QuantLab
주식회사 퀀트랩

Sentiment Analysis

감정사전 & 감정점수 만들기

김형준
Analytic Director / (주) 퀀트랩 / kim@mindscale.kr

오늘의 목표

- 감정 사전 만들기

- 감정 점수 만들기

- 상관관계 이해하기

- 회귀분석 이해하기

- 모형평가 이해하기

왜 감정분석을 하는가?

설문지의 단점

1) 조사 비용 발생

2) 미리 정해진 문항만 측정 가능

3) 사회적 바람직성 등 편향 발생

감정분석

텍스트에서 감정 단어를 추출하여 점수화

1) 기계 학습 (Machine Learning)

2) 단어 사전 기반

사전 기반 분석

장점

- 사용하기 간편

단점

- 주제에 따라 사전이 달라 짐

- 동음이의어 처리 힘듦 e.g) bank

기계학습 기반 분석

장점

- 높은 정확도

단점

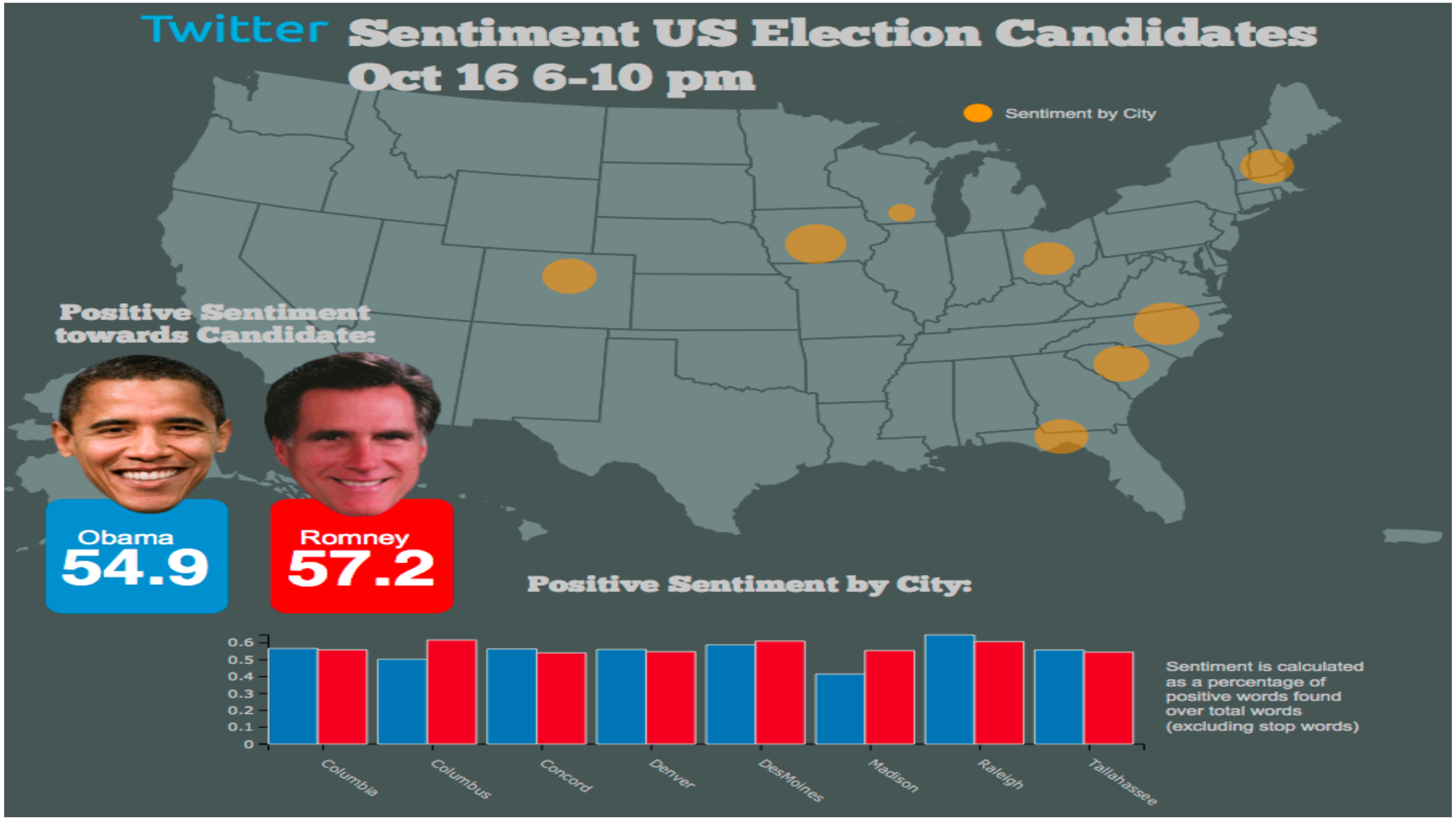
- Over-fitting 해결

- 많은 데이터 필요

예) 나이브 베이즈 / 최대 엔트로피 / 서포트벡터머신 /

랜덤 포레스트 / 토픽 모델

감정 분석 예시


Twitter Sentiment US Election Candidates
Oct 16 6-10 pm
● Sentiment by City
Positive Sentiment towards Candidate:
Obama
54.9
Romney
57.2
Positive Sentiment by City:
0.6
0.5
0.4
0.3
0.2
0.1
0
Columbia
Columbus
Concord
Denver
DesMoines
Madison
Raleigh
Tallahassee
Sentiment is calculated as a percentage of positive words found over total words (excluding stop words)

감정 분석 예시

9359.08
8942.08
8525.08
7691.08
7274.08
6857.08
6440.08
6023.08
5606.08
10.12
11.47
13.12
14.37
16.02
17.27

사전 지식

감정분석: 문장에 사용된 단어로 감정을 예측

예: "이 영화는 좀 길지만 재미있고 신난다"

- 길다 -> 부정

- 재미있다 -> 긍정

- 신나다 -> 긍정

예측 분석

예측분석

선형회귀분석

SVM

RandomForest

Deep Learning

회귀분석(선형(직선) 모형)

