Machine Learning Department
School of Computer Science
Carnegie Mellon University

Experimental Design
+
k-Nearest Neighbors

KNN Readings:
Mitchell 8.2
HTF 13.3
Murphy ---
Bishop 2.5.2

Prob. Readings: (next lecture)
Lecture notes from 10-600
(See Piazza post for the pointers)

Murphy 2
Bishop 2
HTF --
Mitchell --

Matt Gormley
Lecture 3
January 25, 2016

Reminders

- Background Exercises (Homework 1)
  - Released: Wed, Jan. 25
  - Due: Mon, Jan. 30 at 5:30pm
- Website updates
  - Office hours Google calendar on "People"
  - Readings on "Schedule"
- Meet AIs: Sarah, Daniel, Brynn

Outline

- **k-Nearest Neighbors (KNN)**
  - Special cases
  - Choosing k
  - Case Study: KNN on Fisher Iris Data
  - Case Study: KNN on 2D Gaussian Data
- **Experimental Design**
  - Train error vs. test error
  - Train / validation / test splits
  - Cross-validation
- **Function Approximation View of ML**

K-NEAREST NEIGHBORS

k-Nearest Neighbors

Whiteboard:

- Special cases
- Choosing k

KNN ON FISHER IRIS DATA

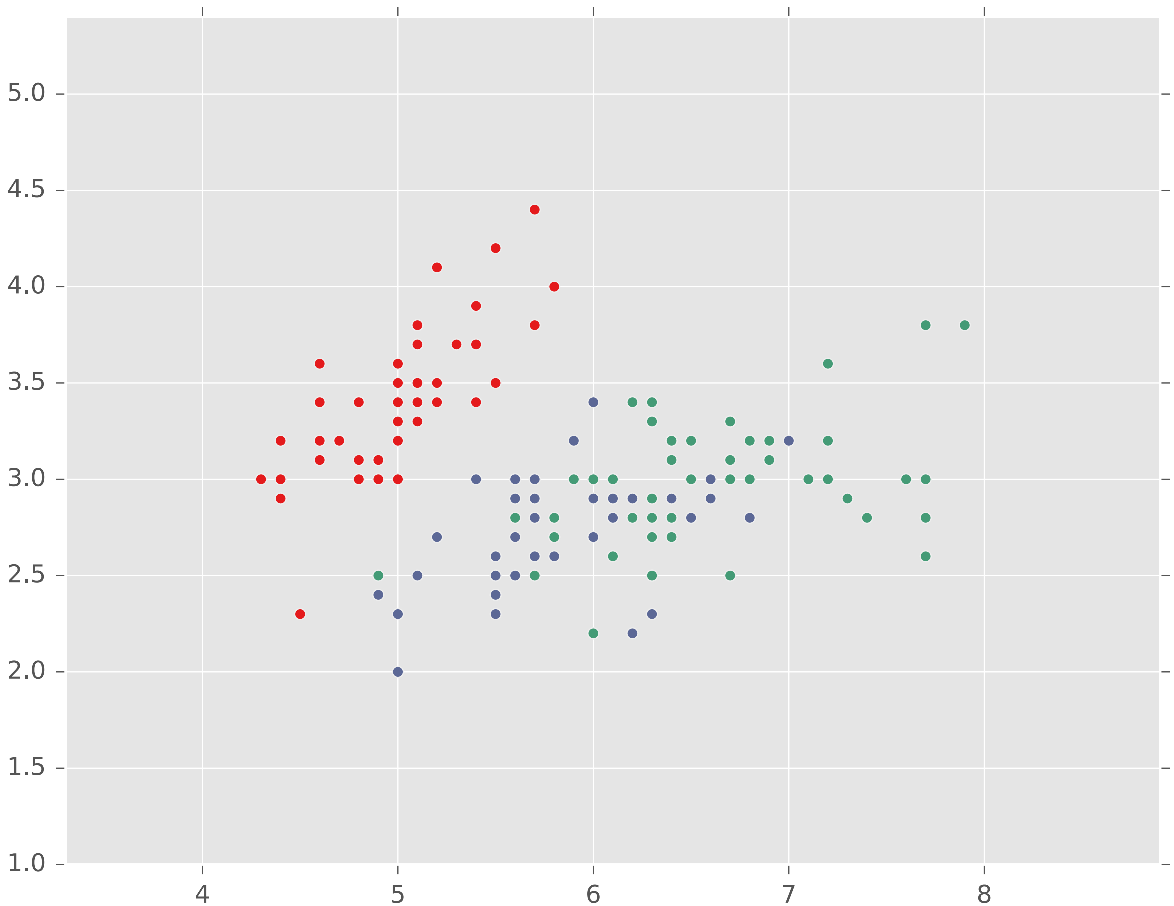
Fisher Iris Dataset

Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
1	6.7	3.0	5.0	1.7

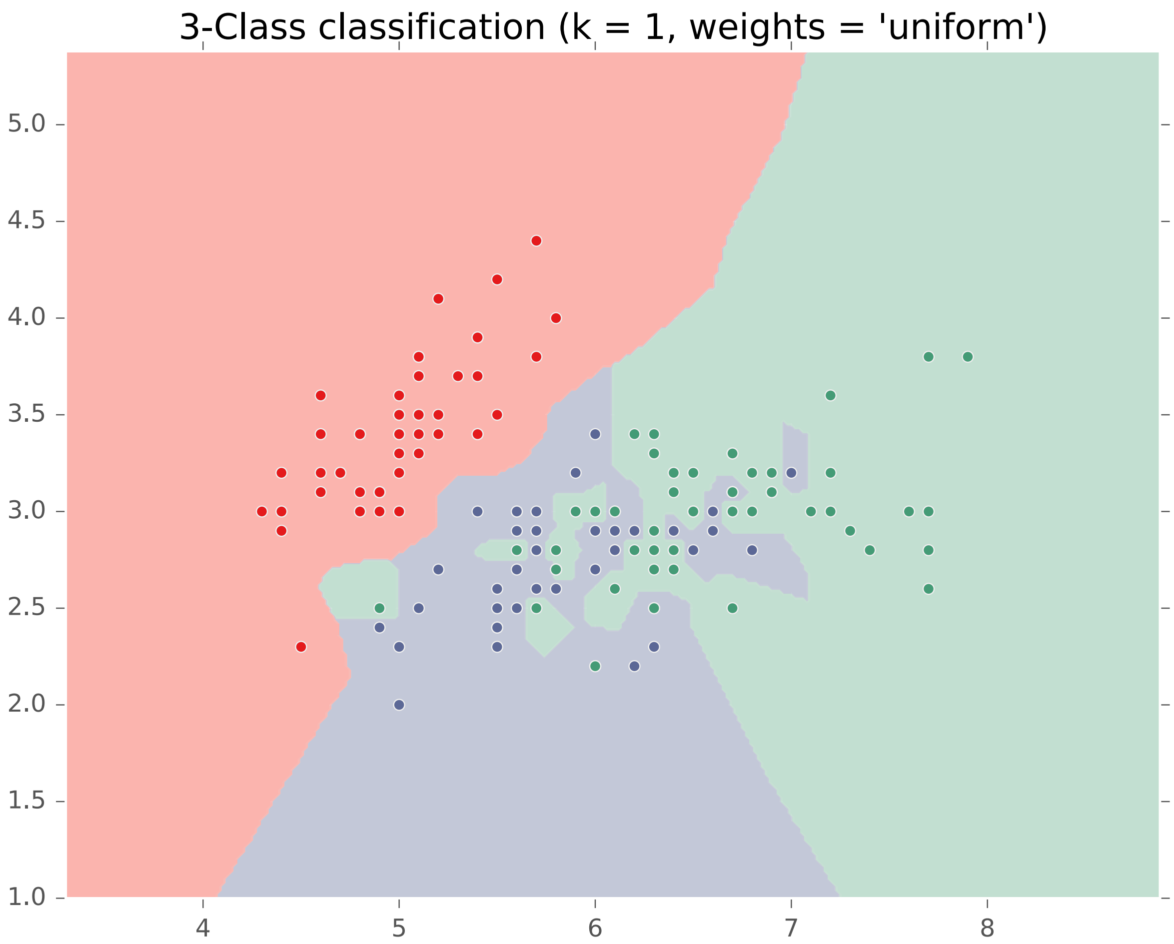
Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

