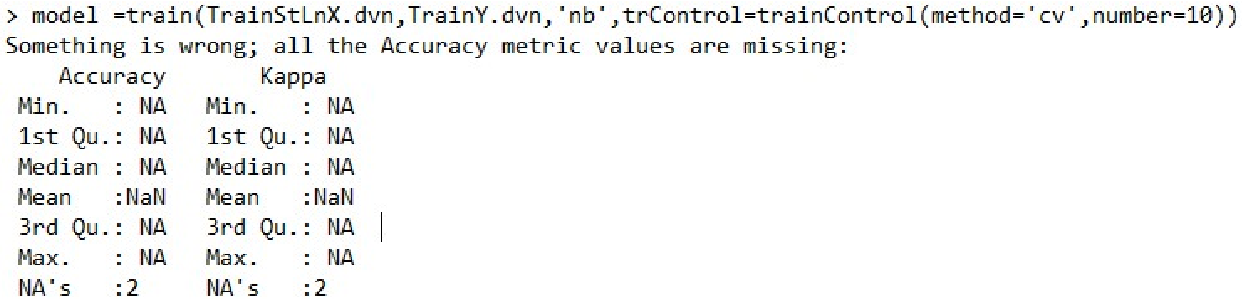
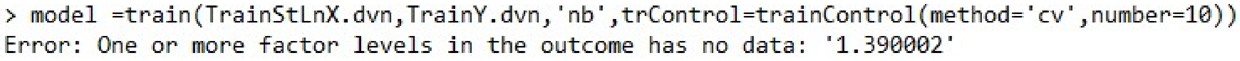
Baruch College, STA-CIS 3920, Exercise # 6
7/31/2020 William Hall 20

6.1 During this exercise I ran into a huge amount of errors with the model= variable.



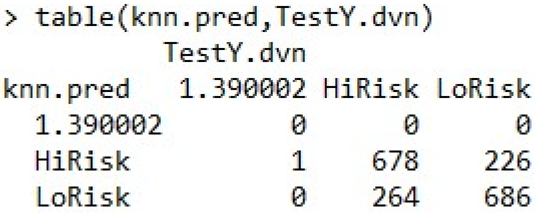
```
> model =train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',number=10))
Something is wrong; all the Accuracy metric values are missing:
    Accuracy       Kappa    
 Min.   : NA   Min.   : NA  
 1st Qu.: NA   1st Qu.: NA  
 Median : NA   Median : NA  
 Mean   :NaN   Mean   :NaN  
 3rd Qu.: NA   3rd Qu.: NA  
 Max.   : NA   Max.   : NA  
 NA's   :2     NA's   :2    
```

After working with it, I got this following error:



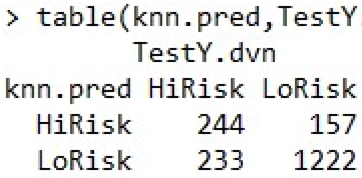
```
> model =train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',number=10))
Error: One or more factor levels in the outcome has no data: '1.390002'
```

Finally, after trying over and over to redo the variables for LN4 and LN6 I finally figured out a way to get it to work. I worked my way up from LN4 and made sure that the table for knn.pred was working. I got a huge amount of errors in the step in LN5 such as:



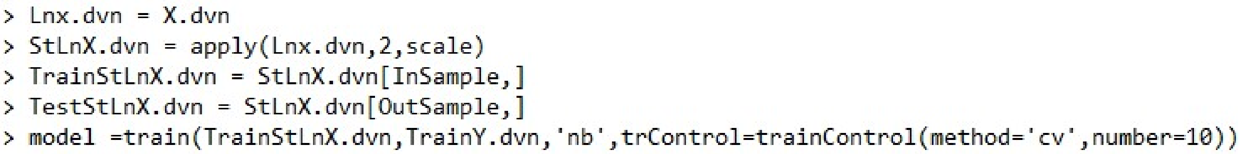
```
> table(knn.pred,TestY.dvn)
          TestY.dvn
knn.pred   1.390002 HiRisk LoRisk
  1.390002        0      0      0
  HiRisk          1    678    226
  LoRisk          0    264    686
```

I made the correct modifications to Y.dvn (which was set to the wrong column) and generated this output:



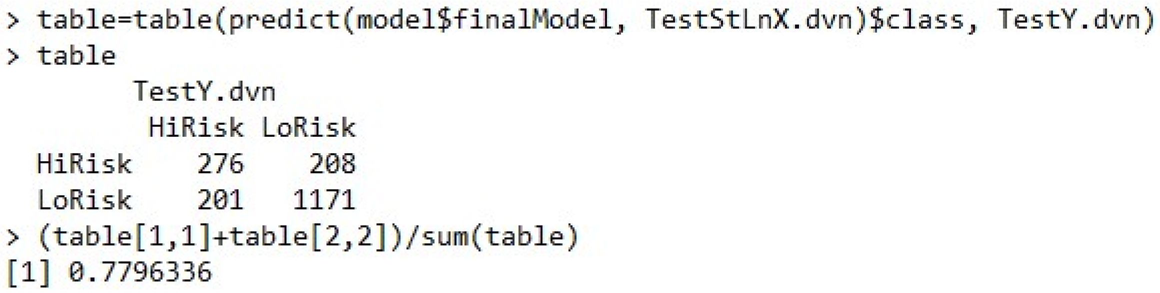
```
> table(knn.pred,TestY.dvn)
        TestY.dvn
knn.pred HiRisk LoRisk
  HiRisk    244    157
  LoRisk    233   1222
```

Immediately after, I tried to get model= to work and was successful this time around.



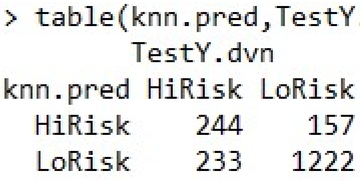
```
> Lnx.dvn = X.dvn
> StLnX.dvn = apply(Lnx.dvn,2,scale)
> TrainStLnX.dvn = StLnX.dvn[InSample,]
> TestStLnX.dvn = StLnX.dvn[OutSample,]
> model =train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',number=10))
```

After getting modell to work, I tried the table to create the table variable and it finally worked. I ended up with a 78% accurate prediction.

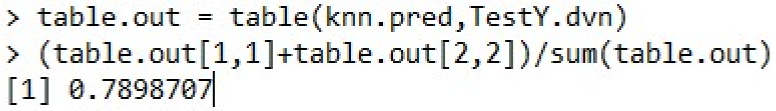


```
> table=table(predict(model$finalModel, TestStLnX.dvn)$class, TestY.dvn)
> table
        TestY.dvn
         HiRisk LoRisk
  HiRisk    276    208
  LoRisk    201   1171
> (table[1,1]+table[2,2])/sum(table)
[1] 0.7796336
```

When using knn, the prediction accuracy fared slightly better than the NB model with a 79%



```
> table(knn.pred,TestY.dvn)
        TestY.dvn
knn.pred HiRisk LoRisk
  HiRisk    244    157
  LoRisk    233   1222
```

