첫시간이니 만큼, 아주 기본적이고 개괄적인 내용들부터 시작합니다. 틀린 내용이 있다면 가차없이 처단바랍니다.

간략한 책구성

```
**목차**

- [GAN?](#gan)
- [간략](##간략한 책구성)
- [간략](#간략한 책구성)
- [Before GAN](#before-gan)
- [Adversarial Nets](#adversarial-nets)
- [How to Train](#how-to-train)
- [Into Deep](#into-deep)
- [Algorithm](#algorithm)
```

```
- [GAN?](#gan)
```

Assessing Model Accuracy

- Load Datasets

preface

시작하기에 앞서, 이 책은 ESL(Elementary of Statistical Learning)에서 수정을 통해 나온 책이다. ISL은 데이터 시대에 접어들면서, statistical learning이 학문적 분야를 넘어 각 분야에 적용되고 있는 흐름에 맞추어 변형된 책이다.

쉽게 말해, 연구자들의 입문서가 아닌 방법을 "적용"하고자 하는 사람들을 위한 입문서이다. 따라서 여러 모델들이 어떻게 구성되어 있고, 어떤 강점이 있는지, 어떤 단점이 있는지, 어떤 상황에 어떤 모델을 적용해야 하는지는 다루고 있지만, 왜 그런 구성이 되었는지에 대한 이론적인 설명은 깊게 들어가지 않으며, 특히 'matrix연산'은 상당부분 피하고 있다.

(추가적인 이론적 배경을 위한 탐구가 필요할것 같다..)

간략한 책구성

study plan에서도 적어놨지만, 무엇을 공부할것인지, 간략하게 알아보고 가자.

2과 : statistical learning에 대한 개괄과 기본 개념들을 설명한다. KNN도 간략하게 설명된다

3,4과 : classical linear model에 대해 다룬다. 구체적으로는 3과에서 linear regression을, 4과에서는 logistic regression을 다룰것이다. (logistic의 경우 generalized linear model에 속한다. 쉽게 말해 로짓(log odds)과 linear 관계가 있음)

5과 : 모델의 성능을 측정하는 방법, cross-validation이나 bootstrap등에 대해 배울 것이다. 교육세션에서 배웠던 것과 비슷하다.

6과 : linear method에 대해 좀더 배우게 된다. model selection, ridge, lasso등에 대해 다룬다. (드디어!)

7과 : non-linear method에 대해 다룬다. 난항이 예상된다

8과 : 인기만점 tree-based model에 대해 다룬다. bagging, boosting, RF등을 다룬다.

9과: 들어는 봤으나 설명하진 못하는, SVM(support vector machine)에 대해 다룬다.

시작!

statistical learning의 가장 일반적인 목표는 x들과 y의 어떠한 '관계'가 있을것이라 '가정'하고, 이를 밝히는 것이다. 좀 더 구체적으로 말하자면, 우리에게 $x_1, x_2, .., x_p$, 즉 p개의 input이 주어졌을때, 우리의 목표 y 를 예측하는 것이다.(사실 x와 y의 관계를 '추론'하는 것도 아주 큰 분야 중 하나이지만, 여기선 예측에 중점을 둔다.) 이는 수식으로 말하면 다음과 같다.

$$Y = f(x)$$

여기서 f는 어떠한 관계든 가능하다.

대표적으로 어떤 사람의 키(x_1), 성별(x_2), 나이(x_3)이 주어졌을때 그 사람의 몸무게(Y)를 예측하는 식일 것이다. 그러나, 실제 현실에서는 당연히 키와 성별, 나이만 가지고 키를 딱 예측할 수 없다. 수많은 인과관계가 얽혀 있어, 키랑 성별, 나이를 통해 키를 어느정도 짐작할수 있을뿐, 오차가 당연히 있을것이다. 이러한 오차를 포함해주기 위해, 통계학에서는 *error term* 'ϵ' 을 넣어준다.

$$Y = f(x) + \epsilon$$

여기서 X는 **예측변수, 독립변수, 변수**(predictors, independent variables, variables) 등의 이름으로 불리고, 목표인 Y는 **반응변수, 종속변수**(response or dependent variable) 등으로 불린다. 또 ϵ $(errorterm)$은 우리가 고려하지 못한, 혹은 현실의 어떠한 기이한 작용들로 생겼을 만한 수많은 오차들을 다 포함하며, 평균이 0일 것이라고 '가정'한다. ϵ은 우리가 가정한 모델로는 아무리 잘 만들어도 줄일 수 없는, 즉 irreducible error를 의미한다.

그러나 우리의 가정대로 정말 키와 몸무게가 어떤 관계가 딱 존재한다 하더라도, 우리는 그 f를 알 수 없다. 왜? 우리의 정보는 한정적이니까. 고로 우리는 우리가 가진 자료를 가지고 f를 '추정'하게 된다. 그리고 그 추정된 관계를 가지고 Y를 추정한다. 이때 실재와 우리의 추정을 구분하기 위해 $\hat{f}$ 와 같이 표현한다. 즉 우리가 만들어낼 관계식은 다음과 같다.

