쉽게 배우는 머신러닝

Ch.2 Statistical Learning

훈 러닝 (Hun Learning)

January 1, 2020

What is Statistical Learning?

머신러닝의 정의

- 데이터를 Y (알고 싶은 것), X (알고 있는 것)으로 나눠 보자.
- $Y = f(X) + \epsilon$ 이런 관계가 있다고 가정해보자.
- 여기서 f를 추정하는 방법들을 머신러닝이라고 한다!

Why Estimate f?

1. Y를 예측하려고! (Prediction)

$$\hat{Y} = \hat{f}(X)$$

- $\hat{f}$의 형태가 뭔지는 크게 관심이 없다. 몰라 그냥 "black box"라고 해. 나는 $\hat{Y}$만 뚝딱 뽑아내면 된다!
- $\hat{Y}$ 제대로 뽑았니? (Accuracy of $\hat{Y}$ as a prediction)

$$\begin{aligned} E[Y - \hat{Y}]^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= [f(X) - \hat{f}(X)]^2 + Var(\epsilon) \end{aligned}$$

  - 여기서 앞에꺼는 "Reducible Error", 뒤에 꺼는 "Irreducible Error".
  - $Var(\epsilon)$ 얘는 애초에 (내가 가정한) 모델이 생겨먹은 데서 나온거라 아무리 예측을 잘 해도 어쩔 수 없음.
  - 어째저째 열심히 해서 $[f(X) - \hat{f}(X)]^2$를 줄이는 게 예측의 목표!

Why Estimate f?

2. f가 궁금해서! (Inference)

$$\hat{Y} = \hat{f}(X)$$

- Y와 X의 관계식인 f 자체가 나의 관심사다! (광고를 얼마나 때려야 접속자 수가 늘까?)
  $\hat{f}$를 그냥 "black box"라고 하면 안돼! 이걸 들여다 보고 이해하는게 목표야!
- $\hat{Y}$ 제대로 뽑았니? (Accuracy of $\hat{Y}$ as a prediction)
  - 수많은 X들 중에서 뭐가 중요할까?
  - 각각의 X들이 Y에 어떻게 영향을 끼치나?
  - 그 영향이 선형(linear, 단순)이야? 비선형(complex, 복잡)이야?
- 나의 목표가 Prediction이나 Inference이냐에 쓰는 방법론이 다르다. 이 책에서는 그 방법론들을 하나하나 간략히 살펴본다!

How Do We Estimate f?

1. Parametric Methods (모수 추정)

- f가 대충 아래 모델(예컨대 선형모델)처럼 생겼다고 가정해보자.

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$$

그렇다 치고 여러 방법들 중 하나로 (예컨대 OLS 최소자승법)로 저 식에 Y와 X를 우겨 넣어 β를 추정하는 것! **즉, 모수 추정법에서는 모수를 추정하는게 곧 모델을 추정하는 것!**

- 장점: 모델이 간단해서 이해하기 편하다!
- 단점: 애초에 모델 가정이 틀리면? 그렇다고 너무 유연한 모델을 가정해버리면 복잡하고,, 내가 가진 데이터에만 꼭 맞는 Overfitting이 될 수도 있고..

How Do We Estimate f?

1. Parametric Methods (모수 추정)

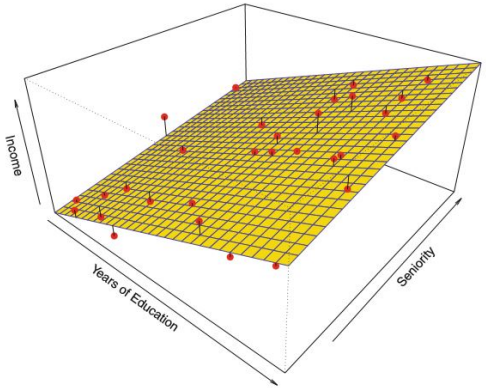


FIGURE 2.4. *A linear model fit by least squares to the* `Income` *data from Figure 2.3. The observations are shown in red, and the yellow plane indicates the least squares fit to the data.*

How Do We Estimate f?

2. Non-Parametric Methods (비모수 추정)

- f의 개형에 대한 가정 따위 없다. 그냥 바로 피팅해! 다만,
  내가 가진 데이터에 잘 들어맞으면서도 너무 괴랄하지 않게! (너무 뜨겁지도 차갑지도 않게?)
  그 "적당한 온도"를 찾는게 관건이다.
  "...gets as close to the data points as possible without being too rough and wiggly"
- 장점: 모델에 대한 가정을 세우지 않으니 유연하다.
- 단점: 데이터가 많이 필요해.. 아주 많이.. 데이터가 적으면 그냥 표본에 선 긋는거야(overfitting)..

