3. Linear Regression

1. 선형회귀 분석이란?

선형 회귀분석은 여러 종속변수들의 선형결합으로 반응변수를 나타내는 것.

쉽게 생각하면 x, y의 산점도를 그려서 산점도를 잘 설명하는 적당한 직선을 찾아내는 것.

2. 선형 회귀분석의 기본 가정

1) 반응변수와 종속변수는 선형성을 띈다.

2) 오차항은 을 만족한다.

3) 독립변수 간에 독립성을 유지한다.

3. Least Square Estimate방법으로 모수추정

위 수식은 일반적인 다중 선형회귀 모형이다.

이를 matrix식으로 나타내면

최소 제곱 추정법(LSE)을 통해 오차 제곱합이 최소가 되도록 하는 값들을 추정하면

를 0으로 만드는 을 구하면

행렬 notation을 사용하면 더욱 간편하게

로 나타낼 수 있다.

위의 식을 normal equation이라 한다.

숙제) 정규 방정식(normal equation)을 직접 손으로 풀기

식을 에 대해 편미분해서 최종 을 구하는 것 하나,

행렬 notation을 사용해서 푼 것 하나.

총 2개의 방식으로 LSE구하는 과정을 직접 손으로 푼 다음 과제에 사진으로 첨부

4. 만약 가 singular-matrix라면?

여기서 만약 변수들 간에 다중 공선성이 있다면, 각 column들이 linear dependent하게 된다.

따라서 의 행렬식이 0에 가깝게 나타난다. (이런 매트릭스를 ill-matrix라고도 부름)

의 행렬식이 0은 아니지만, 0에 가까우면 역행렬이 굉장히 불안정하게 나타난다.

이럴 경우 이론적(Theoretiacal)으론 역행렬이 존재하지만, 계산적(computational)으론 매우 불안정한 역행렬이 나타난다.

따라서 변수들 간에 다중 공선성이 존재하면, LSE방법으로 을 추정하는 것은 잘못되었다.

이를 해결하는 방법으로 LASSO(라쏘), ridge(능형), elastic net(엘라스틱넷)모형 등을 사용하게 된다.

(아래 부분은 이론적인 내용이 다수 포함되 있으므로 수식을 하나하나 이해하기 보다는 흐름을 알아 두는 것을 추천, LASSO와 ridge를 사용하는 이유, 두 방법의 차이정도만 알아도 2학년은 충분, 3학년의 경우 꼼꼼히 읽어 보는 걸 추천)

위의 새로운 3가지 모형들은 기본적인 LSE구하는 식에 패널티(penalty)를 부여함으로써 역행렬이 존재하는 행렬로 만들어 주는 것이다

LASSO는 L1-norm을 사용하고, ridge가 L2-norm을 사용해 패널티를 부여한다.

엘라스틱넷(elasticnet)은 LASSO와 ridge의 조합으로 나타낸다.

(참고)를 이용해 뒤에 새로운 텀을 생성해서 등식을 만드는 것을 라그랑지 승수법이라 한다.

L1-norm은 절댓값을 이용한 거리, L2-norm은 제곱을 이용한 거리라고 생각하면 됨.

간단히 정리하자면, 변수가 굉장히 많은 데이터의 경우, 자기상관이 존재할 가능성이 매우 높다.

따라서 LSE가 아닌 다른 추정치를 구해야 하는데 대표적으로 LASSO, ridge등의 방법이 있다.

Ridge를 이용할 경우, 모든 변수를 사용하고, 각 계수들을 줄이는 효과가 있고,

LASSO는 변수의 개수를 줄이는 효과가 있다.

Elastic net은 데이터의 양이 클 때 잘 돌아가고, ridge와 LASSO의 하이브리드 버전이라고 생각하면 된다. Elastic net에서 알파가 1이면 LASSO, 0이면 ridge를 의미한다.

(추가)이 부분의 설명은 굉장히 어려운 내용이고, 회귀책에서 내용을 따와 쓰는 글이므로 이해하기 보다는 한번 읽어보고 넘어가는 걸 추천. 3학년은 꼭 읽어 보길 바람.

ridge(능형)회귀는 (normal-equation)에서 가 역행렬이 존재하지 않으면, 대신

의 역행렬을 구한다.

