K-Nearest Neighbors

Prof Wells

STA 295: Stat Learning

February 20th, 2024

Outline

In today's class, we will...

- Introduce the KNN algorithm as an example of a non-parametric model
- Discuss benefits and drawbacks of KNN
- Implement KNN in R

Section 1

K-Nearest Neighbors

Overview of Non-Parametric Methods

The overarching goal of supervised learning is to build a model to make predictions for a response Y based on predictors $X_1, \ldots, X_p$.

Overview of Non-Parametric Methods

The overarching goal of supervised learning is to build a model to make predictions for a response Y based on predictors $X_1, \ldots, X_p$.

- Often, we assume that there is a true relationship between Y and $X_1, \ldots, X_p$ given by an (unknown) function f:

$$Y = f(X_1, \ldots, X_p) + \epsilon$$

Overview of Non-Parametric Methods

The overarching goal of supervised learning is to build a model to make predictions for a response Y based on predictors $X_1, \ldots, X_p$.

- Often, we assume that there is a true relationship between Y and $X_1, \ldots, X_p$ given by an (unknown) function f:

$$Y = f(X_1, \ldots, X_p) + \epsilon$$

- **Parametric** methods propose that f belongs to a specific class of functions which are described by a small number of parameters; i.e.

$$f(x_1, x_2, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

Overview of Non-Parametric Methods

The overarching goal of supervised learning is to build a model to make predictions for a response Y based on predictors $X_1, \dots, X_p$.

- Often, we assume that there is a true relationship between Y and $X_1, \dots, X_p$ given by an (unknown) function f:

$$Y = f(X_1, \dots, X_p) + \epsilon$$

- **Parametric** methods propose that f belongs to a specific class of functions which are described by a small number of parameters; i.e.

$$f(x_1, x_2, \dots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

  - These methods then estimate the parameters using data:

$$\hat{f} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$$

Overview of Non-Parametric Methods

The overarching goal of supervised learning is to build a model to make predictions for a response Y based on predictors $X_1, \dots, X_p$.

- Often, we assume that there is a true relationship between Y and $X_1, \dots, X_p$ given by an (unknown) function f:
$$Y = f(X_1, \dots, X_p) + \epsilon$$
- **Parametric** methods propose that f belongs to a specific class of functions which are described by a small number of parameters; i.e.
$$f(x_1, x_2, \dots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$
  - These methods then estimate the parameters using data:
$$\hat{f} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$$
- **Non-Parametric** methods instead make only limited assumptions about the form of f

Overview of Non-Parametric Methods

The overarching goal of supervised learning is to build a model to make predictions for a response Y based on predictors $X_1, \ldots, X_p$.

- Often, we assume that there is a true relationship between Y and $X_1, \ldots, X_p$ given by an (unknown) function f:

$$Y = f(X_1, \ldots, X_p) + \epsilon$$

- **Parametric** methods propose that f belongs to a specific class of functions which are described by a small number of parameters; i.e.

$$f(x_1, x_2, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

  - These methods then estimate the parameters using data:

$$\hat{f} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$$

- **Non-Parametric** methods instead make only limited assumptions about the form of f
  - i.e. they may assume that f is continuous and bounded, but little else

Overview of Non-Parametric Methods

The overarching goal of supervised learning is to build a model to make predictions for a response Y based on predictors $X_1, \dots, X_p$.

- Often, we assume that there is a true relationship between Y and $X_1, \dots, X_p$ given by an (unknown) function f:

$$Y = f(X_1, \dots, X_p) + \epsilon$$

- **Parametric** methods propose that f belongs to a specific class of functions which are described by a small number of parameters; i.e.

$$f(x_1, x_2, \dots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

  - These methods then estimate the parameters using data:

$$\hat{f} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$$

- **Non-Parametric** methods instead make only limited assumptions about the form of f
  - i.e. they may assume that f is continuous and bounded, but little else
  - They instead estimate the outputs of f without being "too wiggly"

Overview of Non-Parametric Methods

The overarching goal of supervised learning is to build a model to make predictions for a response Y based on predictors $X_1, \ldots, X_p$.

- Often, we assume that there is a true relationship between Y and $X_1, \ldots, X_p$ given by an (unknown) function f:

$$Y = f(X_1, \ldots, X_p) + \epsilon$$

- **Parametric** methods propose that f belongs to a specific class of functions which are described by a small number of parameters; i.e.

$$f(x_1, x_2, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

  - These methods then estimate the parameters using data:

$$\hat{f} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$$

- **Non-Parametric** methods instead make only limited assumptions about the form of f
  - i.e. they may assume that f is continuous and bounded, but little else
  - They instead estimate the outputs of f without being "too wiggly"
  - Often, they do not provide an estimate for the function f, only its outputs

K-Nearest Neighbors

- K-Nearest Neighbors (KNN) is an example of a non-parametric supervised learning method

K-Nearest Neighbors

- K-Nearest Neighbors (KNN) is an example of a non-parametric supervised learning method
  - It is notable for its intuitiveness, flexibility, and simplicity, as well as its quick model building time

K-Nearest Neighbors

- K-Nearest Neighbors (KNN) is an example of a non-parametric supervised learning method
  - It is notable for its intuitiveness, flexibility, and simplicity, as well as its quick model building time
  - However, it lacks the structure of linear regression, meaning it provides little information about relationships between variables (i.e. it is a *predictive*, but not *explanatory* model)

