**COMP 4432 Assignment 1**

RuixiangJIANG 19079662D

**Q1**

**a) Classification rules of the DT**

IF DAY\_of\_the\_Week is Mon-Thurs AND From is NT THEN Transportation\_Taken is Bus.

IF DAY\_of\_the\_Week is Mon-Thurs AND From is HK AND TO is NT THEN Transportation\_Taken is Bus.

IF DAY\_of\_the\_Week is Mon-Thurs AND From is HK AND TO is HK THEN Transportation\_Taken is Bus.

IF DAY\_of\_the\_Week is Mon-Thurs AND From is Kln AND TO is HK THEN Transportation\_Taken is MTR.

IF DAY\_of\_the\_Week is Mon-Thurs AND From is Kln AND TO is HK THEN Transportation\_Taken is Bus.

IF DAY\_of\_the\_Week IS Fri-Sun THEN Transportation\_Taken is MTR.

**b) prune the tree**

**图示

描述已自动生成**

**c)**

**i.** when all of the records added did not change the fact that nor it change the majority value in leaves(2MTR - 3Bus -> 4MTR - 3Bus), then the tree need no reconstruction.

**e.**g. adding such record, no reconstruction is needed.

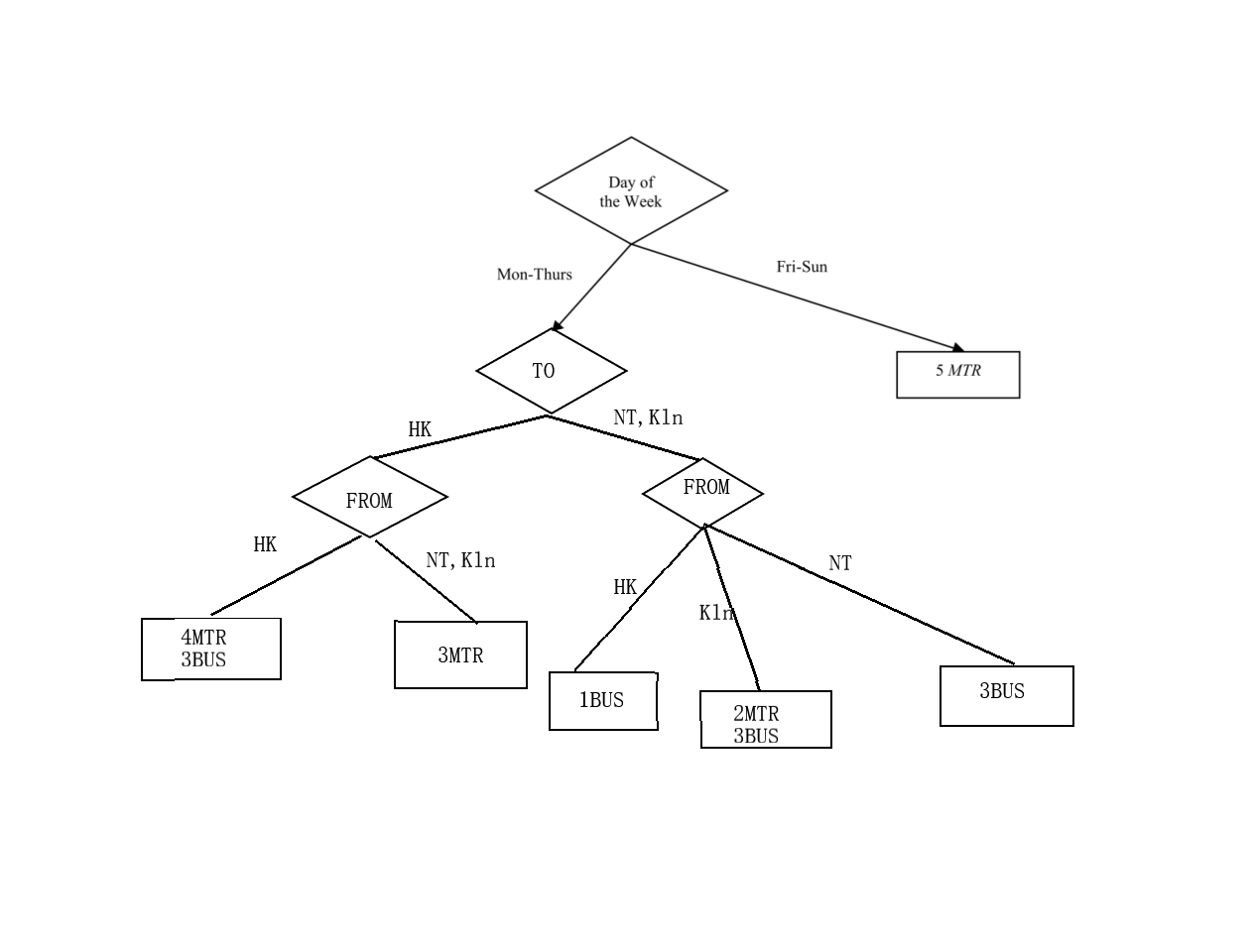
|  |  |  |  |
| --- | --- | --- | --- |
| Fri-Sun | HK | HK | MTR |

