273A hw1

EnYu Huang

January 14, 2021

1 Problem 1: Python & Data Exploration

1.1

[50]:

```
import numpy as np
import matplotlib.pyplot as plt
iris = np.genfromtxt("data/iris.txt",delimiter = None)
Y = iris[:,-1]
X = iris[:,0:-1]
x = X.shape
print(x)
```

```
(148, 4)
```

It means data set X has 4 columns and each of the column include 148 features(data) in it.

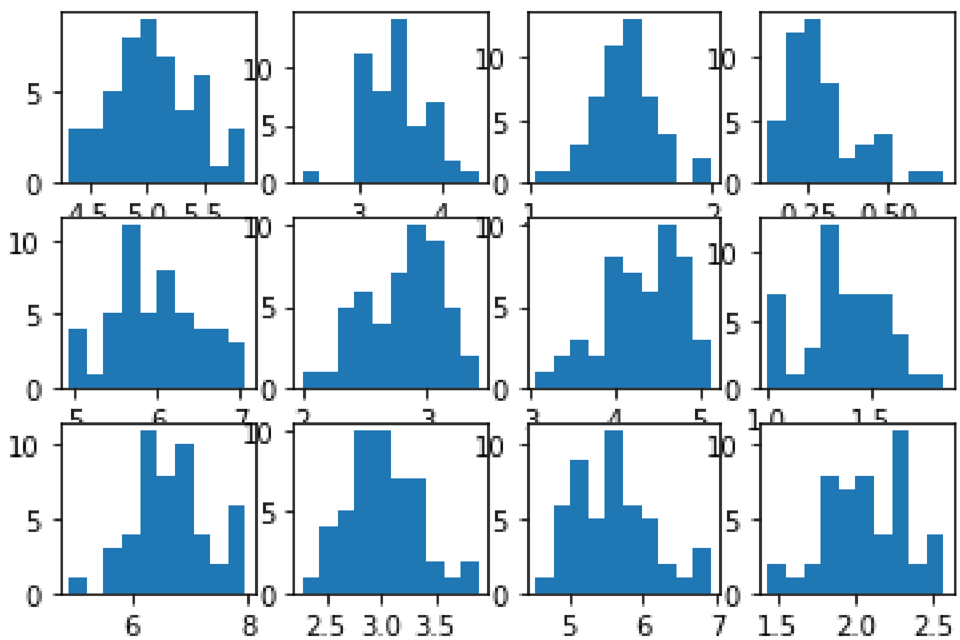
1.2

[51]:

```
data1,data2,data3,data4 = [],[],[],[] #Y = 0
data5,data6,data7,data8 = [],[],[],[] #Y = 1
data9,data10,data11,data12 = [],[],[],[] #Y = 2
for i in range(len(iris)):
    if iris[i,-1] == 0:
        data1.append(iris[i,0])
        data2.append(iris[i,1])
        data3.append(iris[i,2])
        data4.append(iris[i,3])
    if iris[i,-1] == 1:
        data5.append(iris[i,0])
        data6.append(iris[i,1])
        data7.append(iris[i,2])
        data8.append(iris[i,3])
    if iris[i,-1] == 2:
        data9.append(iris[i,0])
        data10.append(iris[i,1])
        data11.append(iris[i,2])
        data12.append(iris[i,3])
plt.subplot(3,4,1)
```

```
plt.hist(data1)
plt.subplot(3,4,2)
plt.hist(data2)
plt.subplot(3,4,3)
plt.hist(data3)
plt.subplot(3,4,4)
plt.hist(data4)
plt.subplot(3,4,5)
plt.hist(data5)
plt.subplot(3,4,6)
plt.hist(data6)
plt.subplot(3,4,7)
plt.hist(data7)
plt.subplot(3,4,8)
plt.hist(data8)
plt.subplot(3,4,9)
plt.hist(data9)
plt.subplot(3,4,10)
plt.hist(data10)
plt.subplot(3,4,11)
plt.hist(data11)
plt.subplot(3,4,12)
plt.hist(data12)
```

[51]:



