Data Mining_Assignment 2
Spring 2017
Kaiqin Huang

Install Anaconda (be sure to use the Python 3.6 version): https://www.continuum.io/downloads

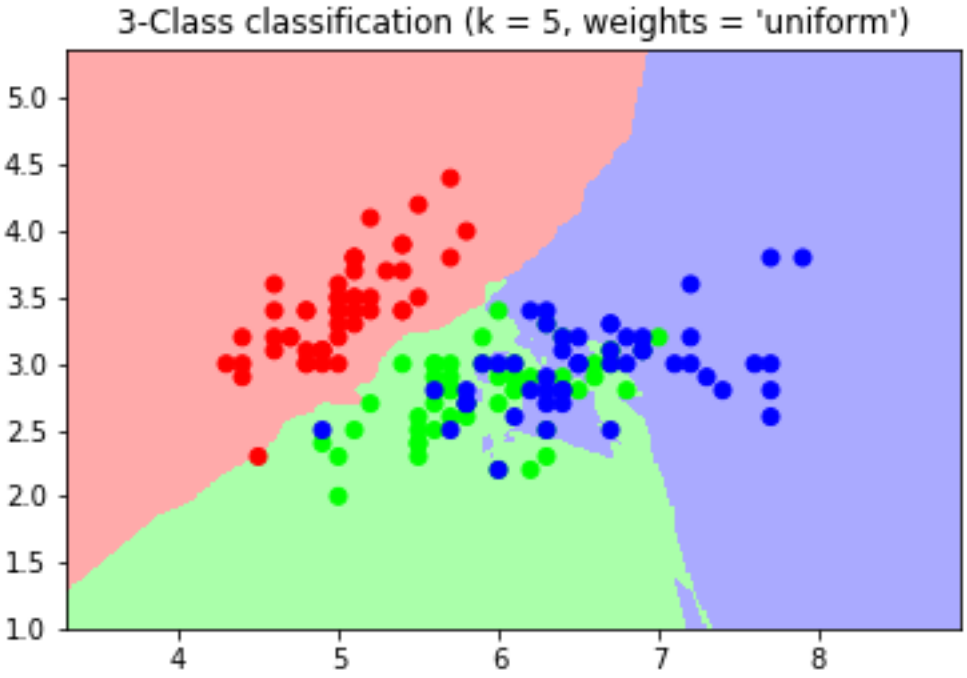
Read section 10.8 in the text and this sci-kit learn overview of nearest neighbor (NN):

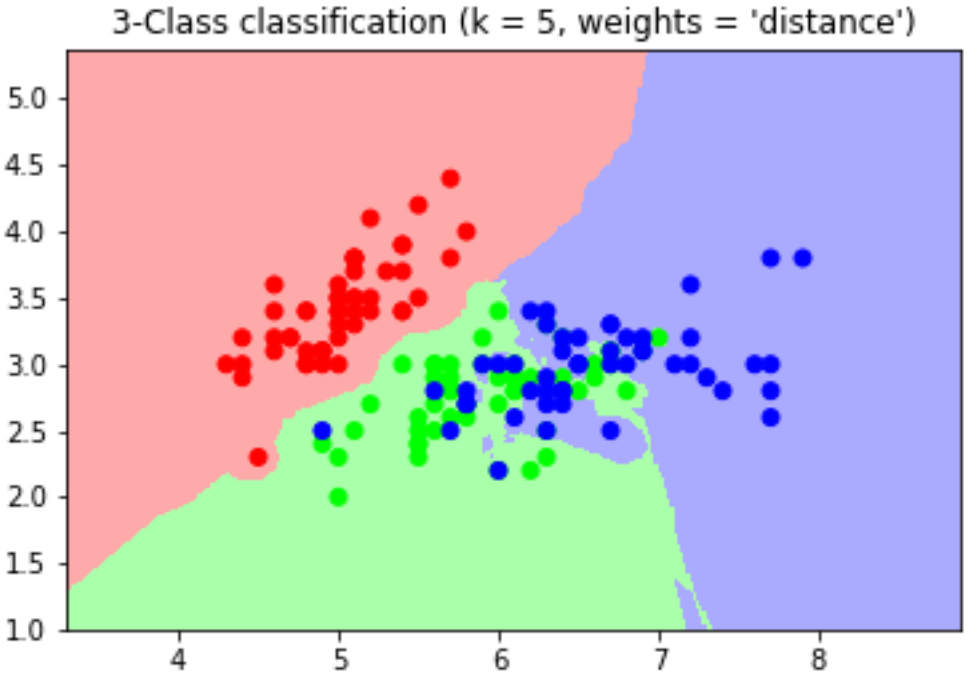
http://scikit-learn.org/stable/modules/neighbors.html