K-Nearest Neighbors

- K-Nearest Neighbors (KNN) is an example of a non-parametric supervised learning method
    - It is notable for its intuitiveness, flexibility, and simplicity, as well as its quick model building time
    - However, it lacks the structure of linear regression, meaning it provides little information about relationships between variables (i.e. it is a *predictive*, but not *explanatory* model)
    - KNN can be used for both regression and classification tasks, as well as some unsupervised tasks

K-Nearest Neighbors

- K-Nearest Neighbors (KNN) is an example of a non-parametric supervised learning method
  - It is notable for its intuitiveness, flexibility, and simplicity, as well as its quick model building time
  - However, it lacks the structure of linear regression, meaning it provides little information about relationships between variables (i.e. it is a *predictive*, but not *explanatory* model)
  - KNN can be used for both regression and classification tasks, as well as some unsupervised tasks
- The algorithm works by assuming the response value of a variable tends to be similar among observations that are similar.
  - What we mean by "similar" will be made formal later, but is a source of problem for the KNN method

KNN Algorithm

1. Divide data into training and test sets

KNN Algorithm

1. Divide data into training and test sets
2. Choose a positive integer K, representing the number of neighbors to be considered.

KNN Algorithm

1. Divide data into training and test sets
2. Choose a positive integer K, representing the number of neighbors to be considered.
3. To make a prediction at a *test* observation x_0, identify the K points in the *training* set whose predictor values are "closest" to the predictor values of x_0. Call this set of neighbors $\mathcal{N}_0$

KNN Algorithm

1. Divide data into training and test sets
2. Choose a positive integer K, representing the number of neighbors to be considered.
3. To make a prediction at a *test* observation x_0, identify the K points in the *training* set whose predictor values are "closest" to the predictor values of x_0. Call this set of neighbors $\mathcal{N}_0$
4. Predict the response $\hat{y}_0$ for x_0 to be the average value of the responses among hte neighbor set:

$$\hat{y}_0 = \frac{1}{K} \sum_{i \in \mathcal{N}_0} y_i$$

KNN Algorithm

1. Divide data into training and test sets
2. Choose a positive integer K, representing the number of neighbors to be considered.
3. To make a prediction at a *test* observation x_0, identify the K points in the *training* set whose predictor values are "closest" to the predictor values of x_0. Call this set of neighbors $\mathcal{N}_0$
4. Predict the response $\hat{y}_0$ for x_0 to be the average value of the responses among hte neighbor set:
$$\hat{y}_0 = \frac{1}{K} \sum_{i \in \mathcal{N}_0} y_i$$
5. Repeat steps 3 and 4 for all points in the *test* set.

KNN in Pictures

- Suppose we want to predict the value of a quantitative response based on two quantitative predictors X_1 and X_2.

KNN in Pictures

- Suppose we want to predict the value of a quantitative response based on two quantitative predictors X_1 and X_2.
- We have a training set of 30 observations, and can plot each observation in 2D **predictor space**, where the horizontal axis is X_1 and the vertical axis is X_2.

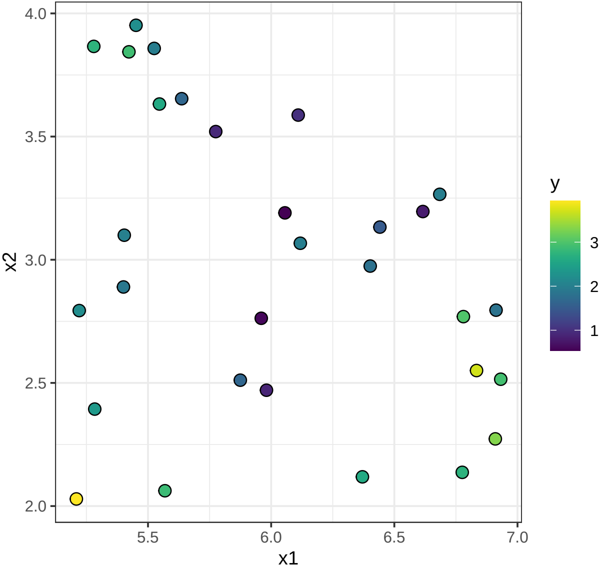
KNN in Pictures

- Suppose we want to predict the value of a quantitative response based on two quantitative predictors X_1 and X_2.
- We have a training set of 30 observations, and can plot each observation in 2D **predictor space**, where the horizontal axis is X_1 and the vertical axis is X_2.
- We color points according to the value of the response variable Y

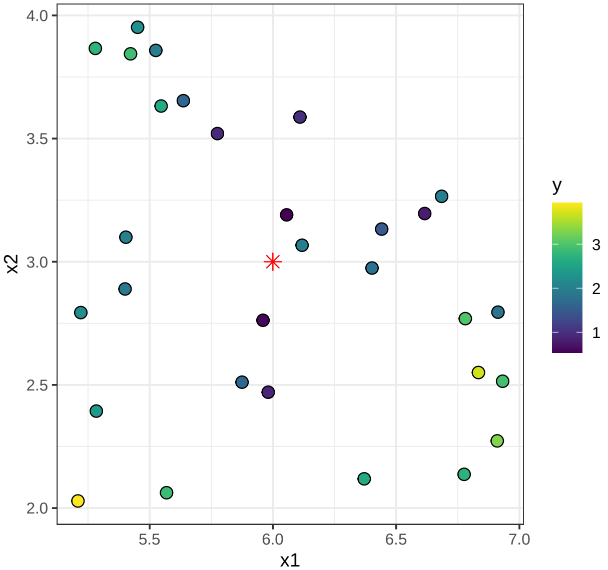
KNN in Pictures

- Suppose we want to predict the value of a quantitative response based on two quantitative predictors X_1 and X_2.
- We have a training set of 30 observations, and can plot each observation in 2D **predictor space**, where the horizontal axis is X_1 and the vertical axis is X_2.
- We color points according to the value of the response variable Y
- We have a new point (*) for which we know the values of its predictors, and wish to estimate the value of its response.

