> getwd()

[1] "C:/Users/Alok Satpathy/Documents"

> setwd("C:/Users/Alok Satpathy/Desktop/Fall 2016/EDA/Final Project")

> Q2csv=read.csv("Question2Dataset.csv")

> attach(Q2csv)

#Selecting columns that would be used for preparing model

> Q2df=data.frame(a\_score, a\_body\_length, a\_body\_has\_code, a\_DaysOld, a\_has\_edited, a\_num\_comment, a\_owner\_reputation, a\_owner\_profile\_summary, a\_owner\_views, a\_owner\_upvotes, a\_owner\_downvotes, q\_score, q\_num\_views, q\_body\_length, q\_body\_has\_code, q\_DaysOld, q\_has\_edited, q\_title\_length, q\_num\_tags, q\_num\_answers, q\_num\_comment, q\_owner\_reputation, q\_owner\_profile\_summary, q\_owner\_views, q\_owner\_upvotes, q\_owner\_downvotes, accepted\_answer\_flag, a\_votes\_up, a\_votes\_down, q\_votes\_up, q\_votes\_down)

> str(Q2df)

'data.frame': 50000 obs. of 31 variables:

$ a\_score : int 1 1 0 0 2 0 9 2 1 0 ...

$ a\_body\_length : int 357 430 267 1323 575 252 225 367 1514 786 ...

$ a\_body\_has\_code : int 1 1 0 1 1 1 1 1 1 1 ...

$ a\_DaysOld : int 98 62 40 39 69 88 88 88 100 35 ...

$ a\_has\_edited : int 1 1 0 1 1 0 1 1 1 0 ...

$ a\_num\_comment : int 0 0 0 0 0 1 5 3 0 0 ...

$ a\_owner\_reputation : int 114 11 705 403 635 22 382 112 1715 46 ...

$ a\_owner\_profile\_summary: int 3 2 2 2 2 0 1 3 3 1 ...

$ a\_owner\_views : int 7 0 65 113 93 10 16 10 117 11 ...

$ a\_owner\_upvotes : int 88 0 519 947 92 6 61 46 90 2 ...

$ a\_owner\_downvotes : int 0 0 1 0 7 0 1 0 5 0 ...

$ q\_score : int 5 31 34 112 1089 1 1 1 1089 1089 ...

$ q\_num\_views : int 8090 12901 15772 94568 1673124 124 124 124 1673124 1673124 ...

$ q\_body\_length : int 534 635 613 649 737 861 861 861 737 737 ...

$ q\_body\_has\_code : int 1 0 0 1 1 1 1 1 1 1 ...

$ q\_DaysOld : int 1033 653 627 2935 1787 88 88 88 1787 1787 ...

$ q\_has\_edited : int 1 1 1 1 1 1 1 1 1 1 ...

$ q\_title\_length : int 48 67 47 44 39 31 31 31 39 39 ...

$ q\_num\_tags : int 2 2 2 4 3 2 2 2 3 3 ...

$ q\_num\_answers : int 11 5 6 8 26 5 5 5 26 26 ...

$ q\_num\_comment : int 6 0 3 3 3 0 0 0 3 3 ...

$ q\_owner\_reputation : int 284 2220 6544 96603 8035 61 61 61 8035 8035 ...

$ q\_owner\_profile\_summary: int 0 3 2 2 1 0 0 0 1 1 ...

$ q\_owner\_views : int 61 286 586 6697 594 19 19 19 594 594 ...

$ q\_owner\_upvotes : int 7 31 178 3641 519 18 18 18 519 519 ...

$ q\_owner\_downvotes : int 0 4 13 100 7 0 0 0 7 7 ...

$ accepted\_answer\_flag : int 0 0 0 0 0 0 1 0 0 0 ...

$ a\_votes\_up : int 1 1 NA NA 2 1 9 2 1 NA ...

$ a\_votes\_down : int NA NA NA NA NA 1 NA NA NA NA ...

$ q\_votes\_up : int 5 31 34 112 1090 2 2 2 1090 1090 ...

$ q\_votes\_down : int NA NA NA NA 1 1 1 1 1 1 ...

#Replacing NA with 0

> Q2df[is.na(Q2df)]=0

> str(Q2df)

'data.frame': 50000 obs. of 31 variables:

$ a\_score : int 1 1 0 0 2 0 9 2 1 0 ...

$ a\_body\_length : int 357 430 267 1323 575 252 225 367 1514 786 ...

$ a\_body\_has\_code : int 1 1 0 1 1 1 1 1 1 1 ...

$ a\_DaysOld : int 98 62 40 39 69 88 88 88 100 35 ...

$ a\_has\_edited : int 1 1 0 1 1 0 1 1 1 0 ...

$ a\_num\_comment : int 0 0 0 0 0 1 5 3 0 0 ...

$ a\_owner\_reputation : int 114 11 705 403 635 22 382 112 1715 46 ...

$ a\_owner\_profile\_summary: int 3 2 2 2 2 0 1 3 3 1 ...

$ a\_owner\_views : int 7 0 65 113 93 10 16 10 117 11 ...

$ a\_owner\_upvotes : int 88 0 519 947 92 6 61 46 90 2 ...

$ a\_owner\_downvotes : int 0 0 1 0 7 0 1 0 5 0 ...

$ q\_score : int 5 31 34 112 1089 1 1 1 1089 1089 ...

$ q\_num\_views : int 8090 12901 15772 94568 1673124 124 124 124 1673124 1673124 ...

$ q\_body\_length : int 534 635 613 649 737 861 861 861 737 737 ...

$ q\_body\_has\_code : int 1 0 0 1 1 1 1 1 1 1 ...

$ q\_DaysOld : int 1033 653 627 2935 1787 88 88 88 1787 1787 ...

$ q\_has\_edited : int 1 1 1 1 1 1 1 1 1 1 ...

$ q\_title\_length : int 48 67 47 44 39 31 31 31 39 39 ...

$ q\_num\_tags : int 2 2 2 4 3 2 2 2 3 3 ...

$ q\_num\_answers : int 11 5 6 8 26 5 5 5 26 26 ...

$ q\_num\_comment : int 6 0 3 3 3 0 0 0 3 3 ...

$ q\_owner\_reputation : int 284 2220 6544 96603 8035 61 61 61 8035 8035 ...

$ q\_owner\_profile\_summary: int 0 3 2 2 1 0 0 0 1 1 ...

$ q\_owner\_views : int 61 286 586 6697 594 19 19 19 594 594 ...

$ q\_owner\_upvotes : int 7 31 178 3641 519 18 18 18 519 519 ...

$ q\_owner\_downvotes : int 0 4 13 100 7 0 0 0 7 7 ...

$ accepted\_answer\_flag : int 0 0 0 0 0 0 1 0 0 0 ...

$ a\_votes\_up : num 1 1 0 0 2 1 9 2 1 0 ...

$ a\_votes\_down : num 0 0 0 0 0 1 0 0 0 0 ...

$ q\_votes\_up : num 5 31 34 112 1090 2 2 2 1090 1090 ...

$ q\_votes\_down : num 0 0 0 0 1 1 1 1 1 1 ...

# Subset Selection Methods

# Best Subset Selection

> lapply(Q2df["accepted\_answer\_flag"], unique)

$accepted\_answer\_flag

[1] 0 1

> library(leaps)

> regfit.full=regsubsets(accepted\_answer\_flag~.,Q2df)

> summary(regfit.full)

Subset selection object

Call: regsubsets.formula(accepted\_answer\_flag ~ ., Q2df)

30 Variables (and intercept)

Forced in Forced out

a\_score FALSE FALSE

a\_body\_length FALSE FALSE

a\_body\_has\_code FALSE FALSE

a\_DaysOld FALSE FALSE

a\_has\_edited FALSE FALSE

a\_num\_comment FALSE FALSE

a\_owner\_reputation FALSE FALSE

a\_owner\_profile\_summary FALSE FALSE

a\_owner\_views FALSE FALSE

a\_owner\_upvotes FALSE FALSE

a\_owner\_downvotes FALSE FALSE

q\_score FALSE FALSE

q\_num\_views FALSE FALSE

q\_body\_length FALSE FALSE

q\_body\_has\_code FALSE FALSE

q\_DaysOld FALSE FALSE

q\_has\_edited FALSE FALSE

q\_title\_length FALSE FALSE

q\_num\_tags FALSE FALSE

q\_num\_answers FALSE FALSE

q\_num\_comment FALSE FALSE

q\_owner\_reputation FALSE FALSE

q\_owner\_profile\_summary FALSE FALSE

q\_owner\_views FALSE FALSE

q\_owner\_upvotes FALSE FALSE

q\_owner\_downvotes FALSE FALSE

a\_votes\_up FALSE FALSE

a\_votes\_down FALSE FALSE

q\_votes\_up FALSE FALSE

q\_votes\_down FALSE FALSE

1 subsets of each size up to 8

Selection Algorithm: exhaustive

a\_score a\_body\_length a\_body\_has\_code a\_DaysOld a\_has\_edited a\_num\_comment

1 ( 1 ) " " " " " " " " " " "\*"

2 ( 1 ) " " " " " " " " " " "\*"

3 ( 1 ) " " " " " " " " " " "\*"

4 ( 1 ) "\*" " " " " " " " " "\*"

5 ( 1 ) "\*" "\*" " " " " " " "\*"

6 ( 1 ) "\*" "\*" " " " " " " "\*"

7 ( 1 ) "\*" "\*" " " " " "\*" "\*"

8 ( 1 ) "\*" "\*" " " " " "\*" "\*"

a\_owner\_reputation a\_owner\_profile\_summary a\_owner\_views a\_owner\_upvotes

1 ( 1 ) " " " " " " " "

2 ( 1 ) " " " " " " " "

3 ( 1 ) "\*" " " " " " "

4 ( 1 ) "\*" " " " " " "

5 ( 1 ) "\*" " " " " " "

6 ( 1 ) "\*" " " " " " "

7 ( 1 ) "\*" " " " " " "

8 ( 1 ) "\*" "\*" " " " "

a\_owner\_downvotes q\_score q\_num\_views q\_body\_length q\_body\_has\_code q\_DaysOld

1 ( 1 ) " " " " " " " " " " " "

2 ( 1 ) " " " " " " " " " " "\*"

3 ( 1 ) " " " " " " " " " " "\*"

4 ( 1 ) " " " " " " " " " " "\*"

5 ( 1 ) " " " " " " " " " " "\*"

6 ( 1 ) "\*" " " " " " " " " "\*"

7 ( 1 ) "\*" " " " " " " " " "\*"

8 ( 1 ) "\*" " " " " " " " " "\*"

q\_has\_edited q\_title\_length q\_num\_tags q\_num\_answers q\_num\_comment

1 ( 1 ) " " " " " " " " " "

2 ( 1 ) " " " " " " " " " "

3 ( 1 ) " " " " " " " " " "

4 ( 1 ) " " " " " " " " " "

5 ( 1 ) " " " " " " " " " "

6 ( 1 ) " " " " " " " " " "

7 ( 1 ) " " " " " " " " " "

8 ( 1 ) " " " " " " " " " "

q\_owner\_reputation q\_owner\_profile\_summary q\_owner\_views q\_owner\_upvotes

1 ( 1 ) " " " " " " " "

2 ( 1 ) " " " " " " " "

3 ( 1 ) " " " " " " " "

4 ( 1 ) " " " " " " " "

5 ( 1 ) " " " " " " " "

6 ( 1 ) " " " " " " " "

7 ( 1 ) " " " " " " " "

8 ( 1 ) " " " " " " " "

q\_owner\_downvotes a\_votes\_up a\_votes\_down q\_votes\_up q\_votes\_down

1 ( 1 ) " " " " " " " " " "

2 ( 1 ) " " " " " " " " " "

3 ( 1 ) " " " " " " " " " "

4 ( 1 ) " " " " " " " " " "

5 ( 1 ) " " " " " " " " " "

6 ( 1 ) " " " " " " " " " "

7 ( 1 ) " " " " " " " " " "

8 ( 1 ) " " " " " " " " " "

> regfit.full=regsubsets(accepted\_answer\_flag~.,data=Q2df,nvmax=32)

> reg.summary=summary(regfit.full)

> names(reg.summary)

[1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"

> reg.summary$rsq

[1] 0.05875058 0.08252065 0.08957599 0.09405824 0.09759187 0.09964930 0.10121843

[8] 0.10282852 0.10408807 0.10466382 0.10522514 0.10571707 0.10599334 0.10627306

[15] 0.10651015 0.10663040 0.10672980 0.10683012 0.10689033 0.10694444 0.10696395

[22] 0.10699031 0.10700963 0.10702273 0.10702986 0.10703243 0.10703253 0.10703257

[29] 0.10703260 0.10703262

> par(mfrow=c(2,2))

> plot(reg.summary$rss,xlab="Number of Variables",ylab="RSS",type="l")

> plot(reg.summary$adjr2,xlab="Number of Variables",ylab="Adjusted

+ RSq",type="l")

> which.max(reg.summary$adjr2)

[1] 23

> points(11,reg.summary$adjr2[11], col="red",cex=2,pch=20)

> plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')

> which.min(reg.summary$cp)

[1] 20

> points(10,reg.summary$cp[10],col="red",cex=2,pch=20)

> which.min(reg.summary$bic)

[1] 15

> plot(reg.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')

> points(6,reg.summary$bic[6],col="red",cex=2,pch=20)

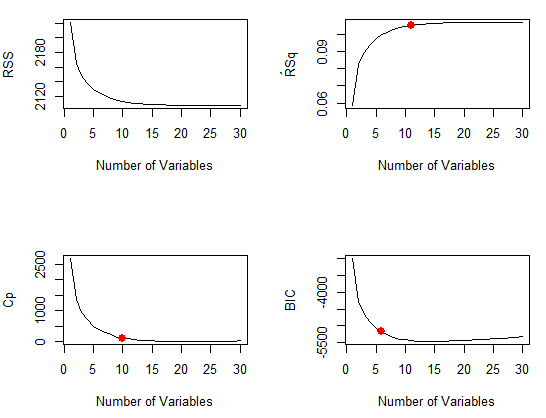
> coef(regfit.full,6)

(Intercept) a\_score a\_body\_length a\_num\_comment

5.427113e-02 2.259559e-03 1.155836e-05 2.171881e-02

a\_owner\_reputation a\_owner\_downvotes q\_DaysOld

3.589930e-07 1.043458e-05 -3.365066e-05



#Forward and Backward stepwise selection

> regfit.fwd=regsubsets(accepted\_answer\_flag~.,data=Q2df,nvmax=32,method="forward")

> summary(regfit.fwd)

Subset selection object

Call: regsubsets.formula(accepted\_answer\_flag ~ ., data = Q2df, nvmax = 32,

method = "forward")

