

Assignment 6

Swapnil Asawa, swa12@pitt.edu

Question 1.

Part a. Done

Part b:

Epoches	0 hidden layer, only logistic regression output	2 neurons in one hidden layer	3 neurons in one hidden layer	5 neurons in one hidden layer	10 neurons in one hidden layer
Mean misclassification error train	0.209606986899563,	0.205240174672489,	0.222707423580786	0.240174672489083,	0.278293135435993
Mean misclassification error test	0.243042671614100	0.226345083487941	0.239332096474954	0.233766233766234	0.227074235807860

Everytime the models were giving different values of mean train error and test error. The value is varying so much that the best network out of above cannot be found. The minimum error comes with 2 neurons in one hidden layer in the above case but if I again call the function, the values may go up making any other network to be better. So I would like to say this much knowledge is insufficient to make comment on which network is better. More training data, better weight initialization, changing learning rate, etc could be few of the many ways to invest further.

Optional:

The experiments are performed as follows:

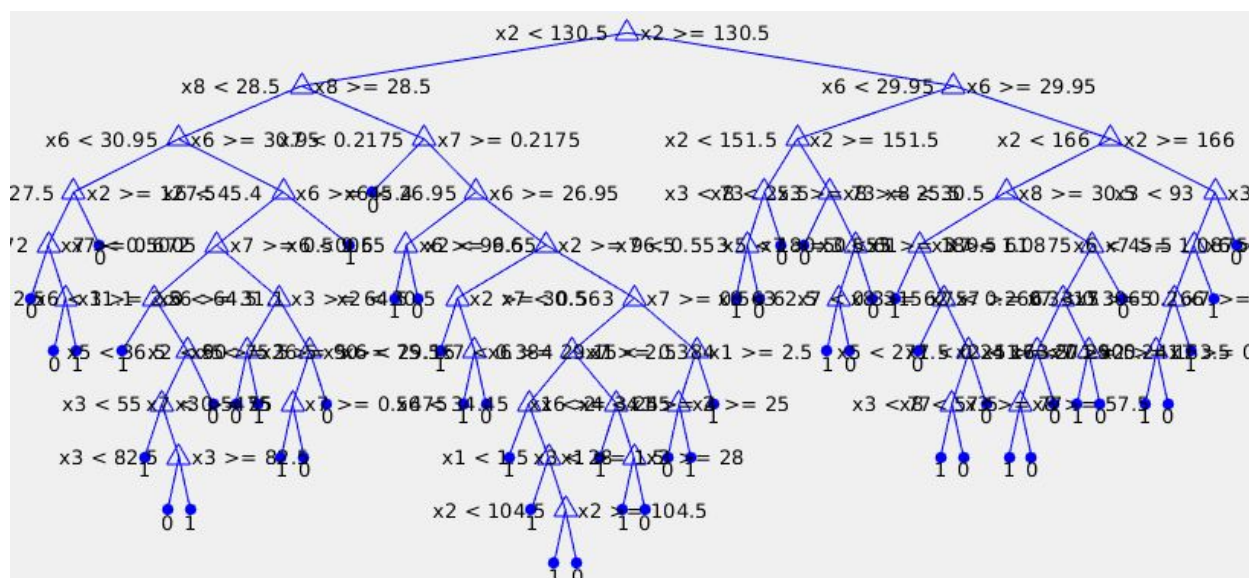
Pattern et paramete rs	5 2	2 2	10 5 2	5 2	5 2	5 5 2	5 5 2
Activatio n of hidden layer	tansig	tansig	tansig	logsig	tansig	tansig	tansig
epochs	2000	2000	2000	2000	20000	20000	2000
Mean train error	0.248908 2969432 31	0.254174 3970315 40	0.278293 1354359 93	0.628942 4860853 43	0.359925 7884972 17	0.231910 9461966 60	0.246753 2467532 47
Mean test error	0.261595 5473098 33	0.262008 7336244 54	0.283842 7947598 25	0.703056 7685589 52	0.318777 2925764 19	0.222707 4235807 86	0.262008 7336244 54

Changing the activation to tan sigmoid did worse.