KNN on Fisher Iris Data



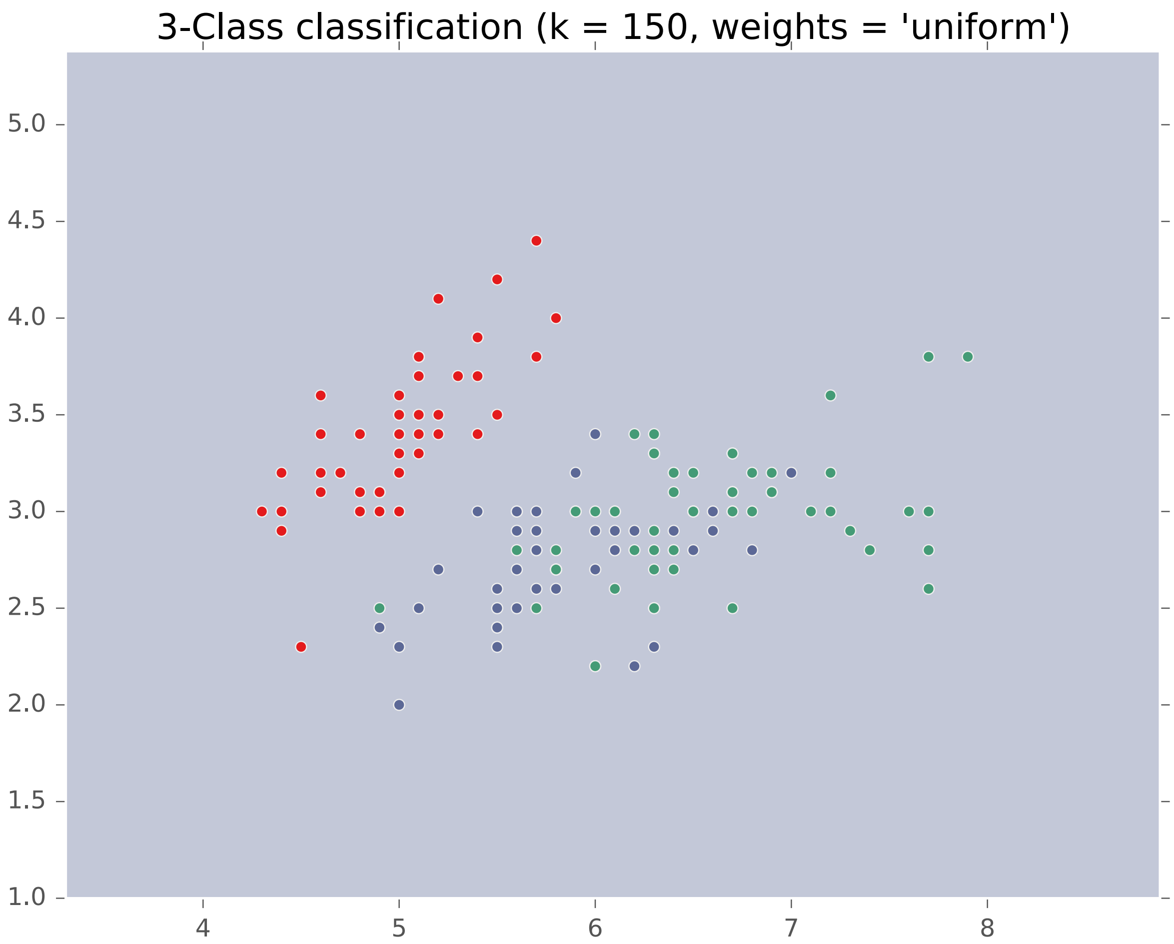
KNN on Fisher Iris Data

Special Case: Nearest Neighbor



KNN on Fisher Iris Data

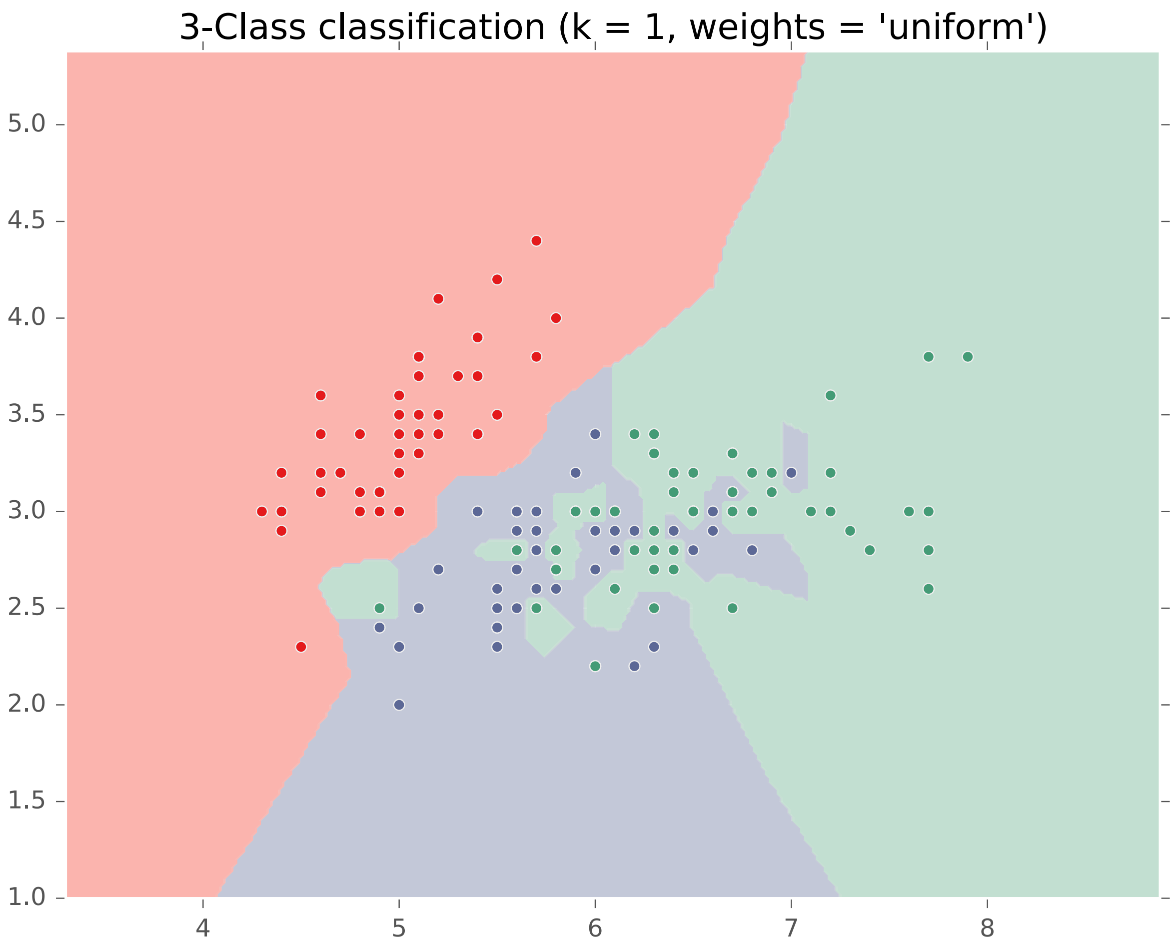
Special Case: Majority Vote



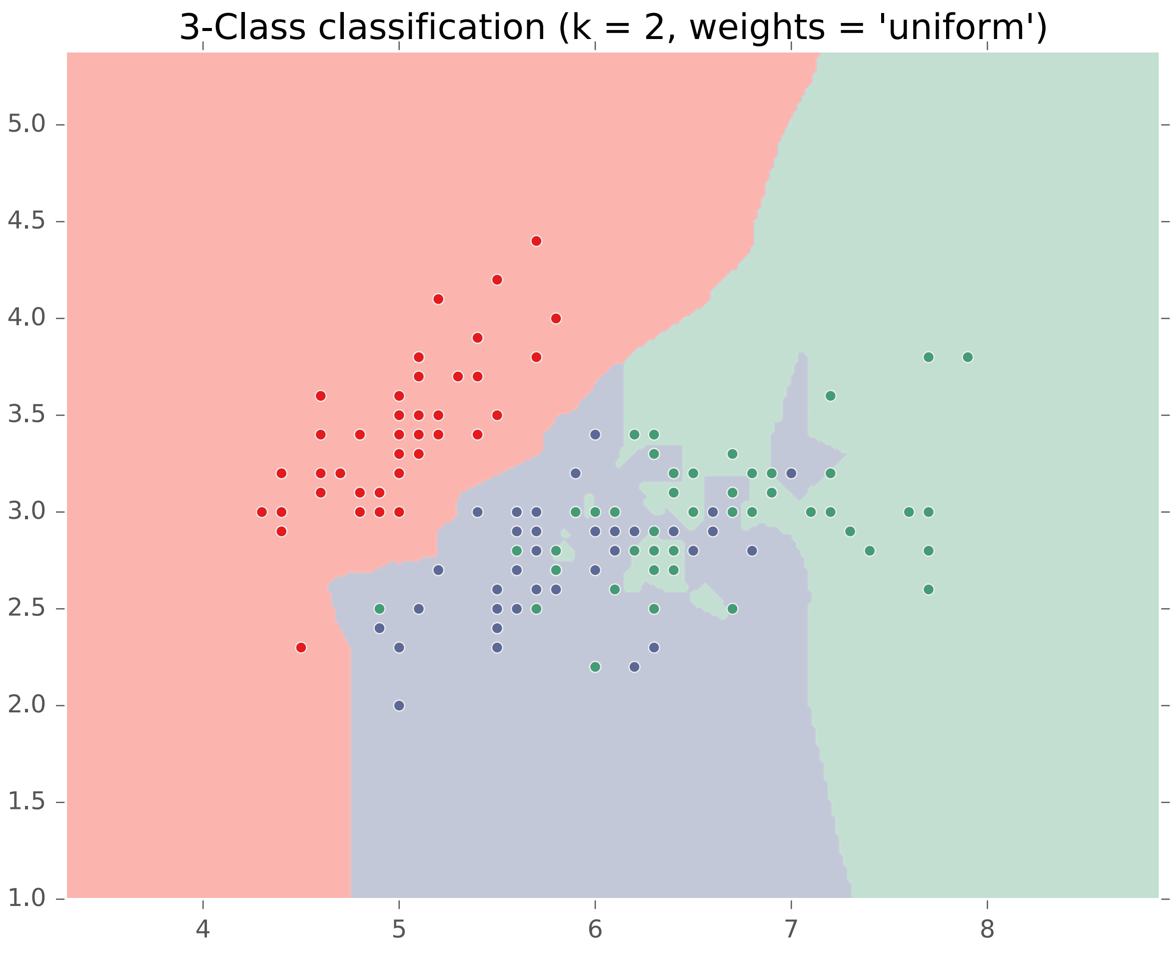
KNN on Fisher Iris Data

KNN on Fisher Iris Data

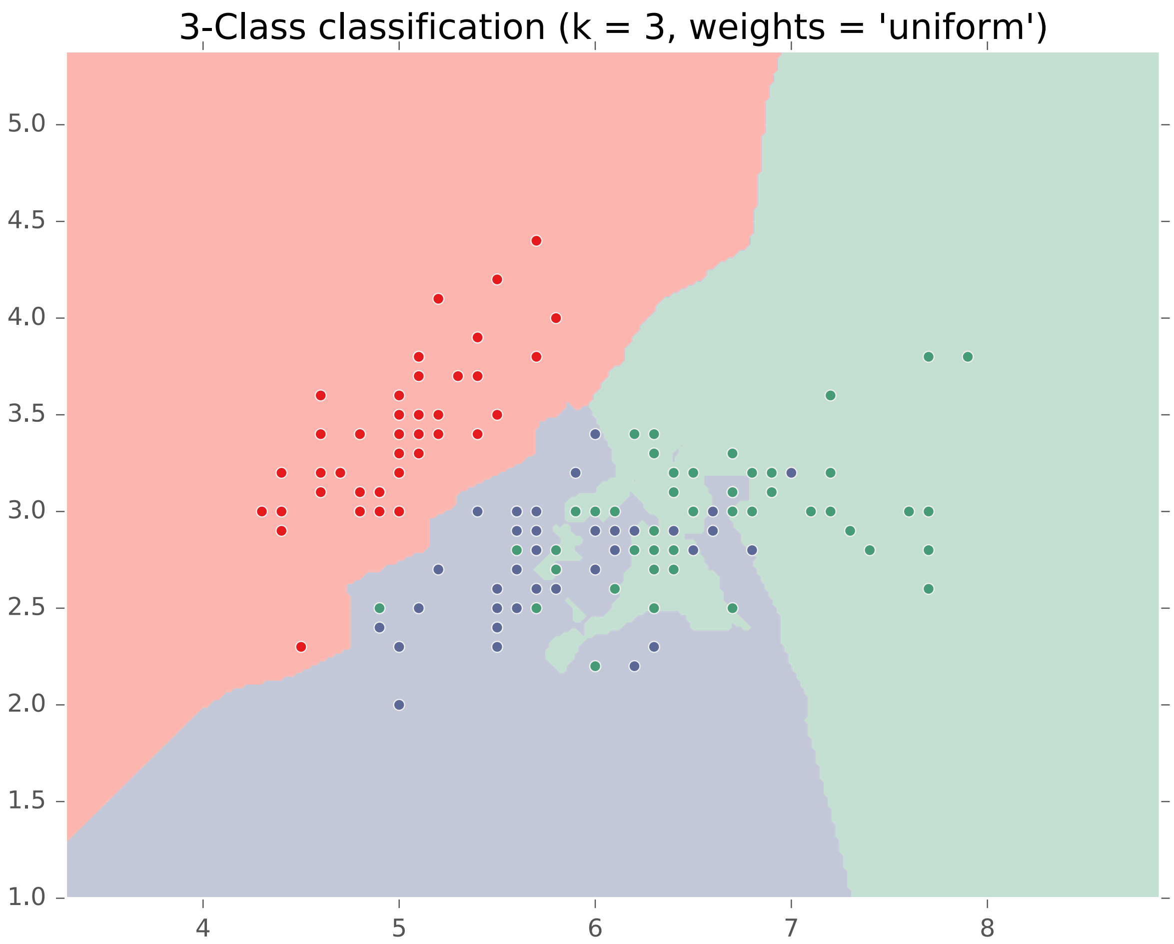
Special Case: Nearest Neighbor



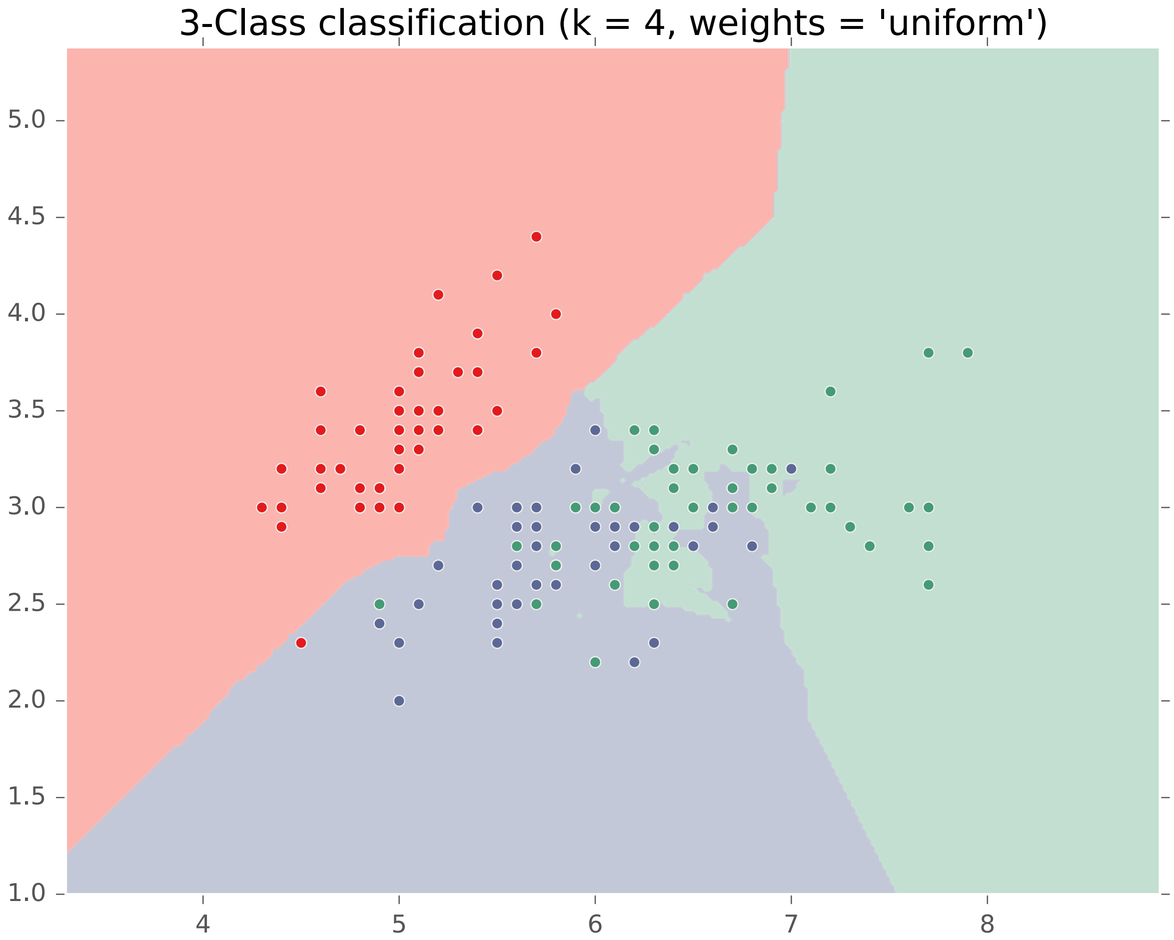
KNN on Fisher Iris Data



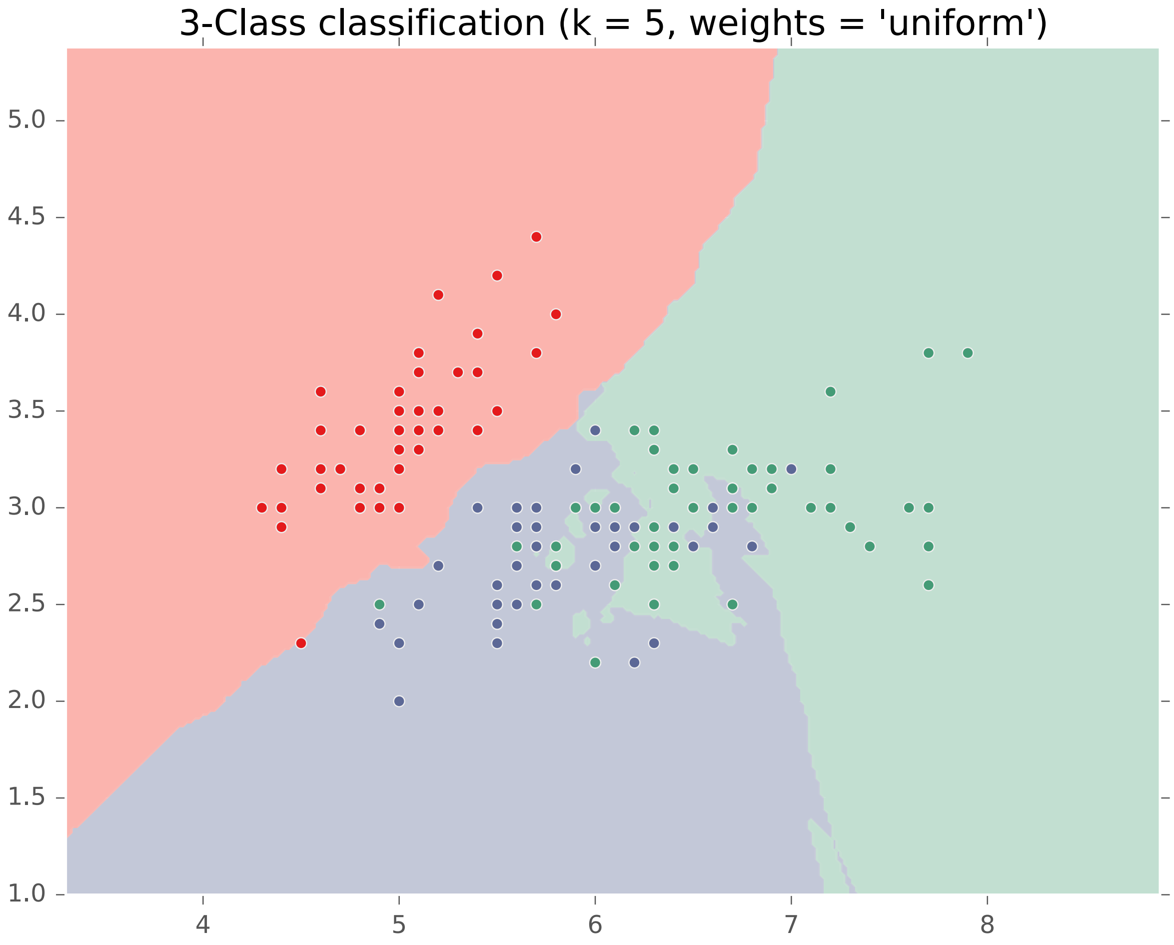
KNN on Fisher Iris Data



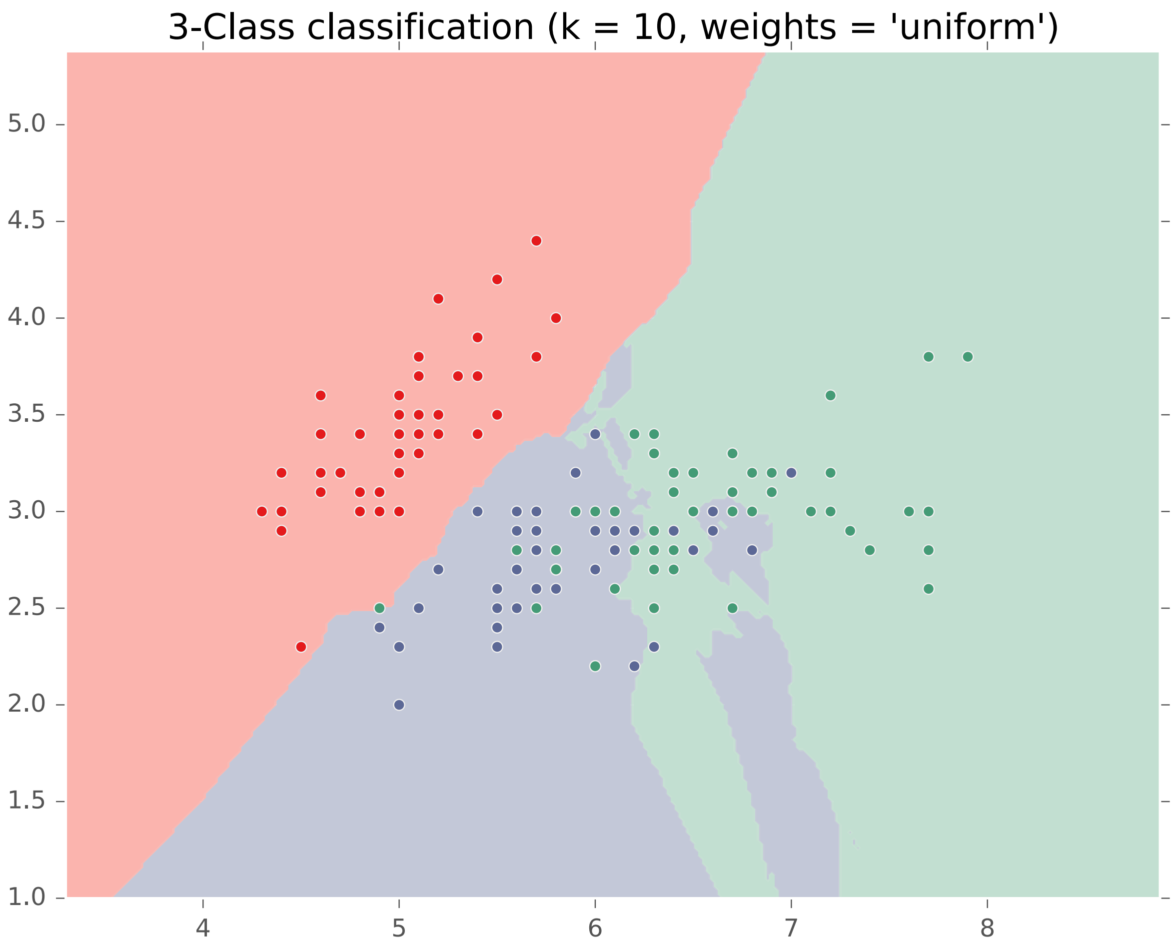
KNN on Fisher Iris Data



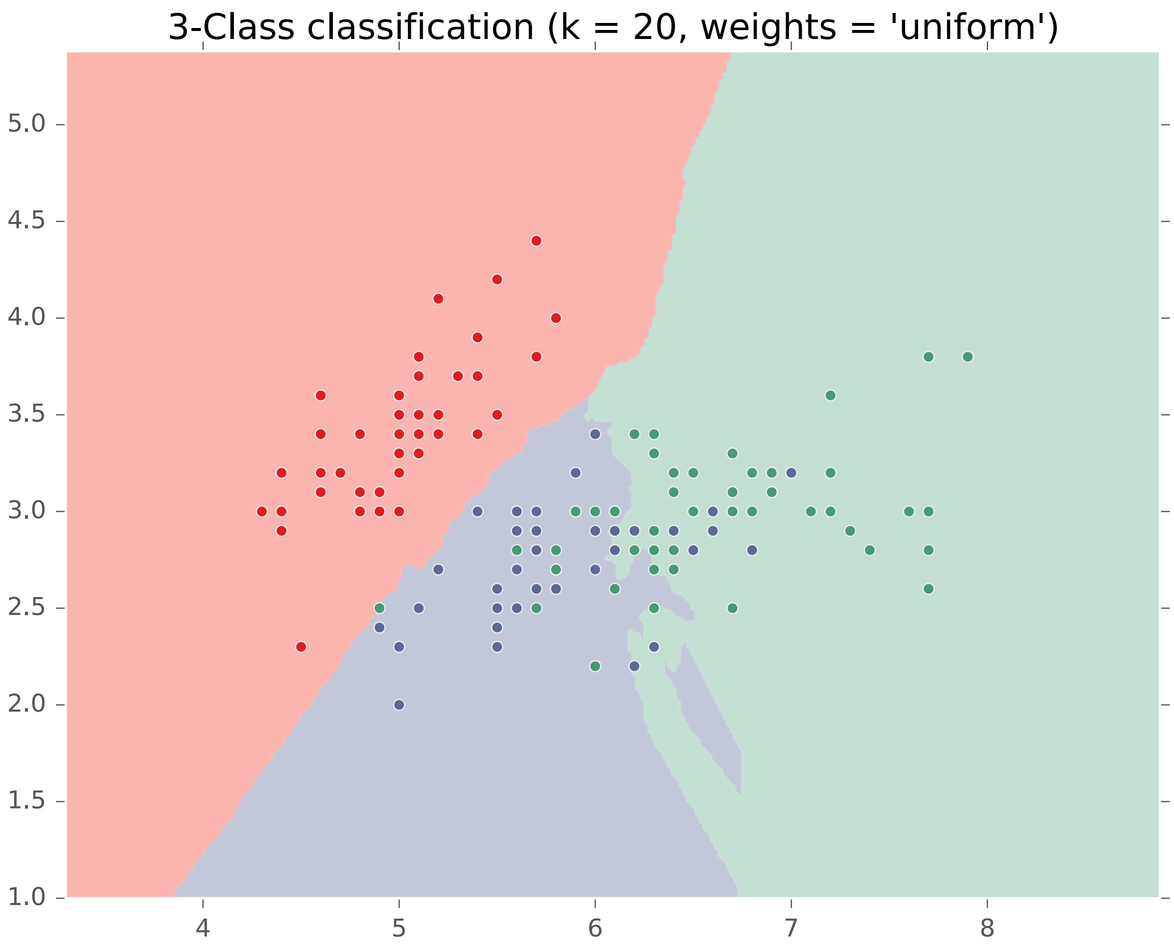
KNN on Fisher Iris Data



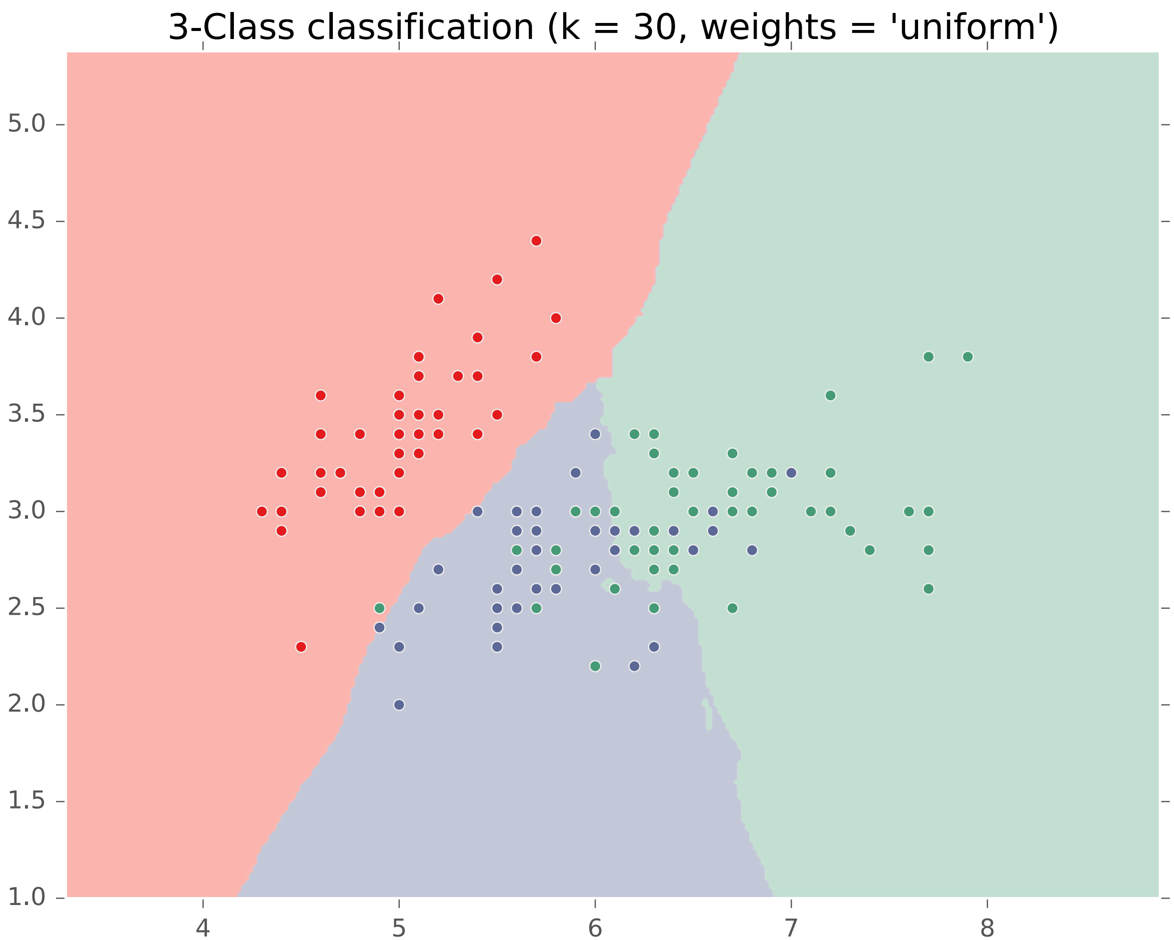
KNN on Fisher Iris Data