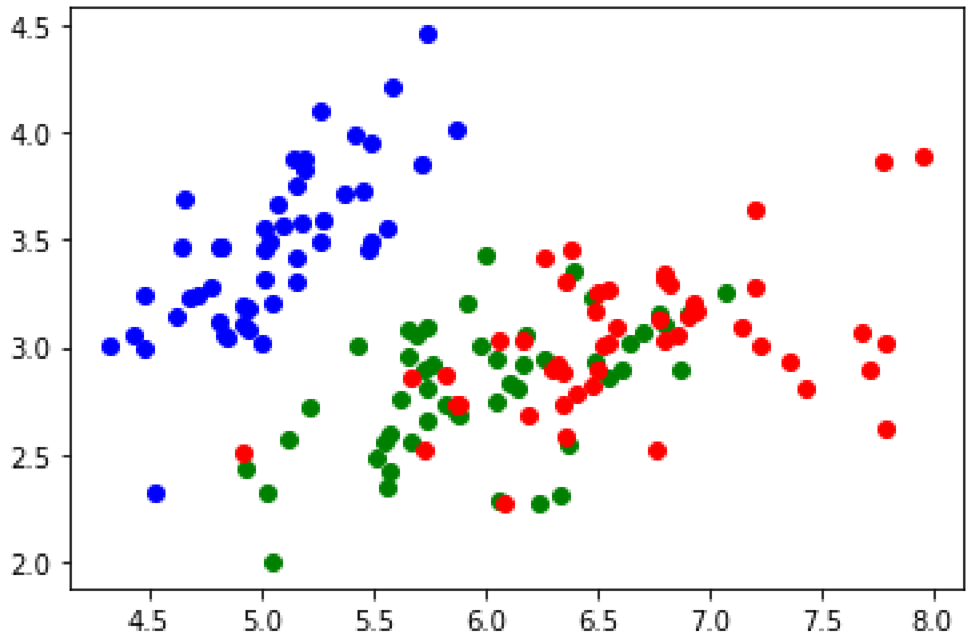
1.3

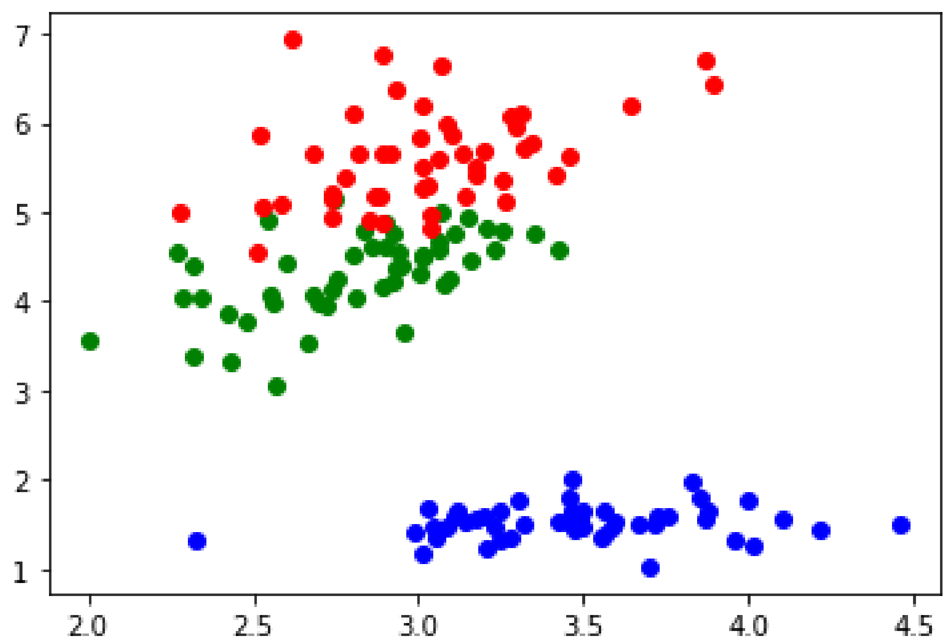
[57]:
```
print("feature in first column : ","mean = ",np.mean(iris[:,0]),"median = ",np.
 ↪median(iris[:,0]),"var = ",np.var(iris[:,0]),"std = ",np.std(iris[:,0]))
print("feature in second column : ","mean = ",np.mean(iris[:,1]),"median = ",np.
 ↪median(iris[:,1]),"var = ",np.var(iris[:,1]),"std = ",np.std(iris[:,1]))
print("feature in third column : ","mean = ",np.mean(iris[:,2]),"median = ",np.
 ↪median(iris[:,2]),"var = ",np.var(iris[:,2]),"std = ",np.std(iris[:,2]))
print("feature in fourth column : ","mean = ",np.mean(iris[:,3]),"median = ",np.
 ↪median(iris[:,3]),"var = ",np.var(iris[:,3]),"std = ",np.std(iris[:,3]))
```