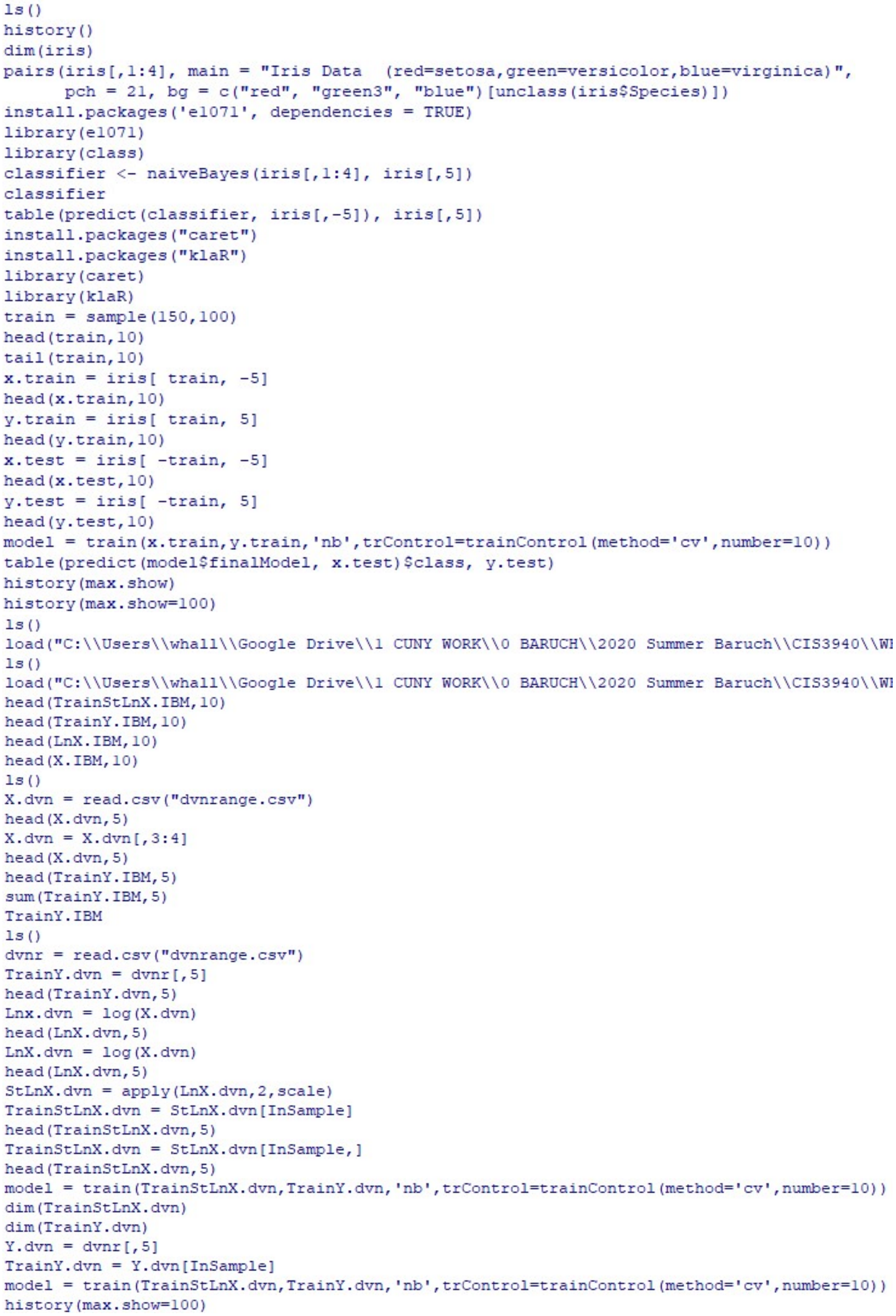


```
> table.out = table(knn.pred,TestY.dvn)
> (table.out[1,1]+table.out[2,2])/sum(table.out)
[1] 0.7898707
```

I not sure as why this happened, I thought the results of the NB would be significantly better but these are the results of my test.