다시 말해서, 의 역행렬이 존재하지 않으므로, 를 더한 의 역행렬을 사용해 의 추정치를 구한다. 여기서 는 작은 값을 사용한다.

ridge(능형)회귀는 제한된 조건 하에서 구한 최소 제곱 추정치이다.

구체적으로, 라는 조건 하에서 구한 오차 제곱 합 를 최소화하는 추정치이다.

변수가 하나인 단순선형 회귀의 경우를 생각해 보자.

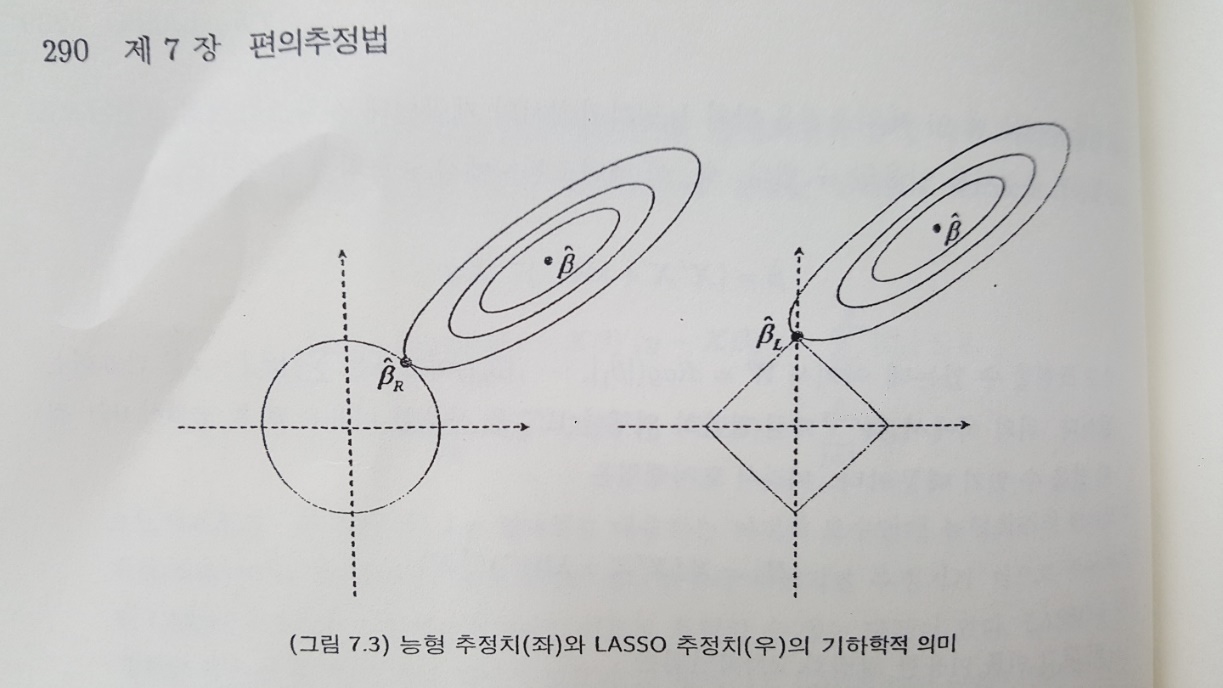
와 을 축으로 하는 2차원 평면상에서, 가 의미하는 것은, 원점을 중심으로 하고, 반지름이 c인 원을 의미한다.

그리고 는 LSE를 중심으로 가지는 타원형의 모습을 띈다.

여기서 원점을 중심으로 하는 원과 LSE를 중심으로 가지는 타원의 접점이 ridge의 해가 된다.

Ridge는 L2-norm을 사용하지만, LASSO는 L1-norm을 사용한다.

Ridge의 제한된 조건은 , 즉 원점을 중심으로 하는 원이지만, LASSO의 제한된 조건은 원점을 중심으로 하는 기울어진 정사각형 모양이다.