30 Variables (and intercept)

Forced in Forced out

a\_score FALSE FALSE

a\_body\_length FALSE FALSE

a\_body\_has\_code FALSE FALSE

a\_DaysOld FALSE FALSE

a\_has\_edited FALSE FALSE

a\_num\_comment FALSE FALSE

a\_owner\_reputation FALSE FALSE

a\_owner\_profile\_summary FALSE FALSE

a\_owner\_views FALSE FALSE

a\_owner\_upvotes FALSE FALSE

a\_owner\_downvotes FALSE FALSE

q\_score FALSE FALSE

q\_num\_views FALSE FALSE

q\_body\_length FALSE FALSE

q\_body\_has\_code FALSE FALSE

q\_DaysOld FALSE FALSE

q\_has\_edited FALSE FALSE

q\_title\_length FALSE FALSE

q\_num\_tags FALSE FALSE

q\_num\_answers FALSE FALSE

q\_num\_comment FALSE FALSE

q\_owner\_reputation FALSE FALSE

q\_owner\_profile\_summary FALSE FALSE

q\_owner\_views FALSE FALSE

q\_owner\_upvotes FALSE FALSE

q\_owner\_downvotes FALSE FALSE

a\_votes\_up FALSE FALSE

a\_votes\_down FALSE FALSE

q\_votes\_up FALSE FALSE

q\_votes\_down FALSE FALSE

1 subsets of each size up to 30

Selection Algorithm: forward

a\_score a\_body\_length a\_body\_has\_code a\_DaysOld a\_has\_edited a\_num\_comment

1 ( 1 ) " " " " " " " " " " "\*"

2 ( 1 ) " " " " " " " " " " "\*"

3 ( 1 ) " " " " " " " " " " "\*"

4 ( 1 ) "\*" " " " " " " " " "\*"

5 ( 1 ) "\*" "\*" " " " " " " "\*"

6 ( 1 ) "\*" "\*" " " " " " " "\*"

7 ( 1 ) "\*" "\*" " " " " "\*" "\*"

8 ( 1 ) "\*" "\*" " " " " "\*" "\*"

9 ( 1 ) "\*" "\*" " " " " "\*" "\*"

10 ( 1 ) "\*" "\*" " " " " "\*" "\*"

11 ( 1 ) "\*" "\*" " " " " "\*" "\*"

12 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

13 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

14 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

15 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

16 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

17 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

18 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

19 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

20 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

21 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

22 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

23 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

24 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

25 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

26 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

27 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

28 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

a\_owner\_reputation a\_owner\_profile\_summary a\_owner\_views a\_owner\_upvotes

1 ( 1 ) " " " " " " " "

2 ( 1 ) " " " " " " " "

3 ( 1 ) "\*" " " " " " "

4 ( 1 ) "\*" " " " " " "

5 ( 1 ) "\*" " " " " " "

6 ( 1 ) "\*" " " " " " "

7 ( 1 ) "\*" " " " " " "

8 ( 1 ) "\*" "\*" " " " "

9 ( 1 ) "\*" "\*" " " " "

10 ( 1 ) "\*" "\*" " " " "

11 ( 1 ) "\*" "\*" "\*" " "

12 ( 1 ) "\*" "\*" "\*" " "

13 ( 1 ) "\*" "\*" "\*" " "

14 ( 1 ) "\*" "\*" "\*" " "

15 ( 1 ) "\*" "\*" "\*" " "

16 ( 1 ) "\*" "\*" "\*" " "

17 ( 1 ) "\*" "\*" "\*" " "

18 ( 1 ) "\*" "\*" "\*" " "

19 ( 1 ) "\*" "\*" "\*" " "

20 ( 1 ) "\*" "\*" "\*" " "

21 ( 1 ) "\*" "\*" "\*" " "

22 ( 1 ) "\*" "\*" "\*" " "

23 ( 1 ) "\*" "\*" "\*" " "

24 ( 1 ) "\*" "\*" "\*" "\*"

25 ( 1 ) "\*" "\*" "\*" "\*"

26 ( 1 ) "\*" "\*" "\*" "\*"

27 ( 1 ) "\*" "\*" "\*" "\*"

28 ( 1 ) "\*" "\*" "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*"

a\_owner\_downvotes q\_score q\_num\_views q\_body\_length q\_body\_has\_code q\_DaysOld

1 ( 1 ) " " " " " " " " " " " "

2 ( 1 ) " " " " " " " " " " "\*"

3 ( 1 ) " " " " " " " " " " "\*"

4 ( 1 ) " " " " " " " " " " "\*"

5 ( 1 ) " " " " " " " " " " "\*"

6 ( 1 ) "\*" " " " " " " " " "\*"

7 ( 1 ) "\*" " " " " " " " " "\*"

8 ( 1 ) "\*" " " " " " " " " "\*"

9 ( 1 ) "\*" " " " " " " " " "\*"

10 ( 1 ) "\*" " " " " " " " " "\*"

11 ( 1 ) "\*" " " " " " " " " "\*"

12 ( 1 ) "\*" " " " " " " " " "\*"

13 ( 1 ) "\*" " " " " " " " " "\*"

14 ( 1 ) "\*" " " " " " " " " "\*"

15 ( 1 ) "\*" " " " " " " " " "\*"

16 ( 1 ) "\*" " " " " " " "\*" "\*"

17 ( 1 ) "\*" " " " " " " "\*" "\*"

18 ( 1 ) "\*" " " "\*" " " "\*" "\*"

19 ( 1 ) "\*" " " "\*" " " "\*" "\*"

20 ( 1 ) "\*" " " "\*" "\*" "\*" "\*"

21 ( 1 ) "\*" " " "\*" "\*" "\*" "\*"

22 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

23 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

24 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

25 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

26 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

27 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

28 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

q\_has\_edited q\_title\_length q\_num\_tags q\_num\_answers q\_num\_comment

1 ( 1 ) " " " " " " " " " "

2 ( 1 ) " " " " " " " " " "

3 ( 1 ) " " " " " " " " " "

4 ( 1 ) " " " " " " " " " "

5 ( 1 ) " " " " " " " " " "

6 ( 1 ) " " " " " " " " " "

7 ( 1 ) " " " " " " " " " "

8 ( 1 ) " " " " " " " " " "

9 ( 1 ) " " " " " " " " "\*"

10 ( 1 ) " " " " " " " " "\*"

11 ( 1 ) " " " " " " " " "\*"

12 ( 1 ) " " " " " " " " "\*"

13 ( 1 ) " " "\*" " " " " "\*"

14 ( 1 ) " " "\*" " " " " "\*"

15 ( 1 ) " " "\*" "\*" " " "\*"

16 ( 1 ) " " "\*" "\*" " " "\*"

17 ( 1 ) " " "\*" "\*" "\*" "\*"

18 ( 1 ) " " "\*" "\*" "\*" "\*"

19 ( 1 ) "\*" "\*" "\*" "\*" "\*"

20 ( 1 ) "\*" "\*" "\*" "\*" "\*"

21 ( 1 ) "\*" "\*" "\*" "\*" "\*"

22 ( 1 ) "\*" "\*" "\*" "\*" "\*"

23 ( 1 ) "\*" "\*" "\*" "\*" "\*"

24 ( 1 ) "\*" "\*" "\*" "\*" "\*"

25 ( 1 ) "\*" "\*" "\*" "\*" "\*"

26 ( 1 ) "\*" "\*" "\*" "\*" "\*"

27 ( 1 ) "\*" "\*" "\*" "\*" "\*"

28 ( 1 ) "\*" "\*" "\*" "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*" "\*"

q\_owner\_reputation q\_owner\_profile\_summary q\_owner\_views q\_owner\_upvotes

1 ( 1 ) " " " " " " " "

2 ( 1 ) " " " " " " " "

3 ( 1 ) " " " " " " " "

4 ( 1 ) " " " " " " " "

5 ( 1 ) " " " " " " " "

6 ( 1 ) " " " " " " " "

7 ( 1 ) " " " " " " " "

8 ( 1 ) " " " " " " " "

9 ( 1 ) " " " " " " " "

10 ( 1 ) " " " " " " " "

11 ( 1 ) " " " " " " " "

12 ( 1 ) " " " " " " " "

13 ( 1 ) " " " " " " " "

14 ( 1 ) " " "\*" " " " "

15 ( 1 ) " " "\*" " " " "

16 ( 1 ) " " "\*" " " " "

17 ( 1 ) " " "\*" " " " "

18 ( 1 ) " " "\*" " " " "

19 ( 1 ) " " "\*" " " " "

20 ( 1 ) " " "\*" " " " "

21 ( 1 ) " " "\*" " " " "

22 ( 1 ) " " "\*" " " " "

23 ( 1 ) " " "\*" " " " "

24 ( 1 ) " " "\*" " " " "

25 ( 1 ) " " "\*" " " " "

26 ( 1 ) " " "\*" " " " "

27 ( 1 ) "\*" "\*" " " " "

28 ( 1 ) "\*" "\*" "\*" " "

29 ( 1 ) "\*" "\*" "\*" " "

30 ( 1 ) "\*" "\*" "\*" "\*"

q\_owner\_downvotes a\_votes\_up a\_votes\_down q\_votes\_up q\_votes\_down

1 ( 1 ) " " " " " " " " " "

2 ( 1 ) " " " " " " " " " "

3 ( 1 ) " " " " " " " " " "

4 ( 1 ) " " " " " " " " " "

5 ( 1 ) " " " " " " " " " "

6 ( 1 ) " " " " " " " " " "

7 ( 1 ) " " " " " " " " " "

8 ( 1 ) " " " " " " " " " "

9 ( 1 ) " " " " " " " " " "

10 ( 1 ) " " "\*" " " " " " "

11 ( 1 ) " " "\*" " " " " " "

12 ( 1 ) " " "\*" " " " " " "

13 ( 1 ) " " "\*" " " " " " "

14 ( 1 ) " " "\*" " " " " " "

15 ( 1 ) " " "\*" " " " " " "

16 ( 1 ) " " "\*" " " " " " "

17 ( 1 ) " " "\*" " " " " " "

18 ( 1 ) " " "\*" " " " " " "

19 ( 1 ) " " "\*" " " " " " "

20 ( 1 ) " " "\*" " " " " " "

21 ( 1 ) " " "\*" " " " " "\*"

22 ( 1 ) " " "\*" " " " " "\*"

23 ( 1 ) " " "\*" " " " " "\*"

24 ( 1 ) " " "\*" " " " " "\*"

25 ( 1 ) " " "\*" " " "\*" "\*"

26 ( 1 ) "\*" "\*" " " "\*" "\*"

27 ( 1 ) "\*" "\*" " " "\*" "\*"

28 ( 1 ) "\*" "\*" " " "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*" "\*"

> regfit.bwd=regsubsets(accepted\_answer\_flag~.,data=Q2df,nvmax=32,method="backward"

+ )

> summary(regfit.bwd)

Subset selection object

Call: regsubsets.formula(accepted\_answer\_flag ~ ., data = Q2df, nvmax = 32,

method = "backward")

30 Variables (and intercept)

Forced in Forced out

a\_score FALSE FALSE

a\_body\_length FALSE FALSE

a\_body\_has\_code FALSE FALSE

a\_DaysOld FALSE FALSE

a\_has\_edited FALSE FALSE

a\_num\_comment FALSE FALSE

a\_owner\_reputation FALSE FALSE

a\_owner\_profile\_summary FALSE FALSE

a\_owner\_views FALSE FALSE

a\_owner\_upvotes FALSE FALSE

a\_owner\_downvotes FALSE FALSE

q\_score FALSE FALSE

q\_num\_views FALSE FALSE

q\_body\_length FALSE FALSE

q\_body\_has\_code FALSE FALSE

q\_DaysOld FALSE FALSE

q\_has\_edited FALSE FALSE

q\_title\_length FALSE FALSE

q\_num\_tags FALSE FALSE

q\_num\_answers FALSE FALSE

q\_num\_comment FALSE FALSE

q\_owner\_reputation FALSE FALSE

q\_owner\_profile\_summary FALSE FALSE

q\_owner\_views FALSE FALSE

q\_owner\_upvotes FALSE FALSE

q\_owner\_downvotes FALSE FALSE

a\_votes\_up FALSE FALSE

a\_votes\_down FALSE FALSE

q\_votes\_up FALSE FALSE

q\_votes\_down FALSE FALSE

1 subsets of each size up to 30

Selection Algorithm: backward

a\_score a\_body\_length a\_body\_has\_code a\_DaysOld a\_has\_edited a\_num\_comment

1 ( 1 ) " " " " " " " " " " "\*"

2 ( 1 ) " " " " " " " " " " "\*"

3 ( 1 ) " " " " " " " " " " "\*"

4 ( 1 ) "\*" " " " " " " " " "\*"

5 ( 1 ) "\*" "\*" " " " " " " "\*"

6 ( 1 ) "\*" "\*" " " " " " " "\*"

7 ( 1 ) "\*" "\*" " " " " "\*" "\*"

8 ( 1 ) "\*" "\*" " " " " "\*" "\*"

9 ( 1 ) "\*" "\*" " " " " "\*" "\*"

10 ( 1 ) "\*" "\*" " " " " "\*" "\*"

11 ( 1 ) "\*" "\*" " " " " "\*" "\*"

12 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

13 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

14 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

15 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

16 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

17 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

18 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

19 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

20 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

21 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

22 ( 1 ) "\*" "\*" "\*" " " "\*" "\*"

23 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

24 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

25 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

26 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

27 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

28 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

a\_owner\_reputation a\_owner\_profile\_summary a\_owner\_views a\_owner\_upvotes

1 ( 1 ) " " " " " " " "

2 ( 1 ) " " " " " " " "

3 ( 1 ) "\*" " " " " " "

4 ( 1 ) "\*" " " " " " "

5 ( 1 ) "\*" " " " " " "

6 ( 1 ) "\*" " " " " " "

7 ( 1 ) "\*" " " " " " "

8 ( 1 ) "\*" "\*" " " " "

9 ( 1 ) "\*" "\*" " " " "

10 ( 1 ) "\*" "\*" " " " "

11 ( 1 ) "\*" "\*" "\*" " "

12 ( 1 ) "\*" "\*" "\*" " "

13 ( 1 ) "\*" "\*" "\*" " "

14 ( 1 ) "\*" "\*" "\*" " "

15 ( 1 ) "\*" "\*" "\*" " "

16 ( 1 ) "\*" "\*" "\*" " "

17 ( 1 ) "\*" "\*" "\*" " "

18 ( 1 ) "\*" "\*" "\*" " "

19 ( 1 ) "\*" "\*" "\*" " "

20 ( 1 ) "\*" "\*" "\*" " "

21 ( 1 ) "\*" "\*" "\*" " "

22 ( 1 ) "\*" "\*" "\*" " "

23 ( 1 ) "\*" "\*" "\*" " "

24 ( 1 ) "\*" "\*" "\*" "\*"

25 ( 1 ) "\*" "\*" "\*" "\*"

26 ( 1 ) "\*" "\*" "\*" "\*"

27 ( 1 ) "\*" "\*" "\*" "\*"

28 ( 1 ) "\*" "\*" "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*"

a\_owner\_downvotes q\_score q\_num\_views q\_body\_length q\_body\_has\_code q\_DaysOld

1 ( 1 ) " " " " " " " " " " " "

2 ( 1 ) " " " " " " " " " " "\*"

3 ( 1 ) " " " " " " " " " " "\*"