**ii.** Adding the new records that cause at 2nd split. This will lead to partial reconstruction

e.g. adding following records:

|  |  |  |  |
| --- | --- | --- | --- |
| Mon-Thurs | KlN | HK | MTR |
| Mon-Thurs | HK | HK | MTR |
| Mon-Thurs | KlN | HK | MTR |
| Mon-Thurs | HK | HK | MTR |

Then tree will be partially reconstructed (at the left sub-tree of root node) as such (this tree is compressed due to the paper size limit and just for demonstration purpose):

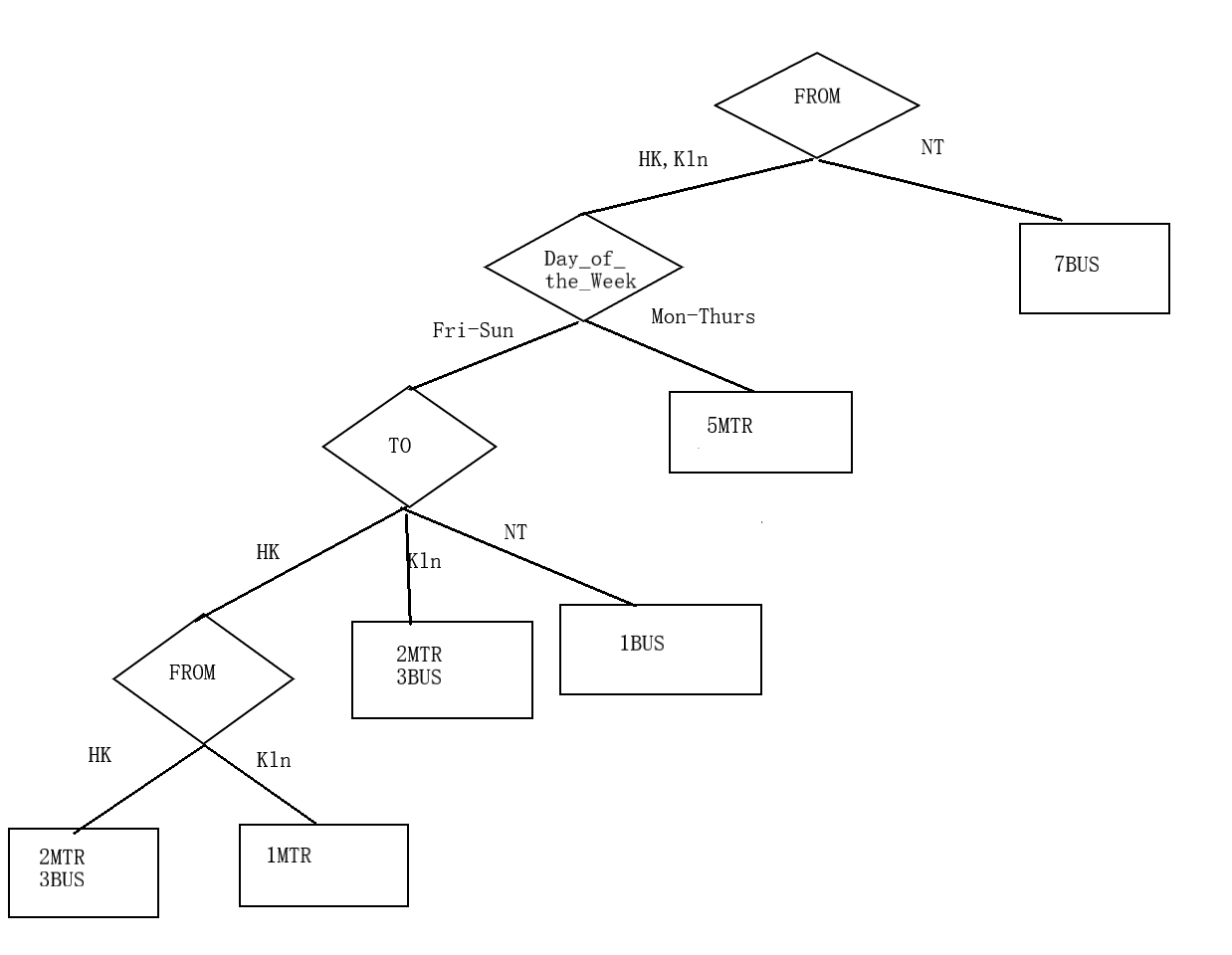


**iii.** Adding the new records that cause or at 1st split (root). This will lead to full reconstruction of the decision tree.

e.g. adding following records:

|  |  |  |  |
| --- | --- | --- | --- |
| Fri-Sun | NT | HK | Bus |
| Fri-Sun | NT | HK | Bus |
| Fri-Sun | NT | Kln | Bus |
| Fri-Sun | NT | Kln | Bus |

Then the tree will need full reconstruction like this (this tree is compressed due to the paper size limit and just for demonstration purpose):



