NC STATE UNIVERSITY

Nonlinear Methods: k Nearest Neighbors

Justin Post

Supervised Learning

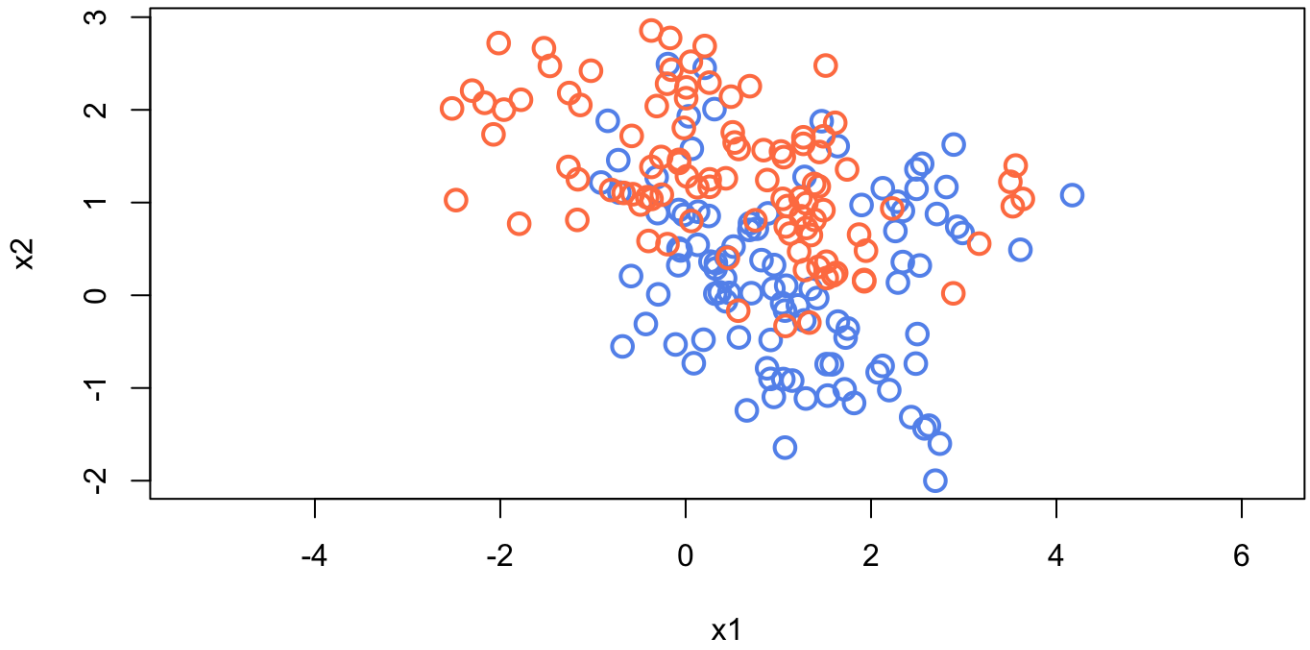
Prediction our goal:

Generally broken into two categories

- Regression
    - Quantitative response
    - Prediction of response
- Classification
    - Categorical response
    - Predict class membership or probability of membership
    - Saw logistic regression and will see classification trees, now we'll look at K Nearest Neighbors (KNN)

Example

Suppose you have two predictors and a categorical response (red or blue)



kNN

Want to predict class membership (red or blue) based on (x1, x2) combination

k Nearest Neighbor idea:

- Use “closest” k observations from training set to predict class (should usually standardize predictors - center/scale)
- Often use Euclidean distance between predictors to determine closest
- $P(red|x1, x2) =$ proportion of k closest values that are red
- $P(blue|x1, x2) =$ proportion of k closest values that are blue
- Classify (predict) to class with highest probability