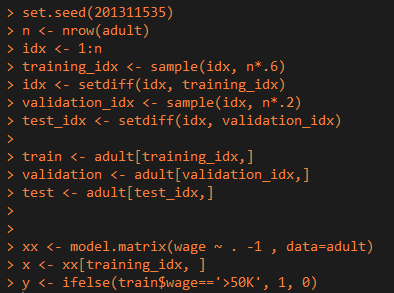
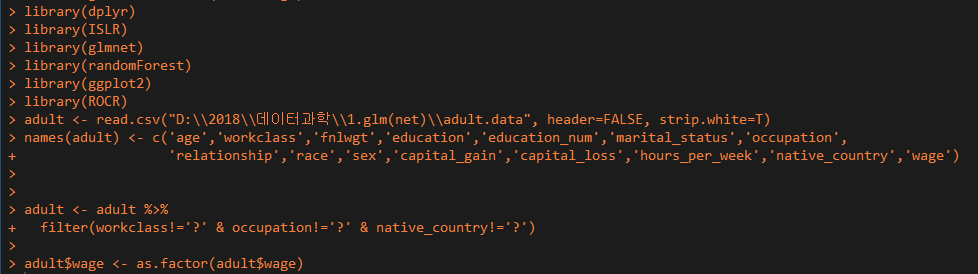
위의 사진에서 볼 수 있듯이, LASSO추정치의 해는 1사분면 위의 점보다 축 위의 점에서 나타날 가능성이 높기 때문에 그림처럼 y축 위의 점에서 접점을 가진다면 x축의 계수는 0으로 나타난다.

정리하면 ridge는 원과 타원의 접점이 1사분면 위에서 생길 확률이 높기 때문에 변수선택의 의미는 지니지 못하고, 단지 LSE를 축소시키는 기능만 한다.

반면에 LASSO는 하나의 축 위에서 접점이 생길 가능성이 높기 때문에 변수선택의 효과 또한 가지고 있다.

5. R코드(glmnet)

0) 데이터 불러오기

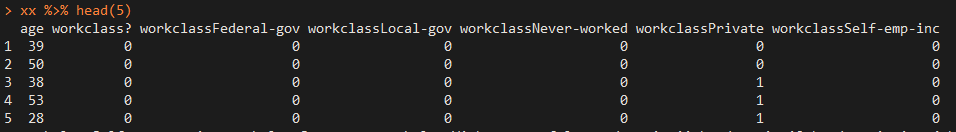
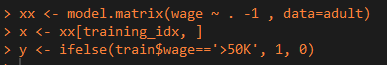


데이터를 불러와서 ‘?’가 있는 관측치들을 모두 제거 후 train, validation, test set 생성

1. model.matrix 생성

glmnet패키지는 값들을 직접 지정해서 함수를 만들 수 있다.

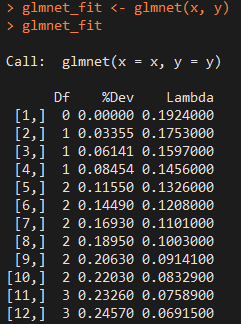
하나 주의할 점은 R의 내장함수인 lm()함수는 범주형 변수인 경우 알아서 가변수화 작업을   
해주지만, glmnet패키지는 직접 model.matrix를 생성한 뒤 회귀분석을 실시해야 한다.



Model.matrix를 통해 생성한 xx를 확인해 보면, 범주형 변수인 workclass의 범주들에 해당하는 새로운 변수가 column으로 생성된 것을 볼 수 있고, 각 관측치 마다 0,1로 코딩되어 있는 것을 확인할 수 있다.

Model.matrix(wage ~. -1, data=adult)에서 -1을 넣은 이유는 절편항을 제거한 model.matrix를 생성하기 위함. -1옵션을 빼면 절편들이 첫 번째 column에 들어간다.

1. Generalized Linear Model

  
가장 기본적인 함수인 glmnet을 이용해서 glm을 생성한 결과.

glmnet은 디폴트로 즉, LASSO모형을 적합시킨다.

glmnet\_fit을 실행시켜 보면 위와 같은 결과를 출력해 준다.

이 결과는 람다값을 변경해 주면서 그때의 자유도와 %Dev를 보여주고, $Dev는 현재의 모델이 얼마나 많은 변이를 설명하는지를 보여준다. 람다값이 내림차순으로 정렬되어 있으며 첫 번째 행은 가장 간단한 y=상수 모형을 적합한 결과.

모형의 자유도가 0, 사용된 변수의 개수가 0개이므로 %Dev=0이다. 가장 간단하고 쓸모없는 모형

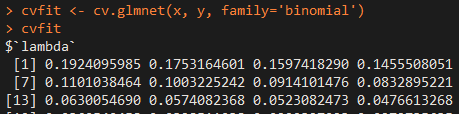
마지막 행은 가장 람다값이 작은 모형, 사용 가능한 모든 변수를 사용한 모델

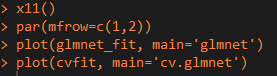
glmnet함수를 통해 여러 람다값에 따른 모형을 확인해 볼 수 있었다.