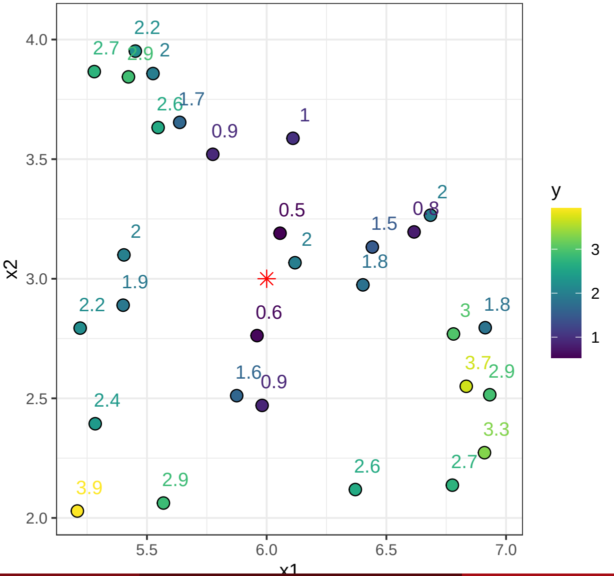
KNN in Pictures



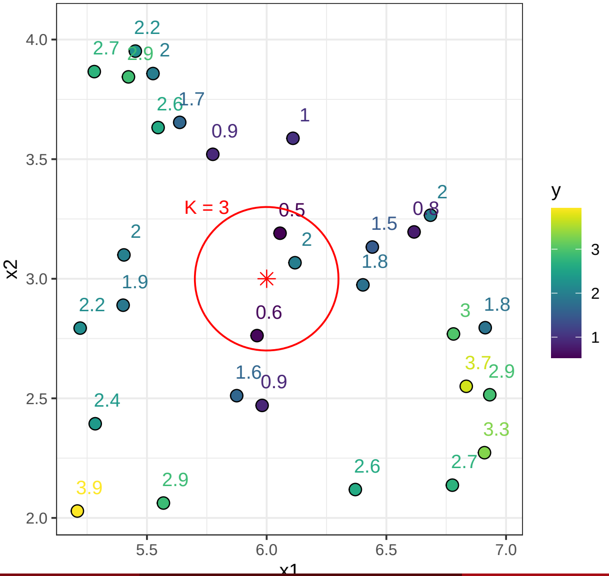
KNN in Pictures



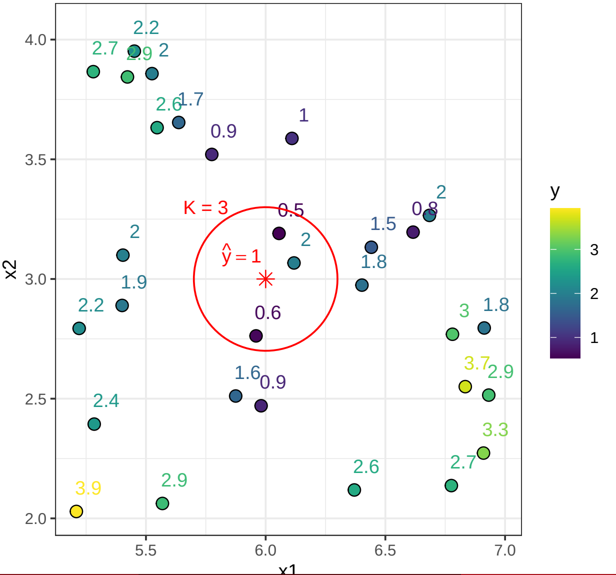
KNN in Pictures



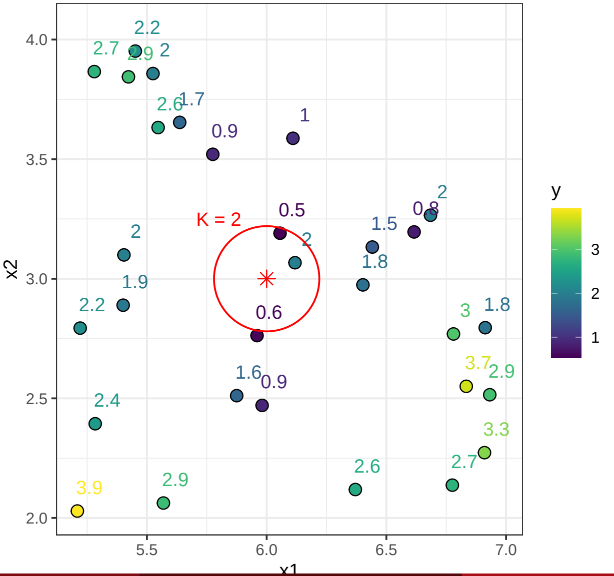
KNN in Pictures



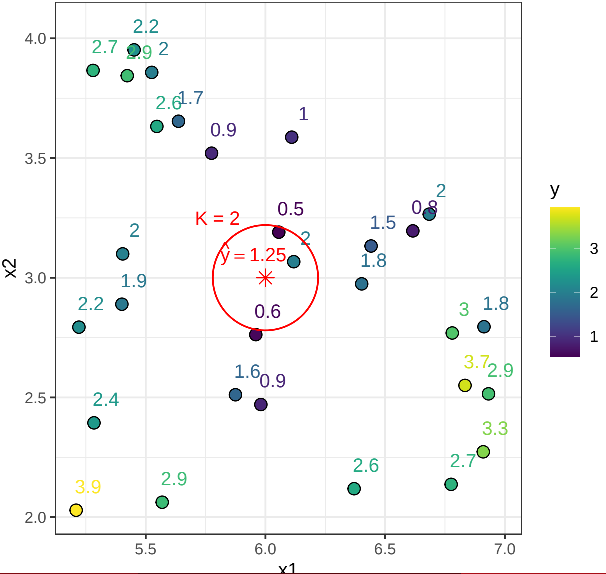
KNN in Pictures



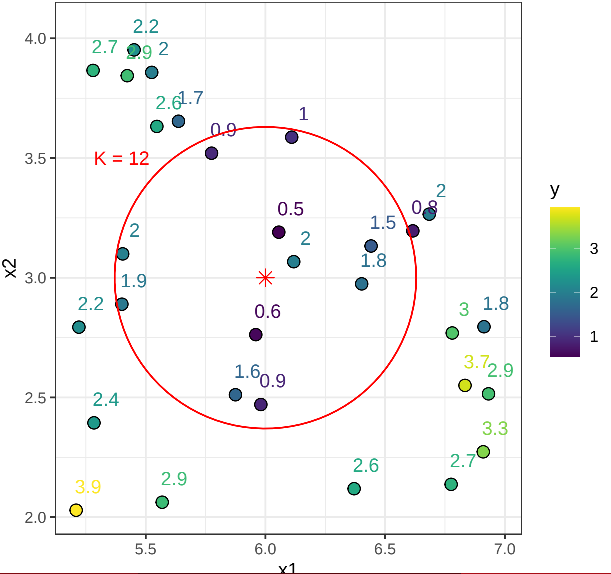
