Big Data Econometrics · ECG - 590

HOMEWORK-2

2. $Pr(Y=K \mid X) = P_K(X)$

Eqn. 4.12

$$P_K(X) = \frac{\pi_K \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_K)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

Denominator is just sum of all classes and will be same for all classes so, we are just concerned with numerator.

$$f_x = \pi_K \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_K)^2\right)$$

Taking log on both side we would have.

$$\cancel{Lf(x)}\ Lf_x = \ln \pi_K + \ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + \left(-\frac{1}{2\sigma^2}(x-\mu_K)^2\right)$$

$\ln \frac{1}{\sqrt{2\pi}\sigma}$ is constant and will be same for all classes.

so, we can just drop it

$$Lf_x = \ln \pi_K - \frac{1}{2\sigma^2}(x^2 + \mu_K^2 - 2x\mu_K)$$

$$Lf_x = \ln \pi_K - \frac{x^2}{2\sigma^2} - \frac{\mu_K^2}{2\sigma^2} + x\frac{\mu_K}{\sigma^2}$$

As we have to maximize the fnctn for particular class.
so, we can drop $-\frac{x^2}{2\sigma^2}$ as it will be same for all classes. so, we are left with.

$$\boxed{\delta_K(x) = x\frac{\mu_K}{\sigma^2} - \frac{\mu_K^2}{2\sigma^2} + \ln \pi_K.}$$ which is eqn. 4.13

we would be assigning ~~its~~ observation for which eqn. 4.13 would be largest for the class.

title: " ECG 590 HW-2“”

6.Suppose we collect data for a group of students in a statistics class with variables X1 =hours studied, X2 =undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, ^ ??0 = ??? 6, ^??1 = 0.05, ^??2 = 1.

a. Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.
b. How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class?

A.

```
prob=function(x1,x2){ logi=exp(-6 + 0.05*x1 + 1*x2); p=logi/(1+logi);return(p)}
prob(40,3.5)
```

```
## [1] 0.3775407
```

B. We have approx 38% probability of getting A in the class.so, let’s see probability for different hours.

```
hours=seq(40,60,1)
probs=mapply(hours, 3.5, FUN=prob)
names(probs)=paste0(hours,"h")
probs
```

```
##       40h       41h       42h       43h       44h       45h       46h
## 0.3775407 0.3893608 0.4013123 0.4133824 0.4255575 0.4378235 0.4501660
##       47h       48h       49h       50h       51h       52h       53h
## 0.4625702 0.4750208 0.4875026 0.5000000 0.5124974 0.5249792 0.5374298
##       54h       55h       56h       57h       58h       59h       60h
## 0.5498340 0.5621765 0.5744425 0.5866176 0.5986877 0.6106392 0.6224593
```

We can see that to have 50% chance, one need to study 50 hours.

7. Suppose that we wish to predict whether a given stock will issue a dividend this year (“Yes” or “No”) based on X, last year’s percent profit.We examine a large number of companies and discover that the mean value of X for companies that issued a dividend was ¯X = 10, while the mean for those that didn’t was ¯X = 0. In addition, the variance of X for these two sets of companies was ^??2 = 36. Finally, 80% of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was X = 4 last year.

Since, X follows a normal distribution. We can use Baye’s theorem with Normal Distribution Function.

```
pdf_normal = function(x, mu_k, sigma){
  (sqrt(2*pi)*sigma)^-1*exp(-(2*sigma^2)^-1*(x-mu_k))
  }

sigma <- 6 # both classes

# class 1, companies that issued a dividend
pi_1= 0.8
mu_1=10

# class2, companies that didn't issue a dividend
pi_2= 0.2
mu_2 = 0

# computing probabilities
x = 4
p_1 = (pi_1*pdf_normal(4,mu_1,sigma))/(pi_1*pdf_normal(4,mu_1,sigma) + pi_2*pdf_normal(4,mu_2,si
gma))
p_2= (pi_2*pdf_normal(4,mu_2,sigma))/(pi_1*pdf_normal(4,mu_1,sigma) + pi_2*pdf_normal(4,mu_2,sig
ma))

# rounding the numbers
p_1 = round(p_1,2)
p_2 = round(p_2,2)

# plot
cbind(c("Dividend", "Non-Dividend"), c(p_1, p_2))
```

```
##      [,1]           [,2]
## [1,] "Dividend"     "0.82"
## [2,] "Non-Dividend" "0.18"
```

So, there is 82% probability that company will issue dividend this year.