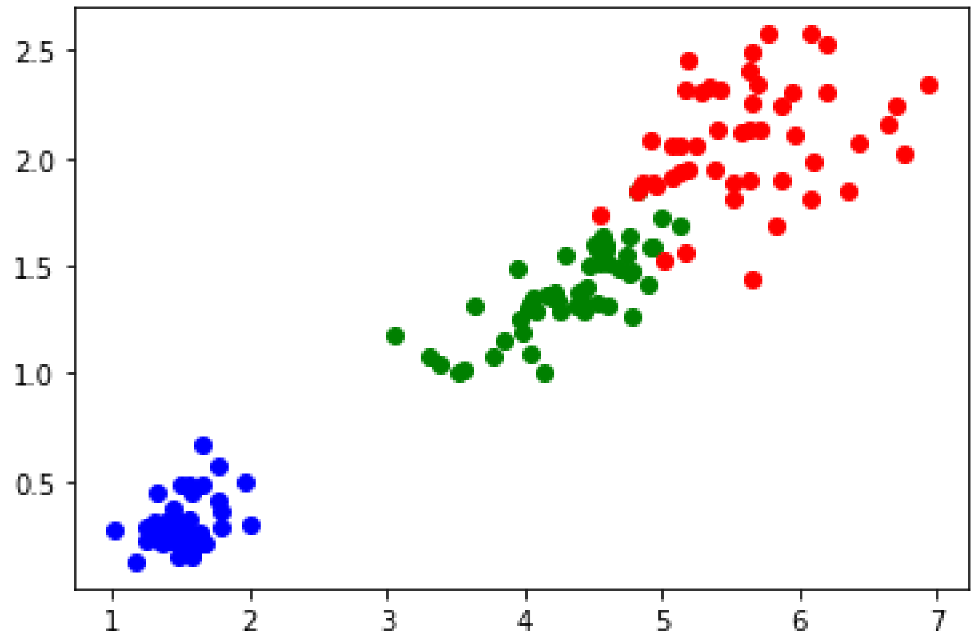
```
Feature in first column :  mean =  5.900103764189188 median =  5.84664255 var =
0.6945590049046649 std =  0.833402066774894
Feature in second column :  mean =  3.098930916891892 median =
3.0598060499999997 var =  0.19035056790635788 std =  0.43629183800107685
Feature in third column :  mean =  3.8195548405405404 median =  4.3998377 var =
3.076716342840002 std =  1.7540571093439352
Feature in fourth column :  mean =  1.2525554845945945 median =  1.361768 var =
0.5757356415417657 std =  0.7587724570263247
```

1.4

[53]:
```
plt.scatter(data1,data2,color ='blue')
plt.scatter(data5,data6,color ='green')
plt.scatter(data9,data10,color ='red')
plt.show() #(feature(1,2) when Y=0,1,2)
plt.scatter(data2,data3,color ='blue')
plt.scatter(data6,data7,color ='green')
plt.scatter(data10,data11,color ='red')
plt.show() #(feature(2,3) when Y=0,1,2)
plt.scatter(data3,data4,color ='blue')
plt.scatter(data7,data8,color ='green')
plt.scatter(data11,data12,color ='red')
plt.show() #(feature(3,4) when Y=0,1,2)
```