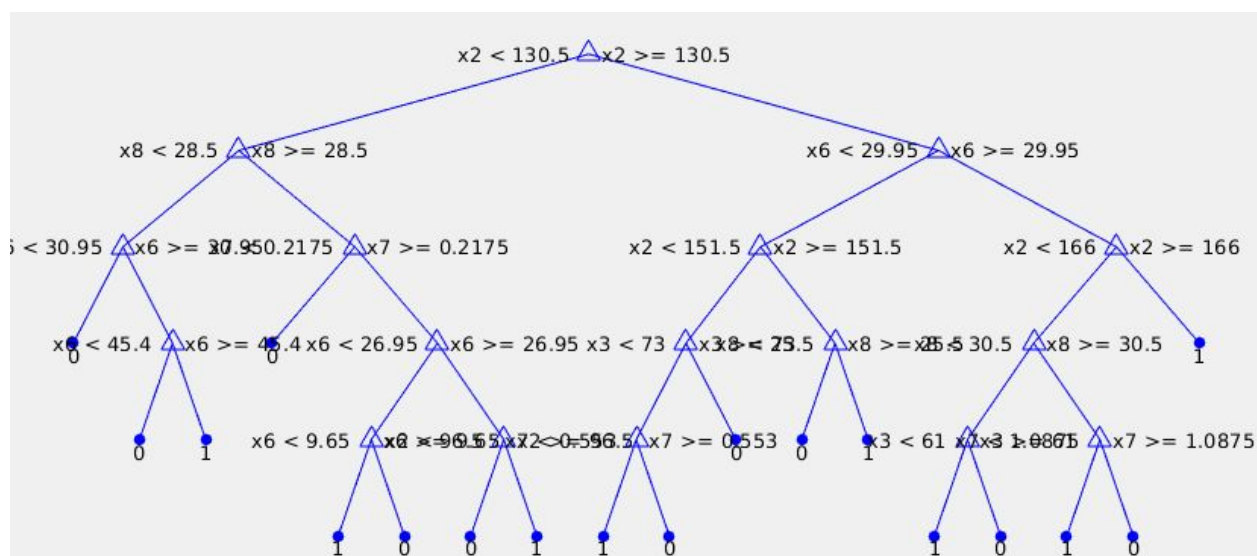
Not much can be inferred with limited number of experiments. Maybe more training data can help better. Right now we only have 530 around training instances.

Question 2

Part a



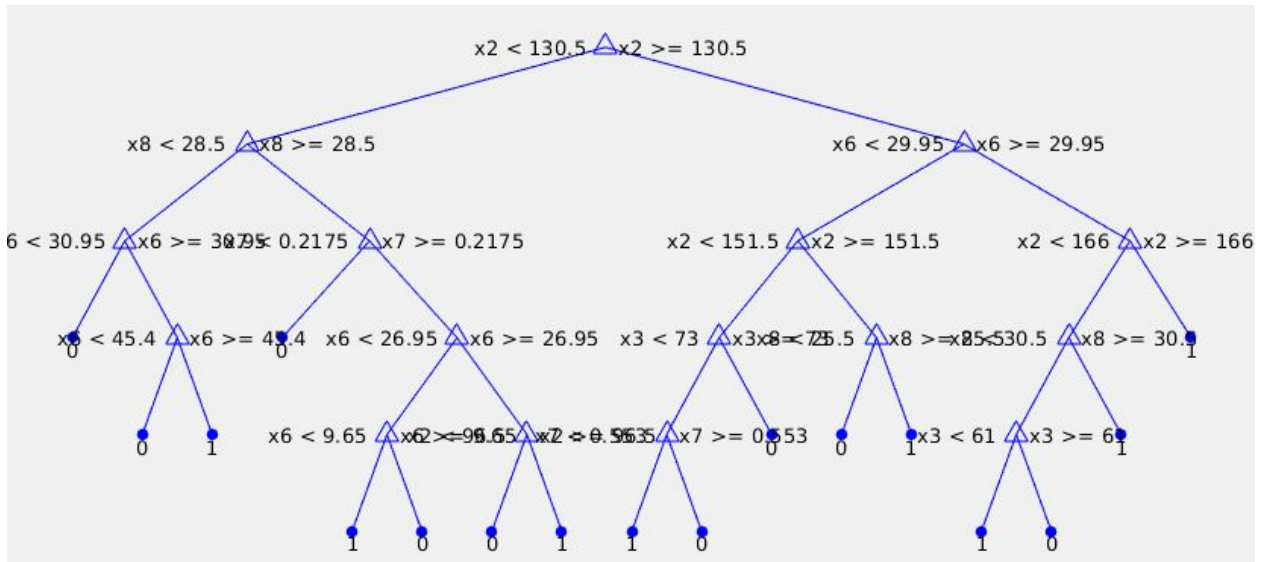
Restricted:



