10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

k-Nearest Neighbors
+
Model Selection

Matt Gormley
Lecture 5
Jan. 30, 2019

Reminders

- **Homework 2: Decision Trees**
  - Out: Wed, Jan 23
  - Due: Wed, Feb 6 at 11:59pm
- **Today's Poll:**
  - http://p5.mlcourse.org

K-NEAREST NEIGHBORS

k-Nearest Neighbors

Chalkboard:

- Nearest Neighbor classifier
- KNN for binary classification

KNN: Remarks

Distance Functions:

- KNN requires a **distance function**

$$g : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}$$

- The most common choice is **Euclidean distance**

$$g(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{m=1}^{M} (u_m - v_m)^2}$$

- But other choices are just fine (e.g. **Manhattan distance**)

$$g(\mathbf{u}, \mathbf{v}) = \sum_{m=1}^{M} |u_m - v_m|$$

KNN: Remarks

In-Class Exercises

1. How can we handle ties for even values of k?

2. What is the inductive bias of KNN?

Answer(s) Here:

KNN: Remarks

Computational Efficiency:

- Suppose we have N training examples, and each one has M features
- Computational complexity for the special case where k=1:

Task	Naive	k-d Tree
Train	O(1)	~ O(M N log N)
Predict (one test example)	O(MN)	~ O(2^M log N) on average

Problem: Very fast for small M, but very slow for large M

In practice: use stochastic approximations (very fast, and empirically often as good)

KNN: Remarks

Theoretical Guarantees:

Cover & Hart (1967)

Let h(x) be a Nearest Neighbor (k=1) binary classifier. As the number of training examples N goes to infinity…

$$\text{error}_{\text{true}}(h) < 2 \times \text{Bayes Error Rate}$$

"In this sense, it may be said that half the classification information in an infinite sample set is contained in the nearest neighbor."

very informally, Bayes Error Rate can be thought of as: *'the best you could possibly do'*

Decision Boundary Example

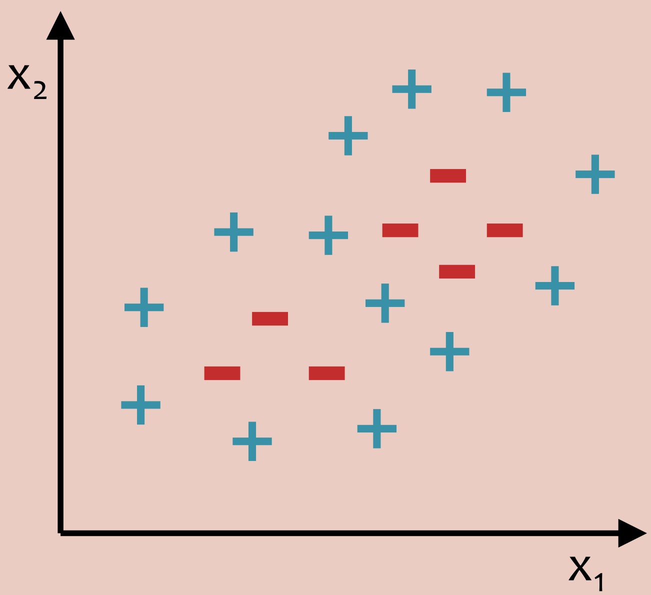
Dataset: Outputs {+,-}; Features x_1 and x_2

In-Class Exercise

Question 1:

a) Can a **k-Nearest Neighbor classifier with k=1** achieve **zero training error** on this dataset?

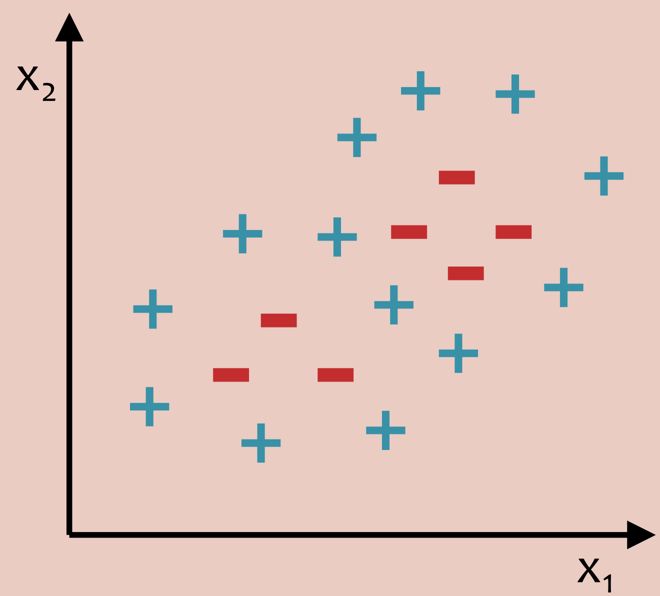
b) If 'Yes', draw the learned decision boundary. **If 'No'**, why not?



Question 2:

a) Can a **Decision Tree classifier** achieve **zero training error** on this dataset?

b) If 'Yes', draw the learned decision boundary for k-Nearest Neighbors with k=1. **If 'No'**, why not?



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

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
0	4.3	3.0
0	4.9	3.6
0	5.3	3.7
1	4.9	2.4
1	5.7	2.8
1	6.3	3.3
1	6.7	3.0

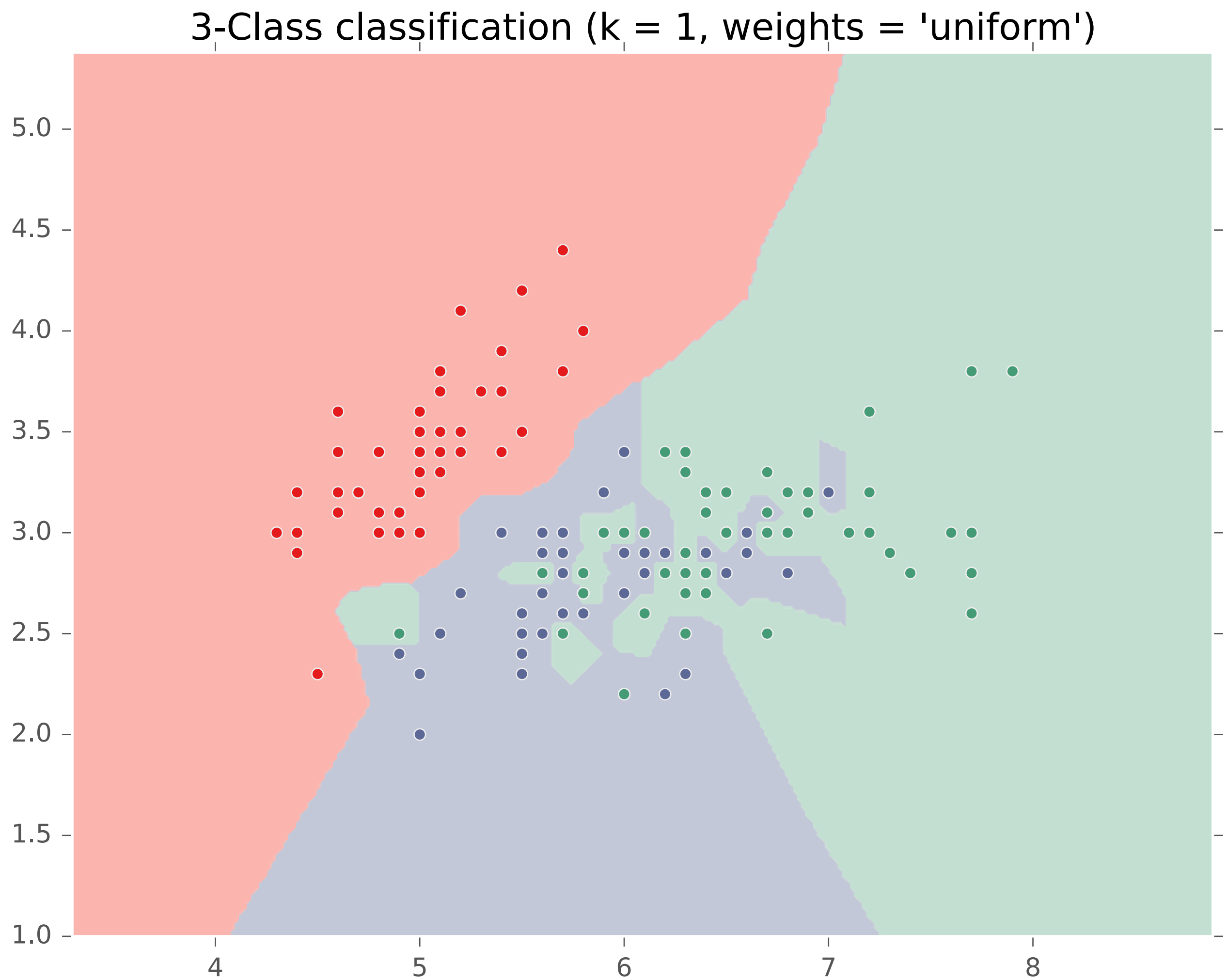
Deleted two of the four features, so that input space is 2D