KNN in Pictures



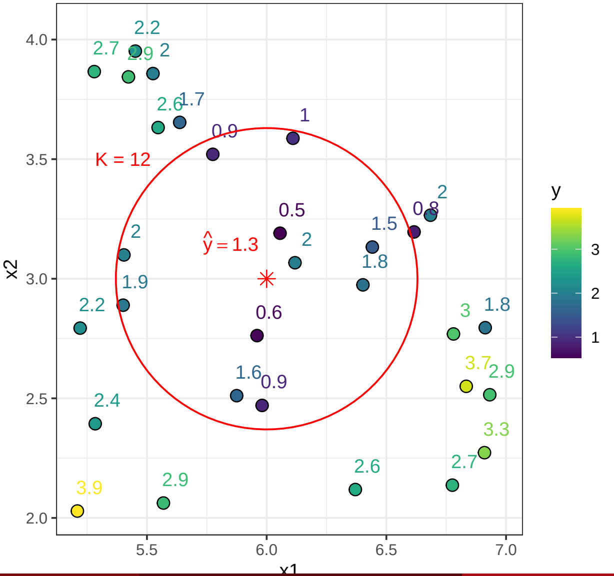
KNN in Pictures



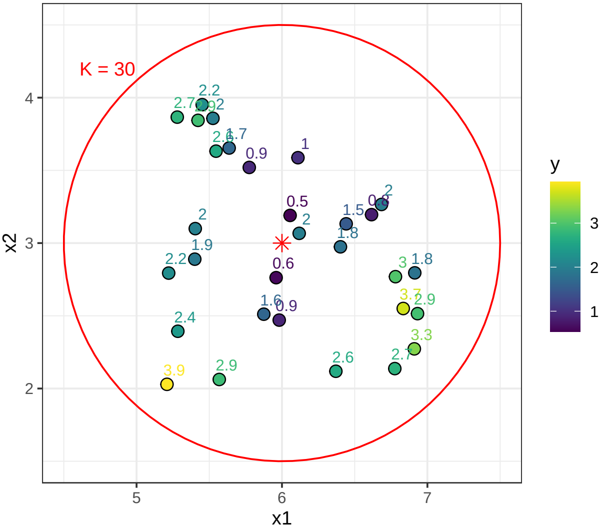
KNN in Pictures



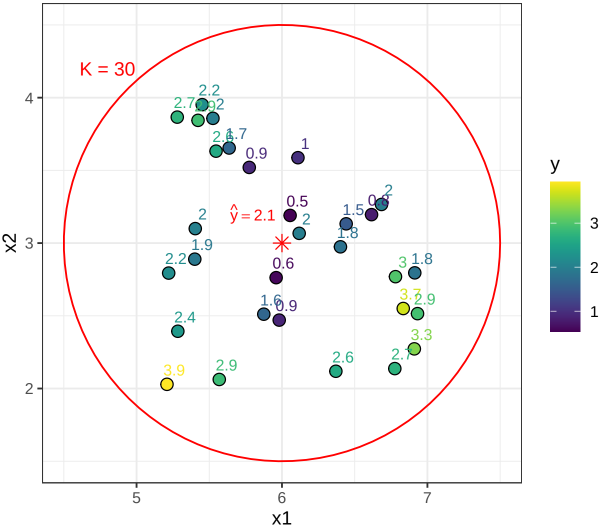
KNN in Pictures



KNN in Pictures

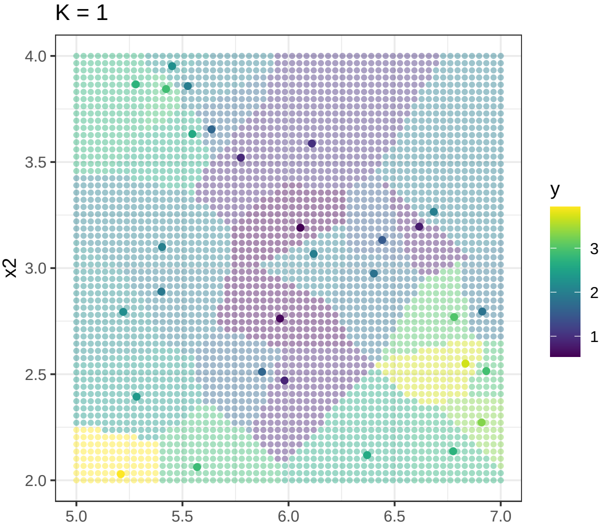


KNN in Pictures



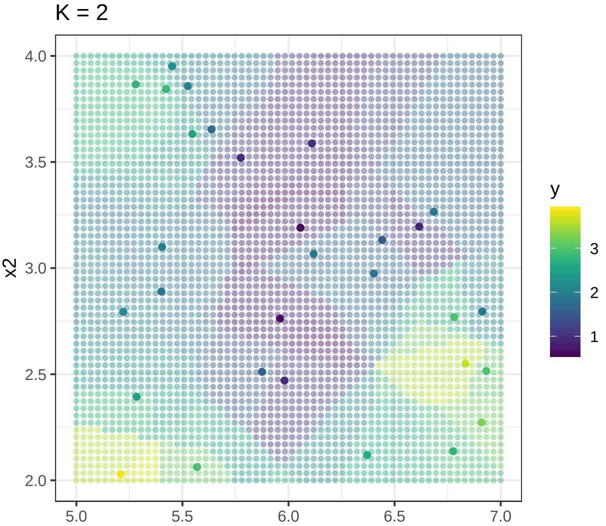
K=1

