virtual file into actual file. Thus it is Not efficient enough to collect linux data in this way. Compared with the data collecting approach I applied on macbook, it still has some advantage that the data is recorded constantly on virtual file. I can still choose to export large gigabytes of data without missing consistent data info since the kernel module is working.

Secondly, when I use python some data are not available such as io counter and net counter for each process. Disk io and network is can contribute to performance of processes too.

Besides, I have some plan for Future work.

1. In the next few days, I plan to Modify and wrap up the ML prediction
2. Create more test cases make sure prediction reliable
3. Find the Answer for my last question from project proposal: Does our model provide accurate prediction of performance? WIP to build test cases to verify.

~~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:

I need to gather data related to various performance metrics and system configurations.

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.

# Steps:

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