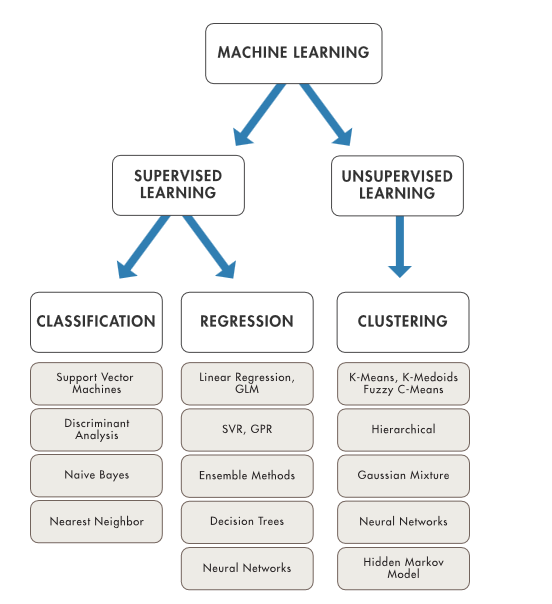
**Tensorflow:**

Everything starts with Google AIY( do-it-yourself artificial intelligence) Vision Kit, which can recognize people smiling face without internet. I tried the kit in the summer time, and found it is really cool! And start digging into what is behind.

Here is a little story when I first try this toy, the light does NOT ON regardless how I smile. The kit has a Raspberry Pi board, a VisionBonnet board (Movidius chip MA2450) with a camera. I connect Raspberry pi board with keyboard and monitor via micro-usb/mini hdmi interface, check the logs and issue testing command. Connect the camera directly to the Raspberry pi board directly to make sure the camera is good. And finally identify the ribbon cable between the two boards was not connected properly.

The first thing come to my mind is google tensorflow, I did some reading on the machine leaning.



Neural networks CNNs(mostly for image, classification prediction), RNNs(sequence predication problems)

**Talk about the knowledge of neural network, layers, activation function**

**Talk about the tensorflow details:**

I used pip to install the tensorflow for CPU only version on windows. Python or JavaScript?

The last but not least to mention here is the VPU. I plan to do some project on it in future.

MA2450 is a vision processing unit VPU, allowing a vision accelerator to be added to devices using ARM processors including PCs, drones, robots, IoT devices and video surveillance for tasks such as identifying people or objects

<https://www.extremetech.com/extreme/222095-google-taps-chipmaker-movidius-to-add-machine-learning-to-phones>

**Quadcopter Drone**

As the project of SE101, I team up with another girl, built a quadcopter drone system from scratch. The system include a frame, motors, ESCs, Propellers, battery, Arduino board, IMU, NodeMcu-ESP8266, cell phone control app.

I designed and implemented an Android app to control the drone via cell phone. The development is done in Java within android studio under windows. It allows user to control the direction of the drone. The app itself is simple and easy. I had experience to build android app before in the IBM’s girls who code activity.

The web server in NodeMcu-ESP8266 is actually cost me lots time and efforts.

* First, the information the web server received is not stable. When the app send out an UP, the server some time get a DOWN. I added logs from the first statement after receiving message, and finally found out it is because filter used to parsing the message is not accurate. I made the filter more specific and solved this issue.
* Second, the response time is long in second. I located the issue receiving message, not sending message. And it is not because of the length of the information. I found out exactly which statement cause the delay via print out the message to serial console. I google the problem symptom online. And tried several approaches: using different library of receiving message, setup different parameters. At last I found one way to setup the timeout parameter that short the response time to million second level, which is fast and still guarantee the whole message is received.

The communication between Arduino UNO boards is ESP8266 is my responsibility as well. I tried both direct Rx/Tx connection and I2C connection. Both works well. To make thing simpler and reliable, I decided to use the direct Rx/Tx approach.

I also involved the debugging of drone gyro stabilization code on Arduino UNO. **Talk about one example what problem found and solved.**

Big lesson learned is to prepare the hardware with backup. The motor is burned several days before the demo date. An extension of couple days was requested and granted, I managed to get the new motor and ESC ASAP, and started the soldering the wires by myself at midnight. And very excited to see the new hardware works in the demo.

In addition, I find the NodeMcu-ESP8266 is very powerful. It is a low cost WIFI micro-controller with full TCP/IP stack, which make is really useful to Internet of things(IoT). And there are lots forum and support web site for it. Eg. <http://esp8266.net/>

**HTBC website:**

I am not sure the technique issues solved, the auto-process the receipt is defiantly one of the highlights.

Here are some questions may need pay attention:

* + - How do you increased the google crawl rate of the web site?
    - What is behind the web site? What software/frame work are used to support the web site? Where is it hosted?
    - What is the work load to rebuild a similar website for another club?