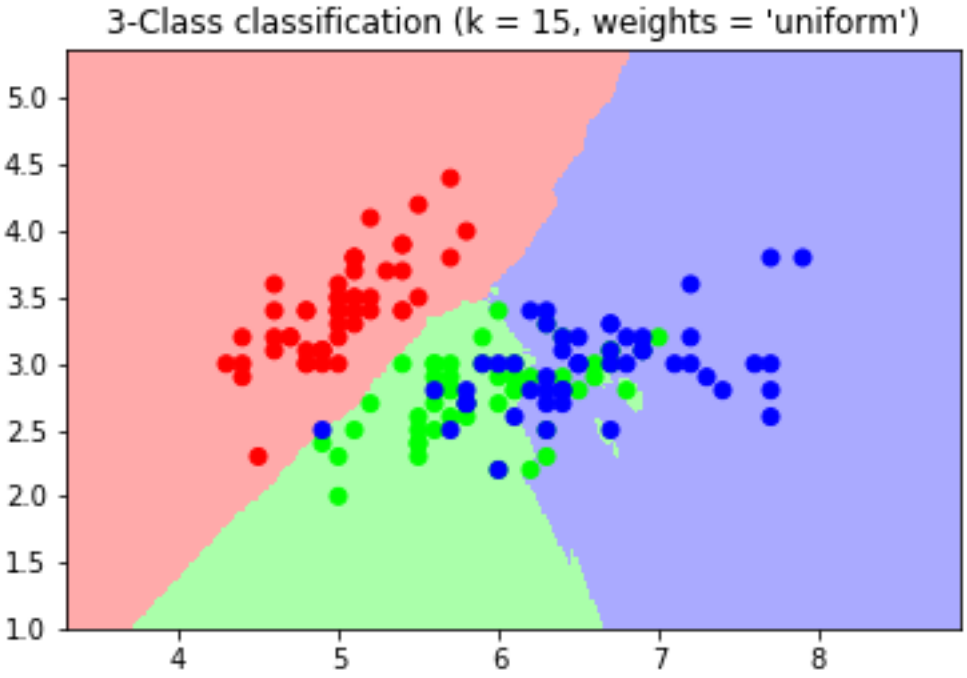
Using Anaconda, execute the example code for NN: http://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py