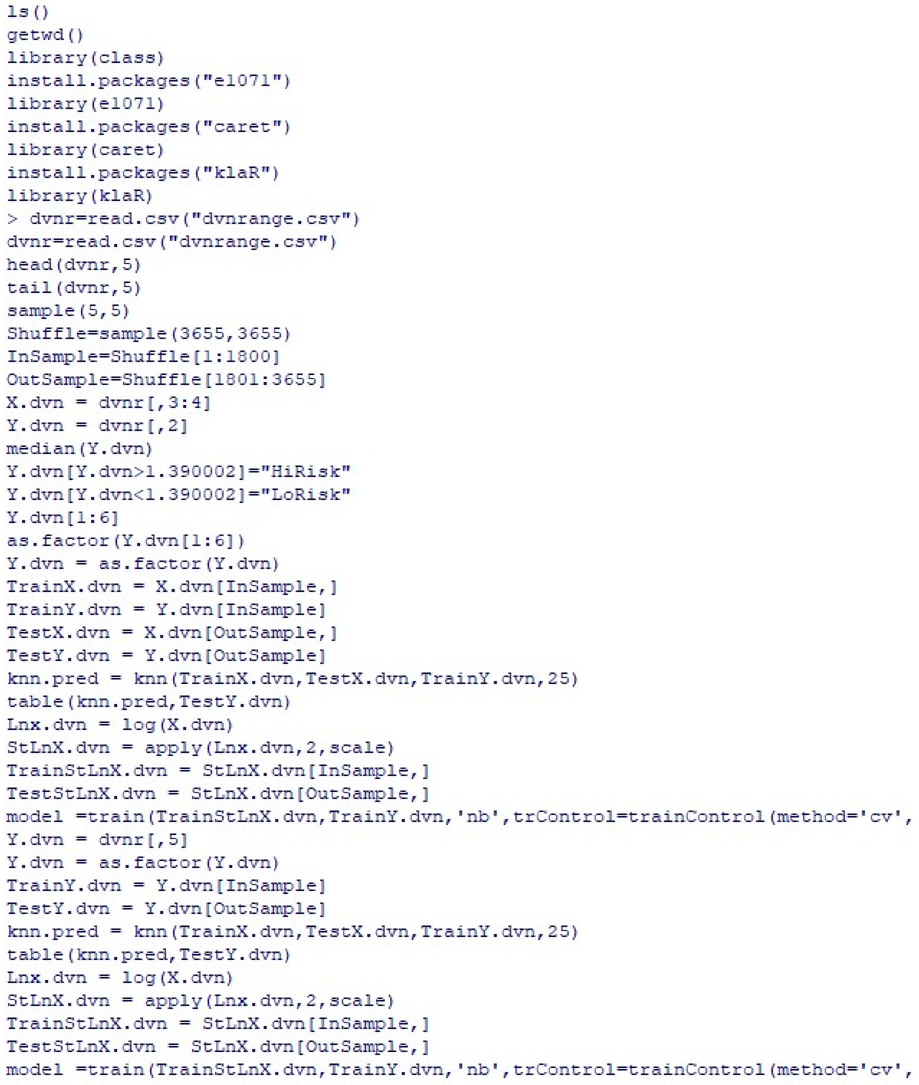
APPENDIX

Baruch College, STA-CIS 3920, Exercise # 6
7/31/2020 William Hall 20

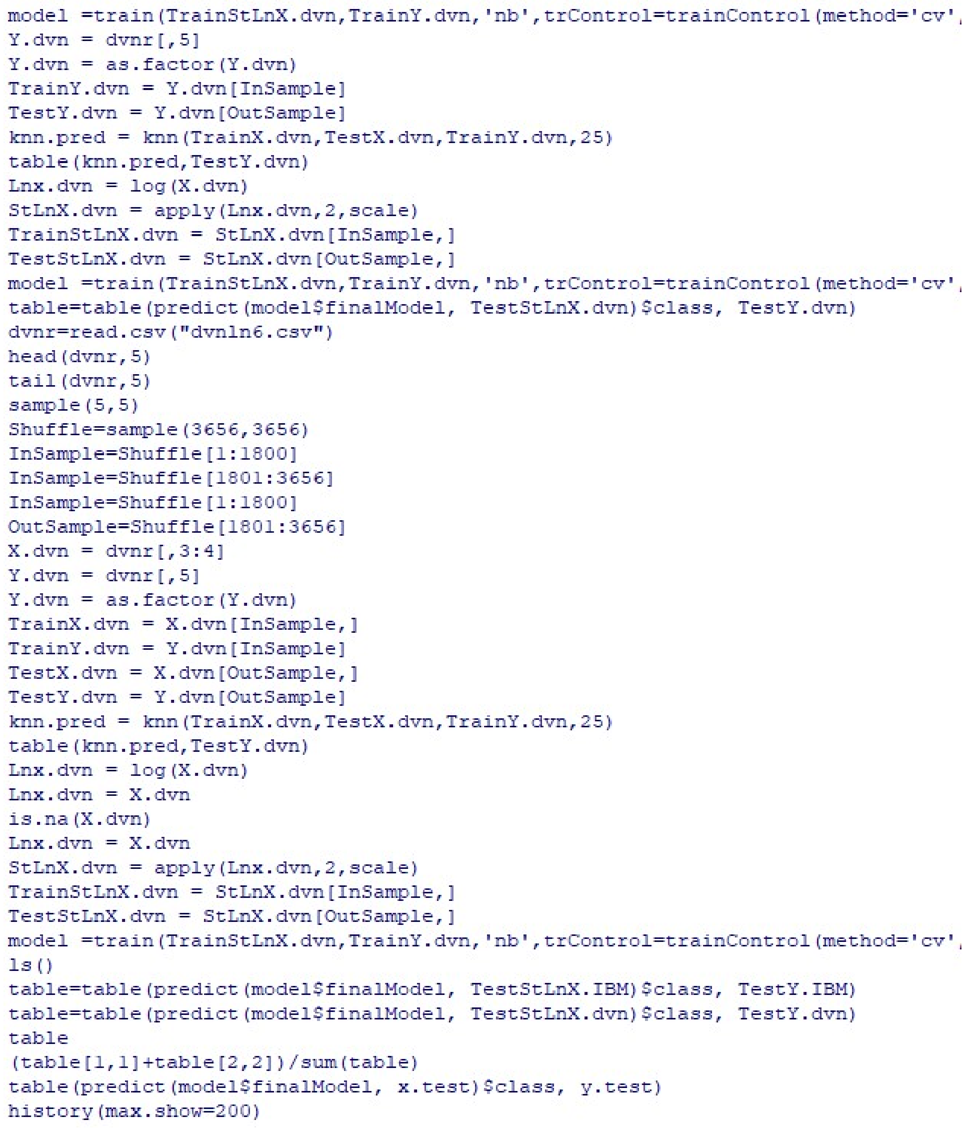


```
ls()
history()
dim(iris)
pairs(iris[,1:4], main = "Iris Data  (red=setosa,green=versicolor,blue=virginica)",
      pch = 21, bg = c("red", "green3", "blue")[unclass(iris$Species)])
install.packages('e1071', dependencies = TRUE)
library(e1071)
library(class)
classifier <- naiveBayes(iris[,1:4], iris[,5])
classifier
table(predict(classifier, iris[,-5]), iris[,5])
install.packages("caret")
install.packages("klaR")
library(caret)
library(klaR)
train = sample(150,100)
head(train,10)
tail(train,10)
x.train = iris[ train, -5]
head(x.train,10)
y.train = iris[ train, 5]
head(y.train,10)
x.test = iris[ -train, -5]
head(x.test,10)
y.test = iris[ -train, 5]
head(y.test,10)
model = train(x.train,y.train,'nb',trControl=trainControl(method='cv',number=10))
table(predict(model$finalModel, x.test)$class, y.test)
history(max.show)