The restricted tree is better because it is giving better accuracy/less error(.2576) on test set than the unrestricted model(.2751). We should try to back prune and check the test error. Until the test error does not increase much, we should back prune it. In this way we avoid model to overfit by reducing its unnecessary complexity which we always try to penalize. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting. We should also use measure of impurity as well.

Part b

1. `new_tree=fitctree(x,y,'MaxNumSplits',20,
'MinParentSize',10,'MinLeafSize',2,'splitcriterion','gdi');`
Test error: 0.2445



Decision tree for classification

- 1 if $x_2 < 130.5$ then node 2 elseif $x_2 \geq 130.5$ then node 3 else 0
- 2 if $x_8 < 28.5$ then node 4 elseif $x_8 \geq 28.5$ then node 5 else 0
- 3 if $x_6 < 29.95$ then node 6 elseif $x_6 \geq 29.95$ then node 7 else 1
- 4 if $x_6 < 30.95$ then node 8 elseif $x_6 \geq 30.95$ then node 9 else 0
- 5 if $x_7 < 0.2175$ then node 10 elseif $x_7 \geq 0.2175$ then node 11 else 0
- 6 if $x_2 < 151.5$ then node 12 elseif $x_2 \geq 151.5$ then node 13 else 0
- 7 if $x_2 < 166$ then node 14 elseif $x_2 \geq 166$ then node 15 else 1
- 8 class = 0
- 9 if $x_6 < 45.4$ then node 16 elseif $x_6 \geq 45.4$ then node 17 else 0
- 10 class = 0
- 11 if $x_6 < 26.95$ then node 18 elseif $x_6 \geq 26.95$ then node 19 else 0
- 12 if $x_3 < 73$ then node 20 elseif $x_3 \geq 73$ then node 21 else 0
- 13 if $x_8 < 25.5$ then node 22 elseif $x_8 \geq 25.5$ then node 23 else 1
- 14 if $x_8 < 30.5$ then node 24 elseif $x_8 \geq 30.5$ then node 25 else 1
- 15 class = 1
- 16 class = 0
- 17 class = 1
- 18 if $x_6 < 9.65$ then node 26 elseif $x_6 \geq 9.65$ then node 27 else 0
- 19 if $x_2 < 96.5$ then node 28 elseif $x_2 \geq 96.5$ then node 29 else 1
- 20 if $x_7 < 0.553$ then node 30 elseif $x_7 \geq 0.553$ then node 31 else 0
- 21 class = 0
- 22 class = 0
- 23 class = 1
- 24 if $x_3 < 61$ then node 32 elseif $x_3 \geq 61$ then node 33 else 0
- 25 class = 1
- 26 class = 1