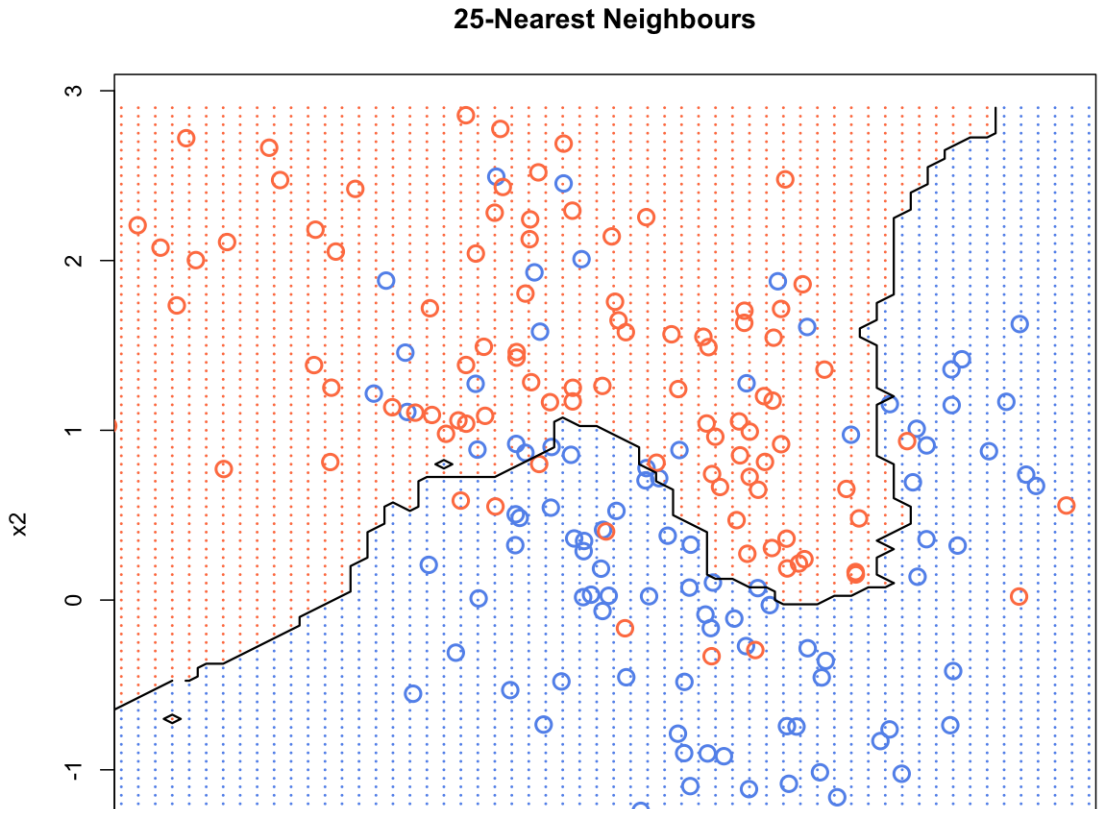
k-Nearest Neighbours Classification

Select the Number of Nearest Neighbours

1 25 150

1 16 31 46 61 76 91 106 121 136 150

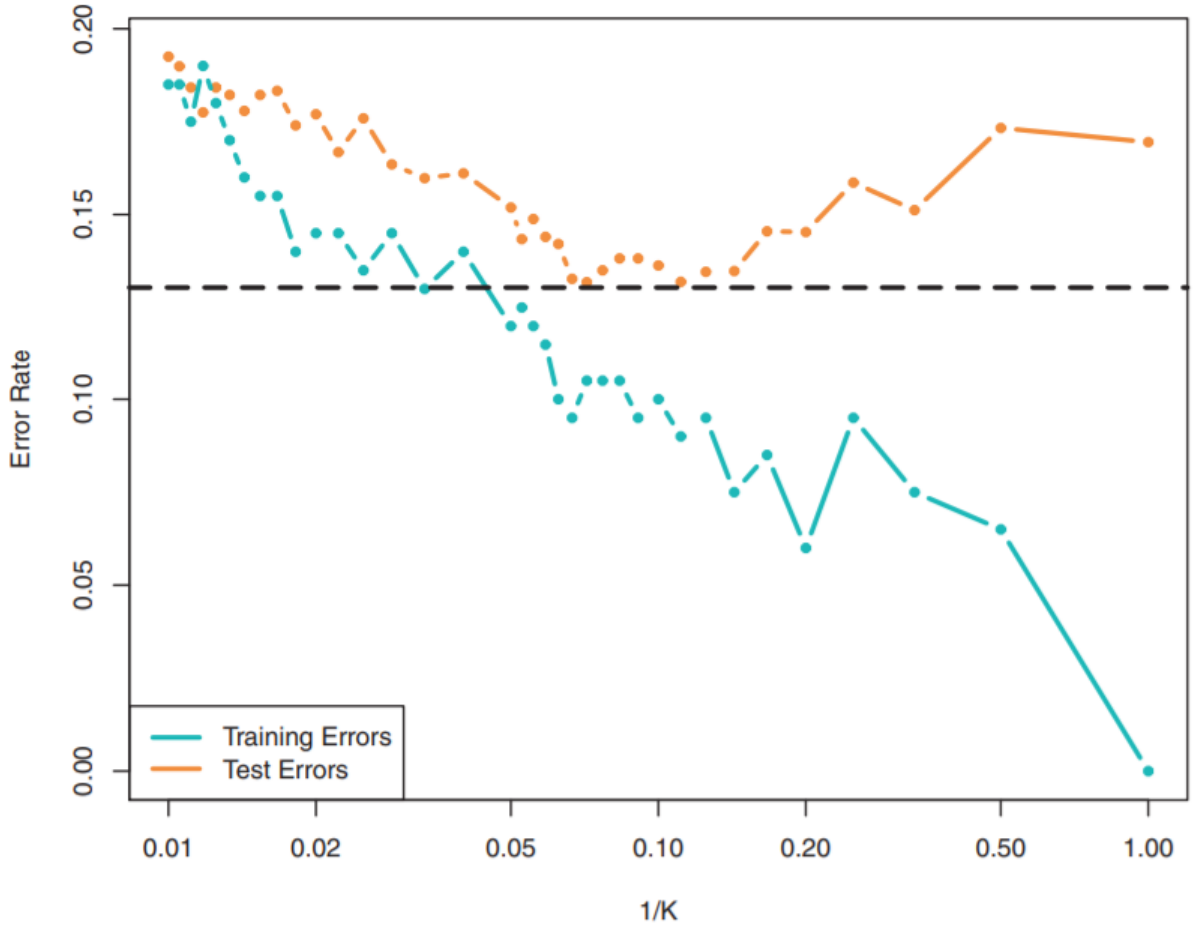
☐ Show the neighbourhood for one point (click to select a point)



kNN

- Small k implies flexible (possibly overfit, higher variance)
    - Training error will be small, may not extend to testing error
- Large k implies more rigid (possibly underfit, lower variance)
- Can do training and test or CV to determine k