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

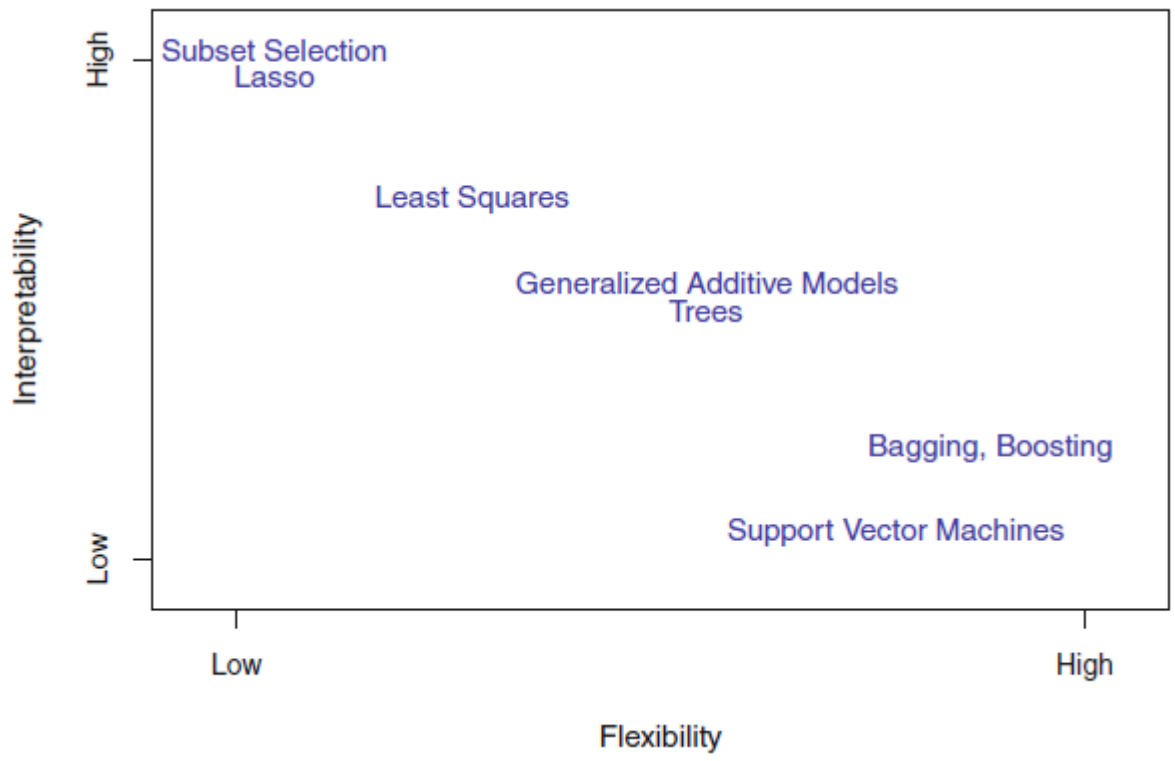
그럼, f를 어떻게 추정할까?

f를 추정하는 방법으로는 **parametric 방법**과 **non-parametric방법** 이 있는데, parametric방법은 X와 Y간에 특정한 관계(대표적으로 예를 들면 선형관계)가 있다고 미리 가정하고 그 틀에 맞추어 추정을 한 후 이를 다시 우리의 가정과 비교하는 방법이다. 해당 가정 사항의 parameter 몇개만 예측하는 것으로 문제가 축소되고, 무엇보다 해당 가정사항 안에서 많은 분석들과 예측을 할 수 있다는 powerful하다는 강점이 있으나 가정이 틀렸을 경우 분석 자체가 도루묵이라는 위험성이 있다.

반면 non-parametric 방법은 f에 대한 어떤 가정도 없이 데이터만을 보고 데이터의 특성을 잘 나타내는 f를 찾는 방법으로, 가정이 틀릴 위험이 없다는 강점이 있으나 기본적으로 많은 데이터를 필요로 하고, parametric 방법 만큼 다양한 분석을 할 수 없다는 약점이 있다. 여기서에서는 parametric 방법에 좀더 집중할 것이다.

flexibility에 따른 다양한 model들

f를 추정하는 여러가지 방법들(method들)이 있는데, 이들을 나누는 가장 큰 기준은 그 방법들의 **flexibility**이다. flexibility란 말그대로 유연성, 즉 우리가 가진 데이터에 얼마나 유연하게 적합하여 f를 추정하는가를 의미하는 것이다. 그러나 flexible하다고 다 좋은건 아닌데, flexible하면 할 수록 해석력(Interpretability)를 잃어버리기 때문이다. 다양한 분석 방법들의 flexible, Interpretability간의 관계를 나타낸 그림은 다음과 같다. 해당 방법들은 뒷장에서 다룰것이기에, 자세한 설명은 생략한다.(사실 아직 잘 모른다.)



