CS 760 Homework 4 by Cheng-Wei Lu

Problem 4.1

For each non-binary feature, I use its average as a standard to decide the binary transformation of each data. I use average because in reality, we often divide data in two parts with average. Therefore, I think it would also be reasonable if I use average here. If the original feature is greater than the average, it will be converted to 1. If not, it will be converted to 0. The average for passenger class is 2.3, age is 29.4, siblings/spouses aboard is 0.5, parents/children aboard is 0.4 and fare is 32.3.

Problem 4.2

Please see the appendix notebook cell of title, ”Calculate mutual information (Problem 4.2).”

Problem 4.3

Please see the appendix notebook cell of title, ”Build Decision Tree (Problem 4.3).” I stop the tree from growing more nodes when the entropy of y is less than 0.2 or when all the features are tested.

Problem 4.4

Please see the appendix notebook cell of title, ”Tree Displayed (Problem 4.4).” There you can see the structure of my tree.

Problem 4.5

Please see the appendix notebook cell of title, ”10 fold cross validation (Problem 4.5).” The accuracy of my prediction is 0.813.

Problem 4.6

My original data is

$$\begin{pmatrix} 2\text{(class)} \\ 0\text{(sex)} \\ 25\text{(age)} \\ 0\text{(siblings/spouses aboard)} \\ 0\text{(parents/children aboard)} \\ 50\text{(fare)} \end{pmatrix}$$

.

According to the conversion rule, my data input to the decision tree will be,

$$\begin{pmatrix} 0(\text{class}) \\ 0(\text{sex}) \\ 0(\text{age}) \\ 0(\text{siblings/spouses aboard}) \\ 0(\text{parents/children aboard}) \\ 1(\text{fare}) \end{pmatrix}$$

.

According to my decision tree, I wouldn't have survived.

Problem 4.7

(a)

Please see the appendix notebook cell of title, "Random Forest on Sampling - Trees Displayed (Problem 4.7(a))." There you can see the structure of my trees.

(b)

The accuracy is 0.831

(c)

According to my random forest, I wouldn't have survived.

Problem 4.8

(a)

Please see the appendix notebook cell of title, " Random Forest on Feature Dropping - Trees Displayed (Problem 4.8(a))." There you can see the structure of my trees.

(b)

The accuracy is 0.825

(c)

According to my random forest, I wouldn't have survived.

Problem 4.9

Yes, my predictions all agree with each other. I would like to use logistic regression more, because with logistic regression, we analyze more detailedly on the relationship between the coefficient of a feature with the prediction result.

hw4_decision_tree_Appendix_2

October 29, 2020

[8]:

```
import csv
import numpy as np
import copy
import time
import math
from scipy import stats
import random
```

[9]:

```
def split_train_label(data):
    train_x = []
    train_y = []
    for i in data:
        train_x.append(i[1:])
        train_y.append([i[0]])

    return train_x,train_y
```

[10]:

```
with open('titanic_data.csv','r') as file:
    temp = csv.reader(file)
    data = list(temp)

header = data[0]
data = data[1:]
for i in range(len(data)):
    row_len = len(data[0])
    for j in range(row_len):
        data[i][j] = float(data[i][j])

train_x, train_y = split_train_label(data)
```

1 Convert all features into binary features

I used average as the criteria to change the data

[12]:

```
#binary conversion
binary_avg = []
for i in range(len(header[1:])):
    total = 0
    avg = 0
    for j in train_x:
        total += j[i]
    avg = total/len(train_x)
    binary_avg.append(avg)

for i in range(len(train_x)):
    for j in range(len(train_x[0])):
        if train_x[i][j] >= binary_avg[j]:
            train_x[i][j] = 1.0
        else:
            train_x[i][j] = 0.0

print(binary_avg)
```

```
[2.305524239007892, 0.35400225479143177, 29.471443066516347, 0.5253664036076663,
0.3833145434047351, 32.30542018038328]
```