Here are the KNN surfaces for a variety of values of K.



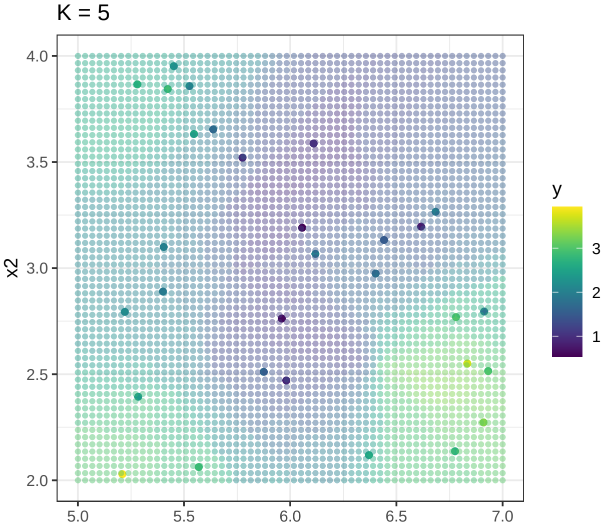
K=2

Here are the KNN surfaces for a variety of values of K.



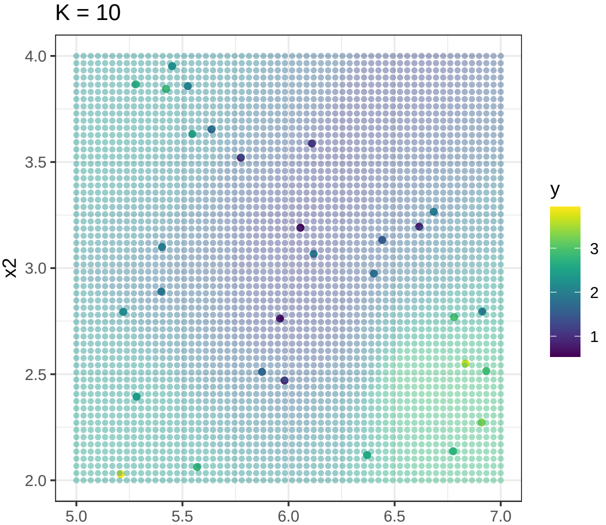
K=5

Here are the KNN surfaces for a variety of values of K.



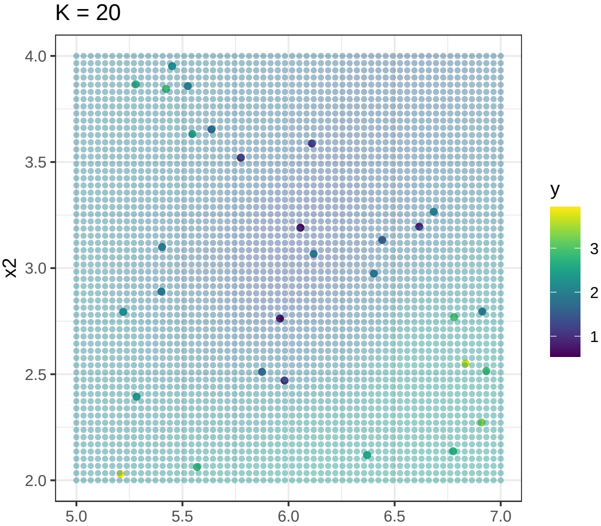
K=10

Here are the KNN surfaces for a variety of values of K.