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


7
6
5
4
3
2
1
2.0
2.5
3.0
3.5
4.0
4.5



2 Problem 2: Basic Linear Algebra

2.1

1.The matrix must be square 2.The determinant of the matrix can not be 0

2.2

```
    A =[1  2  3]
       [2 -1  1]
       [1  3  2]
    B =[0  -8 -2]
       [1  -4 -2]
       [-4  4  1]
det(A)=(-2 + 2 + 18)-(-3 + 3 + 8)=18 - 8 = 10
det(B)=(0 - 64 -8)-(-32 - 8)=-32
```

2.3

$$A^{-1} = \frac{1}{|A|}\begin{bmatrix} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{13} & a_{12} \\ a_{33} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{23} & a_{21} \\ a_{33} & a_{31} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{13} & a_{11} \\ a_{23} & a_{21} \end{vmatrix} \\ \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{11} \\ a_{32} & a_{31} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}.$$

Base on

```
The inverse of A =
           [-5  5  5]   [-0.5  0.5  0.5]
 1/10  *   [-3 -1  5] = [-0.3 -0.1  0.5]
           [ 7 -1 -5]   [ 0.7 -0.1 -0.5]

The inverse of B =
          [4    0   8]     [-0.125   -0.       -0.25   ]
 1/(-32)* [7   -8  -2]  =  [-0.21875  0.25      0.0625 ]
          [-12 32   8]     [ 0.375   -1.       -0.25   ]
```

2.4

$(A^{-1})^T = (A^T)^{-1}$

A^{-1} =

```
 [-0.5  0.5  0.5]
 [-0.3 -0.1  0.5]
 [ 0.7 -0.1 -0.5]
```

$(A^{-1})^T = (A^T)^{-1}$

```
 [-0.5   -0.3   0.7  ]
 [ 0.5   -0.1   -0.1 ]
 [ 0.5   -0.5   -0.5 ]
```

B^{-1} =

```
 [-0.125   0     -0.25  ]
 [-0.21875 0.25  0.0625]
 [ 0.375  -0.1  -0.25  ]
```

$(B^{-1})^T = (B^T)^{-1}$

```
 [-0.125   -0.21875   0.375 ]
 [ 0         0.25     -1     ]
 [-0.25      0.0625   -0.25  ]
```

2.5

$C = AB$, $C^{-1} = (AB)^{-1} = B^{-1}A^{-1}$

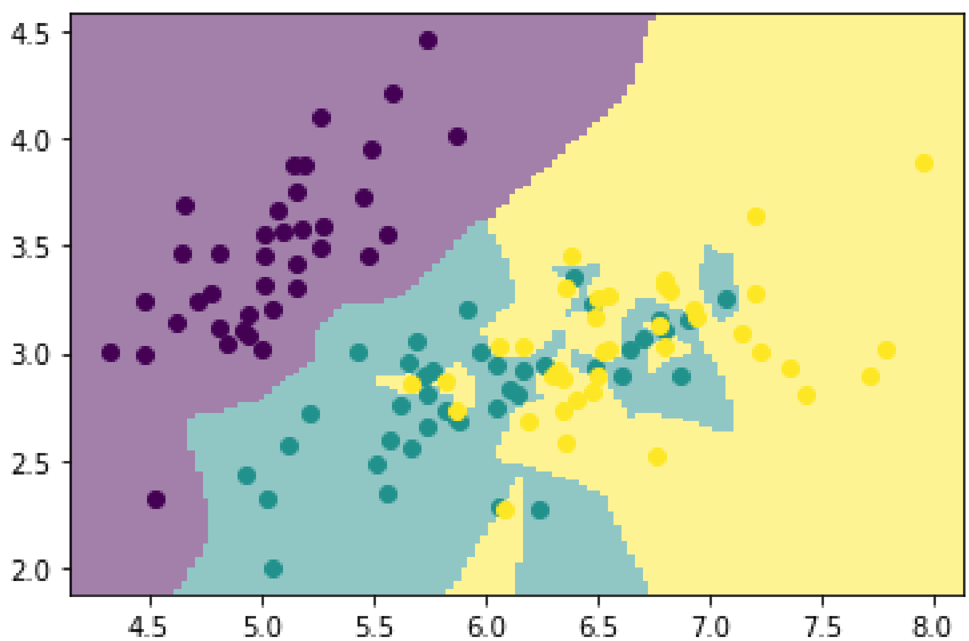
```
B^-1 =
       [-0.125   0    -0.25  ]
       [-0.21875 0.25  0.0625]
       [ 0.375  -0.1  -0.5   ]
A^-1 =
       [-0.5  0.5  0.5]
       [-0.3 -0.1  0.5]
       [ 0.7 -0.1 -0.5]
(B^-1)(A^-1) =
       [-0.125  -0.21875    0.375  ]
       [0.78125 -0.1440625 -0.15625]
       [-0.0625  0.3125    -0.1875 ]
```