예시

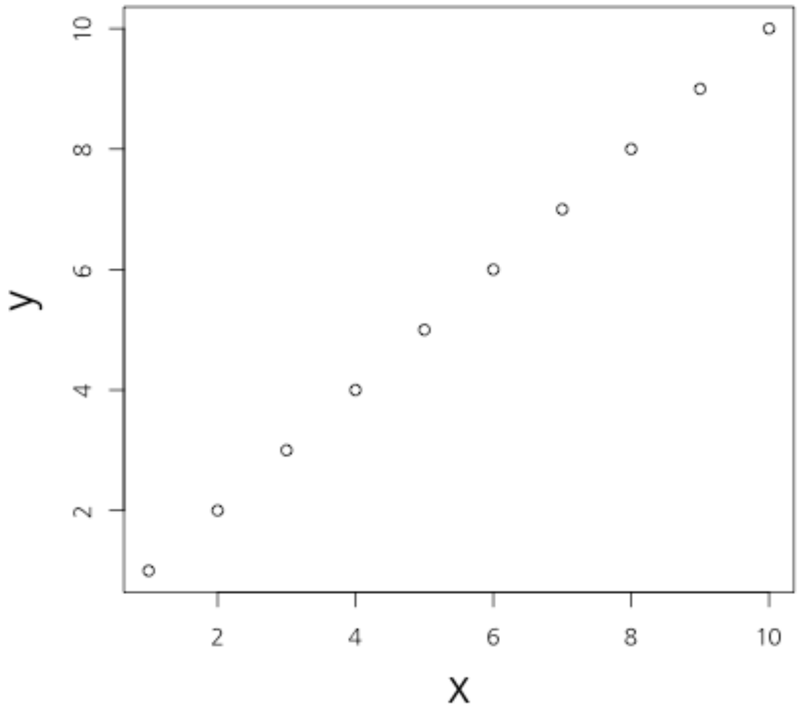
- 키가 1cm 증가할 때마다 몸무게가 1kg 증가

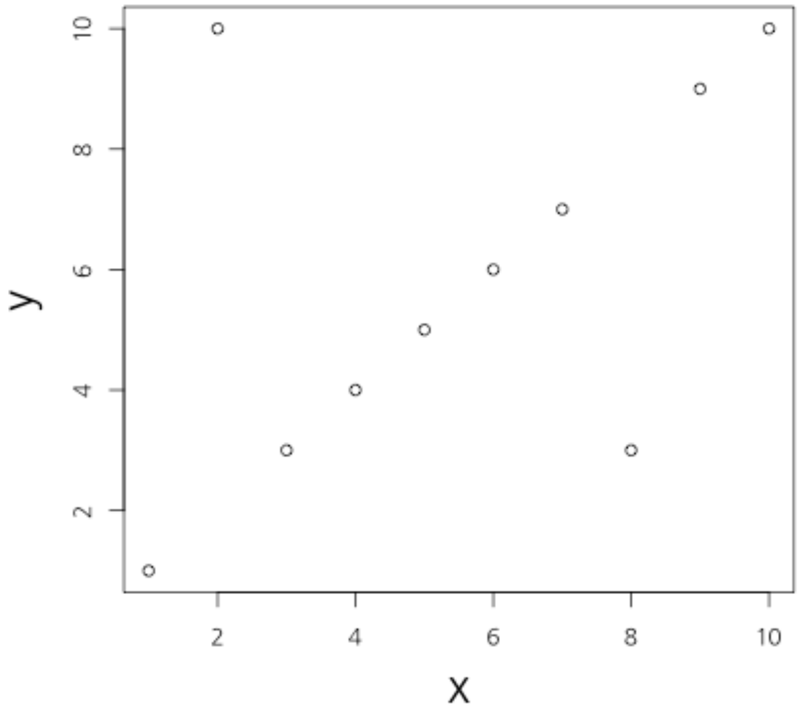
- 월 소득이 100만원 증가할 때마다 몸무게가 1kg 감소

- 부정단어가 1개 증가할 때 마다 평점 1점 감점

- 긍정단어가 1개 증가할 때 마다 평점 1점 증가

상관관계





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

```
## [1] 0.4885042
```

상관관계

x가 증가(혹은 감소)할때 y가 선형적으로 증가(혹은 감소)하는 정도

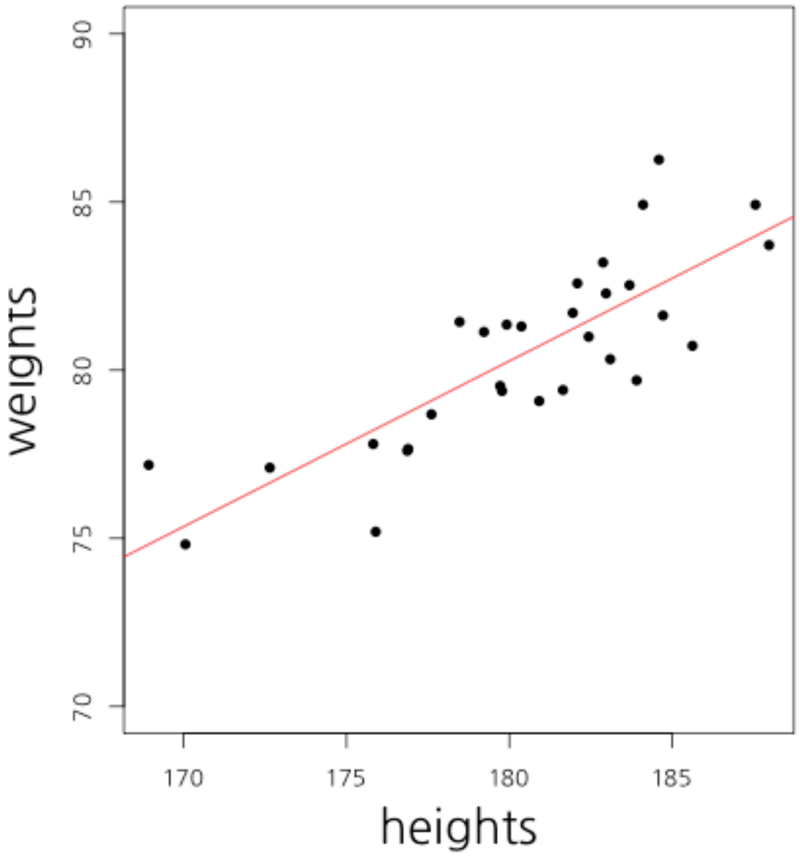
scale

키가 만약 cm라면, 키가 1cm 증가하면 몸무게는 1kg증가

키가 만약 mm라면, 키가 1mm 증가하면 몸무게는 0.1kg 증가

-> 표준화해야 한다

상관관계 및 회귀분석



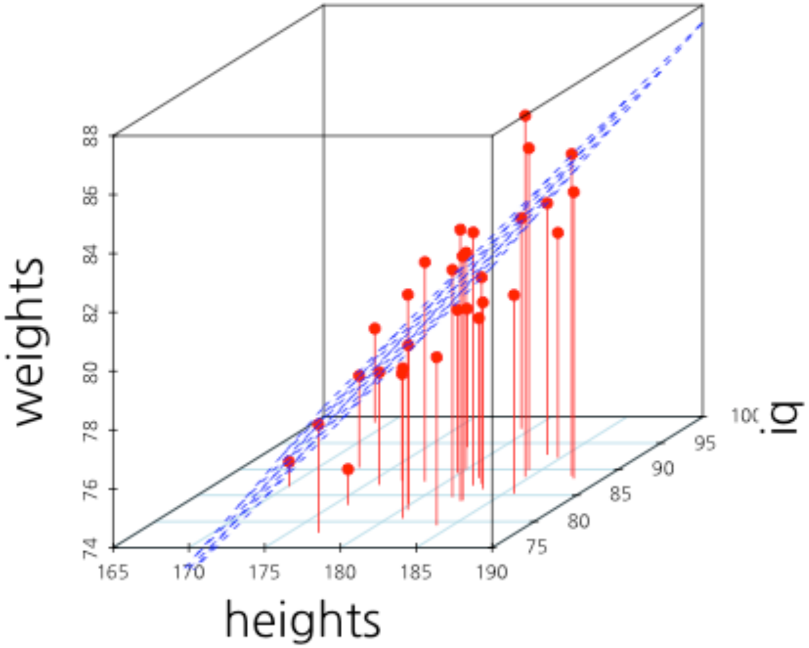
	ESTIMATE	STD. ERROR	T VALUE	PR(>|T|)
(Intercept)	-8.29	11.74	-0.71	0.49
heights	0.49	0.07	7.56	0.00

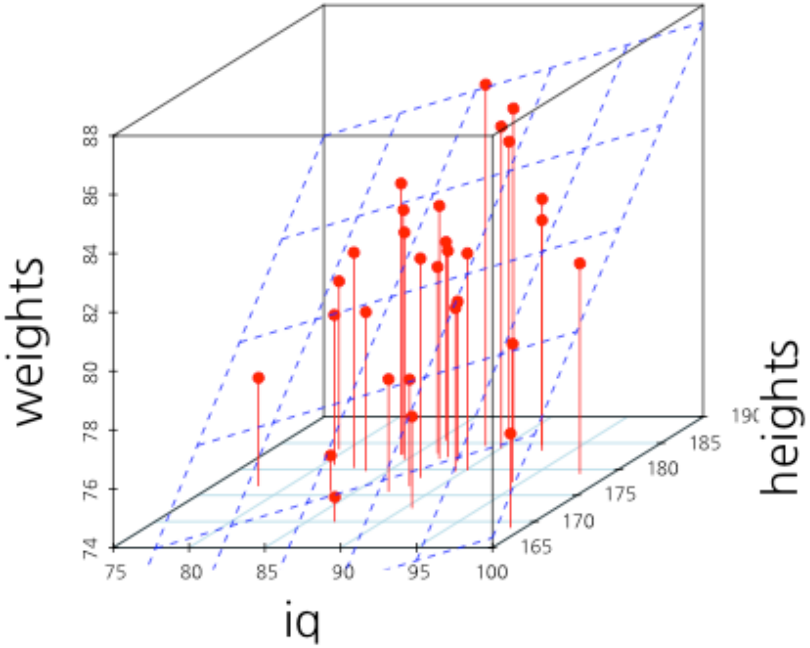
```
cor(weights, heights)
```

```
## [1] 0.8194181
```

키가 1cm 증가하면 몸무게는 .49kg 증가

X가 2개라면?





```
## [1] 0.8194181
```

```
## [1] 0.09818667
```

다중회귀분석

	ESTIMATE	STD. ERROR	T VALUE	PR(>|T|)
(Intercept)	-27.49	12.81	-2.15	0.04
iq	0.15	0.06	2.68	0.01
heights	0.52	0.06	8.72	0.00