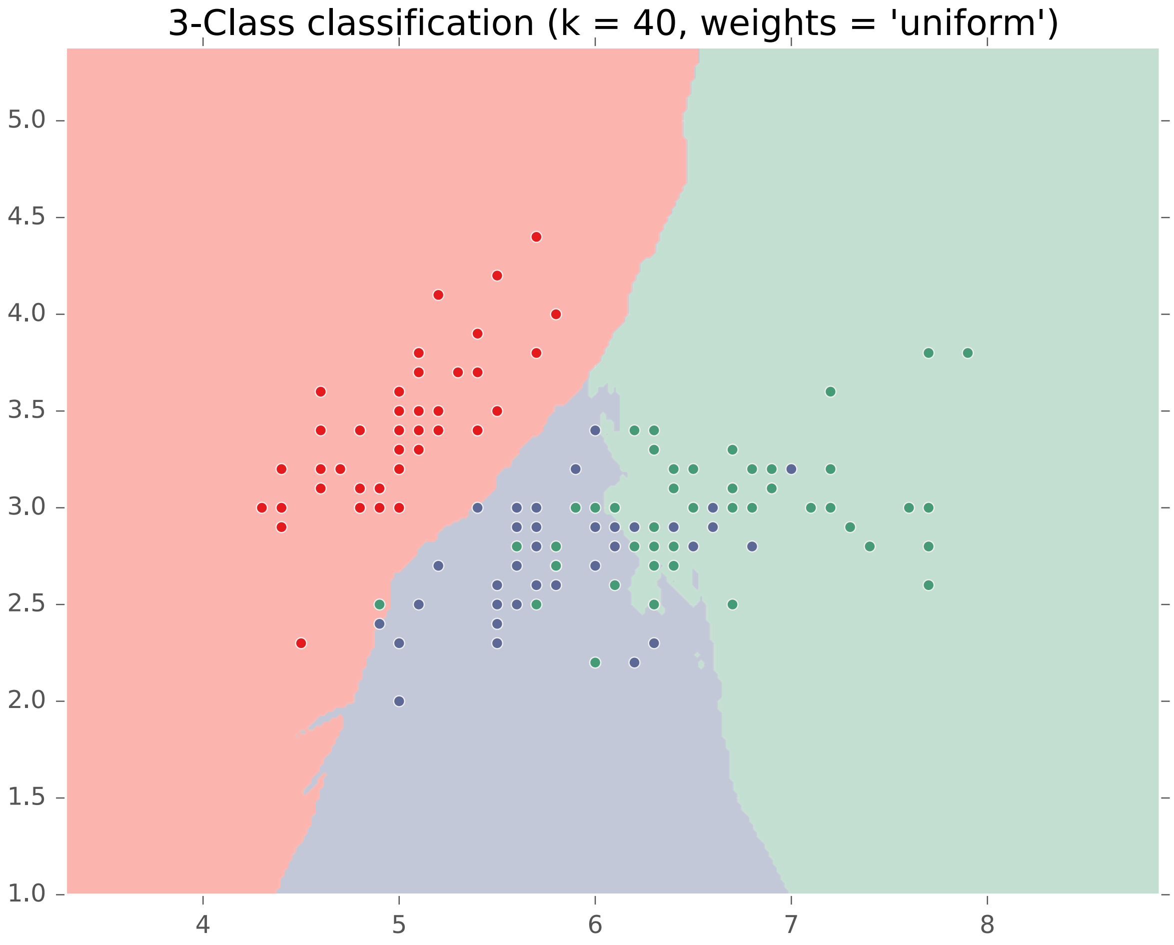
KNN on Fisher Iris Data



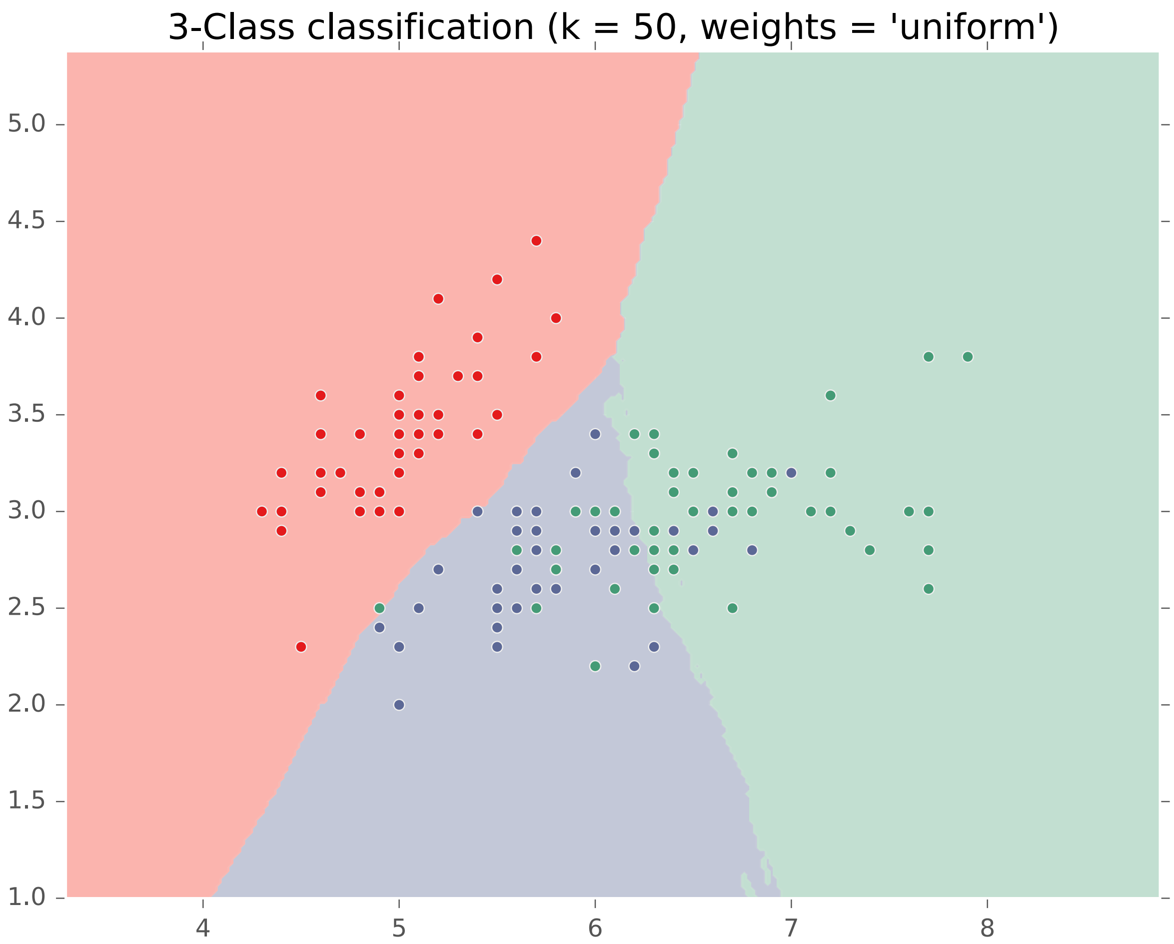
KNN on Fisher Iris Data



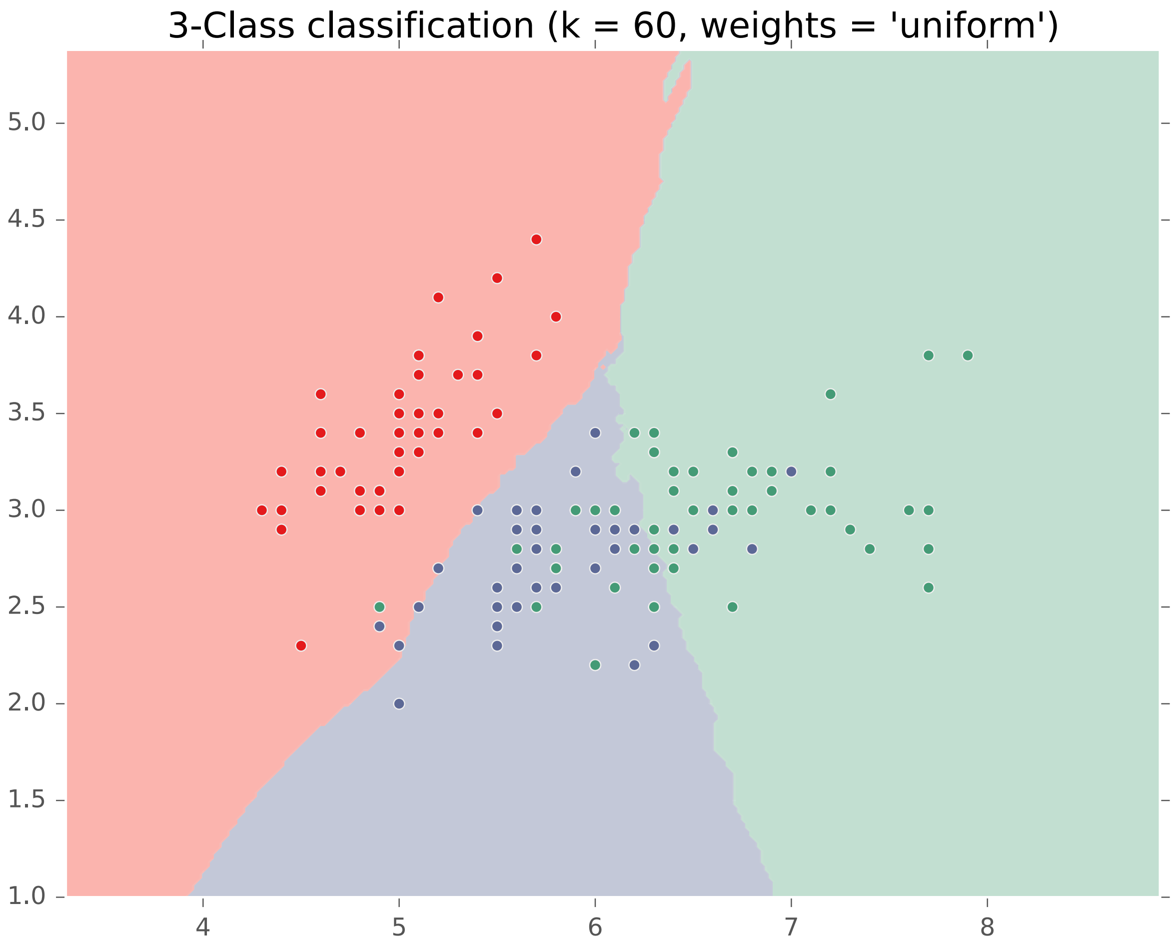
KNN on Fisher Iris Data



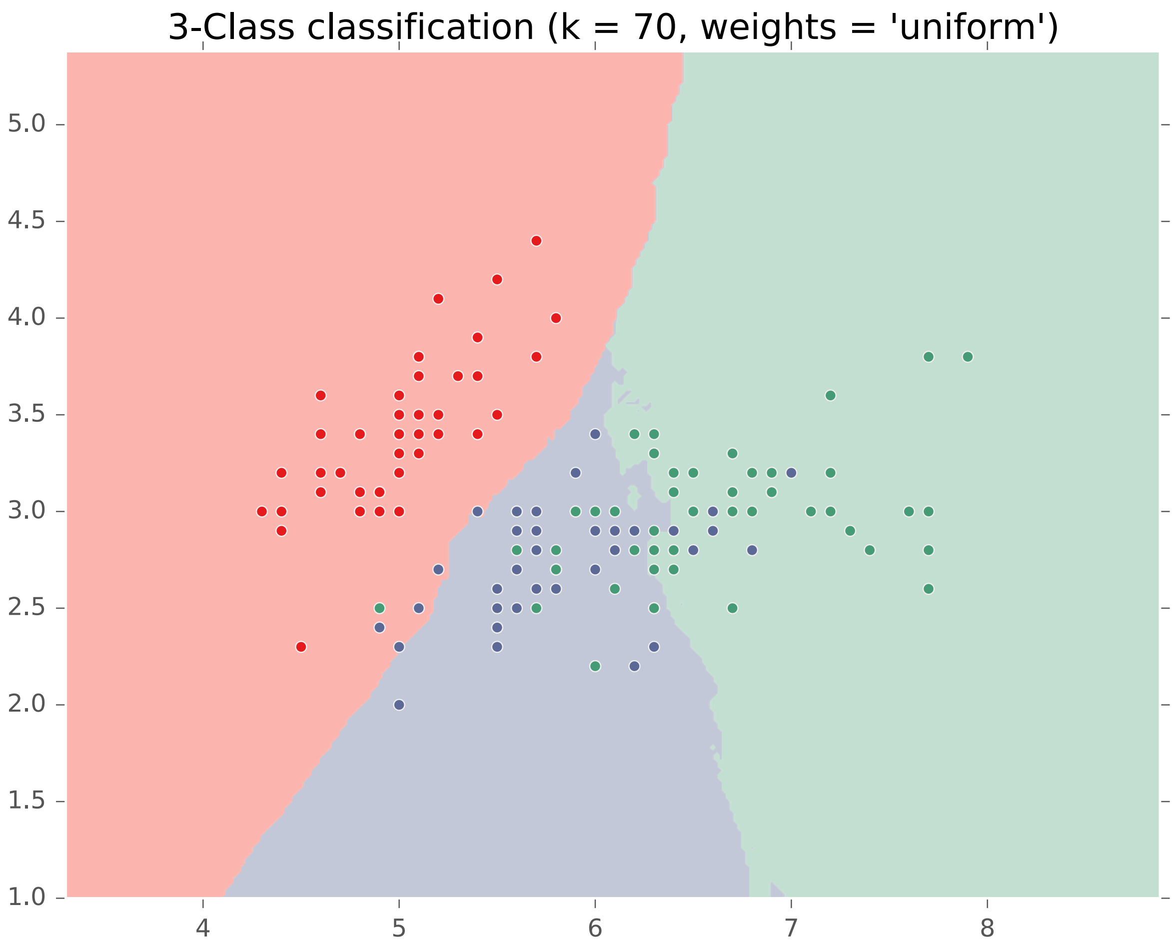
KNN on Fisher Iris Data



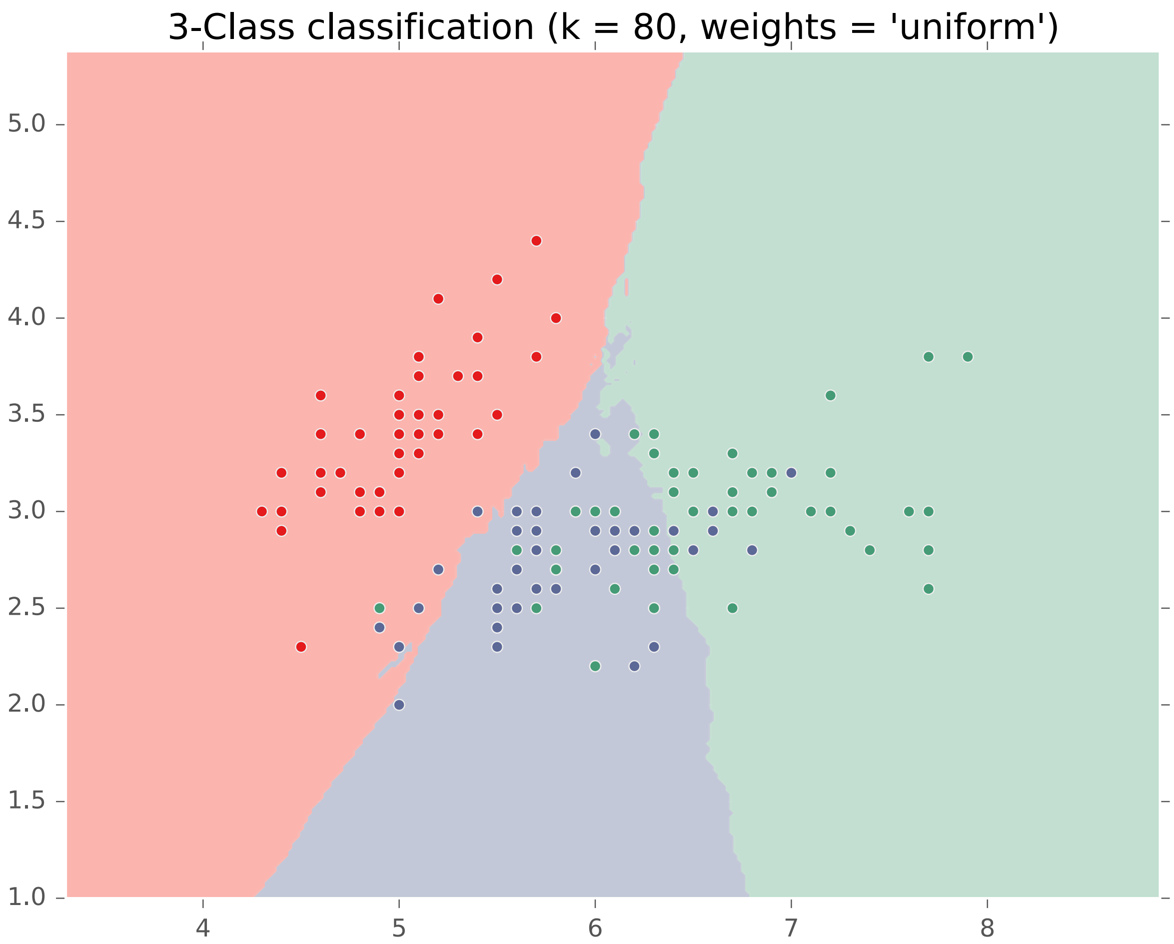
KNN on Fisher Iris Data



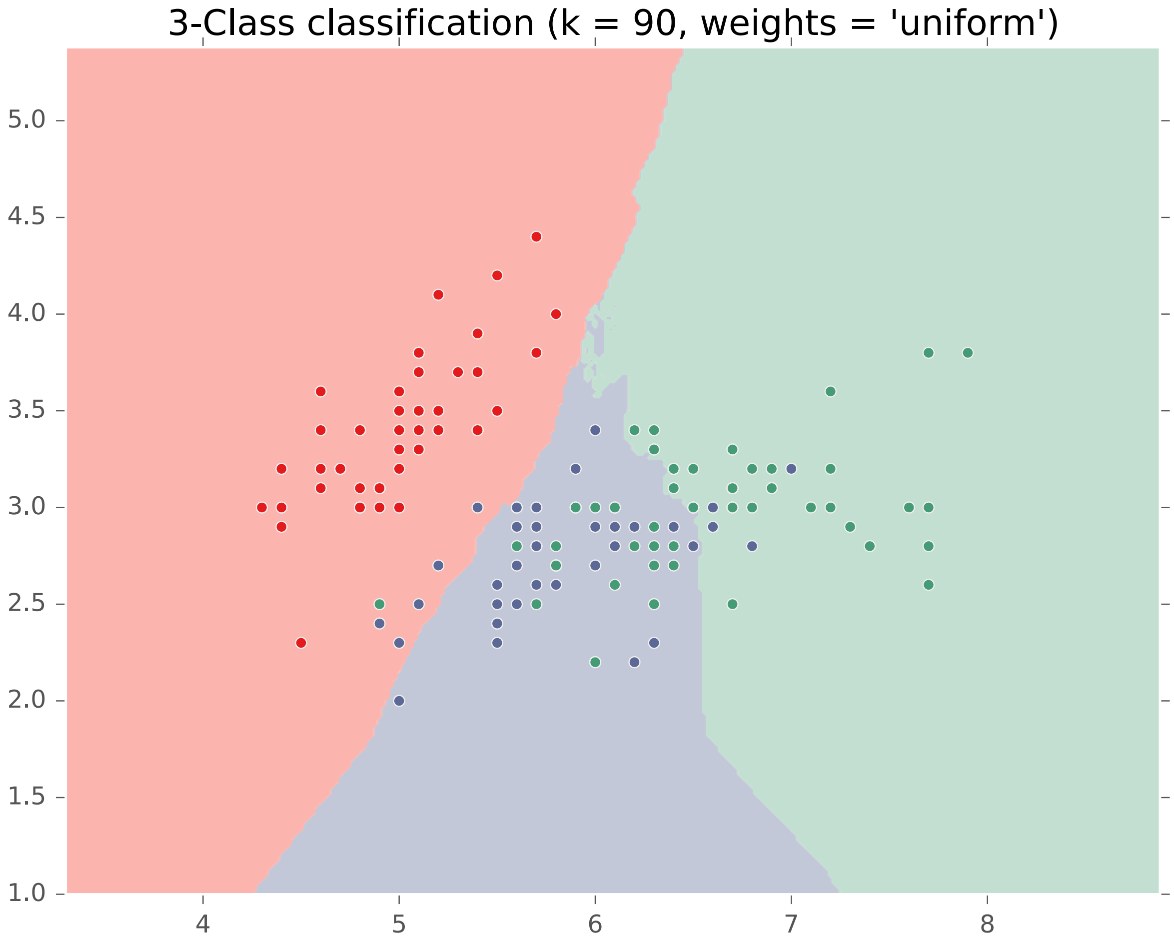
KNN on Fisher Iris Data



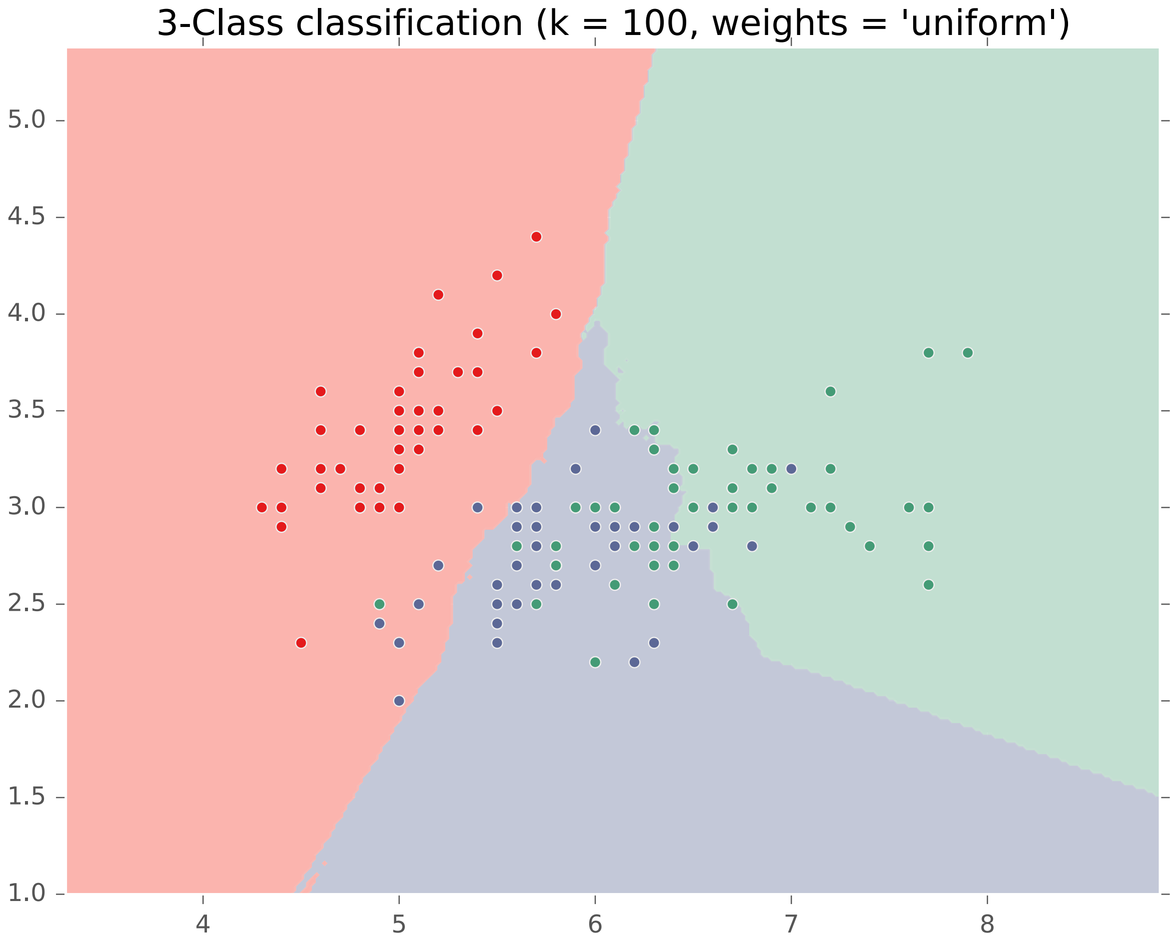
KNN on Fisher Iris Data



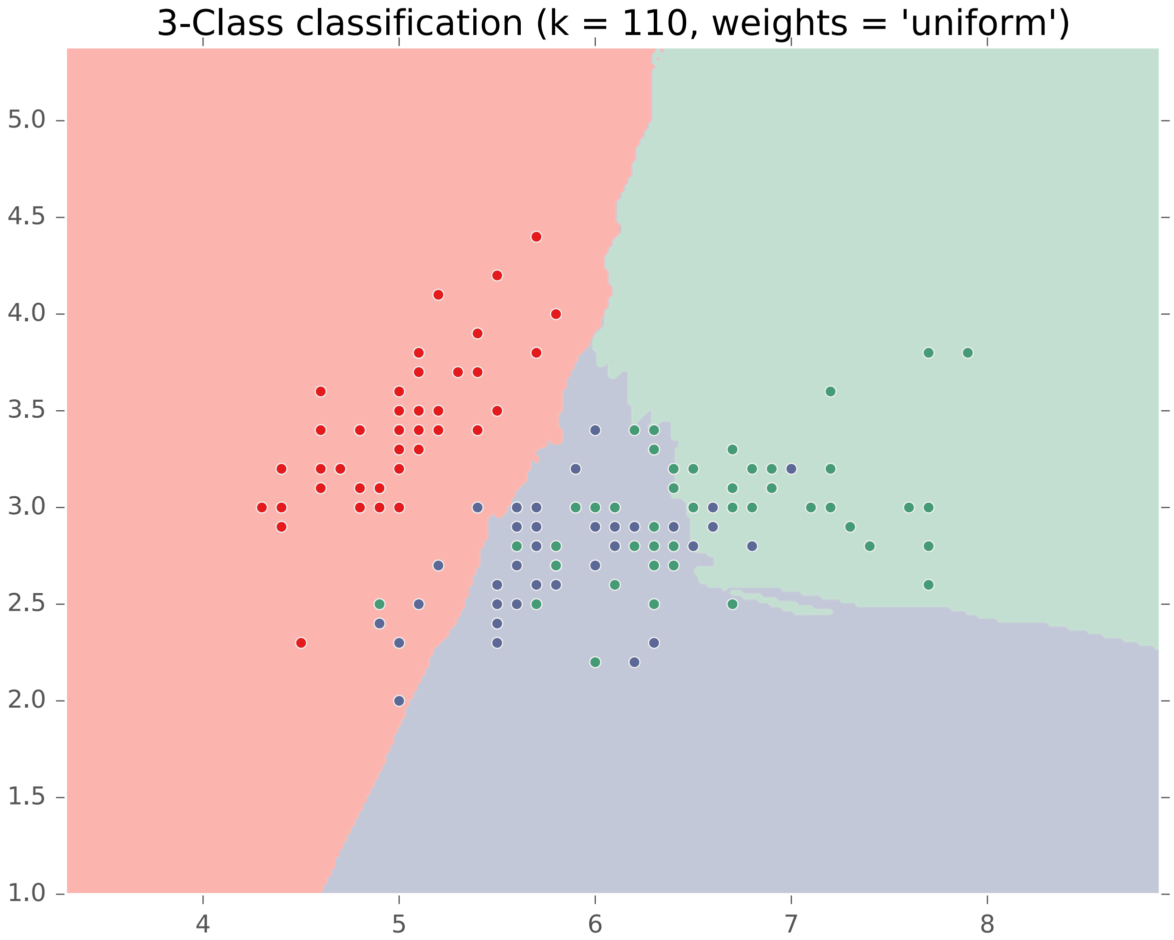
KNN on Fisher Iris Data



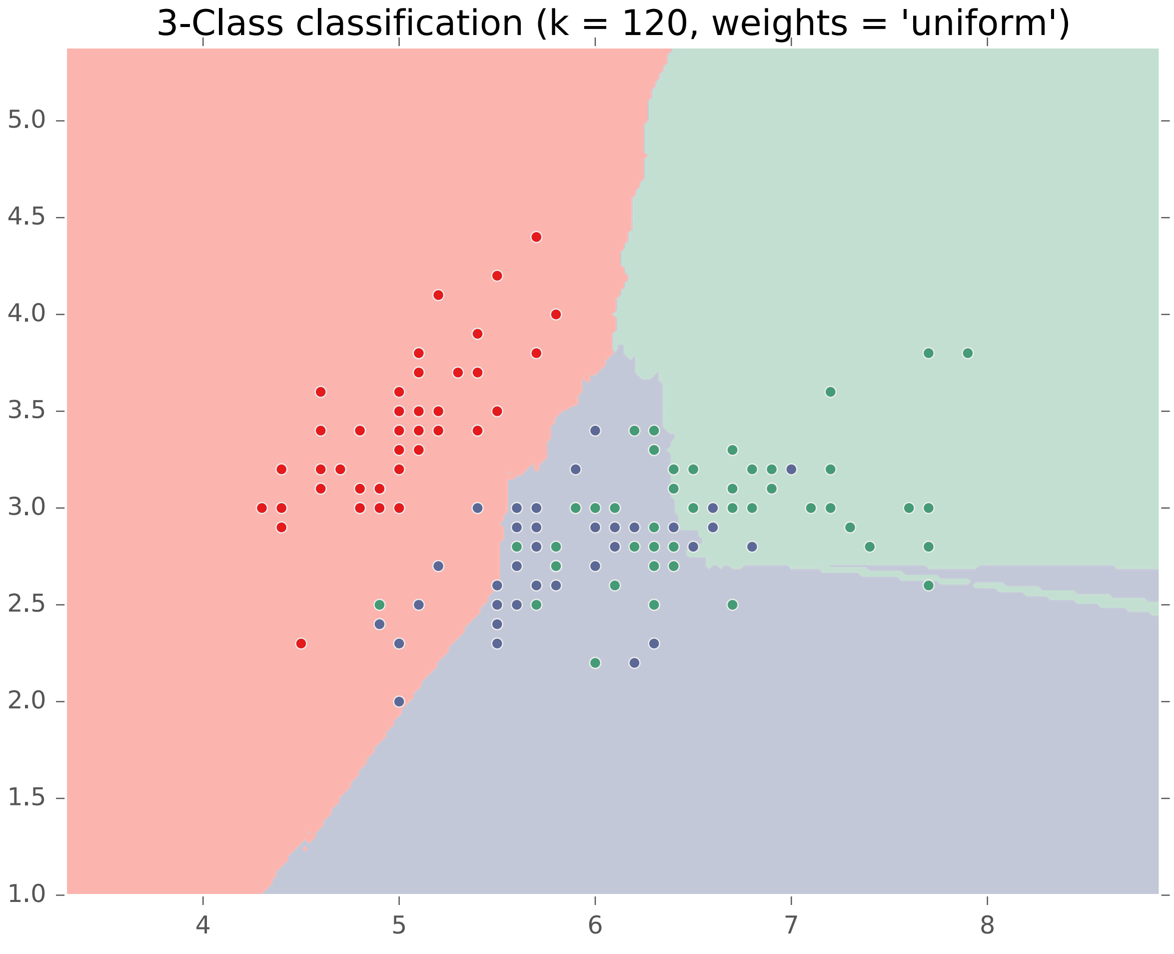
KNN on Fisher Iris Data



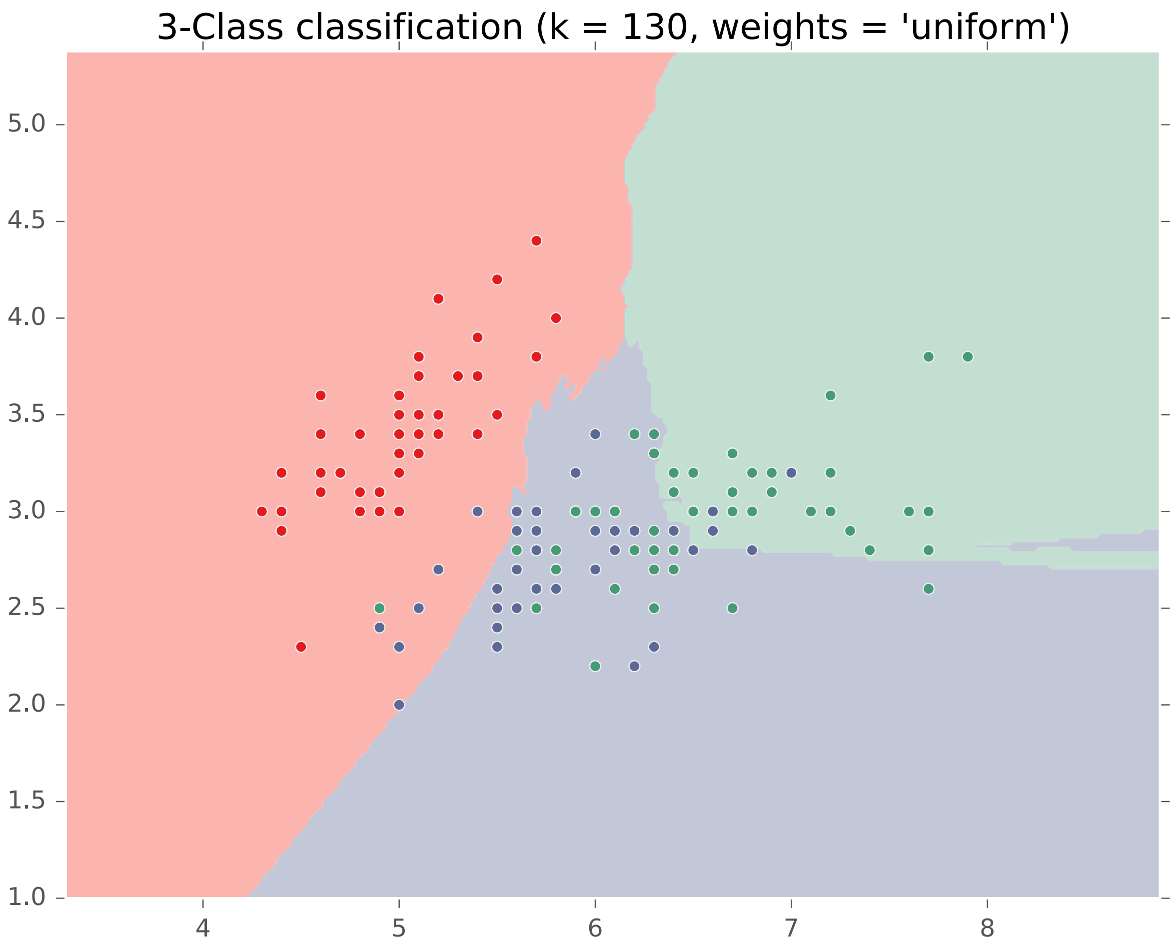
KNN on Fisher Iris Data



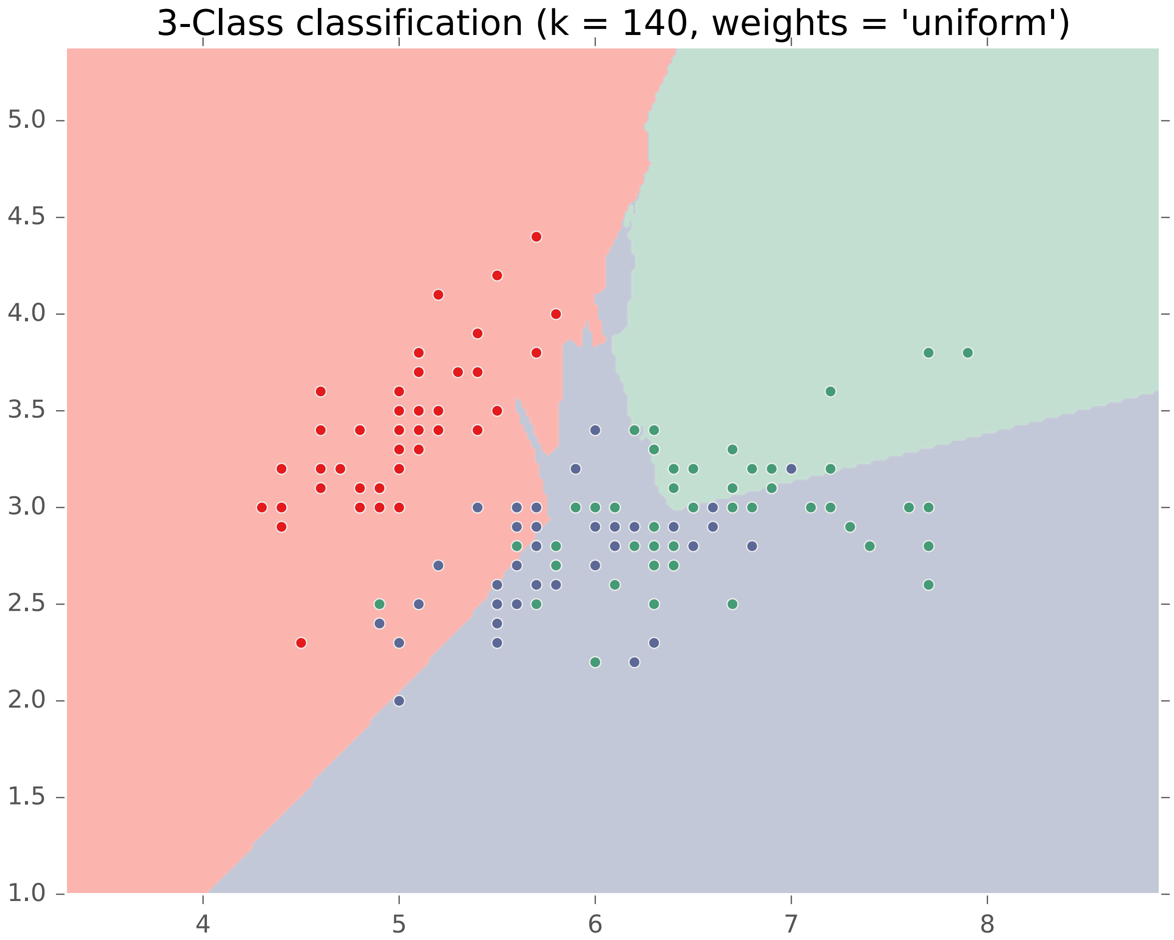
KNN on Fisher Iris Data



