Part1. Decision Tree

```
kahng-01 ~/my_scratch/web/Codes/CS534/HW4 1000$ python3 part1.py
training accuracy with depth: 2 is : 75.5223880597015
validation accuracy with depth: 2 is : 75.91919191919192
training accuracy with depth: 5 is : 78.43781094527364
validation accuracy with depth: 5 is : 78.64646464646464
training accuracy with depth: 10 is : 79.50248756218905
validation accuracy with depth: 10 is : 78.30303030303031
training accuracy with depth: 20 is : 83.66169154228855
validation accuracy with depth: 20 is : 77.55555555555556
training accuracy with depth: 25 is : 86.49751243781094
validation accuracy with depth: 25 is : 76.86868686868686
training accuracy with depth: 30 is : 89.32338308457712
validation accuracy with depth: 30 is : 75.31313131313131
training accuracy with depth: 35 is : 91.39303482587064
validation accuracy with depth: 35 is : 75.15151515151514
training accuracy with depth: 40 is : 92.98507462686567
validation accuracy with depth: 40 is : 74.34343434343434
training accuracy with depth: 45 is : 94.0497512437811
validation accuracy with depth: 45 is : 73.8989898989899
training accuracy with depth: 50 is : 95.2139303482587
validation accuracy with depth: 50 is : 73.45454545454545
```

Figure 1. Decision Tree implementation

(a) What are the first three splits selected by your algorithm? This is for the root, and the two splits immediately beneath the root. What are their respective information gains?

```
{   'depth': 0,
    'information_gain': 0.3075677298204409,
    'label': 1,
    'leaf': False,
    'left': {   'depth': 1,
                'information_gain': 0.03490775046162631,
                'label': 1,
                'leaf': False,
                'left': None,
                'node': 'Vehicle_Damage',
                'node_index': 3,
                'right': None},
    'node': 'Previously_Insured',
    'node_index': 2,
    'right': {   'depth': 1,
                 'information_gain': 0.004327417692095228,
                 'label': 0,
                 'leaf': False,
                 'left': None,
                 'node': 'Vehicle_Damage',
                 'node_index': 3,
                 'right': None}}
training accuracy with depth: 2 is : 75.5223880597015
validation accuracy with depth: 2 is : 75.91919191919192
```

Figure 2. Depth 2 tree node

The figure 1 is the result of decision tree with depth 2. To print out the tree structure, I used the dictionary file format. In the root, appeared with 'depth: 0', the split node is 'Previously_Insured' and its information gain is 0.3076. The split criteria that I used is if the value of feature is zero then go to the left node, and if the value of feature is one then go to the right node. Both nodes after the

first splits, immediately beneath the root, are 'Vehicle_Damage'. The information gain of left 'Vehicle_Damage' node is 0.0349, and the information gain of right 'Vehicle_Damage' node is 0.0043.

(b) Evaluate and plot the training and validation accuracies of your trees as a function for *dmax*. When do you see your tree start overfitting?