4 ( 1 ) " " " " " " " " " " "\*"

5 ( 1 ) " " " " " " " " " " "\*"

6 ( 1 ) "\*" " " " " " " " " "\*"

7 ( 1 ) "\*" " " " " " " " " "\*"

8 ( 1 ) "\*" " " " " " " " " "\*"

9 ( 1 ) "\*" " " " " " " " " "\*"

10 ( 1 ) "\*" " " " " " " " " "\*"

11 ( 1 ) "\*" " " " " " " " " "\*"

12 ( 1 ) "\*" " " " " " " " " "\*"

13 ( 1 ) "\*" " " " " " " " " "\*"

14 ( 1 ) "\*" " " " " " " " " "\*"

15 ( 1 ) "\*" " " " " " " " " "\*"

16 ( 1 ) "\*" " " " " " " "\*" "\*"

17 ( 1 ) "\*" " " " " " " "\*" "\*"

18 ( 1 ) "\*" "\*" " " " " "\*" "\*"

19 ( 1 ) "\*" "\*" " " "\*" "\*" "\*"

20 ( 1 ) "\*" "\*" " " "\*" "\*" "\*"

21 ( 1 ) "\*" "\*" " " "\*" "\*" "\*"

22 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

23 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

24 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

25 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

26 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

27 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

28 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

q\_has\_edited q\_title\_length q\_num\_tags q\_num\_answers q\_num\_comment

1 ( 1 ) " " " " " " " " " "

2 ( 1 ) " " " " " " " " " "

3 ( 1 ) " " " " " " " " " "

4 ( 1 ) " " " " " " " " " "

5 ( 1 ) " " " " " " " " " "

6 ( 1 ) " " " " " " " " " "

7 ( 1 ) " " " " " " " " " "

8 ( 1 ) " " " " " " " " " "

9 ( 1 ) " " " " " " " " "\*"

10 ( 1 ) " " " " " " " " "\*"

11 ( 1 ) " " " " " " " " "\*"

12 ( 1 ) " " " " " " " " "\*"

13 ( 1 ) " " "\*" " " " " "\*"

14 ( 1 ) " " "\*" " " " " "\*"

15 ( 1 ) " " "\*" "\*" " " "\*"

16 ( 1 ) " " "\*" "\*" " " "\*"

17 ( 1 ) " " "\*" "\*" "\*" "\*"

18 ( 1 ) " " "\*" "\*" "\*" "\*"

19 ( 1 ) " " "\*" "\*" "\*" "\*"

20 ( 1 ) "\*" "\*" "\*" "\*" "\*"

21 ( 1 ) "\*" "\*" "\*" "\*" "\*"

22 ( 1 ) "\*" "\*" "\*" "\*" "\*"

23 ( 1 ) "\*" "\*" "\*" "\*" "\*"

24 ( 1 ) "\*" "\*" "\*" "\*" "\*"

25 ( 1 ) "\*" "\*" "\*" "\*" "\*"

26 ( 1 ) "\*" "\*" "\*" "\*" "\*"

27 ( 1 ) "\*" "\*" "\*" "\*" "\*"

28 ( 1 ) "\*" "\*" "\*" "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*" "\*"

q\_owner\_reputation q\_owner\_profile\_summary q\_owner\_views q\_owner\_upvotes

1 ( 1 ) " " " " " " " "

2 ( 1 ) " " " " " " " "

3 ( 1 ) " " " " " " " "

4 ( 1 ) " " " " " " " "

5 ( 1 ) " " " " " " " "

6 ( 1 ) " " " " " " " "

7 ( 1 ) " " " " " " " "

8 ( 1 ) " " " " " " " "

9 ( 1 ) " " " " " " " "

10 ( 1 ) " " " " " " " "

11 ( 1 ) " " " " " " " "

12 ( 1 ) " " " " " " " "

13 ( 1 ) " " " " " " " "

14 ( 1 ) " " "\*" " " " "

15 ( 1 ) " " "\*" " " " "

16 ( 1 ) " " "\*" " " " "

17 ( 1 ) " " "\*" " " " "

18 ( 1 ) " " "\*" " " " "

19 ( 1 ) " " "\*" " " " "

20 ( 1 ) " " "\*" " " " "

21 ( 1 ) " " "\*" " " " "

22 ( 1 ) " " "\*" " " " "

23 ( 1 ) " " "\*" " " " "

24 ( 1 ) " " "\*" " " " "

25 ( 1 ) " " "\*" " " " "

26 ( 1 ) " " "\*" " " " "

27 ( 1 ) "\*" "\*" " " " "

28 ( 1 ) "\*" "\*" "\*" " "

29 ( 1 ) "\*" "\*" "\*" " "

30 ( 1 ) "\*" "\*" "\*" "\*"

q\_owner\_downvotes a\_votes\_up a\_votes\_down q\_votes\_up q\_votes\_down

1 ( 1 ) " " " " " " " " " "

2 ( 1 ) " " " " " " " " " "

3 ( 1 ) " " " " " " " " " "

4 ( 1 ) " " " " " " " " " "

5 ( 1 ) " " " " " " " " " "

6 ( 1 ) " " " " " " " " " "

7 ( 1 ) " " " " " " " " " "

8 ( 1 ) " " " " " " " " " "

9 ( 1 ) " " " " " " " " " "

10 ( 1 ) " " "\*" " " " " " "

11 ( 1 ) " " "\*" " " " " " "

12 ( 1 ) " " "\*" " " " " " "

13 ( 1 ) " " "\*" " " " " " "

14 ( 1 ) " " "\*" " " " " " "

15 ( 1 ) " " "\*" " " " " " "

16 ( 1 ) " " "\*" " " " " " "

17 ( 1 ) " " "\*" " " " " " "

18 ( 1 ) " " "\*" " " " " " "

19 ( 1 ) " " "\*" " " " " " "

20 ( 1 ) " " "\*" " " " " " "

21 ( 1 ) " " "\*" " " "\*" " "

22 ( 1 ) " " "\*" " " "\*" " "

23 ( 1 ) " " "\*" " " "\*" " "

24 ( 1 ) " " "\*" " " "\*" " "

25 ( 1 ) " " "\*" " " "\*" "\*"

26 ( 1 ) "\*" "\*" " " "\*" "\*"

27 ( 1 ) "\*" "\*" " " "\*" "\*"

28 ( 1 ) "\*" "\*" " " "\*" "\*"

29 ( 1 ) "\*" "\*" "\*" "\*" "\*"

30 ( 1 ) "\*" "\*" "\*" "\*" "\*"

> coef(regfit.full,7)

(Intercept) a\_score a\_body\_length a\_has\_edited

5.024723e-02 2.292875e-03 1.033937e-05 2.020404e-02

a\_num\_comment a\_owner\_reputation a\_owner\_downvotes q\_DaysOld

2.046747e-02 3.563762e-07 1.032462e-05 -3.338431e-05

> coef(regfit.fwd,7)

(Intercept) a\_score a\_body\_length a\_has\_edited

5.024723e-02 2.292875e-03 1.033937e-05 2.020404e-02

a\_num\_comment a\_owner\_reputation a\_owner\_downvotes q\_DaysOld

2.046747e-02 3.563762e-07 1.032462e-05 -3.338431e-05

> coef(regfit.bwd,7)

(Intercept) a\_score a\_body\_length a\_has\_edited

5.024723e-02 2.292875e-03 1.033937e-05 2.020404e-02

a\_num\_comment a\_owner\_reputation a\_owner\_downvotes q\_DaysOld

2.046747e-02 3.563762e-07 1.032462e-05 -3.338431e-05

#Choosing among models

> set.seed(1)

> train=sample(c(TRUE,FALSE), nrow(Q2df),rep=TRUE)

> test=(!train)

> regfit.best=regsubsets(accepted\_answer\_flag~.,data=Q2df[train,],nvmax=32)

> test.mat=model.matrix(accepted\_answer\_flag~.,data=Q2df[test,])

> val.errors=rep(NA,31)

> for(i in 1:11){

+ coefi=coef(regfit.best,id=i)

+ pred=test.mat[,names(coefi)]%\*%coefi

+ val.errors[i]=mean((Q2df$accepted\_answer\_flag[test]-pred)^2)

+ }

> val.errors

[1] 0.04417120 0.04307115 0.04263821 0.04240312 0.04232690 0.04225756 0.04221715

[8] 0.04202685 0.04189354 0.04186988 0.04183836 NA NA NA

[15] NA NA NA NA NA NA NA

[22] NA NA NA NA NA NA NA

[29] NA NA NA

> which.min(val.errors)

[1] 11

> coef(regfit.best,30)

(Intercept) a\_score a\_body\_length

5.675943e-02 6.827364e-03 9.133316e-06

a\_body\_has\_code a\_DaysOld a\_has\_edited

1.005307e-02 8.933178e-06 2.144292e-02

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

2.205907e-02 3.755831e-07 7.262864e-03

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

-5.451486e-07 3.737181e-07 5.662703e-06

q\_score q\_num\_views q\_body\_length

3.870241e-03 1.105329e-08 -5.133739e-07

q\_body\_has\_code q\_DaysOld q\_has\_edited

9.744931e-03 -3.377767e-05 -9.028247e-03

q\_title\_length q\_num\_tags q\_num\_answers

-1.574846e-04 -2.719542e-03 -6.679787e-04

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

-1.846335e-03 2.605342e-09 3.017549e-03

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

-6.214334e-10 1.149192e-06 -2.910958e-06

a\_votes\_up a\_votes\_down q\_votes\_up

-5.873875e-03 -3.631236e-03 -3.858212e-03

q\_votes\_down

2.999433e-03

> predict.regsubsets=function(object,newdata,id,...){

+ form=as.formula(object$call[[2]])

+ mat=model.matrix(form,newdata)

+ coefi=coef(object,id=id)

+ xvars=names(coefi)

+ mat[,xvars]%\*%coefi

+ }

> regfit.best=regsubsets(accepted\_answer\_flag~.,data=Q2df,nvmax=32)

> coef(regfit.best,30)

(Intercept) a\_score a\_body\_length

5.371705e-02 1.512208e-02 9.696537e-06

a\_body\_has\_code a\_DaysOld a\_has\_edited

9.893681e-03 2.960808e-05 2.027683e-02

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

2.119361e-02 4.027408e-07 6.872559e-03

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

-5.361746e-07 7.719789e-07 9.647858e-06

q\_score q\_num\_views q\_body\_length

5.681167e-03 1.427731e-08 -1.041190e-06

q\_body\_has\_code q\_DaysOld q\_has\_edited

7.342933e-03 -3.315512e-05 -3.560755e-03

q\_title\_length q\_num\_tags q\_num\_answers

-1.732995e-04 -2.613280e-03 -7.094397e-04

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

-1.920190e-03 6.666180e-09 3.299104e-03

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

-7.358881e-09 -3.740911e-08 6.895790e-07

a\_votes\_up a\_votes\_down q\_votes\_up

-1.277825e-02 -3.620341e-04 -5.674160e-03

q\_votes\_down

4.907028e-03

> k=30

> set.seed(1)

> folds=sample(1:k,nrow(Q2df),replace=TRUE)

> cv.errors=matrix(NA,k,31, dimnames=list(NULL, paste(1:31)))

> for(j in 1:k){

+ best.fit=regsubsets(accepted\_answer\_flag~.,data=Q2df[folds!=j,],nvmax=30)

+ for(i in 1:30){

+ pred=predict(best.fit,Q2df[folds==j,],id=i)

+ cv.errors[j,i]=mean( (Q2df$accepted\_answer\_flag[folds==j]-pred)^2)

+ }

+ }

> mean.cv.errors=apply(cv.errors,2,mean)

> mean.cv.errors

1 2 3 4 5 6 7 8

0.04442586 0.04330650 0.04418829 0.04393870 0.04357534 0.04352206 0.04351628 0.04334557

9 10 11 12 13 14 15 16

0.04324305 0.04324965 0.04320858 0.04316611 0.04317462 0.04316156 0.04312791 0.04312941

17 18 19 20 21 22 23 24

0.04313519 0.04312605 0.04314404 0.04313847 0.04314949 0.04314467 0.04314280 0.04314135

25 26 27 28 29 30 31

0.04314161 0.04314418 0.04314640 0.04314731 0.04314730 0.04314736 NA

> par(mfrow=c(1,1))

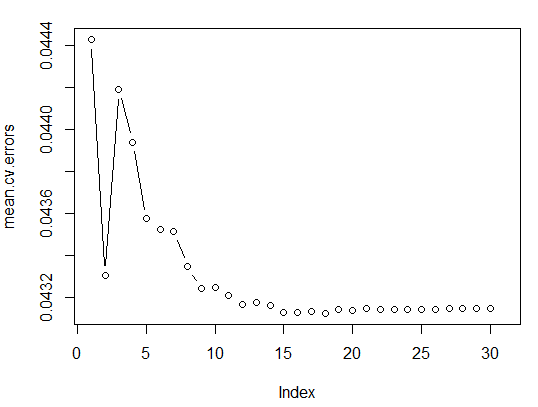
> plot(mean.cv.errors,type='b')

> reg.best=regsubsets(accepted\_answer\_flag~.,data=Q2df, nvmax=30)

> coef(reg.best,3)

(Intercept) a\_num\_comment a\_owner\_reputation q\_DaysOld

6.382674e-02 2.535243e-02 5.218958e-07 -3.352392e-05



# Ridge Regression and LASSO

> fix(Q2df)

> names(Q2df)

[1] "a\_score" "a\_body\_length" "a\_body\_has\_code"

[4] "a\_DaysOld" "a\_has\_edited" "a\_num\_comment"

[7] "a\_owner\_reputation" "a\_owner\_profile\_summary" "a\_owner\_views"

[10] "a\_owner\_upvotes" "a\_owner\_downvotes" "q\_score"

[13] "q\_num\_views" "q\_body\_length" "q\_body\_has\_code"

[16] "q\_DaysOld" "q\_has\_edited" "q\_title\_length"

[19] "q\_num\_tags" "q\_num\_answers" "q\_num\_comment"

[22] "q\_owner\_reputation" "q\_owner\_profile\_summary" "q\_owner\_views"

[25] "q\_owner\_upvotes" "q\_owner\_downvotes" "accepted\_answer\_flag"

[28] "a\_votes\_up" "a\_votes\_down" "q\_votes\_up"

[31] "q\_votes\_down"

> dim(Q2df)

[1] 50000 31

> sum(is.na(Q2df$accepted\_answer\_flag))

[1] 0

> Q2df=na.omit(Q2df)

> dim(Q2df)

[1] 50000 31

> sum(is.na(Q2df))

[1] 0

> install.packages("glmnet")

> library(glmnet)

> x=model.matrix(accepted\_answer\_flag~.,Q2df)[,-1]

> y=Q2df$accepted\_answer\_flag

> grid=10^seq(10,-2,length=100)

> ridge.mod=glmnet(x,y,alpha=0,lambda=grid)

> dim(coef(ridge.mod))

[1] 31 100

#First lambda=50

> ridge.mod$lambda[50]

[1] 11497.57

> coef(ridge.mod)[,50]