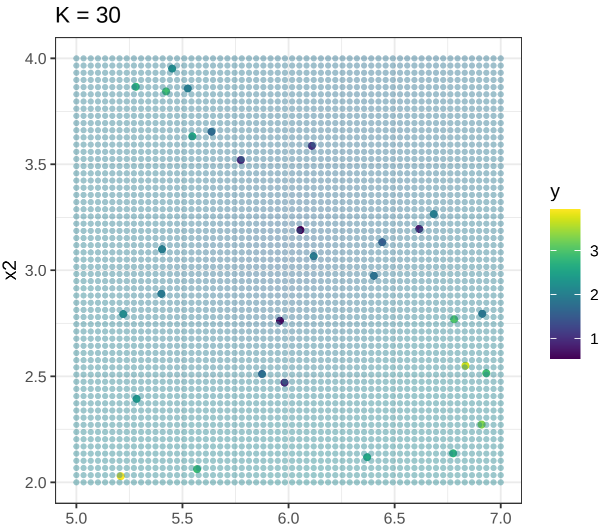
K=20

Here are the KNN surfaces for a variety of values of K.



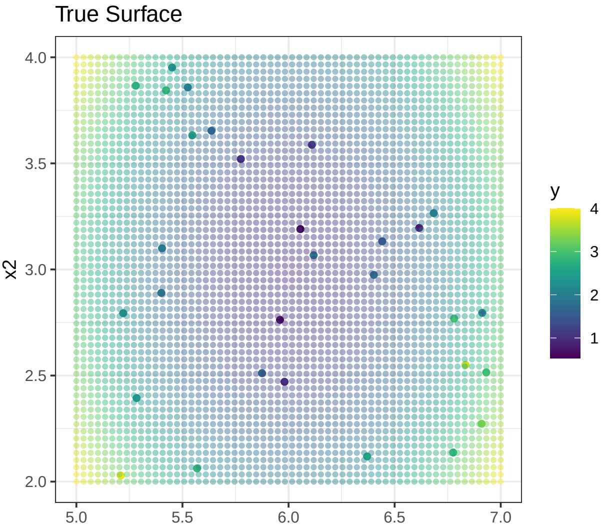
K=30

Here are the KNN surfaces for a variety of values of K.



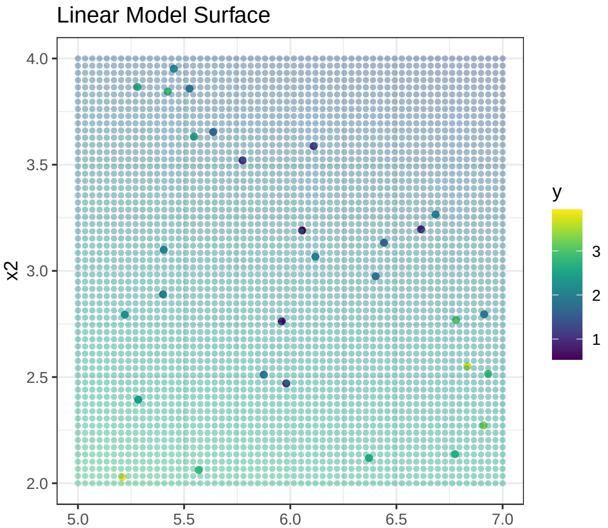
True Surface

Here is the true surface describing $Y = f(X_1, X_2)$:



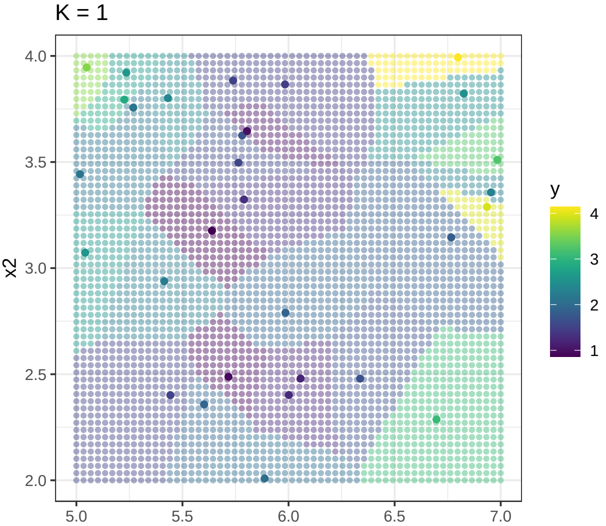
Linear Model

Here is the true surface described by the linear model $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$:



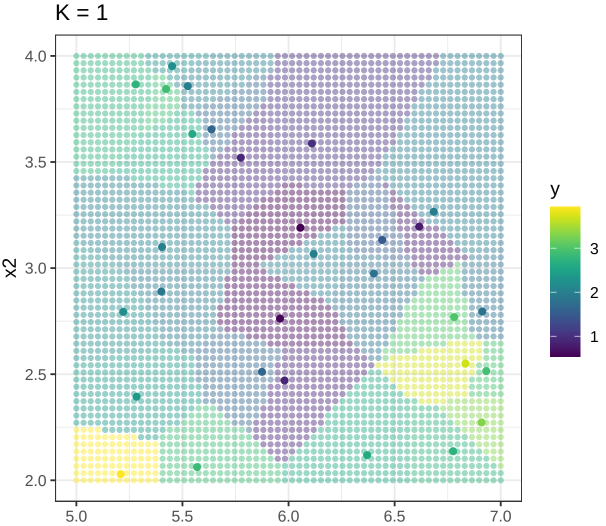
New Training Set, K=1

Here are the KNN surfaces for a variety of values of K.