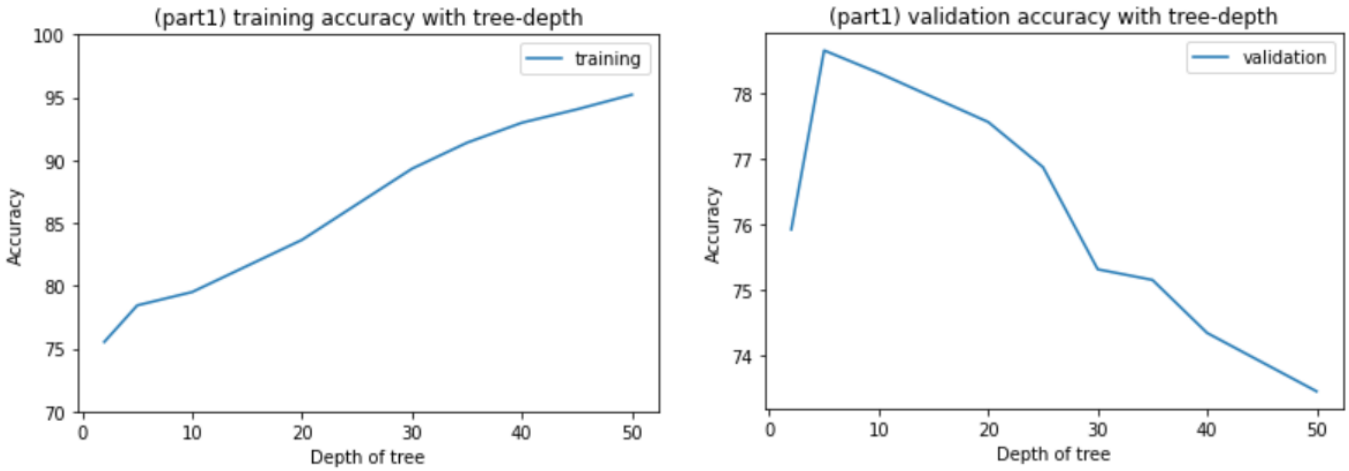

Figure 3. Each Training and Validation accuracy according to the depth of the tree

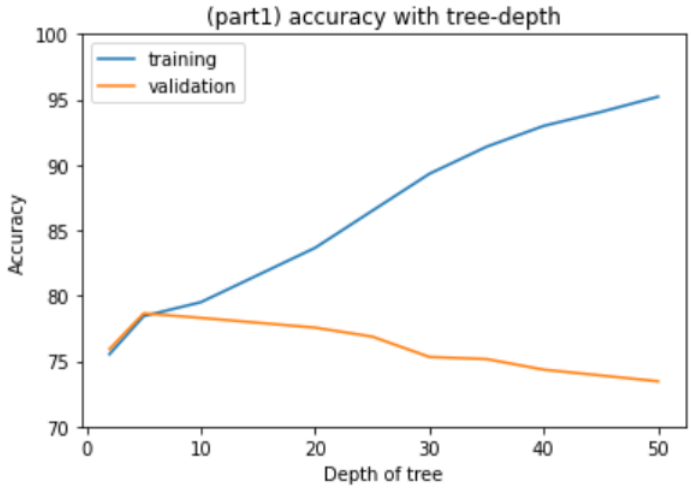

Figure 4. Both Training and Validation accuracy according to the depth of the tree

In the figure 3, left figure is training dataset accuracy and right side is validation dataset accuracy. When see the left figure, the accuracy of training dataset is tending to get higher as the depth of the tree gets higher. However, in the right figure, the accuracy of validation dataset is decreasing after about depth 5 of decision tree. If we compare both accuracies on figure 4, both training and validation accuracy gets higher until the depth of tree is about 5, and after that validation accuracy is decreasing. Therefore, we can say that this tree start overfitting after depth 5.