2 Calculate mutual information (Problem 4.2)

[13]:

```
def count_ones_Hx(featureNumber,data): # choice is 0 or 1, and data is the␣
 ↪training data that will be gone over
    x_prob = {}
    count = 0
    for i in range(len(data)):
        if 1 == data[i][featureNumber]:
            count += 1
    x_prob[1] = count/len(data)
    x_prob[0] = 1-x_prob[1]
    return x_prob

def count_ones_Hxy(featureNumber,dataX,dataY): # choice is 0 or 1, and data is␣
 ↪the training data that will be gone over
    x_y_prob = {(0,0):0,(0,1):0,(1,0):0,(1,1):0}
    x_cond_y = {(0,0):0,(0,1):0,(1,0):0,(1,1):0}
    count_y = [0,0] ## condition when y = 0 or 1

    for i in range(len(dataX)):
        for j in range(2):
            for k in range(2):
                if dataX[i][featureNumber] == j and dataY[i][0] == k:
                    x_y_prob[(j,k)] += 1

    if (x_y_prob[(0,0)]+x_y_prob[(1,0)]) != 0:
        x_cond_y[(0,0)] = x_y_prob[(0,0)]/(x_y_prob[(0,0)]+x_y_prob[(1,0)])
```

```python
        x_cond_y[(1,0)] = x_y_prob[(1,0)]/(x_y_prob[(0,0)]+x_y_prob[(1,0)])
    if (x_y_prob[(0,1)]+x_y_prob[(1,1)]) != 0:
        x_cond_y[(0,1)] = x_y_prob[(0,1)]/(x_y_prob[(0,1)]+x_y_prob[(1,1)])
        x_cond_y[(1,1)] = x_y_prob[(1,1)]/(x_y_prob[(0,1)]+x_y_prob[(1,1)])
    for i in x_y_prob.keys():
        x_y_prob[i] = x_y_prob[i]/len(dataX)

    return x_y_prob,x_cond_y

def MI_eval(train_x,train_y,feature_checked):
    MI_array = []
    for i in range(len(train_x[0])): # i is the feature to calculate mutual information
        if feature_checked[i] == 1:
            MI_array.append(0)
            continue
        Hx = 0
        Hxy = 0
        x_prob = count_ones_Hx(i,train_x)
        x_y_prob,x_cond_y = count_ones_Hxy(i,train_x,train_y)
        for j in range(2):
            if x_prob[j] != 0:
                Hx += x_prob[j]*math.log((1/x_prob[j]),2)
            for k in range(2):
                if x_cond_y[j,k] == 0:
                    Hxy += 0
                else:
                    if x_cond_y[j,k] != 0:
                        Hxy += x_y_prob[j,k]*math.log((1/x_cond_y[j,k]),2)
        MI_array.append(Hx-Hxy)

    MI_y = 0
    prob_y = 0
    for i in train_y:
        MI_y += i[0]
    prob_y = MI_y/len(train_y)

    if prob_y == 0 or prob_y == 1:
        information_y = 0
    else:
        information_y = prob_y*math.log(1/prob_y,2) + (1-prob_y)*math.log(1/(1-prob_y),2)
    return MI_array, information_y
```