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

KNN on Fisher Iris Data

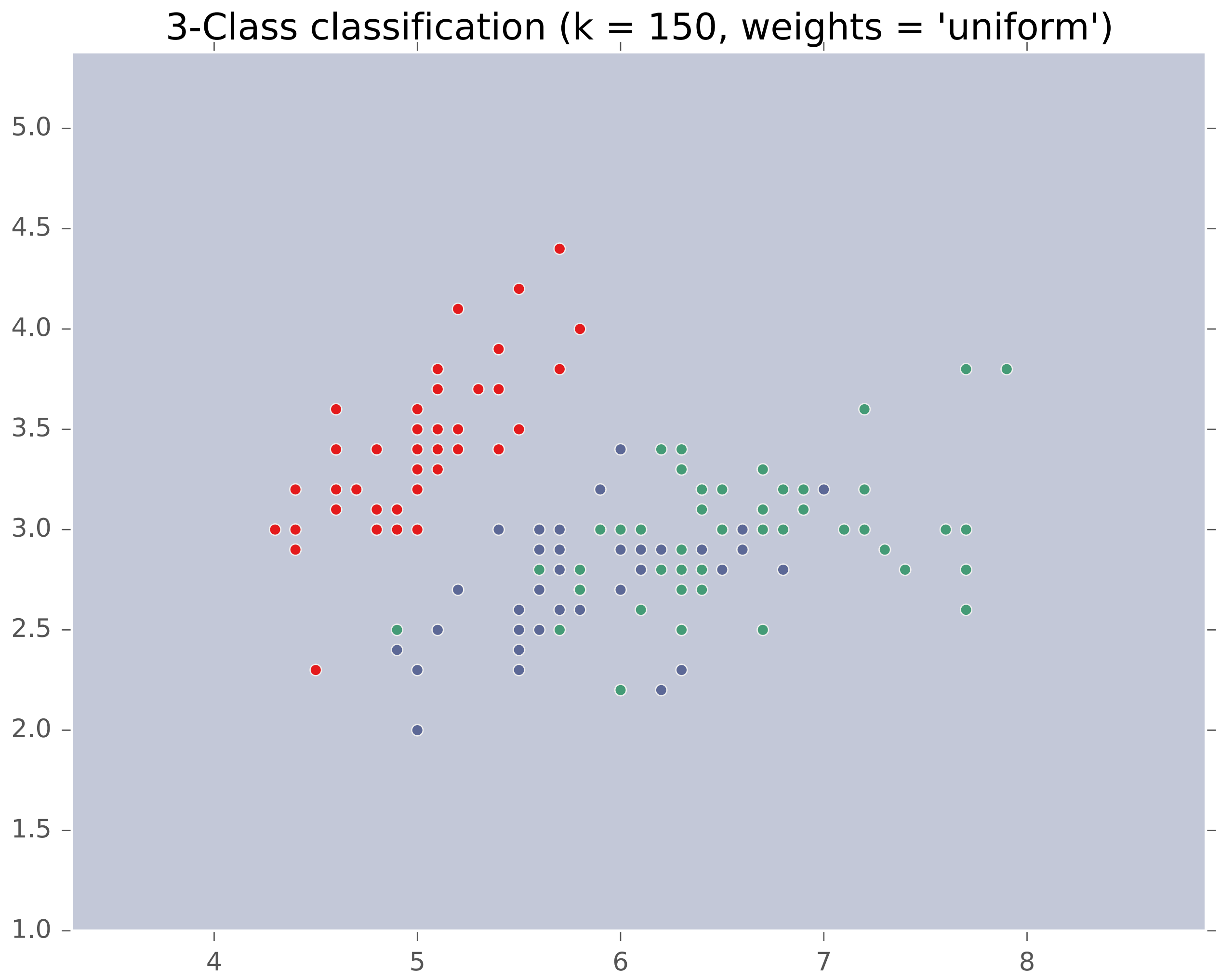
KNN on Fisher Iris Data

Special Case: Nearest Neighbor



KNN on Fisher Iris Data

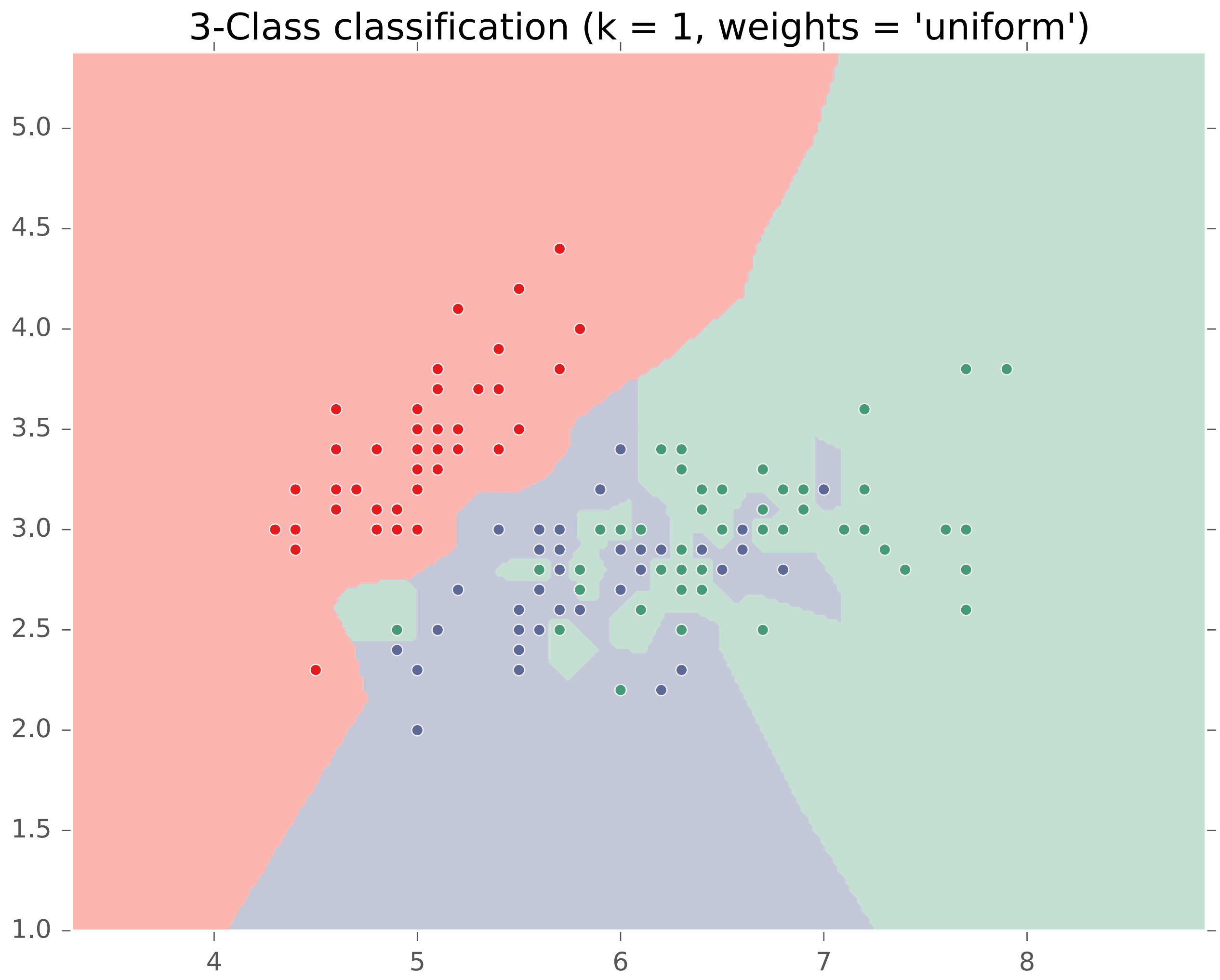
Special Case: Majority Vote



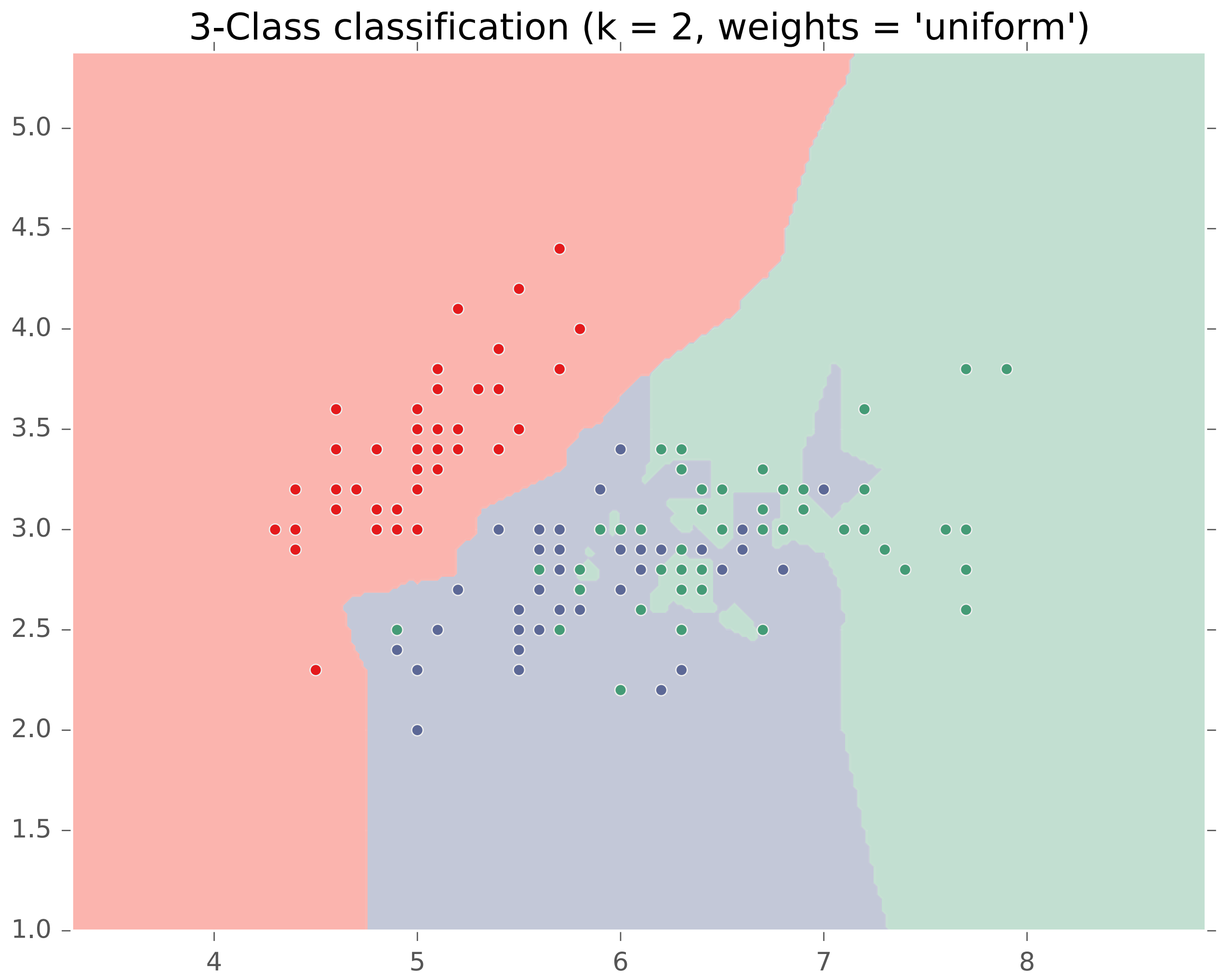
KNN on Fisher Iris Data

KNN on Fisher Iris Data

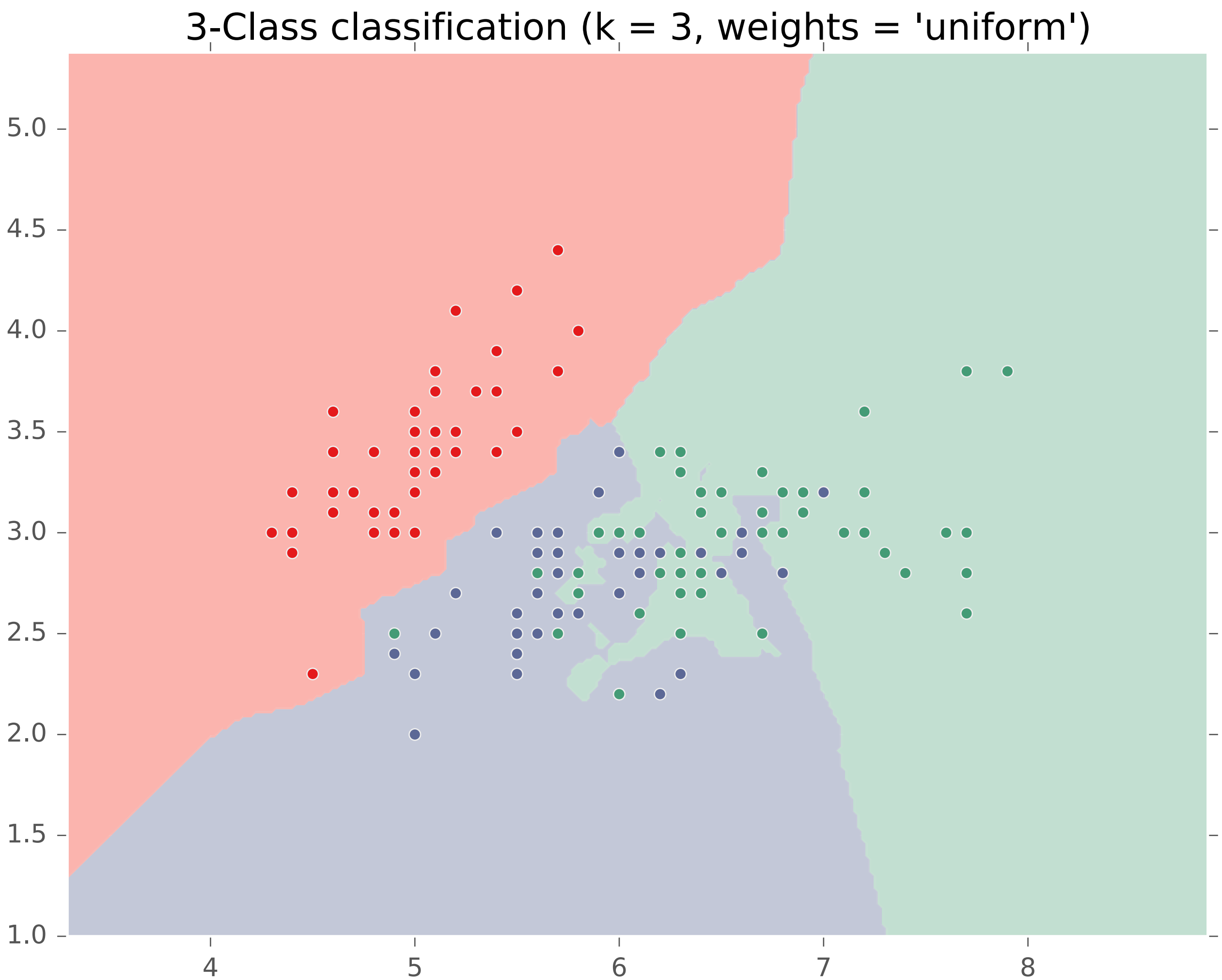
Special Case: Nearest Neighbor



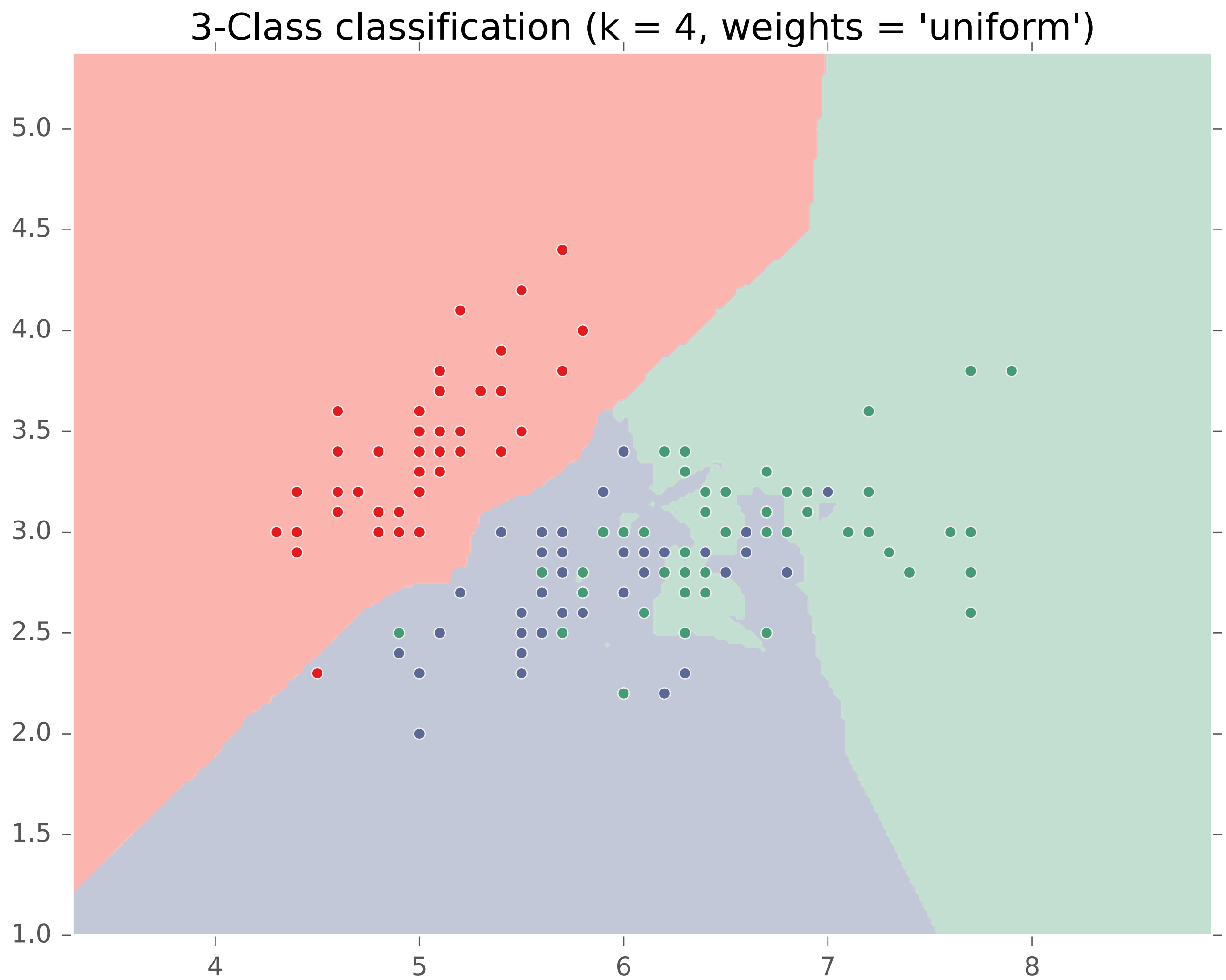
KNN on Fisher Iris Data



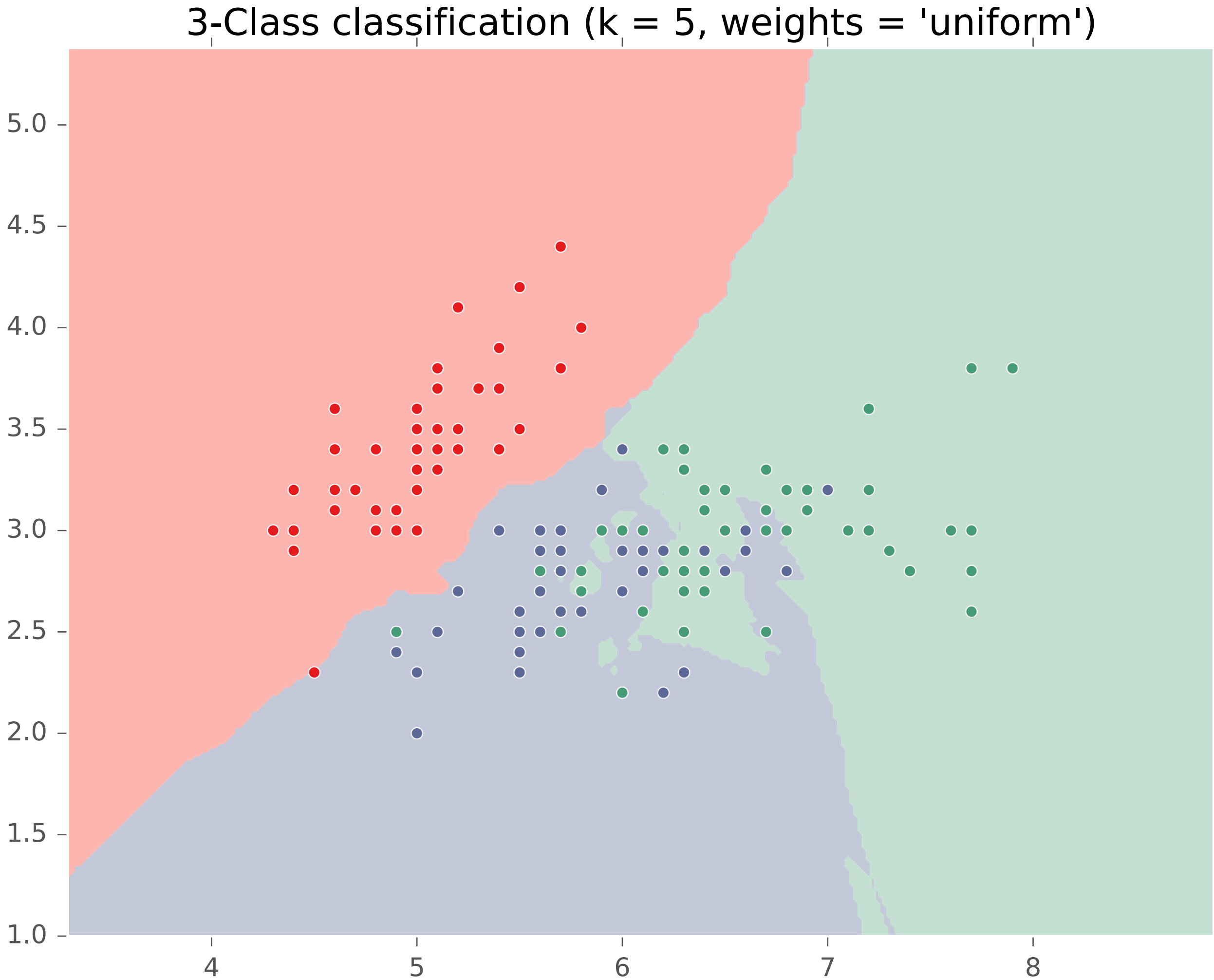
KNN on Fisher Iris Data



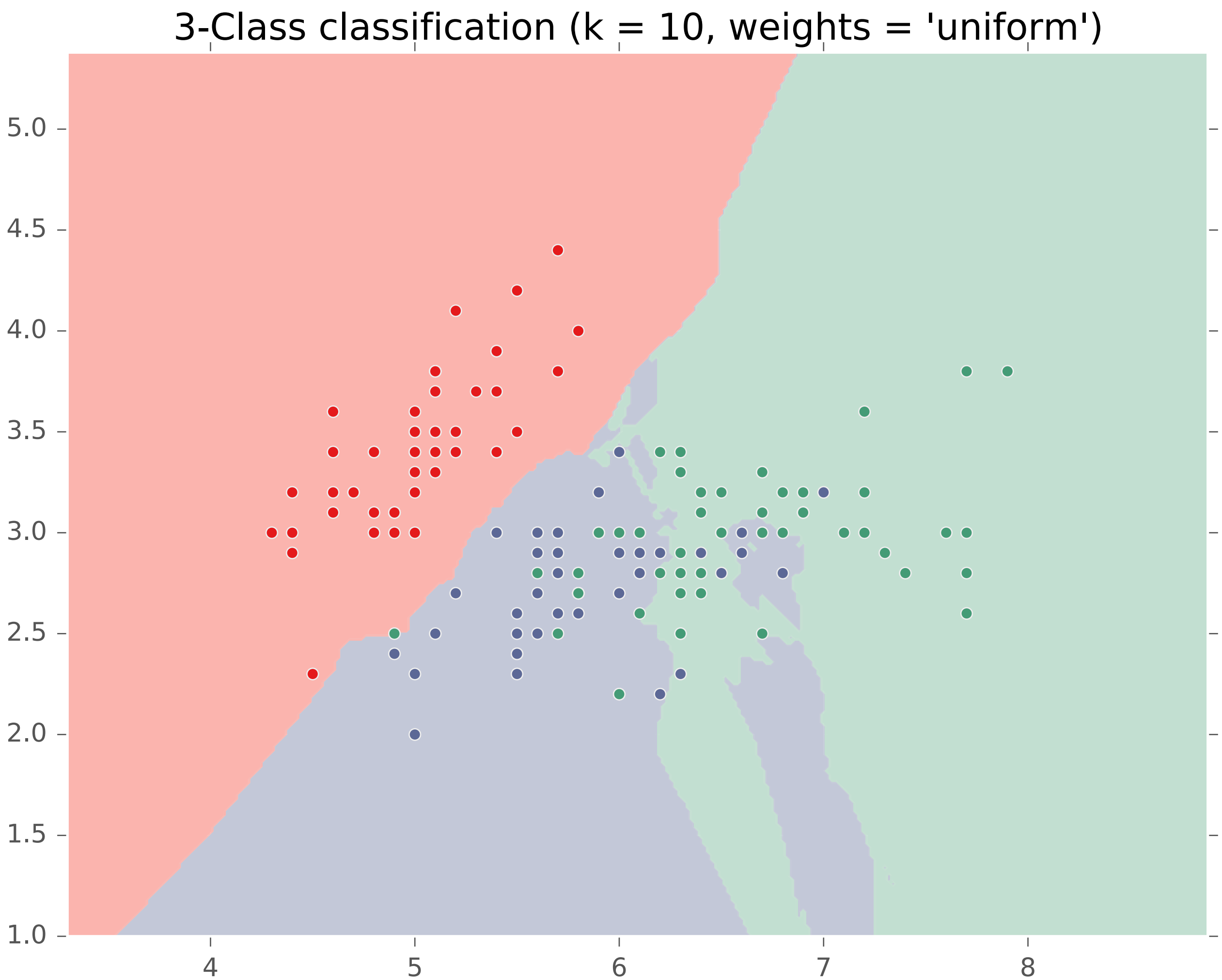
KNN on Fisher Iris Data



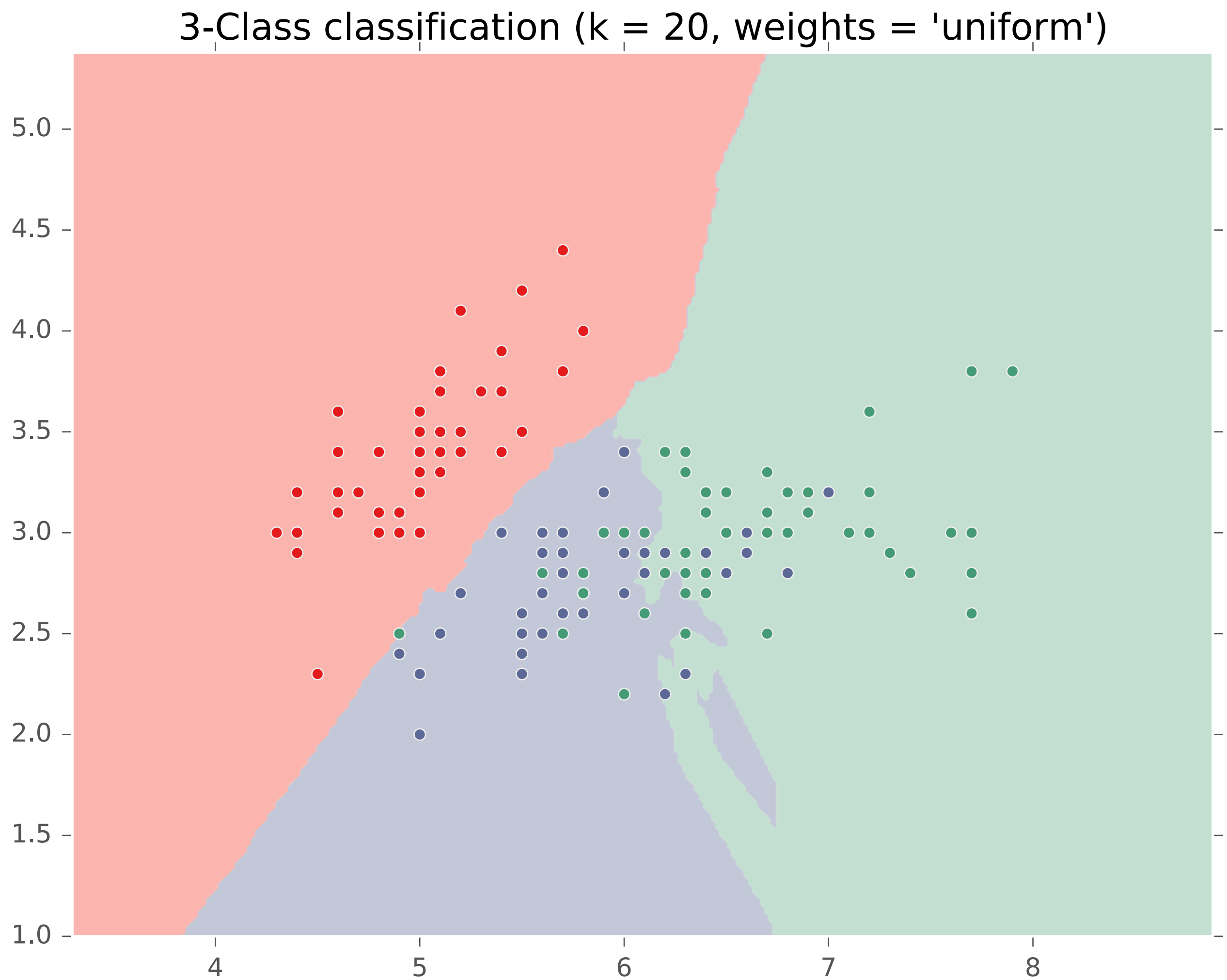
KNN on Fisher Iris Data



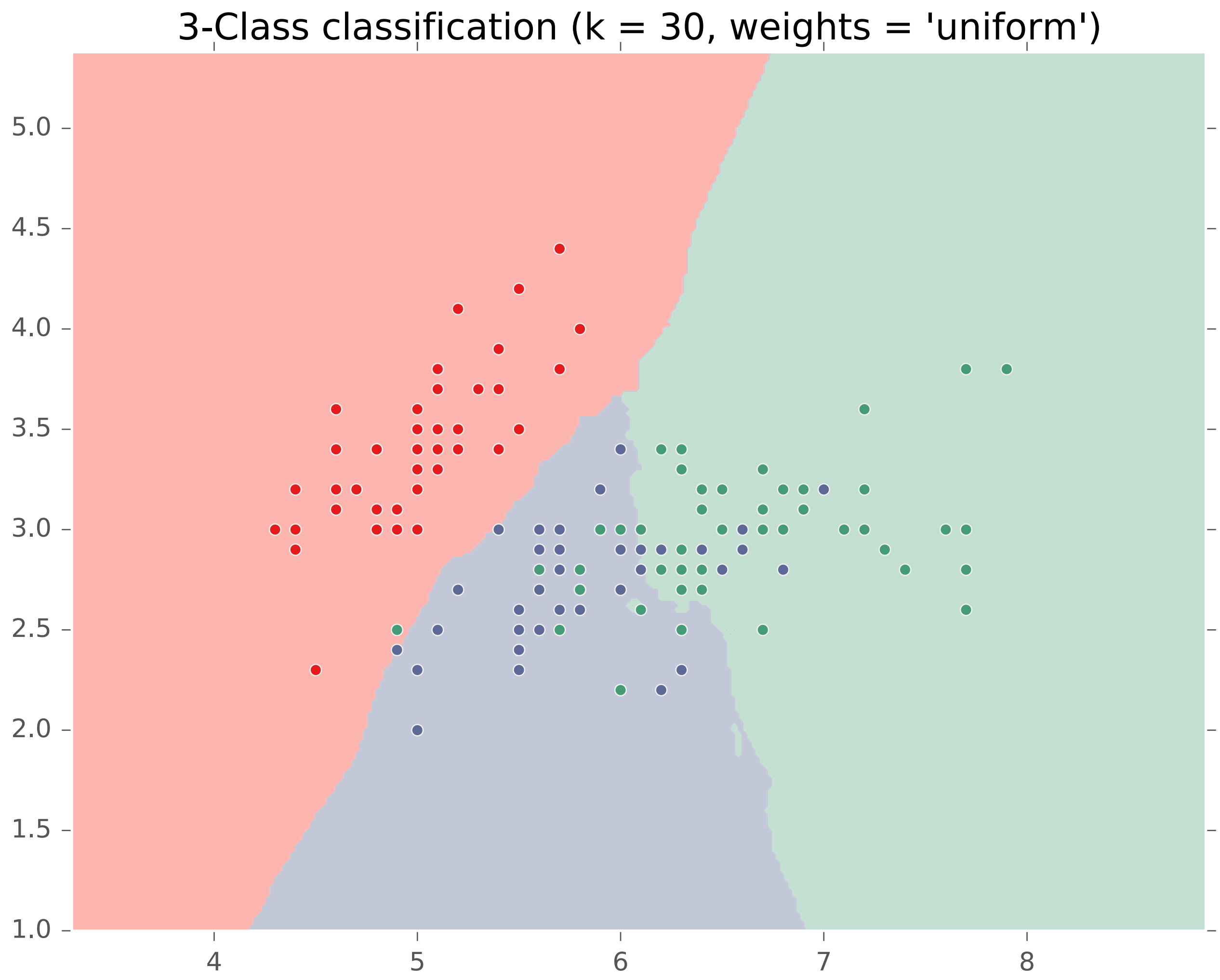
KNN on Fisher Iris Data



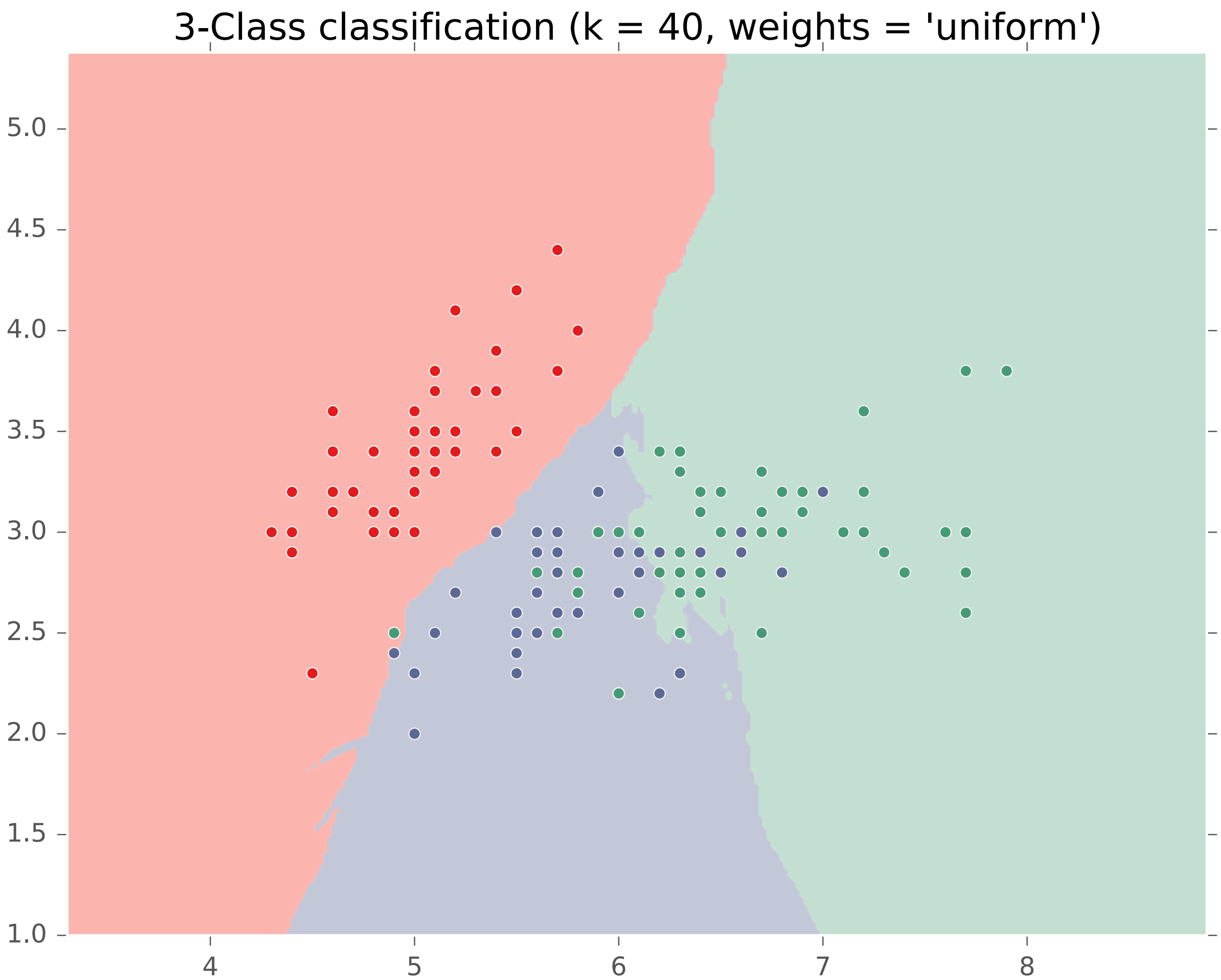
KNN on Fisher Iris Data



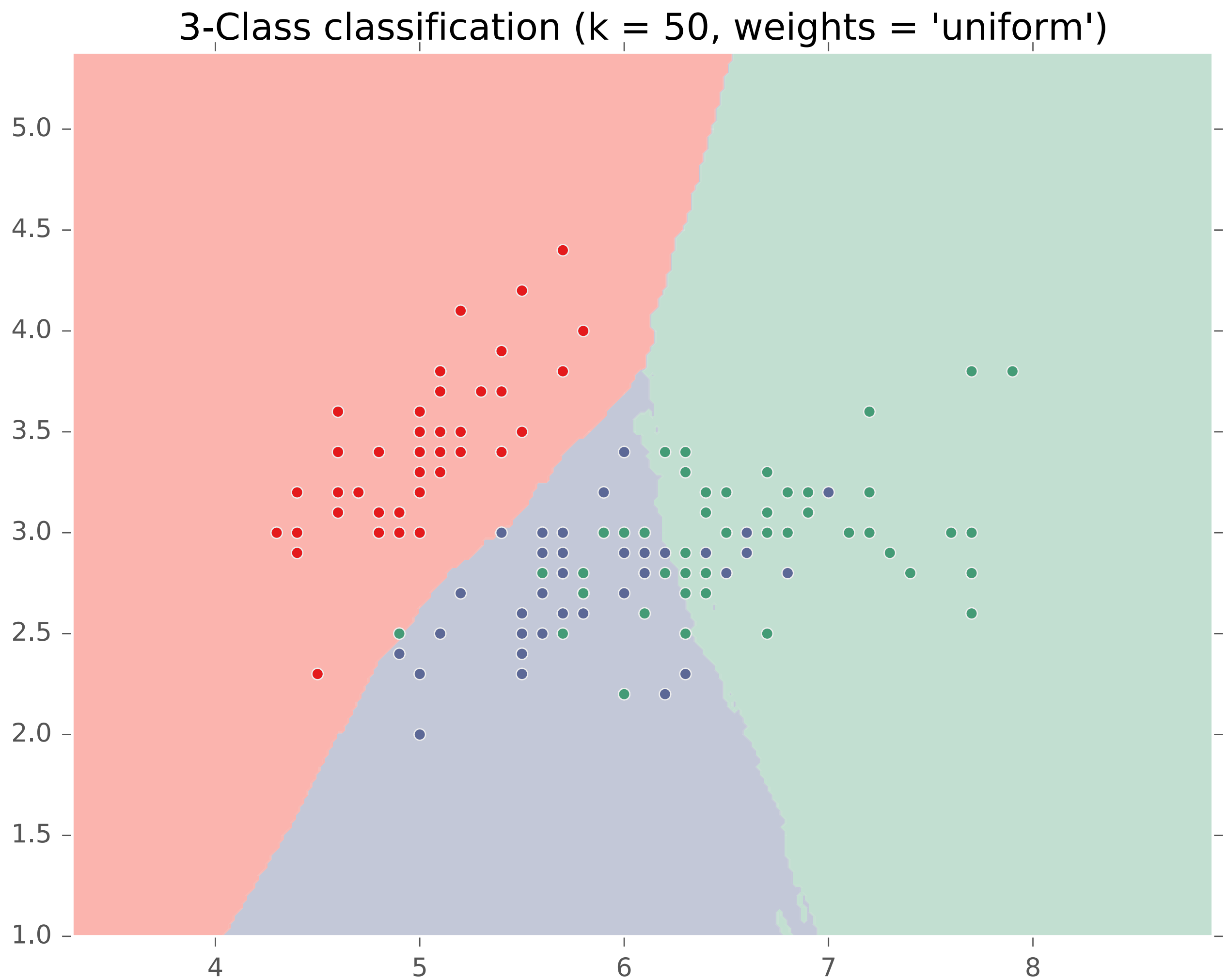
KNN on Fisher Iris Data



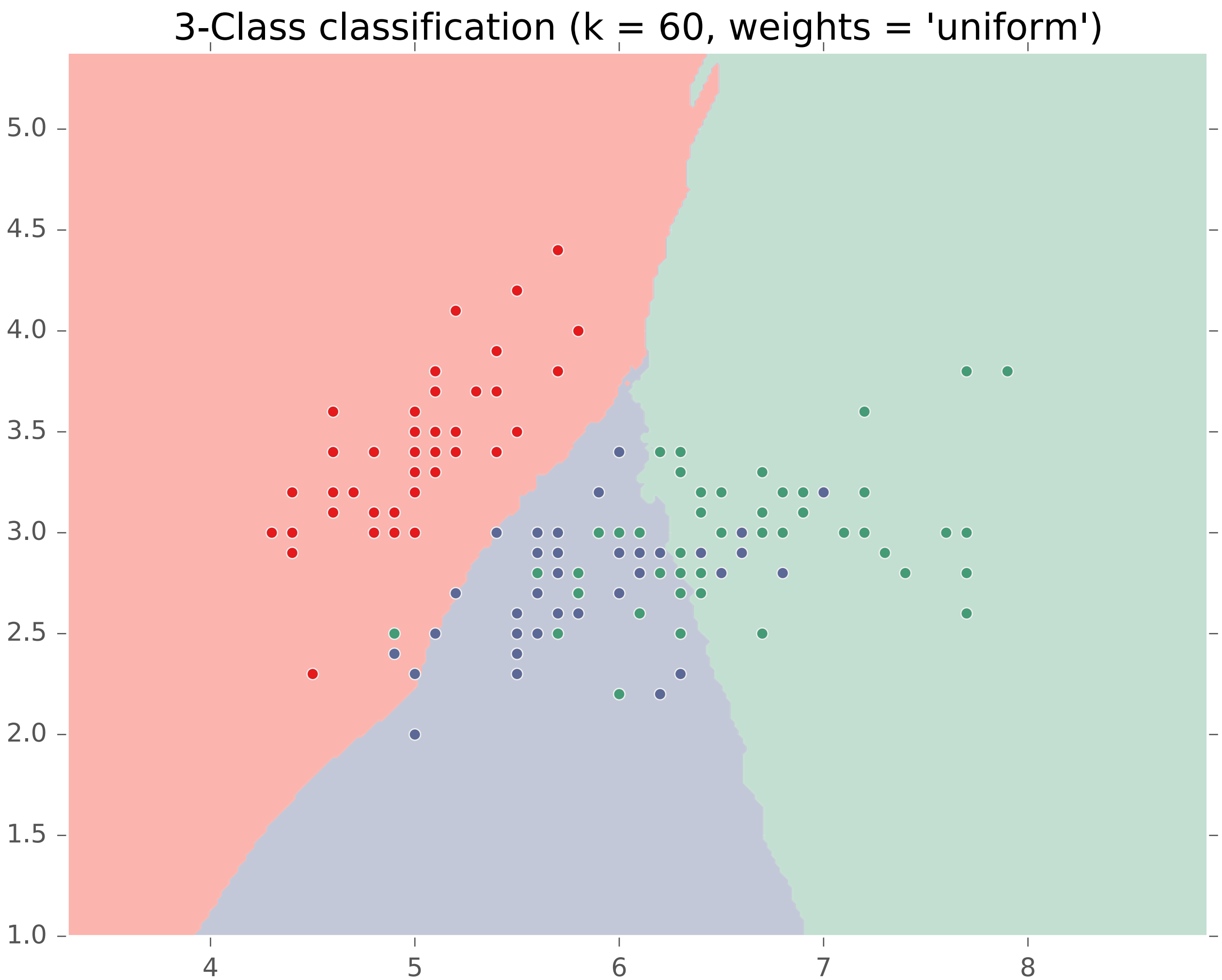
KNN on Fisher Iris Data



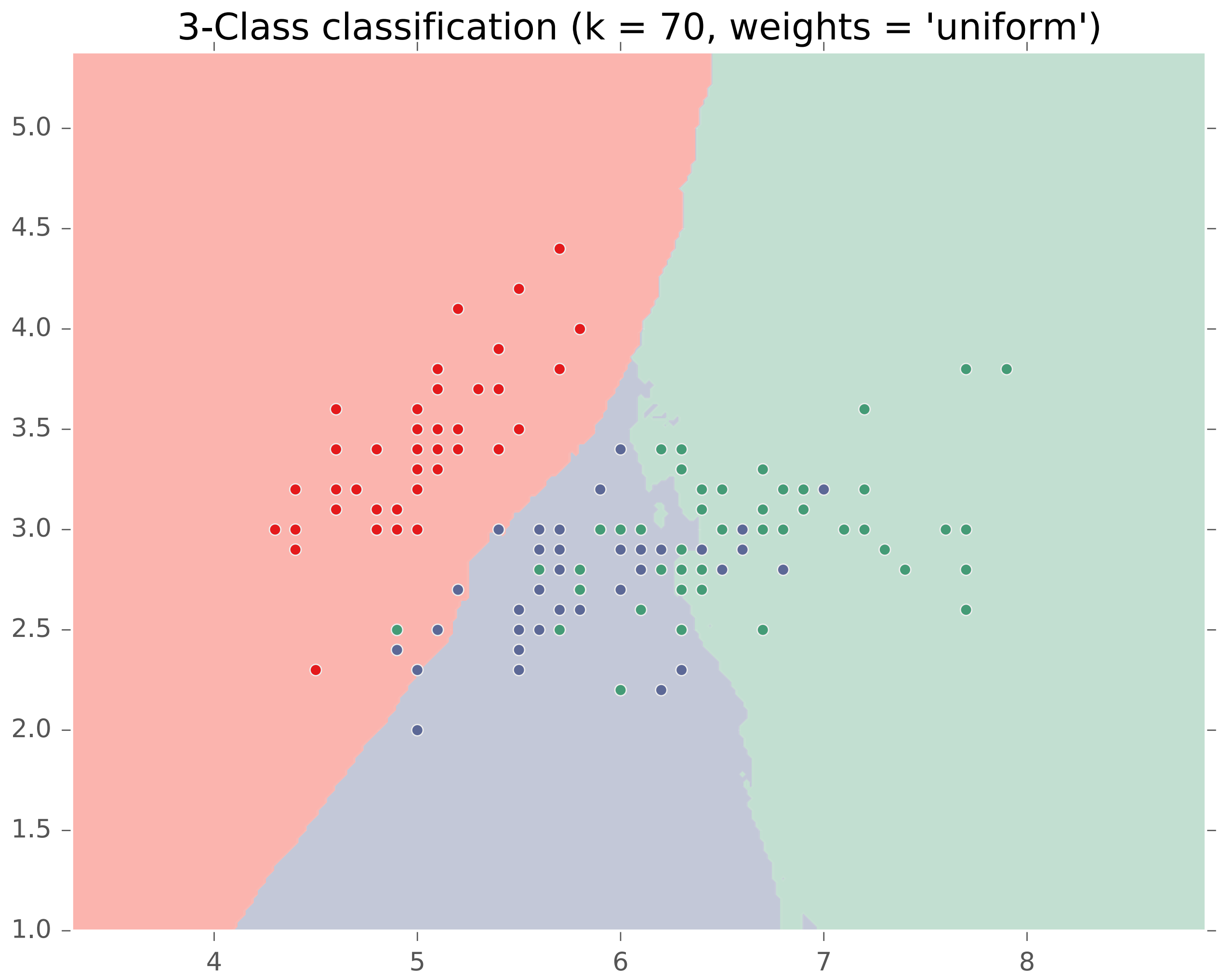
KNN on Fisher Iris Data



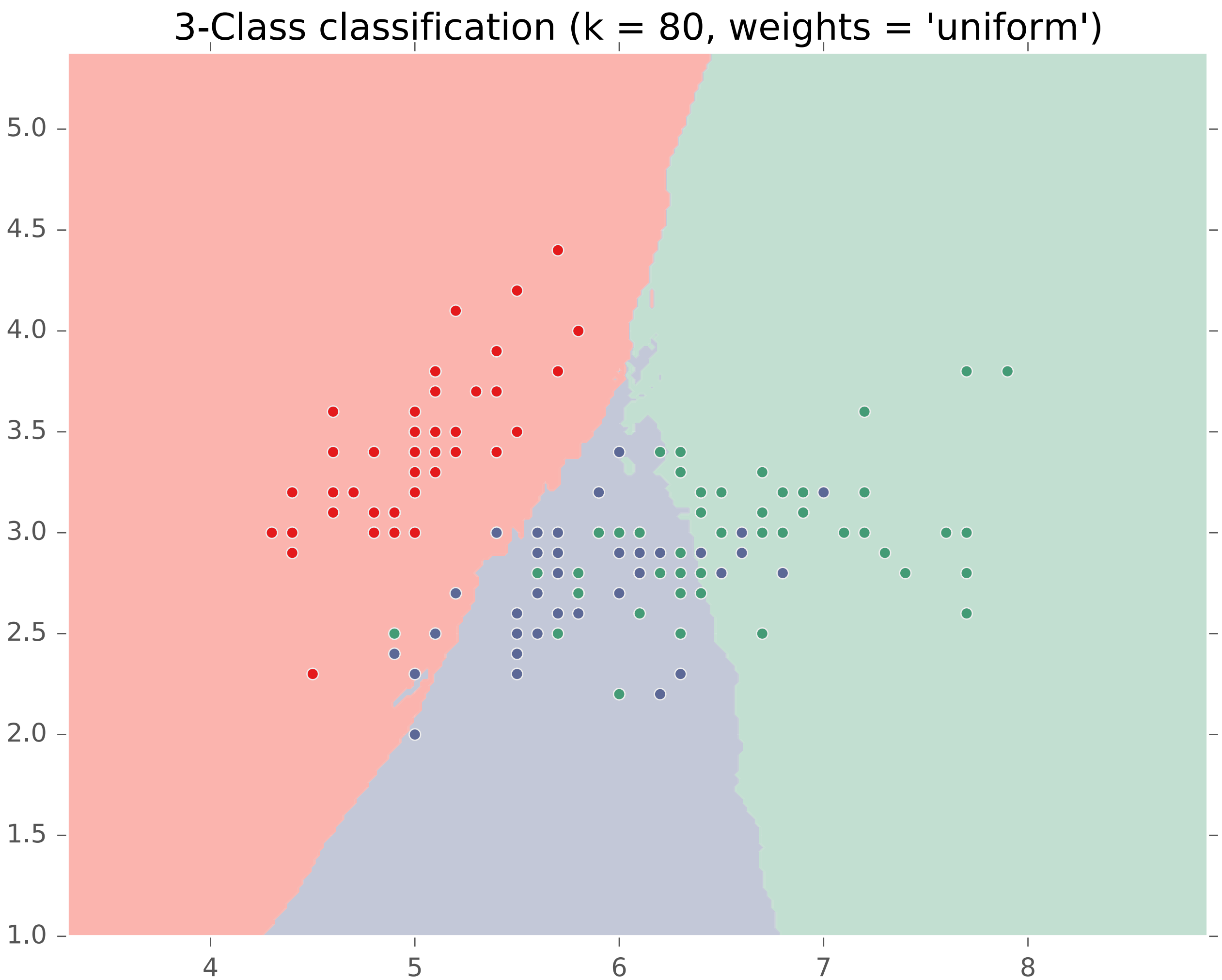
KNN on Fisher Iris Data



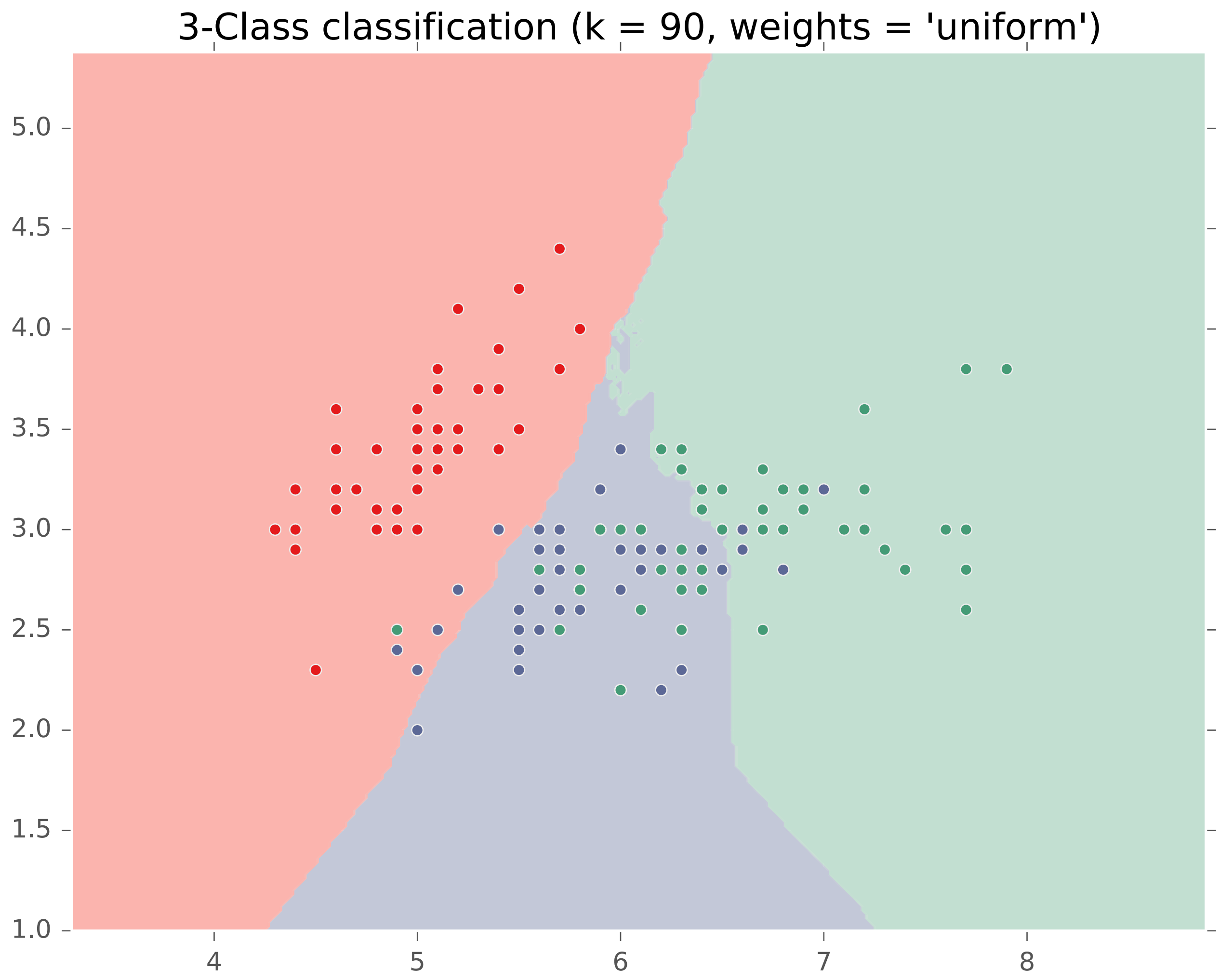
KNN on Fisher Iris Data



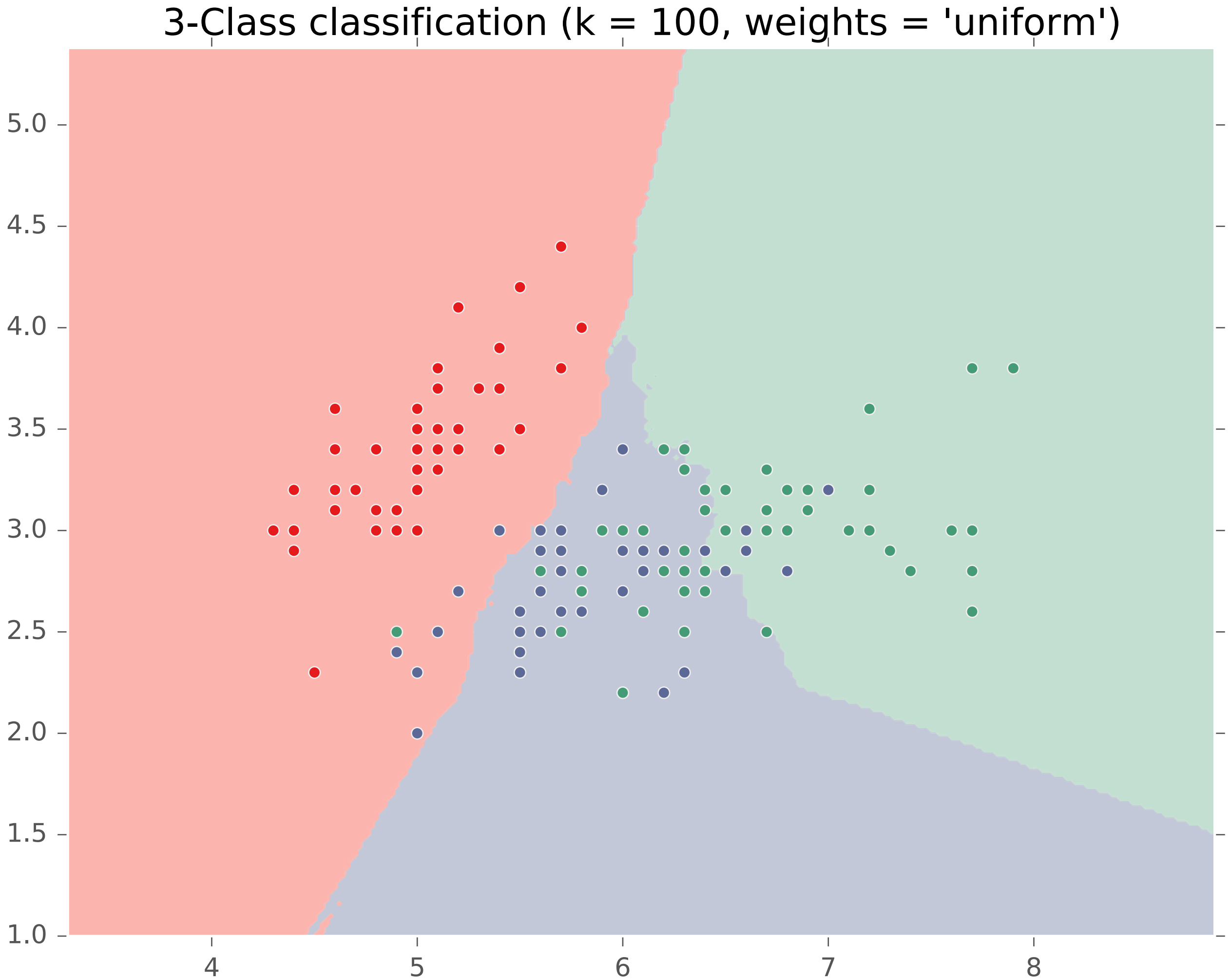
KNN on Fisher Iris Data



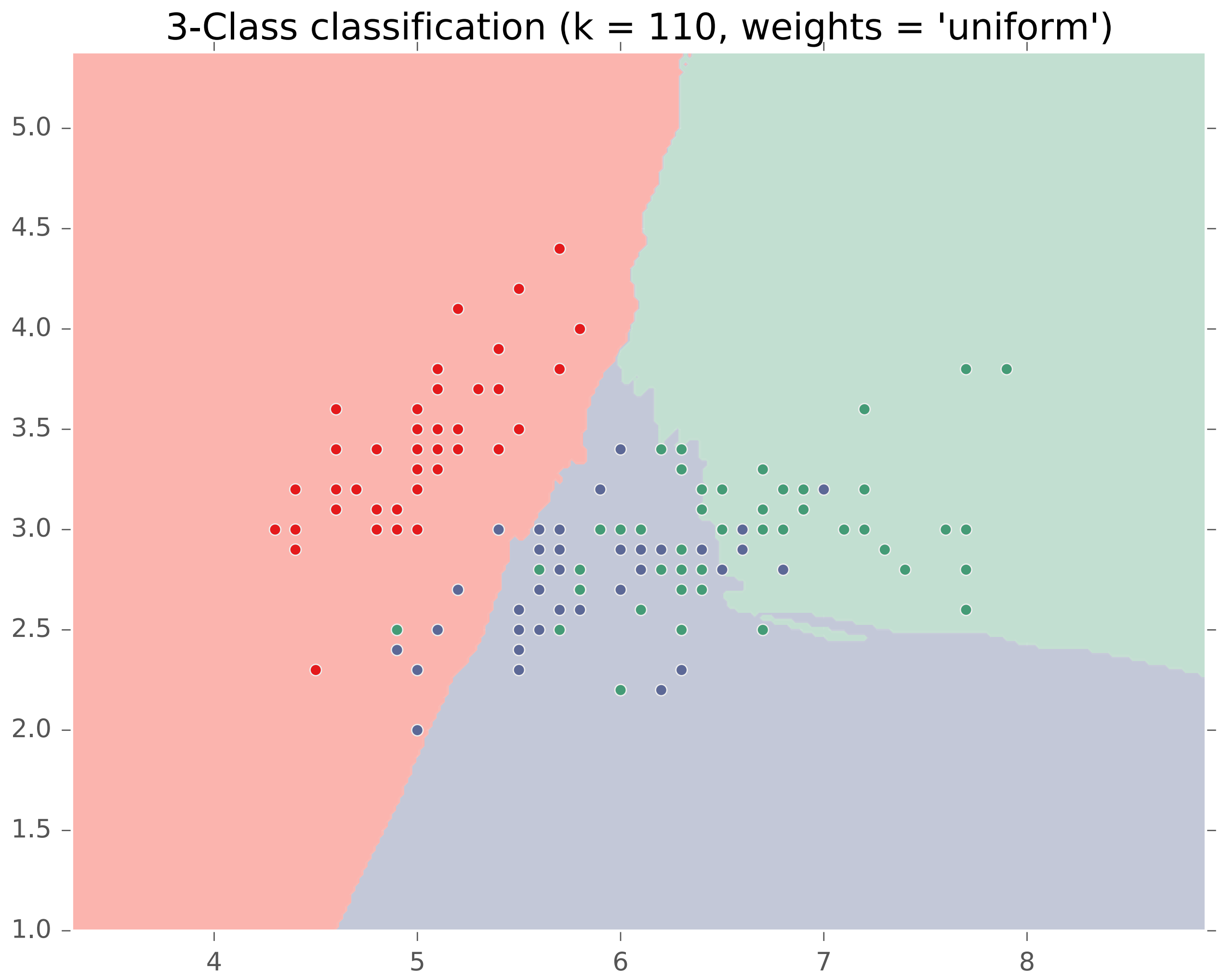
KNN on Fisher Iris Data



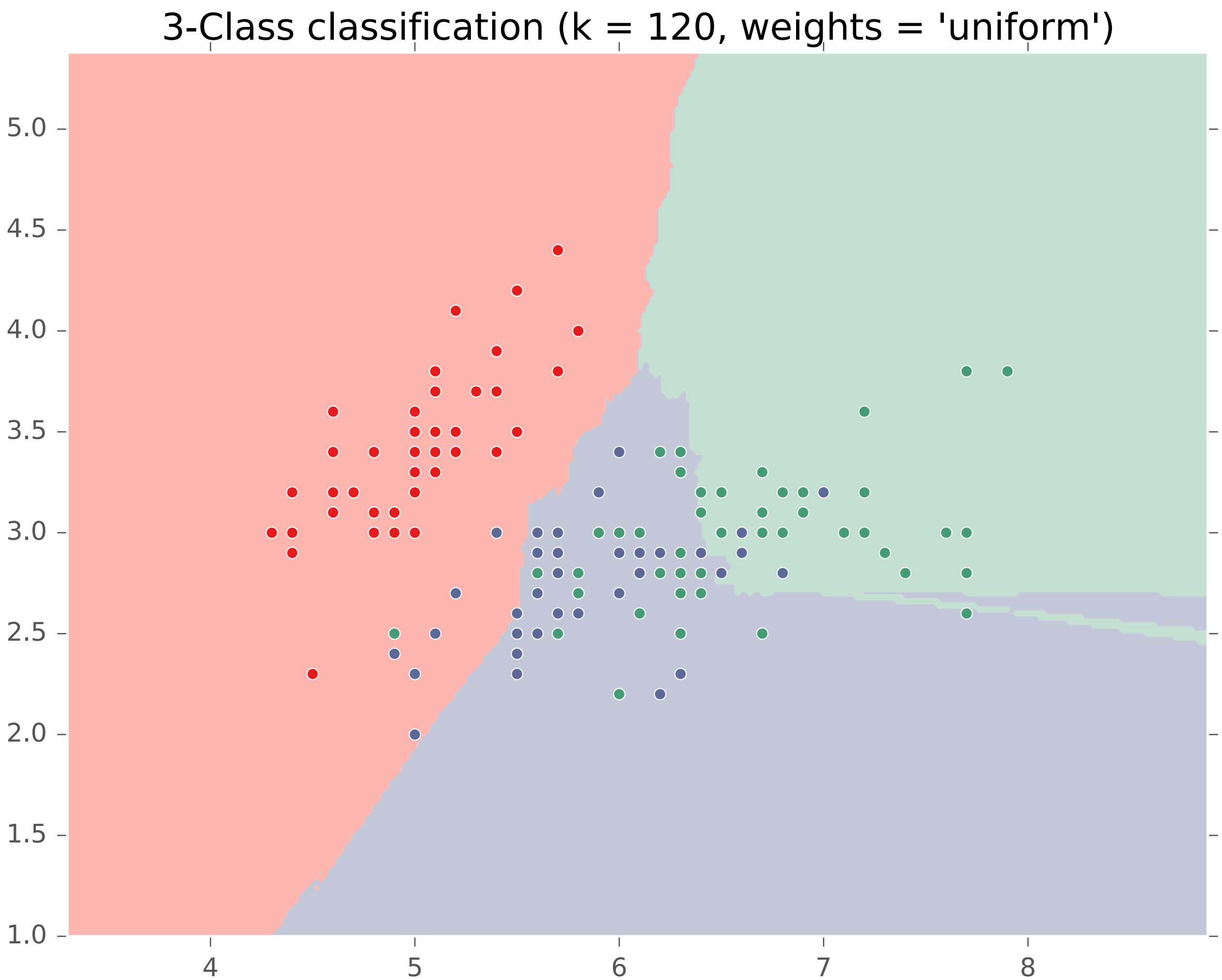
KNN on Fisher Iris Data



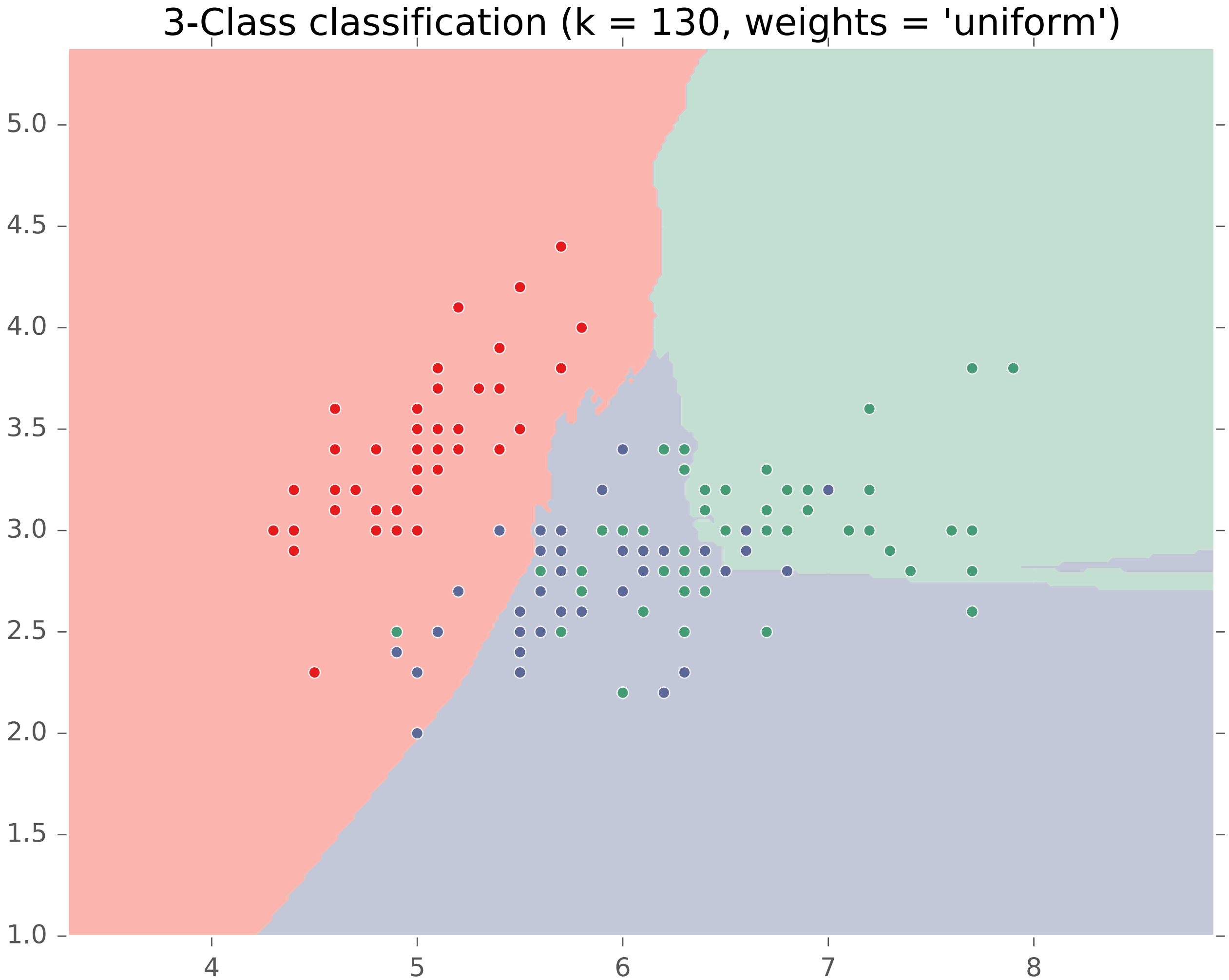
KNN on Fisher Iris Data



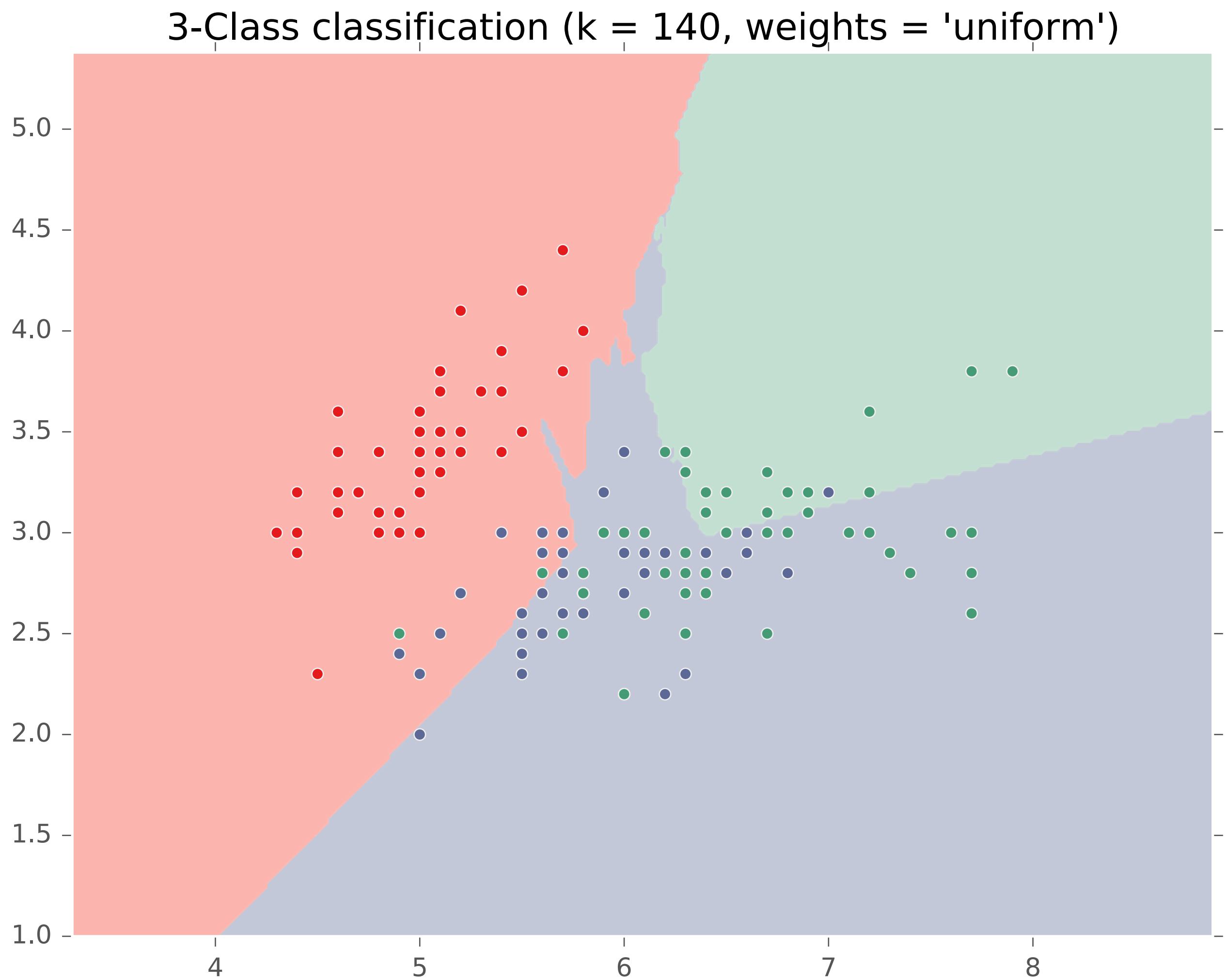
KNN on Fisher Iris Data



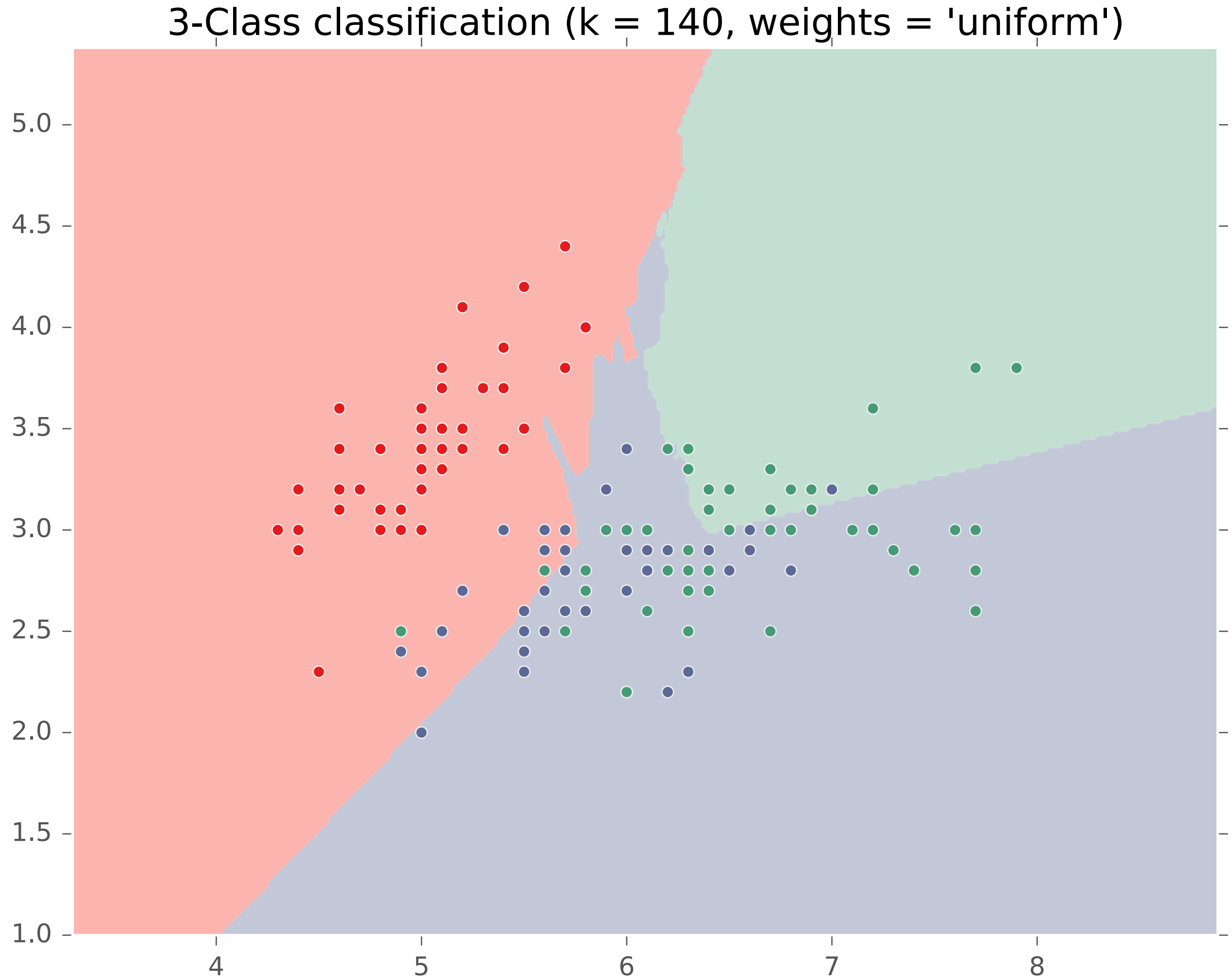
KNN on Fisher Iris Data



KNN on Fisher Iris Data

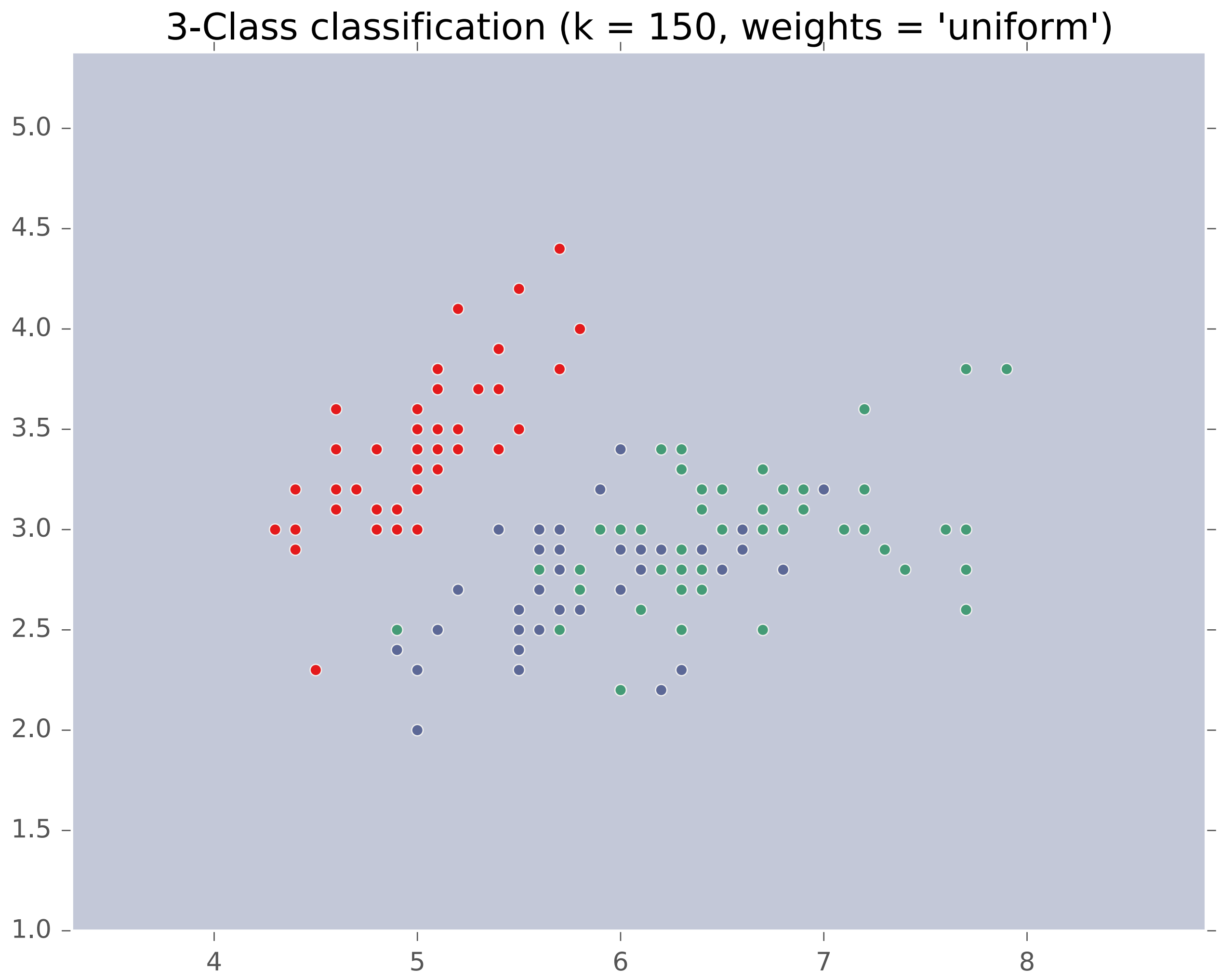


KNN on Fisher Iris Data



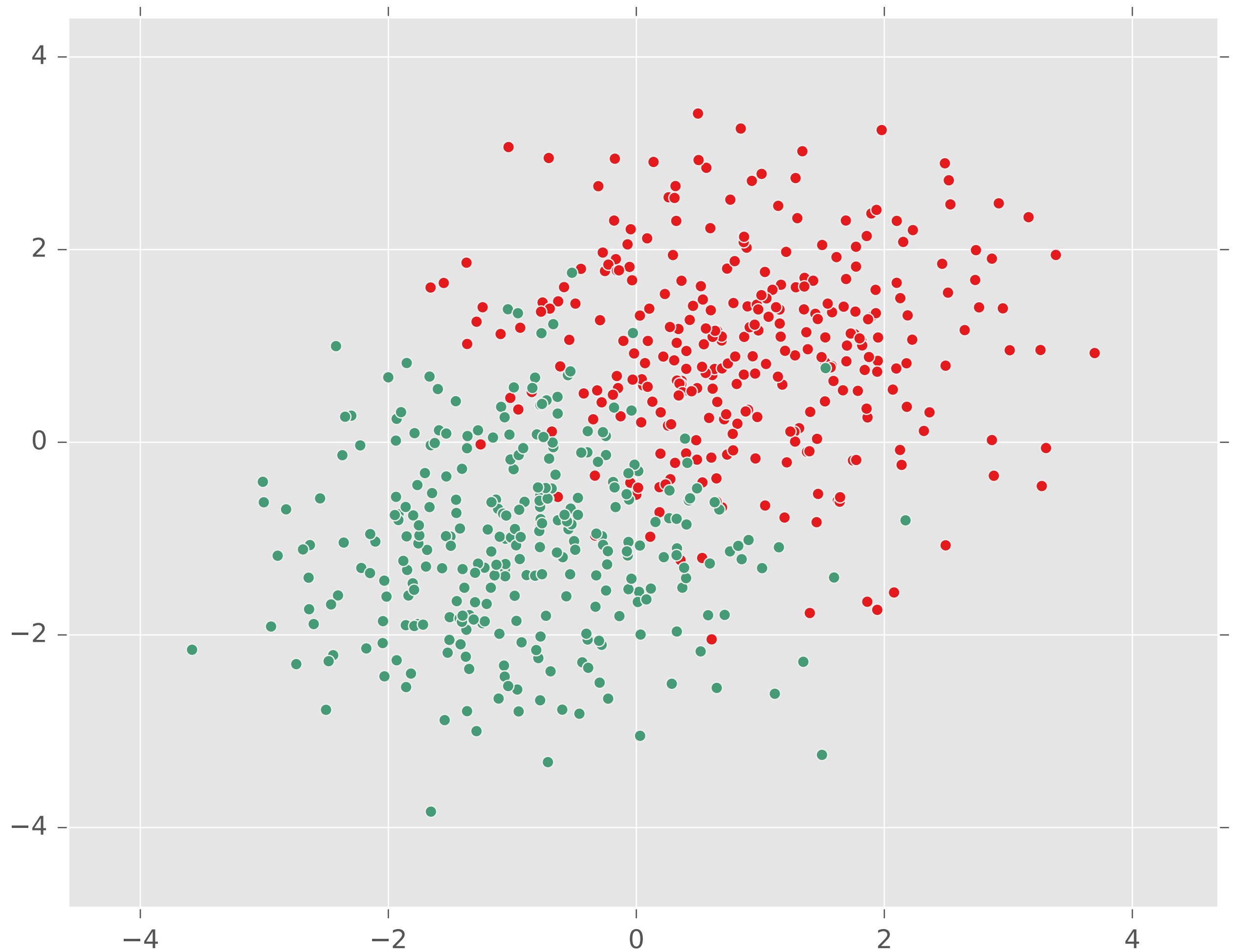
KNN on Fisher Iris Data