Part2. Random Forest

```
==========================================================
train accuracy: d_max : 2, m : 5 with tree num : 10 = 72.80597014925372
valid accuracy: d_max : 2, m : 5 with tree num : 10 = 73.13131313131314
train accuracy: d_max : 2, m : 5 with tree num : 20 = 67.55223880597015
valid accuracy: d_max : 2, m : 5 with tree num : 20 = 67.4949494949495
train accuracy: d_max : 2, m : 5 with tree num : 30 = 68.84577114427861
valid accuracy: d_max : 2, m : 5 with tree num : 30 = 68.76767676767676
train accuracy: d_max : 2, m : 5 with tree num : 40 = 65.7412935323383
valid accuracy: d_max : 2, m : 5 with tree num : 40 = 65.85858585858585
train accuracy: d_max : 2, m : 5 with tree num : 50 = 64.6766169154229
valid accuracy: d_max : 2, m : 5 with tree num : 50 = 64.88888888888889
train accuracy: d_max : 2, m : 5 with tree num : 60 = 66.57711442786069
valid accuracy: d_max : 2, m : 5 with tree num : 60 = 66.82828282828282
train accuracy: d_max : 2, m : 5 with tree num : 70 = 65.50248756218906
valid accuracy: d_max : 2, m : 5 with tree num : 70 = 65.67676767676768
train accuracy: d_max : 2, m : 5 with tree num : 80 = 65.70149253731343
valid accuracy: d_max : 2, m : 5 with tree num : 80 = 65.81818181818181
train accuracy: d_max : 2, m : 5 with tree num : 90 = 65.23383084577115
valid accuracy: d_max : 2, m : 5 with tree num : 90 = 65.39393939393939
train accuracy: d_max : 2, m : 5 with tree num : 100 = 65.31343283582089
valid accuracy: d_max : 2, m : 5 with tree num : 100 = 65.33333333333333
```

```
==========================================================
train accuracy: d_max : 2, m : 25 with tree num : 10 = 63.03482587064677
valid accuracy: d_max : 2, m : 25 with tree num : 10 = 63.21212121212121
train accuracy: d_max : 2, m : 25 with tree num : 20 = 73.53233830845771
valid accuracy: d_max : 2, m : 25 with tree num : 20 = 73.37373737373737
train accuracy: d_max : 2, m : 25 with tree num : 30 = 72.50746268656715
valid accuracy: d_max : 2, m : 25 with tree num : 30 = 72.74747474747475
train accuracy: d_max : 2, m : 25 with tree num : 40 = 72.26865671641791
valid accuracy: d_max : 2, m : 25 with tree num : 40 = 72.1010101010101
train accuracy: d_max : 2, m : 25 with tree num : 50 = 74.85572139303483
valid accuracy: d_max : 2, m : 25 with tree num : 50 = 74.84848484848486
train accuracy: d_max : 2, m : 25 with tree num : 60 = 75.08457711442786
valid accuracy: d_max : 2, m : 25 with tree num : 60 = 75.01010101010101
train accuracy: d_max : 2, m : 25 with tree num : 70 = 72.64676616915423
valid accuracy: d_max : 2, m : 25 with tree num : 70 = 72.7878787878788
train accuracy: d_max : 2, m : 25 with tree num : 80 = 73.33333333333333
valid accuracy: d_max : 2, m : 25 with tree num : 80 = 73.43434343434343
train accuracy: d_max : 2, m : 25 with tree num : 90 = 70.44776119402985
valid accuracy: d_max : 2, m : 25 with tree num : 90 = 70.76767676767676
train accuracy: d_max : 2, m : 25 with tree num : 100 = 68.74626865671641
valid accuracy: d_max : 2, m : 25 with tree num : 100 = 69.31313131313132
```

```
==========================================================
train accuracy: d_max : 2, m : 50 with tree num : 10 = 73.75124378109452
valid accuracy: d_max : 2, m : 50 with tree num : 10 = 73.91919191919192
train accuracy: d_max : 2, m : 50 with tree num : 20 = 75.62189054726367
valid accuracy: d_max : 2, m : 50 with tree num : 20 = 75.83838383838383
train accuracy: d_max : 2, m : 50 with tree num : 30 = 75.4726368159204
valid accuracy: d_max : 2, m : 50 with tree num : 30 = 75.43434343434343
train accuracy: d_max : 2, m : 50 with tree num : 40 = 75.66169154228855
valid accuracy: d_max : 2, m : 50 with tree num : 40 = 75.61616161616162
train accuracy: d_max : 2, m : 50 with tree num : 50 = 75.68159203980099
valid accuracy: d_max : 2, m : 50 with tree num : 50 = 75.67676767676768
train accuracy: d_max : 2, m : 50 with tree num : 60 = 75.67164179104478
valid accuracy: d_max : 2, m : 50 with tree num : 60 = 75.65656565656566
train accuracy: d_max : 2, m : 50 with tree num : 70 = 75.68159203980099
valid accuracy: d_max : 2, m : 50 with tree num : 70 = 75.67676767676768
train accuracy: d_max : 2, m : 50 with tree num : 80 = 75.67164179104478
valid accuracy: d_max : 2, m : 50 with tree num : 80 = 75.65656565656566
train accuracy: d_max : 2, m : 50 with tree num : 90 = 75.67164179104478
valid accuracy: d_max : 2, m : 50 with tree num : 90 = 75.65656565656566
train accuracy: d_max : 2, m : 50 with tree num : 100 = 75.83084577114428
valid accuracy: d_max : 2, m : 50 with tree num : 100 = 75.91919191919192
```

```
=========================================================
train accuracy: d_max : 2, m : 100 with tree num : 10 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 10 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 20 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 20 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 30 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 30 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 40 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 40 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 50 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 50 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 60 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 60 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 70 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 70 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 80 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 80 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 90 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 90 = 75.91919191919192
train accuracy: d_max : 2, m : 100 with tree num : 100 = 75.5223880597015
valid accuracy: d_max : 2, m : 100 with tree num : 100 = 75.91919191919192
```