(Intercept) a\_score a\_body\_length

4.965874e-02 9.544193e-08 3.390952e-10

a\_body\_has\_code a\_DaysOld a\_has\_edited

8.228196e-07 1.418453e-09 1.049119e-06

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

5.937375e-07 1.456741e-11 3.361188e-07

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

1.514413e-11 3.009046e-10 4.290594e-10

q\_score q\_num\_views q\_body\_length

-7.118967e-10 -1.983717e-12 6.352320e-11

q\_body\_has\_code q\_DaysOld q\_has\_edited

8.070656e-07 -8.894927e-10 -1.207279e-07

q\_title\_length q\_num\_tags q\_num\_answers

-1.325088e-09 -5.207115e-08 -7.654759e-08

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

3.826322e-08 -8.786359e-12 -1.458687e-07

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

-7.224793e-12 -1.277464e-10 -3.174198e-11

a\_votes\_up a\_votes\_down q\_votes\_up

9.533802e-08 1.451247e-07 -7.056087e-10

q\_votes\_down

2.704076e-08

> sqrt(sum(coef(ridge.mod)[-1,50]^2))

[1] 1.72639e-06

#Another lambda value as 60

> ridge.mod$lambda[60]

[1] 705.4802

> coef(ridge.mod)[,60]

(Intercept) a\_score a\_body\_length

4.963943e-02 1.554324e-06 5.523137e-09

a\_body\_has\_code a\_DaysOld a\_has\_edited

1.340052e-05 2.309995e-08 1.708737e-05

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

9.671047e-06 2.372238e-10 5.474225e-06

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

2.464673e-10 4.899433e-09 6.987188e-09

q\_score q\_num\_views q\_body\_length

-1.158571e-08 -3.229229e-11 1.034109e-09

q\_body\_has\_code q\_DaysOld q\_has\_edited

1.314255e-05 -1.448678e-08 -1.966776e-06

q\_title\_length q\_num\_tags q\_num\_answers

-2.160856e-08 -8.482427e-07 -1.246325e-06

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

6.227354e-07 -1.430103e-10 -2.374216e-06

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

-1.175278e-10 -2.079164e-09 -5.153232e-10

a\_votes\_up a\_votes\_down q\_votes\_up

1.552400e-06 2.360953e-06 -1.147938e-08

q\_votes\_down

4.408106e-07

> sqrt(sum(coef(ridge.mod)[-1,60]^2))

[1] 2.811649e-05

> predict(ridge.mod,s=500,type="coefficients")[1:20,]

(Intercept) a\_score a\_body\_length

4.963055e-02 2.225301e-06 7.907933e-09

a\_body\_has\_code a\_DaysOld a\_has\_edited

1.918563e-05 3.307160e-08 2.446504e-05

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

1.384707e-05 3.396194e-10 7.837602e-06

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

3.527497e-10 7.013769e-09 1.000325e-08

q\_score q\_num\_views q\_body\_length

-1.658157e-08 -4.622287e-11 1.480238e-09

q\_body\_has\_code q\_DaysOld q\_has\_edited

1.881531e-05 -2.074113e-08 -2.816259e-06

q\_title\_length q\_num\_tags

-3.095666e-08 -1.214579e-06

> set.seed(1)

> predict(ridge.mod,s=500,type="coefficients")[1:31,]

(Intercept) a\_score a\_body\_length

4.963055e-02 2.225301e-06 7.907933e-09

a\_body\_has\_code a\_DaysOld a\_has\_edited

1.918563e-05 3.307160e-08 2.446504e-05

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

1.384707e-05 3.396194e-10 7.837602e-06

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

3.527497e-10 7.013769e-09 1.000325e-08

q\_score q\_num\_views q\_body\_length

-1.658157e-08 -4.622287e-11 1.480238e-09

q\_body\_has\_code q\_DaysOld q\_has\_edited

1.881531e-05 -2.074113e-08 -2.816259e-06

q\_title\_length q\_num\_tags q\_num\_answers

-3.095666e-08 -1.214579e-06 -1.784140e-06

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

8.912803e-07 -2.046897e-10 -3.398197e-06

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

-1.681709e-10 -2.975833e-09 -7.366592e-10

a\_votes\_up a\_votes\_down q\_votes\_up

2.222386e-06 3.378411e-06 -1.642661e-08

q\_votes\_down

6.314052e-07

#Set of instructions for training and validation set testing preparation

> set.seed(1)

> train=sample(1:nrow(x), nrow(x)/2)

> test=(-train)

> y.test=y[test]

> ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid, thresh=1e-12)

> ridge.pred=predict(ridge.mod,s=4,newx=x[test,])

> mean((ridge.pred-y.test)^2)

[1] 0.04545285

> mean((mean(y[train])-y.test)^2)

[1] 0.0466725

> ridge.pred=predict(ridge.mod,s=1e10,newx=x[test,])

> mean((ridge.pred-y.test)^2)

[1] 0.0466725

> ridge.pred=predict(ridge.mod,s=0,newx=x[test,],exact=T)

> mean((ridge.pred-y.test)^2)

[1] 0.04314134

> lm(y~x, subset=train)

Call:

lm(formula = y ~ x, subset = train)

Coefficients:

(Intercept) xa\_score xa\_body\_length

5.381e-02 1.931e-02 1.120e-05

xa\_body\_has\_code xa\_DaysOld xa\_has\_edited

8.432e-03 1.365e-05 1.889e-02

xa\_num\_comment xa\_owner\_reputation xa\_owner\_profile\_summary

2.026e-02 4.301e-07 5.245e-03

xa\_owner\_views xa\_owner\_upvotes xa\_owner\_downvotes

-5.532e-07 1.505e-06 9.270e-06

xq\_score xq\_num\_views xq\_body\_length

8.815e-03 2.315e-08 -8.239e-07

xq\_body\_has\_code xq\_DaysOld xq\_has\_edited

9.619e-03 -3.195e-05 -7.432e-03

xq\_title\_length xq\_num\_tags xq\_num\_answers

-1.297e-04 -2.735e-03 -8.836e-04

xq\_num\_comment xq\_owner\_reputation xq\_owner\_profile\_summary

-1.818e-03 -1.730e-09 2.854e-03

xq\_owner\_views xq\_owner\_upvotes xq\_owner\_downvotes

-3.112e-09 -6.986e-07 4.280e-06

xa\_votes\_up xa\_votes\_down xq\_votes\_up

-1.320e-02 -9.947e-04 -8.810e-03

xq\_votes\_down

7.431e-03

> predict(ridge.mod,s=500,type="coefficients")[1:31,]

(Intercept) a\_score a\_body\_length

5.020358e-02 4.383717e-06 8.288319e-09

a\_body\_has\_code a\_DaysOld a\_has\_edited

1.848033e-05 4.526825e-08 2.530783e-05

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

1.417718e-05 3.542239e-10 7.510325e-06

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

2.978364e-10 7.589482e-09 1.018441e-08

q\_score q\_num\_views q\_body\_length

-1.468888e-08 -4.573746e-11 1.360741e-09

q\_body\_has\_code q\_DaysOld q\_has\_edited

1.941020e-05 -2.096316e-08 -4.046018e-06

q\_title\_length q\_num\_tags q\_num\_answers

4.654568e-09 -1.109864e-06 -1.778827e-06

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

9.660916e-07 -1.820433e-10 -3.267291e-06

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

-8.552710e-11 -2.780471e-09 1.365295e-09

a\_votes\_up a\_votes\_down q\_votes\_up

4.335151e-06 4.085286e-06 -1.457423e-08

q\_votes\_down

5.549107e-07

> set.seed(1)

> train=sample(1:nrow(x), nrow(x)/2)

> test=(-train)

> y.test=y[test]

> ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid, thresh=1e-12)

#The above functions tests and evaluates the test MSE values in addition to the spread of y.test.

#testing with other two lambdas

> ridge.pred=predict(ridge.mod,s=4,newx=x[test,])

> mean((ridge.pred-y.test)^2)

[1] 0.04545285

> mean((mean(y[train])-y.test)^2)

[1] 0.0466725

> ridge.pred=predict(ridge.mod,s=1e10,newx=x[test,])

> mean((ridge.pred-y.test)^2)

[1] 0.0466725

> ridge.pred=predict(ridge.mod,s=0,newx=x[test,],exact=T)

> mean((ridge.pred-y.test)^2)

[1] 0.04314134

#Comparison:

> lm(y~x, subset=train)

Call:

lm(formula = y ~ x, subset = train)

Coefficients:

(Intercept) xa\_score xa\_body\_length

5.381e-02 1.931e-02 1.120e-05

xa\_body\_has\_code xa\_DaysOld xa\_has\_edited

8.432e-03 1.365e-05 1.889e-02

xa\_num\_comment xa\_owner\_reputation xa\_owner\_profile\_summary

2.026e-02 4.301e-07 5.245e-03

xa\_owner\_views xa\_owner\_upvotes xa\_owner\_downvotes

-5.532e-07 1.505e-06 9.270e-06

xq\_score xq\_num\_views xq\_body\_length

8.815e-03 2.315e-08 -8.239e-07

xq\_body\_has\_code xq\_DaysOld xq\_has\_edited

9.619e-03 -3.195e-05 -7.432e-03

xq\_title\_length xq\_num\_tags xq\_num\_answers

-1.297e-04 -2.735e-03 -8.836e-04

xq\_num\_comment xq\_owner\_reputation xq\_owner\_profile\_summary

-1.818e-03 -1.730e-09 2.854e-03

xq\_owner\_views xq\_owner\_upvotes xq\_owner\_downvotes

-3.112e-09 -6.986e-07 4.280e-06

xa\_votes\_up xa\_votes\_down xq\_votes\_up

-1.320e-02 -9.947e-04 -8.810e-03

xq\_votes\_down

7.431e-03

> predict(ridge.mod,s=0,exact=T,type="coefficients")[1:31,]

(Intercept) a\_score a\_body\_length

5.116753e-02 3.276604e-03 1.076287e-05

a\_body\_has\_code a\_DaysOld a\_has\_edited

8.839150e-03 1.569088e-05 1.891559e-02

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

1.948744e-02 3.867904e-07 5.338781e-03

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

-4.732919e-07 2.078318e-06 9.183022e-06

q\_score q\_num\_views q\_body\_length

2.637408e-06 1.658474e-08 -6.885787e-07

q\_body\_has\_code q\_DaysOld q\_has\_edited

1.017657e-02 -3.022116e-05 -7.342307e-03

q\_title\_length q\_num\_tags q\_num\_answers

-1.189693e-04 -2.552383e-03 -9.383281e-04

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

-1.624073e-03 -1.143852e-08 2.430944e-03

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

5.752874e-10 -7.323150e-07 4.188581e-06

a\_votes\_up a\_votes\_down q\_votes\_up

2.796166e-03 -1.564107e-02 2.509044e-06

q\_votes\_down

-1.129070e-03

#Cross validation to get the best lambda

> set.seed(1)

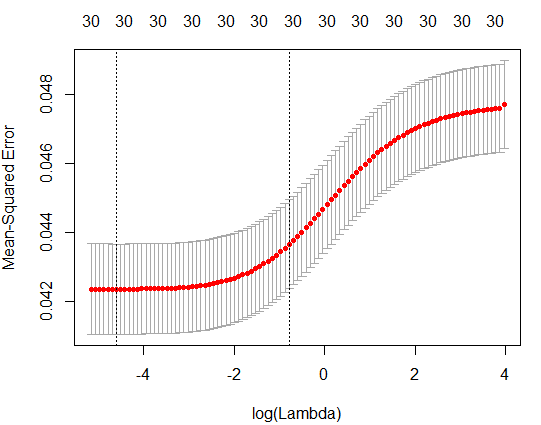
> cv.out=cv.glmnet(x[train,],y[train],alpha=0)

> plot(cv.out)

> bestlam=cv.out$lambda.min

> bestlam

[1] 0.01010282



# Prediction with the best lambda

> ridge.pred=predict(ridge.mod,s=bestlam,newx=x[test,])

> mean((ridge.pred-y.test)^2)

[1] 0.0431409

> out=glmnet(x,y,alpha=0)

> predict(out,type="coefficients",s=bestlam)[1:31,]

(Intercept) a\_score a\_body\_length

5.124974e-02 1.287146e-03 9.322903e-06

a\_body\_has\_code a\_DaysOld a\_has\_edited

1.025606e-02 2.997802e-05 2.025861e-02

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

2.031892e-02 3.600757e-07 6.865925e-03

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

-4.534893e-07 1.399864e-06 9.517149e-06

q\_score q\_num\_views q\_body\_length

3.904759e-06 9.001508e-09 -8.904426e-07

q\_body\_has\_code q\_DaysOld q\_has\_edited

8.092279e-03 -3.128807e-05 -3.696886e-03

q\_title\_length q\_num\_tags q\_num\_answers

-1.609980e-04 -2.458631e-03 -7.791342e-04

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

-1.710223e-03 -4.261023e-09 2.817014e-03

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

-1.510854e-09 -1.465659e-07 7.684724e-07

a\_votes\_up a\_votes\_down q\_votes\_up

1.087446e-03 -1.289248e-02 3.273748e-06

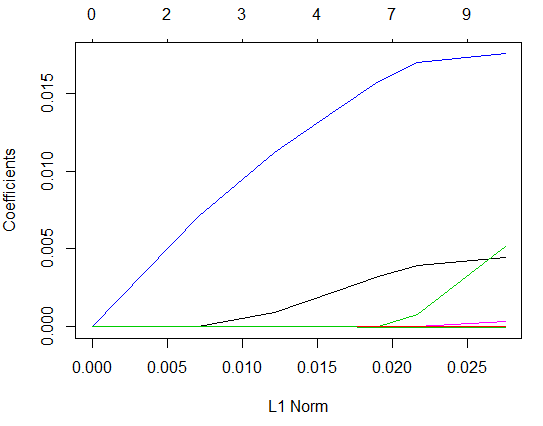
q\_votes\_down

-5.895804e-04

#Lasso

> lasso.mod=glmnet(x[train,],y[train],alpha=1,lambda=grid)

> plot(lasso.mod)



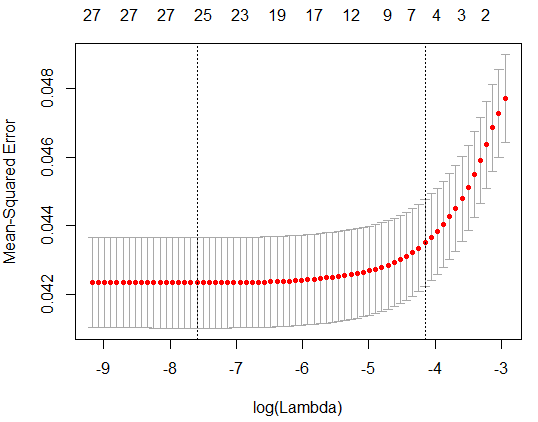
# using CV to get best lambda:

> set.seed(1)

> cv.out=cv.glmnet(x[train,],y[train],alpha=1)

> plot(cv.out)

> bestlam=cv.out$lambda.min



#Using best lambda for prediction

> lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])