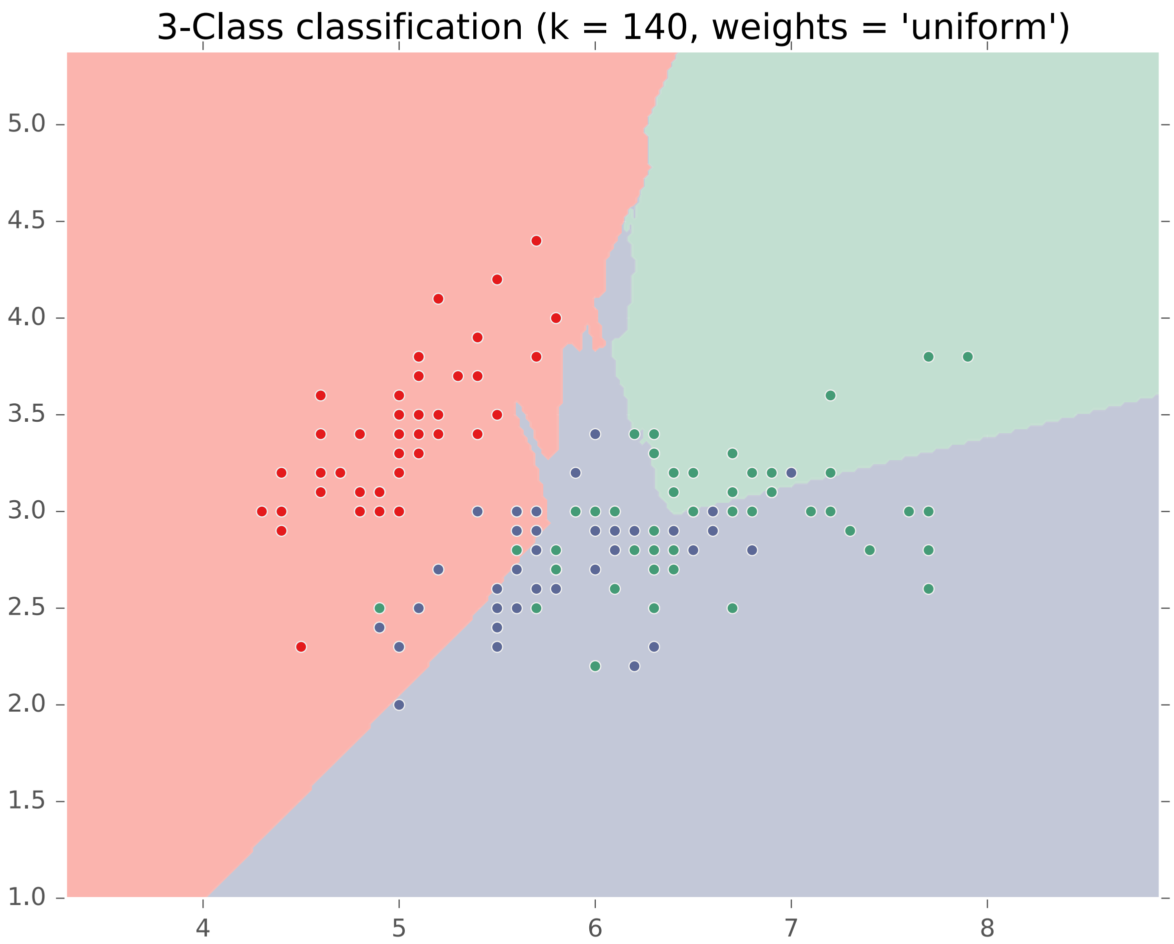
KNN on Fisher Iris Data



KNN on Fisher Iris Data

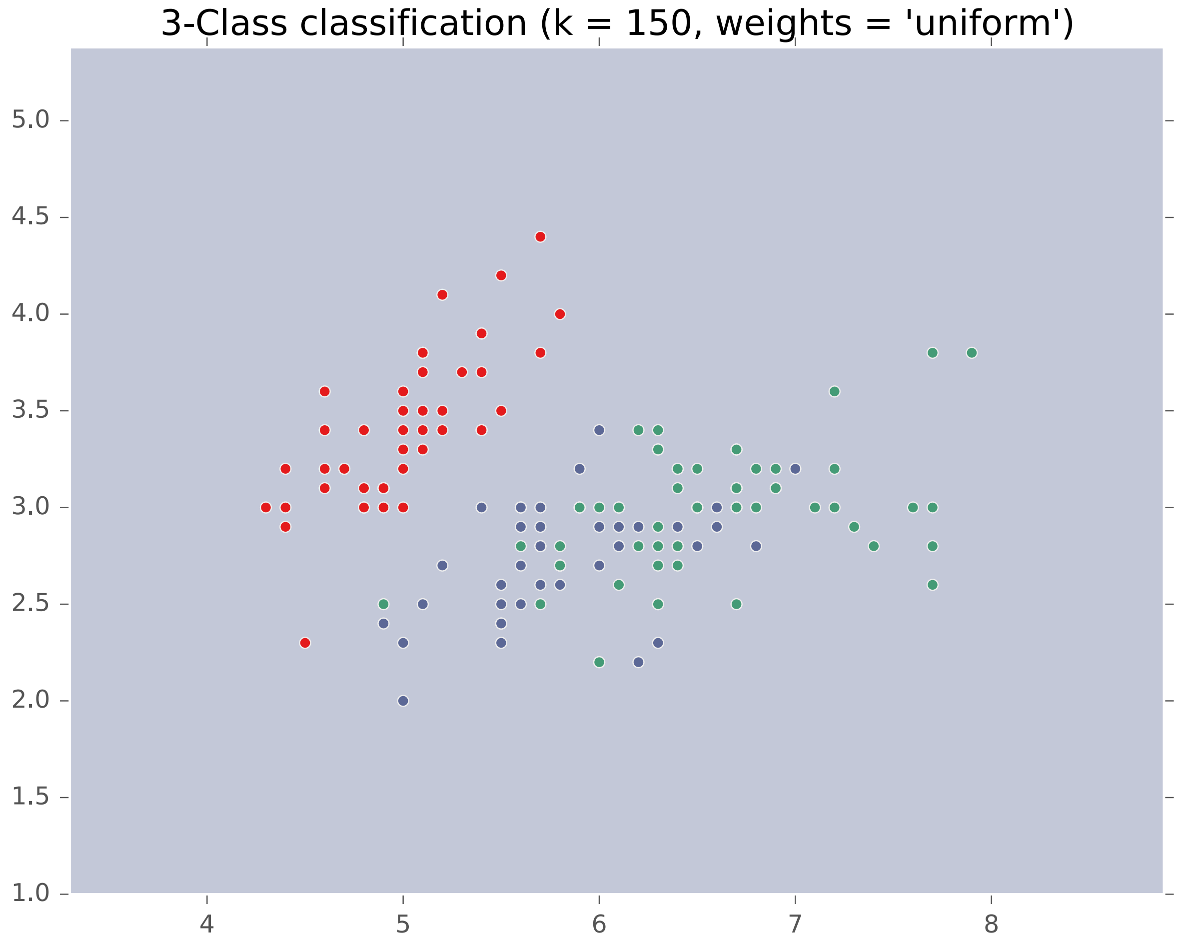


KNN on Fisher Iris Data



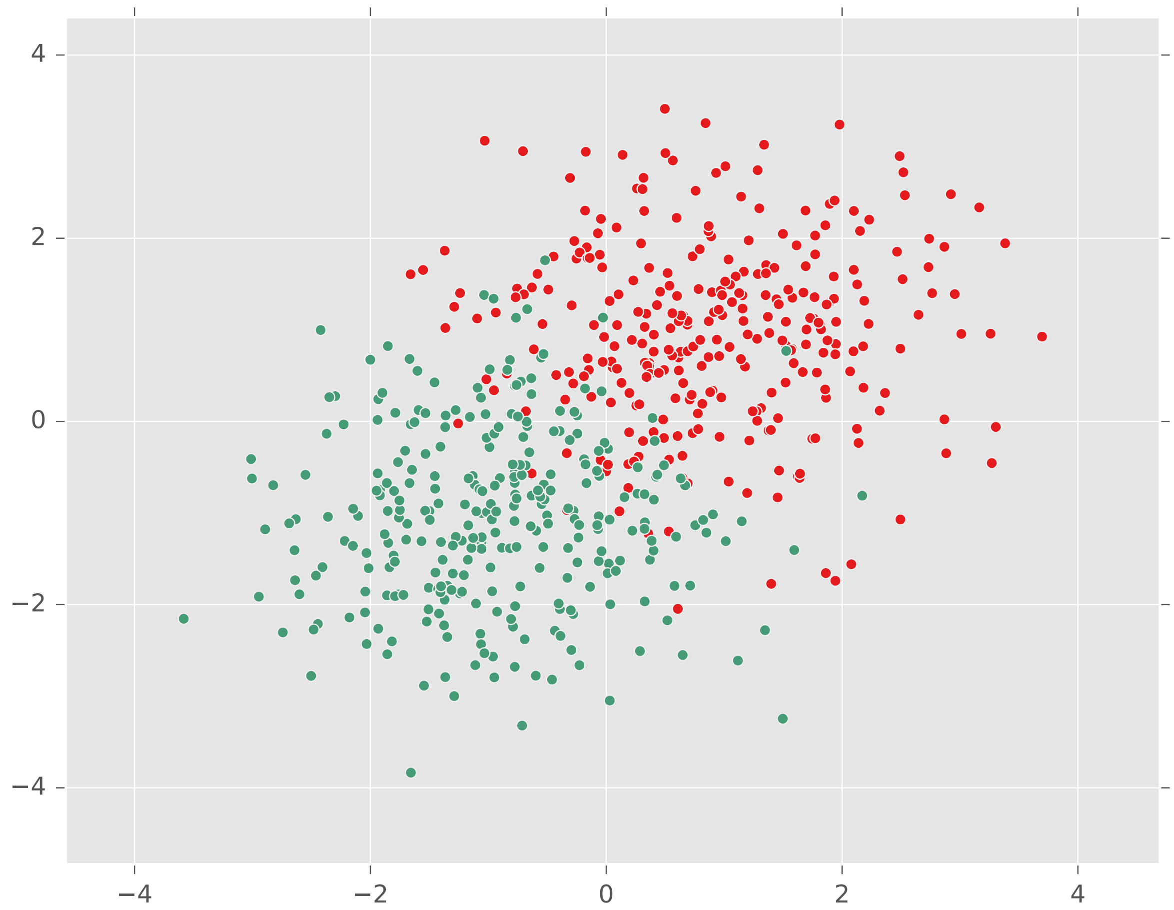
KNN on Fisher Iris Data

Special Case: Majority Vote

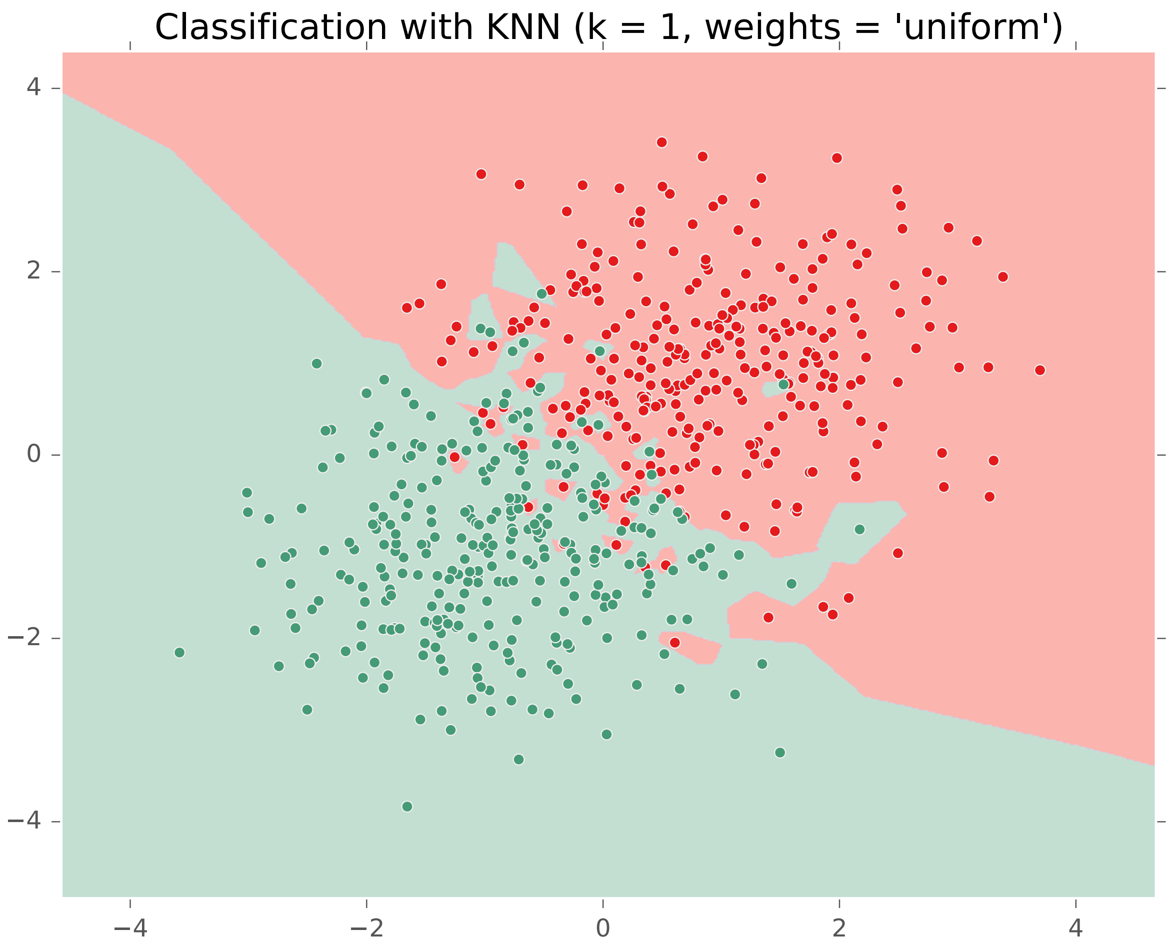


KNN ON GAUSSIAN DATA

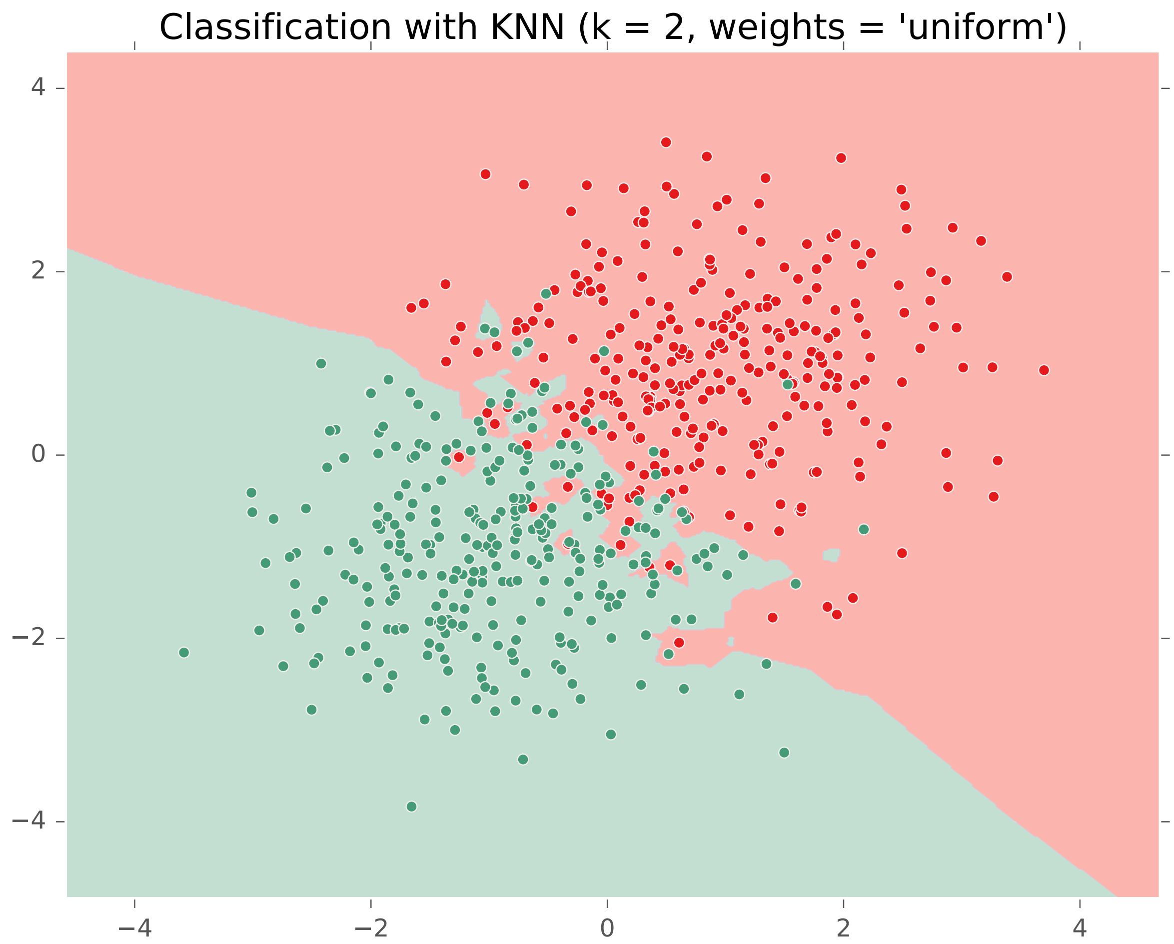
KNN on Gaussian Data



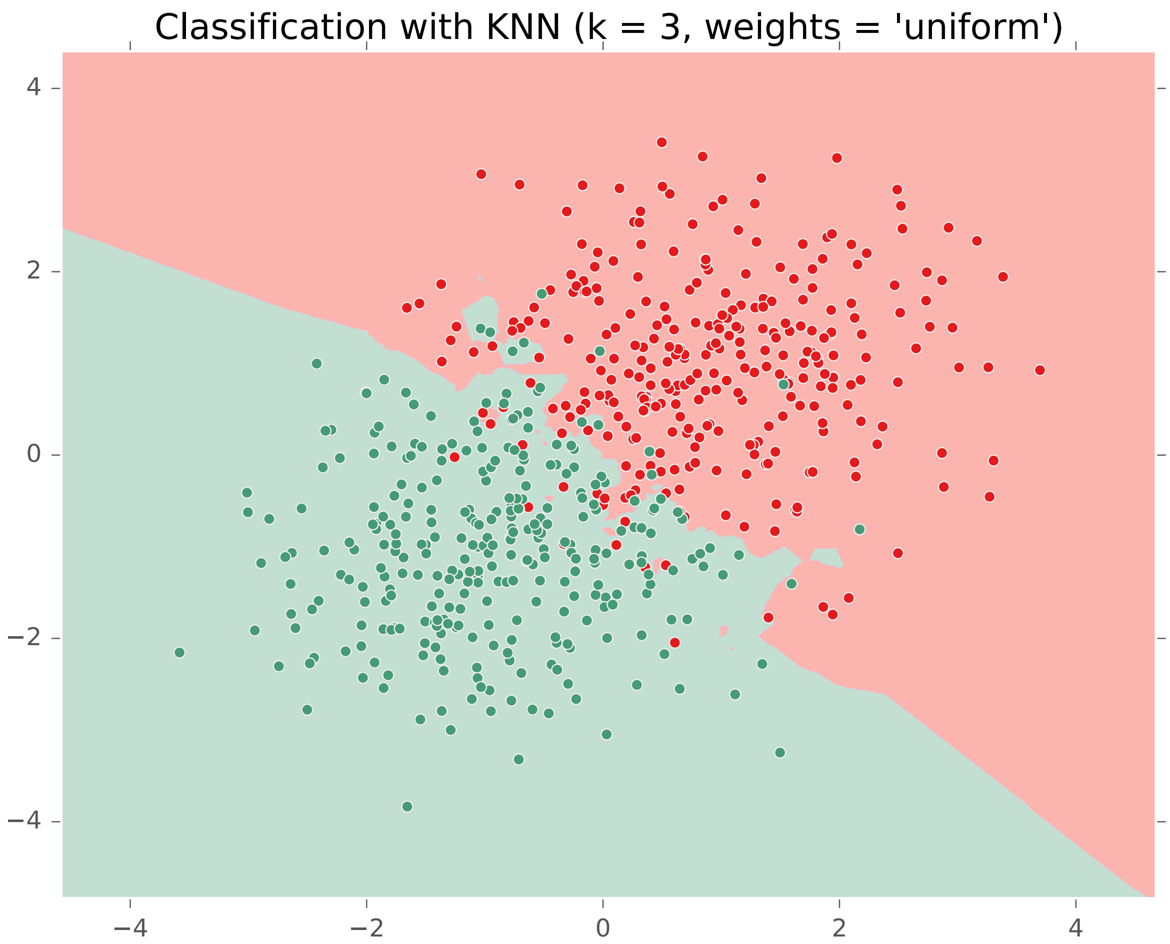
KNN on Gaussian Data



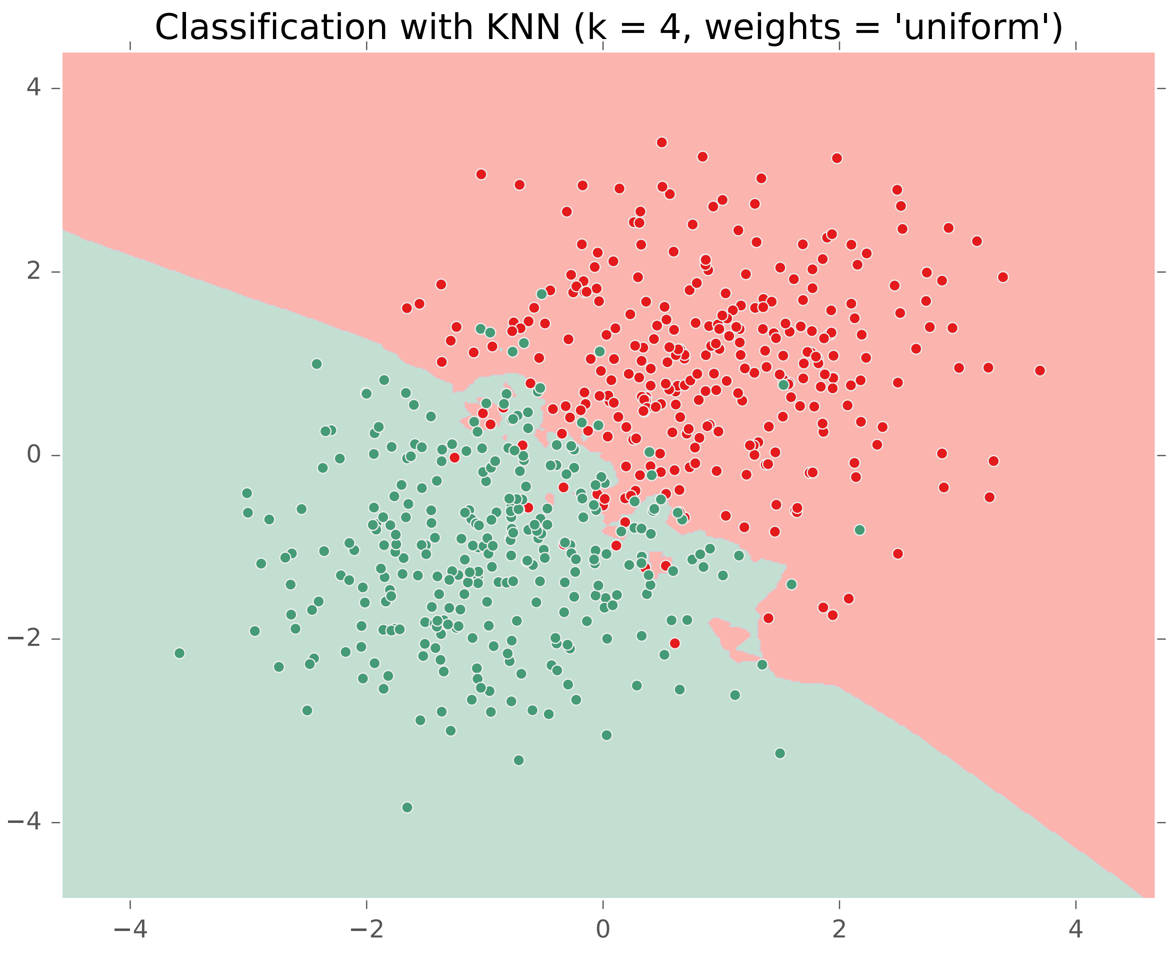
KNN on Gaussian Data



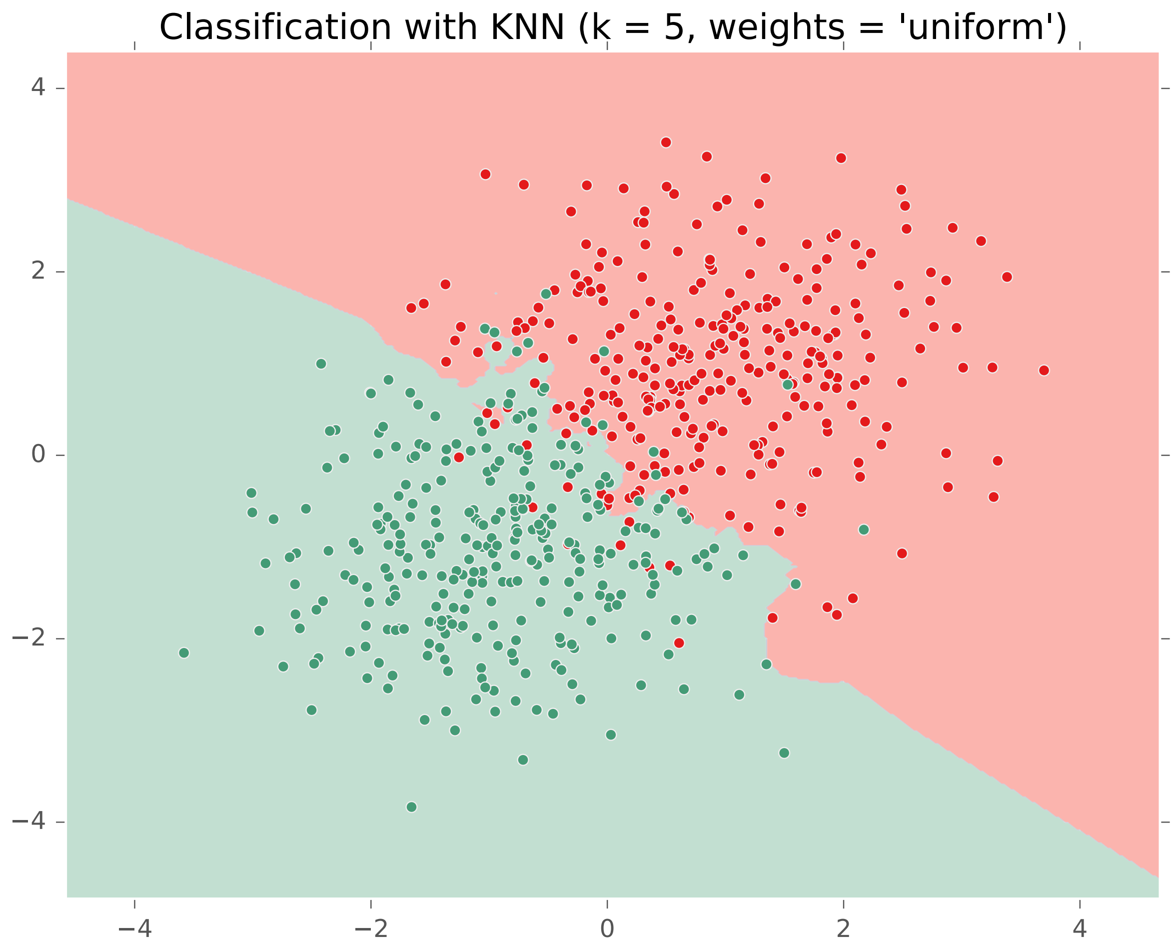
KNN on Gaussian Data



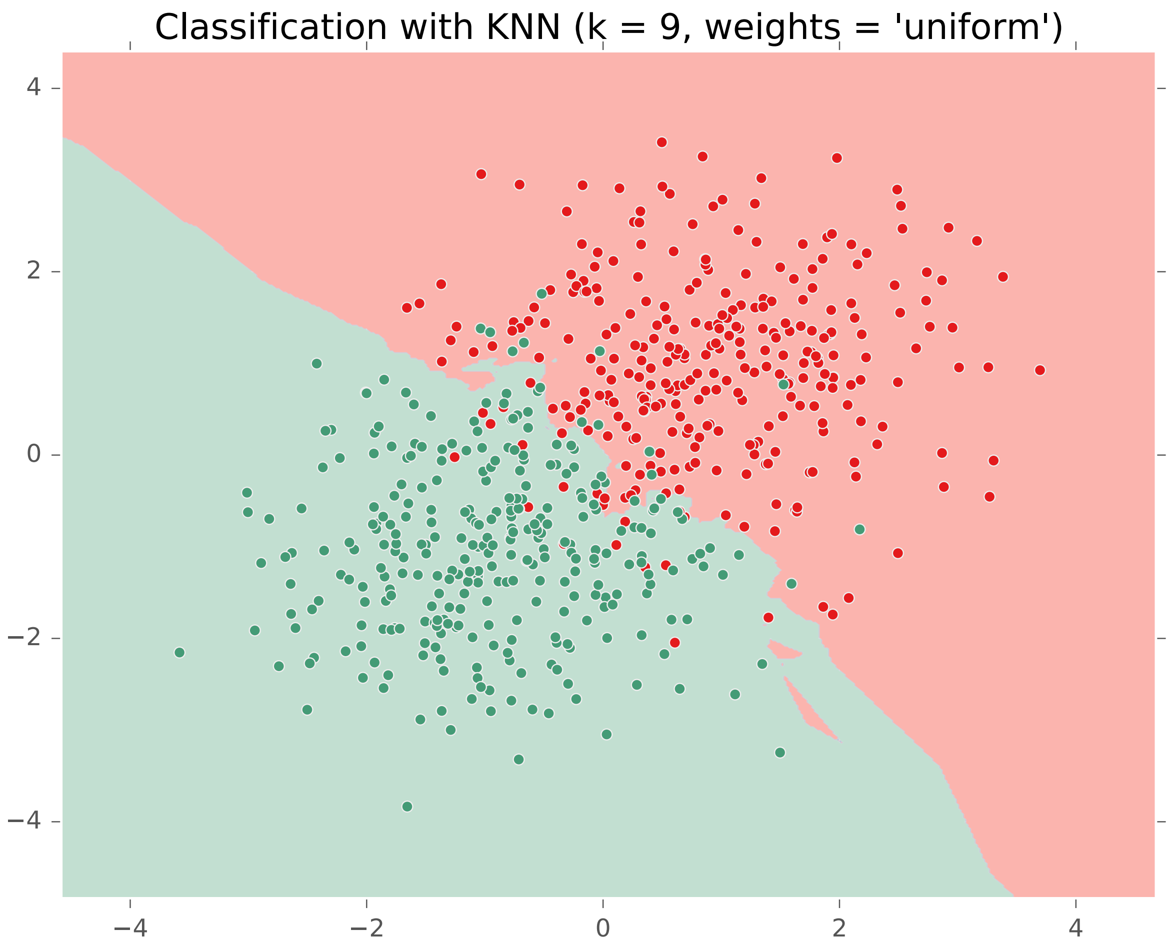
KNN on Gaussian Data



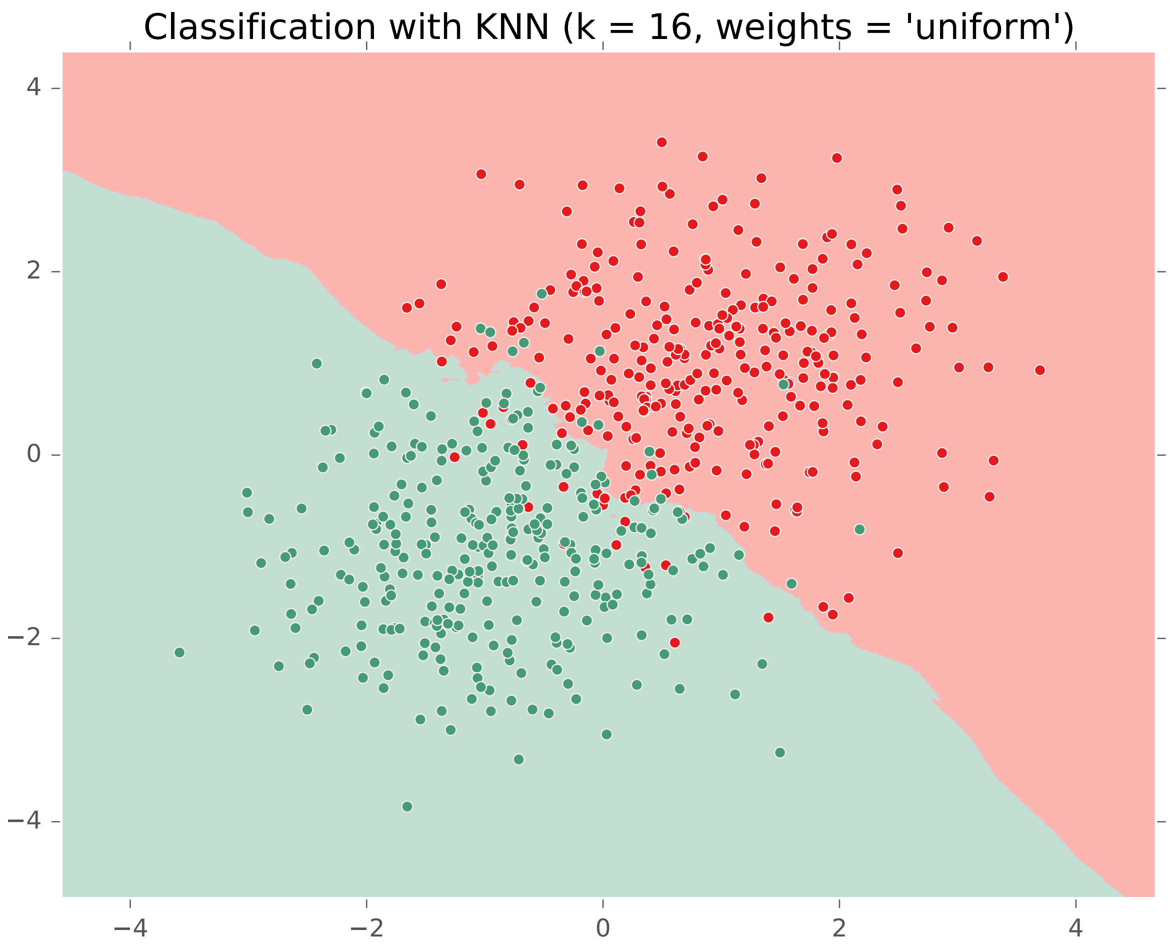
KNN on Gaussian Data