3 Build Decision Tree (Problem 4.3)

[26]:
```
class decision_tree_node:
    def __init__(self, train_x, train_y, feature_checked = [0,0,0,0,0,0], order␣
 ↪= [],depth = 0,print_tree = False):
        self.train_x = train_x
        self.train_y = train_y
        self.feature_checked = feature_checked
        self.isleaf = 0
        self.leftNode = None
        self.rightNode = None
        self.feature_choice = None
        self.order = copy.deepcopy(order)
        self.depth = depth
        self.print_tree = print_tree
        self.build_tree()

    def build_tree(self):
        MI, Hy = MI_eval(self.train_x, self.train_y, self.feature_checked)
        left_x = []
        left_y = []
        right_x = []
        right_y = []

        if Hy <= 0.2 or sum(self.feature_checked) == len(self.train_x[0]):
            self.isleaf = 1
            #print(self.order)

        else:
            max_idx = 0
            for i in range(len(MI)):
                if MI[i] > MI[max_idx]:
                    max_idx = i
            self.feature_choice = max_idx

            for i in range(len(self.train_x)):
                if self.train_x[i][max_idx] == 0:
                    left_x.append(self.train_x[i])
                    left_y.append(self.train_y[i])
                else:
                    right_x.append(self.train_x[i])
                    right_y.append(self.train_y[i])

        if len(left_x) and len(right_x) > 0:
            new_feature_checked = copy.deepcopy(self.feature_checked)
            new_feature_checked[self.feature_choice] = 1
            self.order.append(str(self.feature_choice) + " L ")
            if self.print_tree == True:
                print('|---'*self.depth,'feature ',self.feature_choice+1, ": =␣
↪0")
            self.leftNode =␣
↪decision_tree_node(left_x,left_y,new_feature_checked,self.order,depth = self.
↪depth+1,print_tree=self.print_tree)
            self.order[-1] = str(self.feature_choice) + " R "
            if self.print_tree == True:
                print('|---'*self.depth,'feature ',self.feature_choice+1, ": =␣
↪1")
            self.rightNode =␣
↪decision_tree_node(right_x,right_y,new_feature_checked,self.order,depth =␣
↪self.depth+1,print_tree=self.print_tree)
        else:
            self.isleaf = 1
    def predict(self,train_x):
            if self.isleaf == 1:
                count_1 = 0
                count_0 = 0
                for i in range(len(self.train_y)):
                    if self.train_y[i][0] == 1:
                        count_1 += 1
                    else:
                        count_0 += 1
                prob_1 = count_1/(count_1+count_0)
                prob_0 = count_0/(count_1+count_0)
                if prob_1 > prob_0:
                    return 1
                else:
                    return 0

            elif train_x[self.feature_choice] == 0:
                #print('feature:' ,self.feature_choice, ' go left')
                return(self.leftNode.predict(train_x))
            elif train_x[self.feature_choice] == 1:
                #print('feature:' ,self.feature_choice, ' go right')
                return(self.rightNode.predict(train_x))
```

4 Tree Displayed (Problem 4.4)

[27]:
```
a = decision_tree_node(train_x, train_y,print_tree = True)
```

```
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  6 : = 0
```

```
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  6 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  6 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  6 : = 0
|---|---|---|--- feature  6 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|--- feature  1 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  6 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  6 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  6 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  6 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  6 : = 0
```

```
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|--- feature  6 : = 1
|--- feature  1 : = 1
|---|--- feature  6 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  5 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|--- feature  6 : = 1
```

5 10 fold cross validation (Problem 4.5)

[48]:

```python
interval = math.ceil(len(train_x)/10)
global_accuracy = []
for i in range(10):
    local_accuracy = 0
    count = 0
    tr_x = []
    tr_y = []
    te_x = []
    te_y = []
    te_x += train_x[i*interval : (i+1)*interval]
    te_y += train_y[i*interval : (i+1)*interval]
    tr_x += train_x[0:i*interval]
    tr_x += train_x[(i+1)*interval:]
    tr_y += train_y[0:i*interval]
    tr_y += train_y[(i+1)*interval:]
    tree = decision_tree_node(tr_x, tr_y)

    for j in range(len(te_x)):
         if tree.predict(te_x[j]) == te_y[j][0]:
                count += 1
    global_accuracy.append(count/len(te_x))
print('average accuracy = ', sum(global_accuracy)/len(global_accuracy))
```

```
average accuracy =  0.8130128037627383
```

