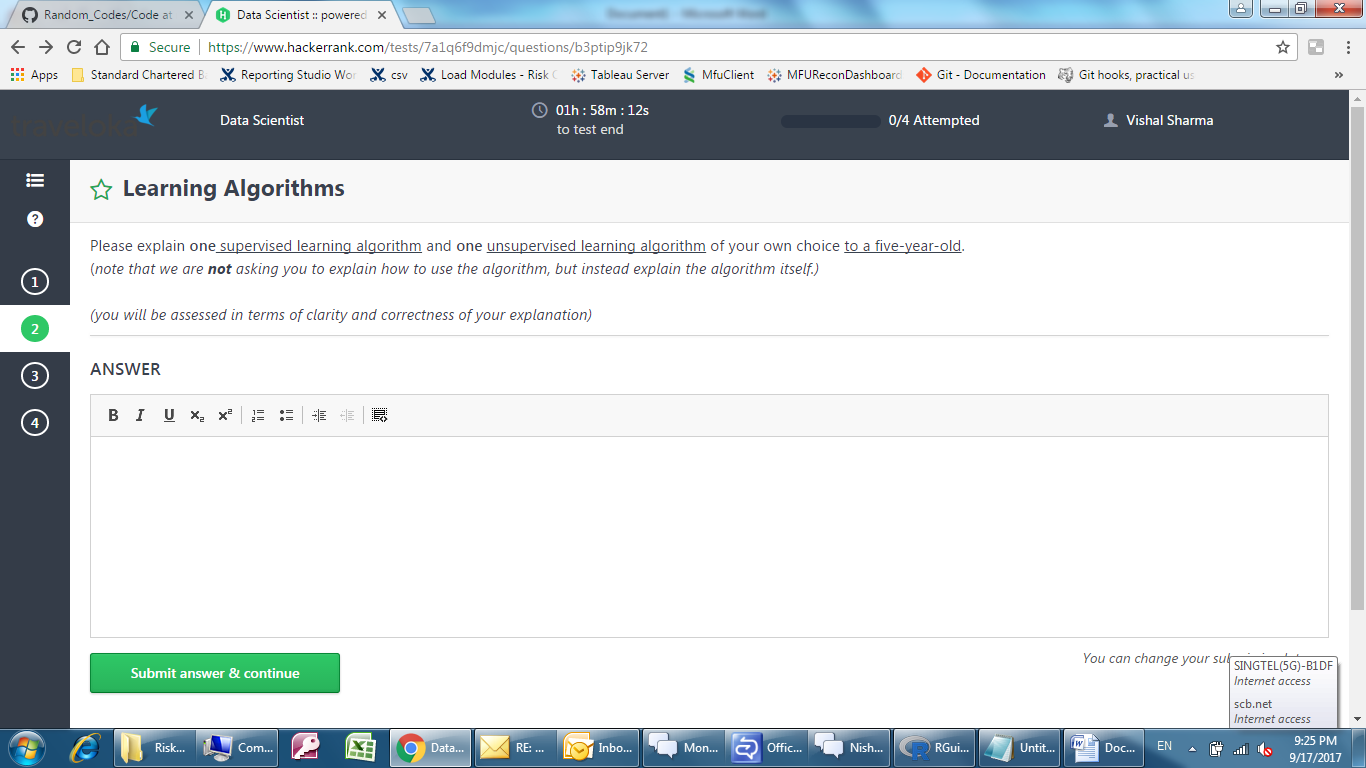


http://pi3.sites.sheffield.ac.uk/tutorials/week-9



Unsupervised Learning Algorithm:

I would like to pick an algorithm known as agglomerative clustering algorithm. Suppose you can show point at the different toys that kid is having. Now I will ask the kid to put together the toys different type based on the appearance on the shelf. He would put together captain America and hulk in one place. Then giJoe characters in one corner of shelf and spiderman in another

* Cluster 1 – Captain America and Hulk
* Cluster 2 – GiJoe characters
* Cluster 3 - Spiderman

It’s not enough to be based on the appearance. Still we can do much better by now looking at the powers of the different characters (not disturbing the cluster appearance leading to increase in size). Then you can combine Cluster 1 and Cluster 3 together as they both have super natural powers.

* Cluster 1 : Captain America and Hulk and Spiderman
* Cluster 2: GiJoe characters

So now you have increased your cluster sizes, but the relation between the toys has decreased. (Earlier the toys were extremely similar because of similar muscular nature, now they are by the superpowers, on different powers)  
  
If you find that the number of clusters are still large, you combine multiple toys together based on their other characteristics, and not just on the muscular nature and superpower.

As you keep doing this, the relation between toys within a cluster keeps reducing, but the size of the cluster keeps on increasing. You choose a size and intra-cluster relation that suits you.  
  
This is agglomerative clustering, where you start by small clusters, and then keep combining smaller clusters to form larger clusters.

Supervised Machine Learning Algorithm:

The algorithm that I would like to discuss is called Support Vector Machine.

I will get some candies of 2 types i.e. chocolate and orange flavour and put on a table. Now I will put the stick on the table and show that the stick is able to separate the 2 types of candies pretty well.