2.

```

graph TD
    Root["x2 < 130.5"] -->|Yes| Node1["x8 < 28.5"]
    Root -->|No| Node2["x6 < 29.95"]
    Node1 -->|Yes| Node3["x6 < 30.95"]
    Node1 -->|No| Node4["x7 >= 0.2175"]
    Node2 -->|Yes| Node5["x2 < 151.5"]
    Node2 -->|No| Node6["x2 < 166"]
    Node3 -->|Yes| Leaf3_0((0))
    Node3 -->|No| Node7["x6 < 45.4"]
    Node4 -->|Yes| Leaf4_0((0))
    Node4 -->|No| Node8["x2 < 96.5"]
    Node5 -->|Yes| Leaf5_0((0))
    Node5 -->|No| Node9["x8 < 25.5"]
    Node6 -->|Yes| Node10["x8 < 30.5"]
    Node6 -->|No| Leaf6_1((1))
    Node7 -->|Yes| Leaf7_1((1))
    Node7 -->|No| Leaf7_0((0))
    Node8 -->|Yes| Leaf8_0((0))
    Node8 -->|No| Leaf8_1((1))
    Node9 -->|Yes| Leaf9_0((0))
    Node9 -->|No| Leaf9_1((1))
    Node10 -->|Yes| Leaf10_1((1))
    Node10 -->|No| Leaf10_0((0))
  
```

```

1 if x2<130.5 then node 2 elseif x2>=130.5 then node 3 else 0
2 if x8<28.5 then node 4 elseif x8>=28.5 then node 5 else 0
3 if x6<29.95 then node 6 elseif x6>=29.95 then node 7 else 1
4 if x6<30.95 then node 8 elseif x6>=30.95 then node 9 else 0
5 if x7<0.2175 then node 10 elseif x7>=0.2175 then node 11 else 0
6 if x2<151.5 then node 12 elseif x2>=151.5 then node 13 else 0
7 if x2<166 then node 14 elseif x2>=166 then node 15 else 1
8 class = 0
9 if x6<45.4 then node 16 elseif x6>=45.4 then node 17 else 0
10 class = 0
11 if x6<26.95 then node 18 elseif x6>=26.95 then node 19 else 0
12 class = 0
13 if x8<25.5 then node 20 elseif x8>=25.5 then node 21 else 1

```

```

14 if x8<30.5 then node 22 elseif x8>=30.5 then node 23 else 1
15 class = 1
16 class = 0
17 class = 1
18 class = 0
19 if x2<96.5 then node 24 elseif x2>=96.5 then node 25 else 1
20 class = 0
21 class = 1
22 if x3<61 then node 26 elseif x3>=61 then node 27 else 0
23 class = 1
24 class = 0
25 class = 1
26 class = 1
27 class = 0

```

Test error = 0.2227

3. MaxNumSplits',16

4. MaxNumSplits',18

5.MaxNumSplits',19

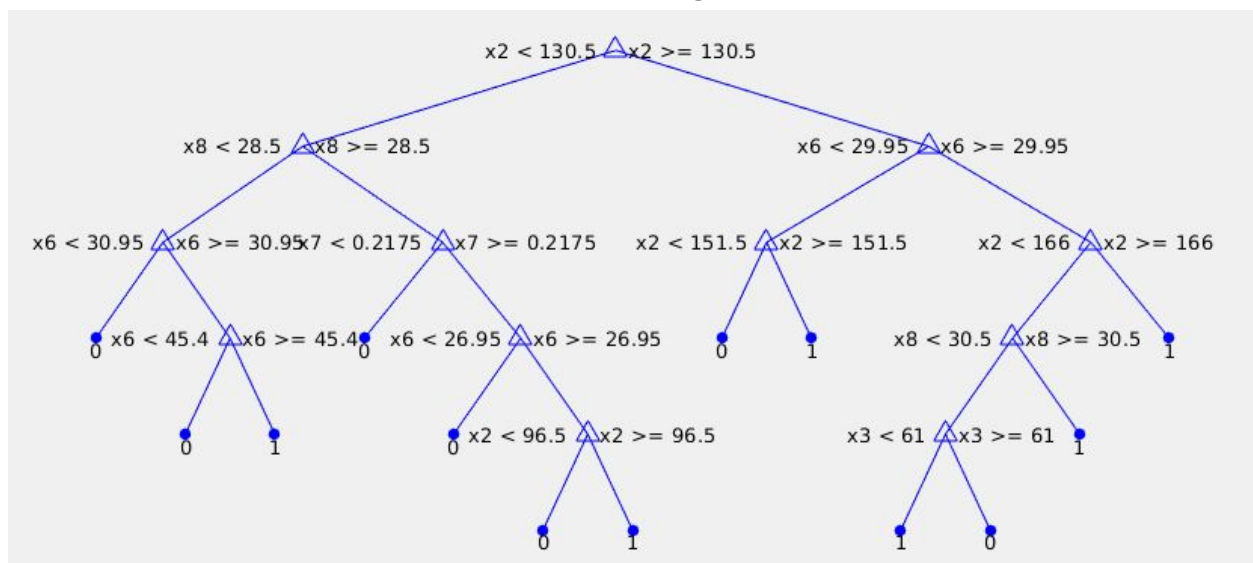
17 is the best max num split I got after trying many different numbers. With least complexity and least error.