회귀분석의 문제

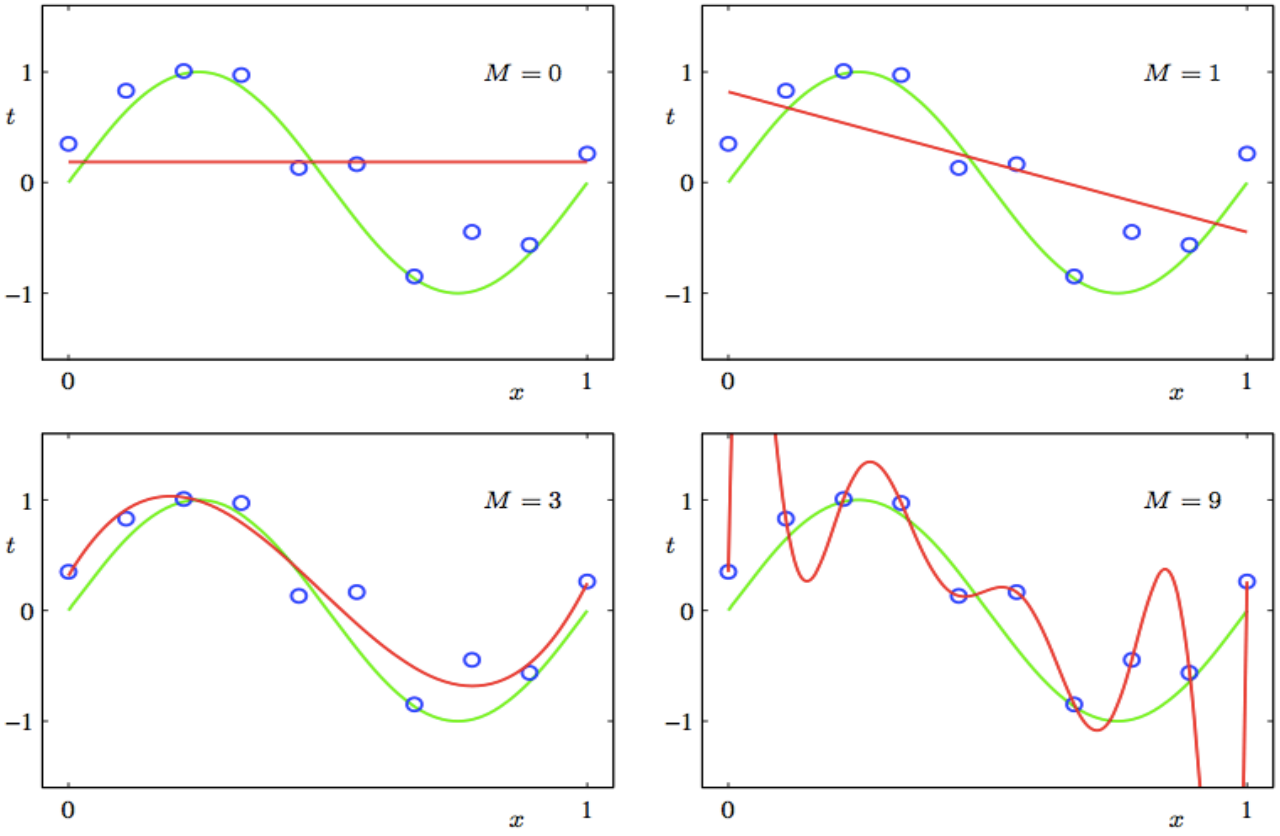
- 변수가 많아지면 과적합(overfitting)이 발생

- 회귀계수가 극단적으로 커지거나 작아짐

- 예측력이 떨어짐

- 과적합을 막아주는 방법이 필요

Over-fitting



Over-fitting

Over-fitting을 피하는 방법들

- Penality of Model Complexity (MSE 보정)

- Regulization (Lasso, Ridge, Elastic Net)