6 Will I survived?

[45]:

```
my_data = [0,0,0,0,0,1]
tree.predict(my_data )
```

[45]:

```
0
```

7 Random Forest on Sampling - Trees Displayed (Problem 4.7(a))

[39]:

```
interval = math.ceil(len(train_x)/10)
forest = []
global_accuracy = []
for i in range(5):
    tr_x = []
    tr_y = []
    te_x = []
    te_y = []
    for j in range(len(train_x)):
        if random.random() <= 0.8:
            tr_x.append(train_x[j])
            tr_y.append(train_y[j])
        else:
            te_x.append(train_x[j])
            te_y.append(train_y[j])
    print('---------------------------tree {}----------------------------'.
 ↪format(i+1))
    tree = decision_tree_node(tr_x, tr_y,print_tree = True)

    forest.append(tree)
```

```
---------------------------tree 1----------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  6 : = 0
```

```
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  6 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  6 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  6 : = 0
|---|---|---|--- feature  6 : = 1
|--- feature  1 : = 1
|---|--- feature  3 : = 0
|---|---|--- feature  6 : = 0
|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  5 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  6 : = 1
|---|--- feature  3 : = 1
|---|---|--- feature  6 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  6 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  6 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
```

```
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|--- feature  6 : = 1
|--- feature  1 : = 1
|---|--- feature  6 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|--- feature  6 : = 1
---------------------------tree 2----------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  6 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  6 : = 0
|---|---|---|--- feature  6 : = 1
```

```
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  6 : = 0
|---|---|---|--- feature  6 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|--- feature  1 : = 1
|---|--- feature  3 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  6 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  6 : = 1
|---|--- feature  3 : = 1
|---|---|--- feature  6 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  6 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  6 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  5 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|--- feature  6 : = 1
|--- feature  1 : = 1
|---|--- feature  6 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
```

```
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|--- feature  6 : = 1
--------------------------tree 3---------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  4 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  6 : = 0
|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  5 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  6 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|--- feature  1 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  6 : = 0
|---|---|---|--- feature  3 : = 0
```

```
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  6 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  6 : = 1
|---|---|--- feature  3 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|--- feature  1 : = 1
|---|--- feature  6 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|--- feature  6 : = 1
---------------------------tree 4---------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  6 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  6 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
```

```
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  6 : = 0
|---|---|---|--- feature  6 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|--- feature  1 : = 1
|---|--- feature  6 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|--- feature  6 : = 1
|---|---|--- feature  3 : = 0
|---|---|--- feature  3 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  6 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 1
|---|--- feature  6 : = 1
|--- feature  1 : = 1
|---|--- feature  6 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
```

```
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|--- feature  6 : = 1
---------------------------tree 5----------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  4 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  6 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  6 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  6 : = 0
|---|---|---|---|--- feature  6 : = 1
|--- feature  1 : = 1
|---|--- feature  6 : = 0
```

```
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  5 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  5 : = 0
|---|---|---|---|--- feature  5 : = 1
|---|---|---|--- feature  4 : = 1
|---|--- feature  6 : = 1
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  6 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|--- feature  6 : = 1
|--- feature  1 : = 1
|---|--- feature  6 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 1
|---|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|---|--- feature  4 : = 0
|---|---|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 1
|---|---|---|---|--- feature  4 : = 0
```

```
|---|---|---|---|--- feature  4 : = 1
|---|--- feature  6 : = 1
```

[46]:

```
for i in forest:
    print(i.predict(my_data))
```

```
0
0
0
0
0
```

8 random forest + cross validation

[51]:

```
interval = math.ceil(len(train_x)/10)
global_accuracy = []
for cros_val in range(10):
    forest = []
    train_x2 = []
    train_y2 = []
    test_x2 = []
    test_y2 = []
    test_x2 += train_x[cros_val*interval : (cros_val+1)*interval]
    test_y2 += train_y[cros_val*interval : (cros_val+1)*interval]
    train_x2 += train_x[0:cros_val*interval]
    train_x2 += train_x[(cros_val+1)*interval:]
    train_y2 += train_y[0:cros_val*interval]
    train_y2 += train_y[(cros_val+1)*interval:]

    for i in range(5):
        tr_x = []
        tr_y = []
        te_x = []
        te_y = []
        for j in range(len(train_x2)):
            if random.random() <= 0.8:
                tr_x.append(train_x2[j])
                tr_y.append(train_y2[j])
            else:
                te_x.append(train_x2[j])
                te_y.append(train_y2[j])

        tree = decision_tree_node(tr_x, tr_y)
        forest.append(tree)
    temp_accuracy = []
    for k in range(len(te_x)):
        temp = []
        answer = None
        for j in forest:
            temp.append(j.predict(te_x[k]))
        if sum(temp) > len(temp)/2 :
            answer = 1
        else :
            answer = 0

        if answer == te_y[k][0]:
            temp_accuracy.append(1)
        else:
            temp_accuracy.append(0)
    global_accuracy.append(sum(temp_accuracy)/len(temp_accuracy))

print('average accuracy = ',sum(global_accuracy)/len(global_accuracy))
```

```
average accuracy =  0.8316802629711637
```