6.

```

new_tree=fitctree(x,y,'MaxNumSplits',17,
'MinParentSize',20,'MinLeafSize',4,'splitcriterion','gdi');

```



Test error =0.2227

Decision tree for classification

```

1 if x2<130.5 then node 2 elseif x2>=130.5 then node 3 else 0
2 if x8<28.5 then node 4 elseif x8>=28.5 then node 5 else 0
3 if x6<29.95 then node 6 elseif x6>=29.95 then node 7 else 1
4 if x6<30.95 then node 8 elseif x6>=30.95 then node 9 else 0
5 if x7<0.2175 then node 10 elseif x7>=0.2175 then node 11 else 0
6 if x2<151.5 then node 12 elseif x2>=151.5 then node 13 else 0
7 if x2<166 then node 14 elseif x2>=166 then node 15 else 1
8 class = 0
9 if x6<45.4 then node 16 elseif x6>=45.4 then node 17 else 0
10 class = 0
11 if x6<26.95 then node 18 elseif x6>=26.95 then node 19 else 0
12 class = 0
13 class = 1
14 if x8<30.5 then node 20 elseif x8>=30.5 then node 21 else 1
15 class = 1
16 class = 0
17 class = 1
18 class = 0
19 if x2<96.5 then node 22 elseif x2>=96.5 then node 23 else 1
20 if x3<61 then node 24 elseif x3>=61 then node 25 else 0
21 class = 1
22 class = 0
23 class = 1
24 class = 1
25 class = 0

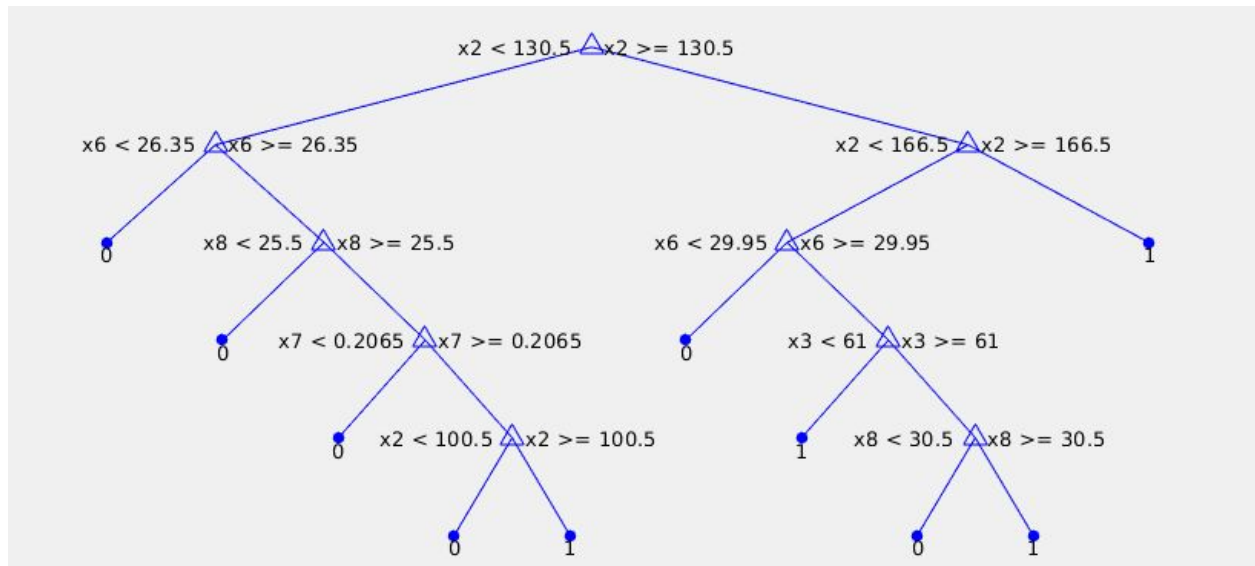
```