- Bayesian

- Drop Out, Bagging, Feature Bagging

과적합을 막는 법

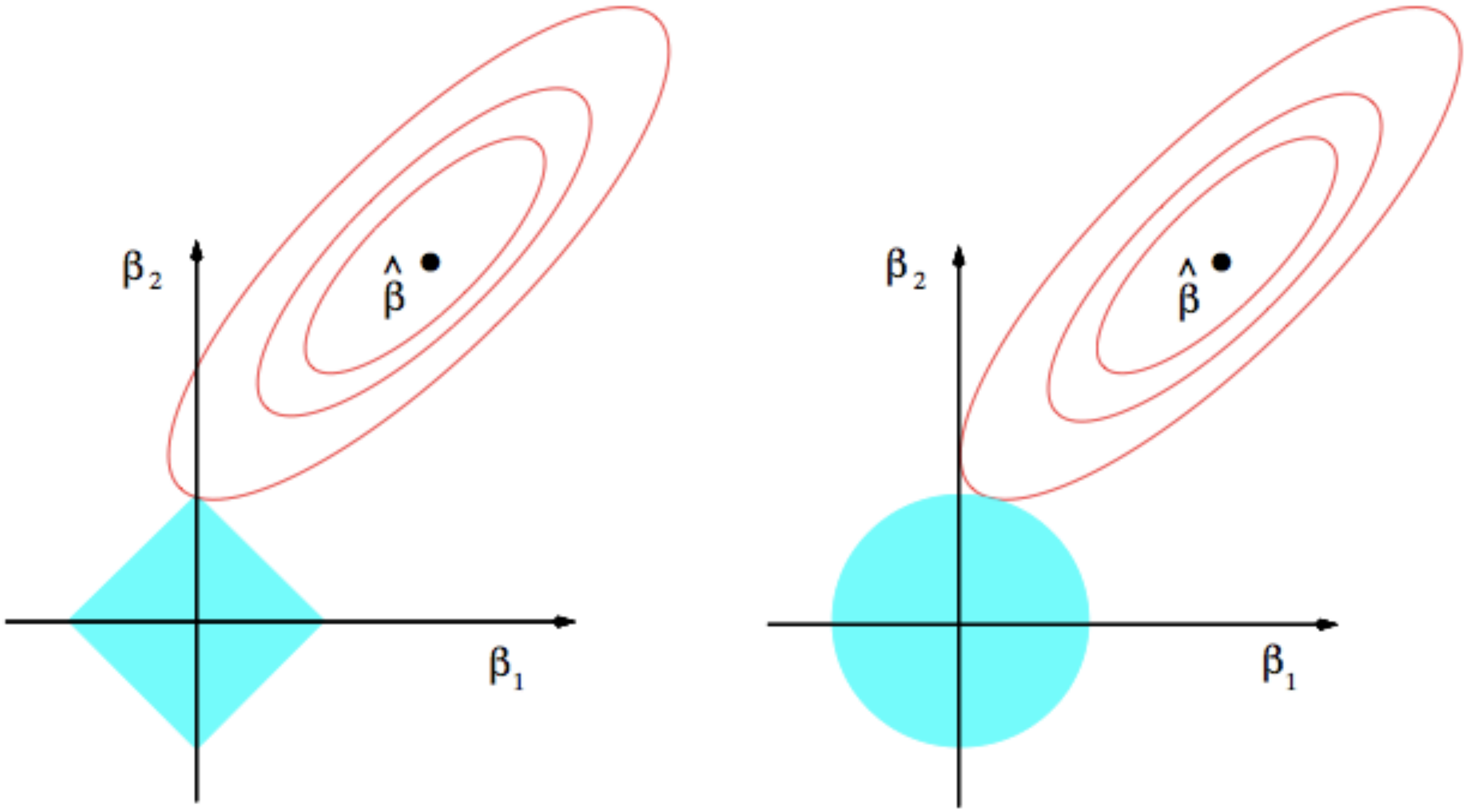
- 라쏘(lasso): 작은 회귀계수를 0으로 만듦

- 릿지(ridge): 전반적으로 회귀계수를 줄여줌

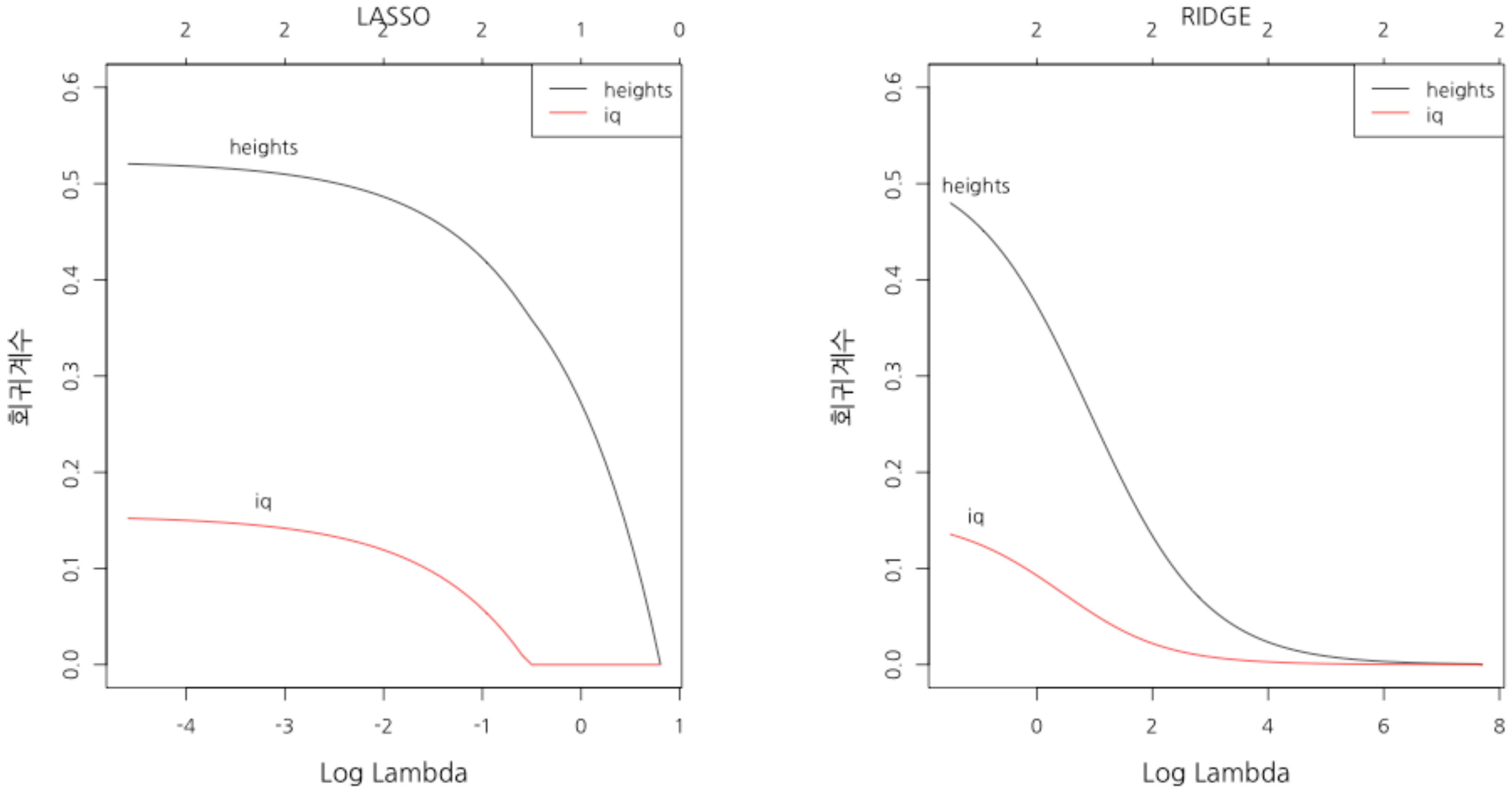
- 엘라스틱넷(elastic net): 라쏘 + 릿지

- 감정분석에서 라쏘를 쓰면 감정 단어만 추출됨

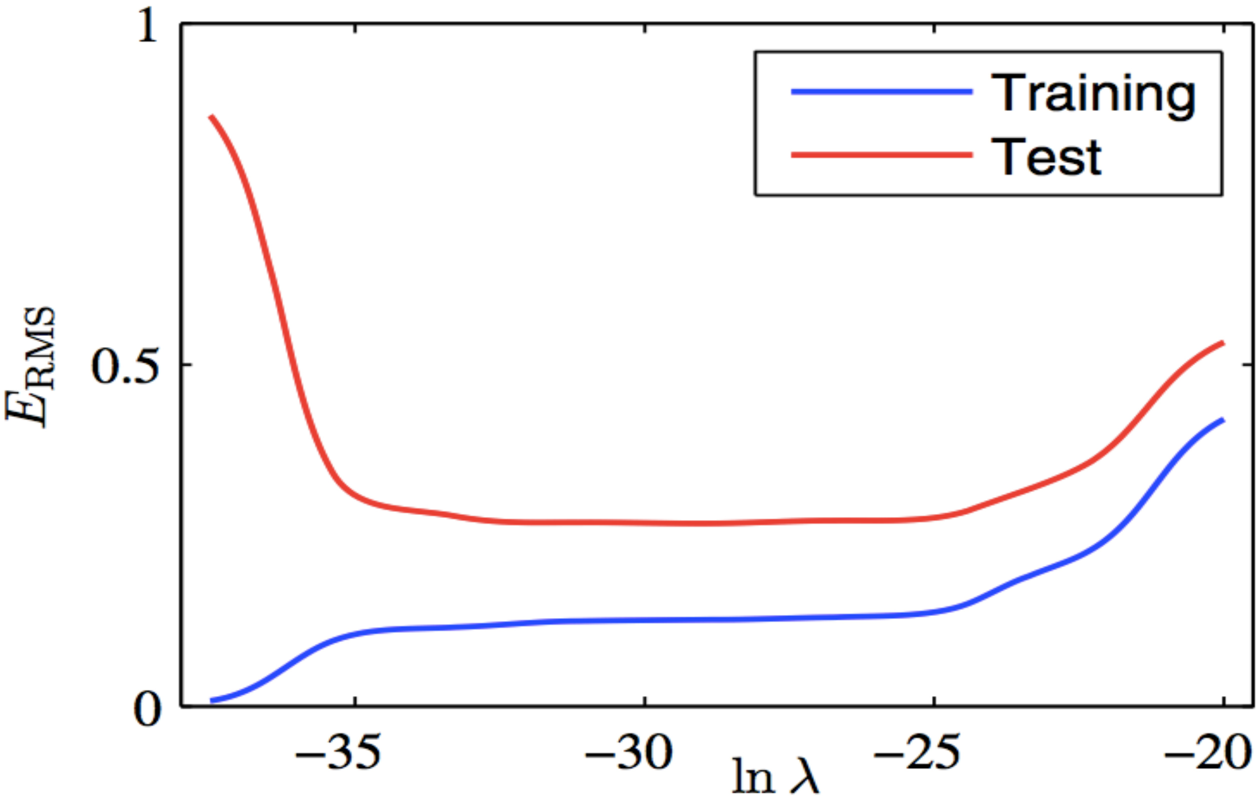
Lasso Vs Ridge



Lasso Vs Ridge



Over-fitting

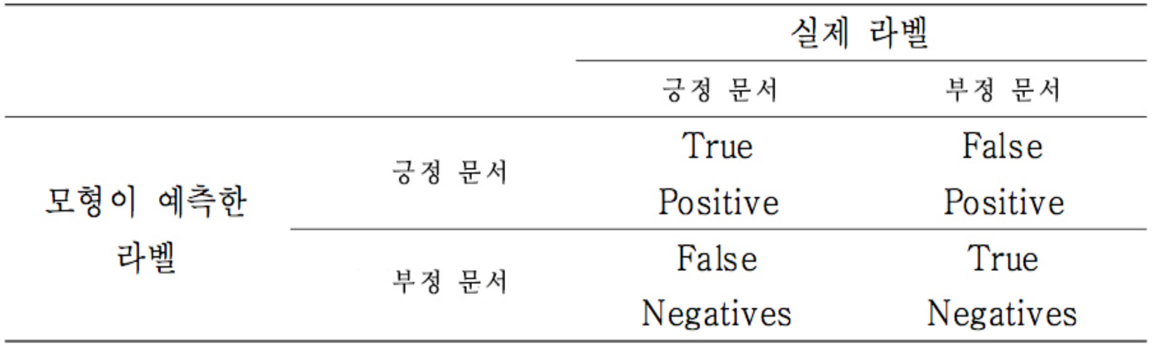


예측력

MSE(Mean of Square Error)

$$MSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

정확도(Accracy)



		실제 라벨	
		긍정 문서	부정 문서
모형이 예측한 라벨	긍정 문서	True Positive	False Positive
	부정 문서	False Negatives	True Negatives

정확도 = (TP + TN) / (TP + FP + TN + FN)

감정분석

Data

아마존 모바일 폰 리뷰 중에서 2,000개만

긍정평 1000개, 부정평 1000개

5. 예제 데이터 불러오기

```r
mobile <- read.csv('mobile2014.csv', stringsAsFactors = F)
```

```r
dim(mobile)
```

```
## [1] 2000    7
```

```r
names(mobile)
```

```
## [1] "X"           "Title"       "Author"      "ReviewID"    "Texts"       "YMD"        
## [7] "Sentiment"
```

```r
table(mobile$Sentiment)
```

```
## 
##    0    1 
## 1000 1000
```

6. DocumentTermMatrix 만들기

```r
library(tm)
```

```r
corpus <- Corpus(VectorSource(mobile$Texts))

##  제거할 단어 목록 확인
stopwords()
```

```
##   [1] "i"          "me"         "my"         "myself"     "we"
##   [6] "our"        "ours"       "ourselves"  "you"        "your"
##  [11] "yours"      "yourself"   "yourselves" "he"         "him"
##  [16] "his"        "himself"    "she"        "her"        "hers"
##  [21] "herself"    "it"         "its"        "itself"     "they"
##  [26] "them"       "their"      "theirs"     "themselves" "what"
##  [31] "which"      "who"        "whom"       "this"       "that"
##  [36] "these"      "those"      "am"         "is"         "are"
##  [41] "was"        "were"       "be"         "been"       "being"
##  [46] "have"       "has"        "had"        "having"     "do"
##  [51] "does"       "did"        "doing"      "would"      "should"
```

6. DocumentTermMatrix 만들기

```
dtm <- DocumentTermMatrix(corpus,
                          control = list(tolower = T,
                                         removePunctuation = T,
                                         removeNumbers = T,
                                         stopwords = stopwords("SMART"),
                                         weighting = weightTfIdf))
```

```
## Warning in weighting(x): empty document(s): 1948
```

```
dtm
```

```
## <<DocumentTermMatrix (documents: 2000, terms: 8446)>>
## Non-/sparse entries: 46461/16845539
## Sparsity           : 100%
## Maximal term length: 132
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

7. 회귀분석으로 감정 사전 만들기

```r
library(glmnet)
```

```r
X <- as.matrix(dtm)
Y <- mobile$Sentiment
```

```r
res.lm <- glmnet(X, Y, family = "binomial", lambda = 0)
```

7. 회귀분석으로 감정 사전 만들기

```
coef.lm <- coef(res.lm)[,1]
pos.lm <- coef.lm[coef.lm > 0]
neg.lm <- coef.lm[coef.lm < 0]
pos.lm <- sort(pos.lm, decreasing = T)
neg.lm <- sort(neg.lm, decreasing = F)
```

7. 회귀분석으로 감정 사전 만들기