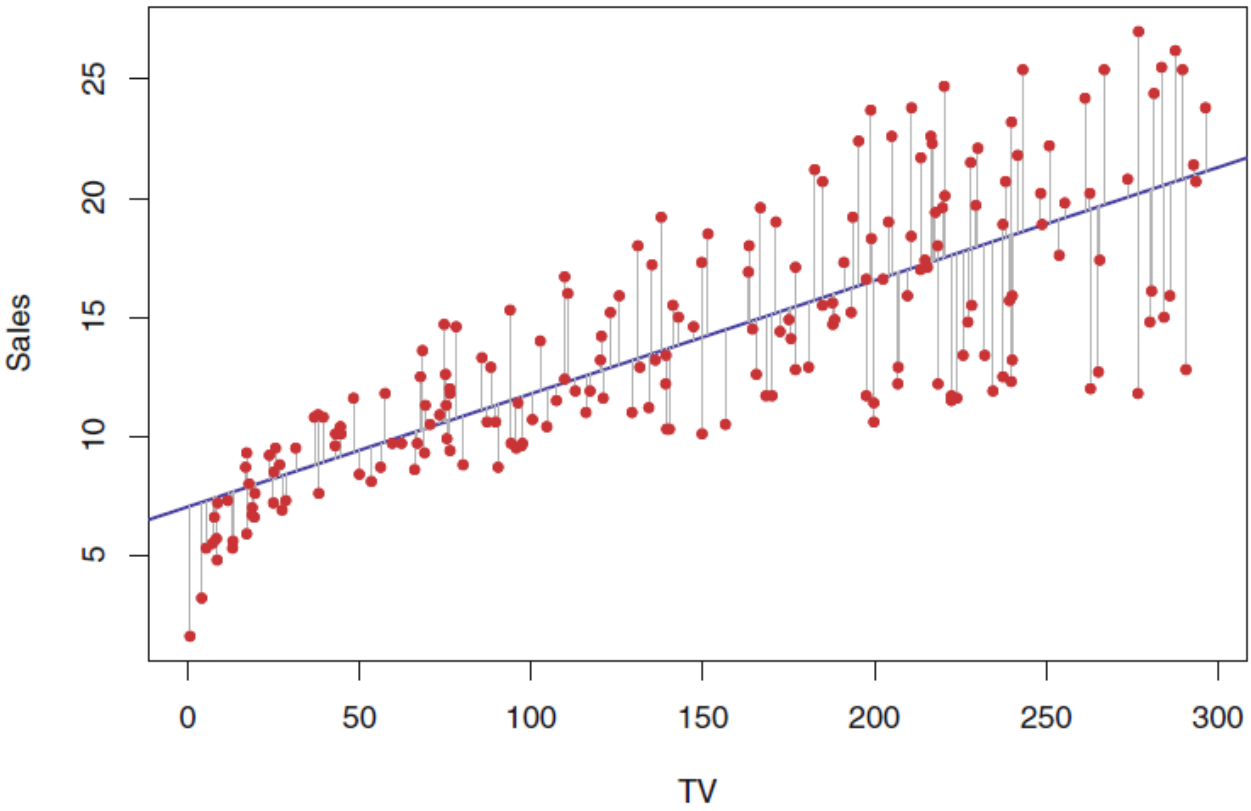
Assessing Model Accuracy

앞에도 말했듯이 하나의 데이터를 다루고자 할때 여러가지 방법들과 모델들이 쓰일수가 있는데, 그럼 그 모델들을 어떻게 평가해야 할까? 우선 첫째로, 우리의 모델이 우리가 가지고 있는 데이터를 얼마나 잘 맞추는가를 볼 수 있다. 회귀문제에서는, 이를 평가하는 지표로 **MSE**라는 것을 가장 많이 쓴다. MSE의 식은 다음과 같다.

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

y_i는 i번째 실제 우리가 가지고 있는 데이터고, $\hat{f}(x_i)$는 i번째 변수들을 통해 우리가 예측한 y값이다.

($\hat{Y} = \hat{f}(x)$ 이 식을 상기하자) 고로, (((예측한 y값과 실제 y값의 오차)의 제곱)의 평균)을 계산한 지표이다. 그림으로 쉽게 보자면, 아래 그림에서 파란색 선이 우리가 예측한 y값들의 모임, 회색선이 실제값과의 차이이다.



낮은 MSE값은 우리의 모델이 주어진 데이터를 잘 설명하고 있다는 것을 의미한다. MSE는 뒷장에서 여러 방식으로 사용되는 아주 중요한 지표이다. 그러나 이는 모델을 평가하는 지표로써는 아주 일차원적인 지표이다. 왜냐면, 우리는 **주어지지 않은**자료들을 잘 예측하고 싶은 것이지, **주어진**자료를 잘 예측하는 것은 우리의 목표가 아니다. 쉬운 예를 들자면, 우리는 지난 6개월간의 주식 데이터를 보고 다음날의 주식의 가격을 알고 싶은 것이지, 일주일 전의 주식 가격을 예측하고 확인하려는게 아니다. 즉, 주어진 $y_1,,,y_p$를 잘 맞추고 싶은게 아니라 아직 접해보지 못한 데이터를 통해 y_0를 확인하고 싶은 것이다. (여기서 0은 아직 접하지 못한 데이터를 통틀어 말한다.)

여기서 주어진 자료들은 training data, 훗날 모델을 직접 돌리며 주어질 자료를 test data라고 부른다.

고로 사실상 모델의 성능은, test data를 얼마나 잘 맞추느냐, 즉 test data에 대한 오차가 적을 수록 좋다고 할 수 있다. test data에 대한 오차는 test MSE라고 말하며 다음과 같이 구할 수 있다.

$Avg(y_0 - \hat{f}(x_0))^2$ (다시, 여기서 0은 아직 접하지 못한 데이터를 통틀어 말한다.)

test data는 아직 주어지지 않은 데이터라는 점에서, 이를 미리 계산하고 낮추기란 쉽지 않다. 이러한 한계를 완화하고자 하는 다양한 트릭들(cross-validation 등등)이 5장에 나올것이다.

얼핏 보았을때는, 주어진 자료들을 잘 맞추면, 나중에 주어질 자료들도 잘 맞추지 않을까? 라는 생각이 든다. 그러나 여기서, 아주 중요한 개념이 등장한다.

Bias-Variance Trade-Off

한번쯤은 들어봤을 만한, ML job-interview에도 꼭 등장하는 단골 문제다.

이 문제의 가장 핵심 개념은, lowest training MSE가 lowest test MSE를 보장하지 못하며, 심지어는 다른 모델들보다 성능이 더 나빠질 수도 있다는 것이다.

용어를 먼저 정의하자.

- Bias : the error that is introduced by approximating a real-life problem. 우리가 f를 추정하고자 할때 단순히 각 점들을 잇는 지그재그의 선을 긋지 않고 나름의 모델을 세워 단순화 시켜 예측을 할것이다. 이런 단순화 작업에 필연적으로 동반하며 생기는 오차가 바로 bias이다. 예를 들어 우리가 선형회귀를 적합하고자 한다면, 우리 모델의 '선형성' 으로 인해 필연적으로 오차가 생길 것이다.('선형성'이라는 가정은 엄청나게 큰 가정이다.) 이는 어떻게 적합하느냐에 따라 조금씩 그 값이 달라지겠지만, 아무리 잘 적합해도 '선형성'으로 인해 줄어들지 않는 오차가 있다. 당연히 **flexible할 수록**, 즉 덜 단순화시킨 모델일 수록 **bias는 줄어들 것이다**.

> 처음에는 bias는 잔차인줄 알았는데, 아니었다. 아래의 식에서도 나와 있듯이,
> $Bias(\hat{f}(x)) = E(\hat{f}(x) - f(x)) = E(\hat{f}(x)) - f(x)$로 받아들여야 한다. 즉 실제 truth 함수인 '$f(x)$'와 우리가 추정한 '$\hat{f}(x)$의 기대값'(무수히 많은 데이터셋에대해 무수히 많은 적합을 시켜보았을때의 생기는 여러 모델들의 평균)과의 '차'로 받아들여야 한다.