7.

```

new_tree=fitctree(x,y,'MaxNumSplits',17,
'MinParentSize',20,'MinLeafSize',4,'splitcriterion','deviance');

```



Decision tree for classification

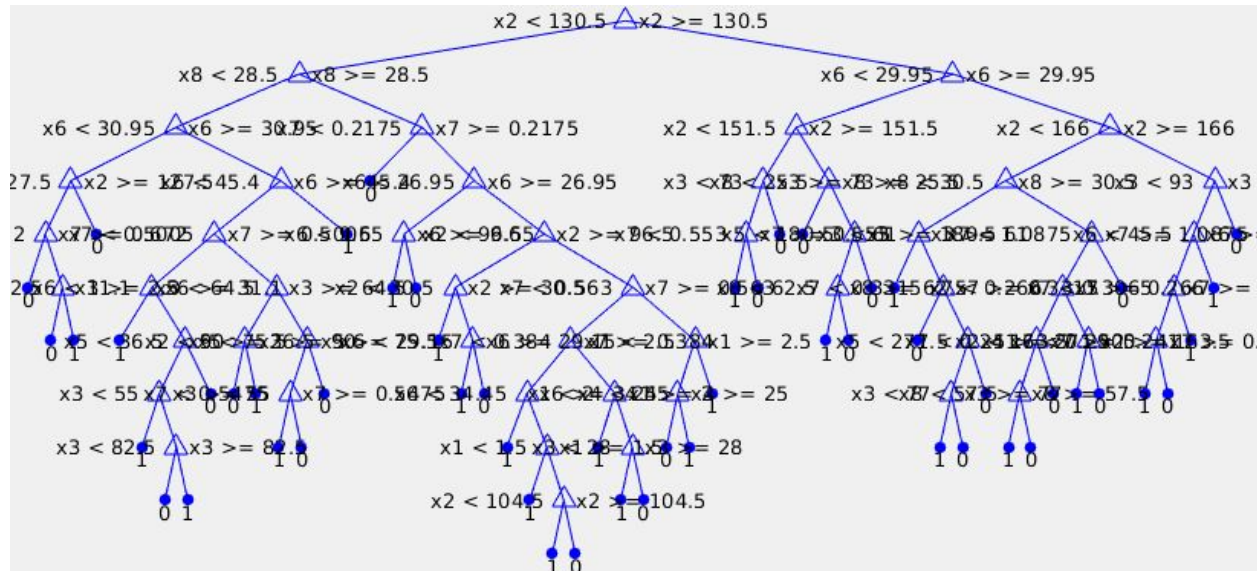
- 1 if $x_2 < 130.5$ then node 2 elseif $x_2 \geq 130.5$ then node 3 else 0
- 2 if $x_6 < 26.35$ then node 4 elseif $x_6 \geq 26.35$ then node 5 else 0
- 3 if $x_2 < 166.5$ then node 6 elseif $x_2 \geq 166.5$ then node 7 else 1
- 4 class = 0
- 5 if $x_8 < 25.5$ then node 8 elseif $x_8 \geq 25.5$ then node 9 else 0
- 6 if $x_6 < 29.95$ then node 10 elseif $x_6 \geq 29.95$ then node 11 else 1
- 7 class = 1
- 8 class = 0
- 9 if $x_7 < 0.2065$ then node 12 elseif $x_7 \geq 0.2065$ then node 13 else 0
- 10 class = 0
- 11 if $x_3 < 61$ then node 14 elseif $x_3 \geq 61$ then node 15 else 1
- 12 class = 0
- 13 if $x_2 < 100.5$ then node 16 elseif $x_2 \geq 100.5$ then node 17 else 0
- 14 class = 1
- 15 if $x_8 < 30.5$ then node 18 elseif $x_8 \geq 30.5$ then node 19 else 1
- 16 class = 0
- 17 class = 1
- 18 class = 0
- 19 class = 1

Test error: 0.2707

Test error increased when split criteria changed to deviance.