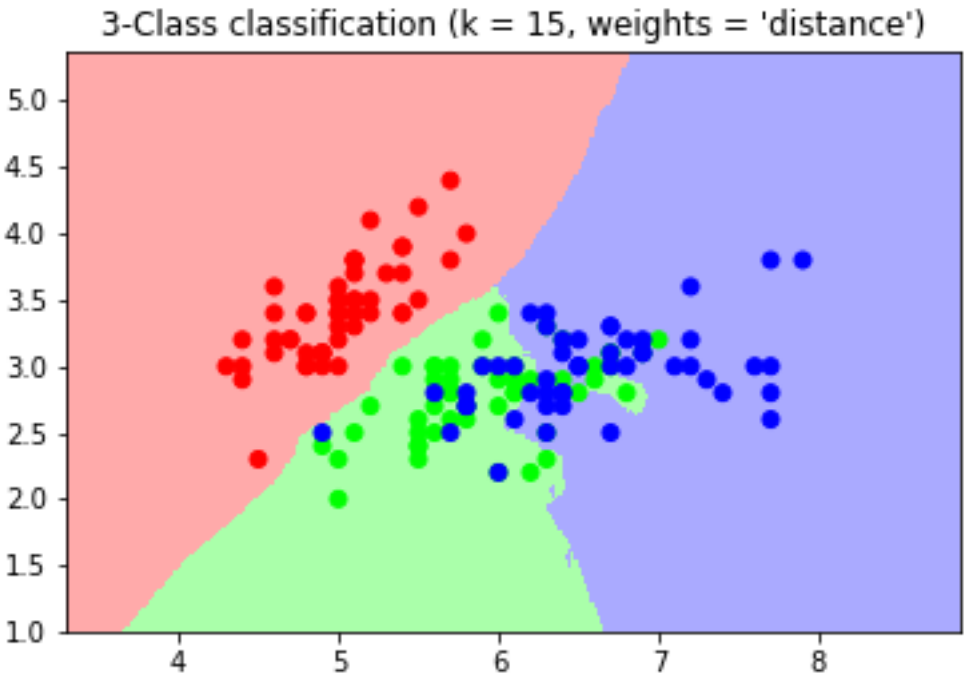
Alter the code to produce plots by varying the number of nearest neighbors, k, from 5 to 50, in increments of 10. What do you notice about the results?

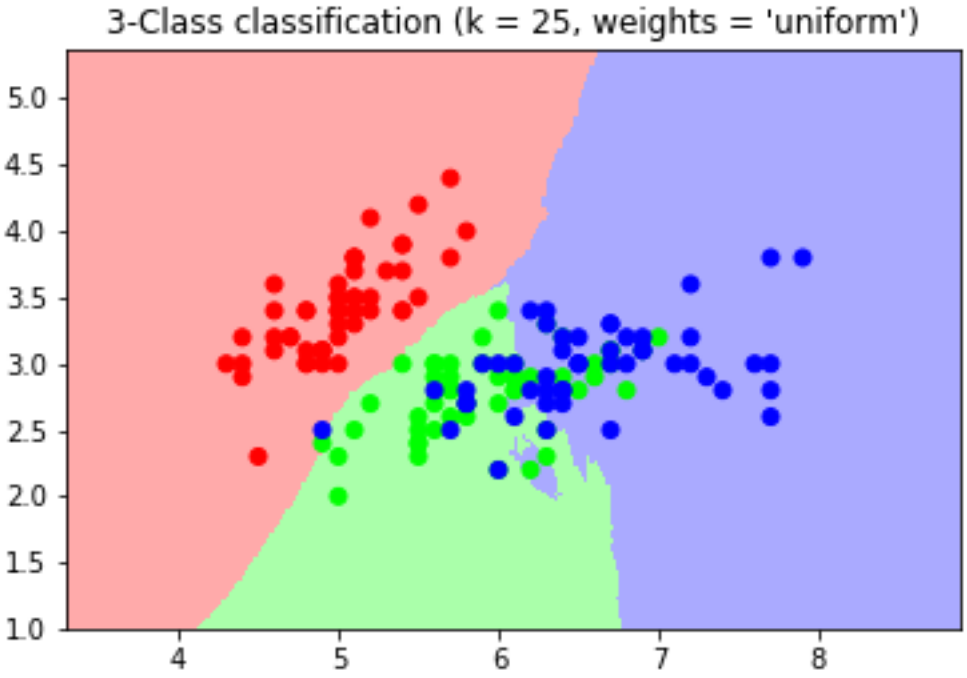
Put your plots in a document along with your analysis and submit