history(max.show=100)
ls()
load("C:\\Users\\whall\\Google Drive\\1 CUNY WORK\\0 BARUCH\\2020 Summer Baruch\\CIS3940\\W
ls()
load("C:\\Users\\whall\\Google Drive\\1 CUNY WORK\\0 BARUCH\\2020 Summer Baruch\\CIS3940\\W
head(TrainStLnX.IBM,10)
head(TrainY.IBM,10)
head(LnX.IBM,10)
head(X.IBM,10)
ls()
X.dvn = read.csv("dvnrange.csv")
head(X.dvn,5)
X.dvn = X.dvn[,3:4]
head(X.dvn,5)
head(TrainY.IBM,5)
sum(TrainY.IBM,5)
TrainY.IBM
ls()
dvnr = read.csv("dvnrange.csv")
TrainY.dvn = dvnr[,5]
head(TrainY.dvn,5)
Lnx.dvn = log(X.dvn)
head(LnX.dvn,5)
LnX.dvn = log(X.dvn)
head(LnX.dvn,5)
StLnX.dvn = apply(LnX.dvn,2,scale)
TrainStLnX.dvn = StLnX.dvn[InSample]
head(TrainStLnX.dvn,5)
TrainStLnX.dvn = StLnX.dvn[InSample,]
head(TrainStLnX.dvn,5)
model = train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',number=10))
dim(TrainStLnX.dvn)
dim(TrainY.dvn)
Y.dvn = dvnr[,5]
TrainY.dvn = Y.dvn[InSample]
model = train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',number=10))
history(max.show=100)
```