How Do We Estimate f?

2. Non-Parametric Methods (비모수 추정)

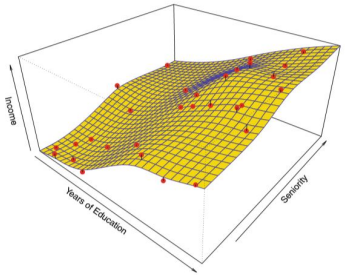


FIGURE 2.5. *A smooth thin-plate spline fit to the* `Income` *data from Figure 2.3 is shown in yellow; the observations are displayed in red. Splines are discussed in Chapter 7.*

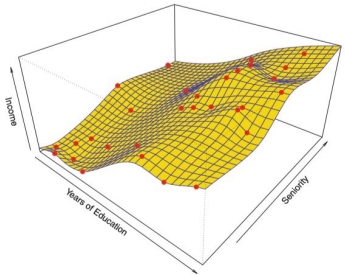


FIGURE 2.6. *A rough thin-plate spline fit to the* `Income` *data from Figure 2.3. This fit makes zero errors on the training data.*

Trade-off: Prediction Accuracy vs Model Interpretability

Prediction 성능이 좋을수록 모델은 더욱 이해하기 힘들어진다!

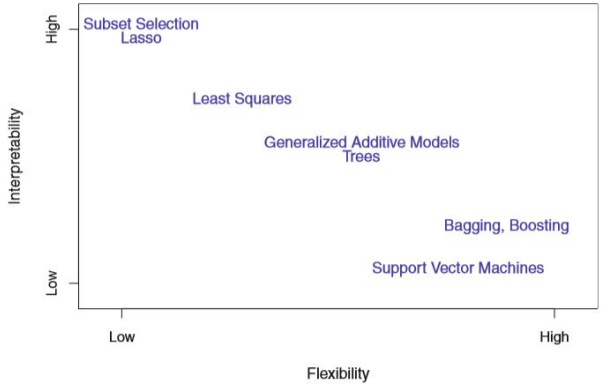


FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

Supervised vs Unsupervised Learning

- **데어터를 봤는데 Y와 X가 이쁘게 나눠져있으면 지도학습(Supervised Learning).**
  우리의 학습을 지도할 Y가 있구나!
- **데이터를 봤는데 Y는 없고 X밖에 없다..? 비지도학습(Unsupervised Learning)**
  ex) 조회수(Y)는 없고 동영상 길이, 동영상 업로드 시간, 댓글 수 등등(X들) 밖에 없다. 이럴때는 할 수 있는게 뭐가 있나?
  - Clustering (Cluster Analysis): Y가 없어도 X들만 보고 데이터를 분류할 수는 있잖아?
    ex) 동영상 조회수(Y)는 모르지만 댓글 수가 적은 동영상들끼리 묶고, 댓글 수가 많은 동영상들끼리 묶어보자.
    ex) 쇼핑몰의 매출(Y)은 모르지만 접속자 수에 따라 분류해보자.
  - X의 종류가 많아질수록 산점도 개수가 많아진다($p(p-1)/2$). 이럴 때 자동으로 Clustering하는 알고리즘은?

Supervised vs Unsupervised Learning

비지도학습(Unsupervised Learning)의 예

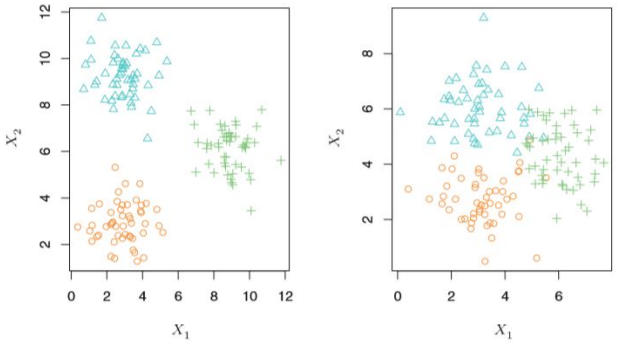


FIGURE 2.8. *A clustering data set involving three groups. Each group is shown using a different colored symbol.* Left: *The three groups are well-separated. In this setting, a clustering approach should successfully identify the three groups.* Right: *There is some overlap among the groups. Now the clustering task is more challenging.*

Regression vs Classification

- **Y가 수치형(Quantitative, Numerical)이면 Regression**
  나이, 몸무게, 소득, 주가 등
- **Y가 범주형(Qualitative, Categorical)이면 Classification**
  성별, 브랜드, 감염 여부 등

Regression: Assessing Model Accuracy

다양한 머신러닝 방법이 있는 이유는 데이터에 따라 뭐가 좋은지 다르기 때문. 그러면 어떤 데이터에 대해 어떤 모델이 "적절"하다고 할 수 있는가?