- Variance : the amount by which $\hat{f}$ would change if we estimated it using a different training data set. 즉 우리가 다른 training data set을 사용할때마다 우리가 추정할 $\hat{f}$가 얼마나 많이 변동할 것인가 이다. 쉽게 말해 데이터에 얼마나 의존적인가이다. **flexible할 수록**, 모델의 **variance는 늘어날 것이다**.

이론적으로, $Avg(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$ 이라는 식을 도출할 수 있다.

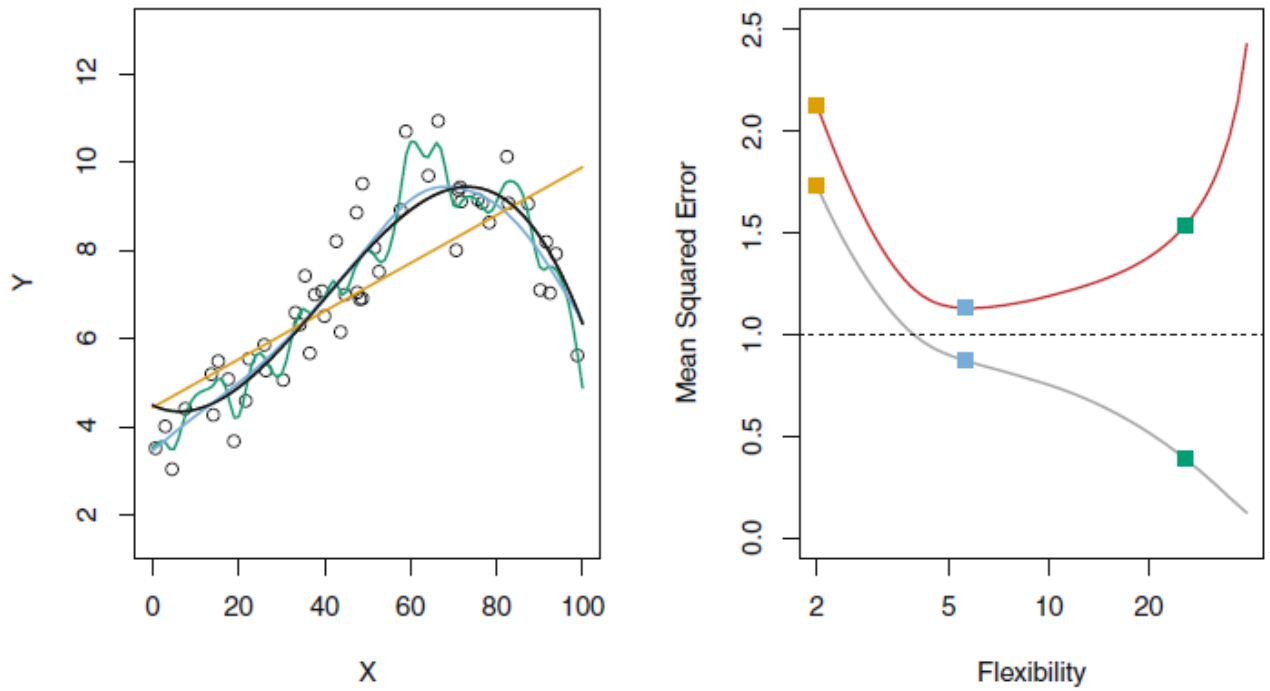
> $y = f(x) + \epsilon$ 이고, ϵ은 위에서 언급했듯이 해당 모델에서 고려하지 못한 irreducible error이다.
>
> $E(\epsilon) = 0, \therefore E(y) = f(x), Var(y) = Var(\epsilon), \because f(x)$는 unknown fixed function의 한 값, 즉 상수.
>
> $Bias(\hat{f}(x)) = E(\hat{f}(x) - f(x)) = E(\hat{f}(x)) - f(x)$
>
> $E[y - \hat{f}(x)]^2 = E[y^2 - 2y\hat{f}(x) + \hat{f}(x)^2]$
>
> $= E(y^2) - 2E(y\hat{f}(x)) + E(\hat{f}(x)^2)$
>
> > $E(y^2) = Var(y) + [E(y)]^2$ $(Var(X) = E(X^2) - (E(x))^2)$
> >
> > $= Var(\epsilon) + (f(x))^2$
> >
> > $E(y\hat{f}(x)) = E[(f(x) + \epsilon) * \hat{f}(x)]$
> >
> > $= E[f(x) * \hat{f}(x) + \epsilon * \hat{f}(x)]$
> >
> > $= f(x)E(\hat{f}(x)) + E(\epsilon * \hat{f}(x))$
> >
> > $= f(x)E(\hat{f}(x)) + E(\epsilon) * E(\hat{f}(x)), \because$irreducible error ϵ와 우리가 추정한 값 $\hat{f}(x)$은 $indep.$
> >
> > $= f(x)E(\hat{f}(x))$
> >
> > $E(\hat{f}(x)^2) = Var(\hat{f}(x)) + [E(\hat{f}(x))]^2$
>
> $\therefore E[y - \hat{f}(x)]^2 = E[y^2 - 2y\hat{f}(x) + \hat{f}(x)^2] = E(y^2) - 2E(y\hat{f}(x)) + E(\hat{f}(x)^2)$
>
> $= Var(\epsilon) + (f(x))^2 - 2[f(x)E(\hat{f}(x))] + Var(\hat{f}(x)) + [E(\hat{f}(x))]^2$
>
> $= Var(\epsilon) + Var(\hat{f}(x)) + [f(x) - E(\hat{f}(x))]^2$

> $= Var(\epsilon) + Var(\hat{f}(x)) + Bias(\hat{f}(x))^2.$

이 중 $Var(\epsilon)$은 우리가 고려하지 못한, irreducible error의 Variance이다. 즉 우리의 목표인 **test MSE를 줄이기 위해서는 모델의 Variance와 Bias를 모두 가능한 낮춰야** 한다. 그러나, 이는 앞의 설명에서 예상할 수 있듯이 쉽지 않다.