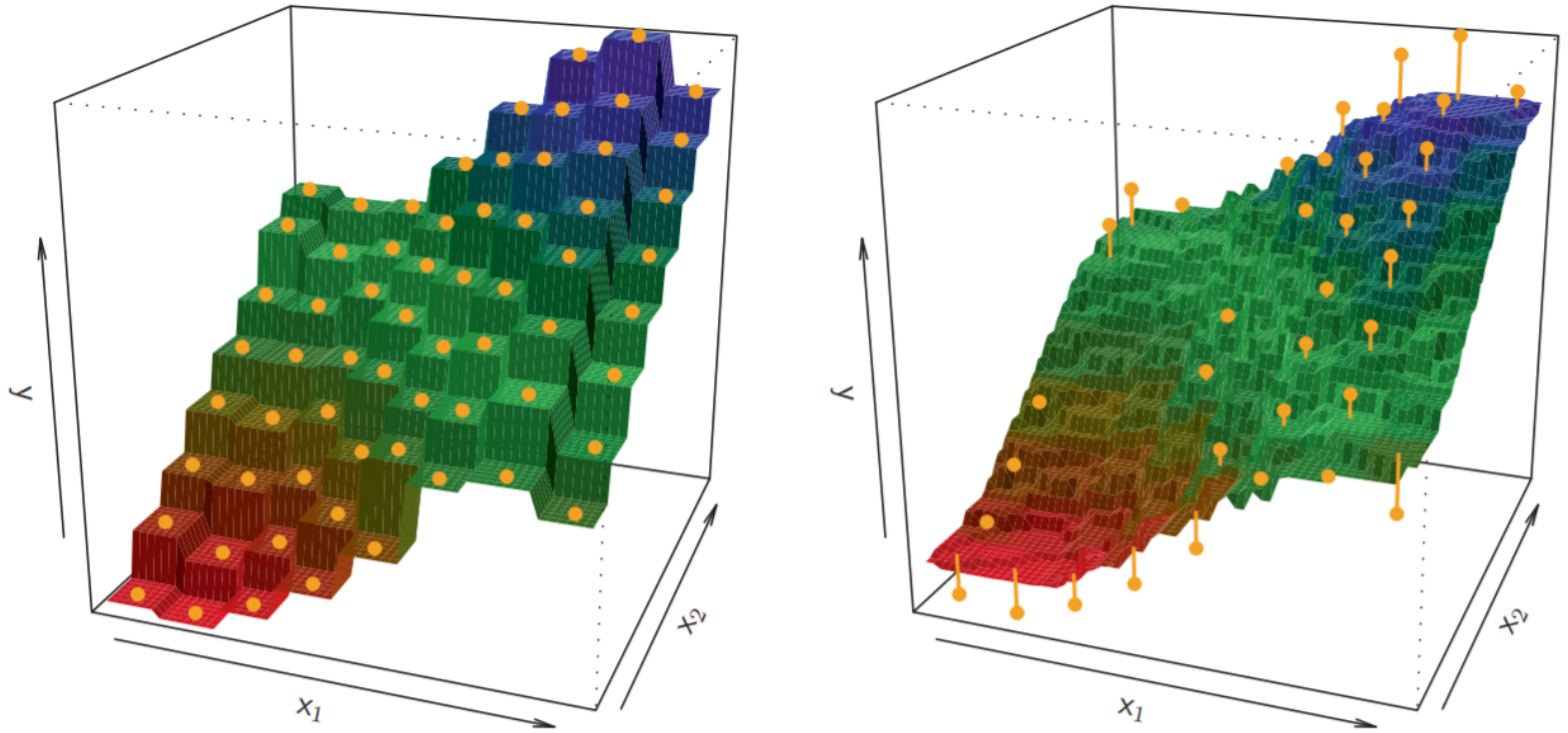
Judge error using misclassification rate.



Courtesy: Introduction to Statistical Learning

kNN for Regression

- Same idea, use average of responses of "closest" k observations in training set as prediction