Measuring the Quality of Fit

- **MSE:** 어떤 방법으로 예측한 $\hat{Y}$와 실제 Y 간의 차이를 나타내는 척도. 나의 예측 모델이 얼마나 정확한가(실제와 맞는가)?

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

  - Y와 X로 이뤄진 데이터셋을 training과 test로 나누자. 나의 모델 $\hat{f}(X)$는 training 자료를 바탕으로 추정한다. training 자료로 계산한 MSE가 training MSE
  - 그러나 중요한 것은 test MSE. 이미 아는 자료를 잘 맞춰봤자 소용이 없으니까. **내가 아는 자료로 추정한 모델이 내가 모르는 것을 얼마나 잘 추정하느냐?**
  - 목적은 test MSE (Averaged squared prediction error)가 최소인 방법을 택하는 것!

$$Argmin\ test\ MSE = Ave(y_0 - \hat{f}(x_0))^2$$

Regression: Assessing Model Accuracy

training MSE만을 기준으로 삼으면 어떤 문제가 벌어지는가?

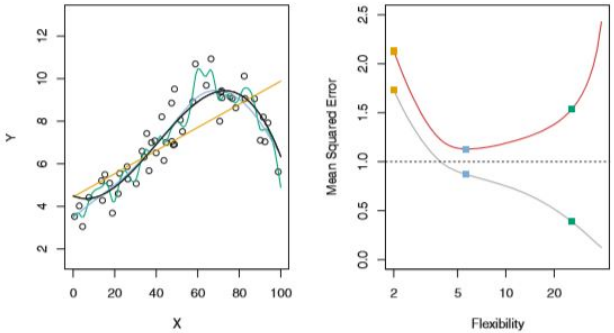


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

Decomposition of Expected Prediction MSE

- 가정
  1. (y_0, x_0): test data, 모르는 거니까 둘 다 확률변수
  2. True Model: $y_0 = f(x_0) + \epsilon$, 여기서 $\epsilon \sim ?(0, Var(\epsilon))$, x_0와 uncorrelated
  3. $\hat{f}(x_0)$는 확률변수 x_0의 함수이므로 확률변수
- Squared Prediction Error, SPE $\equiv (y_0 - \hat{f}(x_0))^2$
- Mean Prediction Error, MSPE $\equiv Ave(y_0 - \hat{f}(x_0))^2$
  ("거리", 이 수치는 우리가 실제 데이터에서 계산할 수 있는 값)
- Expected Prediction Error, E(SPE) $\equiv E(y_0 - \hat{f}(x_0))^2 = E[E(y_0 - \hat{f}(x_0))^2 | x_0)]$
  (아주 많은 test 데이터 x_0들로 아주 많은 $\hat{f}(x_0)$를 계산하여 각각의 Prediction Error(거리)를 계산해 평균한 값, 대수의 법칙에 따라 MSPE $\xrightarrow{P}$ E(SPE))

$$E(SPE) = E(f(x_0) - \hat{f}(x_0))^2 + Var(\epsilon) = Var(\hat{f}(x_0)) + Bias^2(\hat{f}(x_0)) + Var(\epsilon)$$

($\hat{f}$ 자체의 Var + $\hat{f}$가 f와 멀리 떨어진 Bias + True model에서의 오차항의 Var)

