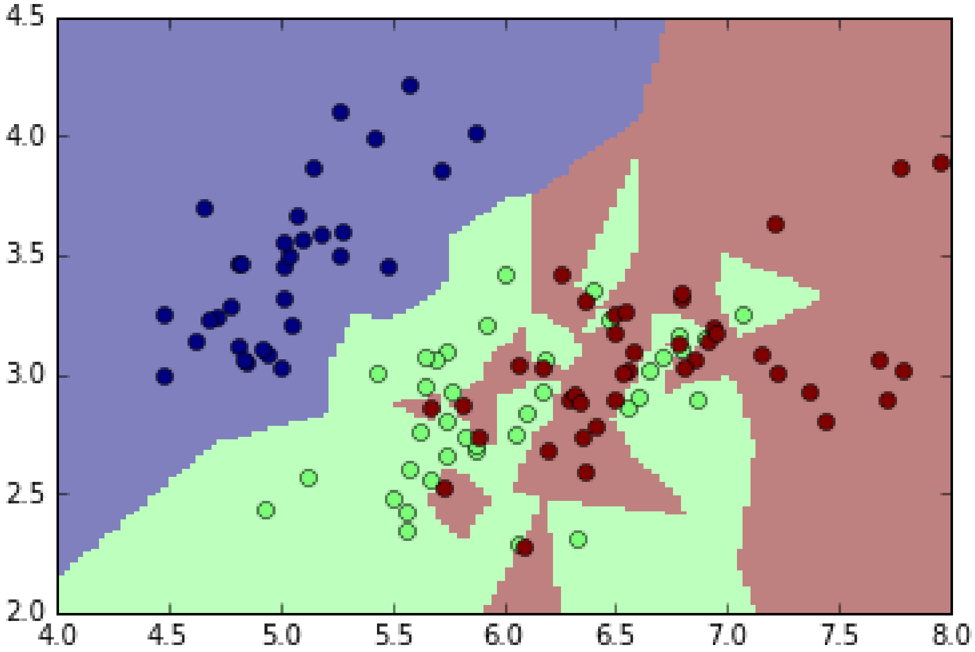
ML_HW1_Problem_2_Code

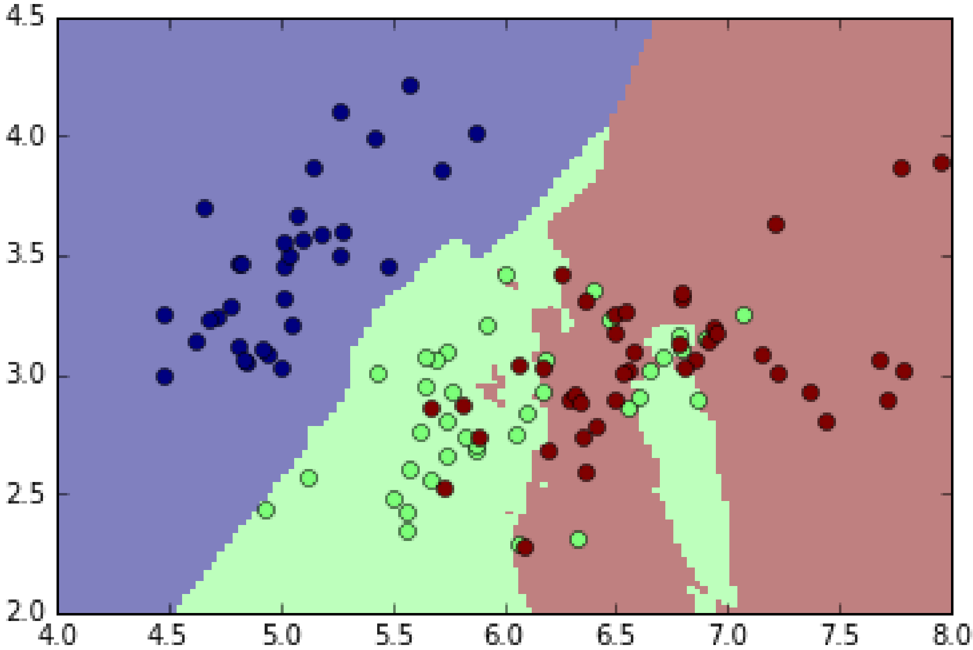
October 3, 2016

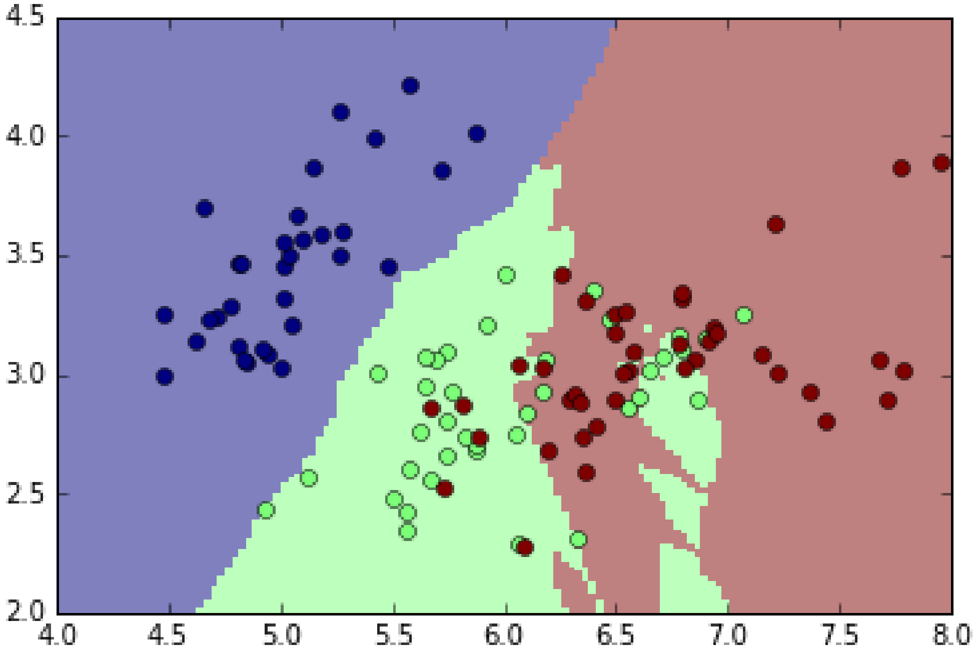
```
In [1]: %matplotlib inline

        import numpy as np
        import matplotlib.pyplot as plt
        import math
        import mltools as ml

        iris = np.genfromtxt("data/iris.txt",delimiter=None) # load the text file
        Y = iris[:,-1] # target value is the last column
        X = iris[:,0:2] # features are the other columns

        X,Y = ml.shuffleData(X,Y)
        Xtr,Xte,Ytr,Yte = ml.splitData(X,Y, 0.75)

        #set  K =1 ,5,10,50
        Kset = [1,5,10,50]


        knn = ml.knn.knnClassify() # create the object and train it
        print("\nSection (a)' Plots are being renderd by K order 1,5,10,50\n")

        #Train it with setting K=1,5,10,50
        for i,K in enumerate(Kset):
           knn.train(Xtr, Ytr, K)
           YteHat = knn.predict(Xte)
           plt.figure(i)
           ml.plotClassify2D(knn,Xtr,Ytr );

        Kset=[1,2,5,10,50,100,200]

        trainDataCount = len(Ytr)
        testDataCount  = len(Yte)

        errTrain =np.empty(7)    # 7 is the length of Kset , its hardcoded or use le
        errTest  =np.empty(7)

        for i,k in enumerate(Kset):
         knn.train(Xtr, Ytr, k)
```