```
pos.lm[1:20]
```

```
##   aboutbattery     absorption     accustomed    accommodate         allthe 
##      1085.0779       971.8079       742.7851       736.4887       680.7802 
##     anywhereno  allconclusion  afterthought   beautifully            agt 
##       673.7496       658.7255       501.8829       491.0924       438.5957 
##  accidentially           blog     alongside      anymorei      acclaimed 
##       370.4009       359.6093       346.5438       279.9606       254.9961 
##           amps       advocate brightnesscall accelerometer       accurate 
##       253.5091       229.8871       202.4650       197.5923       197.3163
```

7. 회귀분석으로 감정 사전 만들기

```
neg.lm[1:20]
```

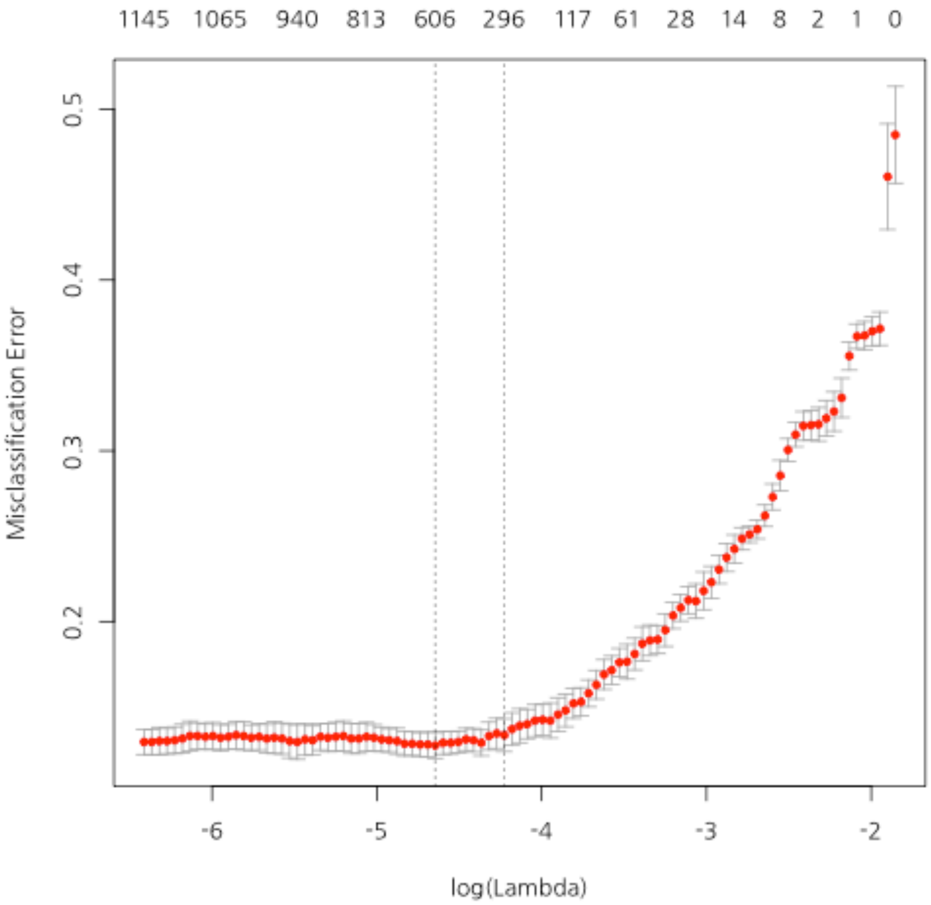
```
##     adverse  accomplish     apology      boasts     applaud     artists 
##   -337.1834   -330.4808   -298.6760   -256.1969   -253.5539   -217.8384 
##   addresses       ainol     amazoni    accesses     arrange      blocks 
##   -216.7636   -213.2872   -202.4802   -191.3860   -181.3013   -179.3567 
##   averaging comparisons   adddelete     annoys        aarp    admitted 
##   -177.2297   -176.4454   -173.5037   -168.7699   -167.4885   -163.6020 
##         aka    amazonit 
##   -156.0933   -154.2843
```

10. 라쏘 회귀분석으로 감정 사전 만들기

```r
set.seed(12345)
res.lasso <- cv.glmnet(X, Y, family = "binomial", alpha = 1,
                       nfolds = 4, type.measure = "class")
```

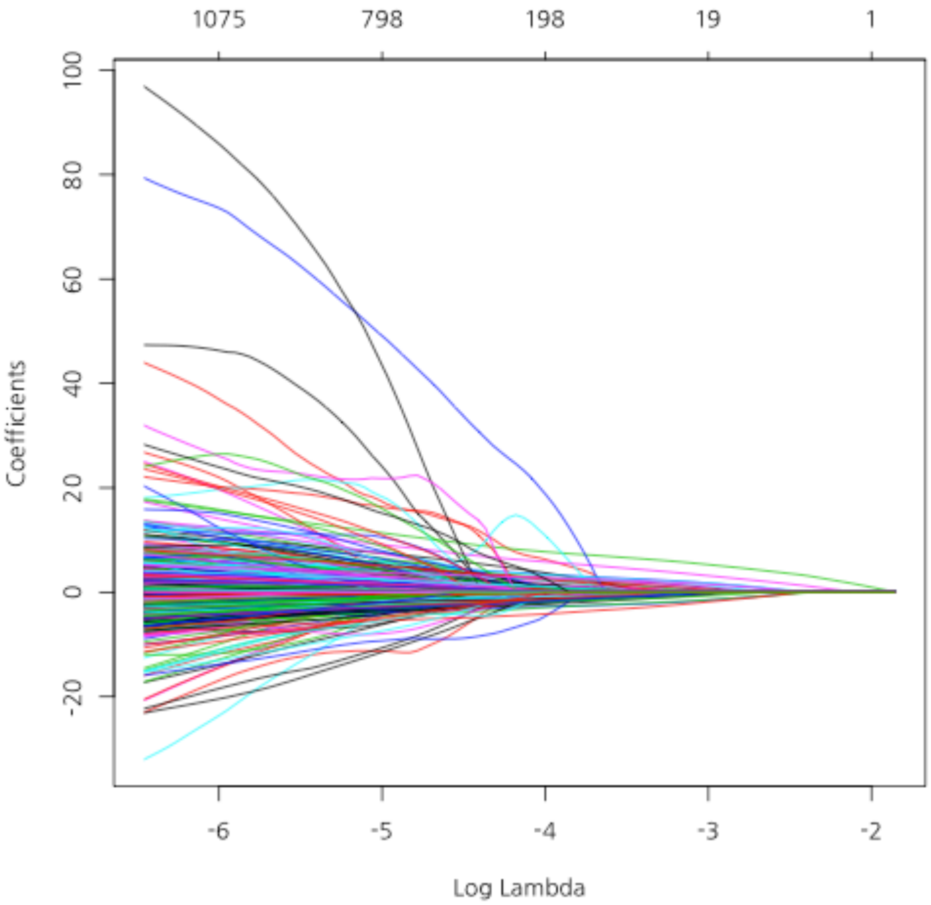
10. 라쏘 회귀분석으로 감정 사전 만들기

```
plot(res.lasso)
```



10. 라쏘 회귀분석으로 감정 사전 만들기

```
plot(res.lasso$glmnet.fit, xvar = "lambda")
```



10. 라쏘 회귀분석으로 감정 사전 만들기

```r
options(scipen = 100)
coef.lasso <- coef(res.lasso, s = "lambda.min")[,1]
pos.lasso <- coef.lasso[coef.lasso > 0]
neg.lasso <- coef.lasso[coef.lasso < 0]
pos.lasso <- sort(pos.lasso, decreasing = T)
neg.lasso <- sort(neg.lasso, decreasing = F)
```

10. 라쏘 회귀분석으로 감정 사전 만들기

```
pos.lasso[1:20]
```

```
##           news      sharpness        section implementation         amoled 
##      38.577474      19.924116      17.276675      14.786139      14.263770 
##          youve      autofocus          great       whatsapp        monster 
##      11.408438      10.652642       9.939520       9.620485       8.232943 
##           love responsiveness        swiping         kitkat           eyes 
##       7.835211       6.849303       6.713517       6.351620       5.576048 
##            pro           pair           fits        perfect           easy 
##       5.325080       5.165146       4.919954       4.671299       4.651886 
```

10. 라쏘 회귀분석으로 감정 사전 만들기