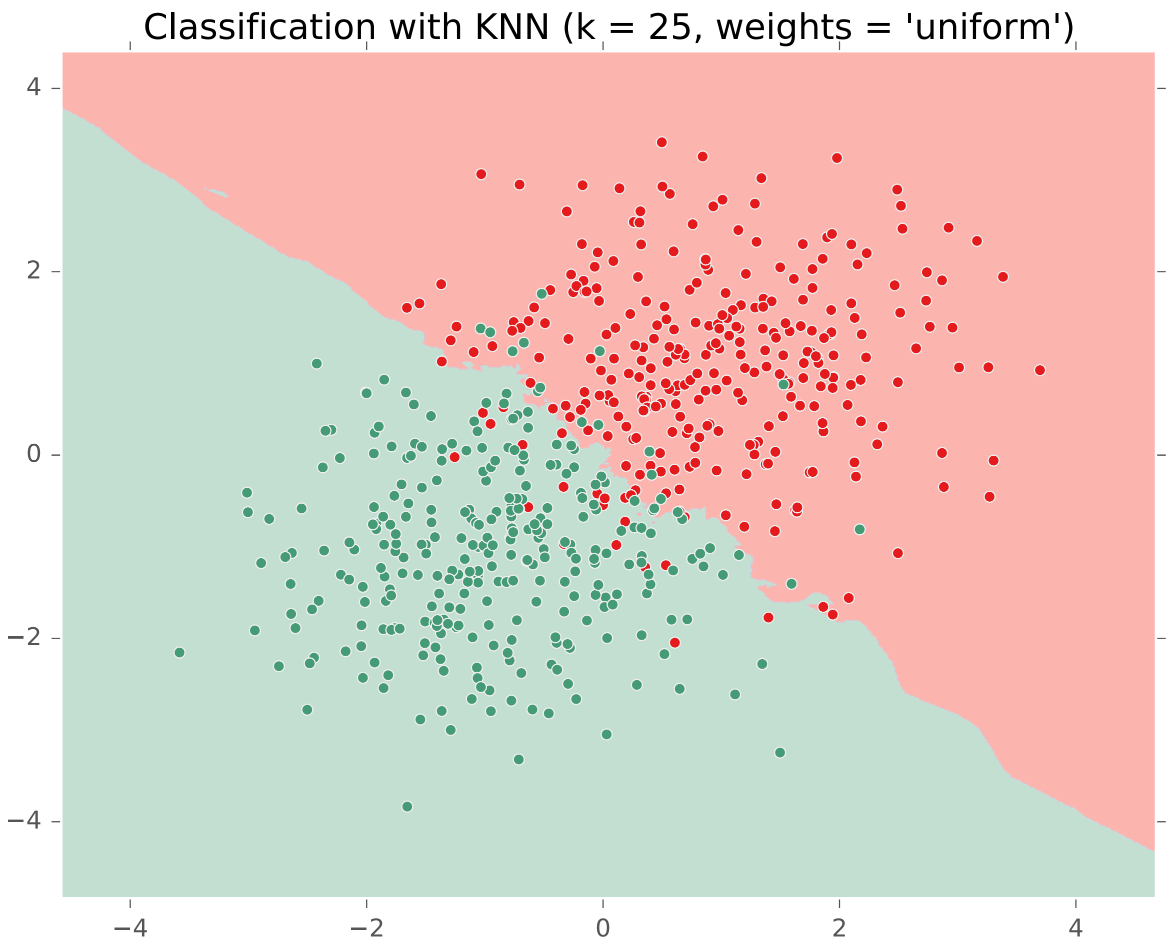
KNN on Gaussian Data



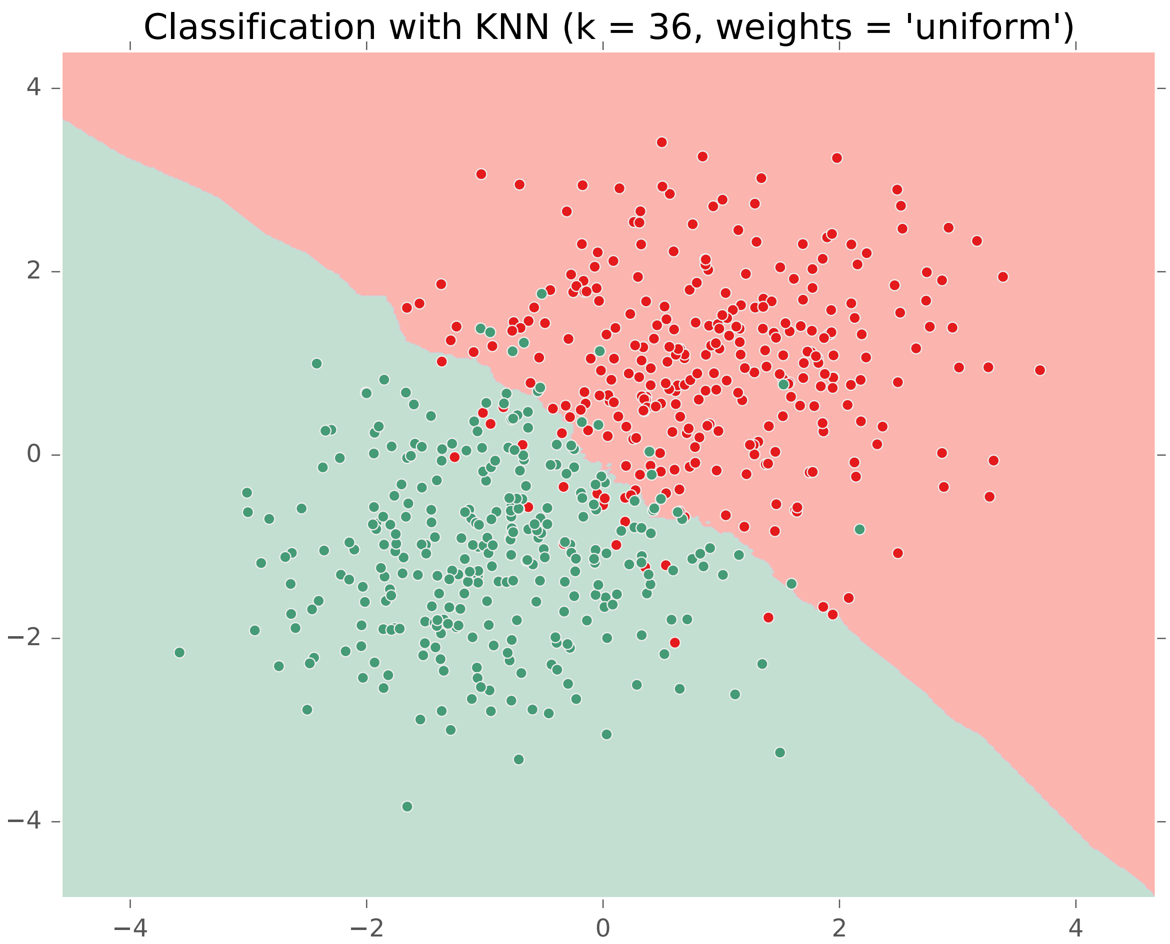
KNN on Gaussian Data



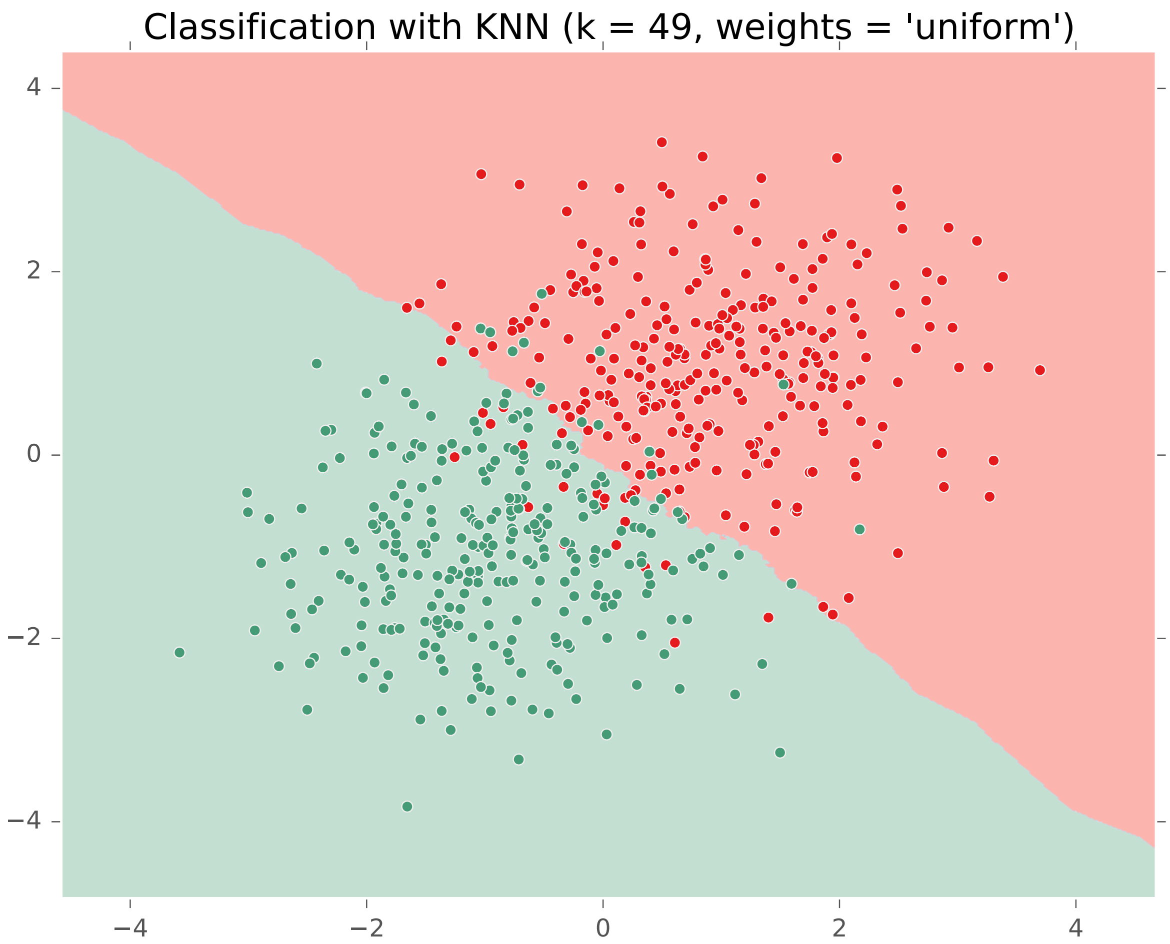
KNN on Gaussian Data



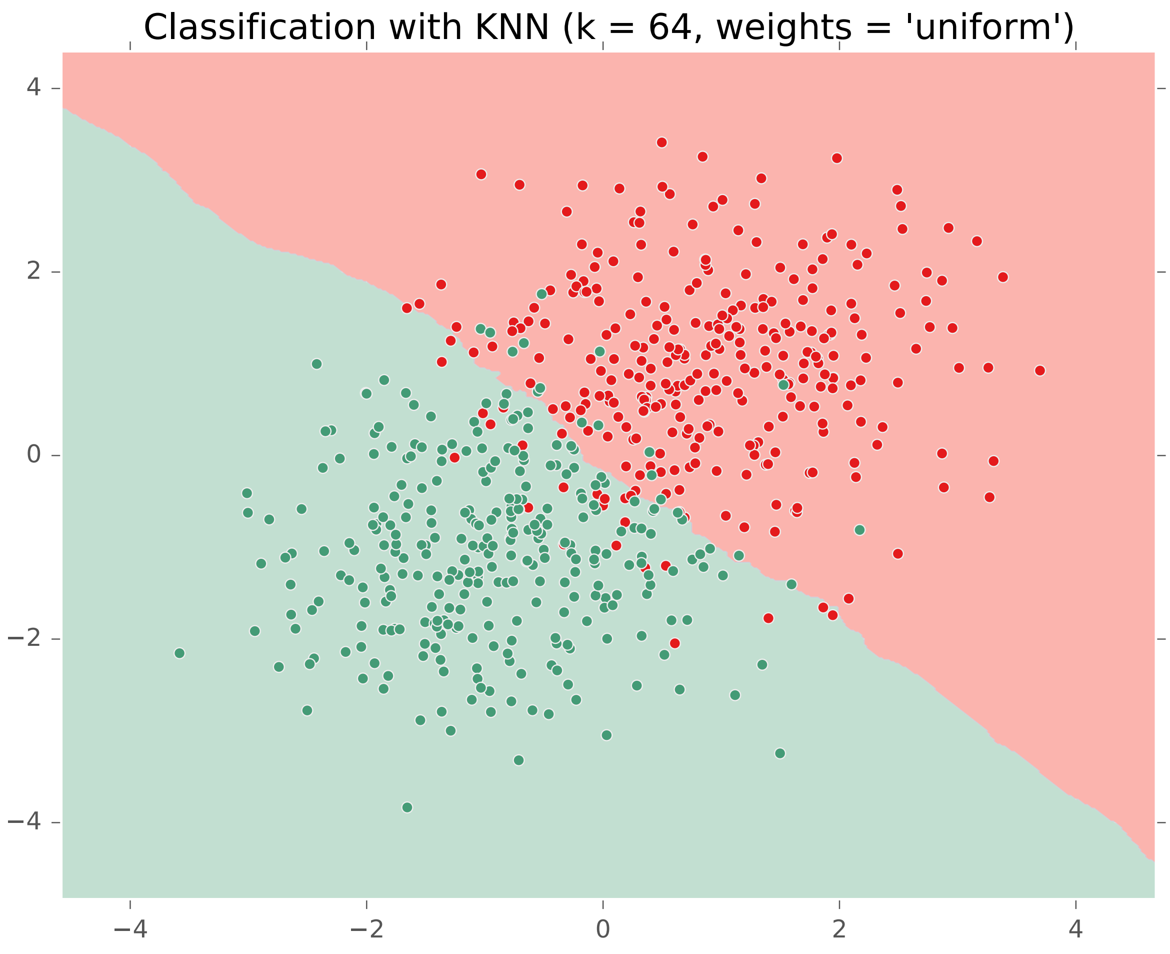
KNN on Gaussian Data



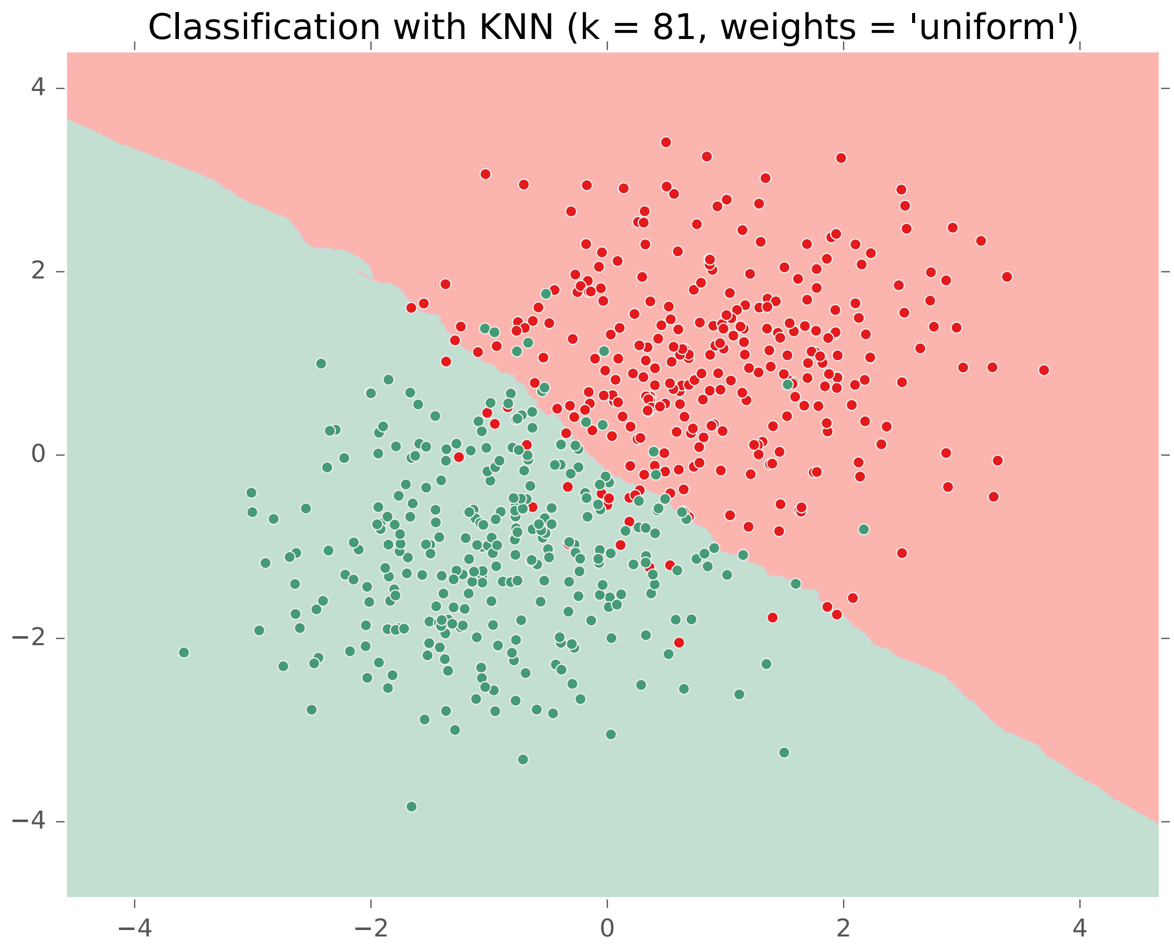
KNN on Gaussian Data



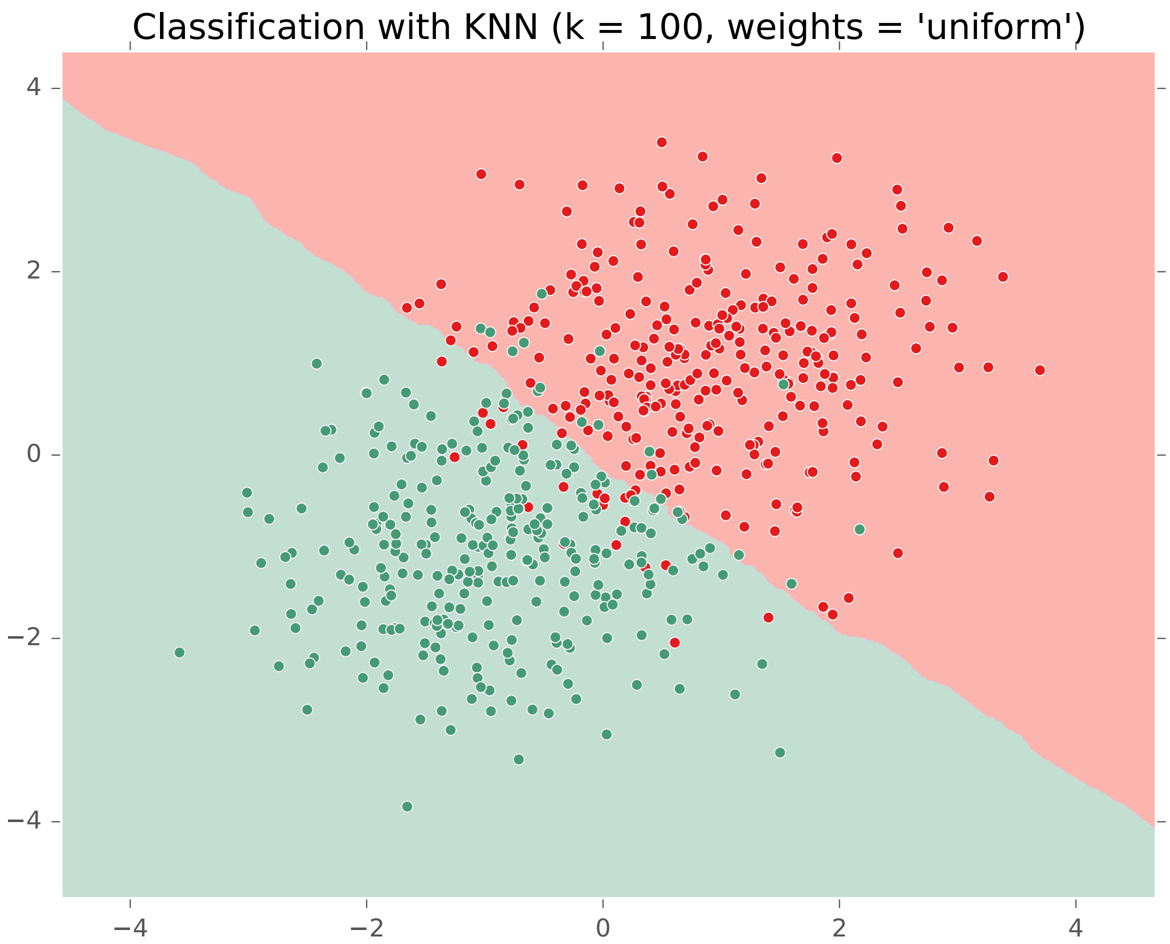
KNN on Gaussian Data



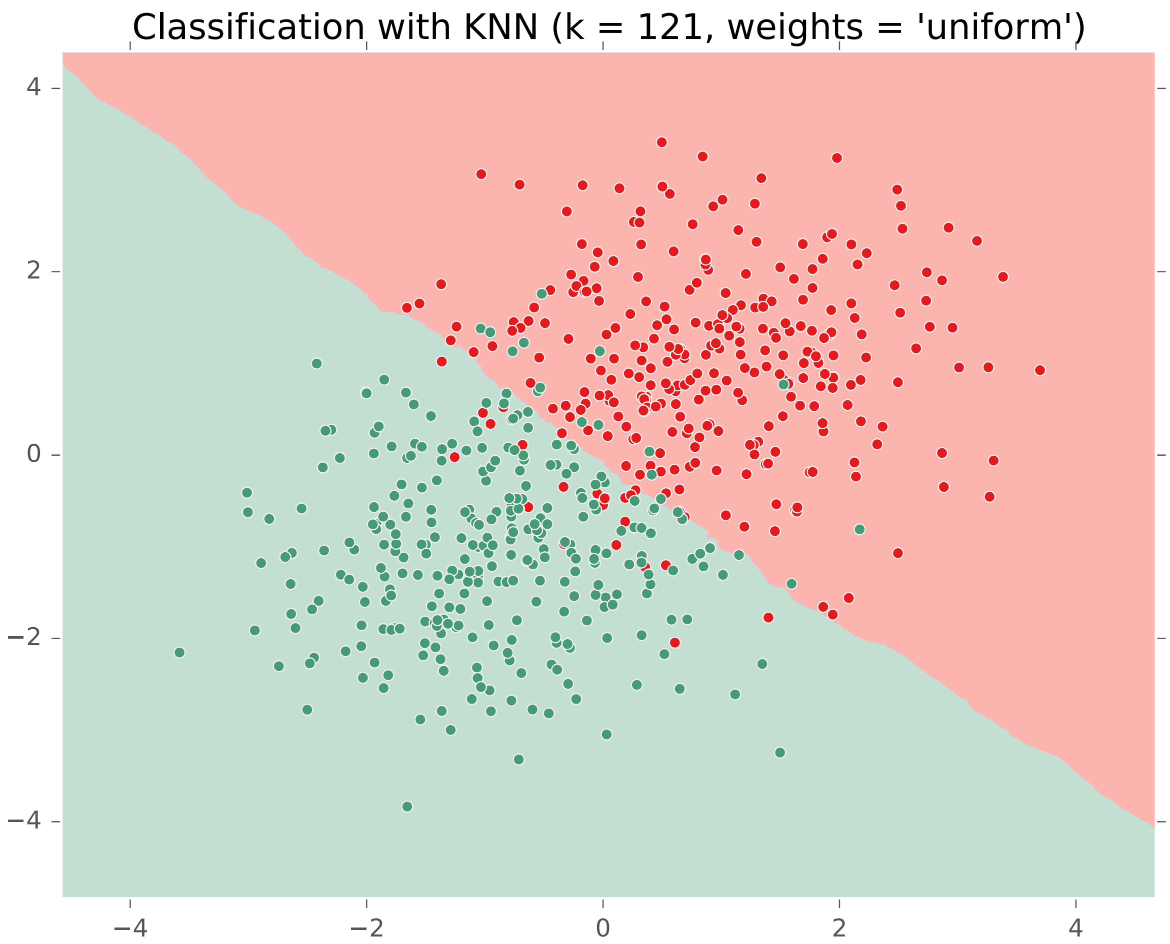
KNN on Gaussian Data



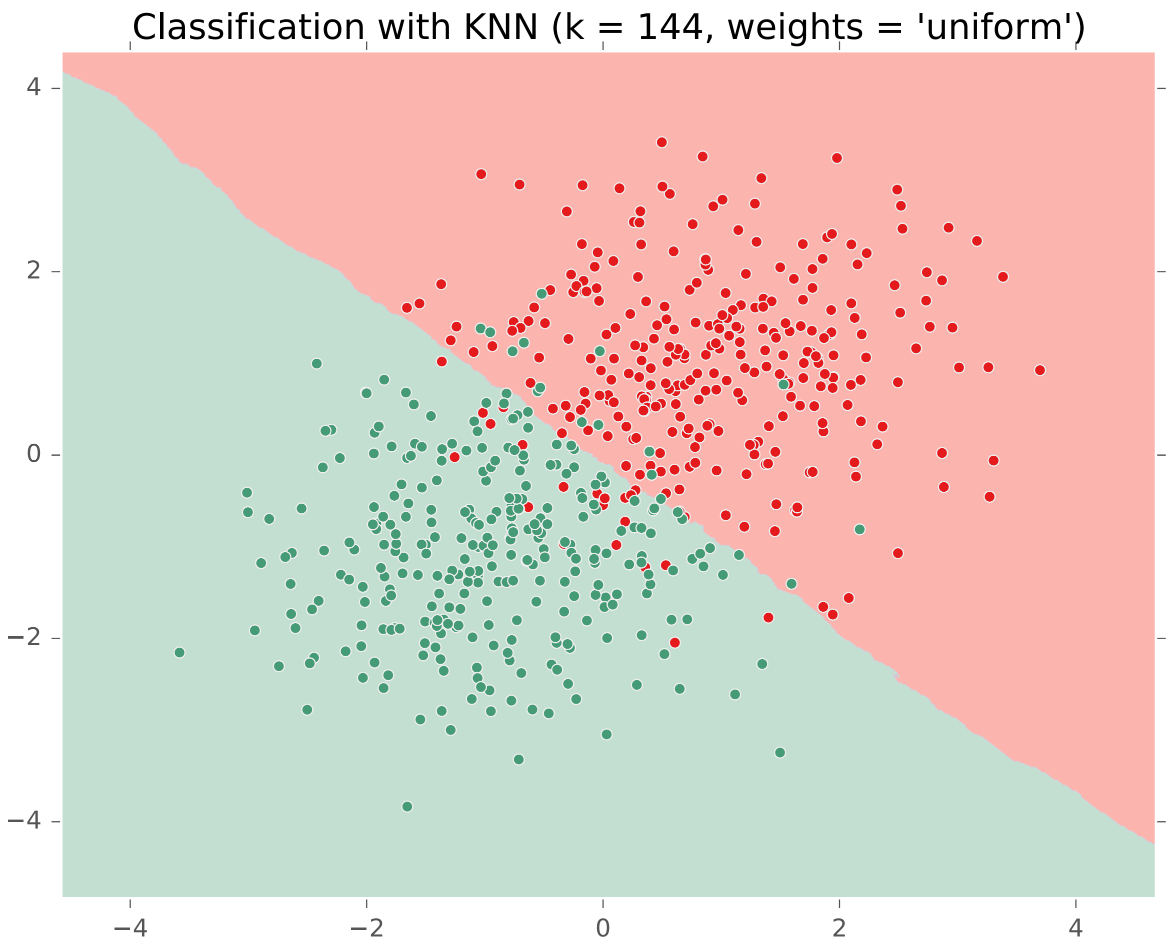
KNN on Gaussian Data



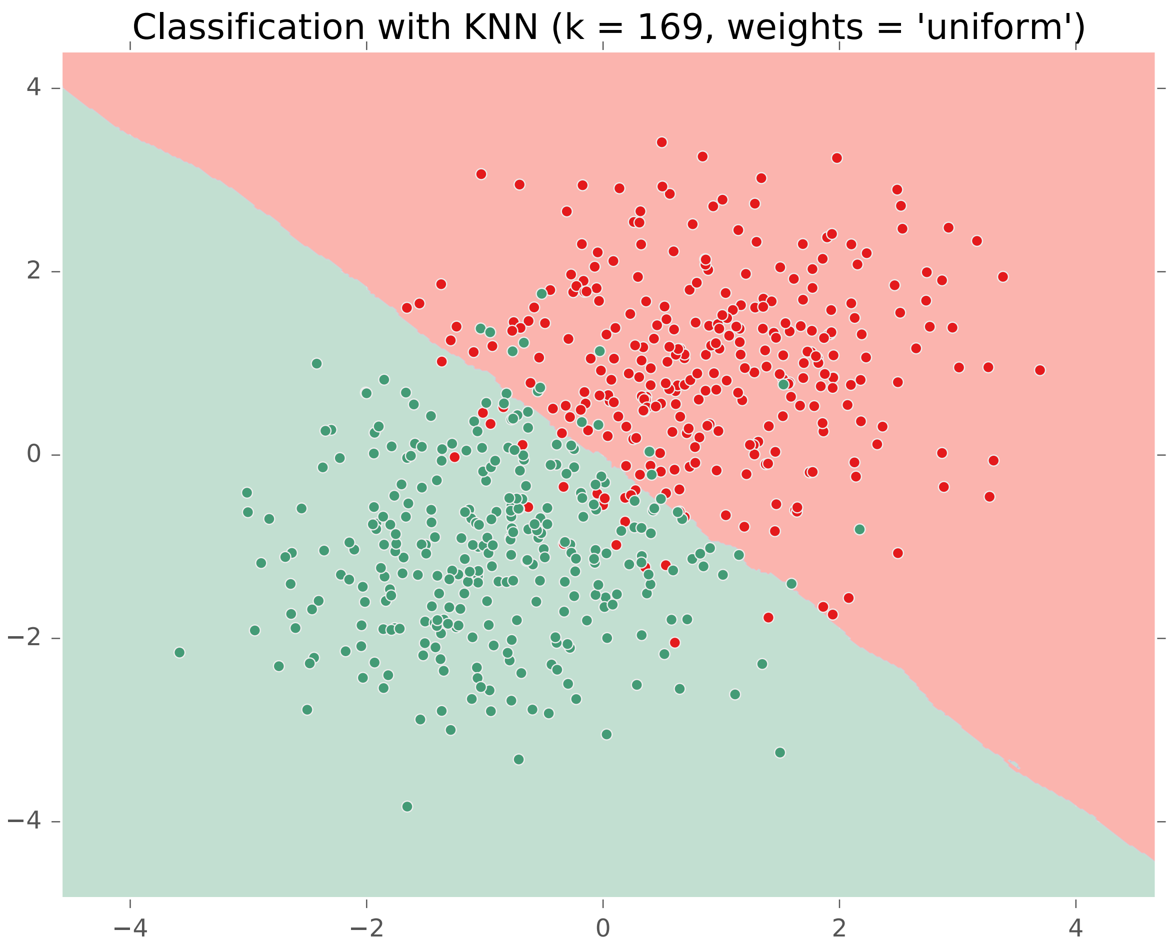
KNN on Gaussian Data



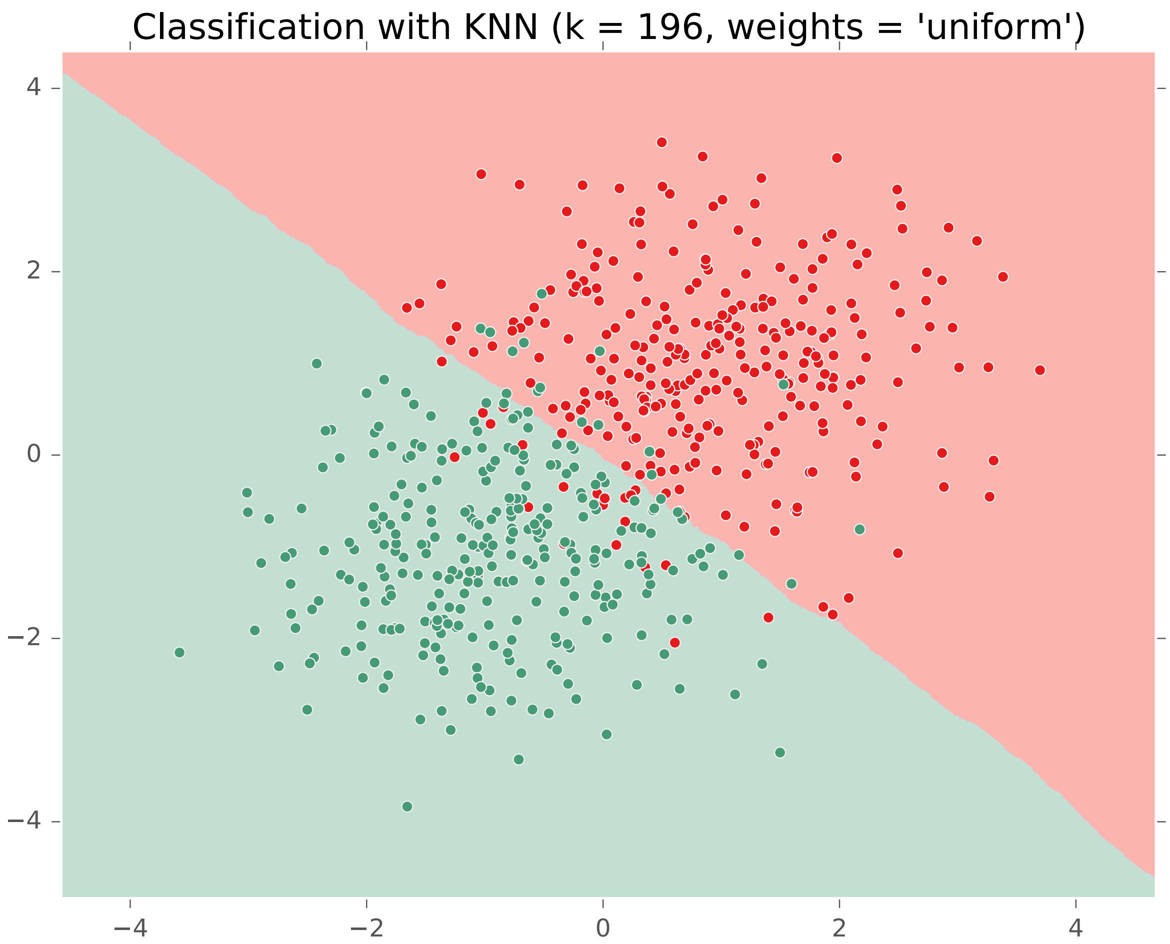
KNN on Gaussian Data



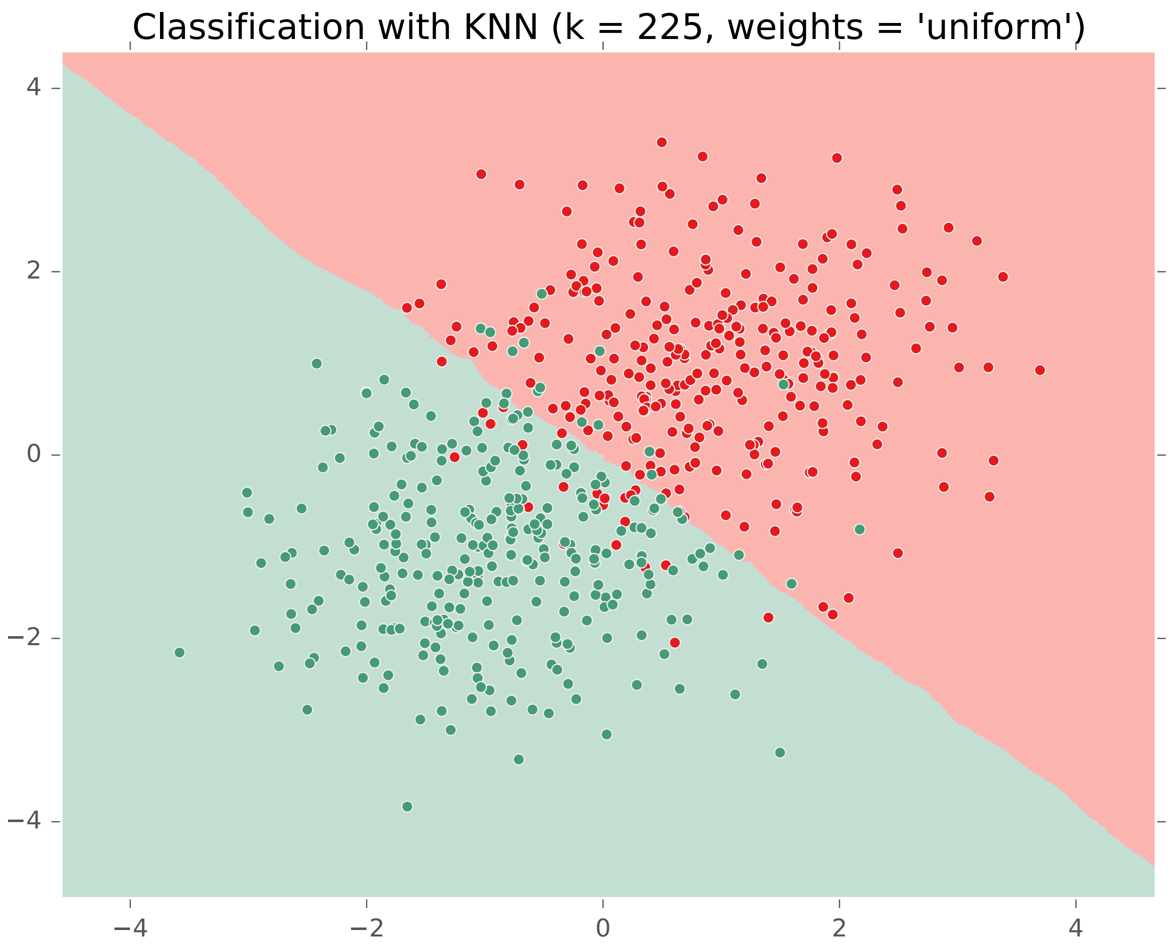
KNN on Gaussian Data



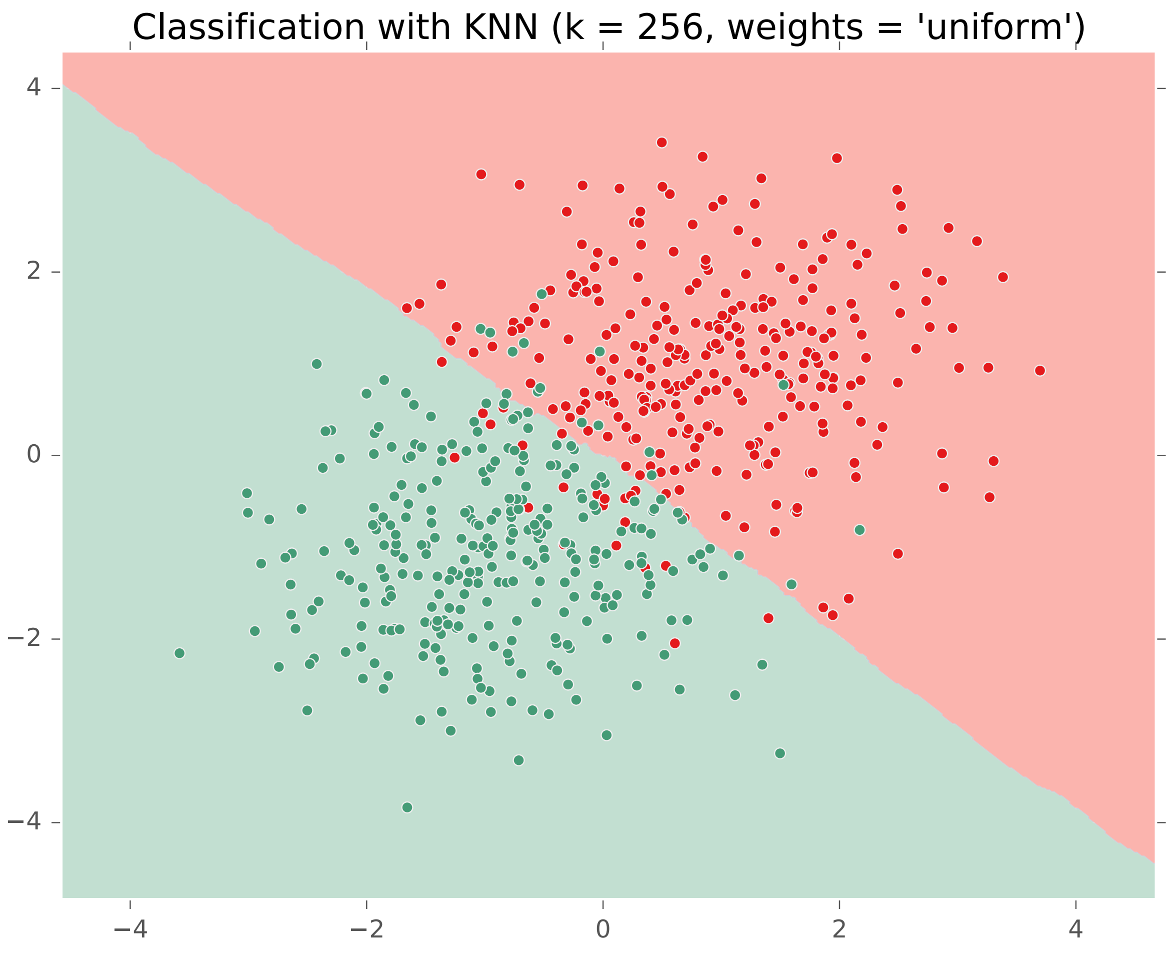