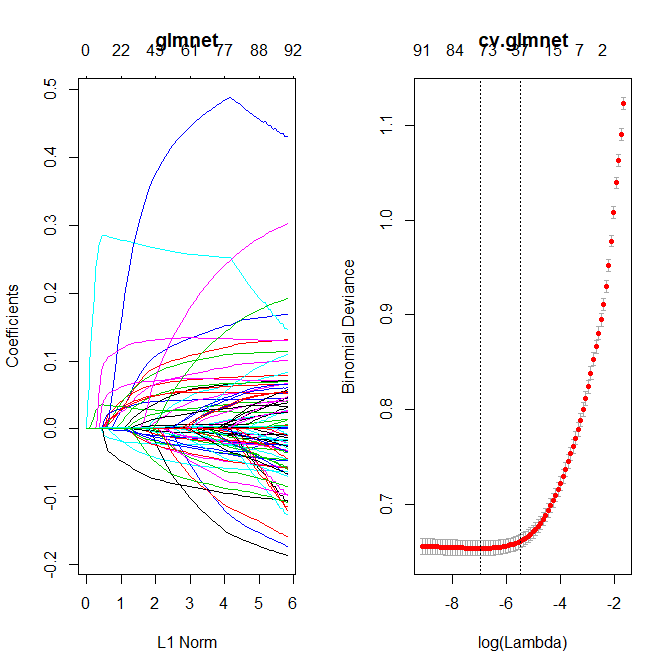
이제 어떤 람다값을 선택해야 하는지의 문제가 남아있는데 이는 cv.glmnet함수를 이용해 자동으로 선택하게 하자.

cv.glmnet은 cross-validation을 해주는 함수로써 최적의 람다값을 알아서 구해준다.

(숙제 : cross-validation이 어떤건지, k-fold교차검증이 무엇인지 간단하게 서술하기)



cv.glmnet은 glmnet으로 돌렸을 때와는 약간 다른 형태의 결과를 보여주고, 결과값도 약간 다름. 



glmnet과 cv.glmnet을 plot시켜본 결과 그림도 서로 다른 것을 확인할 수 있다.

cv.glmnet에서 따로 알파를 지정하지 않았기 때문에 디폴트로 LASSO모형을 나타낸다.

x축의 위의 숫자들은 왼쪽에서 오른쪽으로 갈수록 점점 작아지는 것을 볼 수 있다.

이 숫자들은 선택된 변수의 개수를 나타낸다.

그래프에 찍히는 빨간 점은 해당하는 람다에서 cross-validation을 통해 오차들의 평균을 낸 것.

최적의 람다는 2가지 종류가 있는데 cv.glmnet그래프에서 나타나는 세로 점선에 해당하는 람다가 최적의 람다이다.

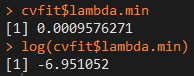
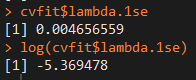
먼저 왼쪽의 점선은 lambda.min 즉, 교차검증 오차의 평균값을 최소화하는 람다.  
즉, 그래프의 최솟값일 때의 람다.

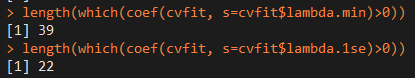
오른쪽의 점선은 lambda.1se, 교차검증 오차의 평균값이 최솟값으로부터 1-표준편차이상 떨어지지 않는 모형 중 가장 간단한 모형에 해당하는 람다.

즉, lambda.min에서 1표준오차 만큼 오차의 평균값은 크지만, 훨씬 단순한 모델을 만드는 람다.

Lambda.min은 오차를 최소화하는 모형을 만들어 준다.

Lambda.1se는 최솟값과 큰 차이는 없지만 훨씬 간단한 모델을 만듦으로써 해석하기 용이하다.



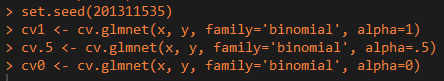


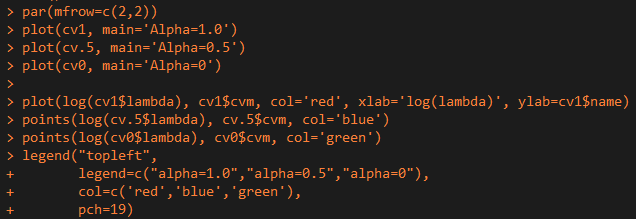
Lambda.1se를 사용할 경우, 모형의 복잡도가 훨씬 낮아지는 것을 확인할 수 있다.