8. Split criterion 'Twoing' worked as good as gdi.

9. `new_tree=fitctree(x,y,'Prune','on','splitcriterion','gdi');`



This worked as bad as unrestricted tree

Test error = 0.2751

Decision tree for classification

- 1 if $x_2 < 130.5$ then node 2 elseif $x_2 \geq 130.5$ then node 3 else 0
- 2 if $x_8 < 28.5$ then node 4 elseif $x_8 \geq 28.5$ then node 5 else 0
- 3 if $x_6 < 29.95$ then node 6 elseif $x_6 \geq 29.95$ then node 7 else 1
- 4 if $x_6 < 30.95$ then node 8 elseif $x_6 \geq 30.95$ then node 9 else 0
- 5 if $x_7 < 0.2175$ then node 10 elseif $x_7 \geq 0.2175$ then node 11 else 0
- 6 if $x_2 < 151.5$ then node 12 elseif $x_2 \geq 151.5$ then node 13 else 0
- 7 if $x_2 < 166$ then node 14 elseif $x_2 \geq 166$ then node 15 else 1
- 8 if $x_2 < 127.5$ then node 16 elseif $x_2 \geq 127.5$ then node 17 else 0
- 9 if $x_6 < 45.4$ then node 18 elseif $x_6 \geq 45.4$ then node 19 else 0
- 10 class = 0
- 11 if $x_6 < 26.95$ then node 20 elseif $x_6 \geq 26.95$ then node 21 else 0
- 12 if $x_3 < 73$ then node 22 elseif $x_3 \geq 73$ then node 23 else 0
- 13 if $x_8 < 25.5$ then node 24 elseif $x_8 \geq 25.5$ then node 25 else 1
- 14 if $x_8 < 30.5$ then node 26 elseif $x_8 \geq 30.5$ then node 27 else 1
- 15 if $x_3 < 93$ then node 28 elseif $x_3 \geq 93$ then node 29 else 1
- 16 if $x_7 < 0.672$ then node 30 elseif $x_7 \geq 0.672$ then node 31 else 0
- 17 class = 0
- 18 if $x_7 < 0.5005$ then node 32 elseif $x_7 \geq 0.5005$ then node 33 else 0
- 19 class = 1

20 if $x_6 < 9.65$ then node 34 elseif $x_6 \geq 9.65$ then node 35 else 0
21 if $x_2 < 96.5$ then node 36 elseif $x_2 \geq 96.5$ then node 37 else 1
22 if $x_7 < 0.553$ then node 38 elseif $x_7 \geq 0.553$ then node 39 else 0
23 class = 0
24 class = 0
25 if $x_5 < 180.5$ then node 40 elseif $x_5 \geq 180.5$ then node 41 else 1
26 if $x_3 < 61$ then node 42 elseif $x_3 \geq 61$ then node 43 else 0
27 if $x_7 < 1.0875$ then node 44 elseif $x_7 \geq 1.0875$ then node 45 else 1
28 if $x_6 < 45.5$ then node 46 elseif $x_6 \geq 45.5$ then node 47 else 1
29 class = 0
30 class = 0
31 if $x_1 < 2.5$ then node 48 elseif $x_1 \geq 2.5$ then node 49 else 0
32 if $x_6 < 31.1$ then node 50 elseif $x_6 \geq 31.1$ then node 51 else 0
33 if $x_3 < 64.5$ then node 52 elseif $x_3 \geq 64.5$ then node 53 else 0
34 class = 1
35 class = 0
36 if $x_2 < 30.5$ then node 54 elseif $x_2 \geq 30.5$ then node 55 else 0
37 if $x_7 < 0.563$ then node 56 elseif $x_7 \geq 0.563$ then node 57 else 1
38 class = 1
39 class = 0
40 if $x_8 < 62.5$ then node 58 elseif $x_8 \geq 62.5$ then node 59 else 1
41 class = 0
42 class = 1
43 if $x_7 < 0.3315$ then node 60 elseif $x_7 \geq 0.3315$ then node 61 else 0
44 if $x_7 < 0.266$ then node 62 elseif $x_7 \geq 0.266$ then node 63 else 1
45 class = 0
46 if $x_7 < 0.3065$ then node 64 elseif $x_7 \geq 0.3065$ then node 65 else 1
47 class = 1
48 class = 0
49 class = 1
50 class = 1
51 if $x_5 < 36.5$ then node 66 elseif $x_5 \geq 36.5$ then node 67 else 0
52 if $x_2 < 90$ then node 68 elseif $x_2 \geq 90$ then node 69 else 1
53 if $x_5 < 75.5$ then node 70 elseif $x_5 \geq 75.5$ then node 71 else 0
54 class = 1
55 if $x_6 < 29.15$ then node 72 elseif $x_6 \geq 29.15$ then node 73 else 0
56 if $x_7 < 0.384$ then node 74 elseif $x_7 \geq 0.384$ then node 75 else 1
57 if $x_1 < 2.5$ then node 76 elseif $x_1 \geq 2.5$ then node 77 else 1
58 class = 1
59 class = 0
60 class = 0
61 if $x_5 < 271.5$ then node 78 elseif $x_5 \geq 271.5$ then node 79 else 1
62 if $x_7 < 0.241$ then node 80 elseif $x_7 \geq 0.241$ then node 81 else 1