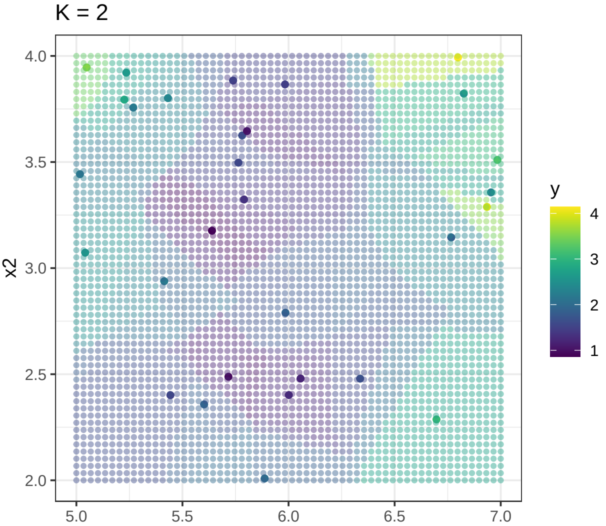
Old Training Set, K = 1

Here are the KNN surfaces for a variety of values of K.



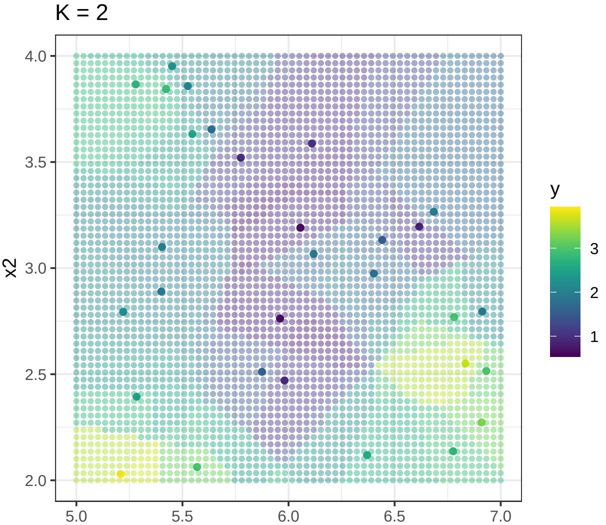
New Training Set, K=2

Here are the KNN surfaces for a variety of values of K.



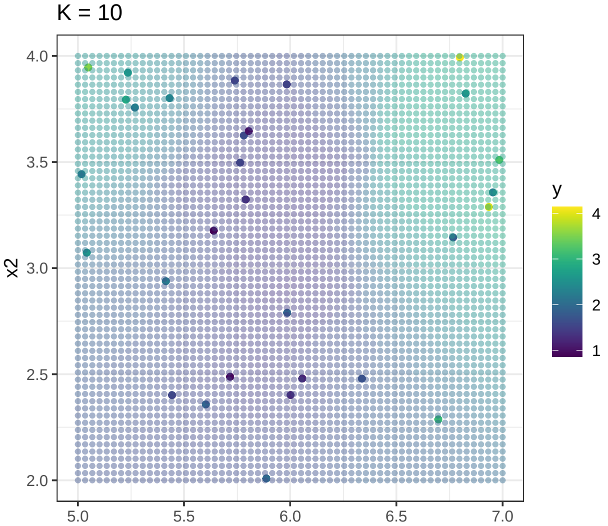
Old Training Set, K = 2

Here are the KNN surfaces for a variety of values of K.



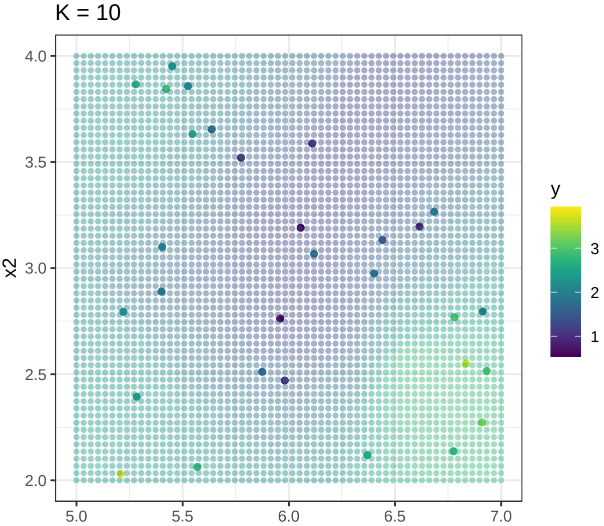
New Training Set, K=10

Here are the KNN surfaces for a variety of values of K.



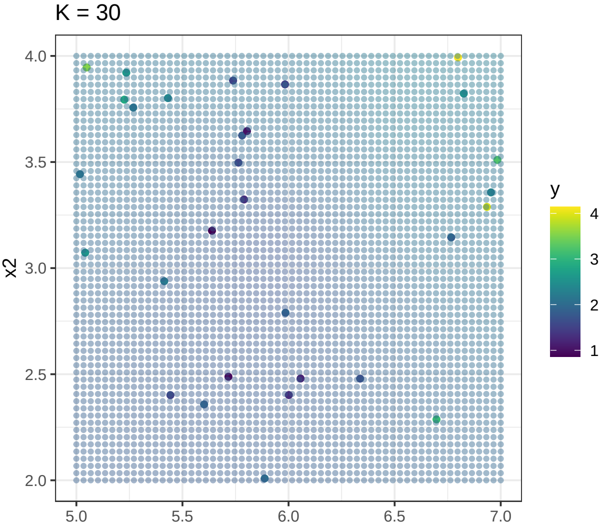
Old Training Set, K=10

Here are the KNN surfaces for a variety of values of K.