**Q2**

**a) minimum number of nodes (other trees with 7 nodes is also correct)**

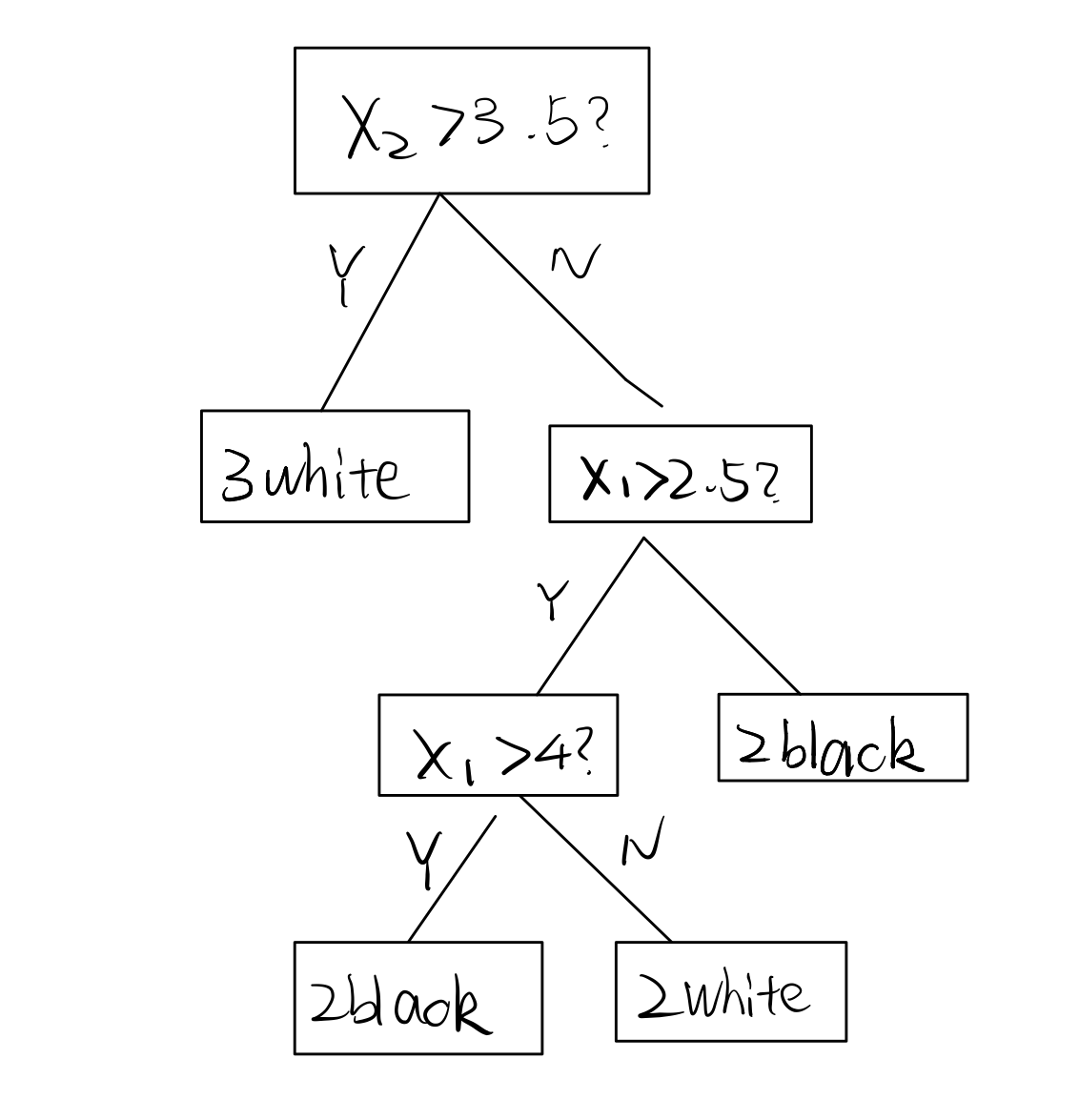


Figure 1 min nodes