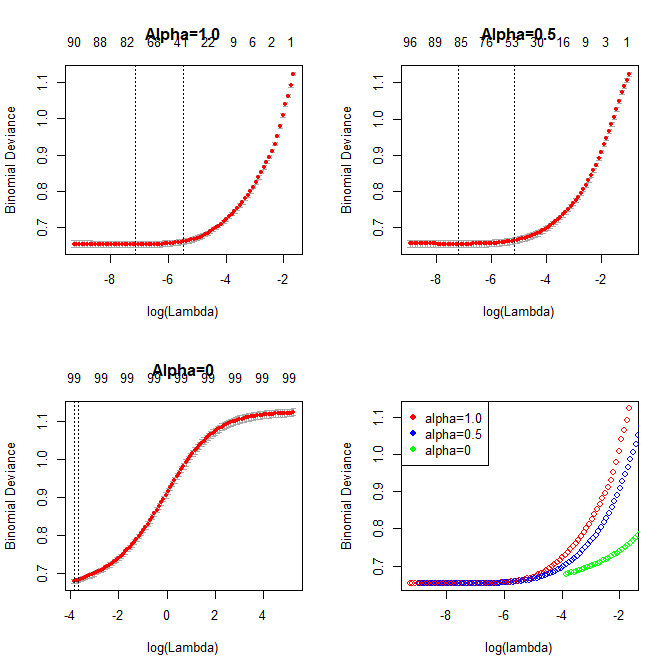
1표준오차만큼의 차이밖에 나지 않지만, 선택되는 변수의 개수는 17개나 줄어듦.

1. LASSO, rideg, elastic net의 비교

이제 LASSO, ridge, elastic net의 모형을 비교해 보자.





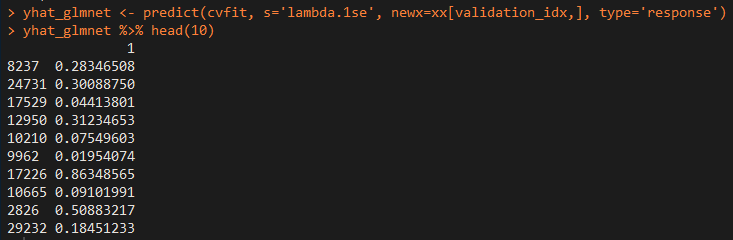
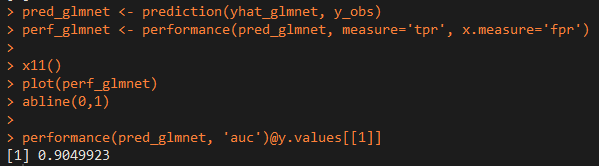
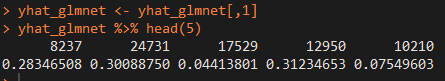
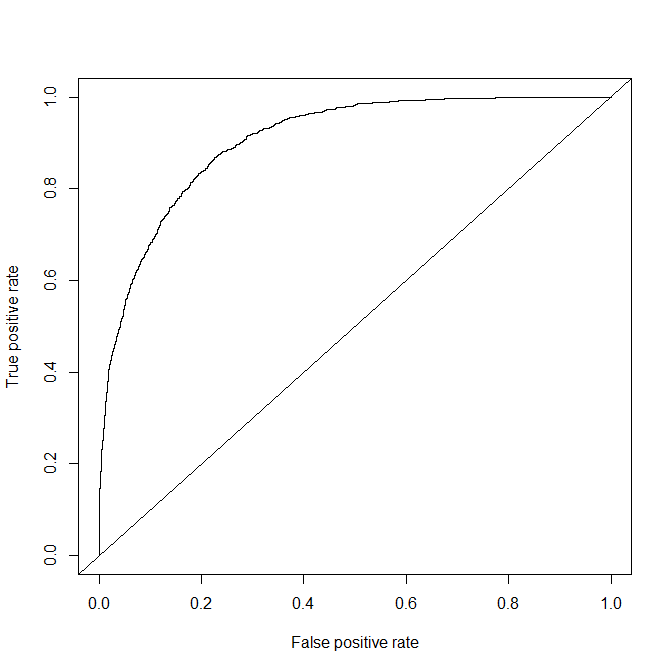


Ridge의 경우(Alpha=0)를 살펴보면 x축 윗부분의 숫자가 선택된 변수의 개수인데 어느 람다이던지 모두 99개로 같은 것을 확인할 수 있다.