Special Case: Majority Vote

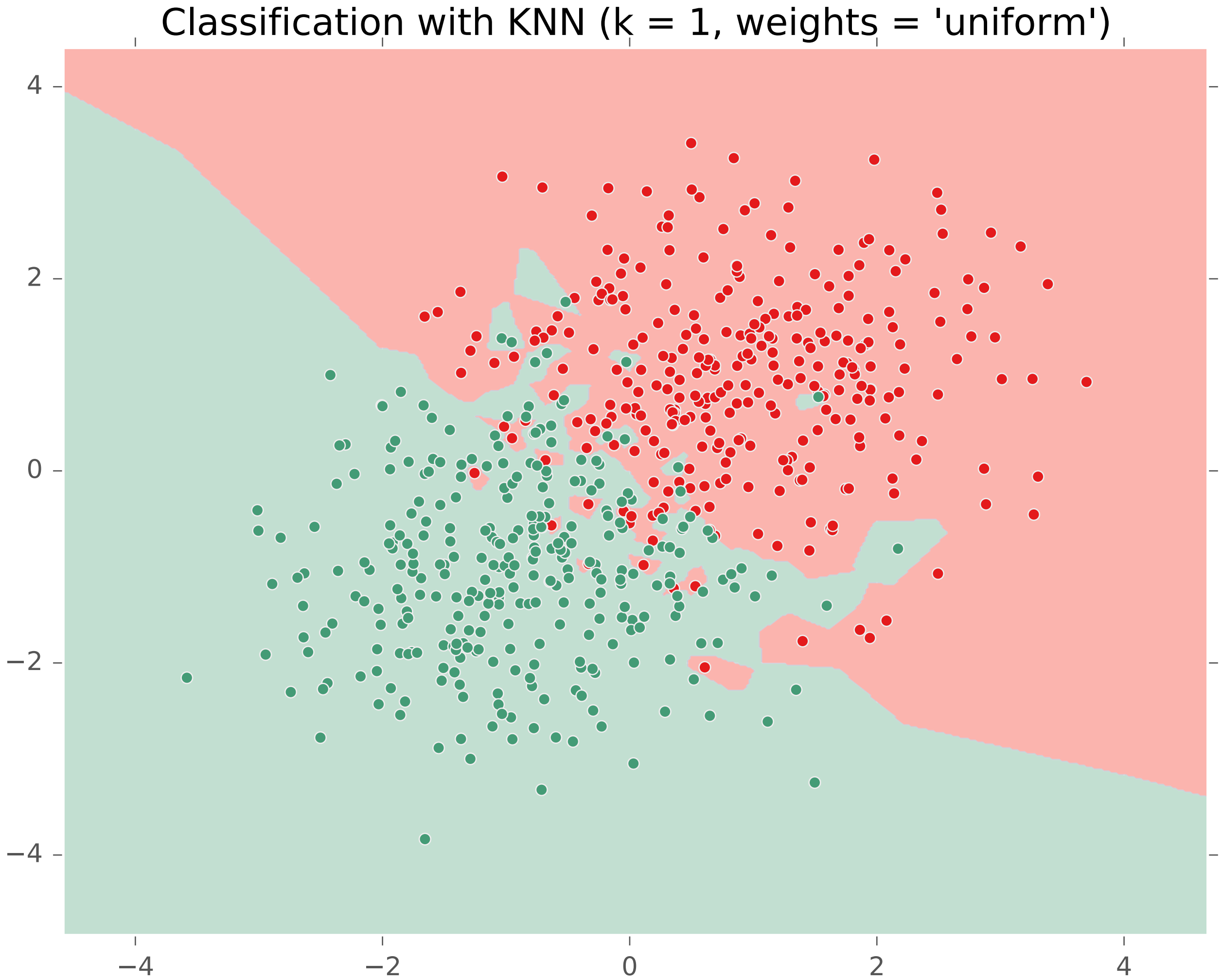


KNN ON GAUSSIAN DATA

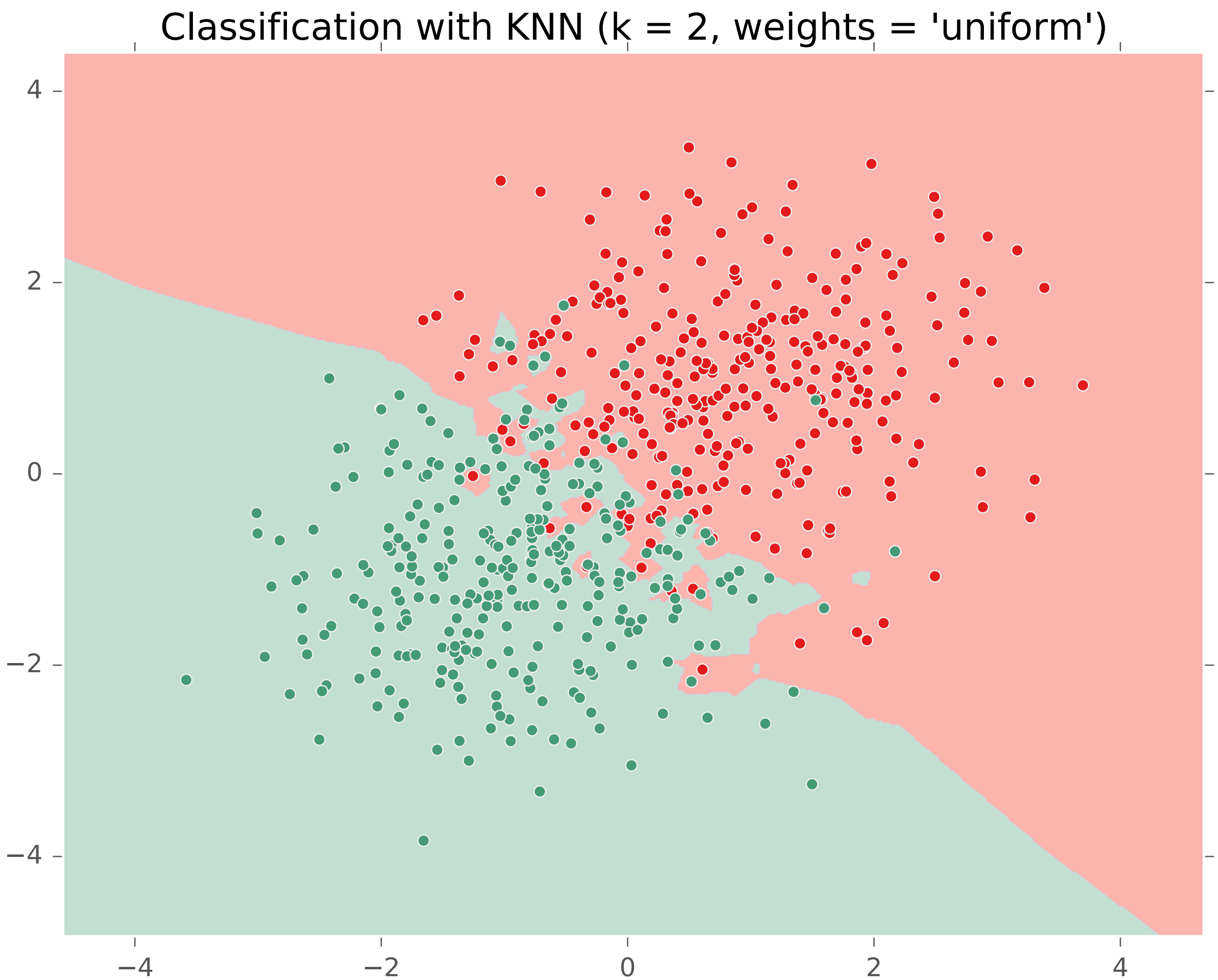
KNN on Gaussian Data



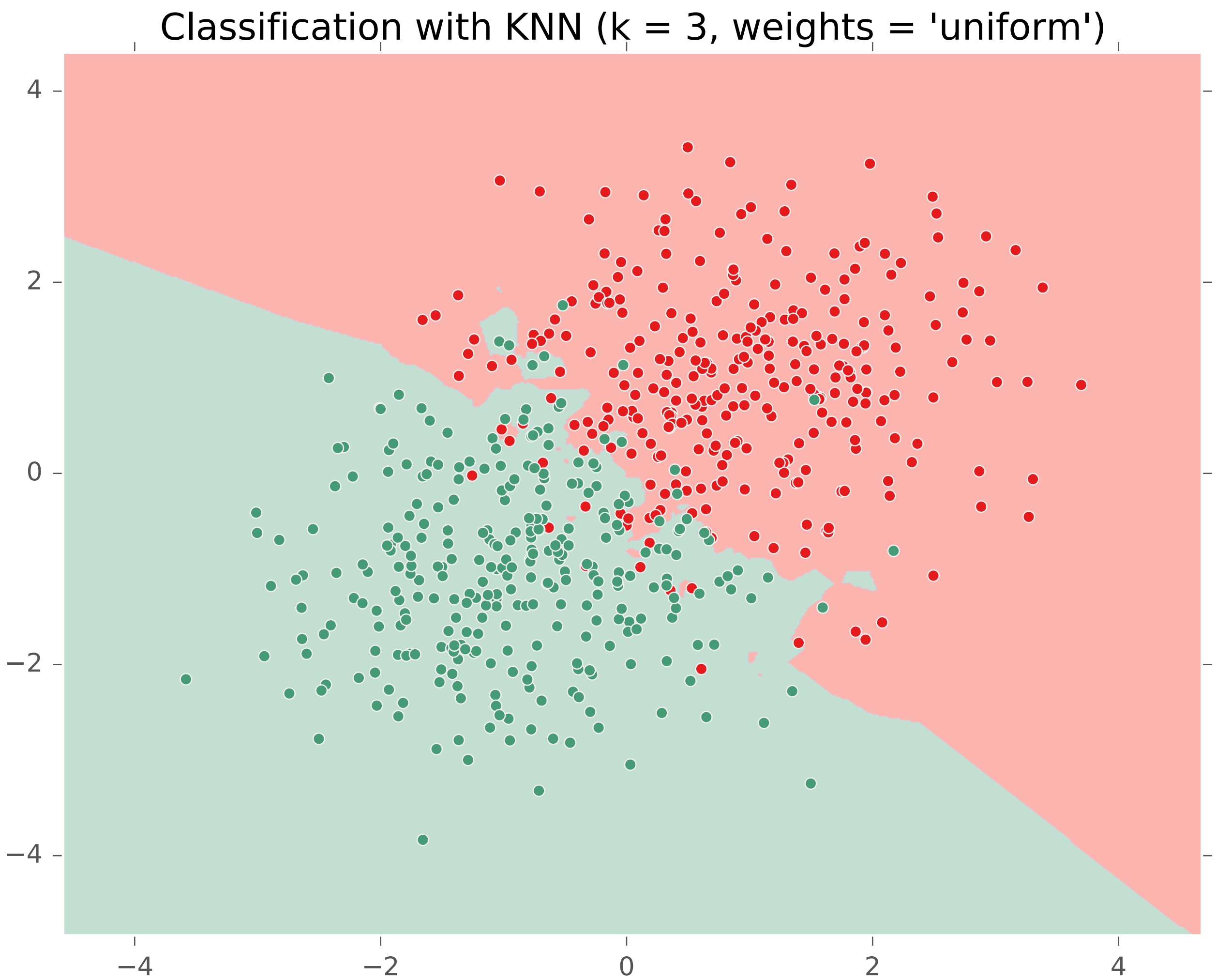
KNN on Gaussian Data



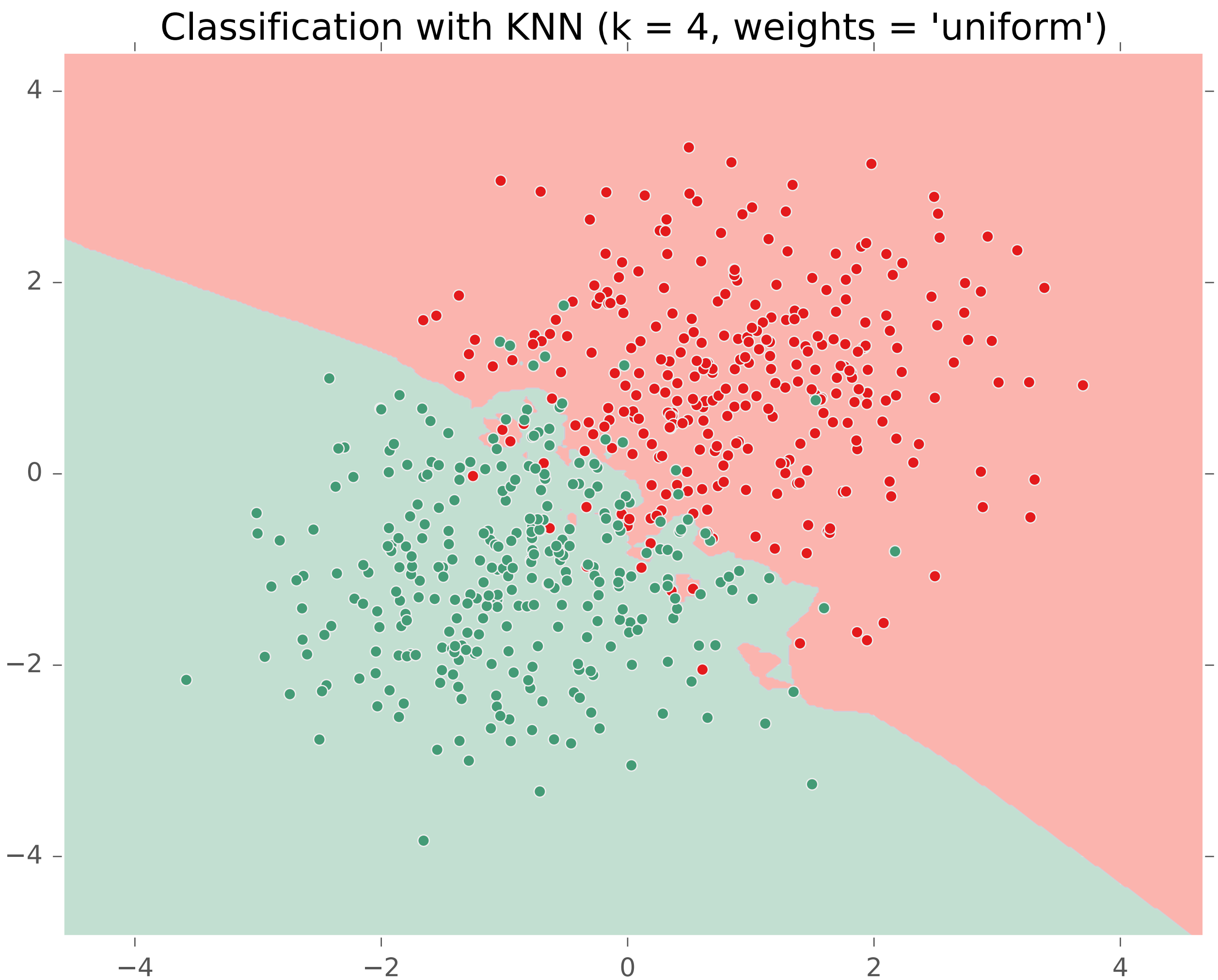
KNN on Gaussian Data



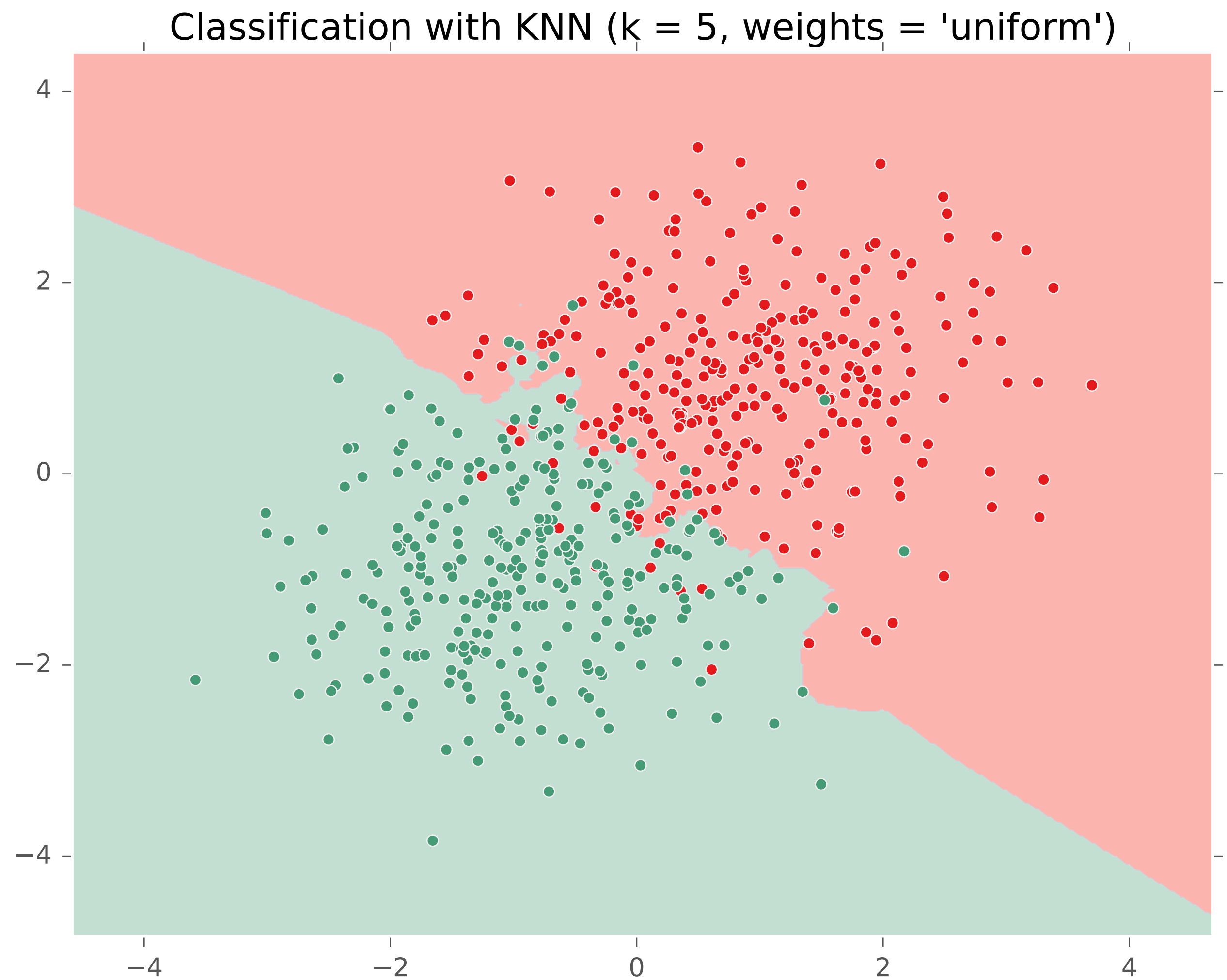
KNN on Gaussian Data



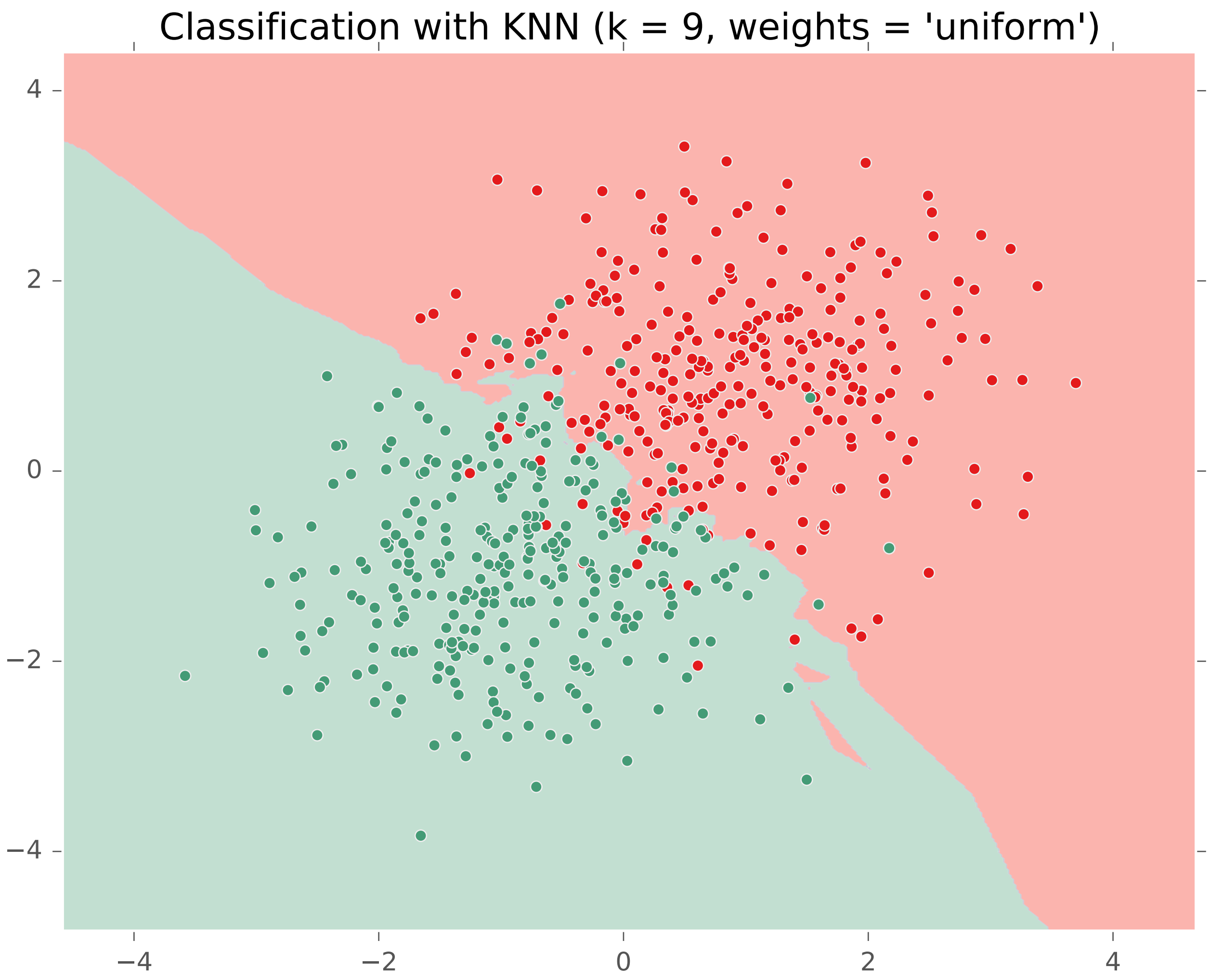
KNN on Gaussian Data



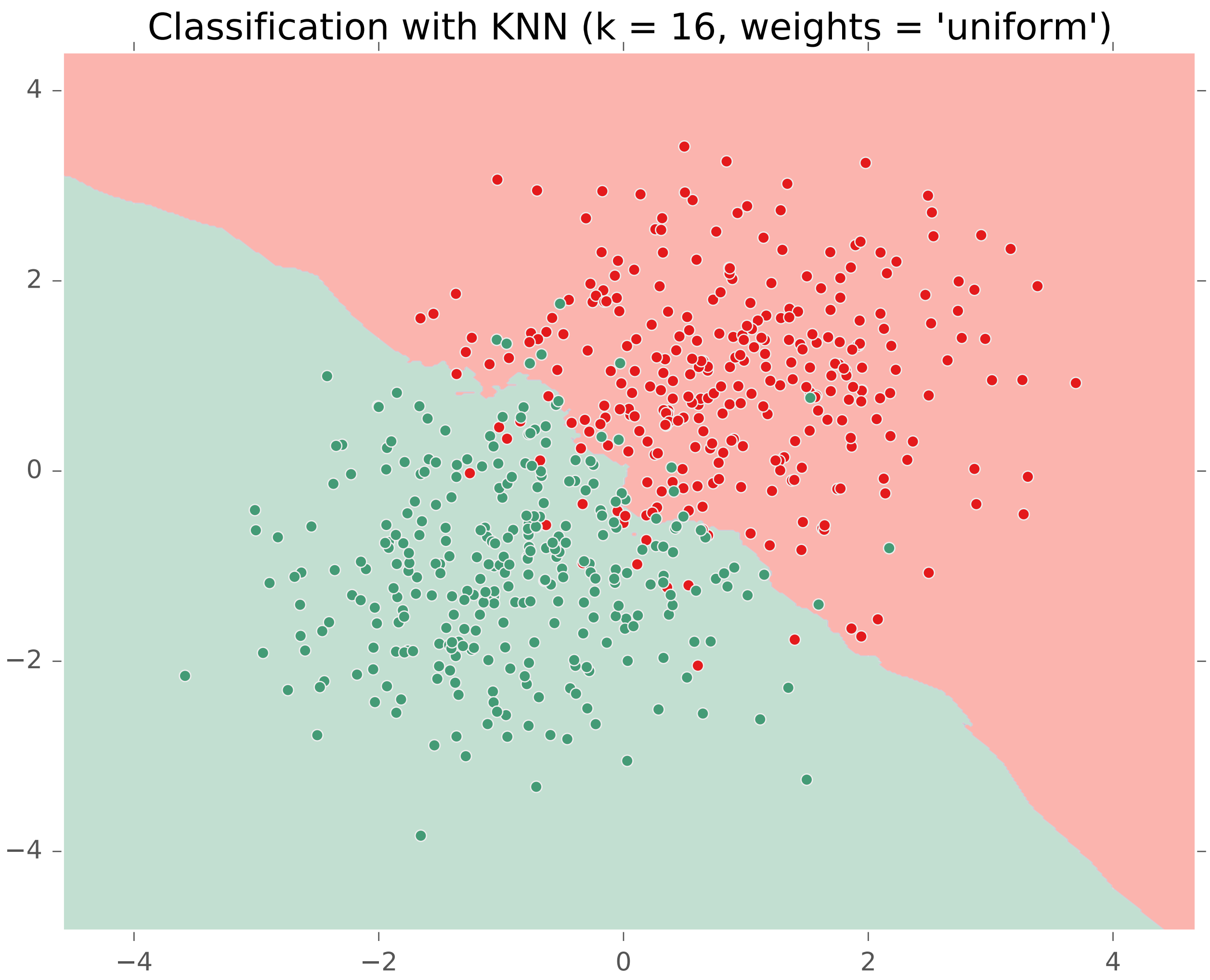
KNN on Gaussian Data



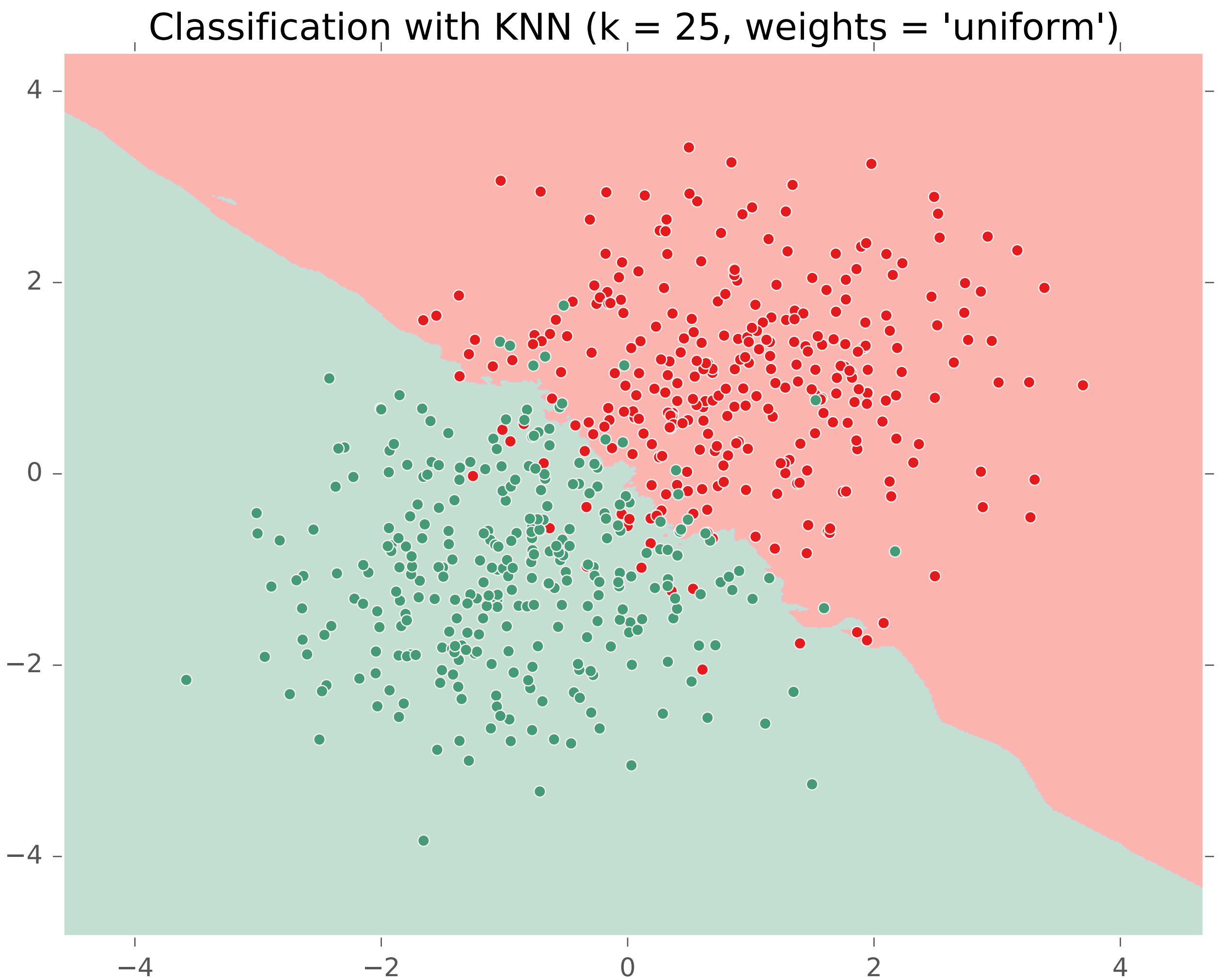
KNN on Gaussian Data



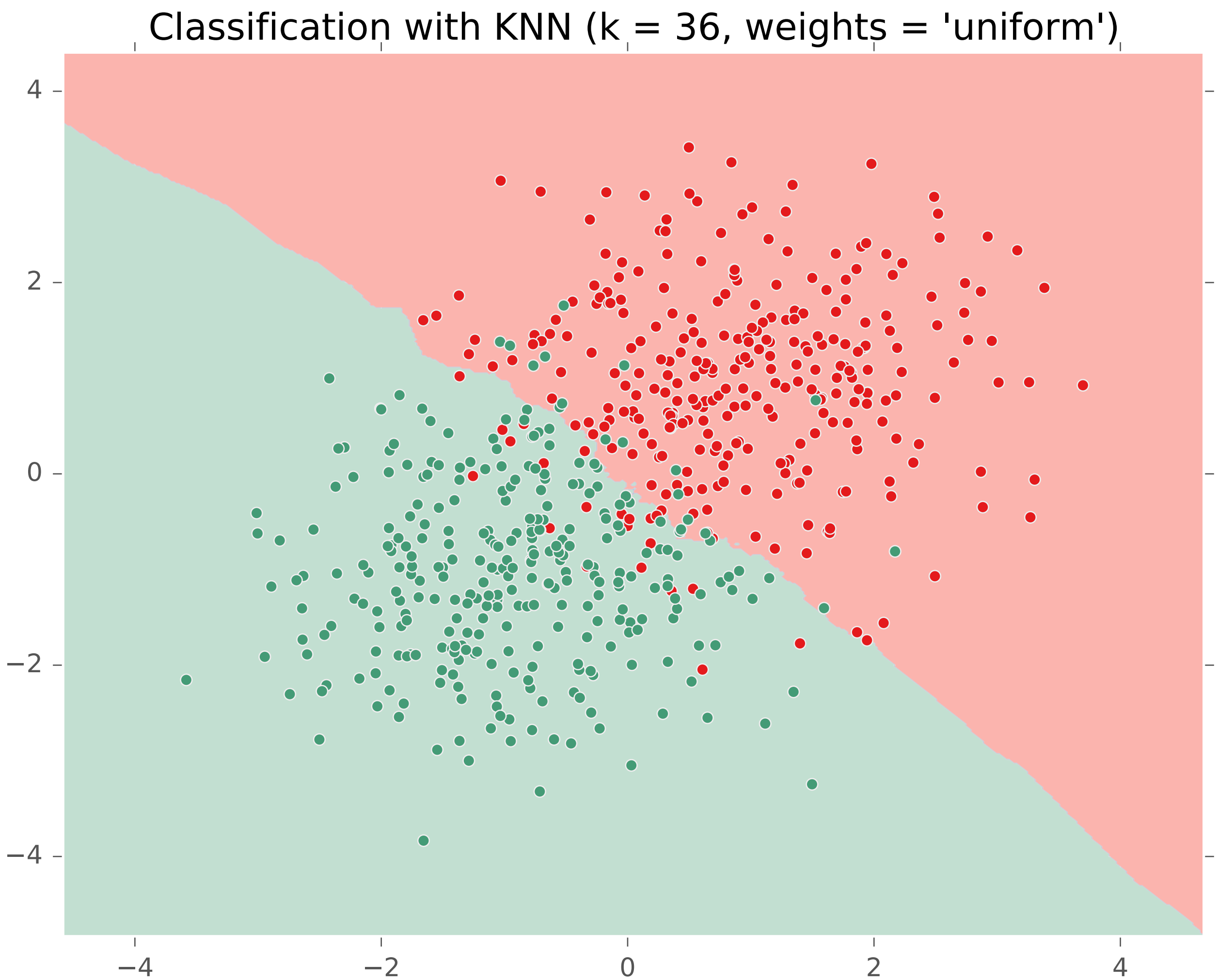
KNN on Gaussian Data



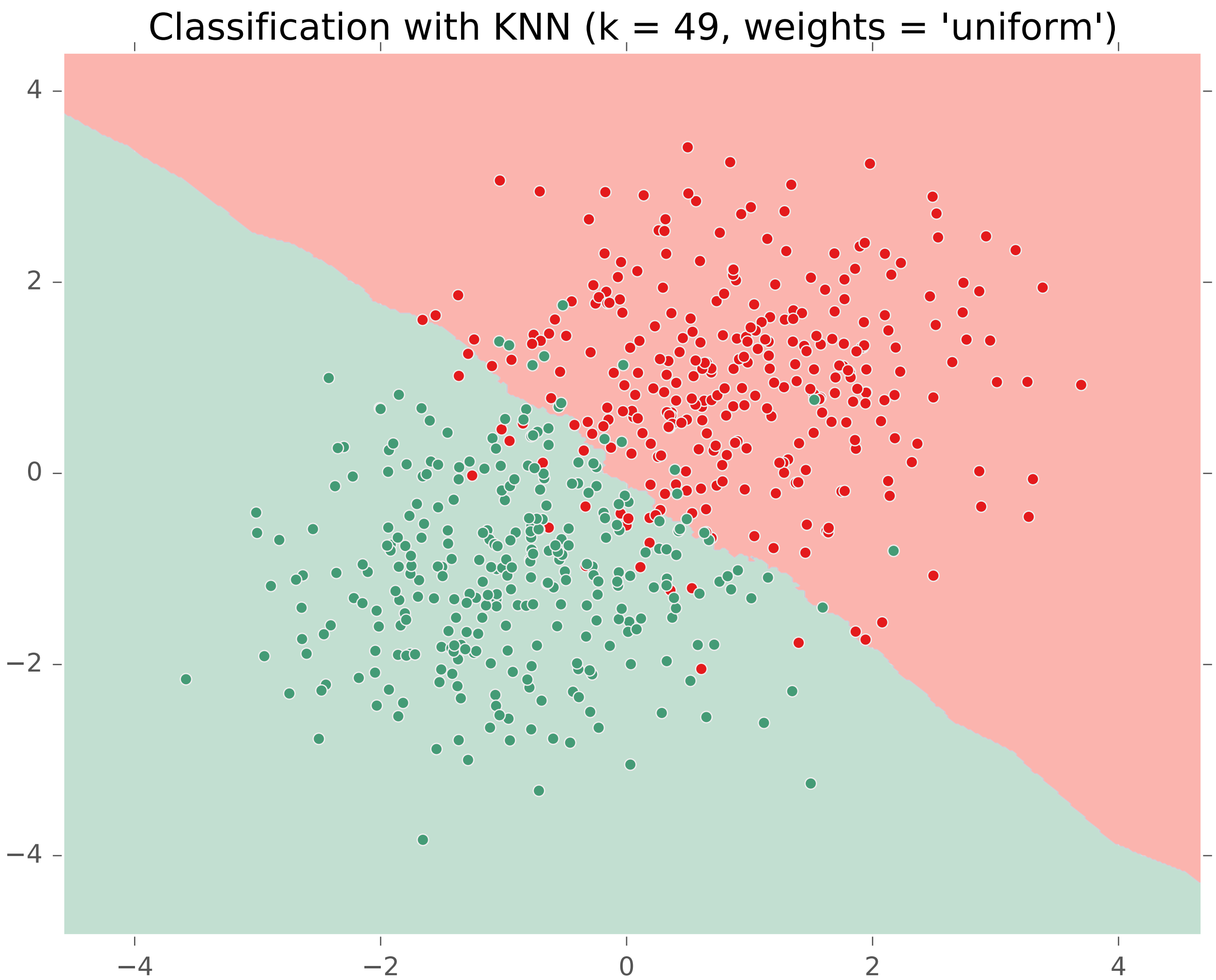
KNN on Gaussian Data



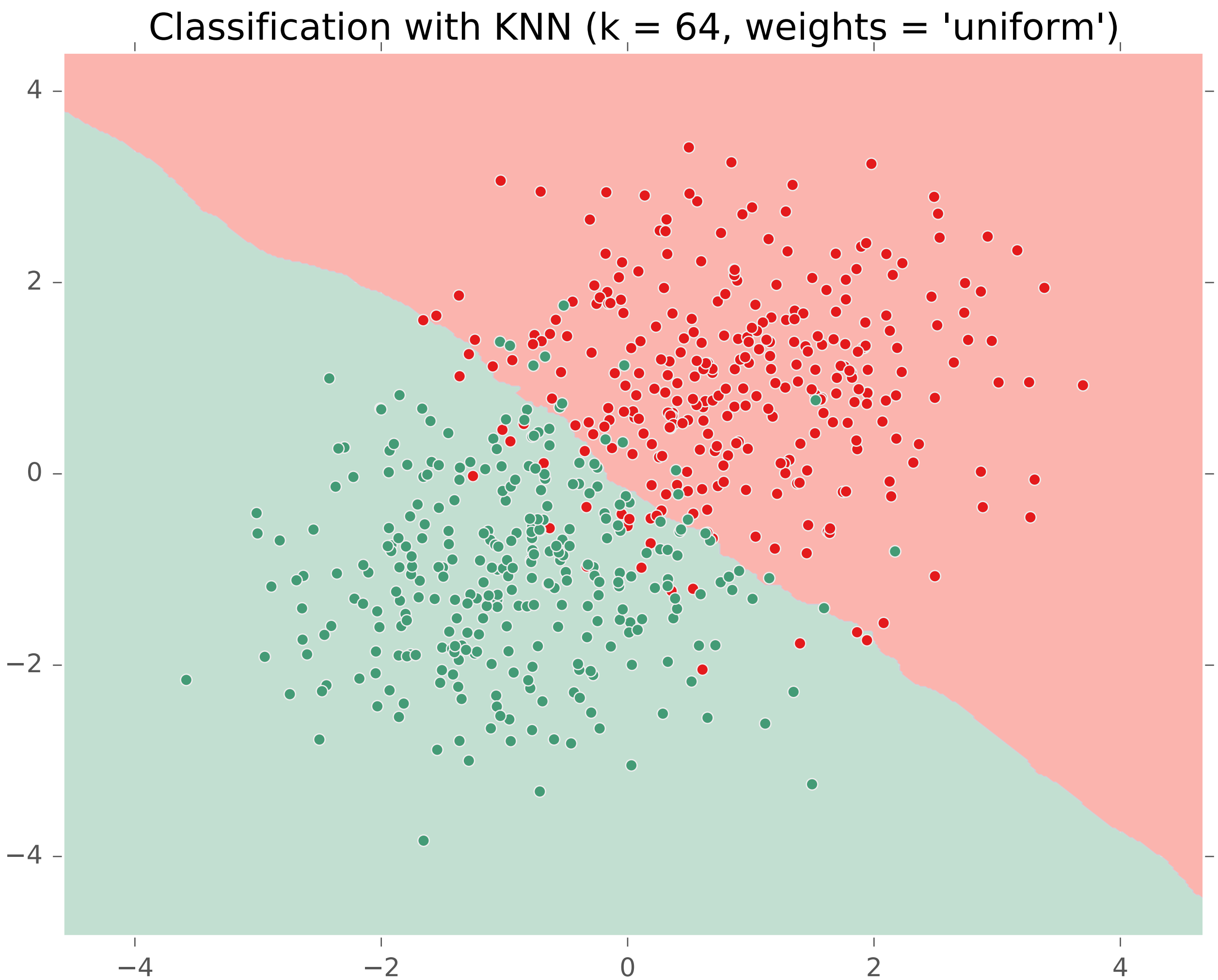
KNN on Gaussian Data



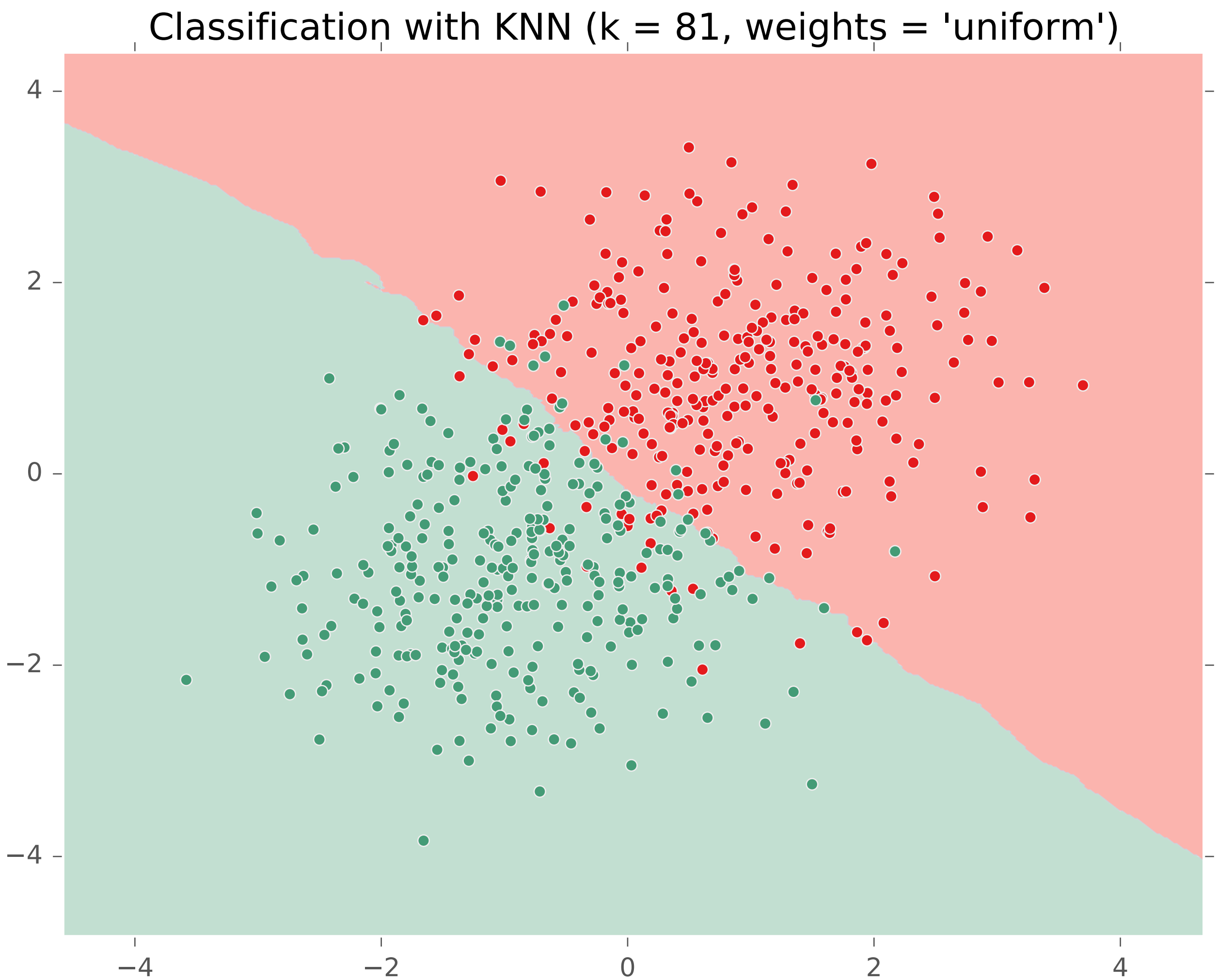
KNN on Gaussian Data



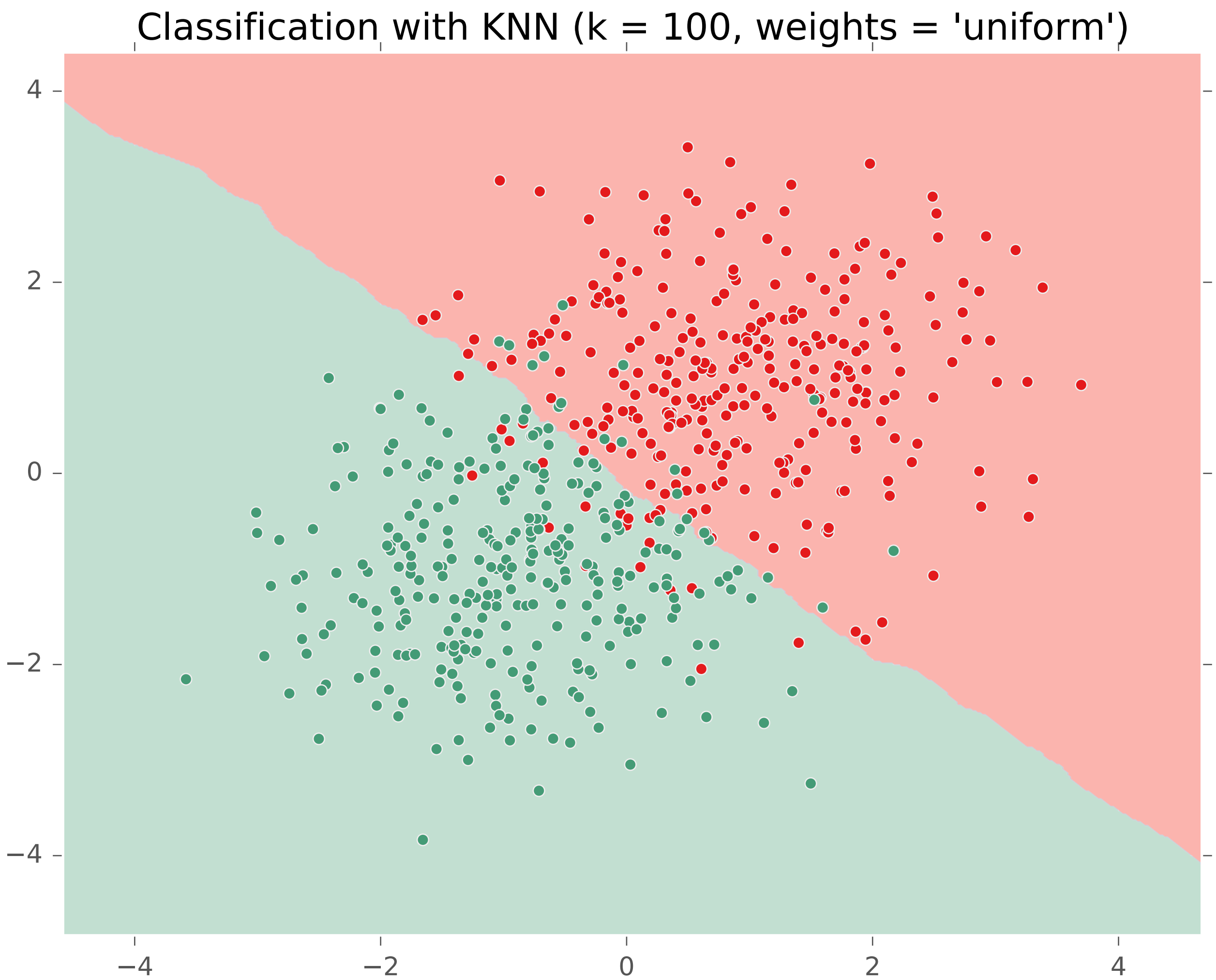
KNN on Gaussian Data



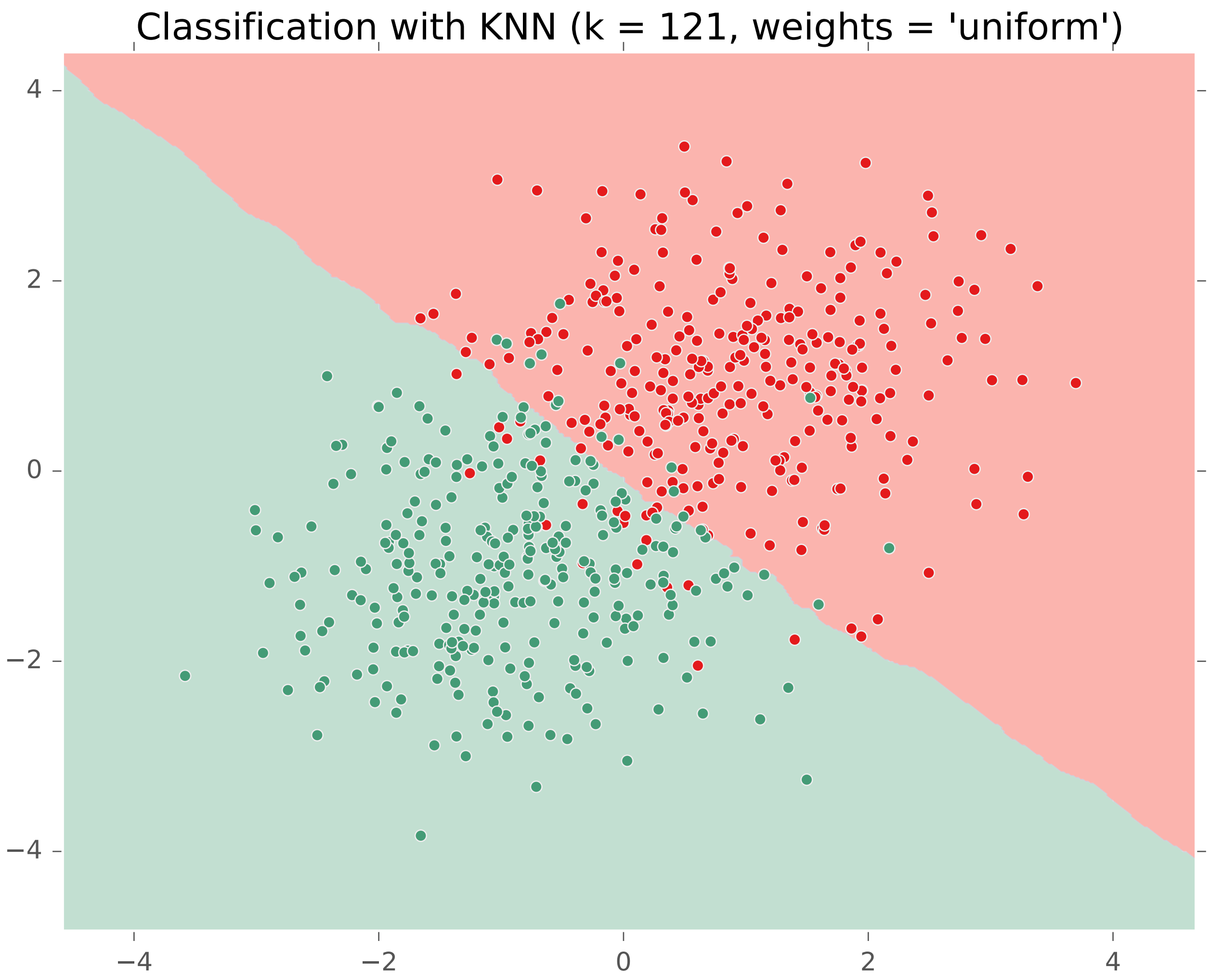
KNN on Gaussian Data



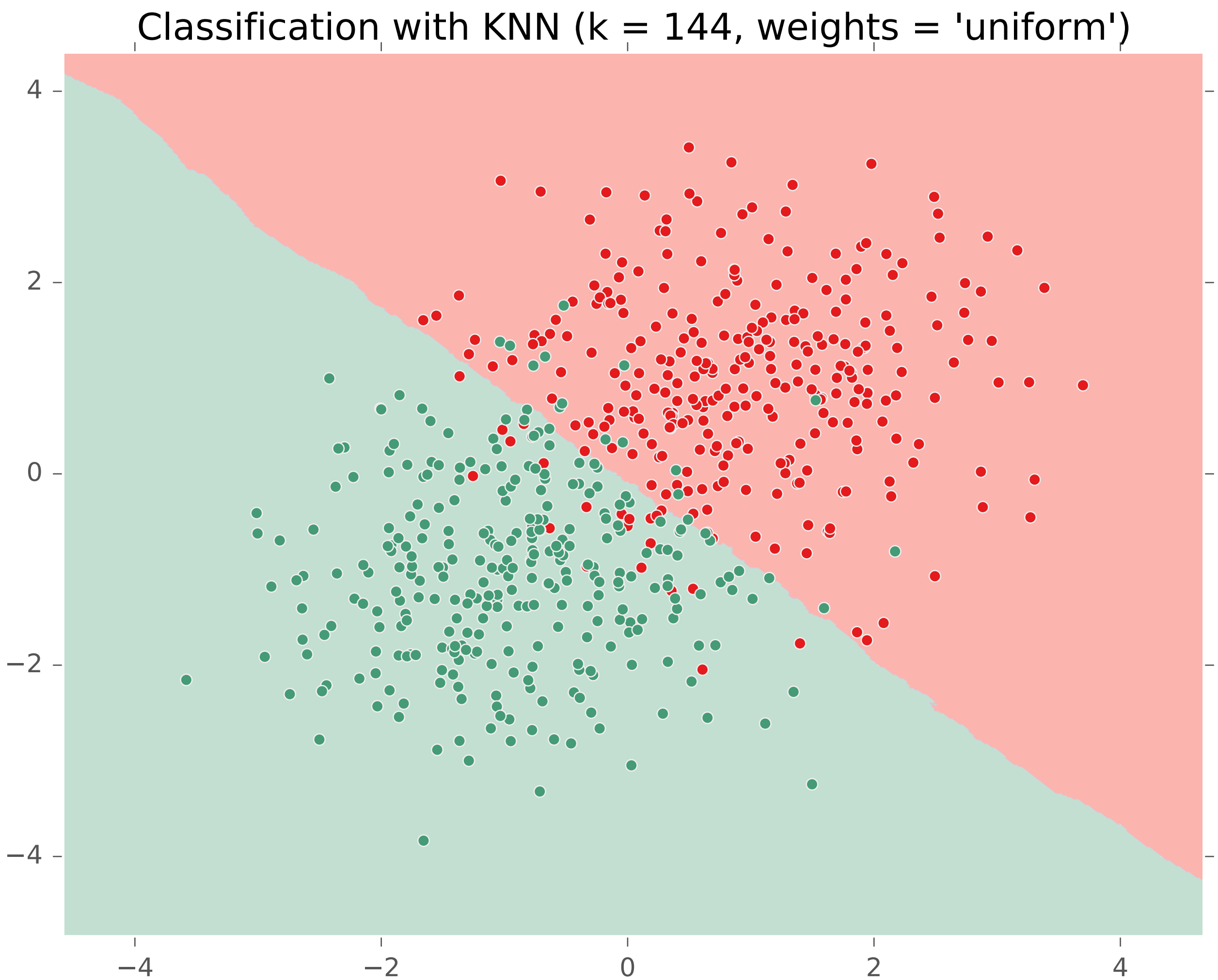
KNN on Gaussian Data



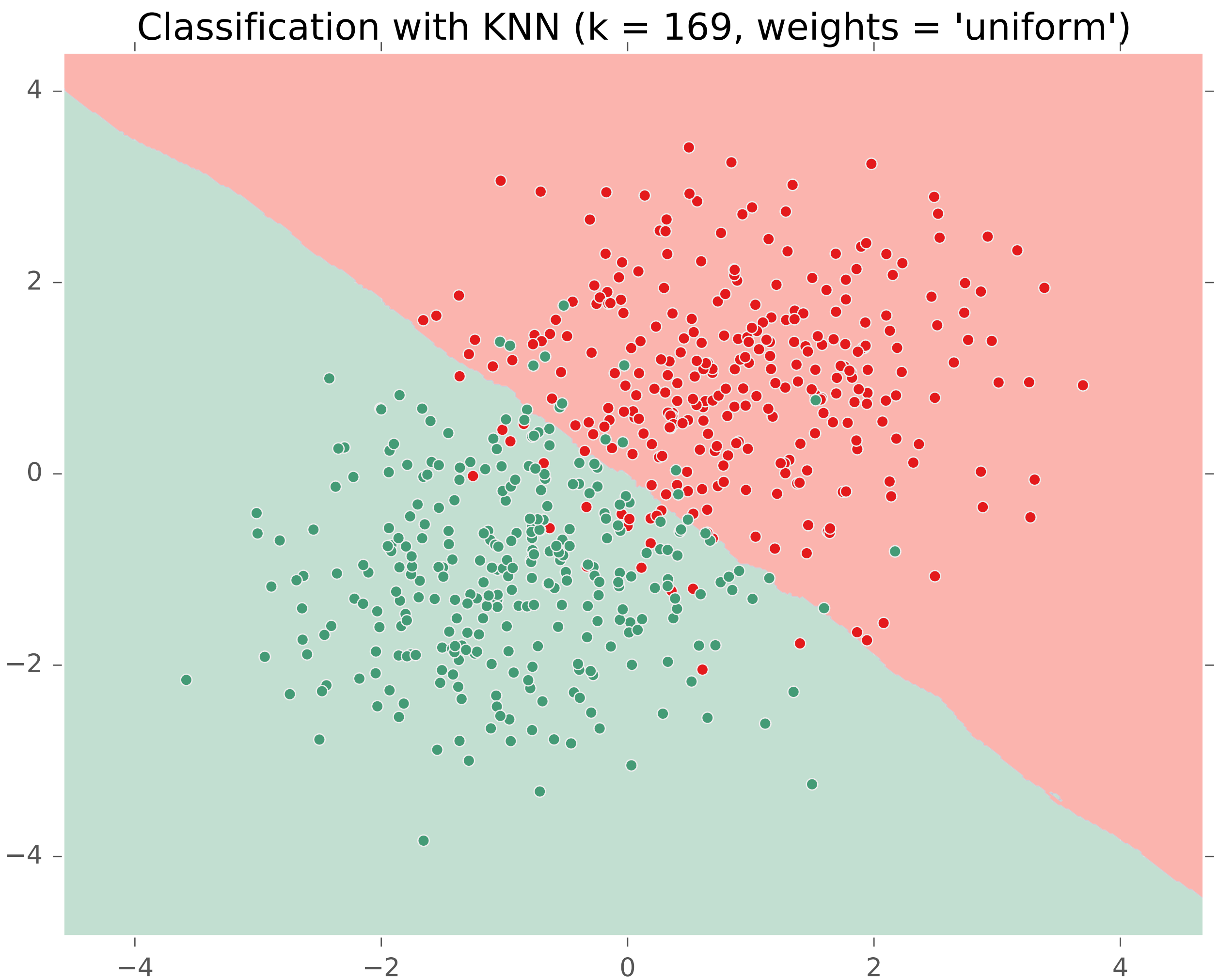
KNN on Gaussian Data



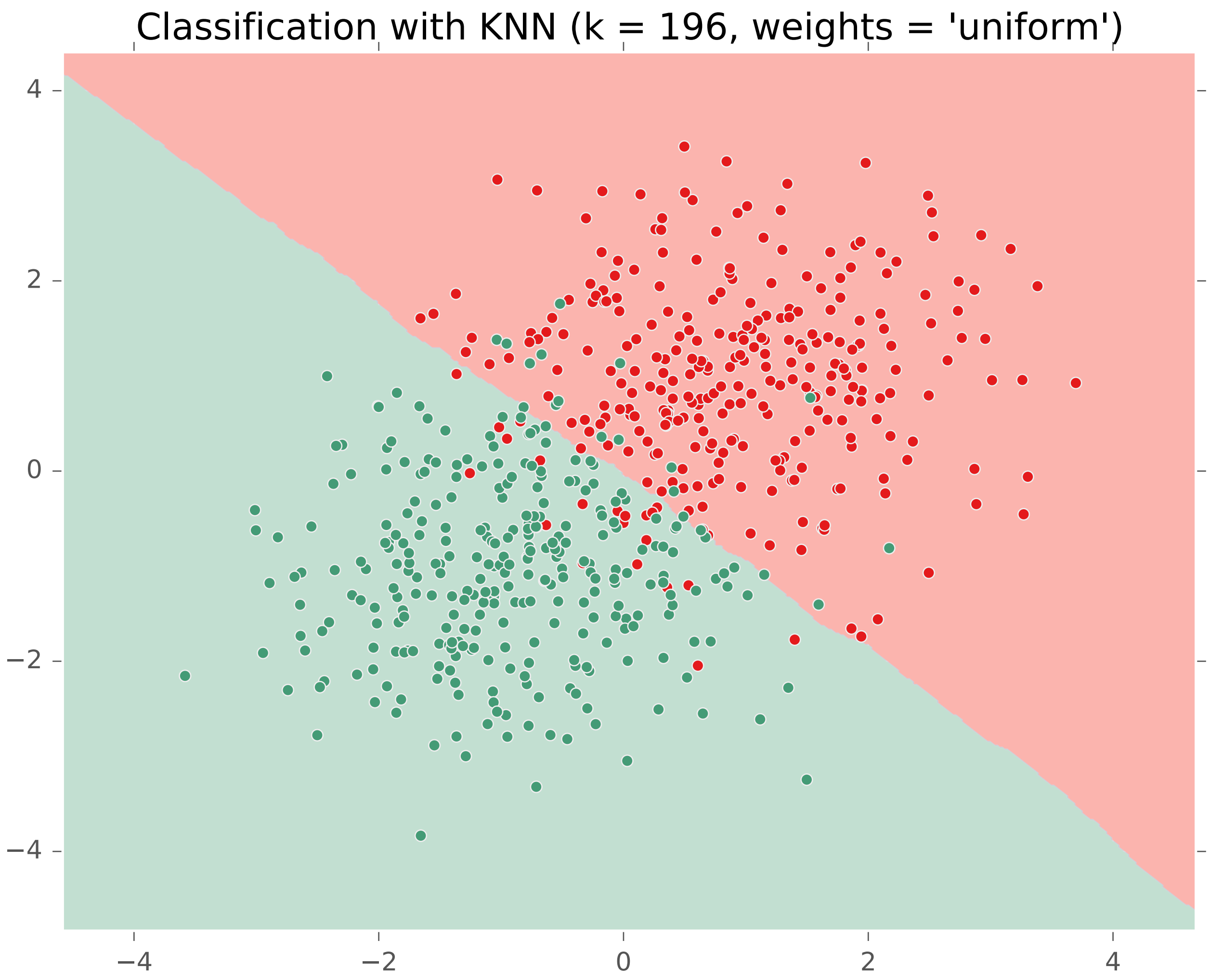
KNN on Gaussian Data



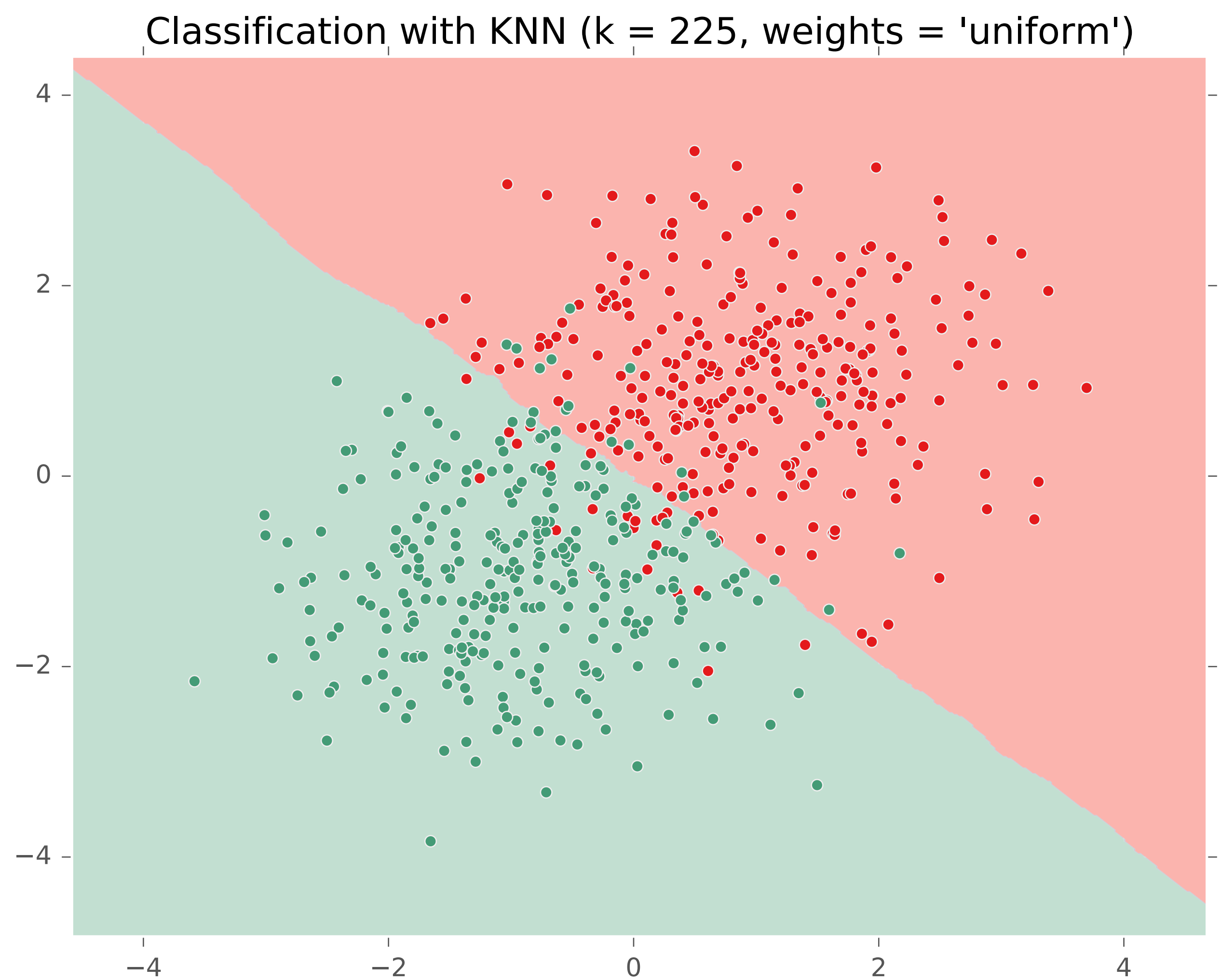
KNN on Gaussian Data



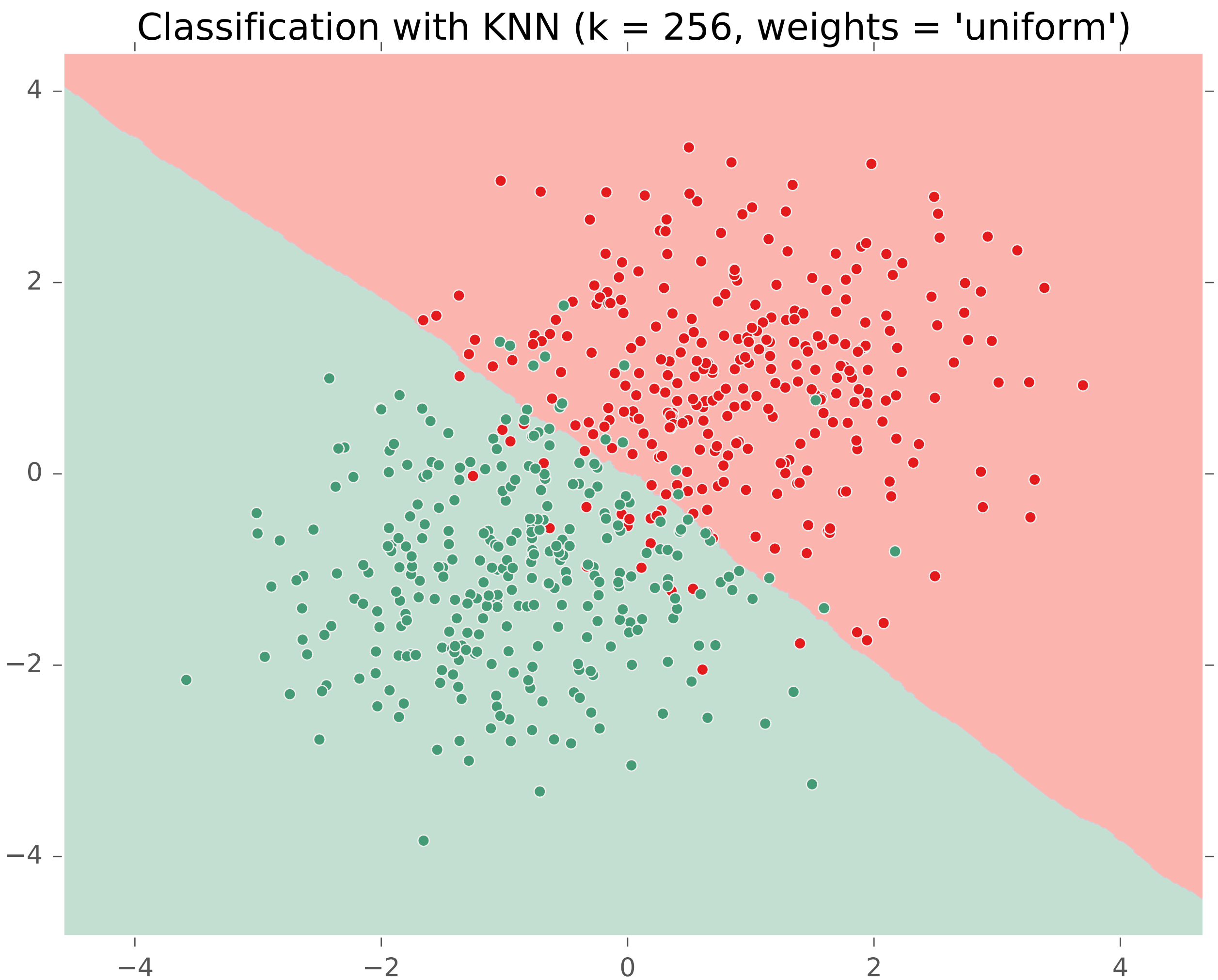
KNN on Gaussian Data



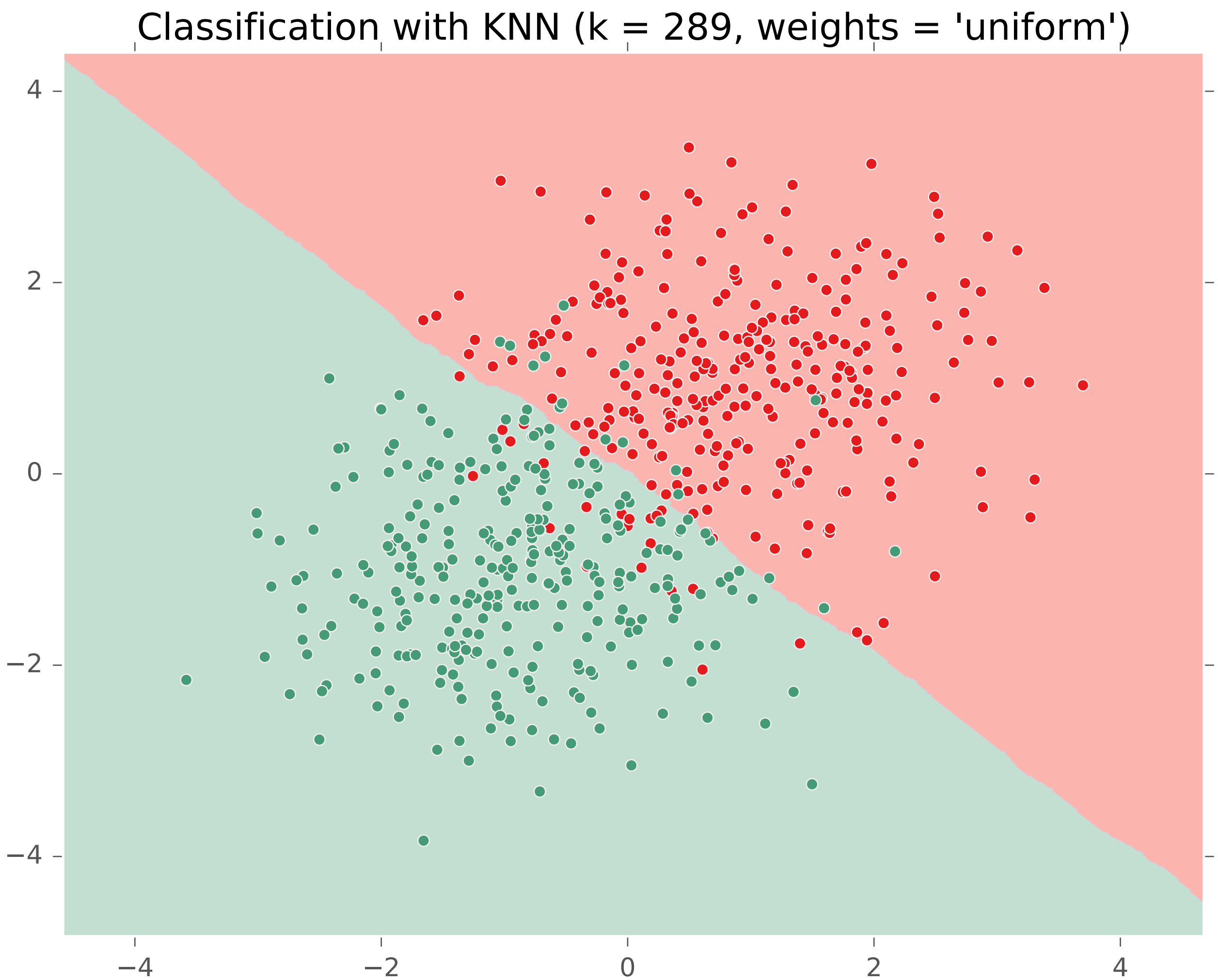
KNN on Gaussian Data



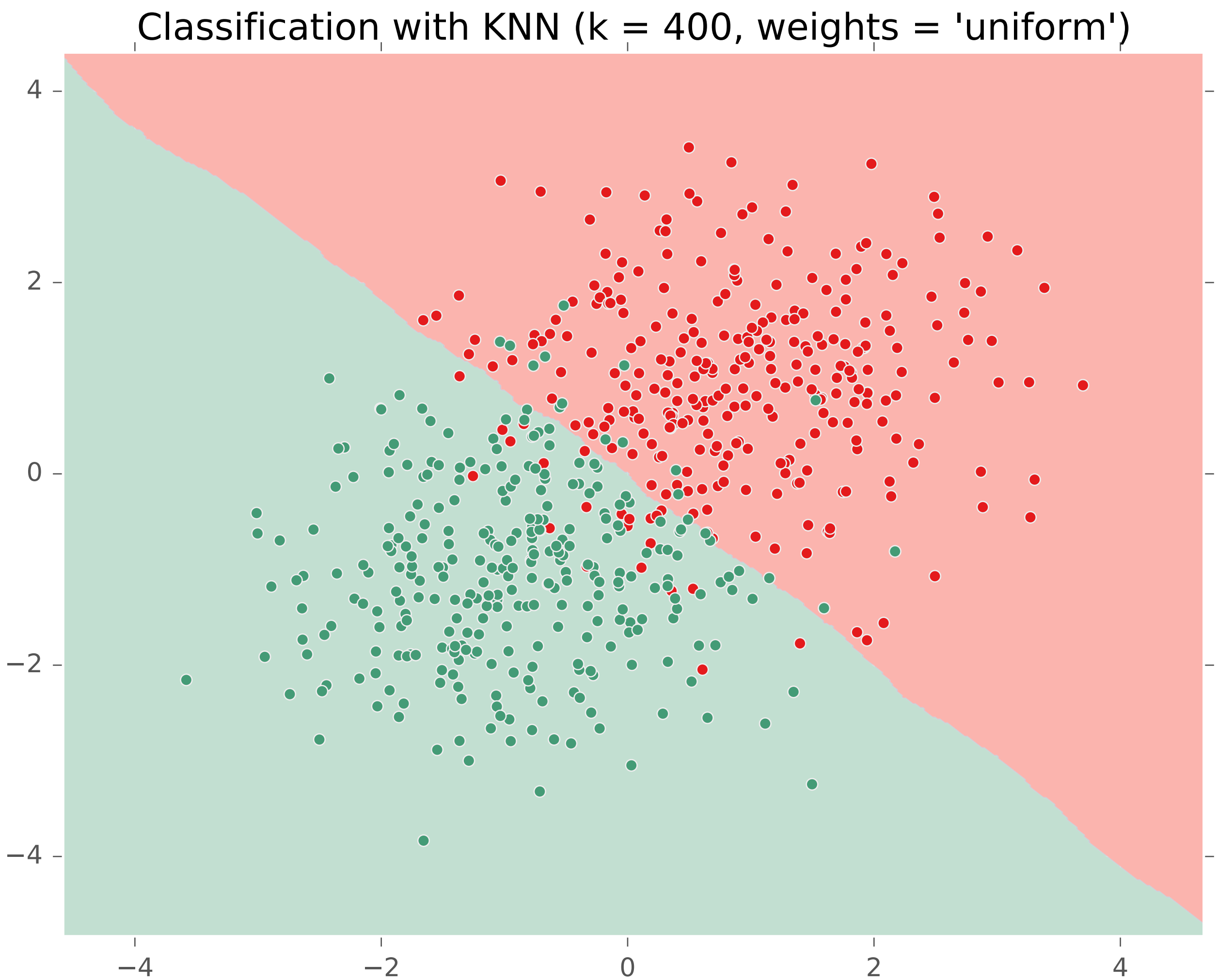
KNN on Gaussian Data



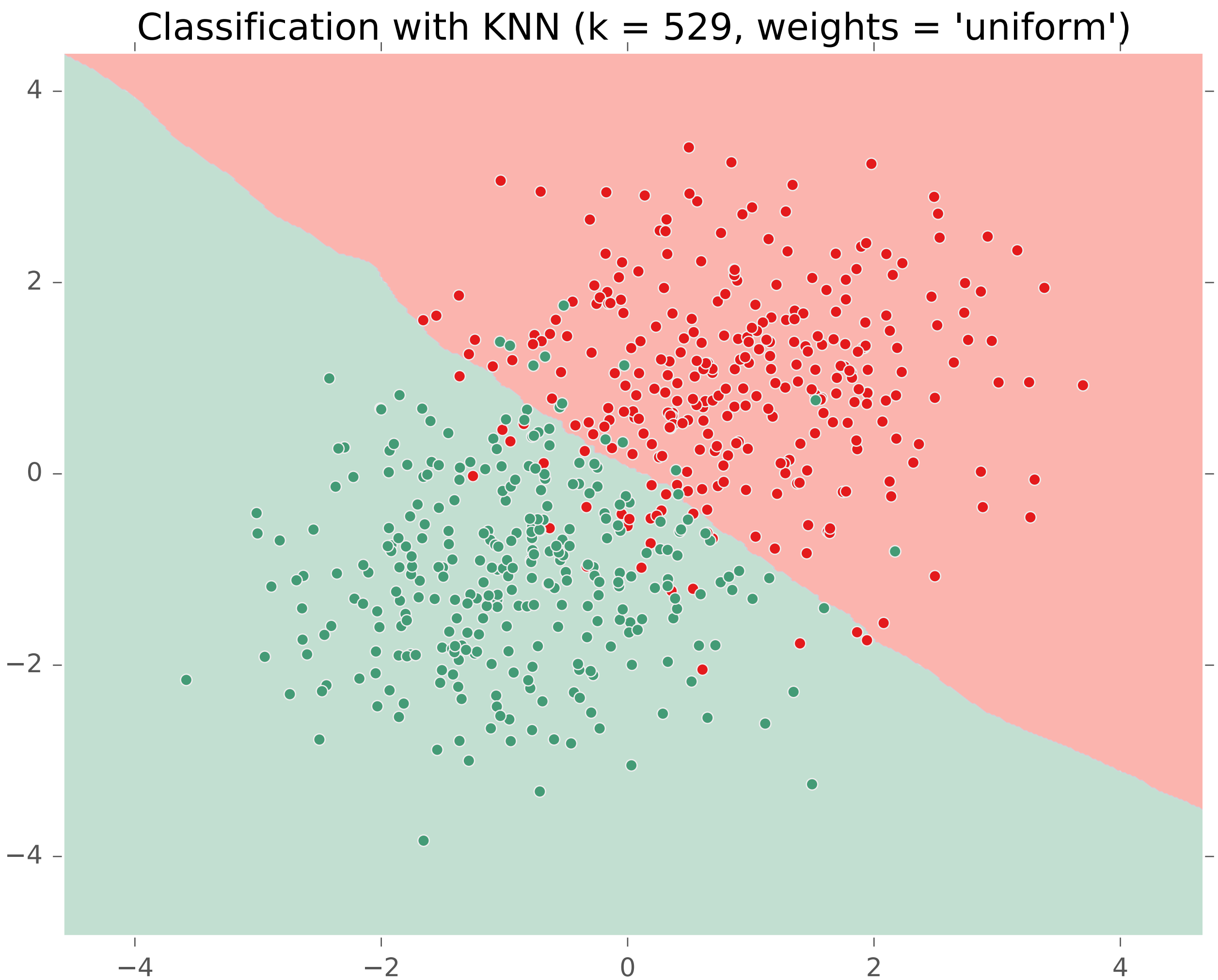
KNN on Gaussian Data



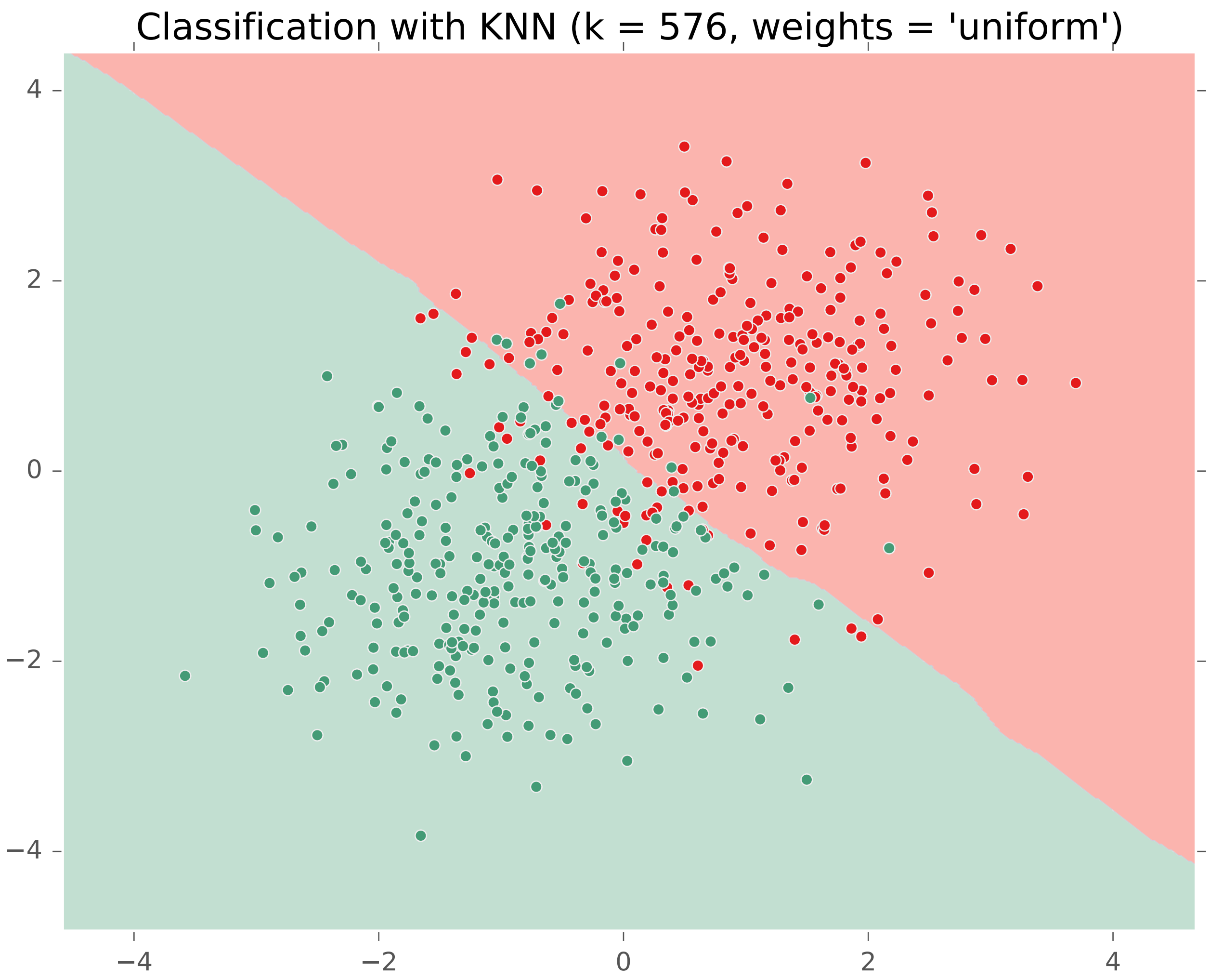
KNN on Gaussian Data



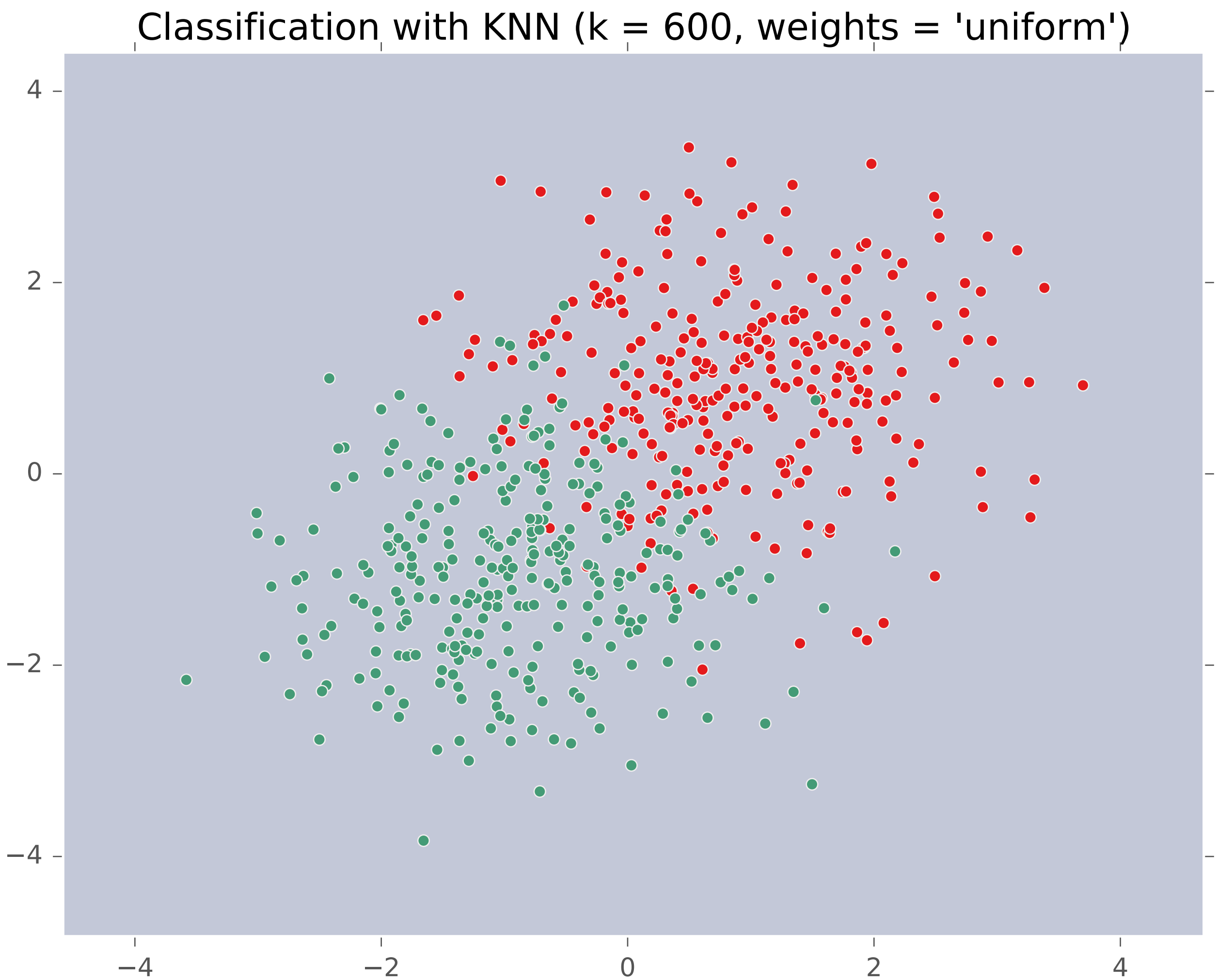
KNN on Gaussian Data



KNN on Gaussian Data



KNN on Gaussian Data



K-NEAREST NEIGHBORS

Questions

- How could k-Nearest Neighbors (KNN) be applied to **regression?**
- Can we do better than majority vote? (e.g. **distance-weighted** KNN)