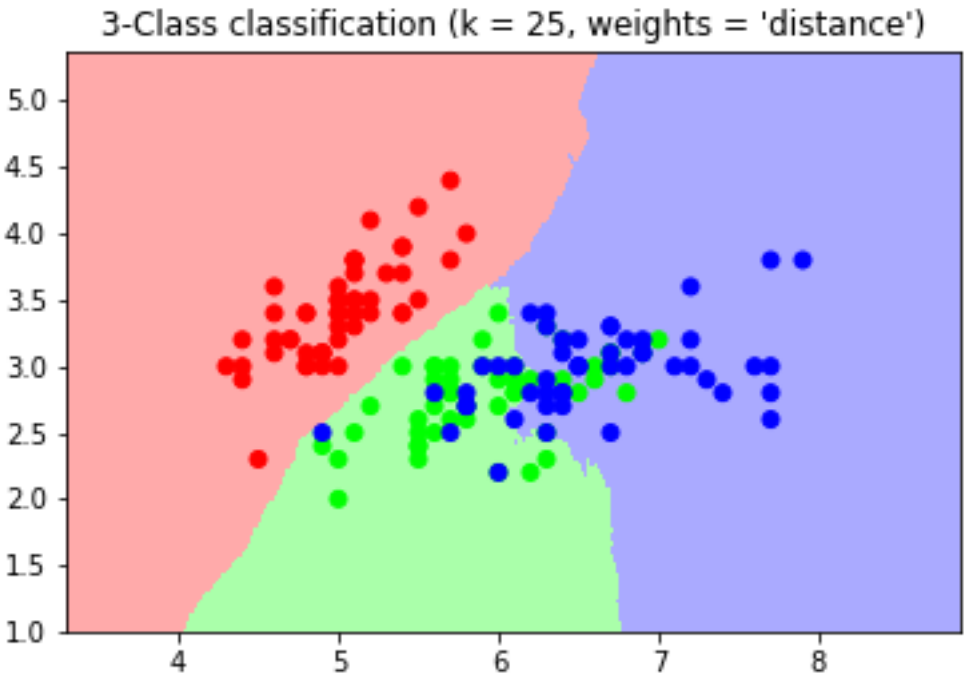

3-Class classification (k = 5, weights = 'uniform')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8