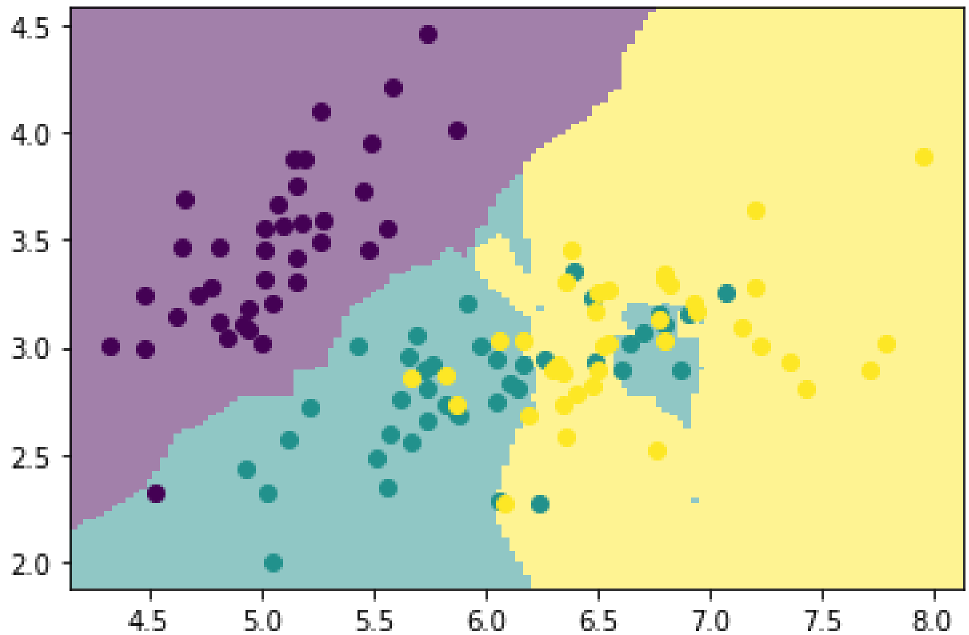
3 Problem 3: kNN predictions

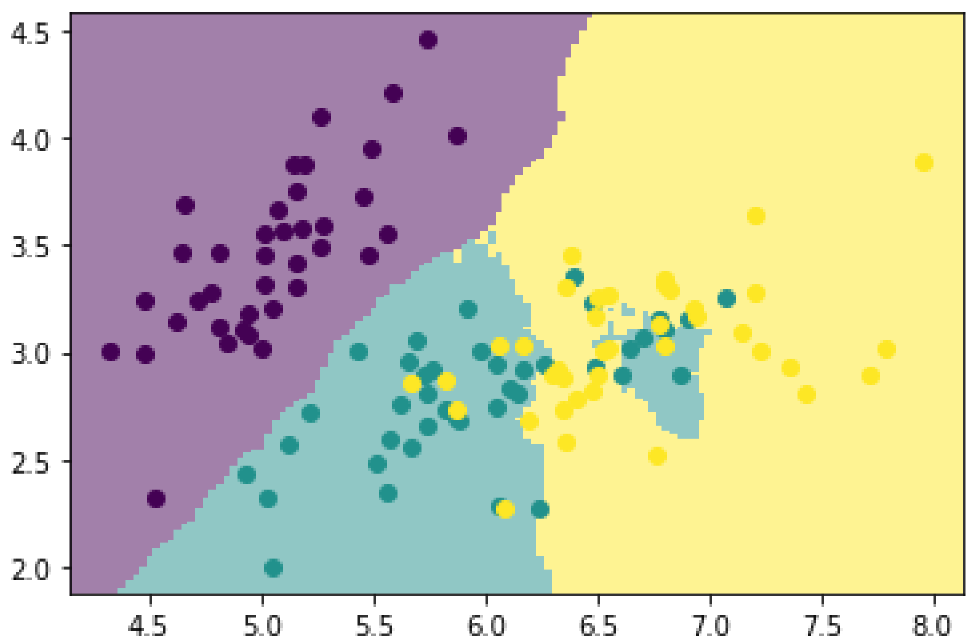
3.1

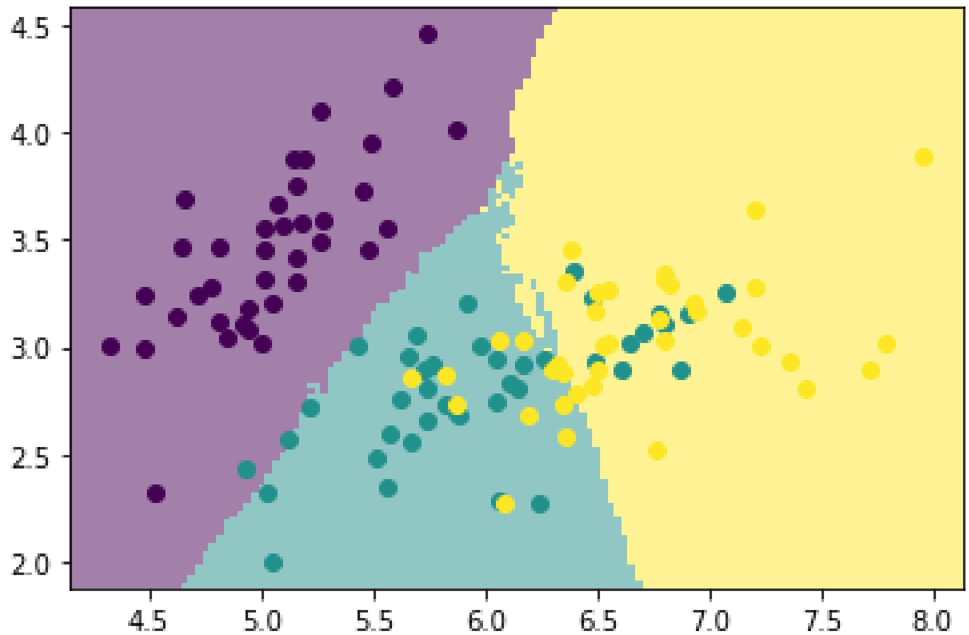
[54]:
```
import mltools as ml
X,Y = ml.shuffleData(X,Y)
Xtr,Xva,Ytr,Yva = ml.splitData(X,Y, 0.8)

k = [1,5,10,50]
for i in k:
    knn = ml.knn.knnClassify()
    knn.train(Xtr[:,0:2], Ytr, i)  #Only first two columns of iris
    ml.plotClassify2D( knn, Xtr[:,0:2], Ytr )
```