- Where does the Cover & Hart (1967) **Bayes error rate bound** come from**?**

KNN Learning Objectives

You should be able to…

- Describe a dataset as points in a high dimensional space [CIML]
- Implement k-Nearest Neighbors with O(N) prediction
- Describe the inductive bias of a k-NN classifier and relate it to feature scale [a la. CIML]
- Sketch the decision boundary for a learning algorithm (compare k-NN and DT)
- State Cover & Hart (1967)'s large sample analysis of a nearest neighbor classifier
- Invent "new" k-NN learning algorithms capable of dealing with even k
- Explain computational and geometric examples of the curse of dimensionality

MODEL SELECTION

Model Selection

WARNING:

- In some sense, our discussion of model selection is premature.
- The models we have considered thus far are fairly simple.
- The models and the many decisions available to the data scientist wielding them will grow to be much more complex than what we've seen so far.

Model Selection

Statistics

- *Def*: a **model** defines the data generation process (i.e. a set or family of parametric probability distributions)
- *Def*: **model parameters** are the values that give rise to a particular probability distribution in the model family
- *Def*: **learning** (aka. estimation) is the process of finding the parameters that best fit the data
- *Def*: **hyperparameters** are the parameters of a prior distribution over parameters

Machine Learning

- *Def*: (loosely) a **model** defines the hypothesis space over which learning performs its search
- *Def*: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- *Def*: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select

Model Selection

Example: Decision Tree