10. This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1, 089 weekly returns for 21 years, from the beginning of 1990 to the end of
11.

a. Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?
b. Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?
c. Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.
d. Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).
e. Repeat (d) using LDA.
f. Repeat (d) using QDA.
g. Repeat (d) using KNN with K = 1.

h. Which of these methods appears to provide the best results on this data?
i. Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

Let's first get all the libraries required to do this question

```
library(class)    # for KNN
library(ISLR)     # for data
```

```
## Warning: package 'ISLR' was built under R version 3.5.3
```

```
library(MASS)     # for LDA
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages ---------------------------------------------------------------------------
-------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.3.0
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'tibble' was built under R version 3.5.3
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

```
## Warning: package 'readr' was built under R version 3.5.3
```

```
## Warning: package 'purrr' was built under R version 3.5.2
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
## Warning: package 'forcats' was built under R version 3.5.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------
-- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.5.3
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
head(Weekly)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
```

A.

```
print("summary")
```

```
## [1] "summary"
```

```
summary(Weekly)
```

```
##       Year           Lag1              Lag2              Lag3        
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
##       Lag4               Lag5              Volume       
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747  
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202  
##  Median :  0.2380   Median :  0.2340   Median :1.00268  
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462  
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373  
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821  
##      Today          Direction 
##  Min.   :-18.1950   Down:484  
##  1st Qu.: -1.1540   Up  :605  
##  Median :  0.2410             
##  Mean   :  0.1499             
##  3rd Qu.:  1.4050             
##  Max.   : 12.0260             
```

```
print("coorelation")
```

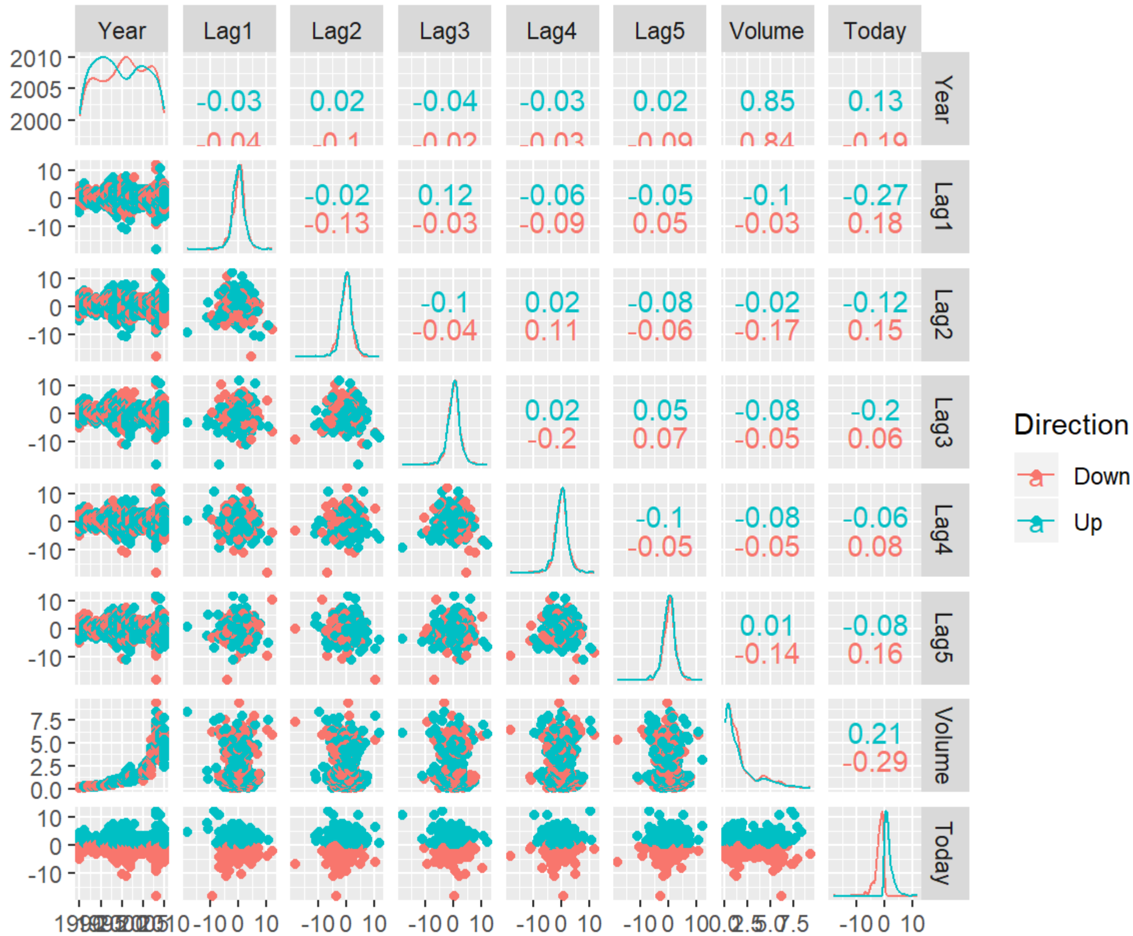
```
## [1] "coorelation"
```