4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0



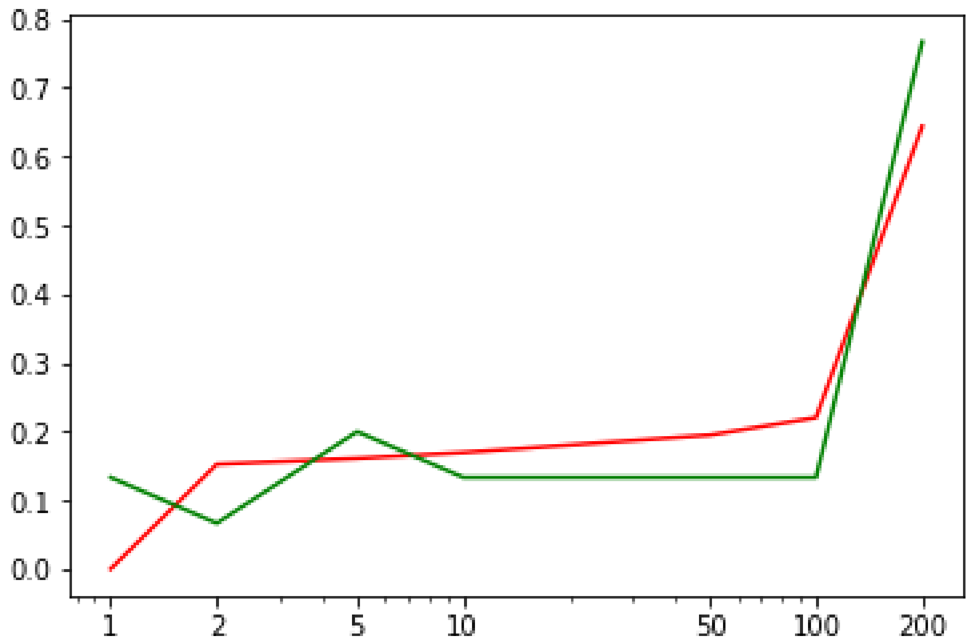
By increasing the value of K, the pros of higher value K can train the model to avoid underfitting problem, however, the cons of increasing K is, if K is too large comparing to the total data, it will cause overfitting problem, raising the train and test error rate in the end.

3.2

[55]:
```
K=[1,2,5,10,50,100,200]
errorTrain = np.zeros(7)
errorTest = np.zeros(7)
for i,k in enumerate(K):
    learner = ml.knn.knnClassify()
    learner.train(Xtr[:,0:2], Ytr, k)
    Yhat = learner.predict(Xtr[:,0:2]) #predict first two columns of Xtr after␣
 ↪training
    Ytest = learner.predict(Xva[:,0:2])

    errorTrain[i] = (np.sum(Yhat != Ytr)) / Xtr.shape[0]
    errorTest[i] = (np.sum(Ytest != Yva)) / Xva.shape[0]

plt.semilogx(K, errorTrain, color='red')
plt.semilogx(K, errorTest, color='green')
plt.show()
```



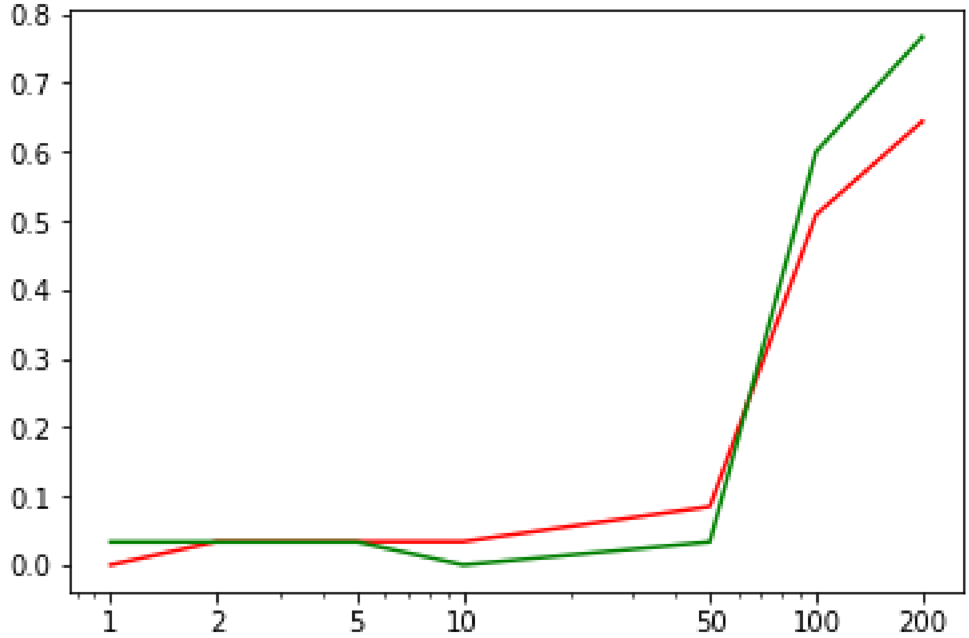
Base on the plot, I will recommend K = 2, due to the reason that since K = 2, the training error rate starts to rise, meaning that the model has been trained by data, and the testing error achieves to the lowest value. This lead to more ideal training and generalization error which is not too low or high.