Baruch College, STA-CIS 3920, Exercise # 6
7/31/2020 William Hall 20



```
ls()
getwd()
library(class)
install.packages("e1071")
library(e1071)
install.packages("caret")
library(caret)
install.packages("klaR")
library(klaR)
> dvnr=read.csv("dvnrange.csv")
dvnr=read.csv("dvnrange.csv")
head(dvnr,5)
tail(dvnr,5)
sample(5,5)
Shuffle=sample(3655,3655)
InSample=Shuffle[1:1800]
OutSample=Shuffle[1801:3655]
X.dvn = dvnr[,3:4]
Y.dvn = dvnr[,2]
median(Y.dvn)
Y.dvn[Y.dvn>1.390002]="HiRisk"
Y.dvn[Y.dvn<1.390002]="LoRisk"
Y.dvn[1:6]
as.factor(Y.dvn[1:6])
Y.dvn = as.factor(Y.dvn)
TrainX.dvn = X.dvn[InSample,]
TrainY.dvn = Y.dvn[InSample]
TestX.dvn = X.dvn[OutSample,]
TestY.dvn = Y.dvn[OutSample]
knn.pred = knn(TrainX.dvn,TestX.dvn,TrainY.dvn,25)
table(knn.pred,TestY.dvn)
Lnx.dvn = log(X.dvn)
StLnX.dvn = apply(Lnx.dvn,2,scale)
TrainStLnX.dvn = StLnX.dvn[InSample,]
TestStLnX.dvn = StLnX.dvn[OutSample,]
model =train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',
Y.dvn = dvnr[,5]
Y.dvn = as.factor(Y.dvn)
TrainY.dvn = Y.dvn[InSample]
TestY.dvn = Y.dvn[OutSample]
knn.pred = knn(TrainX.dvn,TestX.dvn,TrainY.dvn,25)
table(knn.pred,TestY.dvn)
Lnx.dvn = log(X.dvn)
StLnX.dvn = apply(Lnx.dvn,2,scale)
TrainStLnX.dvn = StLnX.dvn[InSample,]
TestStLnX.dvn = StLnX.dvn[OutSample,]
model =train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',
```



```
model =train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',
Y.dvn = dvnr[,5]
Y.dvn = as.factor(Y.dvn)
TrainY.dvn = Y.dvn[InSample]
TestY.dvn = Y.dvn[OutSample]
knn.pred = knn(TrainX.dvn,TestX.dvn,TrainY.dvn,25)
table(knn.pred,TestY.dvn)
Lnx.dvn = log(X.dvn)
StLnX.dvn = apply(Lnx.dvn,2,scale)
TrainStLnX.dvn = StLnX.dvn[InSample,]
TestStLnX.dvn = StLnX.dvn[OutSample,]
model =train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',
table=table(predict(model$finalModel, TestStLnX.dvn)$class, TestY.dvn)
dvnr=read.csv("dvnln6.csv")
head(dvnr,5)
tail(dvnr,5)
sample(5,5)
Shuffle=sample(3656,3656)
InSample=Shuffle[1:1800]
InSample=Shuffle[1801:3656]
InSample=Shuffle[1:1800]
OutSample=Shuffle[1801:3656]
X.dvn = dvnr[,3:4]
Y.dvn = dvnr[,5]
Y.dvn = as.factor(Y.dvn)
TrainX.dvn = X.dvn[InSample,]
TrainY.dvn = Y.dvn[InSample]
TestX.dvn = X.dvn[OutSample,]
TestY.dvn = Y.dvn[OutSample]
knn.pred = knn(TrainX.dvn,TestX.dvn,TrainY.dvn,25)
table(knn.pred,TestY.dvn)
Lnx.dvn = log(X.dvn)
Lnx.dvn = X.dvn
is.na(X.dvn)
Lnx.dvn = X.dvn
StLnX.dvn = apply(Lnx.dvn,2,scale)
TrainStLnX.dvn = StLnX.dvn[InSample,]
TestStLnX.dvn = StLnX.dvn[OutSample,]
model =train(TrainStLnX.dvn,TrainY.dvn,'nb',trControl=trainControl(method='cv',
ls()
table=table(predict(model$finalModel, TestStLnX.IBM)$class, TestY.IBM)
table=table(predict(model$finalModel, TestStLnX.dvn)$class, TestY.dvn)
table
(table[1,1]+table[2,2])/sum(table)
table(predict(model$finalModel, x.test)$class, y.test)
history(max.show=200)
```