이는 위에서 설명한 것처럼 ridge가 변수축소의 의미만 지니고, 변수선택의 의미를 지니지 못하기 때문이다.

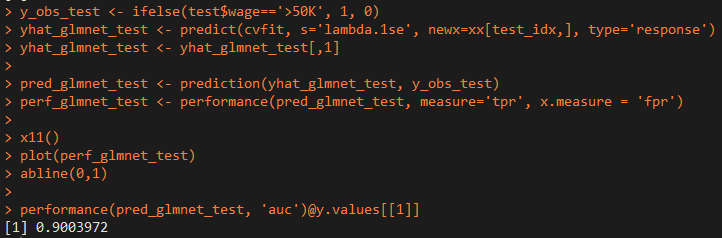
1. 예측 및 결론

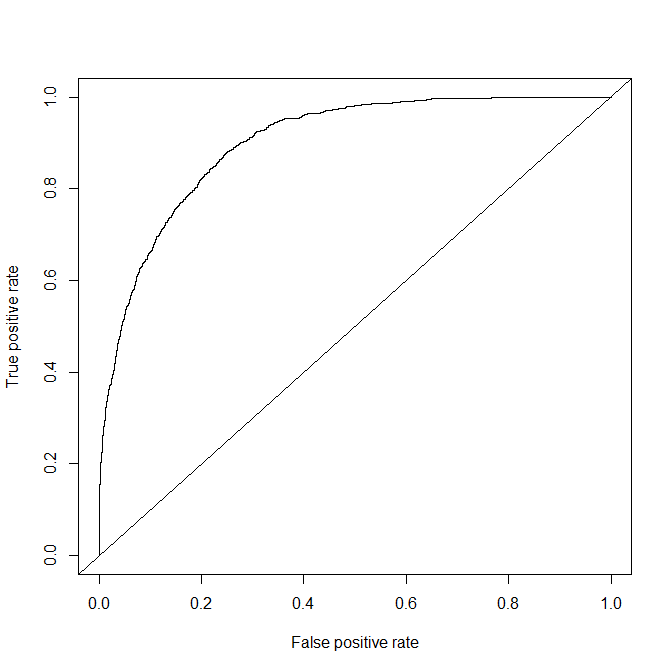
LASSO모형의 lambda.1se를 사용하는 모델을 선택하도록 하자.

Validation set에 적용시킨 결과 auc가 0.905정도로 나왔다.

Test에 적합.





Test에 적합한 최종 결과, auc가 0.9정도로 나온 것을 확인할 수 있다.

(숙제)

1. Normal-equation을 2가지 방법으로 풀기(편미분, 행렬notation) – 손으로 풀어보기
2. Cross-validation(교차검증)이 무엇인지, 그리고 k-fold교차검증은 어떤 방식으로 진행되는지 간단하게 서술
3. 지금까지 했던 의사결정나무, 랜덤포레스트와 이번에 했던 LASSO, ridge, elastic net을 모두 사용해서 validation에 적합한 결과 가장 높은 auc가 어떤 모델에서 나오는지, 그 모델을 test에 적합했을 때 어떻게 나오는지 아래 표를 참고하여 만들 것.

|  |  |  |
| --- | --- | --- |
|  | Validation의 AUC 값 | Validation의 ROC곡선 |
| 의사결정나무 |  |  |
| 랜덤포레스트 |  |  |
| LASSO |  |  |
| Ridge |  |  |
| Elastic net |  |  |

위 5가지 모델 중 validation의 AUC값이 가장 큰 모델 하나만 선택해서 test에 적합 시키고,   
그때의 AUC와 ROC곡선을 첨부할 것.

5가지 모델링을 하는 코드를 캡처해서 올릴 것.

AUC와 ROC곡선을 그리는 코드는 단순반복일 뿐이기 때문에 첫 번째로 사용하는 모델에 대해서만 코드를 캡처해서 올릴 것.

(웬만하면 이번 워드파일처럼 코드를 캡처해서 올리고 해당 결과를 캡처해서 파일에 첨부한 다음 설명을 붙이는 식으로 해주면 좋음)