KNN on Gaussian Data



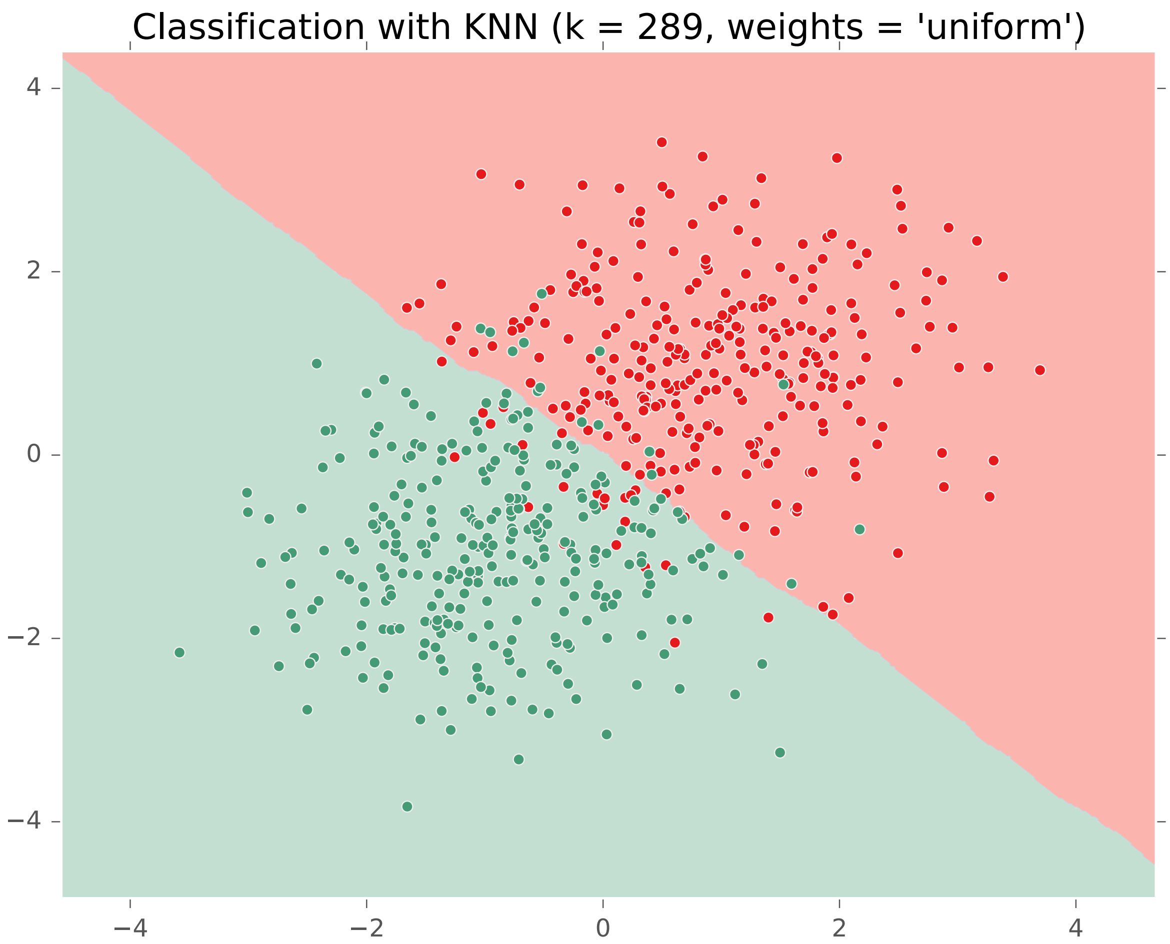
KNN on Gaussian Data



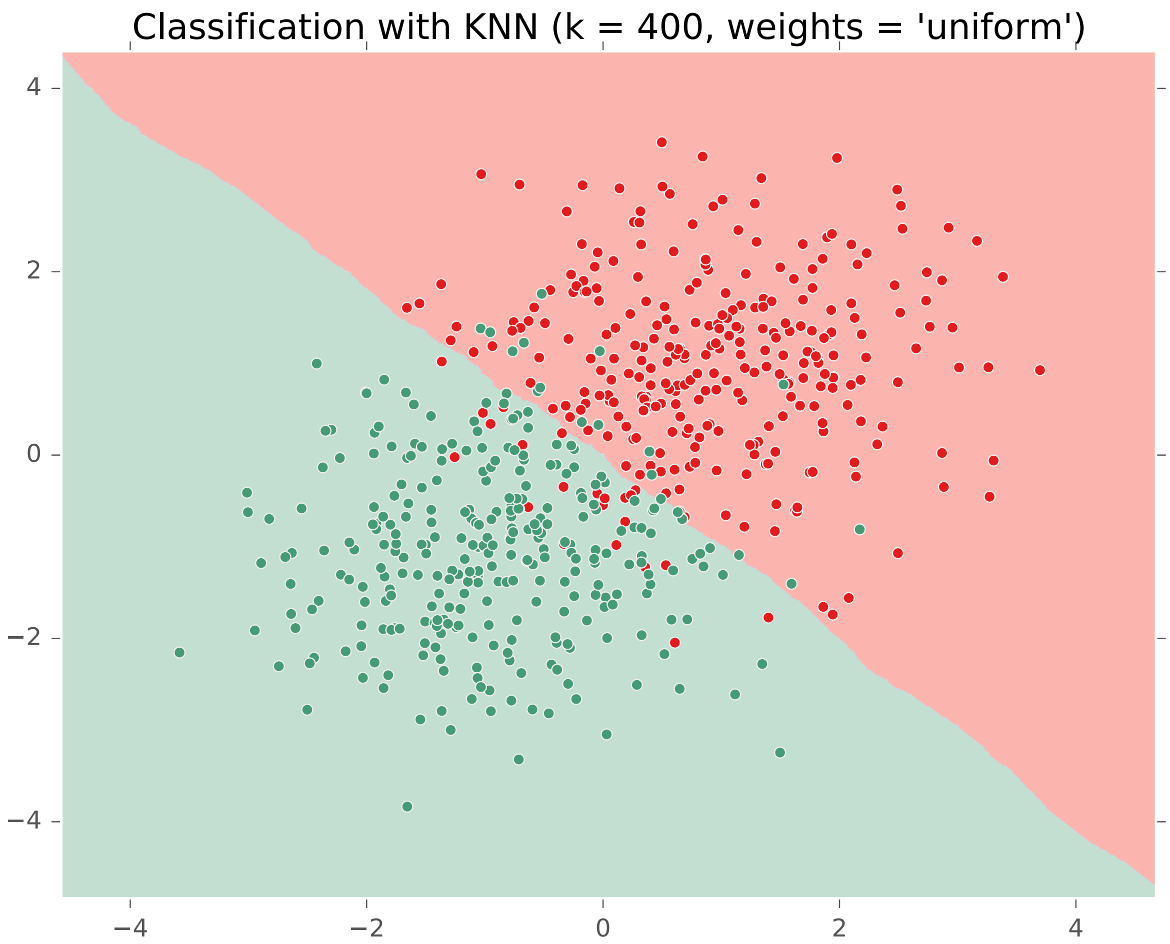
KNN on Gaussian Data



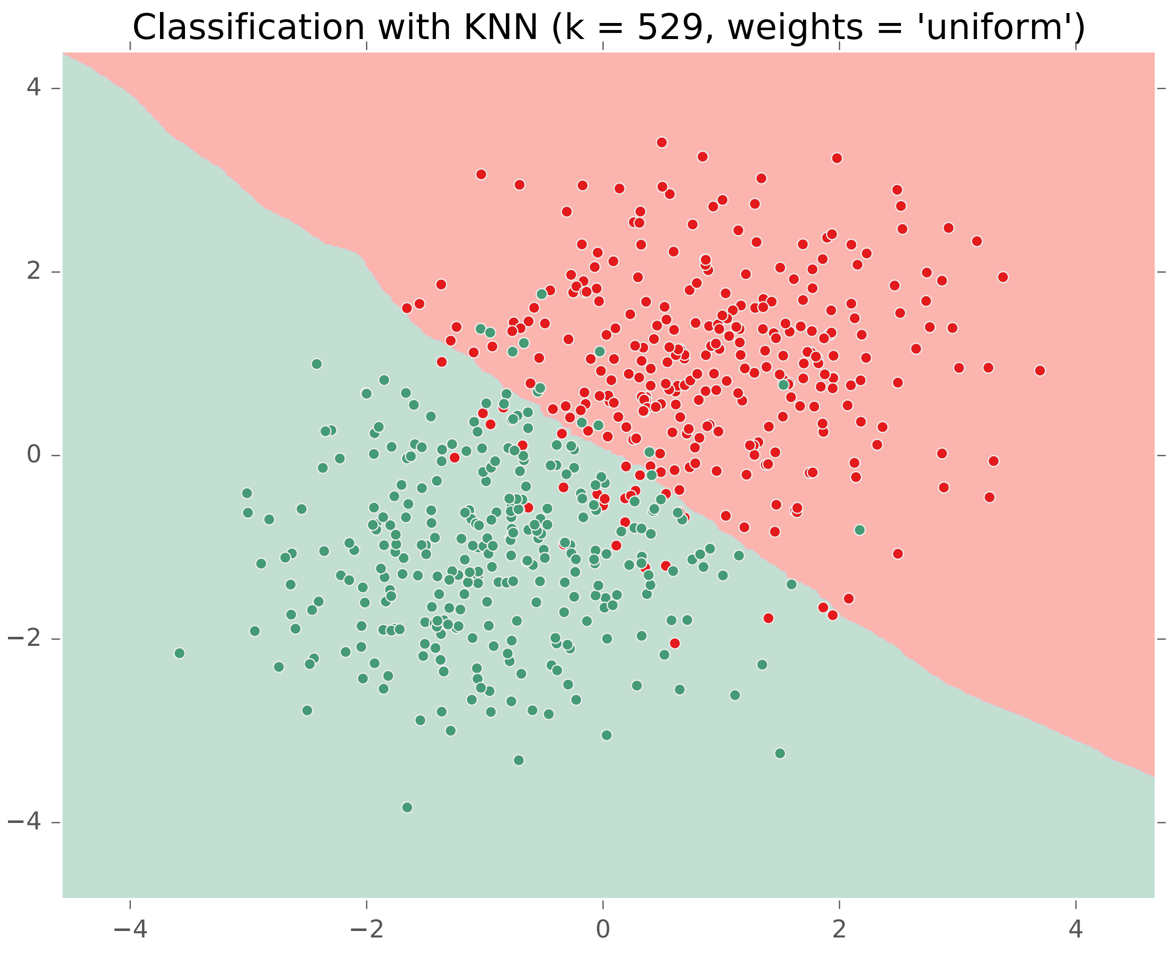
KNN on Gaussian Data



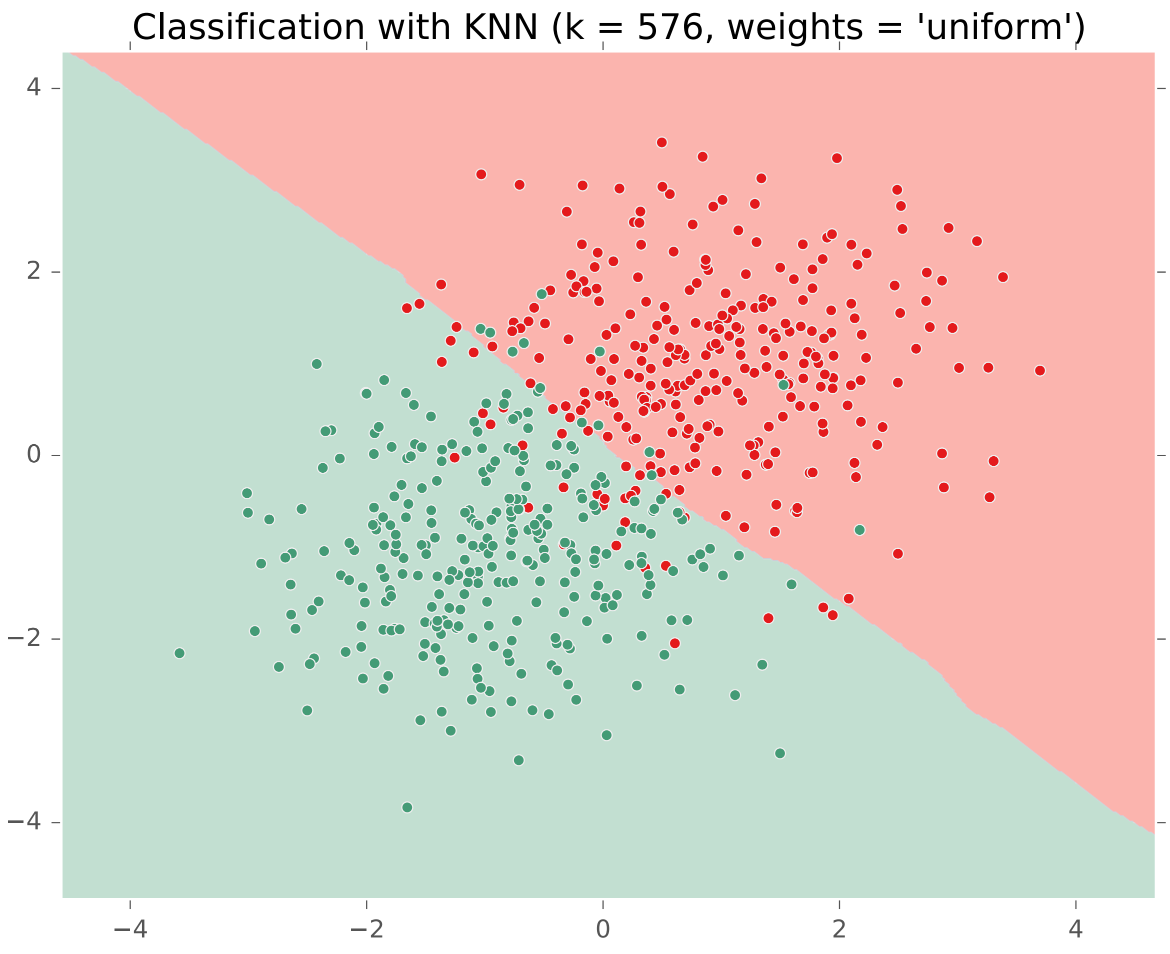
KNN on Gaussian Data



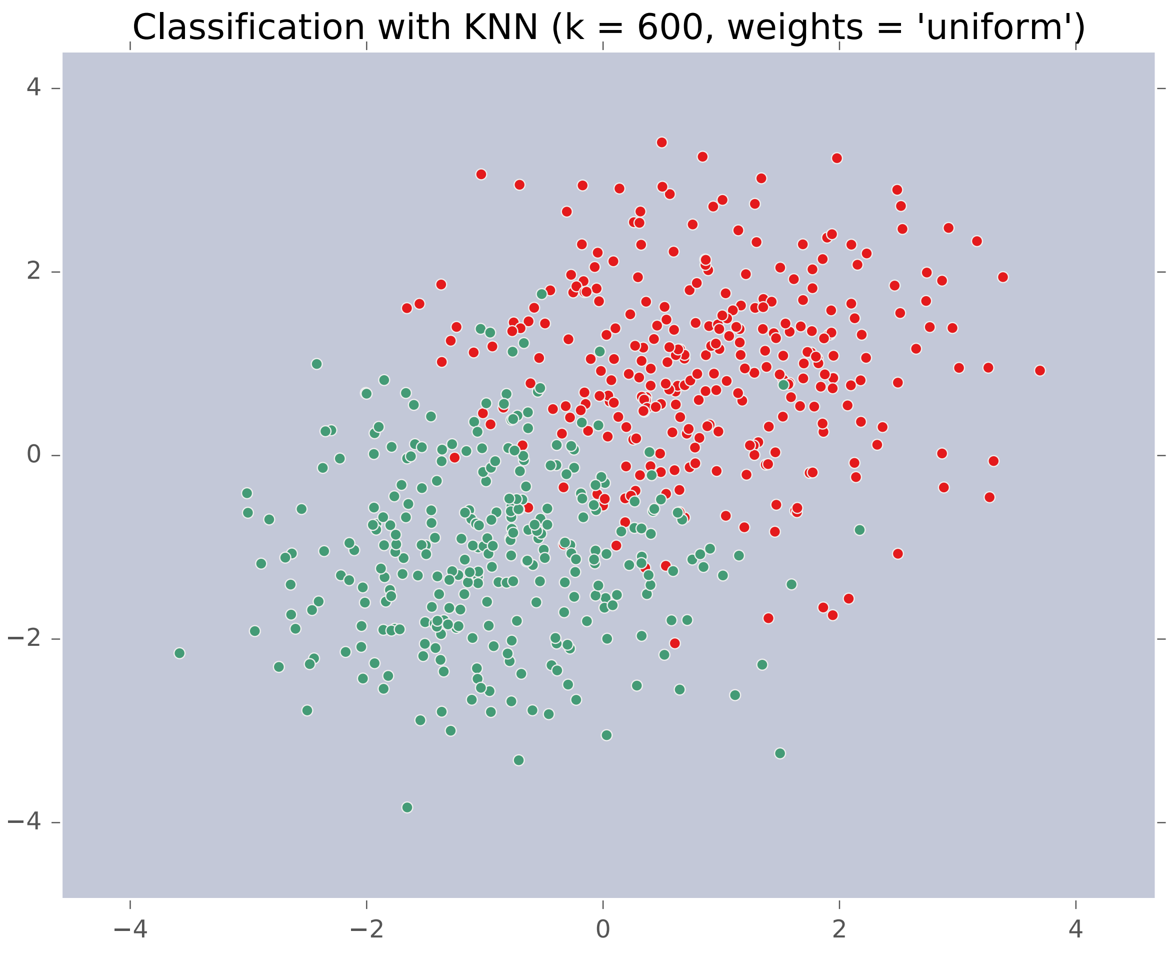
KNN on Gaussian Data



KNN on Gaussian Data



KNN on Gaussian Data

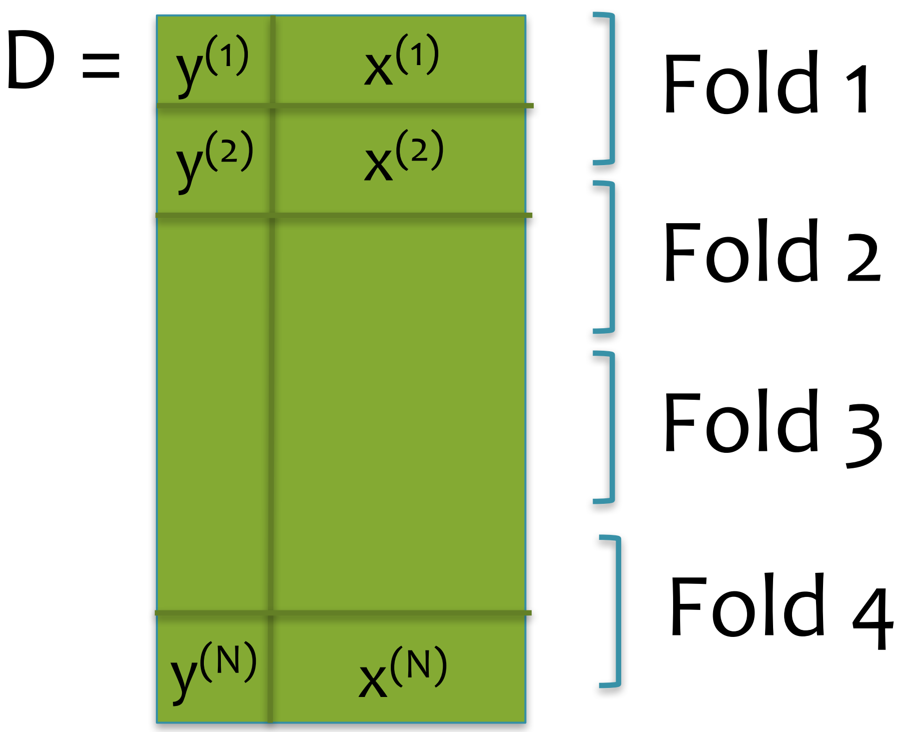


CHOOSING THE NUMBER OF NEIGHBORS

(Name changed from K-Fold Cross-Validation to avoid confusion with KNN)

F-Fold Cross-Validation

Key idea: rather than just a single "validation" set, use many! (Error is more stable. Slower computation.)



Divide data into folds (e.g. 4)