```
cor(Weekly[ ,-9])
```

```
##               Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```

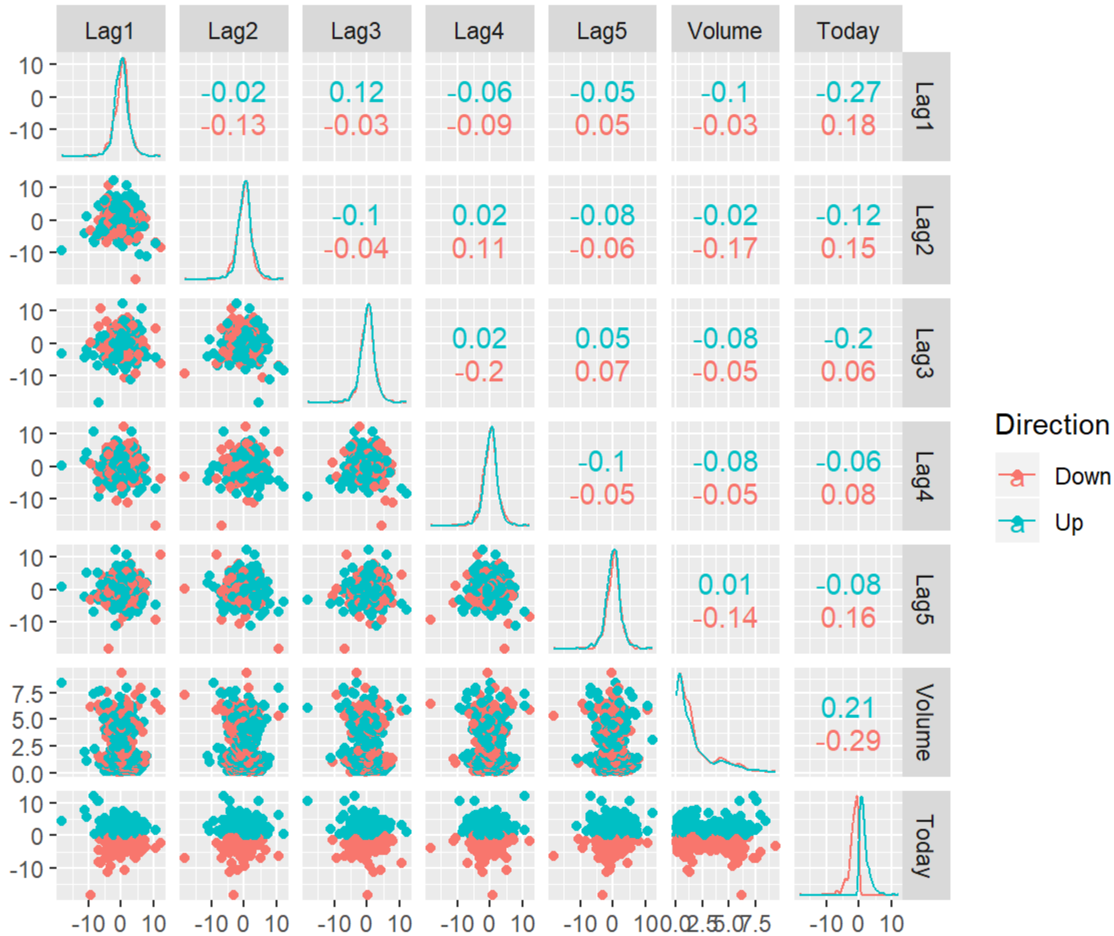
```
ggscatmat(Weekly, color = "Direction")
```

```
## Warning in ggscatmat(Weekly, color = "Direction"): Factor variables are
## omitted in plot
```