3.3

[56]:
```
for i,k in enumerate(K):
    learner = ml.knn.knnClassify()
    learner.train(Xtr, Ytr, k)
    Yhat = learner.predict(Xtr)
    Ytest = learner.predict(Xva)

    errorTrain[i] = (np.sum(Yhat != Ytr)) / Xtr.shape[0]
    errorTest[i] = (np.sum(Ytest != Yva)) / Xva.shape[0]

plt.semilogx(K, errorTrain, color='red')
plt.semilogx(K, errorTest, color='green')
plt.show()
```



After including all the features rather than just first two of the columns, we can observe that the error rate will increase steeply after K = 50. The plot in problem 3 shows lower testing error rate at the beginning than in problem 2, and performs smoother line while K increasing from 2 to 50 (approximate value). In this plot, I will choose K from 10 to 50 since its testing error decreases, meanwhile the generalization error remain ideal difference, hence we can avoid overfitting and underfitting problem by choosing K within this range.

4 Problem 4: Naïve Bayes Classifiers

4.1

From the table, we can know that
P(Y=1)=0.4, P(Y=-1)=0.6
P(X1=1 | Y=1)=0.75, P(X1=0 | Y=1)=0.25,P(X1=1 | Y=-1)=0.5, P(X1=0 | Y=-1)=0.5
P(X2=1 | Y=1)=0, P(X2=0 | Y=1)=1, P(X2=1 | Y=-1)=0.83, P(X2=0 | Y=-1)=0.17
P(X3=1 | Y=1)=0.75, P(X3=0 | Y=1)=0.25,P(X3=1 | Y=-1)=0.66, P(X3=0 | Y=-1)=0.34
P(X4=1 | Y=1)=0.5, P(X4=0 | Y=1)=0.5, P(X4=1 | Y=-1)=0.83, P(X4=0 | Y=-1)=0.17
P(X5=1 | Y=1)=0.25, P(X5=0 | Y=1)=0.75,P(X5=1 | Y=-1)=0.34, P(X5=0 | Y=-1)=0.66

4.2

X = (0 0 0 0 0) =>
P(X | Y=1) P(Y=1) = 0.25 * 1 * 0.25 * 0.5 * 0.75 * 0.4 = 0.0936
P(X | Y=-1)P(Y=-1)= 0.5 * 0.17 * 0.34 * 0.17 * 0.66 * 0.6 = 0.018
P(X | Y=1) P(Y=1) > P(X | Y=-1)P(Y=-1) => P(X) prediction = 1 (read)
X = (1 1 0 1 0) =>

P(X | Y=1) P(Y=1) = 0.75 * 0 * 0.25 * 0.5 * 0.75 * 0.4 = 0
P(X | Y=-1)P(Y=-1)= 0.5 * 0.83 * 0.34 * 0.83 * 0.66 * 0.6 = 0.04629
P(X | Y=1) P(Y=1) <P(X | Y=-1)P(Y=-1) => P(X) prediction = -1 (not read)

4.3

P(Y=1 | X) = (P(X | Y=1) * P(Y=1)) / P(X) => (0 * 0.4) / (0 + 0.04629) = 0 (not read)

4.4

I will submit 4.4, 4.5 in the next assignment.

5 Statement of Collaboration

Josh Ho, we discuss the KNN prediction, pros and cons of chossing value K, and posterior probability in problem 4.
Other than that, I watched the video (https://www.youtube.com/watch?v=CPqOCI0ahss) to get better understanding with Naive Bayes Classifiers.