- Closest again often Euclidean distance
- Very flexible



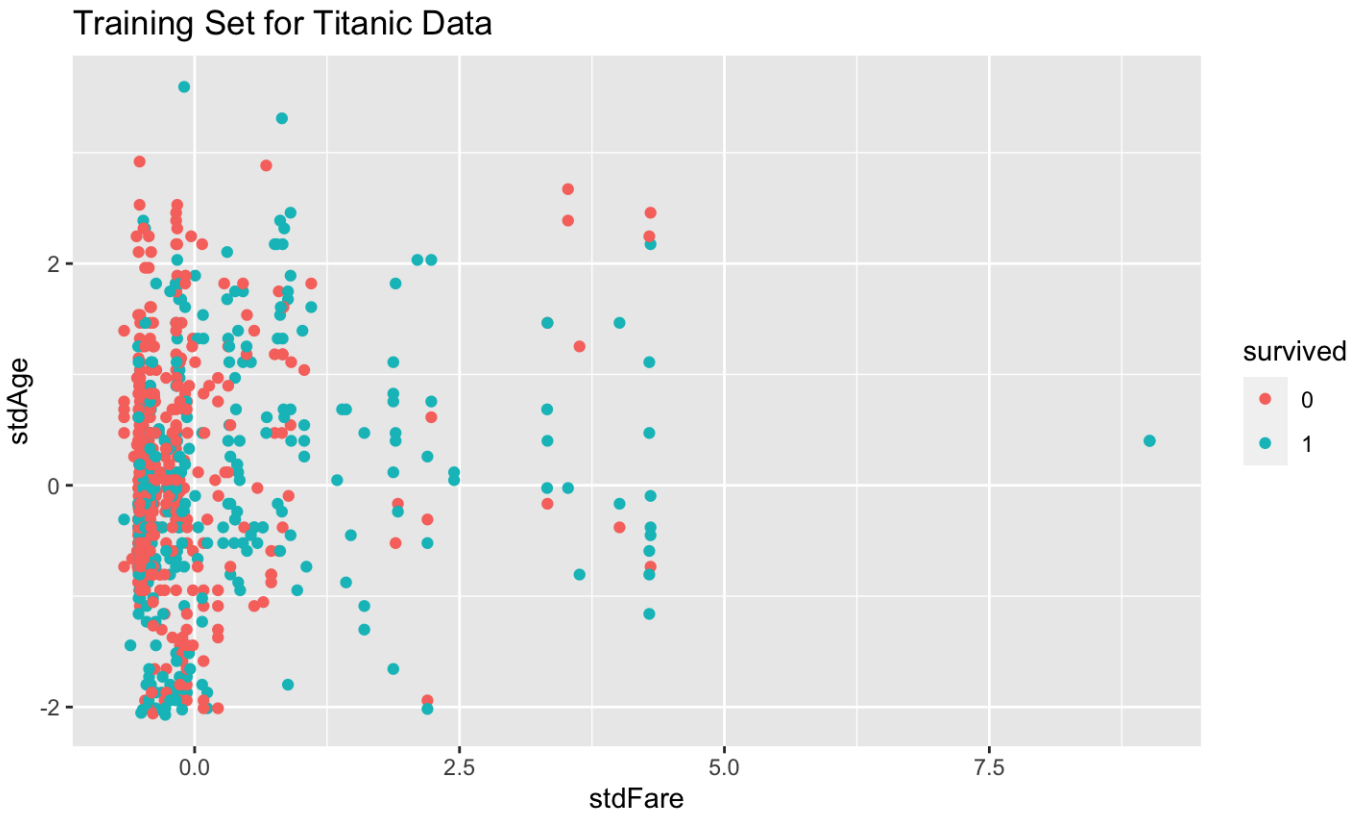
Courtesy: Introduction to Statistical Learning
k = 1 on the left, k = 9 on the right

