

Attributes setting of the random forest model:

Number of trees: 99

Number of features: According some resources from the internet, $\sqrt{\text{total features}}$ seems like a proper number, so I chose 5.

Number of instances to build each tree: variable 'sample_size' is set to 0.8

Min_samples_split: 200

Max_depth: 3

Difficulty I encountered and how I solved them: In my opinion, this assignment is a little hard, since implementing all these functions requires full understanding of what is going on. After studying each step several times, I finally could confirm my understanding about decision trees. Another difficulty is that debugging took me lots of time because I'm not familiar with python, so some bugs I step into, such as difference between shallow copy and deep copy, and after using 'loc' function, index still remains. So 'reset_index' function is required in order to get a correct result. Hope I could get more comfortable when coding python by doing these assignments.

Reflections: Also liked this assignment because it's dealing with real world problems. Additionally, I thought that breaking the whole implementation of decision trees into separate functions is quite good for some reasons. First, it makes debugging much more easier since when one part is wrong, you can focus on a single function and try to find the errors. Second, implementing these functions separately makes the whole problem more solvable compared to putting all the stuff in a single function since the latter way is kind of overwhelming. So I actually learned a lot in terms of how to split the whole process into separate parts thanks to this assignment.