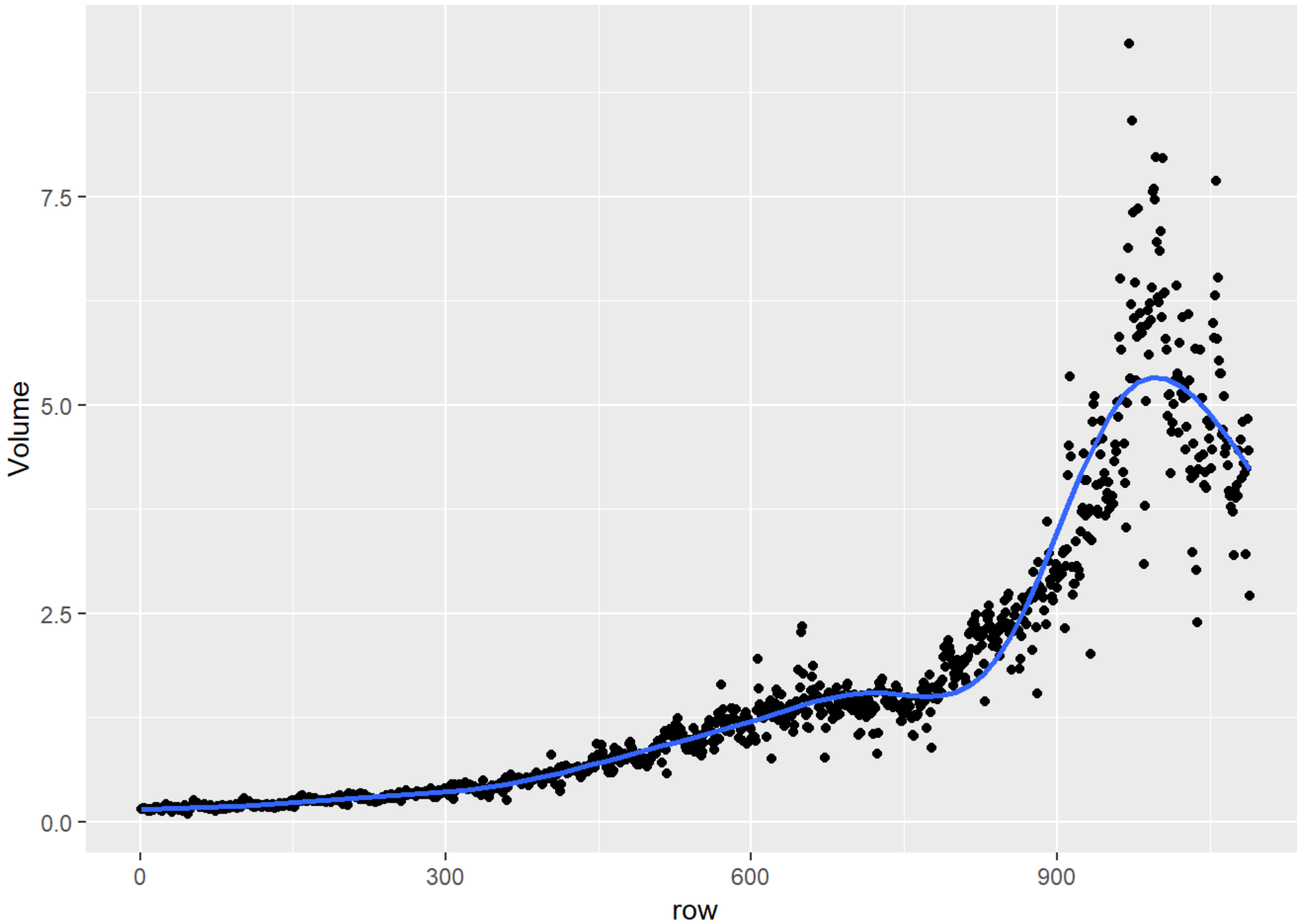

```
ggscatmat(Weekly, columns = 2:9, color = "Direction")
```

```
## Warning in ggscatmat(Weekly, columns = 2:9, color = "Direction"): Factor
## variables are omitted in plot
```




```
Weekly %>% mutate(row = row_number()) %>%
  ggplot(aes(x = row, y = Volume)) +
  geom_point() +
  geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



B.Fitting Logistic Regression Model