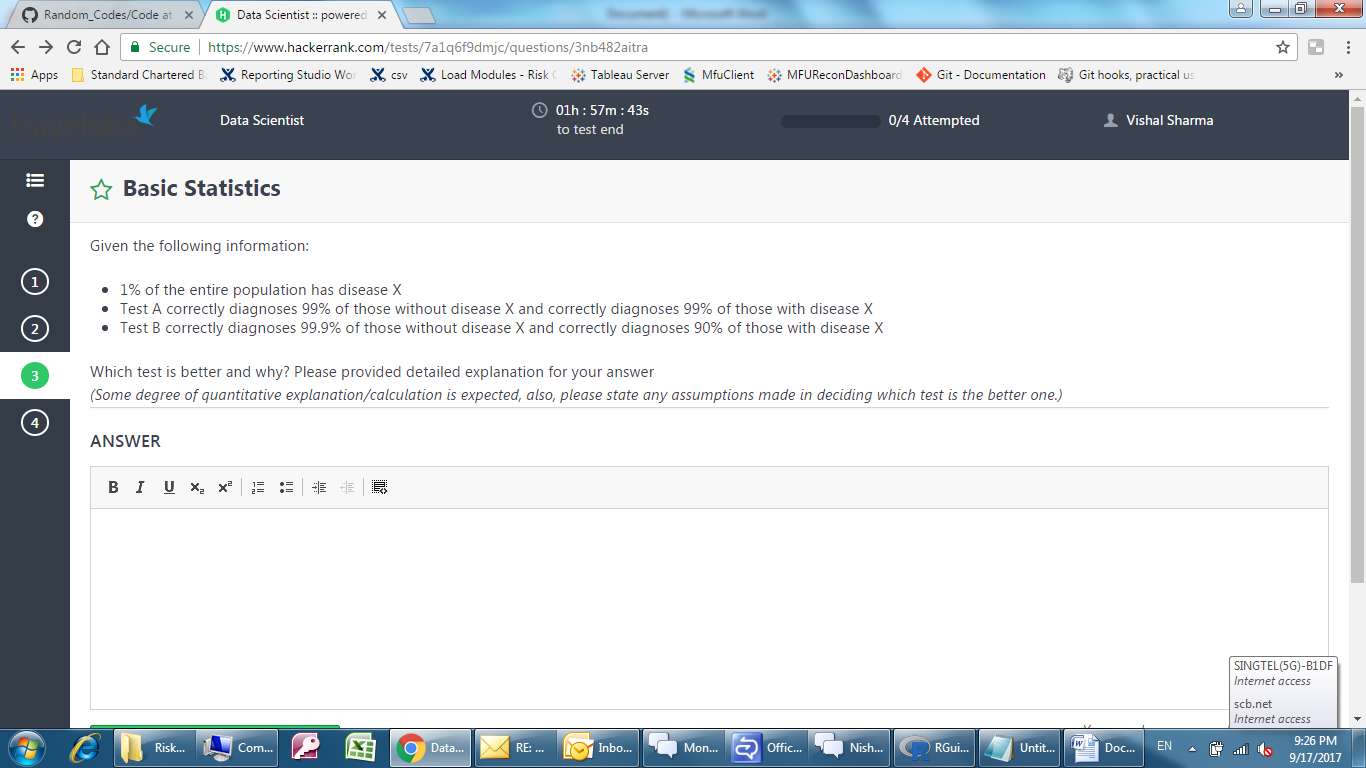
Now let’s say some other mischievous kid sees that the stick is able to separate the candies pretty well and places more candies on the table of the 2 varieties. We see that stick is placed quite well but still one candy is placed on wrong side and we can place the stick in better position now for separation. SVMs try to put the stick in the best possible place by having as big a gap on either side of the stick as possible.

Now mischievous kid returns to see that still we are able to separate the different types of candies. He tries to act more intelligent and changes the location of candies so it can’t be separated with a simple stick. He smiled and gave us this new challenge. Then I will show another trick that you flip the table of course! Throwing the candies into the air. Then, with your pro ninja skills, you grab a sheet of paper and slip it between the balls. Now, looking at the candies from where the mischievous kid is standing, the candies will look split by some curvy line.

Boring adults calls candies as data, the stick a classifier, the biggest gap trick optimization, call flipping the table kernelling and the piece of paper a hyperplane. Once we've found the best place to put the stick or the paper, we realise that we can figure out that best place from just a few of the closest candies to the stick/paper and most of the other candies are unhelpful. We call these important candies- support vectors.

<https://www.quora.com/How-would-you-explain-agglomerative-clustering-to-a-child>

https://www.reddit.com/r/MachineLearning/comments/15zrpp/please\_explain\_support\_vector\_machines\_svm\_like\_i/



We have 2 construct the scenarios with the help of an example of population to be 10,000,000. Let’s now take the scenario one by one:

A table can be constructed which helps in better explaining the probabilities. The scenarios can be compared with the help of example let’s say patient test positive which of the above 2 options performs better:

* Test A:

Disease X No Disease Total

Tests+ 99000 99,000 198,000

Test- 1000 9,801,000 9,802,000

Total 100,000 9,900,000 10,000,000

Now the probability of disease given that patient tests positive is having the probability of 99000/198000=0.50 chance of correct diagnosis

* Test B:

Disease X No Disease Total

Tests+ 90,000 9,900 99,900

Test- 10,000 9,890,100 9,900,100

Total 100,000 9,900,000 10,000,000

Now the probability that patient has disease given that it tests positive is having the probability of 90000/189900=0.90 chance of correct diagnosis

So Looking at this scenario Test B is better as it’s able to diagnose the scenario that when doctor is getting the test as positive and able to detect the disease.

We can also look at 2 other metrics for comparison accuracy as well as F score:

Accuracy: (TP + TN) / (TP + TN + FP + FN)

Test A: = 99,000+9,801,000/99,000+9,801,000+99,000+1,000=0.99

Test B = 90,000+9,890,100/90,000+9,890,100+ 9,900+ 10,000=0.998

Here test B has higher accuracy so its better.

http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704\_probability/bs704\_probability6.html