Regression: Assessing Model Accuracy

- $Var(\hat{f}(x_0))$: Variance of a model
  - y의 추정을 위해 어떤 모델 를 쓴다고 하자. 똑같은 y_0를 서로 다른 training 데이터로 추정하면 $\hat{y_0} = \hat{f}(x_0)$의 값이 얼마나 달라지나?
  - 모델의 분산이 높으면 어떤 training 데이터를 쓰냐에 따라 $\hat{y_0}$의 값이 널뛰기할 것. 환장할 노릇.
  - 더 유연하고 복잡한 모델을 쓸수록 train 데이터에 더 fitting이 되므로, 다른 데이터를 썼을 때 값이 크게 달라진다. 즉 Var가 커진다.
- $Bias^2(\hat{f}(x_0))$: Bias of a model
  - 실제보다 단순한 모델 를 씀에 따라 발생하는 실제 모델과의 편차.
  - 예컨대 선형모델 를 쓰면 다른 train 데이터를 가져와도 예측값이 크게 다르지 않을 것. 그러나 실제 모델이 비선형이며 복잡할수록 모델이 가지는 편차가 커진다.
- **일반적으로, 더 유연한(복잡한) 모델을 쓸수록 Var은 증가하고 Bias는 감소한다. 전체 MSE는 Bias와 Var의 증감 속도에 따라 증가 혹은 감소한다.** $\rightarrow$ Trade-off!!

Regression: Assessing Model Accuracy

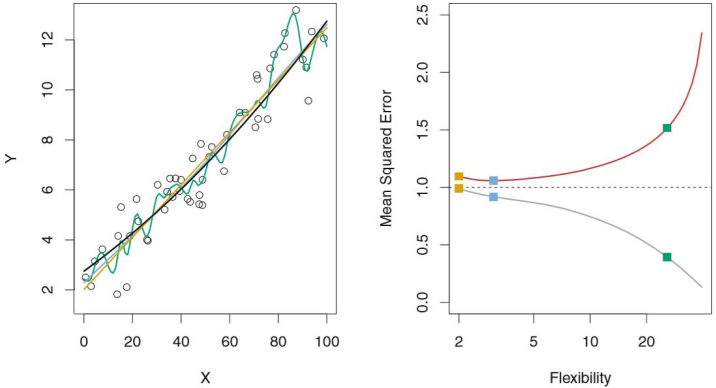


FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

Regression: Assessing Model Accuracy

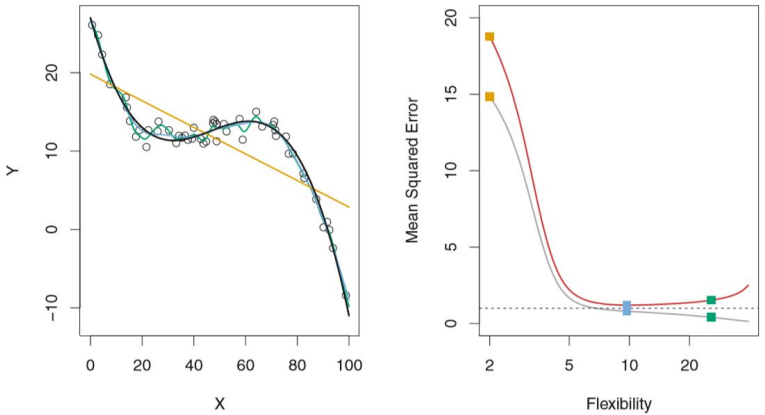


FIGURE 2.11. *Details are as in Figure 2.9, using a different f that is far from linear. In this setting, linear regression provides a very poor fit to the data.*

Classification: Assessing Model Accuracy

Y가 범주형 자료인 Classification에서는 모델의 성능을 어떻게 측정함?

- Error Rate: 전체 N번의 예측 중에서 예측이 틀린(실제와 다른 범주인) 비율

$$Test\ Error\ Rate = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq \hat{y}_i)$$

$$Expected\ Test\ Error\ Rate = E[I(y_0 \neq \hat{y_0})]$$

- y_0은 model $\hat{f}$를 사용해 예측한 x_0의 범주. 이때 $\hat{f}$를 classifier 라고 한다.
  "좋은" classifier는 $E[I(y_0 \neq \hat{y_0})]$를 최소화!

The Bayes Classifier: The unattainable gold standard

- $X = x_0$로 주어져있을 때 Y가 취할 수 있는 값들의 확률, 즉 조건부 분포를 안다고 하자.

$$P(Y = j|X = x_0)$$

- 이 X가 주어진 Y의 조건부 분포에 따라, 주어진 x_0에 대해 조건부 확률이 가장 큰 Y를 y_0로 하면 $E[I(y_0 \neq \hat{y_0})]$가 최소화!