> mean((lasso.pred-y.test)^2)

[1] 0.04280294

> out=glmnet(x,y,alpha=1,lambda=grid)

> lasso.coef=predict(out,type="coefficients",s=bestlam)[1:31,]

> lasso.coef

(Intercept) a\_score a\_body\_length

5.367260e-02 1.216393e-03 3.820922e-06

a\_body\_has\_code a\_DaysOld a\_has\_edited

0.000000e+00 0.000000e+00 5.951271e-03

a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

1.880085e-02 2.058404e-07 1.832329e-03

a\_owner\_views a\_owner\_upvotes a\_owner\_downvotes

0.000000e+00 0.000000e+00 4.565356e-06

q\_score q\_num\_views q\_body\_length

0.000000e+00 0.000000e+00 0.000000e+00

q\_body\_has\_code q\_DaysOld q\_has\_edited

0.000000e+00 -2.599163e-05 0.000000e+00

q\_title\_length q\_num\_tags q\_num\_answers

0.000000e+00 0.000000e+00 0.000000e+00

q\_num\_comment q\_owner\_reputation q\_owner\_profile\_summary

0.000000e+00 0.000000e+00 0.000000e+00

q\_owner\_views q\_owner\_upvotes q\_owner\_downvotes

0.000000e+00 0.000000e+00 0.000000e+00

a\_votes\_up a\_votes\_down q\_votes\_up

0.000000e+00 0.000000e+00 0.000000e+00

q\_votes\_down

0.000000e+00

> lasso.coef[lasso.coef!=0]

(Intercept) a\_score a\_body\_length

5.367260e-02 1.216393e-03 3.820922e-06

a\_has\_edited a\_num\_comment a\_owner\_reputation

5.951271e-03 1.880085e-02 2.058404e-07

a\_owner\_profile\_summary a\_owner\_downvotes q\_DaysOld

1.832329e-03 4.565356e-06 -2.599163e-05

#Dimensionality Reduction PCA:

#> install.packages("pls")

> library(pls)

> set.seed(2)

> pcr.fit=pcr(accepted\_answer\_flag~., data=Q2df,scale=TRUE,validation="CV")

> summary(pcr.fit)

Data: X dimension: 50000 30

Y dimension: 50000 1

Fit method: svdpc

Number of components considered: 30

VALIDATION: RMSEP

Cross-validated using 10 random segments.

(Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps

CV 0.2172 0.2156 0.2107 0.2107 0.2119 0.2115 0.2109 0.2108 0.2110

adjCV 0.2172 0.2156 0.2106 0.2107 0.2117 0.2113 0.2107 0.2106 0.2108

9 comps 10 comps 11 comps 12 comps 13 comps 14 comps 15 comps 16 comps

CV 0.2104 0.2104 0.2104 0.2103 0.2103 0.2103 0.2103 0.2097

adjCV 0.2102 0.2102 0.2102 0.2101 0.2101 0.2101 0.2101 0.2094

17 comps 18 comps 19 comps 20 comps 21 comps 22 comps 23 comps 24 comps

CV 0.2094 0.2093 0.2092 0.2082 0.2081 0.2081 0.2081 0.2081

adjCV 0.2093 0.2091 0.2091 0.2080 0.2080 0.2079 0.2079 0.2079

25 comps 26 comps 27 comps 28 comps 29 comps 30 comps

CV 0.2077 0.2077 0.2076 0.2076 0.2076 0.2076

adjCV 0.2075 0.2075 0.2075 0.2074 0.2074 0.2074

TRAINING: % variance explained

1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps

X 12.598 21.858 29.354 36.466 41.837 46.274 50.343

accepted\_answer\_flag 1.559 7.154 7.168 7.645 7.906 8.371 8.424

8 comps 9 comps 10 comps 11 comps 12 comps 13 comps 14 comps

X 54.182 57.642 60.963 64.194 67.299 70.301 73.249

accepted\_answer\_flag 8.436 8.768 8.774 8.792 8.821 8.826 8.831

15 comps 16 comps 17 comps 18 comps 19 comps 20 comps 21 comps

X 75.943 78.569 81.171 83.550 85.865 88.03 90.13

accepted\_answer\_flag 8.854 9.388 9.407 9.522 9.522 10.27 10.29

22 comps 23 comps 24 comps 25 comps 26 comps 27 comps 28 comps

X 92.16 94.11 95.90 97.33 98.42 99.29 100.0

accepted\_answer\_flag 10.34 10.34 10.34 10.63 10.63 10.65 10.7

29 comps 30 comps

X 100.0 100.0

accepted\_answer\_flag 10.7 10.7

> validationplot(pcr.fit,val.type="MSEP")

> set.seed(1)

> pcr.fit=pcr(accepted\_answer\_flag~., data=Q2df,subset=train,scale=TRUE, validation="CV")

> validationplot(pcr.fit,val.type="MSEP")

> pcr.pred=predict(pcr.fit,x[test,],ncomp=7)

> mean((pcr.pred-y.test)^2)

[1] 0.04461562

> pcr.fit=pcr(y~x,scale=TRUE,ncomp=7)

> summary(pcr.fit)

Data: X dimension: 50000 30

Y dimension: 50000 1

Fit method: svdpc

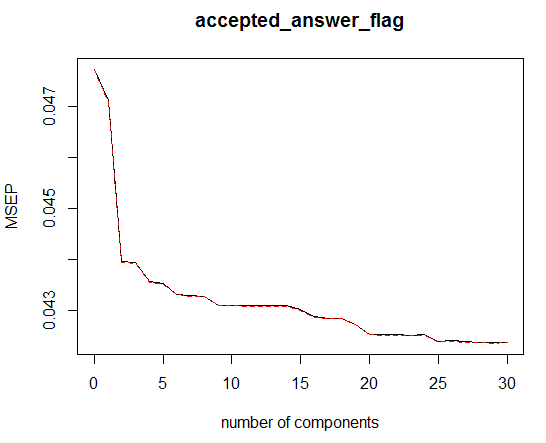
Number of components considered: 7

TRAINING: % variance explained

1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps

X 12.598 21.858 29.354 36.466 41.837 46.274 50.343

y 1.559 7.154 7.168 7.645 7.906 8.371 8.424



# Partial Least Squares

> set.seed(1)

> pls.fit=plsr(accepted\_answer\_flag~., data=Q2df,subset=train,scale=TRUE, validation="CV")

> summary(pls.fit)

Data: X dimension: 25000 30

Y dimension: 25000 1

Fit method: kernelpls

Number of components considered: 30

VALIDATION: RMSEP

Cross-validated using 10 random segments.

(Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps

CV 0.2184 0.2074 0.2064 0.2060 0.2059 0.2058 0.2058 0.2058 0.2058

adjCV 0.2184 0.2074 0.2064 0.2059 0.2058 0.2058 0.2058 0.2058 0.2058

9 comps 10 comps 11 comps 12 comps 13 comps 14 comps 15 comps 16 comps

CV 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058

adjCV 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058

17 comps 18 comps 19 comps 20 comps 21 comps 22 comps 23 comps 24 comps

CV 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058

adjCV 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058

25 comps 26 comps 27 comps 28 comps 29 comps 30 comps

CV 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058

adjCV 0.2058 0.2058 0.2058 0.2058 0.2058 0.2058

TRAINING: % variance explained

1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps

X 9.854 18.75 24.91 30.20 34.41 37.30 40.55

accepted\_answer\_flag 10.163 11.09 11.55 11.63 11.65 11.67 11.67

8 comps 9 comps 10 comps 11 comps 12 comps 13 comps 14 comps

X 43.82 47.56 53.38 56.48 58.64 61.41 63.68

accepted\_answer\_flag 11.67 11.67 11.67 11.67 11.67 11.67 11.67

15 comps 16 comps 17 comps 18 comps 19 comps 20 comps 21 comps

X 65.06 67.22 70.08 72.26 74.09 76.74 79.24

accepted\_answer\_flag 11.67 11.68 11.68 11.68 11.68 11.68 11.68

22 comps 23 comps 24 comps 25 comps 26 comps 27 comps 28 comps

X 81.68 84.27 86.75 89.25 90.28 92.30 94.51

accepted\_answer\_flag 11.68 11.68 11.68 11.68 11.68 11.68 11.68

29 comps 30 comps

X 97.31 100.00

accepted\_answer\_flag 11.68 11.68

> validationplot(pls.fit,val.type="MSEP")

> pls.pred=predict(pls.fit,x[test,],ncomp=2)

> mean((pls.pred-y.test)^2)

[1] 0.0442294

> pls.fit=plsr(accepted\_answer\_flag~., data=Q2df,scale=TRUE,ncomp=2)

> summary(pls.fit)

Data: X dimension: 50000 30

Y dimension: 50000 1

Fit method: kernelpls

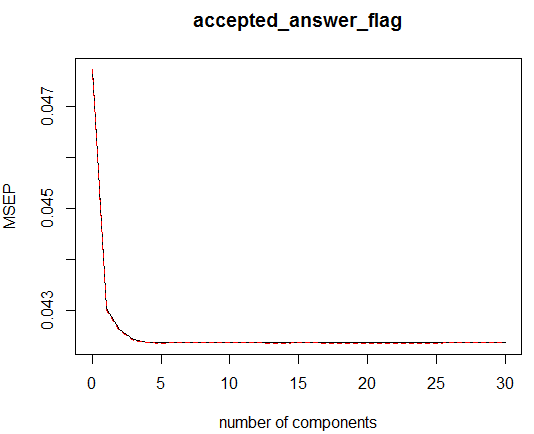
Number of components considered: 2

TRAINING: % variance explained

1 comps 2 comps

X 10.030 18.65

accepted\_answer\_flag 9.015 10.07



> Q2smp\_size=floor(0.75\*nrow(Q2df))

> set.seed(123)

> Q2train\_ind=sample(seq\_len(nrow(Q2df)), size = Q2smp\_size)

> Q2Train=Q2df[Q2train\_ind, ]

> Q2Test=Q2df[-Q2train\_ind, ]

> Q2lm=glm(accepted\_answer\_flag~.,data=Q2Train,family="binomial")

> summary(Q2lm)

Call:

glm(formula = accepted\_answer\_flag ~ ., family = "binomial",

data = Q2Train)

Deviance Residuals:

Min 1Q Median 3Q Max

-6.8859 -0.3633 -0.0969 -0.0216 8.4904

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.569e+00 2.290e-01 -11.221 < 2e-16 \*\*\*

a\_score 1.128e-01 1.793e-01 0.629 0.529072

a\_body\_length 1.865e-04 1.852e-05 10.072 < 2e-16 \*\*\*

a\_body\_has\_code 5.915e-01 1.187e-01 4.983 6.28e-07 \*\*\*

a\_DaysOld 2.838e-03 8.441e-04 3.363 0.000772 \*\*\*

a\_has\_edited 3.246e-01 5.662e-02 5.733 9.86e-09 \*\*\*

a\_num\_comment 1.297e-01 9.886e-03 13.121 < 2e-16 \*\*\*

a\_owner\_reputation 2.916e-06 1.097e-06 2.657 0.007893 \*\*

a\_owner\_profile\_summary 1.584e-01 2.553e-02 6.203 5.54e-10 \*\*\*

a\_owner\_views -8.620e-06 5.314e-06 -1.622 0.104750

a\_owner\_upvotes 9.821e-06 1.658e-05 0.592 0.553659

a\_owner\_downvotes 5.171e-05 1.434e-05 3.606 0.000310 \*\*\*

q\_score -7.681e-02 1.811e-01 -0.424 0.671538

q\_num\_views 1.442e-05 1.516e-06 9.509 < 2e-16 \*\*\*

q\_body\_length -1.360e-05 1.697e-05 -0.801 0.422901

q\_body\_has\_code 1.647e-01 1.118e-01 1.473 0.140852

q\_DaysOld -2.024e-03 1.271e-04 -15.928 < 2e-16 \*\*\*

q\_has\_edited -4.341e-02 5.741e-02 -0.756 0.449542

q\_title\_length -3.599e-03 1.441e-03 -2.497 0.012542 \*

q\_num\_tags -1.612e-02 2.239e-02 -0.720 0.471716

q\_num\_answers -1.468e-01 2.534e-02 -5.793 6.90e-09 \*\*\*

q\_num\_comment -2.460e-02 8.333e-03 -2.952 0.003158 \*\*

q\_owner\_reputation 3.273e-06 4.539e-06 0.721 0.470760

q\_owner\_profile\_summary 9.921e-02 2.583e-02 3.840 0.000123 \*\*\*

q\_owner\_views -2.253e-05 2.122e-05 -1.062 0.288282

q\_owner\_upvotes 2.104e-05 4.117e-05 0.511 0.609194

q\_owner\_downvotes 1.924e-04 7.034e-05 2.736 0.006226 \*\*

a\_votes\_up 1.881e-02 1.795e-01 0.105 0.916543

a\_votes\_down -1.381e-01 1.911e-01 -0.723 0.469918

q\_votes\_up 2.539e-02 1.813e-01 0.140 0.888621

q\_votes\_down -1.074e-01 1.820e-01 -0.590 0.555166

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 14622 on 37499 degrees of freedom

Residual deviance: 10624 on 37469 degrees of freedom

AIC: 10686

Number of Fisher Scoring iterations: 13

> Q2modelstepforward=step(Q2lm, direction="forward")

Start: AIC=10686.44

accepted\_answer\_flag ~ a\_score + a\_body\_length + a\_body\_has\_code +

a\_DaysOld + a\_has\_edited + a\_num\_comment + a\_owner\_reputation +

a\_owner\_profile\_summary + a\_owner\_views + a\_owner\_upvotes +

a\_owner\_downvotes + q\_score + q\_num\_views + q\_body\_length +

q\_body\_has\_code + q\_DaysOld + q\_has\_edited + q\_title\_length +

q\_num\_tags + q\_num\_answers + q\_num\_comment + q\_owner\_reputation +

q\_owner\_profile\_summary + q\_owner\_views + q\_owner\_upvotes +

q\_owner\_downvotes + a\_votes\_up + a\_votes\_down + q\_votes\_up +

q\_votes\_down

> summary(Q2modelstepforward)

Call:

glm(formula = accepted\_answer\_flag ~ a\_score + a\_body\_length +

a\_body\_has\_code + a\_DaysOld + a\_has\_edited + a\_num\_comment +

a\_owner\_reputation + a\_owner\_profile\_summary + a\_owner\_views +

a\_owner\_upvotes + a\_owner\_downvotes + q\_score + q\_num\_views +

q\_body\_length + q\_body\_has\_code + q\_DaysOld + q\_has\_edited +

q\_title\_length + q\_num\_tags + q\_num\_answers + q\_num\_comment +

q\_owner\_reputation + q\_owner\_profile\_summary + q\_owner\_views +

q\_owner\_upvotes + q\_owner\_downvotes + a\_votes\_up + a\_votes\_down +

q\_votes\_up + q\_votes\_down, family = "binomial", data = Q2Train)

Deviance Residuals:

Min 1Q Median 3Q Max

-6.8859 -0.3633 -0.0969 -0.0216 8.4904

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.569e+00 2.290e-01 -11.221 < 2e-16 \*\*\*

a\_score 1.128e-01 1.793e-01 0.629 0.529072

a\_body\_length 1.865e-04 1.852e-05 10.072 < 2e-16 \*\*\*

a\_body\_has\_code 5.915e-01 1.187e-01 4.983 6.28e-07 \*\*\*

a\_DaysOld 2.838e-03 8.441e-04 3.363 0.000772 \*\*\*

a\_has\_edited 3.246e-01 5.662e-02 5.733 9.86e-09 \*\*\*

a\_num\_comment 1.297e-01 9.886e-03 13.121 < 2e-16 \*\*\*

a\_owner\_reputation 2.916e-06 1.097e-06 2.657 0.007893 \*\*

a\_owner\_profile\_summary 1.584e-01 2.553e-02 6.203 5.54e-10 \*\*\*

a\_owner\_views -8.620e-06 5.314e-06 -1.622 0.104750

a\_owner\_upvotes 9.821e-06 1.658e-05 0.592 0.553659

a\_owner\_downvotes 5.171e-05 1.434e-05 3.606 0.000310 \*\*\*

q\_score -7.681e-02 1.811e-01 -0.424 0.671538

q\_num\_views 1.442e-05 1.516e-06 9.509 < 2e-16 \*\*\*

q\_body\_length -1.360e-05 1.697e-05 -0.801 0.422901

q\_body\_has\_code 1.647e-01 1.118e-01 1.473 0.140852

q\_DaysOld -2.024e-03 1.271e-04 -15.928 < 2e-16 \*\*\*

q\_has\_edited -4.341e-02 5.741e-02 -0.756 0.449542

q\_title\_length -3.599e-03 1.441e-03 -2.497 0.012542 \*

q\_num\_tags -1.612e-02 2.239e-02 -0.720 0.471716

q\_num\_answers -1.468e-01 2.534e-02 -5.793 6.90e-09 \*\*\*

q\_num\_comment -2.460e-02 8.333e-03 -2.952 0.003158 \*\*

q\_owner\_reputation 3.273e-06 4.539e-06 0.721 0.470760

q\_owner\_profile\_summary 9.921e-02 2.583e-02 3.840 0.000123 \*\*\*

q\_owner\_views -2.253e-05 2.122e-05 -1.062 0.288282

q\_owner\_upvotes 2.104e-05 4.117e-05 0.511 0.609194

q\_owner\_downvotes 1.924e-04 7.034e-05 2.736 0.006226 \*\*

a\_votes\_up 1.881e-02 1.795e-01 0.105 0.916543

a\_votes\_down -1.381e-01 1.911e-01 -0.723 0.469918

q\_votes\_up 2.539e-02 1.813e-01 0.140 0.888621

q\_votes\_down -1.074e-01 1.820e-01 -0.590 0.555166

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 14622 on 37499 degrees of freedom

Residual deviance: 10624 on 37469 degrees of freedom

AIC: 10686

Number of Fisher Scoring iterations: 13

> Q2modelstepbackward=step(Q2lm, direction="backward")

Start: AIC=10686.44

accepted\_answer\_flag ~ a\_score + a\_body\_length + a\_body\_has\_code +

a\_DaysOld + a\_has\_edited + a\_num\_comment + a\_owner\_reputation +

a\_owner\_profile\_summary + a\_owner\_views + a\_owner\_upvotes +

a\_owner\_downvotes + q\_score + q\_num\_views + q\_body\_length +

q\_body\_has\_code + q\_DaysOld + q\_has\_edited + q\_title\_length +

q\_num\_tags + q\_num\_answers + q\_num\_comment + q\_owner\_reputation +

q\_owner\_profile\_summary + q\_owner\_views + q\_owner\_upvotes +

q\_owner\_downvotes + a\_votes\_up + a\_votes\_down + q\_votes\_up +

q\_votes\_down

> summary(Q2modelstepbackward)

Call:

glm(formula = accepted\_answer\_flag ~ a\_score + a\_body\_length +

a\_body\_has\_code + a\_DaysOld + a\_has\_edited + a\_num\_comment +

a\_owner\_reputation + a\_owner\_profile\_summary + a\_owner\_views +

a\_owner\_downvotes + q\_score + q\_num\_views + q\_DaysOld + q\_title\_length +

q\_num\_answers + q\_num\_comment + q\_owner\_profile\_summary +

q\_owner\_downvotes + q\_votes\_down, family = "binomial", data = Q2Train)

Deviance Residuals:

Min 1Q Median 3Q Max

-6.9446 -0.3645 -0.0968 -0.0213 8.4904

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.528e+00 2.033e-01 -12.434 < 2e-16 \*\*\*

a\_score 1.339e-01 8.826e-03 15.177 < 2e-16 \*\*\*

a\_body\_length 1.830e-04 1.809e-05 10.121 < 2e-16 \*\*\*

a\_body\_has\_code 6.321e-01 1.168e-01 5.409 6.32e-08 \*\*\*

a\_DaysOld 2.797e-03 8.412e-04 3.326 0.000882 \*\*\*

a\_has\_edited 3.257e-01 5.646e-02 5.768 8.01e-09 \*\*\*

a\_num\_comment 1.262e-01 9.692e-03 13.025 < 2e-16 \*\*\*

a\_owner\_reputation 3.463e-06 9.162e-07 3.780 0.000157 \*\*\*

a\_owner\_profile\_summary 1.609e-01 2.541e-02 6.331 2.43e-10 \*\*\*

a\_owner\_views -1.029e-05 5.202e-06 -1.977 0.048004 \*

a\_owner\_downvotes 5.146e-05 1.433e-05 3.590 0.000331 \*\*\*

q\_score -5.340e-02 5.312e-03 -10.053 < 2e-16 \*\*\*

q\_num\_views 1.476e-05 1.510e-06 9.774 < 2e-16 \*\*\*

q\_DaysOld -2.029e-03 1.259e-04 -16.116 < 2e-16 \*\*\*

q\_title\_length -3.737e-03 1.435e-03 -2.604 0.009222 \*\*

q\_num\_answers -1.477e-01 2.533e-02 -5.829 5.59e-09 \*\*\*

q\_num\_comment -2.599e-02 8.202e-03 -3.169 0.001531 \*\*

q\_owner\_profile\_summary 1.047e-01 2.509e-02 4.170 3.04e-05 \*\*\*

q\_owner\_downvotes 1.631e-04 5.564e-05 2.930 0.003385 \*\*

q\_votes\_down -8.762e-02 2.317e-02 -3.781 0.000156 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 14622 on 37499 degrees of freedom

Residual deviance: 10631 on 37480 degrees of freedom

AIC: 10671

Number of Fisher Scoring iterations: 13

> Q2lm2=glm(accepted\_answer\_flag~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_num\_views+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary+q\_owner\_downvotes,data=Q2Train,family="binomial")

Warning message:

glm.fit: fitted probabilities numerically 0 or 1 occurred

> summary(Q2lm2)

Call:

glm(formula = accepted\_answer\_flag ~ a\_body\_length + a\_body\_has\_code +

a\_DaysOld + a\_has\_edited + a\_num\_comment + a\_owner\_reputation +

a\_owner\_profile\_summary + a\_owner\_downvotes + q\_num\_views +

q\_DaysOld + q\_title\_length + q\_num\_answers + q\_num\_comment +

q\_owner\_profile\_summary + q\_owner\_downvotes, family = "binomial",

data = Q2Train)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.4551 -0.3746 -0.1073 -0.0306 5.4927

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.481e+00 1.888e-01 -13.141 < 2e-16 \*\*\*

a\_body\_length 1.732e-04 1.749e-05 9.902 < 2e-16 \*\*\*

a\_body\_has\_code 6.442e-01 1.117e-01 5.765 8.18e-09 \*\*\*

a\_DaysOld 3.456e-03 8.267e-04 4.181 2.91e-05 \*\*\*

a\_has\_edited 3.853e-01 5.536e-02 6.959 3.42e-12 \*\*\*

a\_num\_comment 1.545e-01 9.180e-03 16.827 < 2e-16 \*\*\*

a\_owner\_reputation 2.060e-06 4.254e-07 4.843 1.28e-06 \*\*\*

a\_owner\_profile\_summary 1.666e-01 2.495e-02 6.677 2.45e-11 \*\*\*

a\_owner\_downvotes 6.383e-05 1.389e-05 4.596 4.30e-06 \*\*\*

q\_num\_views 1.481e-06 1.032e-06 1.435 0.15127

q\_DaysOld -2.383e-03 1.243e-04 -19.177 < 2e-16 \*\*\*

q\_title\_length -4.018e-03 1.398e-03 -2.874 0.00406 \*\*

q\_num\_answers -1.637e-01 2.190e-02 -7.478 7.56e-14 \*\*\*

q\_num\_comment -3.407e-02 7.472e-03 -4.560 5.11e-06 \*\*\*

q\_owner\_profile\_summary 9.931e-02 2.448e-02 4.056 4.99e-05 \*\*\*

q\_owner\_downvotes 1.196e-04 6.402e-05 1.869 0.06163 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 14622 on 37499 degrees of freedom

Residual deviance: 11024 on 37484 degrees of freedom

AIC: 11056

Number of Fisher Scoring iterations: 9

> Q2lm3=glm(accepted\_answer\_flag~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary,data=Q2Train,family="binomial")

Warning message:

glm.fit: fitted probabilities numerically 0 or 1 occurred

> summary(Q2lm3)

Call:

glm(formula = accepted\_answer\_flag ~ a\_body\_length + a\_body\_has\_code +

a\_DaysOld + a\_has\_edited + a\_num\_comment + a\_owner\_reputation +

a\_owner\_profile\_summary + a\_owner\_downvotes + q\_DaysOld +

q\_title\_length + q\_num\_answers + q\_num\_comment + q\_owner\_profile\_summary,

family = "binomial", data = Q2Train)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.4521 -0.3746 -0.1072 -0.0306 5.7524

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.498e+00 1.880e-01 -13.287 < 2e-16 \*\*\*

a\_body\_length 1.729e-04 1.746e-05 9.898 < 2e-16 \*\*\*

a\_body\_has\_code 6.414e-01 1.116e-01 5.748 9.04e-09 \*\*\*

a\_DaysOld 3.435e-03 8.260e-04 4.158 3.21e-05 \*\*\*

a\_has\_edited 3.850e-01 5.537e-02 6.954 3.56e-12 \*\*\*

a\_num\_comment 1.549e-01 9.177e-03 16.881 < 2e-16 \*\*\*

a\_owner\_reputation 2.074e-06 4.250e-07 4.881 1.06e-06 \*\*\*

a\_owner\_profile\_summary 1.659e-01 2.495e-02 6.650 2.93e-11 \*\*\*

a\_owner\_downvotes 6.426e-05 1.389e-05 4.625 3.74e-06 \*\*\*

q\_DaysOld -2.338e-03 1.194e-04 -19.579 < 2e-16 \*\*\*

q\_title\_length -4.034e-03 1.398e-03 -2.885 0.00391 \*\*

q\_num\_answers -1.604e-01 2.169e-02 -7.395 1.41e-13 \*\*\*

q\_num\_comment -3.358e-02 7.469e-03 -4.496 6.92e-06 \*\*\*

q\_owner\_profile\_summary 1.030e-01 2.438e-02 4.224 2.40e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 14622 on 37499 degrees of freedom

Residual deviance: 11028 on 37486 degrees of freedom

AIC: 11056

Number of Fisher Scoring iterations: 9

> BIC(Q2lm3)

[1] 11175.39

> logistic\_probs=predict(Q2lm3, Q2Train, type="response")

> head(logistic\_probs)

14379 39415 20449 44149 47020 2278

0.0598964805 0.0492997030 0.0269305387 0.0020040065 0.0001021921 0.0362776801

> testing\_y=Q2Test$accepted\_answer\_flag

> logistic\_pred\_y=rep(0,length(testing\_y))

> logistic\_pred\_y[logistic\_probs>0.5]=1

> training\_y=Q2Train$accepted\_answer\_flag

> table(logistic\_pred\_y,training\_y)

training\_y

logistic\_pred\_y 0 1

0 11861 592

1 84 77

> mean(logistic\_pred\_y!=training\_y,na.rm=TRUE)

[1] 0.05359125

> logistic\_probs=predict(Q2lm3, Q2Test, type="response")

> head(logistic\_probs)

2 8 10 15 20 23

4.139617e-02 2.182282e-01 4.832195e-05 1.992405e-03 2.790292e-02 1.882245e-03

> logistic\_pred\_y=rep(0,length(testing\_y))

> logistic\_pred\_y[logistic\_probs>0.5]=1

> logistic\_probs=predict(Q2lm3, Q2Test, type="response")

> head(logistic\_probs)

2 8 10 15 20 23

4.139617e-02 2.182282e-01 4.832195e-05 1.992405e-03 2.790292e-02 1.882245e-03

> logistic\_pred\_y=rep(0,length(testing\_y))

> logistic\_pred\_y[logistic\_probs>0.5]=1

> table(logistic\_pred\_y,testing\_y)

testing\_y

logistic\_pred\_y 0 1

0 11818 633

1 29 20

> mean(logistic\_pred\_y!=testing\_y,na.rm=TRUE)

[1] 0.05296

#Cross Validation for logistic regression

#LOOCV

> library(boot)

> MSE\_LOOCV=cv.glm(Q2Train, Q2lm3)$delta[1]

> MSE\_LOOCV

> MSE\_LOOCV=NULL

> for(i in 1:10){

+ model=glm(accepted\_answer\_flag~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary,data=Q2Train)

+ MSE\_LOOCV[i]=cv.glm(Q2Train,model)$delta[1]

+ }

> MSE\_LOOCV

#K-fold

> MSE\_10\_Fold\_CV=cv.glm(Q2Train,Q2lm3,K=10)$delta[1]

> MSE\_10\_Fold\_CV

[1] 0.04083843

> MSE\_10\_Fold\_CV=NULL

> for(i in 1:10){

+ model=glm(accepted\_answer\_flag~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary,data=Q2Train)

+ MSE\_10\_Fold\_CV[i]=cv.glm(Q2Train,model,K=10)$delta[1]

+ }

> MSE\_10\_Fold\_CV

[1] 0.04165307 0.04164606 0.04165980 0.04166714 0.04165900 0.04167902 0.04167137

[8] 0.04165470 0.04167492 0.04165797

> install.packages("ROCR")

> library(ROCR)