Bias-Variance Trade-off란, **모델의 flexibility에 따라 bias와 variance는 필연적으로 trade-off 관계에 있다는 것**을 나타낸다.

빠른 이해를 위해 그림을 보자.



위의 왼쪽 그림에서, 하얀색(?검은색?)점은 우리가 관찰한 실제 값들이고, 검은색 선은 실제 X와 Y와의 관계, 즉 f이다. (물론 해당 실제 관계는 현실에서는 미리 알 수 없다.) 그리고 황색, 하늘색, 초록색 선들은 각각 다른 flexibility를 가지고 주어진 training data에서 나름의 f를 추정한 선들($\hat{f}$)이다. 황색선이 가장 덜 flexible하고, 하늘색이 중간정도로 flexible하고, 초록색이 가장 많이 flexible하다. 주어진 자료를 가장 잘 설명(혹은 예측)하는 선은 초록색, 즉 주어진 자료에 맞춰서 구불구불하게 꼬은 선일 것이다. (이는 smoothing spline이라는 방법을 통해 그은 선이다. 7장에 나온다.) 그러나 실제의 관계, 즉 검은색선을 가장 잘 예측하는 선은 파란색 선이다.

이는 오른쪽 그림을 통해 잘 나타나 있는데, 오른쪽 그림에서 U자 곡선은 test MSE, S자 곡선은 training MSE이다. x축은 flexibility를 나타내는 정도를 의미한다.(df에 대해서 어케 해석해야 될까요???) 모델이 flexible해지면 해질수록, 즉 주어진 데이터에 맞추어 꼬불꼬불해지면 해질수록 training MSE는 지속적으로 감소한다. 그러나 test MSE, 즉 우리의 진짜 목표는 처음에는 감소하다가, 어느순간 다시 증가하게 된다. training MSE는 작은데 test MSE는 커지는 이러한 상황은 **overfitting**이라 부른다

다시 한번 상기. $Avg(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$

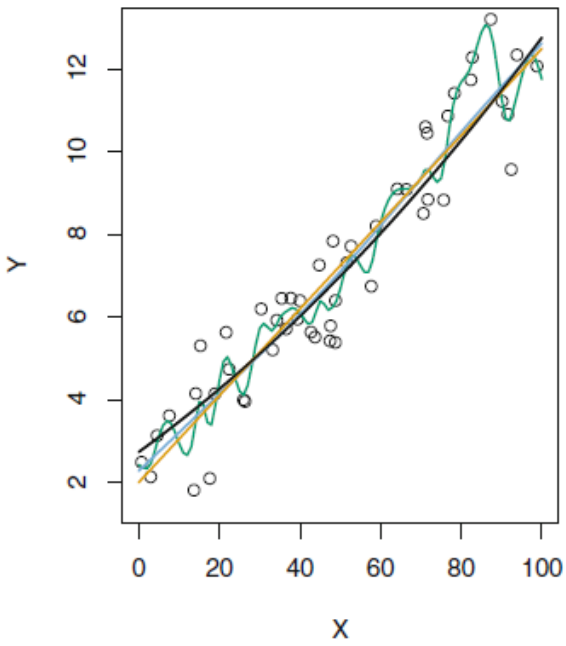
우리의 목표인 test MSE를 줄이려면, 줄일 수 없는 $Var(\epsilon)$을 제외하고, 우리 모델의 Variance도 가능한 줄어야 하고, Bias도 가능한 줄어야 한다.

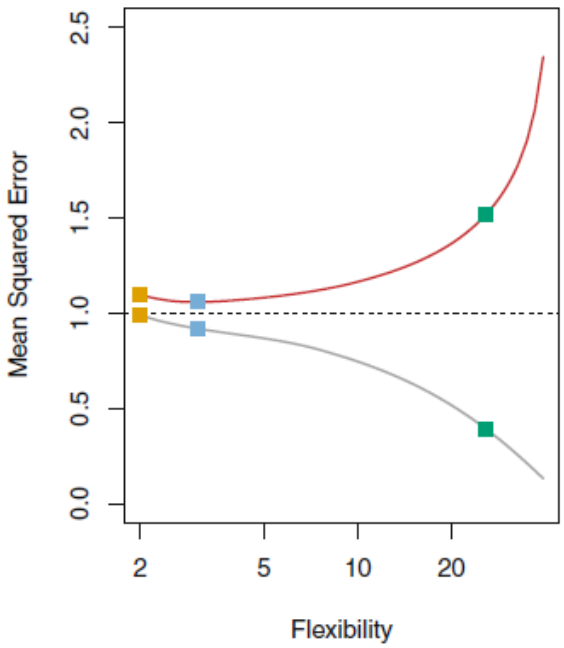
더 flexible한 method를 사용하게 되면 bias는 줄고, variance는 늘어난다. 다만, 처음에는 bias가 줄어드는 정도가 variance가 늘어나는 정도보다 더 컸기에, 더욱 flexible한 방법을 사용하는 것이 더 낮은 test MSE를 만들어 내었다. 그러나 어느 시점, 즉 적당한 수준을 넘어 더욱 flexible한 방법을 사용하고자 하면 bias가 줄어드는 정도보다 variance가 늘어나는 정도가 더 크기에, 결과적으로 test MSE는 더 늘어나게 된다.