9 Random Forest on Feature Dropping - Trees Displayed (Problem 4.8(a))

[53]:
```
forest = []

for k in range(len(train_x[0])):

    #drop features
    tr_x_drop = []
    for row in train_x:
        tr_x_drop.append(row[0:k] + row[k+1:])
    print('---------------------------tree {}----------------------------'.
 ↪format(k+1))
    forest.append(decision_tree_node(tr_x_drop, train_y,print_tree = True))
```

```
---------------------------tree 1----------------------------
 feature  1 : = 0
|--- feature  5 : = 0
|---|--- feature  4 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  2 : = 0
|---|---|---|--- feature  3 : = 0
```

```
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  2 : = 1
|--- feature  5 : = 1
|---|--- feature  4 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
 feature  1 : = 1
|--- feature  5 : = 0
|---|--- feature  3 : = 0
|---|---|--- feature  2 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  2 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|--- feature  5 : = 1
|---|--- feature  4 : = 0
|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
---------------------------tree 2----------------------------
 feature  1 : = 0
|--- feature  4 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  2 : = 0
```

```
|---|---|---|--- feature  2 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  2 : = 0
|---|---|---|--- feature  2 : = 1
|--- feature  4 : = 1
|---|--- feature  2 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  2 : = 1
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
 feature  1 : = 1
|--- feature  2 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|--- feature  3 : = 1
|--- feature  2 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  4 : = 0
```

```
|---|---|--- feature  4 : = 1
---------------------------tree 3----------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  4 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|--- feature  1 : = 1
|---|--- feature  4 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|--- feature  1 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|--- feature  5 : = 1
```

```
---------------------------tree 4---------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  4 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|--- feature  1 : = 1
|---|--- feature  4 : = 0
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|--- feature  1 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
```

```
|---|--- feature  5 : = 1
---------------------------tree 5----------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  4 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|--- feature  1 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|--- feature  1 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
```

```
|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
---------------------------tree 6---------------------------
 feature  2 : = 0
|--- feature  1 : = 0
|---|--- feature  5 : = 0
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|--- feature  1 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
 feature  2 : = 1
|--- feature  1 : = 0
|---|--- feature  3 : = 0
|---|---|--- feature  4 : = 0
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  5 : = 0
|---|---|---|--- feature  5 : = 1
|---|--- feature  3 : = 1
|---|---|--- feature  5 : = 0
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  5 : = 1
|--- feature  1 : = 1
|---|--- feature  5 : = 0
|---|---|--- feature  3 : = 0
|---|---|---|--- feature  4 : = 0
```

```
|---|---|---|--- feature  4 : = 1
|---|---|--- feature  3 : = 1
|---|---|---|--- feature  4 : = 0
|---|---|---|--- feature  4 : = 1
|---|--- feature  5 : = 1
|---|---|--- feature  4 : = 0
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
|---|---|--- feature  4 : = 1
|---|---|---|--- feature  3 : = 0
|---|---|---|--- feature  3 : = 1
```

10 Will I survived?

[54]:

```python
for i in forest:
    print(i.predict(my_data))
```

```
0
0
0
0
0
0
```

11 Random forest + feature dropping(10 fold cross validation)

[57]:

```python
interval = math.ceil(len(train_x)/10)
global_accuracy = []
for i in range(len(train_x[0])):

    #drop features
    tr_x_drop = []
    for row in train_x:
        tr_x_drop.append(row[0:i] + row[i+1:])
    forest.append(decision_tree_node(tr_x_drop, train_y))

    for cros_val in range(10):
        forest = []
        train_x2 = []
        train_y2 = []
        test_x2 = []
        test_y2 = []
        test_x2 += tr_x_drop[cros_val*interval : (cros_val+1)*interval]
        test_y2 += train_y[cros_val*interval : (cros_val+1)*interval]
        train_x2 += tr_x_drop[0:cros_val*interval]
        train_x2 += tr_x_drop[(cros_val+1)*interval:]
        train_y2 += train_y[0:cros_val*interval]
        train_y2 += train_y[(cros_val+1)*interval:]

        tree = decision_tree_node(train_x2, train_y2)
        forest.append(tree)
    temp_accuracy = []
    for k in range(len(test_x2)):
        temp = []
        answer = None
        for j in forest:
            temp.append(j.predict(test_x2[k]))
        if sum(temp) > len(temp)/2 :
            answer = 1
        else :
            answer = 0

        if answer == test_y2[k][0]:
            temp_accuracy.append(1)
        else:
            temp_accuracy.append(0)
    global_accuracy.append(sum(temp_accuracy)/len(temp_accuracy))

print('average accuracy = ',sum(global_accuracy)/len(global_accuracy))
```

```
average accuracy =  0.8255813953488372
```