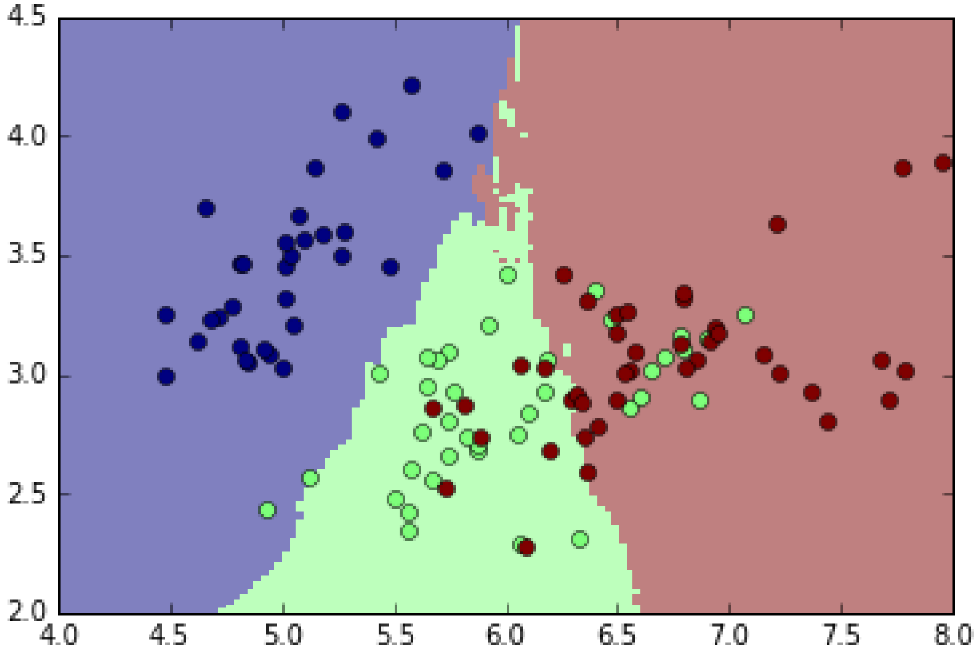
```
 YtrHat = knn.predict(Xtr)
 errTrain[i]= np.sum(Ytr !=YtrHat) / trainDataCount
 #print("Error Rate  Validating using Training Data and K = "+str(k) + " is

print('\n')
for i,k in enumerate(Kset):
 knn.train(Xtr, Ytr, k)
 YteHat = knn.predict(Xte)
 errTest[i]=  np.sum(Yte !=YteHat) / testDataCount
 #print("Error Rate Validating using Test Data and K = "+str(k) + " is : "

plt.figure(4)
plt.semilogx(Kset,errTrain,color='red')
plt.semilogx(Kset,errTest,color='green')
plt.show()

print("THE K to choose is the one that minimizes the curve of error on Test
print("Recommended K is : " + str(Kset[np.argmin(errTest)]))




#K=1
#knn.train(Xtr, Ytr, K)
#YteHat = knn.predict(Xte)
#plt.figure(1)
#ml.plotClassify2D( knn, Xtr, Ytr )


#K=5
#knn.train(Xtr, Ytr, K)
#YteHat = knn.predict(Xte)
#plt.figure(2)
#ml.plotClassify2D( knn, Xtr, Ytr )


#K=10
#knn.train(Xtr, Ytr, K)
#YteHat = knn.predict(Xte)
#plt.figure(3)
#ml.plotClassify2D( knn, Xtr, Ytr )

#K=50
#knn.train(Xtr, Ytr, K)
#YteHat = knn.predict(Xte)
#plt.figure(4)
#ml.plotClassify2D( knn, Xtr, Ytr )
```

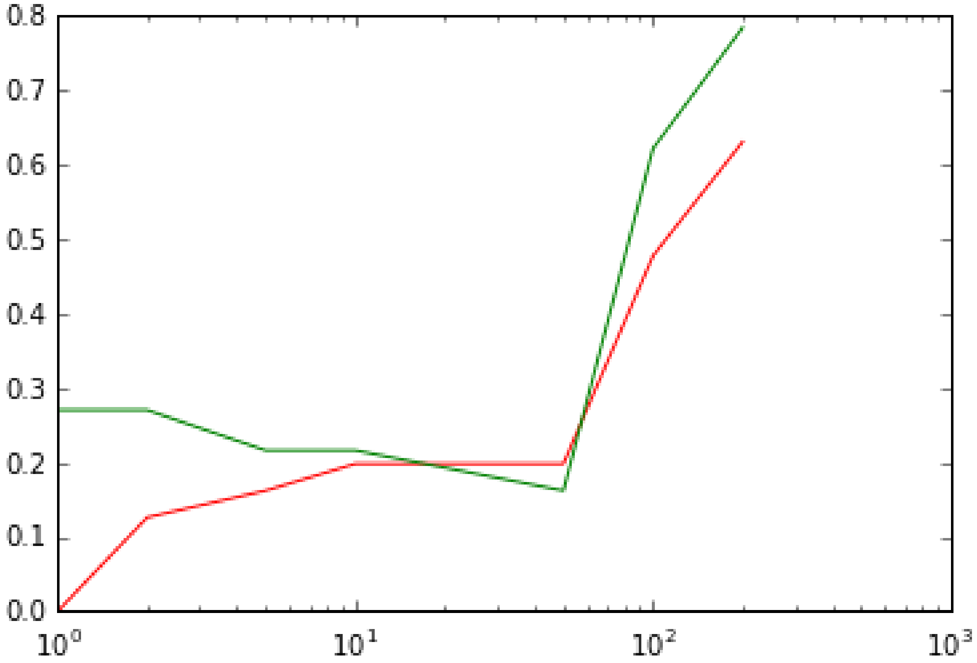
```
Section (a)' Plots are being renderd by K order 1,5,10,50
```




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





```
THE K to choose is the one that minimizes the curve of error on Testing Data
Recommended K is : 50
```

```
In [ ]:
```