```
neg.lasso[1:20]
```

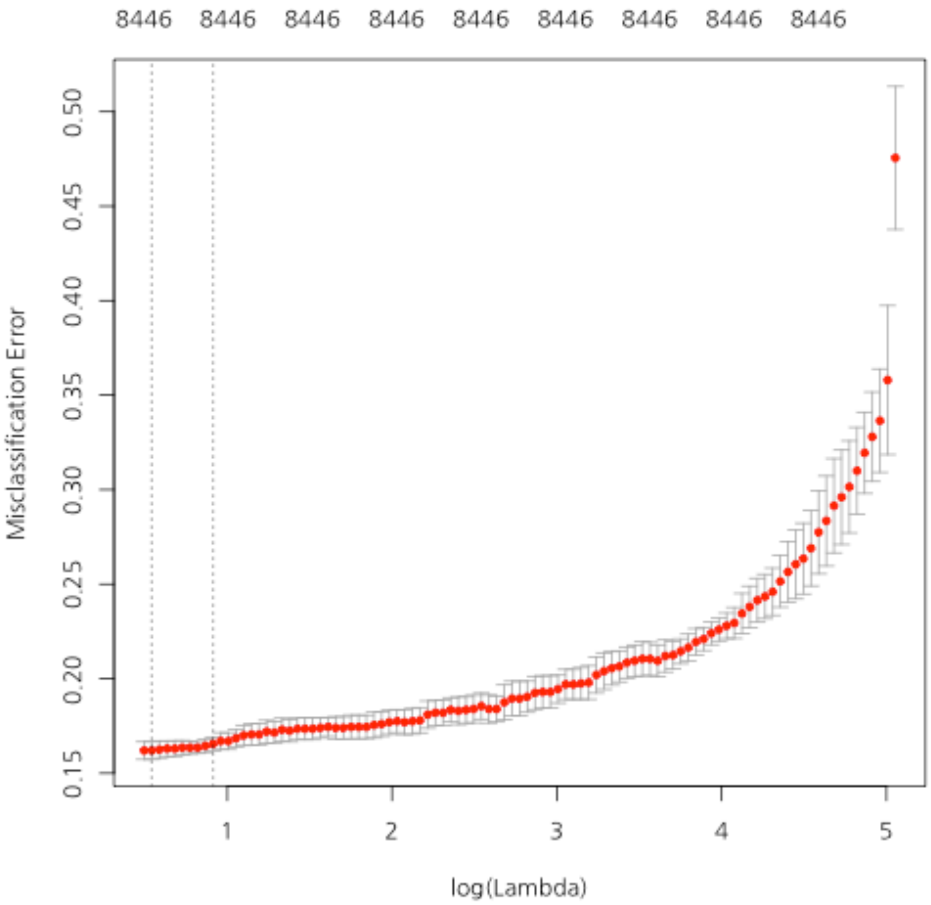
```
##    addresses       pushed    promising consistently   repeatedly
##    -9.274913    -8.947030    -7.709165    -6.557389    -5.723836
##       return     versions        money        wakes        utter
##    -5.028760    -4.801251    -4.104239    -4.060679    -3.854104
##        elses         back          zip    contacted     swindled
##    -3.561807    -3.413399    -3.284574    -3.154823    -3.122682
##        sucks         july     horrible         slow     received
##    -3.117836    -3.014597    -2.965226    -2.871529    -2.654189
```

11. 릿지 회귀분석으로 감정 사전 만들기

```
set.seed(12345)
res.ridge <- cv.glmnet(X, Y, family = "binomial", alpha = 0,
                       nfolds = 4, type.measure = "class")
```

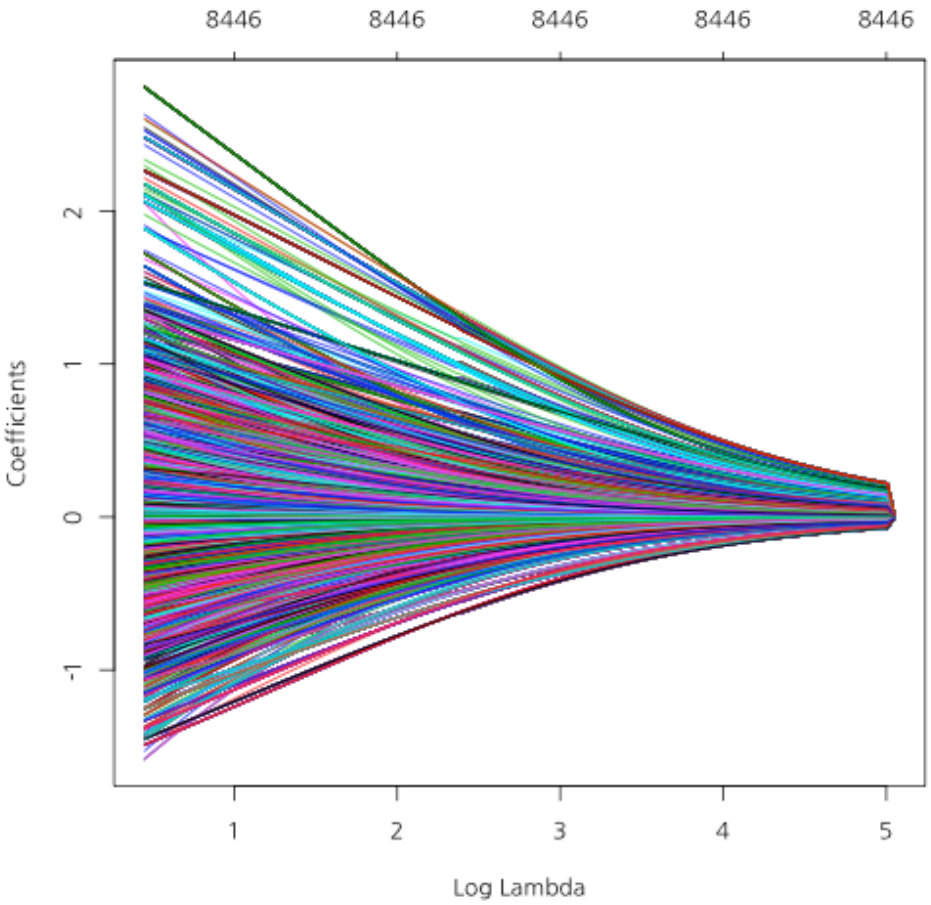
11. 릿지 회귀분석으로 감정 사전 만들기

```
plot(res.ridge)
```



11. 릿지 회귀분석으로 감정 사전 만들기

```
plot(res.ridge$glmnet.fit, xvar = "lambda")
```



11. 릿지 회귀분석으로 감정 사전 만들기

```r
coef.ridge <- coef(res.ridge, s = "lambda.min")[,1]
pos.ridge <- coef.ridge[coef.ridge > 0]
neg.ridge <- coef.ridge[coef.ridge < 0]
pos.ridge <- sort(pos.ridge, decreasing = T)
neg.ridge <- sort(neg.ridge, decreasing = F)
```

11. 릿지 회귀분석으로 감정 사전 만들기

```
pos.ridge[1:20]
```

```
##   anywhereno brightnessto    yourselfi        wifig      whistle 
##     2.739847     2.738926     2.738648     2.738635     2.738580 
##    waistband commentsapps     vertical   thatbottom       copied 
##     2.738508     2.738440     2.738419     2.738313     2.738093 
##    speedcons    spacealso     smallest       cutter     sleeping 
##     2.738082     2.737937     2.737801     2.737719     2.737640 
##   screenvery    detailing       puffin       doable    phonetips 
##     2.737409     2.737377     2.737238     2.737078     2.737075
```

11. 릿지 회귀분석으로 감정 사전 만들기

```
neg.ridge[1:20]
```

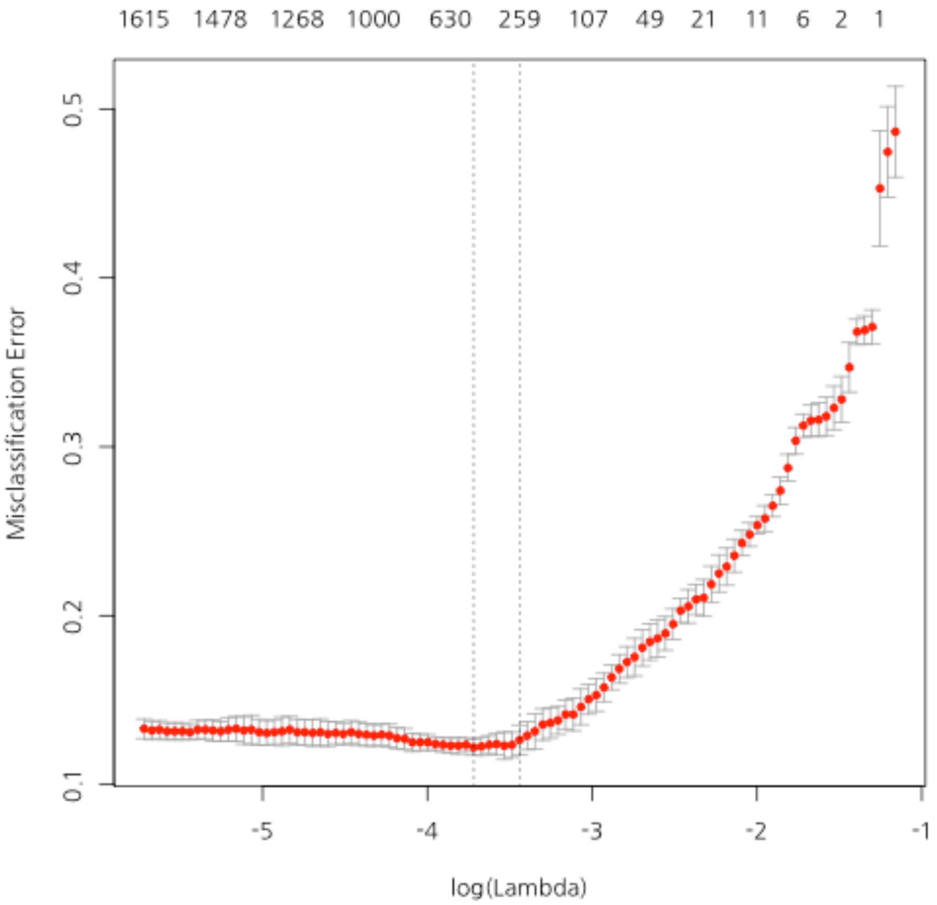
```
##      slowness      limiting        groove        engage discontinuing 
##     -1.500915     -1.500909     -1.500890     -1.500862     -1.500859 
##       apology        carpet   disgruntled   explanation   hypothesize 
##     -1.449195     -1.448927     -1.448633     -1.448471     -1.448232 
##      whomever       voltage           toi       jumping         scuff 
##     -1.448227     -1.448198     -1.448162     -1.448128     -1.448125 
##   returnclaim       loosely   refurbishes       reflect        paused 
##     -1.448084     -1.448064     -1.448048     -1.448016     -1.447995 
```

12. 엘라스틱넷으로 감정 사전 만들기