Classification: Assessing Model Accuracy

Indicator Variable은 다음과 같이 정의되는데,

$$I(y_0 \neq y_0) = \begin{cases} 0 & \text{if } y_0 = \hat{y_0} \\ 1 & \text{if } y_0 \neq \hat{y_0} \end{cases}$$

Bayes Classifier를 사용하면 $\hat{y_0}$은 무조건 주어진 x_0에서 확률이 가장 큰 y값을 y_0로 한다. 즉

$$\hat{y_0} = Argmax_j P(Y = j | X = x_0)$$

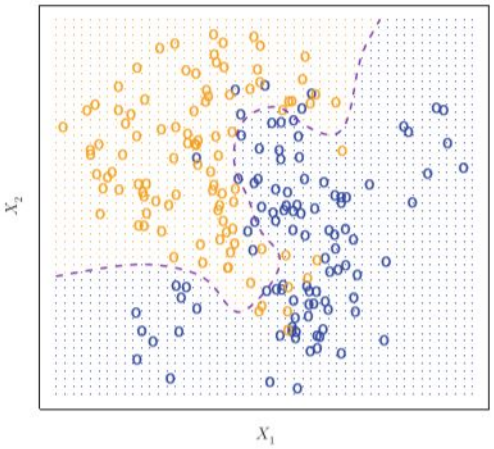
때문에 Bayes Classifier에서 Indicator Variable은

$$I(y_0 \neq y_0) = \begin{cases} 0 & \text{w.p. } Max_j P(Y = j | X = x_0) \\ 1 & \text{w.p. } 1 - Max_j P(Y = j | X = x_0) \end{cases}$$

Bayes Error Rate at x_0 = $1 - Max_j P(Y = j | X = x_0)$
Overall Bayes Error Rate = $1 - E[Max_j P(Y = j | X = x_0)]$
모든 x_0 값에서 $Max_j P(Y = j | X = x_0)$를 구해(0, 1, 혹은 그 사이) 평균한 값. Irreducible Error

Classification: Assessing Model Accuracy

Ex) Bayes Classifier in Two Classes-Two Predictors case



- $P(Y|X_1, X_2)$를 안다고 가정

$$P(Y|X_1, X_2) = \begin{cases} 1 \\ 0 \\ \in [0, 1] \end{cases}$$

- **Bayes Decision Boundary:**
  $P(Y|X_1, X_2) = 0.5$가 되는 (X_1, X_2)을 이은 선
- **Bayes Classification**

$$\hat{y_0} = \begin{cases} 1 & \text{if } P(Y|X_1, X_2) > 0.5 \\ 0 & \textit{if } P(Y|X_1, X_2) \leq 0.5 \end{cases}$$

Classification: Assessing Model Accuracy

실제로는 $P(Y = j|X = x_0)$을 모르기 때문에 Bayes Classifier는 사용할 수 없음. 때문에 많은 경우 $P(Y = j|X = x_0)$을 추정하여 그를 바탕으로 Classification. 대표적인 경우가

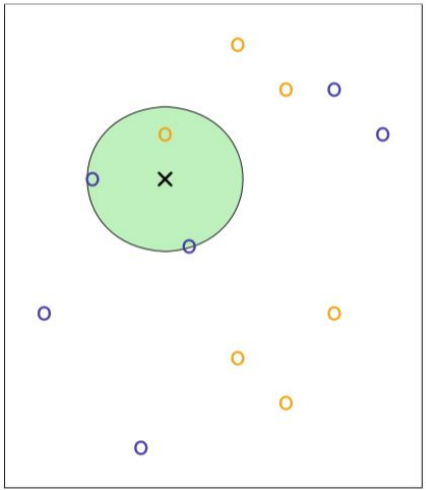
K-Nearest Neighbors(KNN)

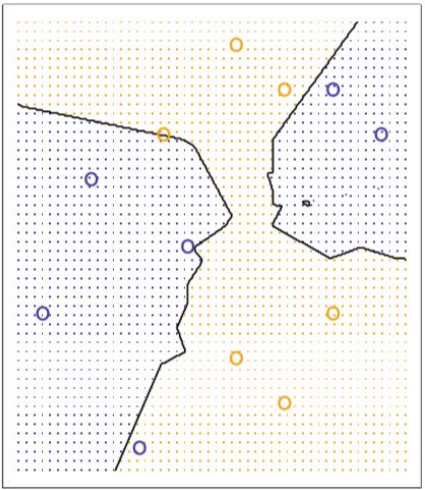
$$P(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