- model = set of all possible trees, possibly restricted by some hyperparameters (e.g. max depth)
- parameters = structure of a specific decision tree
- learning algorithm = ID3, CART, etc.
- hyperparameters = max-depth, threshold for splitting criterion, etc.

Machine Learning

- *Def*: (loosely) a **model** defines the hypothesis space over which learning performs its search
- *Def*: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- *Def*: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select

Model Selection

Example: k-Nearest Neighbors

- model = set of all possible nearest neighbors classifiers
- parameters = none (KNN is an instance-based or non-parametric method)
- learning algorithm = for naïve setting, just storing the data
- hyperparameters = *k*, the number of neighbors to consider

Machine Learning

- *Def*: (loosely) a **model** defines the hypothesis space over which learning performs its search
- *Def*: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- *Def*: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select

Model Selection

Example: Perceptron

- model = set of all linear separators
- parameters = vector of weights (one for each feature)
- learning algorithm = mistake based updates to the parameters
- hyperparameters = none (unless using some variant such as averaged perceptron)

Machine Learning

- *Def*: (loosely) a **model** defines the hypothesis space over which learning performs its search
- *Def*: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- *Def*: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select

Model Selection

Statistics

- *Def*: a **model** defines the data generation pro... family of para... distributions)
- *Def*: **model pa**... values that giv... particular prob... distribution in...