Figure 5. Random Forest result with d_max = 2

```
=========================================================
train accuracy: d_max : 10, m : 5 with tree num : 10 = 74.88557213930348
valid accuracy: d_max : 10, m : 5 with tree num : 10 = 74.08080808080808
train accuracy: d_max : 10, m : 5 with tree num : 20 = 75.1542288557214
valid accuracy: d_max : 10, m : 5 with tree num : 20 = 74.96969696969697
train accuracy: d_max : 10, m : 5 with tree num : 30 = 75.31343283582089
valid accuracy: d_max : 10, m : 5 with tree num : 30 = 74.68686868686869
train accuracy: d_max : 10, m : 5 with tree num : 40 = 74.98507462686567
valid accuracy: d_max : 10, m : 5 with tree num : 40 = 74.56565656565657
train accuracy: d_max : 10, m : 5 with tree num : 50 = 76.02985074626865
valid accuracy: d_max : 10, m : 5 with tree num : 50 = 75.71717171717171
train accuracy: d_max : 10, m : 5 with tree num : 60 = 76.32835820895522
valid accuracy: d_max : 10, m : 5 with tree num : 60 = 76.20202020202021
train accuracy: d_max : 10, m : 5 with tree num : 70 = 75.98009950248756
valid accuracy: d_max : 10, m : 5 with tree num : 70 = 75.79797979797979
train accuracy: d_max : 10, m : 5 with tree num : 80 = 75.13432835820896
valid accuracy: d_max : 10, m : 5 with tree num : 80 = 74.82828282828284
train accuracy: d_max : 10, m : 5 with tree num : 90 = 74.09950248756219
valid accuracy: d_max : 10, m : 5 with tree num : 90 = 73.81818181818181
train accuracy: d_max : 10, m : 5 with tree num : 100 = 73.7412935323383
valid accuracy: d_max : 10, m : 5 with tree num : 100 = 73.43434343434343
```

```
=========================================================
train accuracy: d_max : 10, m : 25 with tree num : 10 = 79.0547263681592
valid accuracy: d_max : 10, m : 25 with tree num : 10 = 79.0909090909091
train accuracy: d_max : 10, m : 25 with tree num : 20 = 78.92537313432835
valid accuracy: d_max : 10, m : 25 with tree num : 20 = 79.01010101010101
train accuracy: d_max : 10, m : 25 with tree num : 30 = 78.8955223880597
valid accuracy: d_max : 10, m : 25 with tree num : 30 = 78.98989898989899
train accuracy: d_max : 10, m : 25 with tree num : 40 = 78.98507462686567
valid accuracy: d_max : 10, m : 25 with tree num : 40 = 79.11111111111111
train accuracy: d_max : 10, m : 25 with tree num : 50 = 79.08457711442786
valid accuracy: d_max : 10, m : 25 with tree num : 50 = 79.0909090909091
train accuracy: d_max : 10, m : 25 with tree num : 60 = 79.12437810945273
valid accuracy: d_max : 10, m : 25 with tree num : 60 = 79.13131313131314
train accuracy: d_max : 10, m : 25 with tree num : 70 = 79.07462686567163
valid accuracy: d_max : 10, m : 25 with tree num : 70 = 79.11111111111111
train accuracy: d_max : 10, m : 25 with tree num : 80 = 79.14427860696517
valid accuracy: d_max : 10, m : 25 with tree num : 80 = 79.11111111111111
train accuracy: d_max : 10, m : 25 with tree num : 90 = 79.12437810945273
valid accuracy: d_max : 10, m : 25 with tree num : 90 = 79.15151515151516
train accuracy: d_max : 10, m : 25 with tree num : 100 = 79.11442786069651
valid accuracy: d_max : 10, m : 25 with tree num : 100 = 79.11111111111111
```