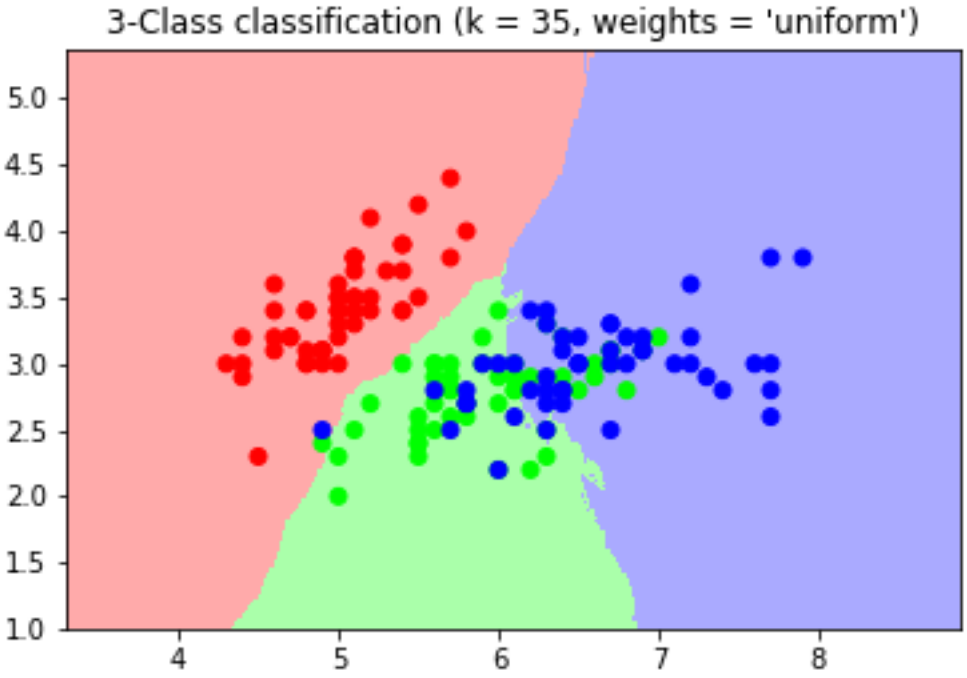

3-Class classification (k = 5, weights = 'distance')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8

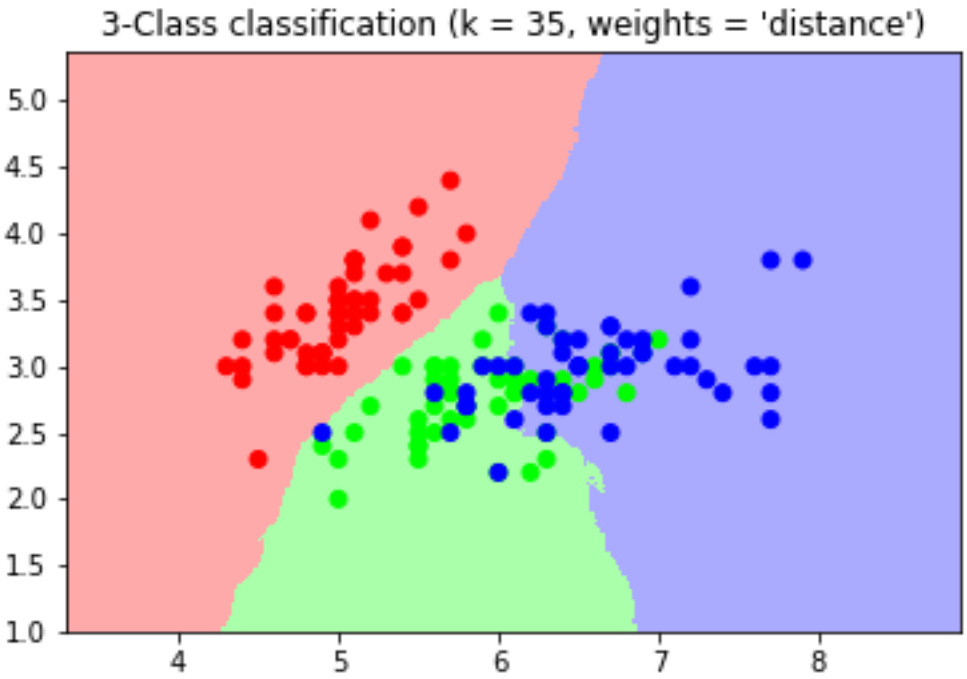

3-Class classification (k = 15, weights = 'uniform')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8