그럼, **어느 정도 flexible한게 '적당한' flexibility인가?**

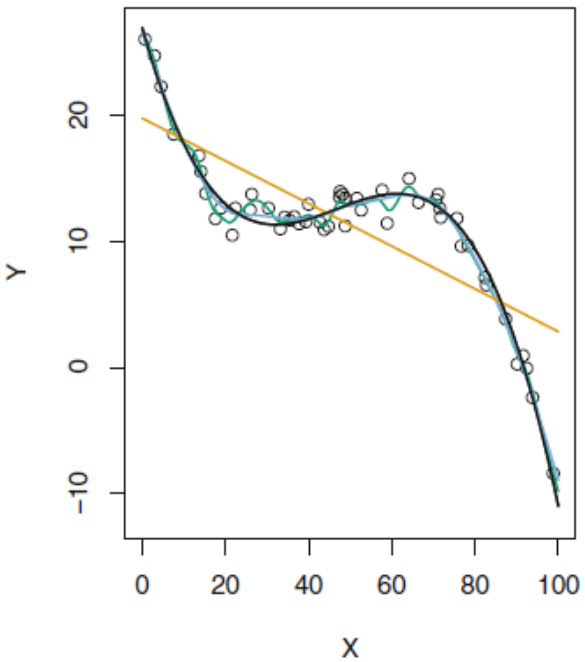
답은 **'그때그때 다르다'** 이다.

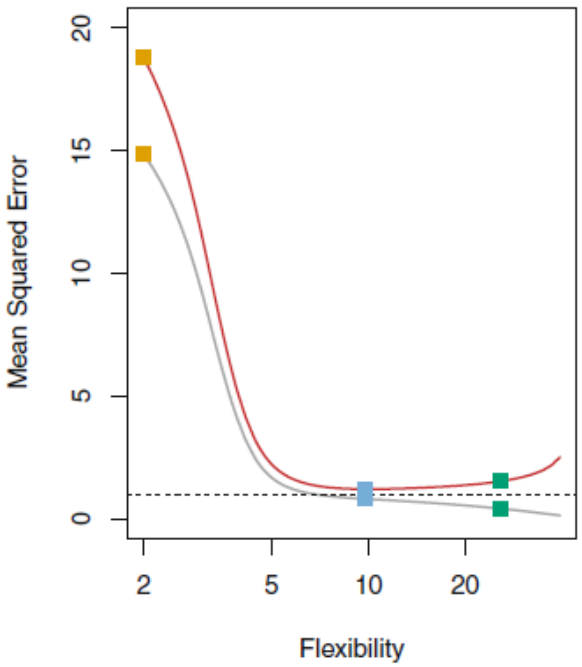
1. 만약 실제 truth, 즉 f가 선형에 거의 가까운 관계였다면, 아주 약간만 flexible한 방법이 최적의 결과를 가져올 것이다.





2. 반대로 만약 실제의 f가 그 자체로 구불구불한 모양이 'truth'라면, 자연스레 더욱 flexible한 모델이 최적의 결과를 가져올 것이다.





이 처럼, bias와 variance는 trade-off의 관계이고, 어느정도의 수준이 제일 좋은지는 상황에 따라 다르기 때문에, 분석자는 항상 이를 염두하고 어느 모델이 좋을지를 생각해봐야 한다.

Assessing Model Accuracy at Classification

앞서서는 회귀, 즉 regression의 경우에 대한 모델 평가방법에 대해 말했다. 그럼 이제 classification의 경우에 대한 모델 평가 방법에 대해 알아보자.

- Q. Regression과 Classification의 차이는?
- A. 간단하다. 반응변수(y)가 양적변수일때는 regression 문제, 반응변수가 질적변수일때는 classification 문제이다.

p개의 예측변수 $x_1, x_2, .., x_p$ 가 있을때 해당 자료가 어느 class에 속할지를 예측하는 것이 classification문제이다. 예를 들면 몸무게, 키가 주어졌을때 해당 사람이 '남자'에 속할지, '여자'에 속할지를 맞추는 것이다.

여기서의 평가방법도 위의 MSE와 개념적으로 크게 다르지 않은데, 여기에서는 실제 답, 즉 실제 class에 맞게 분류를 했는지 안했는지를 보고 오류율을 구하면 된다. 이를 식으로 나타내면 다음과 같다.

$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$

여기서 $\hat{y}_i$는 '몇번째 클래스에 속하는지'에 대한 class label이다. $I()$는 indicator function으로, 쉽게 말하자면 안의 조건문이 참이라면 1, 거짓이라면 0을 반환한다고 보면 된다. 따라서 위의 식은 '전체 n개중 몇개나 틀리게 label을 부여했는가'를 의미한다 보면 된다.

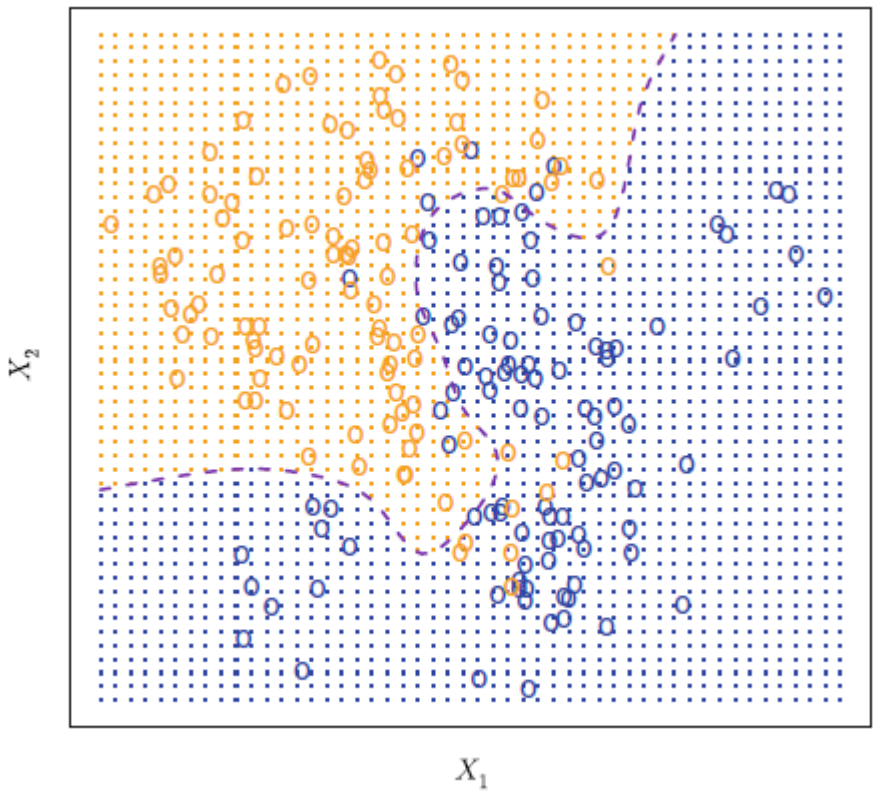
역시 여기서도 결국 중요한건 test data에 대한 error rate, 즉 $Avg(I(y_0 \neq \hat{y}_0))$이다.