```
==========================================================
train accuracy: d_max : 10, m : 50 with tree num : 10 = 78.87562189054727
valid accuracy: d_max : 10, m : 50 with tree num : 10 = 79.01010101010101
train accuracy: d_max : 10, m : 50 with tree num : 20 = 78.92537313432835
valid accuracy: d_max : 10, m : 50 with tree num : 20 = 79.03030303030303
train accuracy: d_max : 10, m : 50 with tree num : 30 = 79.09452736318407
valid accuracy: d_max : 10, m : 50 with tree num : 30 = 79.03030303030303
train accuracy: d_max : 10, m : 50 with tree num : 40 = 79.11442786069651
valid accuracy: d_max : 10, m : 50 with tree num : 40 = 78.98989898989899
train accuracy: d_max : 10, m : 50 with tree num : 50 = 79.1542288557214
valid accuracy: d_max : 10, m : 50 with tree num : 50 = 78.98989898989899
train accuracy: d_max : 10, m : 50 with tree num : 60 = 79.1542288557214
valid accuracy: d_max : 10, m : 50 with tree num : 60 = 79.03030303030303
train accuracy: d_max : 10, m : 50 with tree num : 70 = 79.14427860696517
valid accuracy: d_max : 10, m : 50 with tree num : 70 = 79.07070707070707
train accuracy: d_max : 10, m : 50 with tree num : 80 = 79.18407960199005
valid accuracy: d_max : 10, m : 50 with tree num : 80 = 79.07070707070707
train accuracy: d_max : 10, m : 50 with tree num : 90 = 79.22388059701493
valid accuracy: d_max : 10, m : 50 with tree num : 90 = 79.13131313131314
train accuracy: d_max : 10, m : 50 with tree num : 100 = 79.22388059701493
valid accuracy: d_max : 10, m : 50 with tree num : 100 = 79.17171717171718
```

```
==========================================================
train accuracy: d_max : 10, m : 100 with tree num : 10 = 79.28358208955224
valid accuracy: d_max : 10, m : 100 with tree num : 10 = 79.01010101010101
train accuracy: d_max : 10, m : 100 with tree num : 20 = 79.48258706467661
valid accuracy: d_max : 10, m : 100 with tree num : 20 = 78.92929292929293
train accuracy: d_max : 10, m : 100 with tree num : 30 = 79.46268656716417
valid accuracy: d_max : 10, m : 100 with tree num : 30 = 79.17171717171718
train accuracy: d_max : 10, m : 100 with tree num : 40 = 79.67164179104478
valid accuracy: d_max : 10, m : 100 with tree num : 40 = 79.25252525252525
train accuracy: d_max : 10, m : 100 with tree num : 50 = 79.73134328358209
valid accuracy: d_max : 10, m : 100 with tree num : 50 = 79.1919191919192
train accuracy: d_max : 10, m : 100 with tree num : 60 = 79.69154228855722
valid accuracy: d_max : 10, m : 100 with tree num : 60 = 79.31313131313131
train accuracy: d_max : 10, m : 100 with tree num : 70 = 79.67164179104478
valid accuracy: d_max : 10, m : 100 with tree num : 70 = 79.17171717171718
train accuracy: d_max : 10, m : 100 with tree num : 80 = 79.66169154228857
valid accuracy: d_max : 10, m : 100 with tree num : 80 = 79.25252525252525
train accuracy: d_max : 10, m : 100 with tree num : 90 = 79.62189054726369
valid accuracy: d_max : 10, m : 100 with tree num : 90 = 79.41414141414141
train accuracy: d_max : 10, m : 100 with tree num : 100 = 79.61194029850746
valid accuracy: d_max : 10, m : 100 with tree num : 100 = 79.37373737373737
```