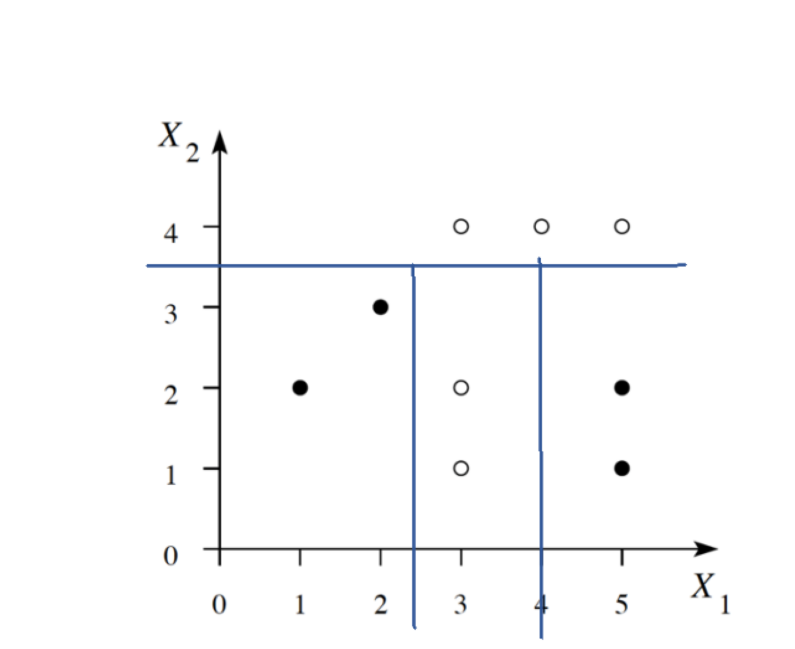


Figure 2 min nodes boundary

**b) minimum number of levels (other trees with 3 levels is also correct)**

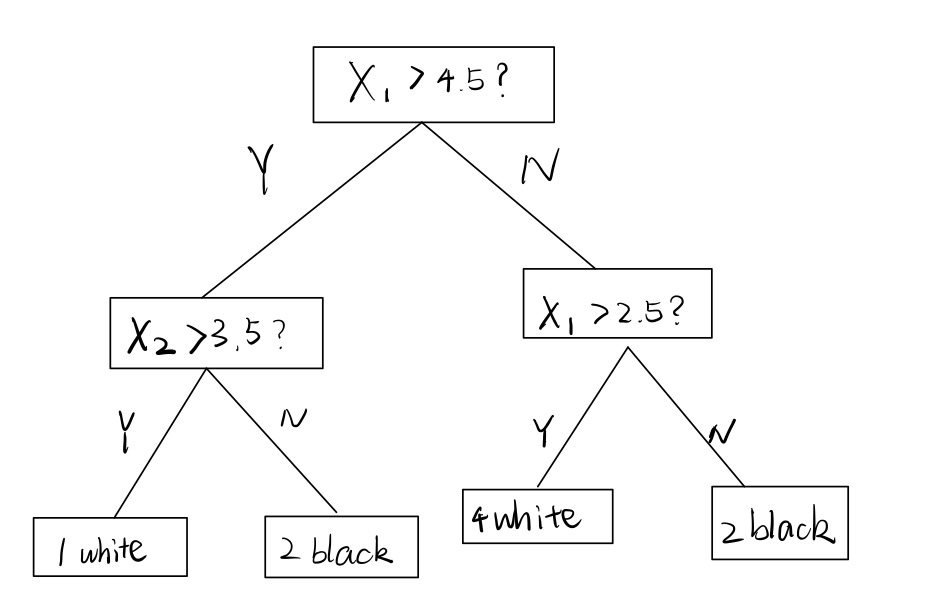