```r
set.seed(12345)
res.elastic <- cv.glmnet(X, Y, family = "binomial", alpha = .5,
                         nfolds = 4, type.measure="class")
```

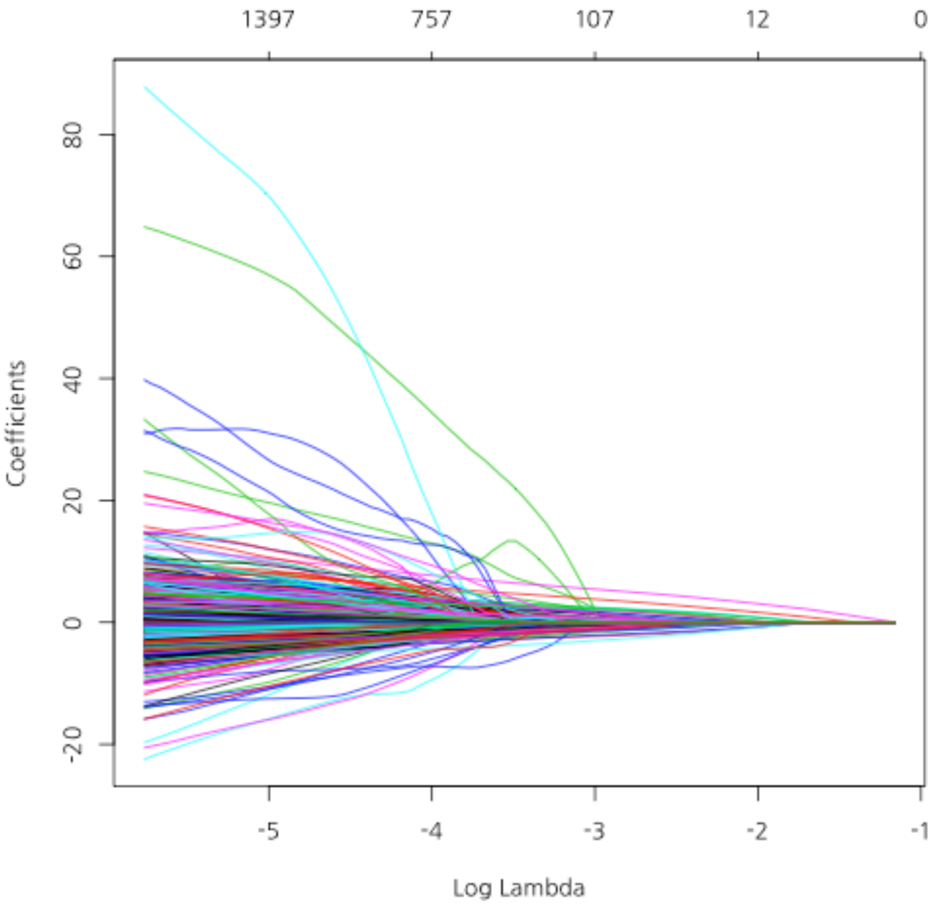
12. 엘라스틱넷으로 감정 사전 만들기

```
plot(res.elastic)
```



12. 엘라스틱넷으로 감정 사전 만들기

```
plot(res.elastic$glmnet.fit, xvar = "lambda")
```



12. 엘라스틱넷으로 감정 사전 만들기

```
coef.elastic <- coef(res.elastic, s = "lambda.min")[,1]
pos.elastic <- coef.elastic[coef.elastic > 0]
neg.elastic <- coef.elastic[coef.elastic < 0]
pos.elastic <- sort(pos.elastic, decreasing = T)
neg.elastic <- sort(neg.elastic, decreasing = F)
```

12. 엘라스틱넷으로 감정 사전 만들기

```
pos.elastic[1:20]
```

```
##           news         amoled      sharpness      intrusive implementation 
##      27.487451      10.570031      10.333007       9.966970       8.992294 
##          great          youve           love         kitkat           eyes 
##       6.975481       6.270957       5.106715       4.591255       4.382721 
##       whatsapp        perfect           easy           fits         remote 
##       4.193411       3.499910       3.440100       3.432532       3.380695 
##            pro      impressed        windows           fast          loves 
##       3.188608       3.149368       3.116624       3.035995       2.988360 
```

12. 엘라스틱넷으로 감정 사전 만들기

```
neg.elastic[1:20]
```

```
##       pushed    addresses        wakes       return    promising
##    -7.409465    -4.587790    -4.123647    -4.070836    -3.998956
##   repeatedly consistently        money         back    contacted
##    -3.940290    -3.238517    -3.099030    -2.986000    -2.903110
##        sucks        utter     versions     horrible         told
##    -2.556260    -2.391758    -2.374289    -2.330850    -2.312851
##       refund         slow      wouldnt     terrible     received
##    -2.273095    -2.250336    -2.126634    -2.101997    -2.027309
```

14. 감정사전을 이용한 감정분석

```
library(tm.plugin.sentiment)
```

```
senti.lm <- polarity(dtm, names(pos.lm), names(neg.lm))
senti.lasso <- polarity(dtm, names(pos.lasso), names(neg.lasso))
senti.ridge <- polarity(dtm, names(pos.ridge), names(neg.ridge))
senti.elastic <- polarity(dtm, names(pos.elastic), names(neg.elastic))
```

14. 감정사전을 이용한 감정분석

```
senti.lm <- polarity(dtm, names(pos.lm), names(neg.lm))
senti.lasso <- polarity(dtm, names(pos.lasso), names(neg.lasso))
senti.ridge <- polarity(dtm, names(pos.ridge), names(neg.ridge))
senti.elastic <- polarity(dtm, names(pos.elastic), names(neg.elastic))
```

15. 감정분석이 얼마나 정확한가 확인하기

```
senti.lm.b <- ifelse(senti.lm > 0, 1, 0)
senti.lasso.b <- ifelse(senti.lasso > 0, 1, 0)
senti.ridge.b <- ifelse(senti.ridge > 0, 1, 0)
senti.elastic.b <- ifelse(senti.elastic > 0, 1, 0)
```

15. 감정분석이 얼마나 정확한가 확인하기

```
library(caret)
```

15. 감정분석이 얼마나 정확한가 확인하기

```
confusionMatrix(senti.lm.b, mobile$Sentiment)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 962 535
##          1  38 464
## 
##                Accuracy : 0.7134
##                  95% CI : (0.693, 0.7331)
##     No Information Rate : 0.5003
##     P-Value [Acc > NIR] : < 0.0000000000000002
## 
##                   Kappa : 0.4266
##  Mcnemar's Test P-Value : < 0.0000000000000002
## 
##             Sensitivity : 0.9620
##             Specificity : 0.4645
```

15. 감정분석이 얼마나 정확한가 확인하기

```
confusionMatrix(senti.lasso.b, mobile$Sentiment)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 978  47
##          1  18 952
##
##                Accuracy : 0.9674
##                  95% CI : (0.9587, 0.9748)
##     No Information Rate : 0.5008
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.9348
##  Mcnemar's Test P-Value : 0.0005147
##
##             Sensitivity : 0.9819
##             Specificity : 0.9530
```

15. 감정분석이 얼마나 정확한가 확인하기

```
confusionMatrix(senti.ridge.b, mobile$Sentiment)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 984  20
##          1  16 979
## 
##                Accuracy : 0.982
##                  95% CI : (0.9752, 0.9874)
##     No Information Rate : 0.5003
##     P-Value [Acc > NIR] : <0.0000000000000002
## 
##                   Kappa : 0.964
##  Mcnemar's Test P-Value : 0.6171
## 
##             Sensitivity : 0.9840
##             Specificity : 0.9800
```

15. 감정분석이 얼마나 정확한가 확인하기

```
confusionMatrix(senti.elastic.b, mobile$Sentiment)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 977  61
##          1  18 934
## 
##                Accuracy : 0.9603
##                  95% CI : (0.9508, 0.9684)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 0.0000000000000022
## 
##                   Kappa : 0.9206
##  Mcnemar's Test P-Value : 0.000002297
## 
##             Sensitivity : 0.9819
##             Specificity : 0.9387
```

15. 감정분석이 얼마나 정확한가 확인하기