Bayes Classifier

test data에 대한 error rate를 줄이기 위해서는 Bayes Classifier라 불리는 아주 간단한 원리의 classification을 하면 된다. Bayes Classifier는 x_0라는 input이 주어졌을때 Y가 어느 클래스에 속할지 확률(즉 conditional probability)을 구하고 그 확률이 최대가 되는 class에 분류를 하는 것이다. 이를 식으로 나타내자면 다음과 같다.

$$Pr(Y = j|X = x_0), (j = 1, 2, \ldots, K)$$

총 K개의 클래스가 있을때, x_0라는 condition, 즉 조건이 있을때 해당 자료가 1번째 클래스에 속할 확률, 2번째 클래스에 속할 확률,..., 등을 전부 구해, 속할 확률이 가장 큰 클래스에 배정해주면 된다.



대충 이러한 그림으로 나오는는데, 이는 x_1, x_2가 주어지는 경우의 classification이다. 모든 점에 대해 노랑색 클래스에 속할지 파랑색 클래스에 속할지 확률을 구하고 그에 따라 classification해준 것이다. 물론 해당 확률은 그 input일때 해당 클래스에 속하는 경우가 더 많다는 것이다. 특정 input의 경우 100% 한 클래스에 속할수도 있겠지만(그림에서 맨 오른쪽, 혹은 맨 아래의 점들은 모두 파란색이므로 이때의 conditional probability

$$Pr(Y = \text{파랑} | X = x0)$$

는 1이다.) 애초에 모집단의 분포 자체가 섞여 있을 수도 있다(그림에서 정 중앙라인 부분). 그럴 경우 100% 그 클래스에 속하는 것은 아니고, 해당 경우

$$Pr(Y = \text{파랑} | X = x0)$$

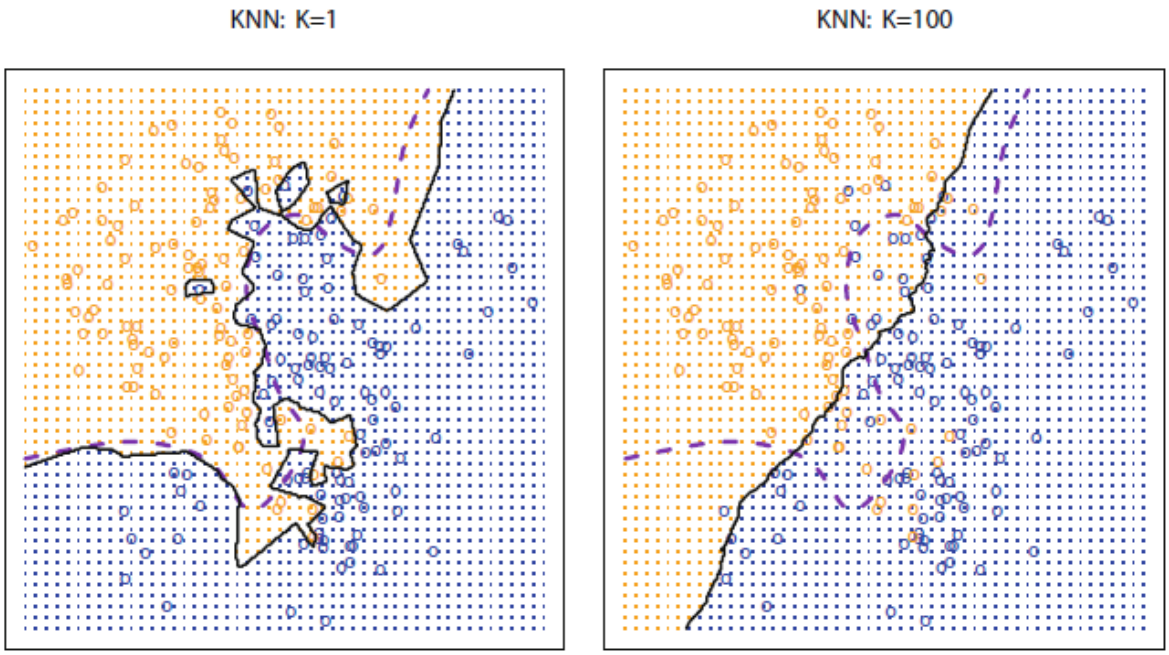
는 1보다 작다.

KNN(K-Nearest Neighbors)

그러나 Bayes Classification에서 구하고자 하는 conditional probability는 실제 구하기가 힘들다. 우리는 실제 모집단의 분포를 모르기에, 실제의 conditional probability도 알 수 없기 때문이다. (이는 conditional probability를 구하는 Bayes Theorem에 대한 이해와 부가적인 설명 필요한데, 다소 세부적인 내용이라 판단되어 참조링크만 걸어둔다. unbiased한 conditional probability를 구하기 위해선 엄청나게 많은 수의 데이터가 필요하기 때문이라는 것. Bayes and Naive Bayes Classifier)

이에 따라 우리는 conditional probability를 추정하고자 하는 다양한 방법들을 사용하는데, 가장 대표적인 것이 KNN이다. 여기서 K는 양수인 하나의 숫자인데, input이 들어왔을때 우리의 training data를 기준으로 K-nearest, 즉 K개의 가장 가까운 이웃 데이터를 살펴보고(가까운을 무슨 기준으로 평가할까?=>분석자의 판단. 코사인유사도 던 유클리드놈이던) 이를 토대로 conditional probability를 계산하는 것이다. 다시 위의 그림을 기준으로 설명하자면, K=5일 경우 가장 가까운 5개의 데이터를 살펴보고 데이터가 각각 ['노랑색', '노랑색','파랑색', '파랑색','파랑색'] 클래스 였다면 파랑색 클래스에 속할 것이라 판단하는 것이다.