- *Def*: **learnin**... aka. estimation) is the proce... finding the parame... that best fit the data
- *Def*: **hyperparameters** are the parameters of a prior distribution over parameters

If "learning" is all about picking the best **parameters** how do we pick the best **hyperparameters?**

Machine Learning

- *Def*: (loosely) a **model** defines the ...pace over which ...forms its search
- ...**arameters** are the ...ues or structure ...the learning algorithm ...e to a hypothesis
- ...**ning algorithm** defines the d...a-driven search over the hyp...esis space (i.e. search for goo... ...rameters)
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select

Model Selection

- Two *very* similar definitions:
  - *Def*: **model selection** is the process by which we choose the "best" model from among a set of candidates
  - *Def*: **hyperparameter optimization** is the process by which we choose the "best" hyperparameters from among a set of candidates **(could be called a special case of model selection)**

- **Both** assume access to a function capable of measuring the quality of a model

- **Both** are typically done "outside" the main training algorithm --- typically training is treated as a black box

Example of Hyperparameter Opt.

Chalkboard:

- Special cases of k-Nearest Neighbors
- Choosing k with validation data
- Choosing k with cross-validation

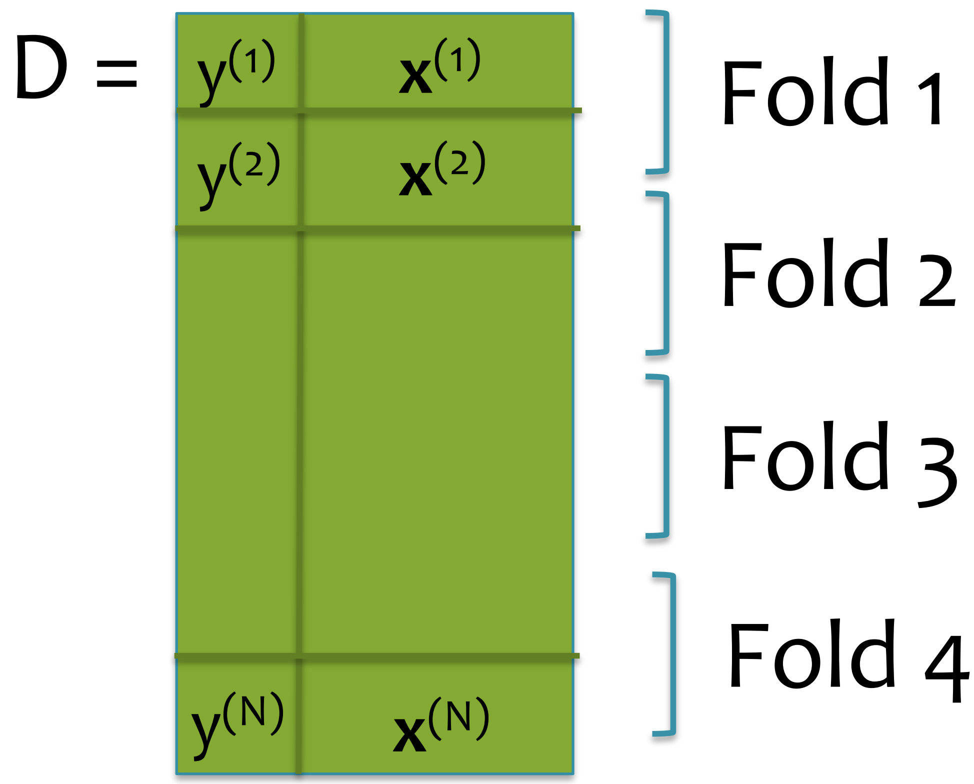
Cross-Validation

Cross validation is a method of estimating loss on held out data

Input: training data, learning algorithm, loss function (e.g. 0/1 error)

Output: an estimate of loss function on held-out data

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



Algorithm:

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 loss (*almost* equivalent to averaging loss over the folds)

Model Selection

WARNING (again):

- This section is only scratching the surface!
- Lots of methods for hyperparameter optimization: (to talk about later)
  - Grid search
  - Random search
  - Bayesian optimization
  - Graduate-student descent
  - …

Main Takeaway:

- Model selection / hyperparameter optimization is just another form of learning

Model Selection Learning Objectives

You should be able to…

- Plan an experiment that uses training, validation, and test datasets to predict the performance of a classifier on unseen data (without cheating)
- Explain the difference between (1) training error, (2) validation error, (3) cross-validation error, (4) test error, and (5) true error
- For a given learning technique, identify the model, learning algorithm, parameters, and hyperparamters
- Define "instance-based learning" or "nonparametric methods"
- Select an appropriate algorithm for optimizing (aka. learning) hyperparameters

THE PERCEPTRON ALGORITHM

Perceptron: History

Imagine you are trying to build a new machine learning technique… your name is Frank Rosenblatt… and the year is 1957

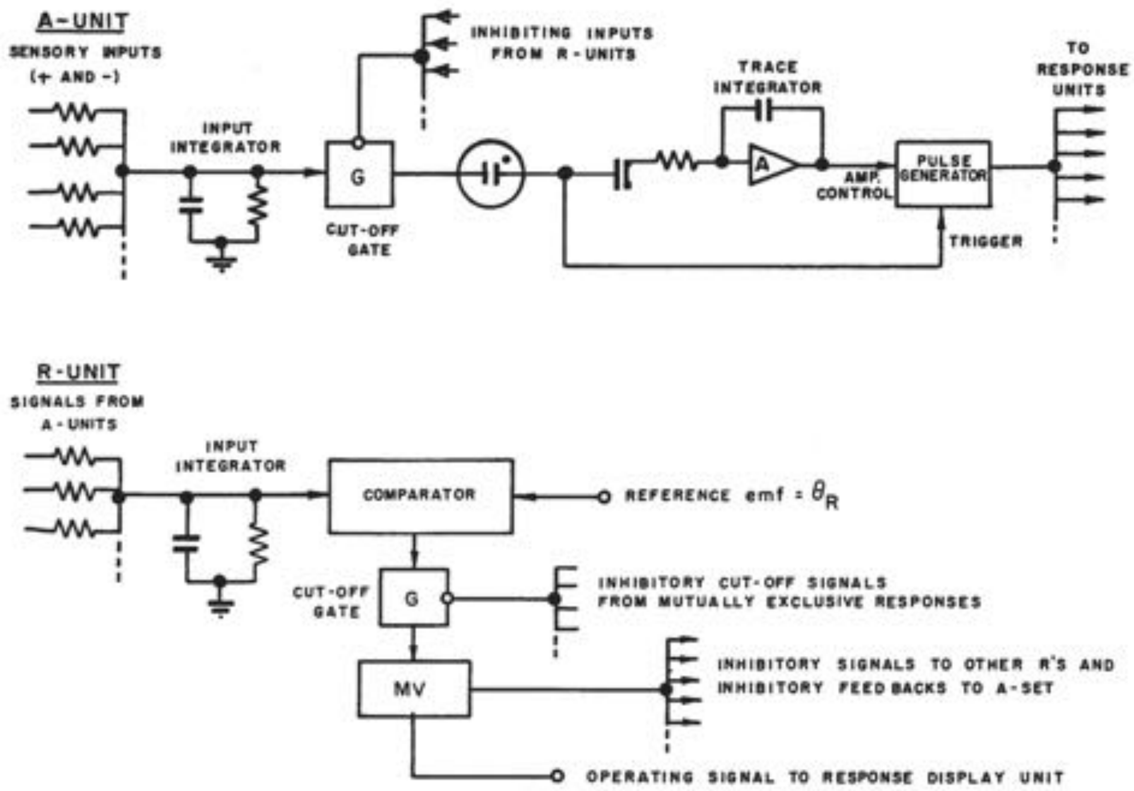
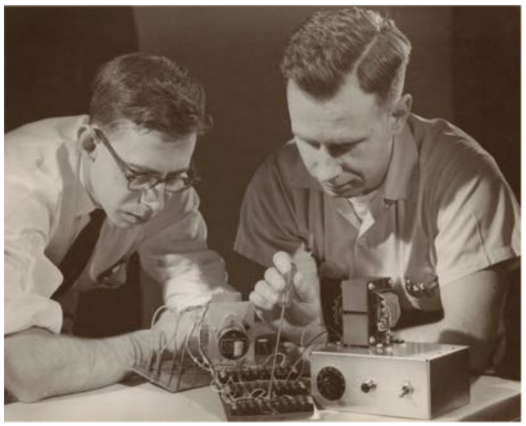
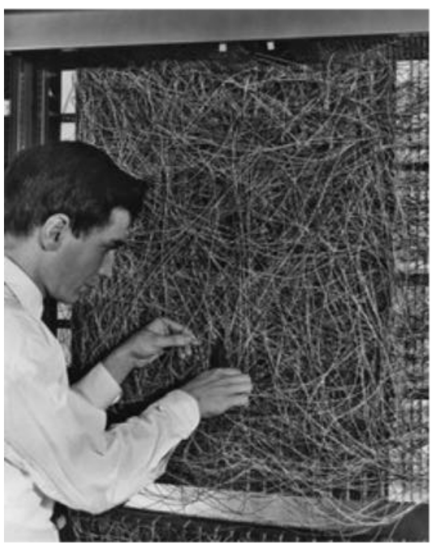


FIGURE 5
DESIGN OF TYPICAL UNITS

Perceptron: History

Imagine you are trying to build a new machine learning technique… your name is Frank Rosenblatt… and the year is 1957

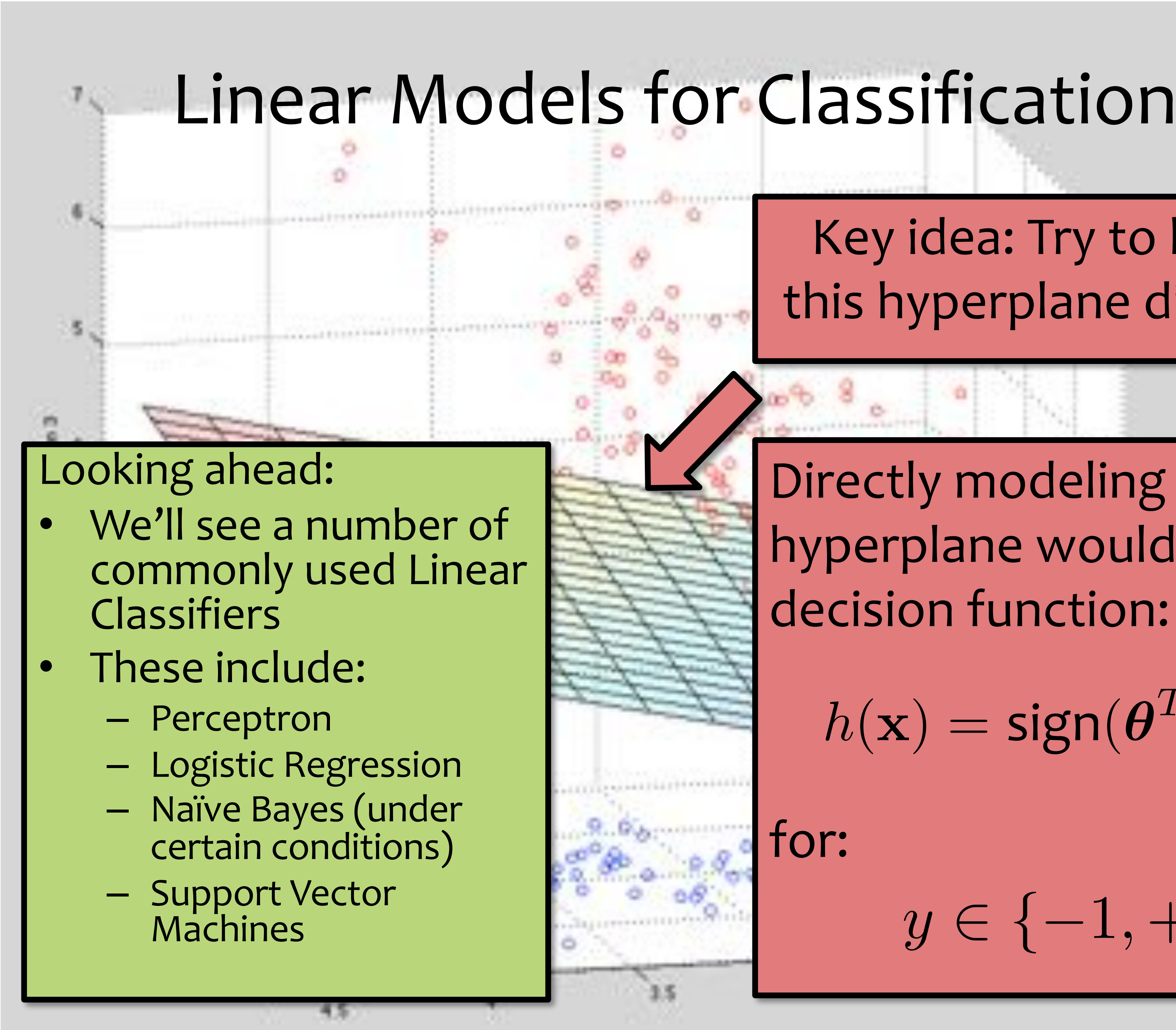




The New Yorker, December 6, 1958 P. 44