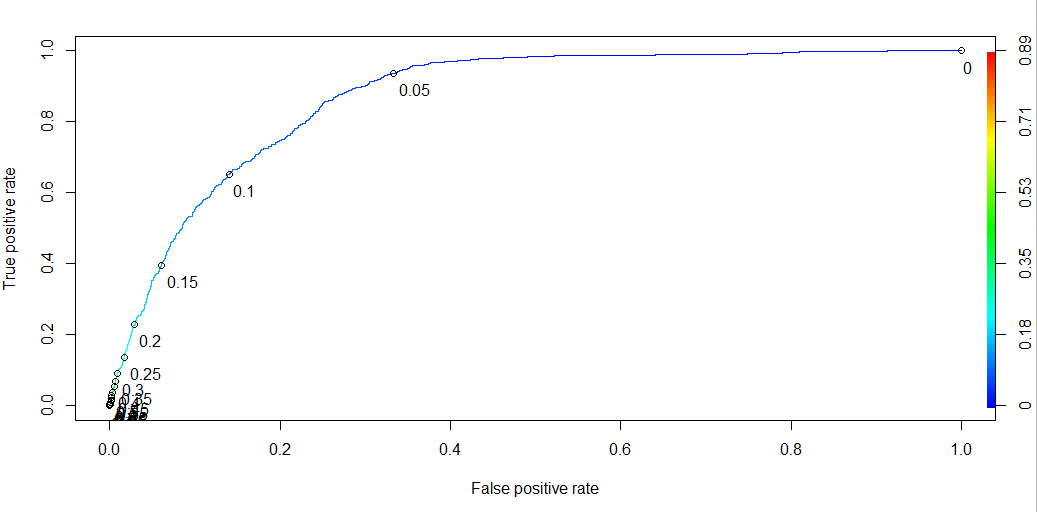
> ROCRpred<- prediction(logistic\_probs,testing\_y)

> ROCRperf<- performance(ROCRpred,"tpr","fpr")

> plot(ROCRperf)

> plot(ROCRperf,colorize=TRUE)

> plot(ROCRperf, colorize = TRUE, print.cutoffs.at = seq(0,1,0.05), text.adj = c(-0.2,1.7))



> as.numeric(performance(ROCRpred,"auc")@y.values)

[1] 0.871199

**LDA:**

> install.packages("lda")

> library(lda)

> library(MASS)

> lda.model = lda(accepted\_answer\_flag~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary, data=Q2Train, family = binomial)

> lda.model

Call:

lda(accepted\_answer\_flag ~ a\_body\_length + a\_body\_has\_code +

a\_DaysOld + a\_has\_edited + a\_num\_comment + a\_owner\_reputation +

a\_owner\_profile\_summary + a\_owner\_downvotes + q\_DaysOld +

q\_title\_length + q\_num\_answers + q\_num\_comment + q\_owner\_profile\_summary,

data = Q2Train, family = binomial)

Prior probabilities of groups:

0 1

0.9512 0.0488

Group means:

a\_body\_length a\_body\_has\_code a\_DaysOld a\_has\_edited a\_num\_comment a\_owner\_reputation a\_owner\_profile\_summary

0 802.0828 0.8086347 55.91189 0.2624054 0.522596 6191.805 1.291197

1 1318.1306 0.9480874 57.57869 0.4928962 2.477049 27575.838 1.772131

a\_owner\_downvotes q\_DaysOld q\_title\_length q\_num\_answers q\_num\_comment q\_owner\_profile\_summary

0 116.3752 1046.6772 51.40401 8.013429 2.833081 1.0808803

1 650.8727 106.6191 50.71803 5.646448 3.372131 0.8387978

Coefficients of linear discriminants:

LD1

a\_body\_length 1.801253e-04

a\_body\_has\_code 1.827253e-01

a\_DaysOld 6.394856e-04

a\_has\_edited 2.537278e-01

a\_num\_comment 3.527535e-01

a\_owner\_reputation 5.219942e-06

a\_owner\_profile\_summary 1.059400e-01

a\_owner\_downvotes 1.763384e-04

q\_DaysOld -4.921785e-04

q\_title\_length -2.951783e-03

q\_num\_answers -6.274203e-03

q\_num\_comment -2.784911e-02

q\_owner\_profile\_summary 4.180869e-02

#predicting the LDA model with training data

> lda\_pred=predict(lda.model,Q2Train)

> lda\_class=lda\_pred$class

#CONFUSION MATRIX with LDA Training data

> table(lda\_class,Q2Train$accepted\_answer\_flag)

lda\_class 0 1

0 34970 1525

1 700 305

#Misclassification Rate with LDA training data

> mean(lda\_class!=Q2Train$accepted\_answer\_flag)

[1] 0.05933333

#predicting the LDA model with test data

> lda\_prediction=predict(lda.model,Q2Test)

> lda\_class\_n=lda\_prediction$class

#CONFUSION MATRIX with LDA Testing data

> table(lda\_class\_n,Q2Test$accepted\_answer\_flag)

lda\_class\_n 0 1

0 11616 565

1 231 88

#Misclassification Rate with LDA testing data

> mean(lda\_class\_n!=Q2Test$accepted\_answer\_flag)

[1] 0.06368

#CROSS VALIDATION of LDA Model

> library(rpart)

> library(ipred)

> ip.lda <- function(object, newdata) predict(object, newdata = newdata)$class

> errorest(factor(Q2Train$accepted\_answer\_flag)~ Q2Train$a\_body\_length+Q2Train$a\_body\_has\_code+Q2Train$a\_DaysOld+Q2Train$a\_has\_edited+Q2Train$a\_num\_comment+Q2Train$a\_owner\_reputation+Q2Train$a\_owner\_profile\_summary+Q2Train$a\_owner\_downvotes+Q2Train$q\_DaysOld+Q2Train$q\_title\_length+Q2Train$q\_num\_answers+Q2Train$q\_num\_comment+Q2Train$q\_owner\_profile\_summary, data=Q2Train, model=lda, estimator="cv",est.para=control.errorest(k=10), predict=ip.lda)$err

[1] 0.07325333

#ROC for LDA

> S = lda\_prediction$posterior[,2]

> roc.curve=function(s,print=FALSE){

+ Ps=(S>s)\*1

+ FP=sum((Ps==1)\*(Q2Test$accepted\_answer\_flag == 0))/sum(Q2Test$accepted\_answer\_flag == 0)

+ TP=sum((Ps==1)\*(Q2Test$accepted\_answer\_flag == 1))/sum(Q2Test$accepted\_answer\_flag == 1)

+ if(print==TRUE){

+ print(table(Observed=Q2Test$accepted\_answer\_flag,Predicted=Ps))

+ }

+ vect=c(FP,TP)

+ names(vect)=c("FPR","TPR")

+ return(vect)

+ }

> threshold = 0.5

> roc.curve(threshold,print=TRUE)

Predicted

Observed 0 1

0 11616 231

1 565 88

FPR TPR

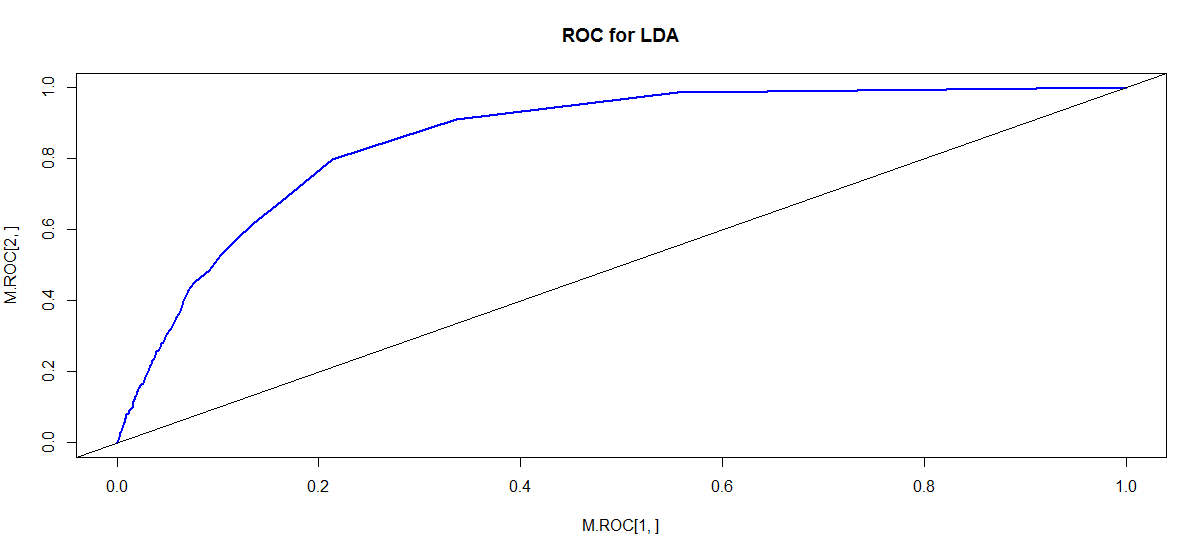
0.01949861 0.13476263

> ROC.curve=Vectorize(roc.curve)

> M.ROC=ROC.curve(seq(0,1,by=.01))

> plot(M.ROC[1,],M.ROC[2,],col="blue",lwd=2,type="l", main = "ROC for LDA")

> abline(0,1)



**QDA**

> qda\_model = qda(accepted\_answer\_flag~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary, data=Q2Train, family=binomial)

> qda\_model

Call:

qda(accepted\_answer\_flag ~ a\_body\_length + a\_body\_has\_code +

a\_DaysOld + a\_has\_edited + a\_num\_comment + a\_owner\_reputation +

a\_owner\_profile\_summary + a\_owner\_downvotes + q\_DaysOld +

q\_title\_length + q\_num\_answers + q\_num\_comment + q\_owner\_profile\_summary,

data = Q2Train, family = binomial)

Prior probabilities of groups:

0 1

0.9512 0.0488

Group means:

a\_body\_length a\_body\_has\_code a\_DaysOld a\_has\_edited a\_num\_comment a\_owner\_reputation

0 802.0828 0.8086347 55.91189 0.2624054 0.522596 6191.805

1 1318.1306 0.9480874 57.57869 0.4928962 2.477049 27575.838

a\_owner\_profile\_summary a\_owner\_downvotes q\_DaysOld q\_title\_length q\_num\_answers

0 1.291197 116.3752 1046.6772 51.40401 8.013429

1 1.772131 650.8727 106.6191 50.71803 5.646448

q\_num\_comment q\_owner\_profile\_summary

0 2.833081 1.0808803

1 3.372131 0.8387978

#predicting the model with training data

> qda\_pred=predict(qda\_model, Q2Train)

> qda\_class=qda\_pred$class

#CONFUSION MATRIX for QDA training data

> table(qda\_class,Q2Train$accepted\_answer\_flag)

qda\_class 0 1

0 34021 1301

1 1649 529

#Misclassification Rate for QDA training data

> mean(qda\_class!=Q2Train$accepted\_answer\_flag)

[1] 0.07866667

#predicting the model with test data for QDA

> qda\_pred1=predict(qda\_model,Q2Test)

> qda\_class\_n=qda\_pred1$class

#CONFUSION MATRIX for QDA test data

> table(qda\_class\_n,Q2Test$accepted\_answer\_flag)

qda\_class\_n 0 1

0 11279 478

1 568 175

#Misclassification Rate for QDA test data

> mean(qda\_class\_n!=Q2Test$accepted\_answer\_flag)

[1] 0.08368

#CROSS VALIDATION for QDA

> ip.qda <- function(object, newdata) predict(object, newdata = newdata)$class

> errorest(factor(Q2Train$accepted\_answer\_flag)~ Q2Train$a\_body\_length+Q2Train$a\_body\_has\_code+Q2Train$a\_DaysOld+Q2Train$a\_has\_edited+Q2Train$a\_num\_comment+Q2Train$a\_owner\_reputation+Q2Train$a\_owner\_profile\_summary+Q2Train$a\_owner\_downvotes+Q2Train$q\_DaysOld+Q2Train$q\_title\_length+Q2Train$q\_num\_answers+Q2Train$q\_num\_comment+Q2Train$q\_owner\_profile\_summary, data=Q2Train, model=qda, estimator="cv",est.para=control.errorest(k=10), predict=ip.qda)$err

[1] 0.10176

#ROC for QDA

> qda.S = qda\_pred1$posterior[,2]

> roc.curve=function(s,print=FALSE){

+ Ps=(qda.S>s)\*1

+ FP=sum((Ps==1)\*(Q2Test$accepted\_answer\_flag == 0))/sum(Q2Test$accepted\_answer\_flag == 0)

+ TP=sum((Ps==1)\*(Q2Test$accepted\_answer\_flag == 1))/sum(Q2Test$accepted\_answer\_flag == 1)

+ if(print==TRUE){

+ print(table(Observed= Q2Test$accepted\_answer\_flag, Predicted=Ps))

+ }

+ vect=c(FP,TP)

+ names(vect)=c("FPR","TPR")

+ return(vect)

+ }

> threshold = 0.5

> roc.curve(threshold,print=TRUE)

Predicted

Observed 0 1

0 11279 568

1 478 175

FPR TPR

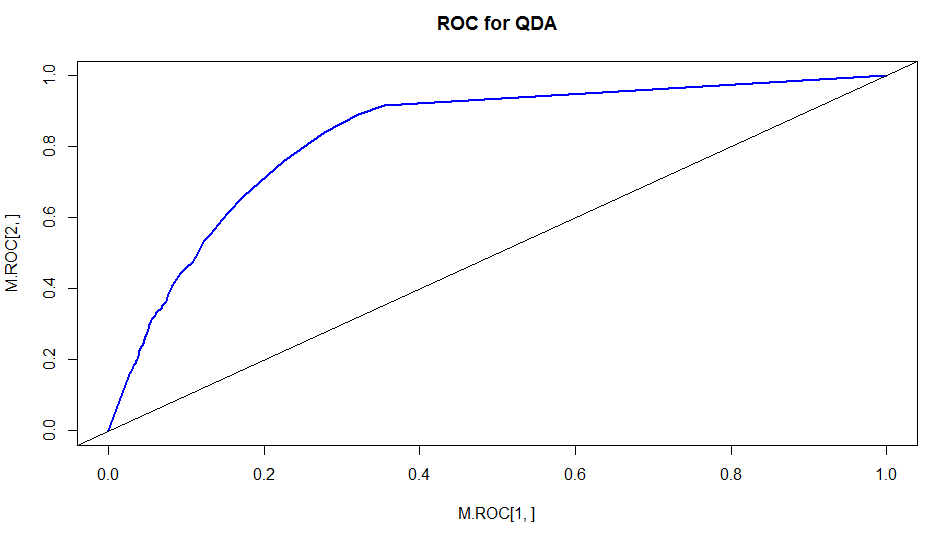
0.04794463 0.26799387

> ROC.curve=Vectorize(roc.curve)

> M.ROC=ROC.curve(seq(0,1,by=.01))

> plot(M.ROC[1,],M.ROC[2,],col="blue",lwd=2,type="l", main = "ROC for QDA")

> abline(0,1)



**KNN:**

> library(class)

> train.x<-cbind(Q2Train$a\_body\_length+Q2Train$a\_body\_has\_code+Q2Train$a\_DaysOld+Q2Train$a\_has\_edited+Q2Train$a\_num\_comment+Q2Train$a\_owner\_reputation+Q2Train$a\_owner\_profile\_summary+Q2Train$a\_owner\_downvotes+Q2Train$q\_DaysOld+Q2Train$q\_title\_length+Q2Train$q\_num\_answers+Q2Train$q\_num\_comment+Q2Train$q\_owner\_profile\_summary)