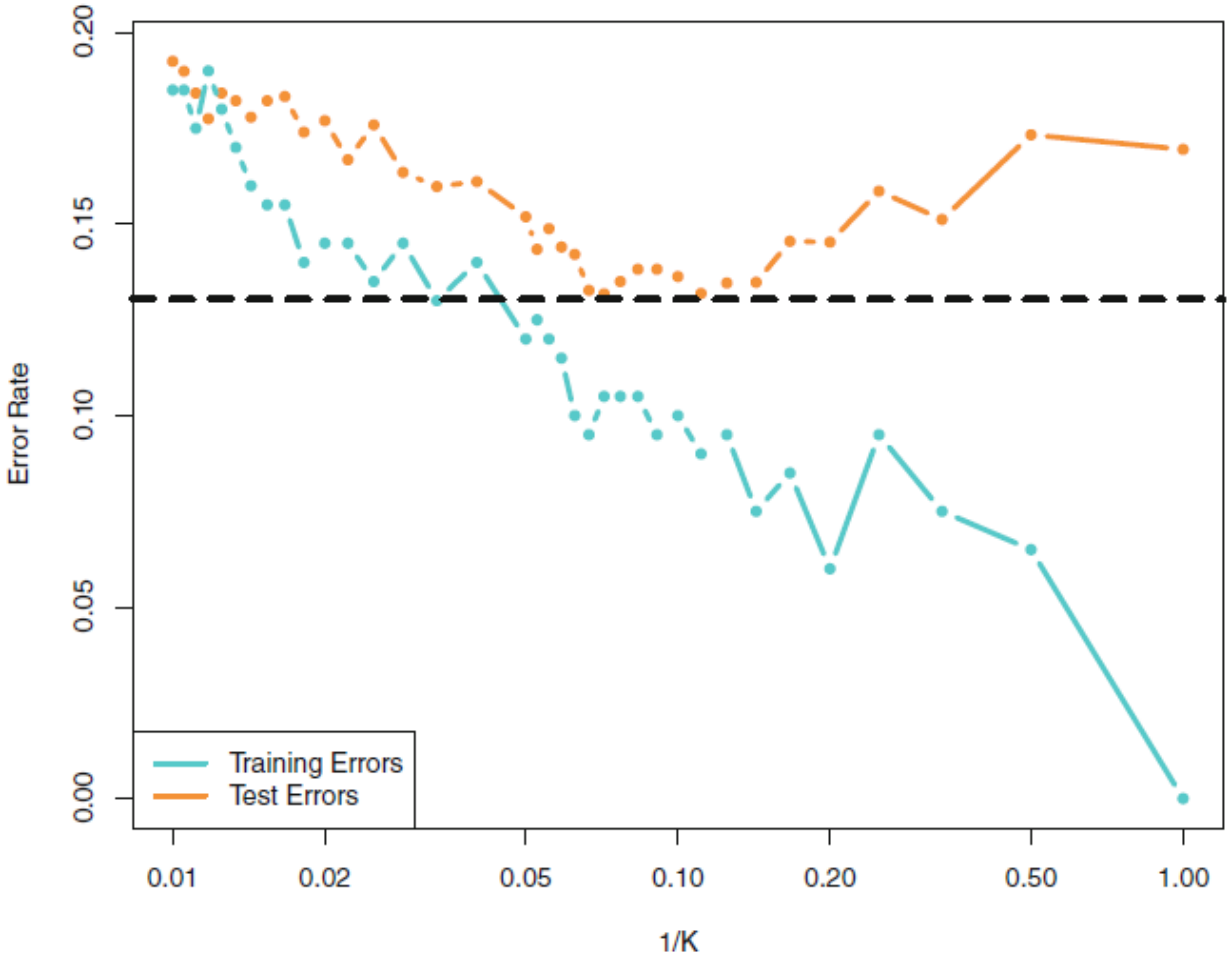
K를 몇으로 설정할 것인지에 따라 결과가 천차만별인데, 여기에선 K를 몇으로 하느냐에 따라 위에서 언급한 flexibility의 개념이 결정된다.



위의 그림에서 점선은 Bayes Classifier이고 왼쪽 그림은 K=1인 경우의 KNN, 오른쪽은 K=100인경우의 KNN결과이다.

K=1일 경우 단 한개의 점만을 보고 판단을 하여 결과적으론 주어진 데이터에 따라 쉽게 변동하는 flexible한 모델이 되고, K=100개일 경우 엄청 많은 데이터를 보고 판단하기에, 매우 variance가 적은 flexibility가 적은 모델이 되서 그림에서 처럼 사실상 직선에 가까운 형태가 된다. (flexibility가 커지면 Variance는 커지고 Bias는 줄어든다는 것을 다시 상기하자.)

이 경우에도 역시나 Bias-Variance Trade-off의 개념이 적용되며, 아래 그림을 보면 마찬가지로 '적당한'수준의 flexibility을 넘을 경우 Test Error는 늘어나는 U자 형태임을 볼 수 있다.



즉 여기서도 '적당한'수준의 flexibility를 찾아 모델을 수립하는 것이 중요한 관건이다. 이 과제를 풀기 위한 다양한 방법들이 논의되고 있는데, 이는 5장에서 다룰 것이다.