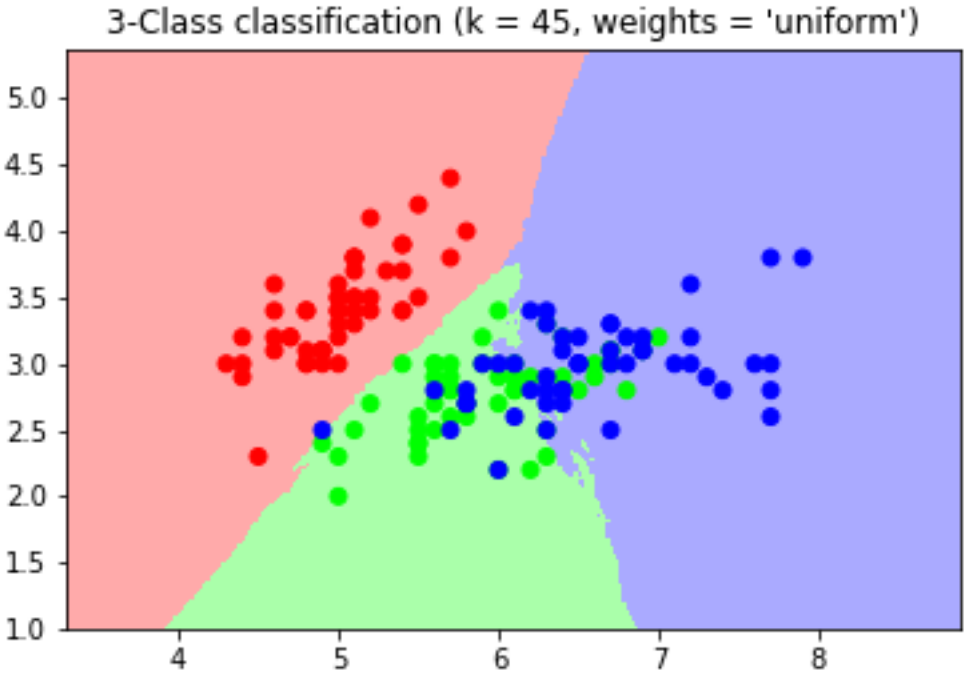

3-Class classification (k = 15, weights = 'distance')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8

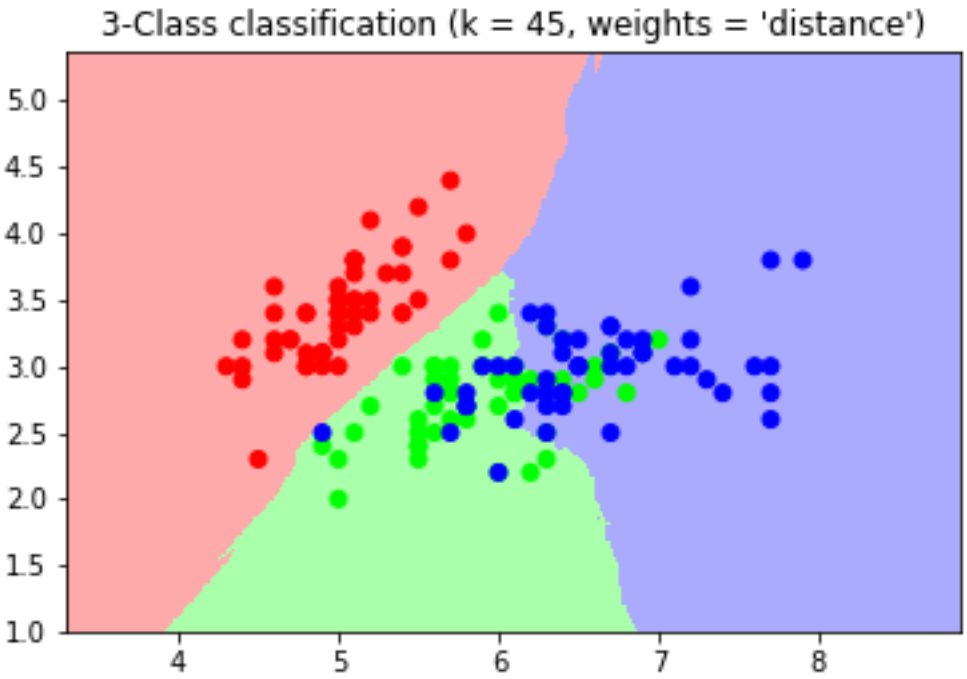

3-Class classification (k = 25, weights = 'uniform')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8


3-Class classification (k = 25, weights = 'distance')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8


3-Class classification (k = 35, weights = 'uniform')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8


3-Class classification (k = 35, weights = 'distance')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8


3-Class classification (k = 45, weights = 'uniform')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8


3-Class classification (k = 45, weights = 'distance')
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8