> test.x<- cbind(Q2Test$a\_body\_length+Q2Test$a\_body\_has\_code+Q2Test$a\_DaysOld+Q2Test$a\_has\_edited+Q2Test$a\_num\_comment+Q2Test$a\_owner\_reputation+Q2Test$a\_owner\_profile\_summary+Q2Test$a\_owner\_downvotes+Q2Test$q\_DaysOld+Q2Test$q\_title\_length+Q2Test$q\_num\_answers+Q2Test$q\_num\_comment+Q2Test$q\_owner\_profile\_summary)

> set.seed(123)

#predicting the model with testing data for KNN with k=1

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=1,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11558 600

1 289 53

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.07112

#Trying to predict KNN with different k values to get the lowest misclassification rate

#predicting the KNN model with k=2

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=2,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11604 610

1 243 43

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.06824

#predicting the KNN model with k=3

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=3,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11754 630

1 93 23

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05784

#predicting the KNN model with k=4

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=4,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11761 632

1 86 21

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05744

#predicting the KNN model with k=5

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=5,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11797 636

1 50 17

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05488

#predicting the KNN model with k=6

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=6,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11805 636

1 42 17

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05424

#predicting the KNN model with k=7

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=7,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11827 646

1 20 7

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05328

#predicting the KNN model with k=8

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=8,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11829 643

1 18 10

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05288

#predicting the KNN model with k=9

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=9,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11841 646

1 6 7

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05216

#predicting the KNN model with k=10

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=10,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11844 648

1 3 5

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05208

#predicting the KNN model with k=11

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=11,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11846 651

1 1 2

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05216

#predicting the KNN model with k=12

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=12,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11842 651

1 5 2

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05248

#predicting the KNN model with k=13

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=13,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11845 653

1 2 0

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.0524

#predicting the KNN model with k=14

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=14,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11843 653

1 4 0

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05256

#predicting the KNN model with k=15

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=15,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11845 652

1 2 1

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05232

#predicting the KNN model with k=16

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=16,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11844 652

1 3 1

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.0524

#predicting the KNN model with k=17

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=17,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11845 652

1 2 1

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05232

#predicting the KNN model with k=18

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=18,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11845 652

1 2 1

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05232

#predicting the KNN model with k=19

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=19,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11847 652

1 0 1

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05216

#predicting the KNN model with k=20

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=20,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11846 652

1 1 1

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05224

#predicting the KNN model with k=21

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=21,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11847 653

1 0 0

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05224

#predicting the KNN model with k=22

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=22,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11847 653

1 0 0

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05224

#predicting the KNN model with k=30

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=30,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11847 653

1 0 0

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05224

#predicting the KNN model with k=40

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=40,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11847 653

1 0 0

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05224

#predicting the KNN model with k=100

> knn.pred<- knn(data.frame(train.x),data.frame(test.x),Q2Train$accepted\_answer\_flag,k=100,prob=TRUE)

> table(knn.pred,Q2Test$accepted\_answer\_flag)

knn.pred 0 1

0 11847 653

1 0 0

> mean(knn.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05224

#taking the model built using knn=8

#CROSS VALIDATION of KNN

> bwpredict.knn <- function(object, newdata) predict.ipredknn(object, newdata, type="class")

> errorest(factor(Q2Train$accepted\_answer\_flag) ~ Q2Train$a\_body\_length+Q2Train$a\_body\_has\_code+Q2Train$a\_DaysOld+Q2Train$a\_has\_edited+Q2Train$a\_num\_comment+Q2Train$a\_owner\_reputation+Q2Train$a\_owner\_profile\_summary+Q2Train$a\_owner\_downvotes+Q2Train$q\_DaysOld+Q2Train$q\_title\_length+Q2Train$q\_num\_answers+Q2Train$q\_num\_comment+Q2Train$q\_owner\_profile\_summary, data=Q2Train, model=ipredknn, estimator="cv", est.para=control.errorest(k=10), predict=bwpredict.knn, kk=10)$err

[1] 0.0584

#ROC for KNN

> library(ROCR)

> prob<- attr(knn.pred,"prob")

> prob

> prob <- (2\*(ifelse(knn.pred == "0", 1-prob, prob)) - 1)

> knn <- prediction(prob, testing\_y)

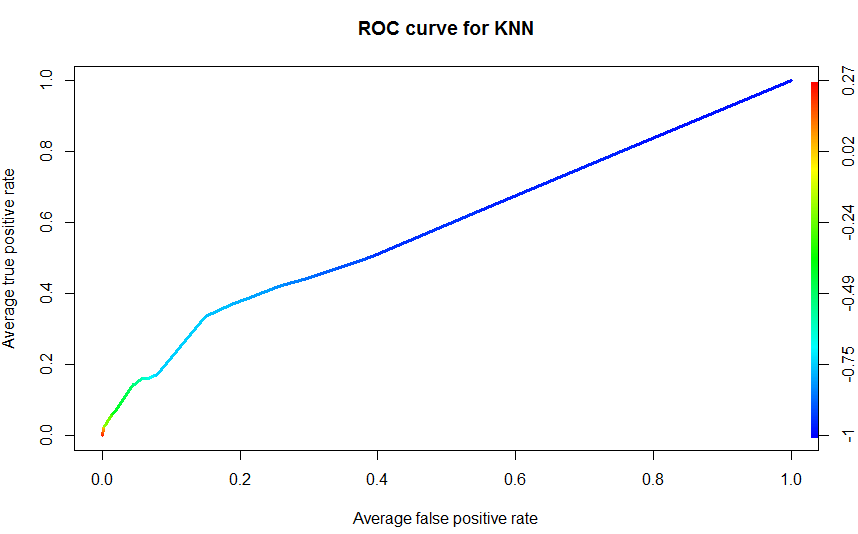
> pred\_knn <- prediction(prob, testing\_y)

> pred\_knn <- performance(pred\_knn, "tpr", "fpr")

> plot(pred\_knn, avg= "threshold", colorize=T, lwd=3, main="ROC curve for KNN")

> as.numeric(performance(ROCRpred,"auc")@y.values)

[1] 0.871199



**CART**:

> install.packages("tree")

> library(tree)

> tree.CMPTrain = tree(accepted\_answer\_flag~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary,Q2Train)

> summary(tree.CMPTrain)

Regression tree:

tree(formula = accepted\_answer\_flag ~ a\_body\_length + a\_body\_has\_code +

a\_DaysOld + a\_has\_edited + a\_num\_comment + a\_owner\_reputation +

a\_owner\_profile\_summary + a\_owner\_downvotes + q\_DaysOld +

q\_title\_length + q\_num\_answers + q\_num\_comment + q\_owner\_profile\_summary,

data = Q2Train)

Variables actually used in tree construction:

[1] "q\_DaysOld" "a\_num\_comment" "a\_owner\_reputation"

Number of terminal nodes: 4

Residual mean deviance: 0.04134 = 1550 / 37500

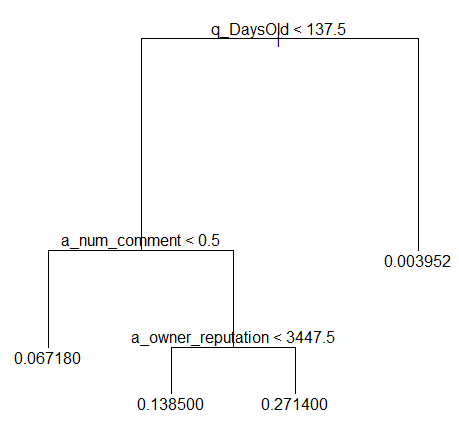
Distribution of residuals:

Min. 1st Qu. Median Mean 3rd Qu. Max.

-0.271400 -0.067180 -0.003952 0.000000 -0.003952 0.996000

> plot(tree.CMPTrain)

> text(tree.CMPTrain,pretty=0)



> lf=seq(1,nrow(Q2Train))

> tree.Q2Train=tree(as.factor(accepted\_answer\_flag)~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary,subset=lf)

> tree.pred=predict(tree.Q2Train,Q2Test,type="class")

> summary(tree.Q2Train)

Classification tree:

tree(formula = as.factor(accepted\_answer\_flag) ~ a\_body\_length +

a\_body\_has\_code + a\_DaysOld + a\_has\_edited + a\_num\_comment +

a\_owner\_reputation + a\_owner\_profile\_summary + a\_owner\_downvotes +

q\_DaysOld + q\_title\_length + q\_num\_answers + q\_num\_comment +

q\_owner\_profile\_summary, subset = lf)

Variables actually used in tree construction:

[1] "q\_DaysOld" "a\_num\_comment" "a\_owner\_reputation"

Number of terminal nodes: 4

Residual mean deviance: 0.2834 = 10630 / 37500

Misclassification error rate: 0.04656 = 1746 / 37500

> plot(tree.Q2Train)

> text(tree.Q2Train,pretty=0)

#Confusion matrix for CART

> table(tree.pred,testing\_y)

testing\_y

tree.pred 0 1

0 11847 653

1 0 0

> mean(tree.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05224

#Pruning and Cross Validation of the Classification Tree

> cv.Q2Train=cv.tree(tree.Q2Train,FUN=prune.misclass)

> names(cv.Q2Train)

[1] "size" "dev" "k" "method"

> cv.Q2Train

$size

[1] 4 1

$dev

[1] 1746 1746

$k

[1] -Inf 0

$method

[1] "misclass"

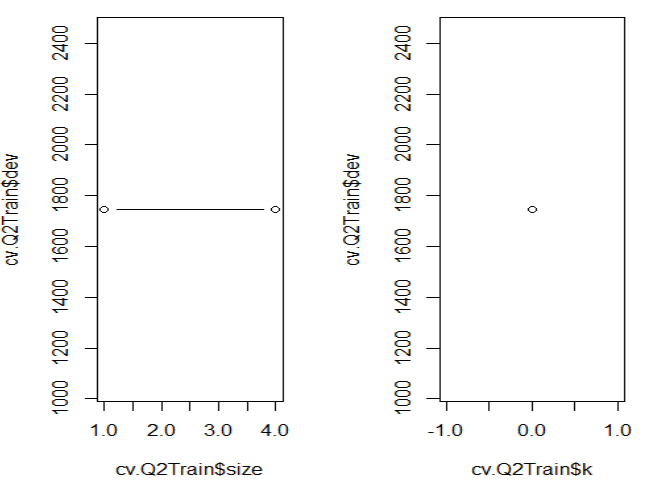
attr(,"class")

[1] "prune" "tree.sequence"

> par(mfrow=c(1,2))

> plot(cv.Q2Train$size, cv.Q2Train$dev, type="b")

> plot(cv.Q2Train$k, cv.Q2Train$dev, type="b")



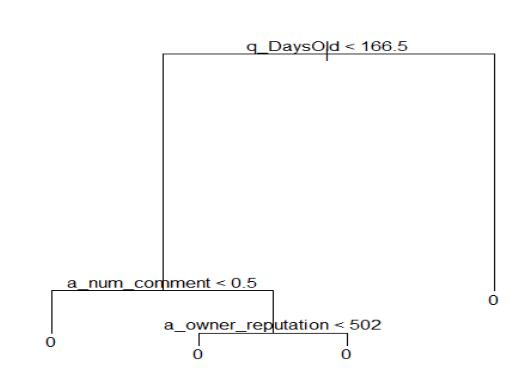
> par(mfrow=c(1,1))

> prune.Q2Train=prune.misclass(tree.Q2Train, best=9)

Warning message:

In prune.tree(tree = tree.Q2Train, best = 9, method = "misclass") :

best is bigger than tree size



> tree.pred=predict(prune.Q2Train, Q2Test, type="class")

> table(tree.pred, testing\_y)

testing\_y

tree.pred 0 1

0 11847 653

1 0 0

> mean(tree.pred!=Q2Test$accepted\_answer\_flag)

[1] 0.05224

**Random Forest**

> library(randomForest)

> bag.Q2Train

> bag.Q2Train=randomForest(accepted\_answer\_flag~a\_body\_length+a\_body\_has\_code+a\_DaysOld+a\_has\_edited+a\_num\_comment+a\_owner\_reputation+a\_owner\_profile\_summary+a\_owner\_downvotes+q\_DaysOld+q\_title\_length+q\_num\_answers+q\_num\_comment+q\_owner\_profile\_summary,data=Q2Train, subset=lf, mtry=4, importance=TRUE)

> bag.Q2Train

Call:

randomForest(formula = accepted\_answer\_flag ~ a\_body\_length + a\_body\_has\_code + a\_DaysOld + a\_has\_edited + a\_num\_comment + a\_owner\_reputation + a\_owner\_profile\_summary + a\_owner\_downvotes + q\_DaysOld + q\_title\_length + q\_num\_answers + q\_num\_comment + q\_owner\_profile\_summary, data = Q2Train, mtry = 4, importance = TRUE, subset = lf)

Type of random forest: regression

Number of trees: 500

No. of variables tried at each split: 4

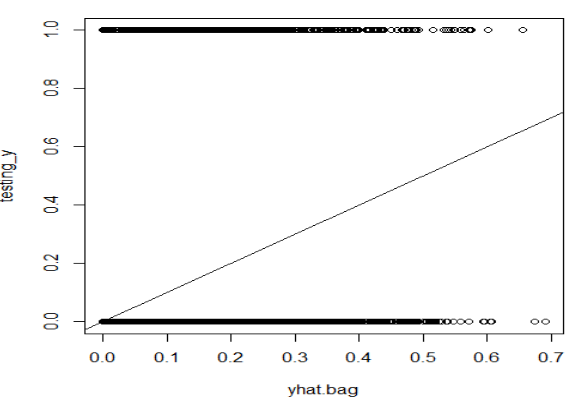
Mean of squared residuals: 0.04126036

% Var explained: 11.11

> yhat.bag=predict(bag.Q2Train, Q2Test)

> plot(yhat.bag, testing\_y)

> abline(0,1)



> mean(yhat.bag!=testing\_y)

[1] 0.8875

#ROC for CART

> tree.pred=predict(tree.Q2Train,Q2Test,type="vector",prob=TRUE)

> tree.S = tree.pred[,2]

> roc.curve=function(s,print=FALSE){

+ Ps=(tree.S>s)\*1

+ FP=sum((Ps==1)\*(Q2Train$accepted\_answer\_flag == 0))/sum(Q2Train$accepted\_answer\_flag == 0)

+ TP=sum((Ps==1)\*(Q2Train$accepted\_answer\_flag == 1))/sum(Q2Train$accepted\_answer\_flag == 1)

+ if(print==TRUE){

+ print(table(Observed=Q2Test$accepted\_answer\_flag,Predicted=Ps))

+ }

+ vect=c(FP,TP)

+ names(vect)=c("FPR","TPR")

+ return(vect)

+ }

> threshold = 0.5

> roc.curve(threshold,print=TRUE)

Predicted

Observed 0

0 11847

1 653

FPR TPR

0 0

> ROC.curve=Vectorize(roc.curve)

> M.ROC=ROC.curve(seq(0,1,by=.01))

> plot(M.ROC[1,],M.ROC[2,],col="blue",lwd=2,type="l", main = "ROC for CART")

> abline(0,1)