Figure 6. Random Forest result with d_max = 10

```
==========================================================
train accuracy: d_max : 25, m : 5 with tree num : 10 = 79.80099502487562
valid accuracy: d_max : 25, m : 5 with tree num : 10 = 77.55555555555556
train accuracy: d_max : 25, m : 5 with tree num : 20 = 80.49751243781095
valid accuracy: d_max : 25, m : 5 with tree num : 20 = 78.12121212121212
train accuracy: d_max : 25, m : 5 with tree num : 30 = 80.13930348258707
valid accuracy: d_max : 25, m : 5 with tree num : 30 = 78.26262626262627
train accuracy: d_max : 25, m : 5 with tree num : 40 = 80.09950248756219
valid accuracy: d_max : 25, m : 5 with tree num : 40 = 78.24242424242425
train accuracy: d_max : 25, m : 5 with tree num : 50 = 80.05970149253731
valid accuracy: d_max : 25, m : 5 with tree num : 50 = 78.32323232323233
train accuracy: d_max : 25, m : 5 with tree num : 60 = 79.97014925373135
valid accuracy: d_max : 25, m : 5 with tree num : 60 = 78.28282828282829
train accuracy: d_max : 25, m : 5 with tree num : 70 = 79.98009950248756
valid accuracy: d_max : 25, m : 5 with tree num : 70 = 78.28282828282829
train accuracy: d_max : 25, m : 5 with tree num : 80 = 79.99004975124379
valid accuracy: d_max : 25, m : 5 with tree num : 80 = 78.32323232323233
train accuracy: d_max : 25, m : 5 with tree num : 90 = 79.90049751243781
valid accuracy: d_max : 25, m : 5 with tree num : 90 = 78.14141414141415
train accuracy: d_max : 25, m : 5 with tree num : 100 = 79.83084577114428
valid accuracy: d_max : 25, m : 5 with tree num : 100 = 78.04040404040404
```