Figure 3 min level

图表, 散点图

描述已自动生成

Figure 4 min level boundary

**Q3**

**a) *k = 1***

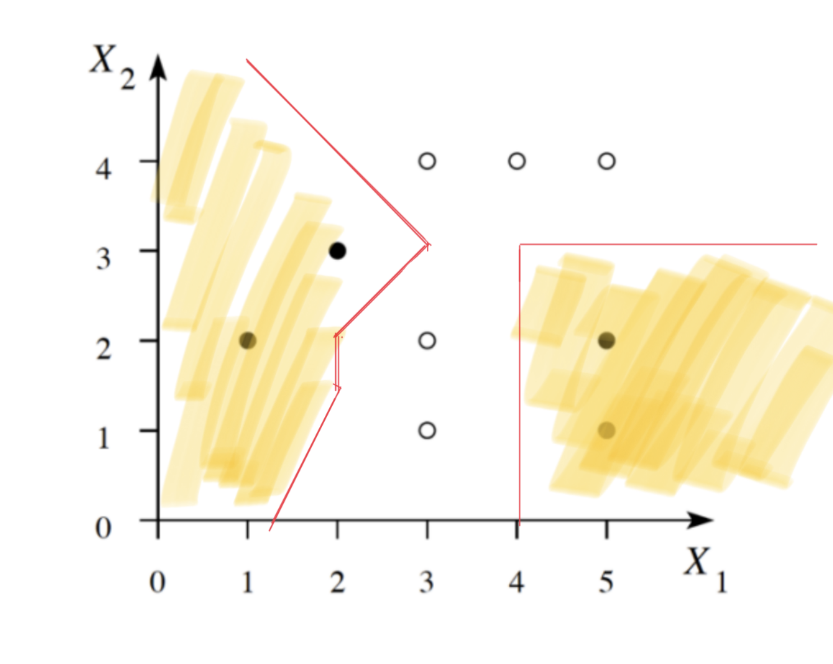


Figure 5 (k = 1)

3.