Talk story about the perceptron, a new electronic brain which hasn't been built, but which has been successfully simulated on the I.B.M. 704. Talk with Dr. Frank Rosenblatt, of the Cornell Aeronautical Laboratory, who is one of the two men who developed the prodigy; the other man is Dr. Marshall C. Yovits, of the Office of Naval Research, in Washington. Dr. Rosenblatt defined the perceptron as the first non-biological object which will achieve an organization o its external environment in a meaningful way. It interacts with its environment, forming concepts that have not been made ready for it by a human agent. If a triangle is held up, the perceptron's eye picks up the image & conveys it along a random succession of lines to the response units, where the image is registered. It can tell the difference betw. a cat and a dog, although it wouldn't be able to tell whether the dog was to theleft or right of the cat. Right now it is of no practical use, Dr. Rosenblatt conceded, but he said that one day it might be useful to send one into outer space to take in impressions for us.

Linear Models for Classification



Key idea: Try to learn this hyperplane directly

Looking ahead:

- We'll see a number of commonly used Linear Classifiers
- These include:
  - Perceptron
  - Logistic Regression
  - Naïve Bayes (under certain conditions)
  - Support Vector Machines

Directly modeling the hyperplane would use a decision function:

$$h(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{x})$$

for:

$$y \in \{-1, +1\}$$

Geometry

In-Class Exercise

Draw a picture of the region corresponding to:

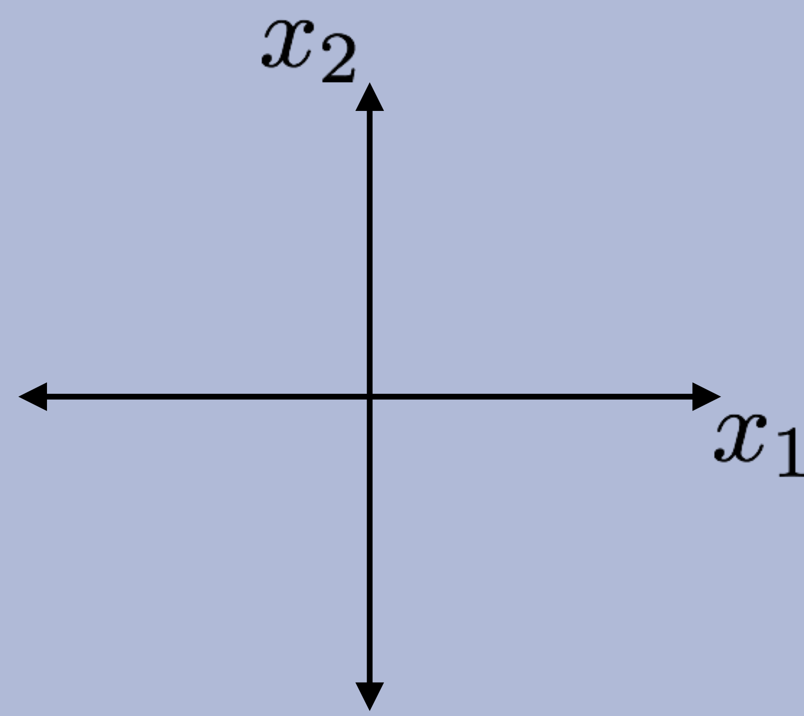
$$w_1 x_1 + w_2 x_2 + b > 0$$

where $w_1 = 2, w_2 = 3, b = 6$

Draw the vector

$\mathbf{w} = [w_1, w_2]$

Answer Here:



Visualizing Dot-Products

Chalkboard:

- – vector in 2D
- – line in 2D
- – adding a bias term
- – definition of orthogonality
- – vector projection
- – hyperplane definition
- – half-space definitions