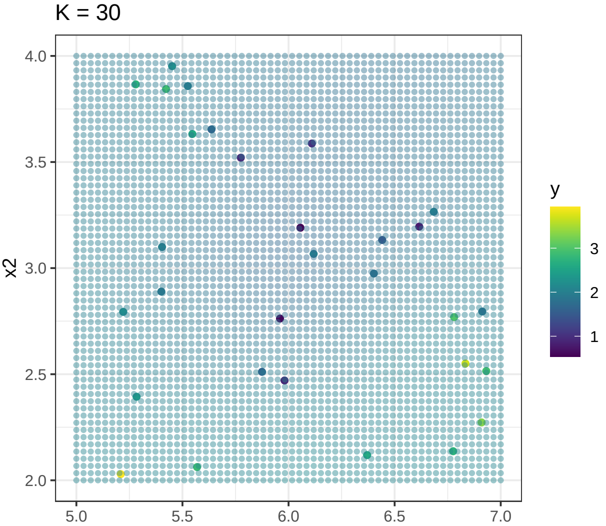
New Training Set, K=30

Here are the KNN surfaces for a variety of values of K.



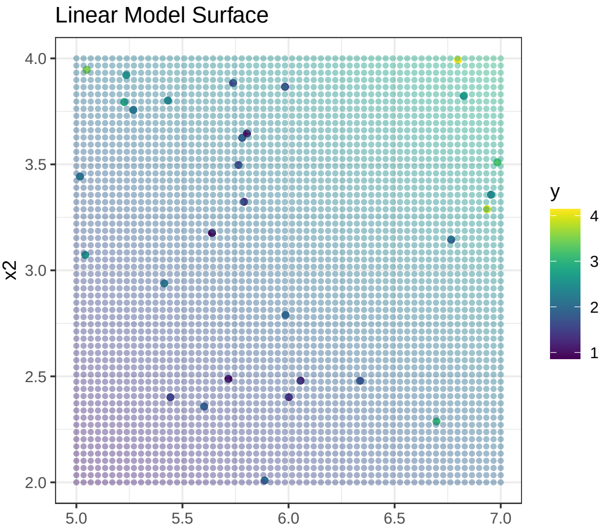
Old Training Set, K=30

Here are the KNN surfaces for a variety of values of K.



New Training Set Linear Model

Here is the true surface described by the linear model $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$:



Reflections

Different values of K lead to different estimates $\hat{y}$.

Reflections

Different values of K lead to different estimates $\hat{y}$.

- Small K mean that we only look at closest or "most similar" points, which should have response values closest to the true response at the test point.

Reflections

Different values of K lead to different estimates $\hat{y}$.

- Small K mean that we only look at closest or "most similar" points, which should have response values closest to the true response at the test point.
    - However, with only a small number of neighbors, predictions are likely to change significantly from training set to training set.

Reflections

Different values of K lead to different estimates $\hat{y}$.

- Small K mean that we only look at closest or "most similar" points, which should have response values closest to the true response at the test point.
  - However, with only a small number of neighbors, predictions are likely to change significantly from training set to training set.
  - Low K yields a high **Variance**, low **Bias** model

Reflections

Different values of K lead to different estimates $\hat{y}$.

- Small K mean that we only look at closest or "most similar" points, which should have response values closest to the true response at the test point.
  - However, with only a small number of neighbors, predictions are likely to change significantly from training set to training set.
  - Low K yields a high **Variance**, low **Bias** model
- Small K mean that we look points that are both close and distant; some of these points have responses that might not be close to the true response at the test point.

Reflections

Different values of K lead to different estimates $\hat{y}$.

- Small K mean that we only look at closest or "most similar" points, which should have response values closest to the true response at the test point.
    - However, with only a small number of neighbors, predictions are likely to change significantly from training set to training set.
    - Low K yields a high **Variance**, low **Bias** model
- Small K mean that we look points that are both close and distant; some of these points have responses that might not be close to the true response at the test point.
    - However, by averaging across a large number of points, predictions will not change much between different training sets.

Reflections

Different values of K lead to different estimates $\hat{y}$.

- Small K mean that we only look at closest or "most similar" points, which should have response values closest to the true response at the test point.
  - However, with only a small number of neighbors, predictions are likely to change significantly from training set to training set.
  - Low K yields a high **Variance**, low **Bias** model
- Small K mean that we look points that are both close and distant; some of these points have responses that might not be close to the true response at the test point.
  - However, by averaging across a large number of points, predictions will not change much between different training sets.
  - High K yields a high **Bias**, low **Variance** model

Reflections

Different values of K lead to different estimates $\hat{y}$.

- Small K mean that we only look at closest or "most similar" points, which should have response values closest to the true response at the test point.
  - However, with only a small number of neighbors, predictions are likely to change significantly from training set to training set.
  - Low K yields a high **Variance**, low **Bias** model
- Small K mean that we look points that are both close and distant; some of these points have responses that might not be close to the true response at the test point.
  - However, by averaging across a large number of points, predictions will not change much between different training sets.
  - High K yields a high **Bias**, low **Variance** model
- In Ch. 5, we discuss methods for choosing optimal K