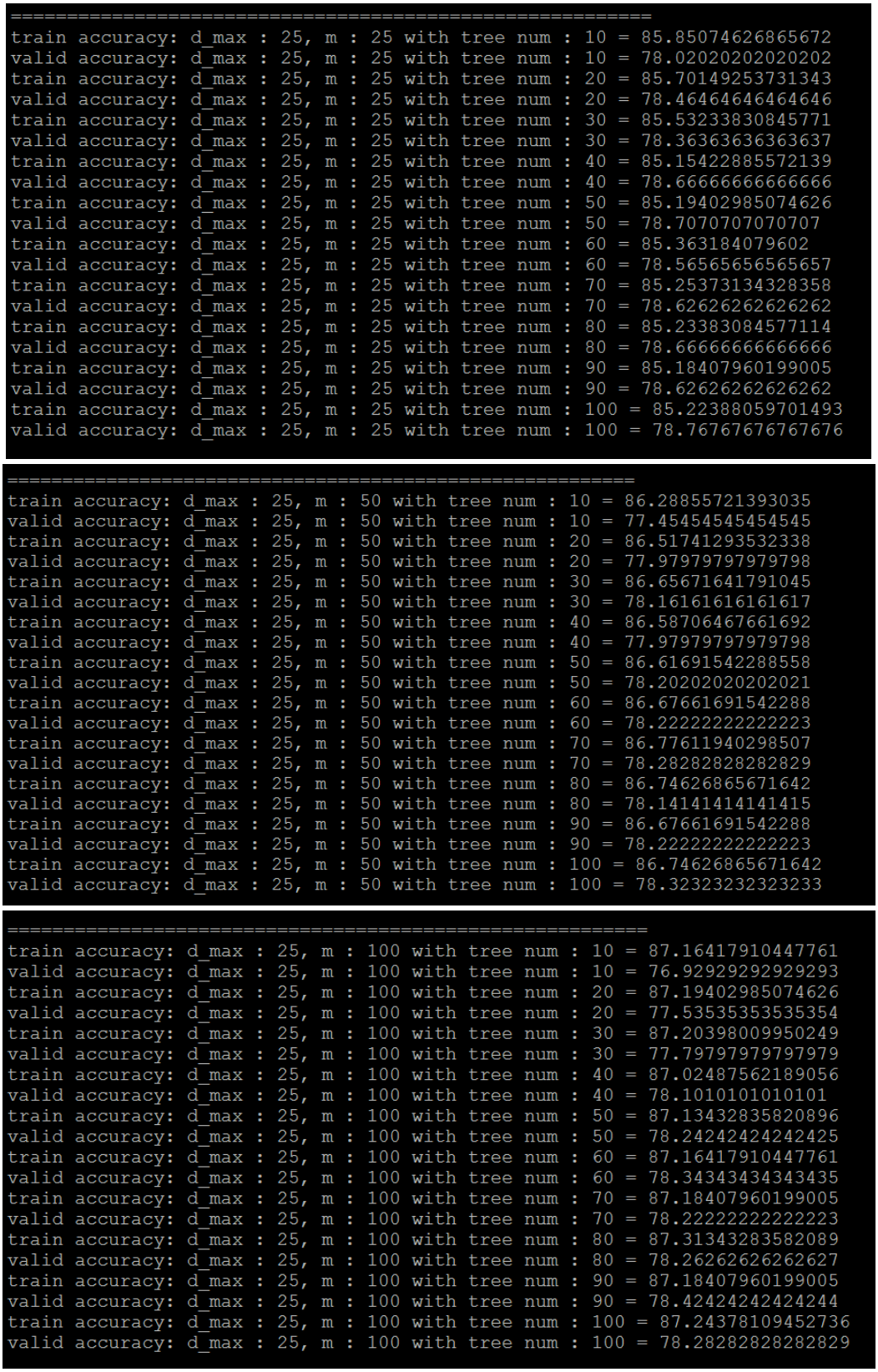

Figure 7. Random Forest result with d_max = 25

(a) Compare your training curves with the validation curves, do you think your model is overfitting or underfitting for particular parameter combinations? And why?