**b) *k = 3***

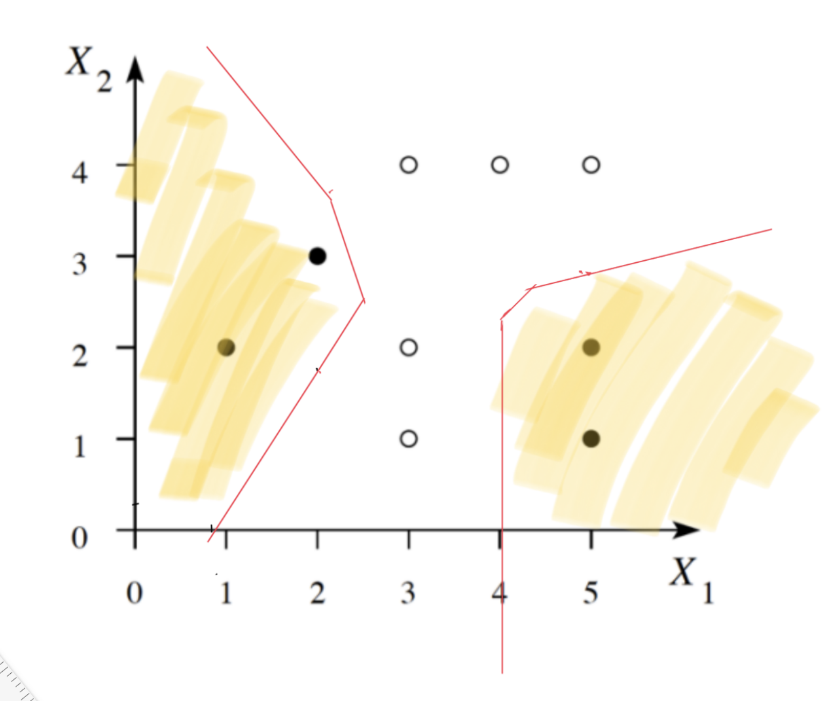


Figure 6(k = 3)

