Except for reading, the first two parts are supposed to be finished in 2 weeks.

1.Coding

We were discussing K-Means algorithm carefully and thoroughly. And please do remember that you need to be able to implement that all by yourselves in Python or in C++.

Besides, if your target are those big names, K-Means++ is also compulsory.

So you are highly recommended to implement K-Means++ in Python or in C++. And you are welcome to accomplish this based on the K-Means code we provided in our course.

2.Mathematical Problems

The last topic of our last week is Decision Tree [You can find it in our slides] where we have 3 algorithms (ID3, C4.5 and CART) in which the fundamental one is the ID3 algorithm which was implemented in 1979 initially by Quinlan.

In our slides, we provide a specific example of implementing a decision tree by using ID3 algorithm step by step. The only problem of that is we intentionaly missed some computing details of how to build up the tree.

Therefore your task is to supplyment details of calculation of each node to the tree and provide the evidence why you split a node in such way.

You can follow what we did in slides. And to check whether you set up the correctly or not, you can also find the final tree in our slides.

P.S.

A. If you still think this is too hard to complete, please don't worry. We can cover this topic in Week 15.

B. If you still have time, you can learn C4.5 and CART by yourself as well from our slides and please try to answer questions:

I. What is Gain Ratio?
II. Why we are prone to use Gain Ratio?
III. How to split a node by using Gain Ratio?
IV. What Gini Index?
V. How to split a node by using Gini Index?
VI. Why people are likely to use C4.5 or CART rather than ID3?

3.Reading

[If you have limited time, just ignore this part. It's not critical for most companies.

If you are fascinated in machine learning or if your targets are those superstar companies, then please go on.]

There is another classical machine tool named AdaBoost that we were not talking about. The most famous application for that in CV field is face detection implemented by Viola and Jones alongside Haar features (Actually this algorithm is so successful that it has been included in OpenCV which means you can just call this function to do simple face detection in just no more than 3 lines).

So you can read the materials we provide in our slides and try to understand these two concepts:

A. What is AdaBoost algorithm and B. What is Haar Feature.

Then try to answer the following questions one by one:

For AdaBoost:
a. What is a weak classifier?
b. What is a strong classifier?
c. How to combine those weak classifiers?
d. How to update a weak classifier?
e. How to update the strong classifier?
f. Can you complete the mathematical derivation by hand?

For Haar feature:
g. What is a Haar feature?
h. Can you find out any upgrade Haar features besides the original one?
i. Can you implement a Haar feature in Python or C++?
j. Can you implement the algorithm in a accelerated way? Like integral image?
k. How to combine Haar feature with AdaBoost?

If you can answer them all, then trust me, you've got a full score.