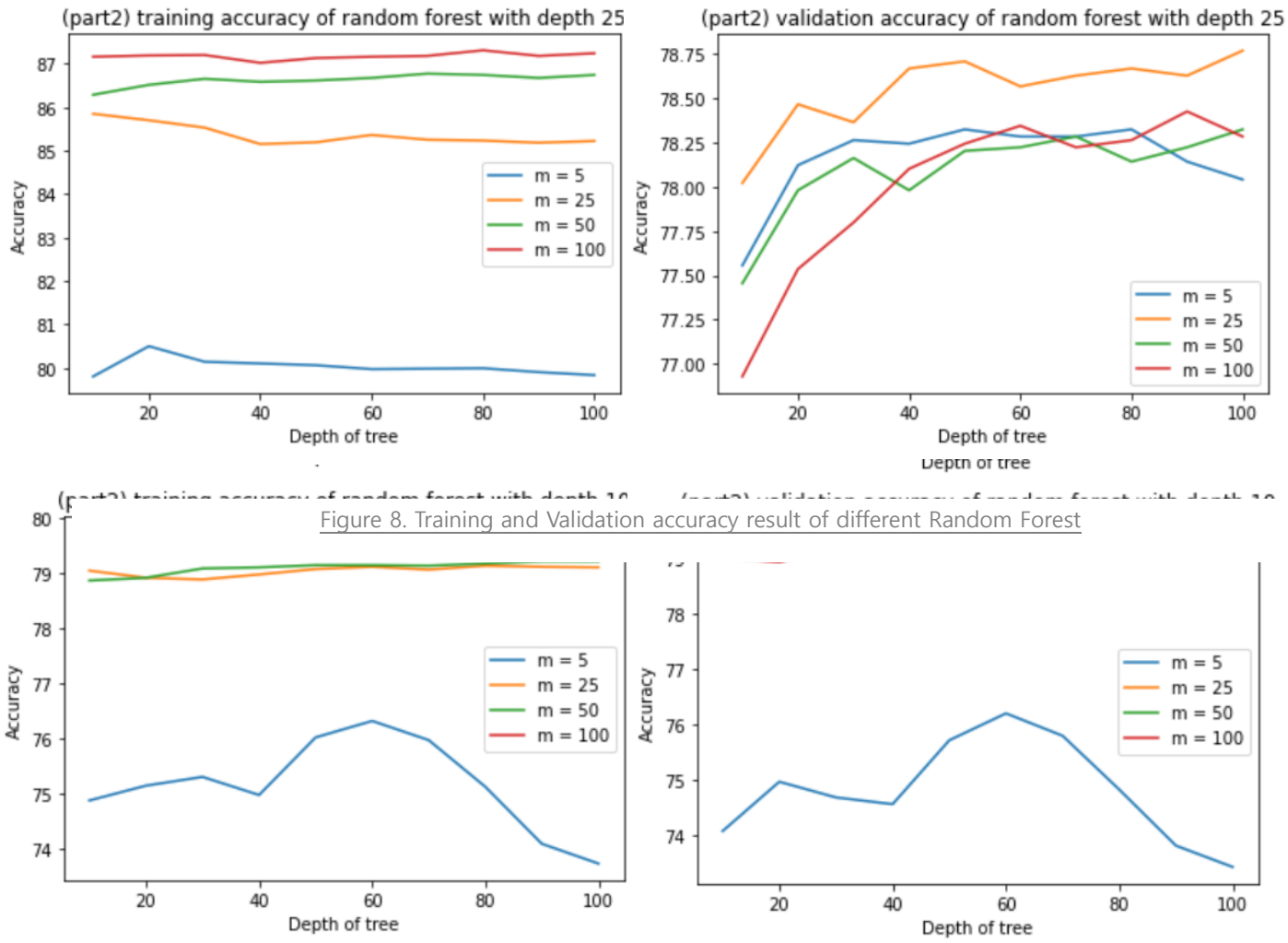

Figure 8. Training and Validation accuracy result of different Random Forest

In figure 8, figures on the left side represent training accuracy according to the depth of tree, and right side represent validation accuracy according to the depth of tree. I think random forest with depth 2 with different values of the number of features selected in each step shows some underfitting. If we compare the training accuracy and validation accuracy of random forest with depth 2, we can see that both accuracies show almost same result. Also, random forest with depth 25 with different number of selected features in each step such as 100 or 50 is shows overfitting. If we compare the training accuracy and validation accuracy of this random forest, we can see that validation accuracy is getting lower as the training accuracy gets higher. This result is showing some overfitting.

(b)

Random forest with depth 2 is showing underfitting result, validation accuracy shows very high accuracy. It means this model is predicting the new data too well. This is because even though there are lots of features to consider when training, tree with depth 2 only compare 2 features, and it will cause too biased trained model. Therefore, it will predict data with biased model and could cause the underfitting result. On the other hands, random forest with depth 25 with high m values are showing overfitting result. It means that this model is not good for predicting new data. This is because it considers too many features which leads to a high variance trained model. The model overly trains the features that are not even common in one decision boundary. So, the model will be good at predicting trained data but not with new data.

Therefore, based on the two result, we can see that model that is trained with too small number of features will be highly biased and show low variance. However, model that is trained with too many features will show high variance result but with low bias. Both models will have high errors. Then we can train this random forest tree model with medium amount split criteria features because it will show low biased and low variance and will generate acceptable and understandable validation accuracy.

If we create large depth tree, the more the depth gets dipper, there will be smaller number of data to be split. Then it will cause the model to be more specific and highly variance, and this will make the performance increase difficult.