```
mobile.test <- read.csv("mobile2014_test.csv", stringsAsFactors = F)
```

```
dim(mobile.test)
```

```
## [1] 1000    7
```

```
names(mobile.test)
```

```
## [1] "X"         "Title"     "Author"    "ReviewID"  "Texts"     "YMD"
## [7] "Sentiment"
```

```
table(mobile.test$Sentiment)
```

```
##
##   0   1
## 500 500
```

15. 감정분석이 얼마나 정확한가 확인하기

```
corpus <- Corpus(VectorSource(mobile.test$Texts))
dtm.test <- DocumentTermMatrix(corpus,
                          control = list(tolower = T,
                                         removePunctuation = T,
                                         removeNumbers = T,
                                         stopwords = stopwords("SMART"),
                                         weighting = weightTfIdf,
                                         dictionary = Terms(dtm)))
```

```
## Warning in weighting(x): empty document(s): 883
```

```
## Warning in weighting(x): unreferenced term(s):  downloading aah aaps
## abombada aboutbattery aboutprice abroad abruptly absolutamente absulotly
## accelerometer accent accepting accessed accesses accessibility
## accessoriesi accessoriesin accidentially acclaimed acclimating accomplish
## accomplishing accordinglyupgrading accounts accurate accurateif
## accurateremote accustomed ace achieve aclarar acquainted acted action
## activating activitate activity actuall actualmente acurate adapt adapted
```

15. 감정분석이 얼마나 정확한가 확인하기

```
senti.lm.test <- polarity(dtm.test, names(pos.lm), names(neg.lm))
senti.lasso.test <- polarity(dtm.test, names(pos.lasso), names(neg.lasso))
senti.ridge.test <- polarity(dtm.test, names(pos.ridge), names(neg.ridge))
senti.elastic.test <- polarity(dtm.test, names(pos.elastic), names(neg.elastic))
```

15. 감정분석이 얼마나 정확한가 확인하기

```r
senti.lm.b.test <- ifelse(senti.lm.test > 0, 1, 0)
senti.lasso.b.test <- ifelse(senti.lasso.test > 0, 1, 0)
senti.ridge.b.test <- ifelse(senti.ridge.test > 0, 1, 0)
senti.elastic.b.test <- ifelse(senti.elastic.test > 0, 1, 0)
```

15. 감정분석이 얼마나 정확한가 확인하기

```
confusionMatrix(senti.lm.b.test, mobile.test$Sentiment)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 445 301
##          1  55 198
##                                              
##                Accuracy : 0.6436             
##                  95% CI : (0.6131, 0.6734)   
##     No Information Rate : 0.5005             
##     P-Value [Acc > NIR] : < 0.0000000000000002
##                                              
##                   Kappa : 0.2869             
##  Mcnemar's Test P-Value : < 0.0000000000000002
##                                              
##             Sensitivity : 0.8900             
##             Specificity : 0.3968             
```

15. 감정분석이 얼마나 정확한가 확인하기

```
confusionMatrix(senti.lasso.b.test, mobile.test$Sentiment)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 447  86
##          1  45 411
## 
##                Accuracy : 0.8675
##                  95% CI : (0.8448, 0.8881)
##     No Information Rate : 0.5025
##     P-Value [Acc > NIR] : < 0.00000000000000022
## 
##                   Kappa : 0.7352
##  Mcnemar's Test P-Value : 0.0004744
## 
##             Sensitivity : 0.9085
##             Specificity : 0.8270
```

15. 감정분석이 얼마나 정확한가 확인하기

```
confusionMatrix(senti.ridge.b.test, mobile.test$Sentiment)
```

```
## Confusion Matrix and Statistics
## 
##            Reference
## Prediction   0   1
##          0 444  85
##          1  56 414
## 
##                Accuracy : 0.8589
##                  95% CI : (0.8357, 0.8799)
##     No Information Rate : 0.5005
##     P-Value [Acc > NIR] : < 0.0000000000000002
## 
##                   Kappa : 0.7177
##  Mcnemar's Test P-Value : 0.01837
## 
##             Sensitivity : 0.8880
##             Specificity : 0.8297
```

15. 감정분석이 얼마나 정확한가 확인하기

```
confusionMatrix(senti.elastic.b.test, mobile.test$Sentiment)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 454  82
##          1  39 416
## 
##                Accuracy : 0.8779
##                  95% CI : (0.8559, 0.8976)
##     No Information Rate : 0.5025
##     P-Value [Acc > NIR] : < 0.00000000000000022
## 
##                   Kappa : 0.7559
##  Mcnemar's Test P-Value : 0.0001344
## 
##             Sensitivity : 0.9209
##             Specificity : 0.8353
```

18. 감정사전의 회귀계수를 이용한 감정분석

```
X.test <- as.matrix(dtm.test)
senti.lm.test.coef <- predict(res.lm , newx = X.test)
senti.lasso.test.coef <- predict(res.lasso, newx = X.test, s = "lambda.min")
senti.ridge.test.coef <- predict(res.ridge, newx = X.test, s = "lambda.min")
senti.elastic.test.coef <- predict(res.elastic, newx = X.test, s = "lambda.min")
```

18. 감정사전의 회귀계수를 이용한 감정분석

```r
senti.lm.b.test.coef <- ifelse(senti.lm.test.coef > 0, 1, 0)
senti.lasso.b.test.coef <- ifelse(senti.lasso.test.coef > 0, 1, 0)
senti.ridge.b.test.coef <- ifelse(senti.ridge.test.coef > 0, 1, 0)
senti.elastic.b.test.coef <- ifelse(senti.elastic.test.coef > 0, 1, 0)
```

18. 감정사전의 회귀계수를 이용한 감정분석

```
confusionMatrix(senti.lm.b.test.coef, mobile.test$Sentiment)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 363 111
##          1 137 389
##
##                Accuracy : 0.752
##                  95% CI : (0.724, 0.7785)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.504
##  Mcnemar's Test P-Value : 0.1124
##
##             Sensitivity : 0.7260
##             Specificity : 0.7780
```

18. 감정사전의 회귀계수를 이용한 감정분석

```
confusionMatrix(senti.lasso.b.test.coef, mobile.test$Sentiment)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 459  72
##          1  41 428
## 
##                Accuracy : 0.887
##                  95% CI : (0.8657, 0.906)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 0.0000000000000002
## 
##                   Kappa : 0.774
##  Mcnemar's Test P-Value : 0.00477
## 
##             Sensitivity : 0.9180
##             Specificity : 0.8560
```

18. 감정사전의 회귀계수를 이용한 감정분석

```
confusionMatrix(senti.ridge.b.test.coef, mobile.test$Sentiment)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0 417  67
##          1  83 433
##
##                Accuracy : 0.85
##                  95% CI : (0.8263, 0.8716)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.7
##  Mcnemar's Test P-Value : 0.2207
##
##             Sensitivity : 0.8340
##             Specificity : 0.8660
```

18. 감정사전의 회귀계수를 이용한 감정분석

```
confusionMatrix(senti.elastic.b.test.coef, mobile.test$Sentiment)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 464  68
##          1  36 432
##
##                Accuracy : 0.896
##                  95% CI : (0.8754, 0.9142)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 0.0000000000000022
##
##                   Kappa : 0.792
##  Mcnemar's Test P-Value : 0.002367
##
##             Sensitivity : 0.9280
##             Specificity : 0.8640
```

종합

```
confusionMatrix(senti.lm.b, mobile$Sentiment)$overall[1]
```

```
##  Accuracy
## 0.7133567
```

```
confusionMatrix(senti.lm.b.test, mobile.test$Sentiment)$overall[1]
```

```
##  Accuracy
## 0.6436436
```

```
confusionMatrix(senti.lm.b.test.coef, mobile.test$Sentiment)$overall[1]
```

```
## Accuracy
##    0.752
```

종합

```
confusionMatrix(senti.lasso.b, mobile$Sentiment)$overall[1]
```

```
##  Accuracy
## 0.9674185
```

```
confusionMatrix(senti.lasso.b.test, mobile.test$Sentiment)$overall[1]
```

```
## Accuracy
## 0.867543
```

```
confusionMatrix(senti.lasso.b.test.coef, mobile.test$Sentiment)$overall[1]
```

```
## Accuracy
##    0.887
```

종합

```
confusionMatrix(senti.ridge.b, mobile$Sentiment)$overall[1]
```

```
## Accuracy
## 0.981991
```

```
confusionMatrix(senti.ridge.b.test, mobile.test$Sentiment)$overall[1]
```

```
##  Accuracy
## 0.8588589
```

```
confusionMatrix(senti.ridge.b.test.coef, mobile.test$Sentiment)$overall[1]
```

```
## Accuracy
##     0.85
```

종합

```
confusionMatrix(senti.elastic.b, mobile$Sentiment)$overall[1]
```

```
##  Accuracy
## 0.9603015
```

```
confusionMatrix(senti.elastic.b.test, mobile.test$Sentiment)$overall[1]
```

```
##  Accuracy
## 0.8779011
```

```
confusionMatrix(senti.elastic.b.test.coef, mobile.test$Sentiment)$overall[1]
```

```
## Accuracy
##    0.896
```