Fitting kNN in R with `class::knn()` Function

- Titanic Data set: Predict survival status (removed NAs) as a function of traveler age and traveler fare
- Should center and scale (divide by SD) data
    - Use training mean and training SD to do standardization for both training and test set

Fitting kNN in R with `class::knn()` Function

- Titanic Data set: Predict survival status (removed NAs) as a function of traveler age and traveler fare



kNN in R

```
knnFit <- knn(train = select(titanicDataTrain, stdFare, stdAge),
                           test = select(titanicDataTest, stdFare, stdAge),
                           cl = titanicDataTrain$survived,
                           k = 3) #could use CV to determine k
fitInfo <- tbl_df(data.frame(knnFit, select(titanicDataTest, survived, stdFare, stdAge)))
```

```
## Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
```

```
fitInfo
```

```
## # A tibble: 209 x 4
##   knnFit survived stdFare  stdAge
##   <fct>  <fct>      <dbl>   <dbl>
## 1 1      0          2.20   0.0464
## 2 0      0          0.269  2.96
## 3 1      1          0.776  0.188
## 4 1      1          3.63   0.898
## 5 1      1          1.05  -0.308
## # … with 204 more rows
```

kNN in R

```
tbl1 <- table(fitInfo$knnFit,fitInfo$survived)
tbl1
```

```
##
##      0  1
##   0 84 39
##   1 44 42
```

```
#misclass rate in test set
misClass <- 1 - sum(diag(tbl1))/sum(tbl1)
misClass
```

```
## [1] 0.3971292
```

kNN vs Logistic Regression

```
## [1] "Logistic Regression Predictions"
```

```
##
## glmPreds   0   1
##        0 113  54
##        1  15  27
```

```
## [1] "Predicting everyone dies"
```

```
##
##       0   1
##   0 128  81
```

kNN vs Logistic Regression

- kNN does very poorly here - perhaps choose k with CV!

```
## Logistic Regression Misclassification Rate
##                                   0.3301435
##                  kNN Misclassification Rate
##                                   0.3971292
##                    Predicting Death for All
##                                   0.3875598
```

With k = 10

```
##
##      0  1
##   0 98 41
##   1 30 40
```

```
## [1] 0.3397129
```

```
## Logistic Regression Misclassification Rate
##                                  0.3301435
##                 kNN Misclassification Rate
##                                  0.3397129
##                   Predicting Death for All
##                                  0.3875598
```

Recap

- k nearest neighbors uses close observations from the training set for prediction
- Very flexible to not flexible
- Can be used for both regression and classification