63 if $x_2 < 163.5$ then node 82 elseif $x_2 \geq 163.5$ then node 83 else 1
64 if $x_7 < 0.2905$ then node 84 elseif $x_7 \geq 0.2905$ then node 85 else 1
65 class = 1
66 if $x_3 < 55$ then node 86 elseif $x_3 \geq 55$ then node 87 else 0
67 class = 0
68 class = 0
69 class = 1
70 if $x_7 < 0.5475$ then node 88 elseif $x_7 \geq 0.5475$ then node 89 else 0
71 class = 0
72 class = 1
73 class = 0
74 if $x_6 < 34.45$ then node 90 elseif $x_6 \geq 34.45$ then node 91 else 1
75 if $x_1 < 2$ then node 92 elseif $x_1 \geq 2$ then node 93 else 0
76 if $x_4 < 25$ then node 94 elseif $x_4 \geq 25$ then node 95 else 0
77 class = 1
78 if $x_3 < 77$ then node 96 elseif $x_3 \geq 77$ then node 97 else 1
79 class = 0
80 if $x_8 < 57.5$ then node 98 elseif $x_8 \geq 57.5$ then node 99 else 1
81 class = 0
82 class = 1
83 class = 0
84 class = 1
85 class = 0
86 class = 1
87 if $x_3 < 82.5$ then node 100 elseif $x_3 \geq 82.5$ then node 101 else 0
88 class = 1
89 class = 0
90 class = 1
91 if $x_1 < 1.5$ then node 102 elseif $x_1 \geq 1.5$ then node 103 else 0
92 class = 1
93 if $x_3 < 28$ then node 104 elseif $x_3 \geq 28$ then node 105 else 0
94 class = 0
95 class = 1
96 class = 1
97 class = 0
98 class = 1
99 class = 0
100 class = 0
101 class = 1
102 class = 1
103 if $x_2 < 104.5$ then node 106 elseif $x_2 \geq 104.5$ then node 107 else 0
104 class = 1
105 class = 0

```
106 class = 1
107 class = 0
```

```
tree =
```

```
ClassificationTree
  ResponseName: 'Y'
  CategoricalPredictors: []
  ClassNames: [0 1]
  ScoreTransform: 'none'
  NumObservations: 539
```

```
Properties, Methods
```

```
ans =
```

```
'Test error (new tree):'
```

```
error =
```

```
0.2751
```

Best performer:

```
new_tree=fitctree(x,y,'MaxNumSplits',17,  
'MinParentSize',20,'MinLeafSize',4,'splitcriterion','gdi');  
Least complexity and least error.
```