- 어떤 점 x_0에서 "가장 가까운" K개의 점들의 집합을 N_0
- 그 중에서 j의 개수를 $\sum_{i \in N_0} I(y_i = j)$
- 이렇게 구한 추정 조건부 분포를 바탕으로 Bayes Classifier 적용!
- K의 값은 점 주변에 다른 점을 몇개를 볼 거냐의 문제, 더 많은 점을 보면 볼수록 더욱 경직적인 Decision Boundary

Classification: Assessing Model Accuracy

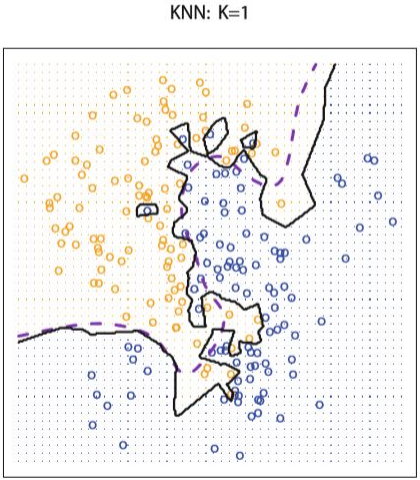
K-Nearest Neighbors(KNN)

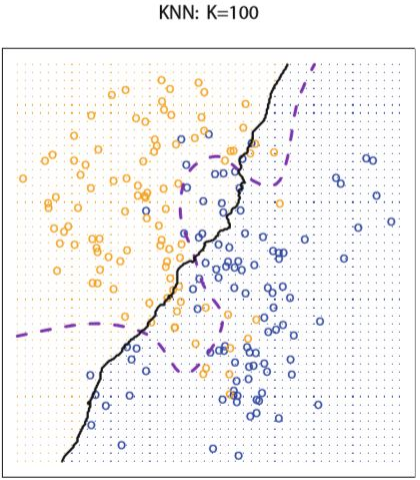




Classification: Assessing Model Accuracy

K-Nearest Neighbors(KNN)





Classification: Assessing Model Accuracy

K-Nearest Neighbors(KNN)

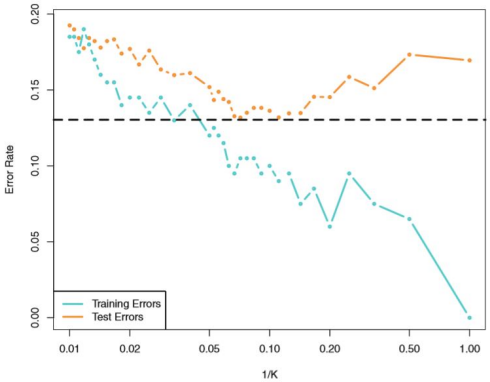
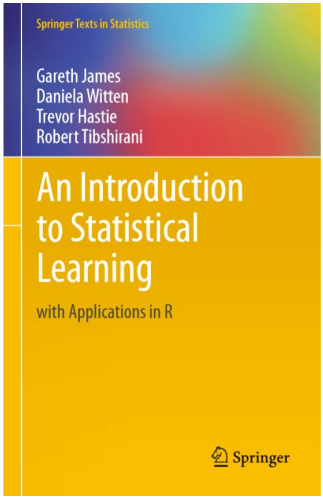


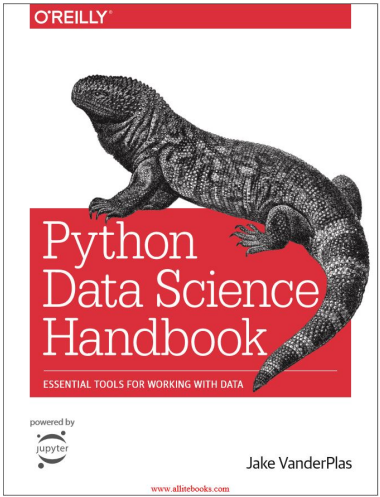
FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.*

About The Textbook



- 주 교재:
  An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013.
- 목차:
  1. Intro
  2. Statistical Learning
  3. Linear Regression
  4. Classification
  5. Resampling Methods
  6. Linear Model Selection and Regularization
  7. Moving Beyond Linearity
  8. Tree-based Methods
  9. Support Vector Machines
  10. Unsupervised Learning

교재



- 보조 교재:
  Jake VanderPlas. Python Data Science Handbook: Essential Tools for Working with Data. :O'Reilly Media, 2016
- 목차:
  1. iPython
  2. Numpy
  3. Pandas
  4. Matplotlib
  5. Machine Learning