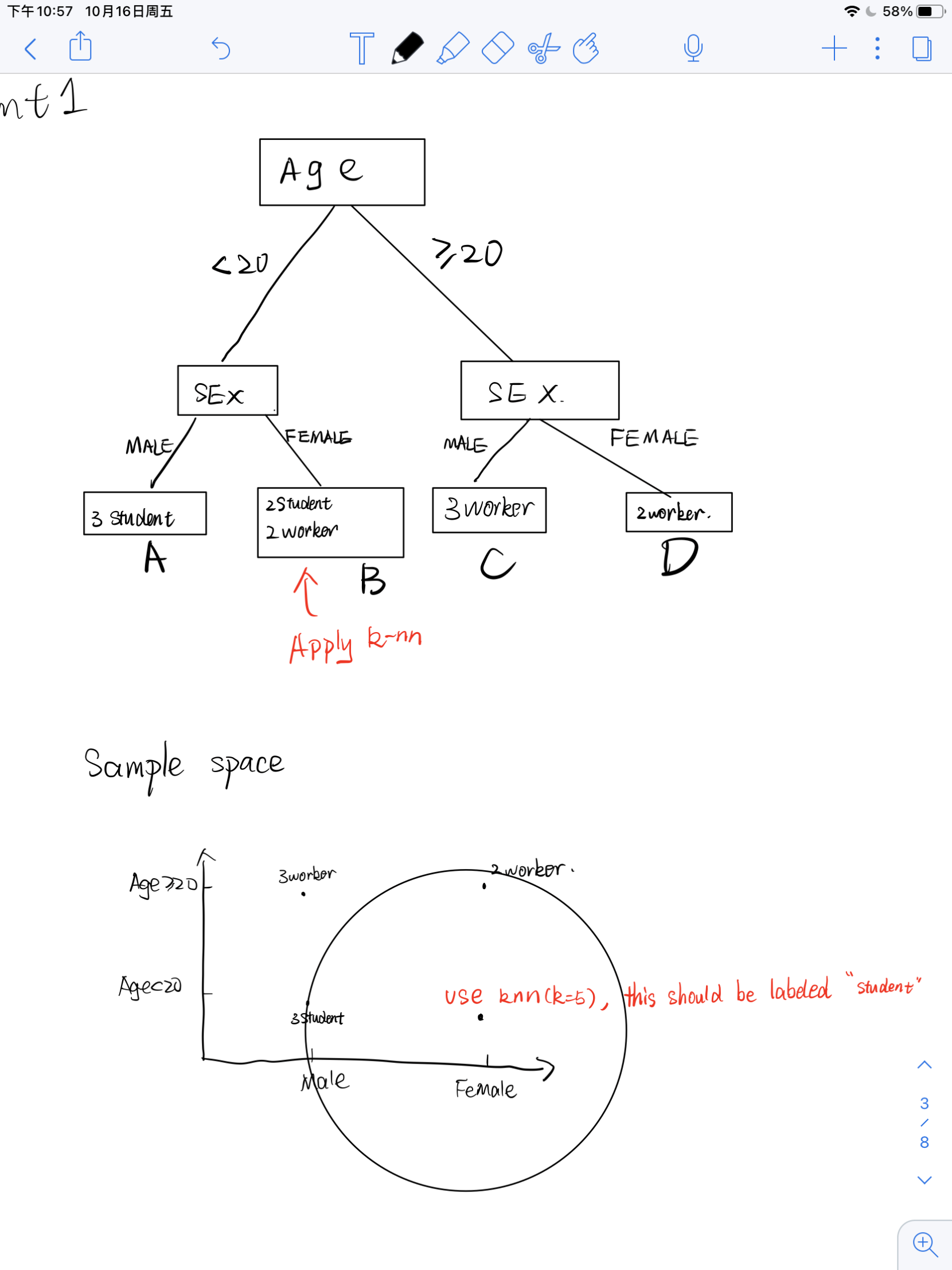
**Q4.**

In A DT classifier (ID3/C4.5), the leaf node maybe impure due to limited attributes. In those cases, a workable strategy is to follow the majority. This may work well on some leaves where the information entropy is relatively low (e.g. 1 vs 9, ). However, when information entropy is high, such as 4 vs 6, , this method may be quite unsatisfactory. If this is the case (information entropy of a leaf node is higher than certain threshold), knn maybe used to enhance the performance (in terms of accuracy). This will be further illustrated below.

My proposal is to use knn to the obtain the expected value of those impure leaves with high information entropy.