1. Train on folds {1,2,3} and predict on {4}
2. Train on folds {1,2,4} and predict on {3}
3. Train on folds {1,3,4} and predict on {2}
4. Train on folds {2,3,4} and predict on {1}

Concatenate all the predictions and evaluate error

Math as Code

How to implement?

$$y^{max} = \underset{y \in Y}{\operatorname{argmax}} f(y)$$

It depends on how large the set Y is!

If it's a small enumerable set $Y = \{1,2,\ldots,77\}$, then:

```
ymax = -inf
for y in {1,2,…77}:
    if f(y) > ymax:
        ymax = y
return ymax
```

Math as Code

How to implement?

$$v^{max} = \max_{y \in Y} f(y)$$

It depends on how large the set Y is!

If it's a small enumerable set Y = {1,2,…,77}, then:

```
vmax = -inf
for y in {1,2,…77}:
    if f(y) > vmax:
        vmax = f(y)
return vmax
```

Function Approximation View of ML

Whiteboard

Beyond the Scope of This Lecture

- k-Nearest Neighbors (KNN) for **Regression**
- **Distance-weighted** KNN
- Cover & Hart (1967) **Bayes error rate bound**
- KNN for Facial Recognition (see **Eigenfaces** in PCA lecture)

Takeaways

- **k-Nearest Neighbors**
  - Requires careful choice of k (# of neighbors)
  - Experimental design can be just as important as the learning algorithm itself
- **Function Approximation View**
  - Assumption: inputs are sampled from some unknown distributions
  - Assumption: outputs come from a fixed unknown function (e.g. human annotator)
  - Goal: Learn a hypothesis which closely approximates that function