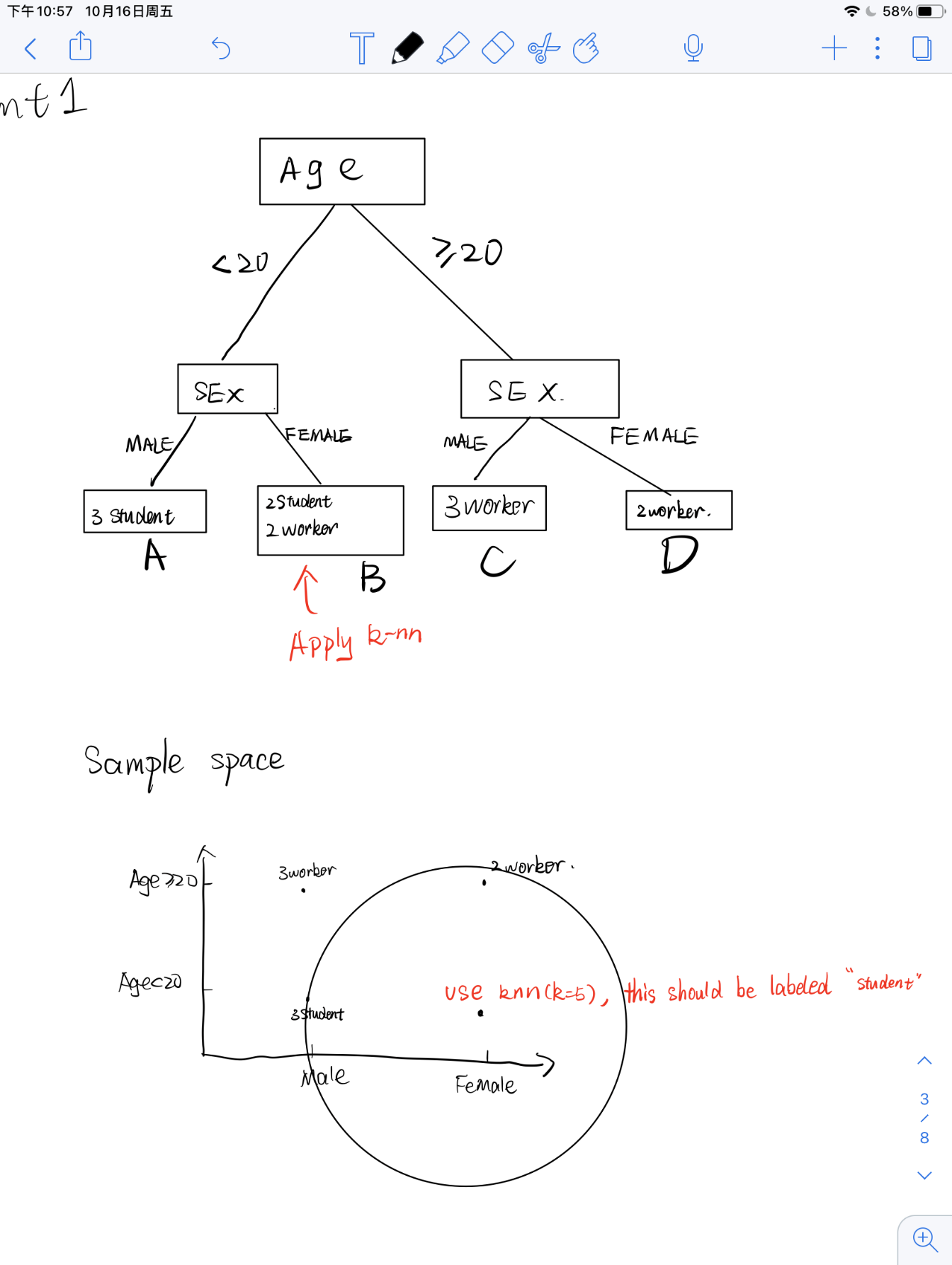
Conceptually speaking, a leaf should be “near” to its sibling leaf in the sample space, which means that the majority of attributes of them should be same. Thus, we can use nearby leaves to predict this leaf.

To find the nearest leaves, we need to traverse the tree and find the nearby nodes. By saying “nearby”, it means attributes should match as much as possible. According to the greedy intrinsic of ID3/C4.5 decision tree, the node close to root should have higher weight on distance as they are of higher information gain (not sure if this is not compulsory).

For example: consider such a decision tree on two attributes:

B is impure and thus is to be evaluated by knn, here we take *k=5* for demonstration. A is the nearest nodes since the parent of parent node is same *(Age < 20)*. D is the second nearest because it has one node as same (Female). C is the furthest node because none of the attribute is same.

So, B node should be considered ‘student’ when k = 5.

In knn perspective, a corresponding demonstration on sample space should look like this: 

To sum up, for this univariate-trees-based hybrid model, the decision boundary should still be axis-parallel because knn is only applied to predict the value of the leaf nodes when constructing the tree. Only leaf nodes will be affected (disregarding later post-pruning). The shape of decision boundary should be very similar to that of DT. Intuitively, however, its boundary maybe flattened (less “zig-zag”) compared to the original tree, as leaf nodes are much more likely to have same value as its nearby nodes due to knn method. It suggests better generalization performance.