```
glm_fit_wk <- glm(Direction ~
                    Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                  data = Weekly,
                  family = binomial)
summary(glm_fit_wk)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Based on p-value, lag 2 with p-value of 0.0296 seems to be significant among all the 6 predictors along with the intercept

C.

```
glm_probs_wk = predict(glm_fit_wk, type = "response")
glm_pred_wk = rep("Down", length(glm_probs_wk))
glm_pred_wk[glm_probs_wk > 0.5] <- "Up"

table(glm_pred_wk, Weekly$Direction)
```

```
##
## glm_pred_wk Down  Up
##        Down   54  48
##        Up    430 557
```

```
mean(glm_pred_wk == Weekly$Direction)
```

```
## [1] 0.5610652
```

On an average 56% times, logistic regression model is predicting the response direction correctly. 557 out of total 605 times of UP, Logistic regression is predicting UP, which is very good but out of 484 times of down, logistic regression is predicting 54 times down only. It seems Logistic Regression is biased towards UP direction.

D.Let's create a training and test data set as follows:

```
train <- (Weekly$Year < 2009)
Weekly_train <- Weekly[train,]
Weekly_test <- Weekly[!train,]
Direction_train <- Weekly_train$Direction
Direction_test <- Weekly_test$Direction
```

Now's let's create a logistic model on Train data sets from 1990 to 2008:

```
logistic_wkly <- glm(Direction ~ Lag2,
                     data = Weekly_train,
                     family = binomial)
summary(logistic_wkly)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

Now let's test the model on test data

```
logistic_probs <- predict(logistic_wkly, Weekly_test, type = "response")
logistic_pred = rep("Down", length(Direction_test))
logistic_pred[logistic_probs > 0.5] <- "Up"
table(logistic_pred, Direction_test)
```

```
##               Direction_test
## logistic_pred Down Up
##          Down    9  5
##          Up     34 56
```

```
mean(logistic_pred == Direction_test)
```

```
## [1] 0.625
```

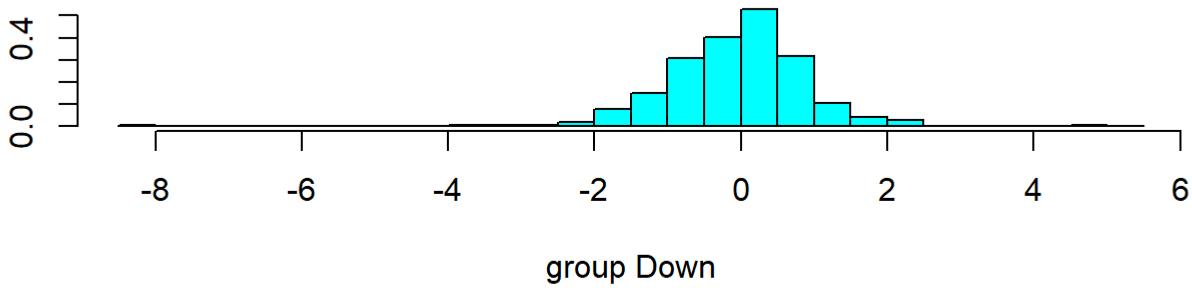
We can see now 62.5% times the logistic regression model with only lag2 as predictor is predicting directions correctly which is more than previous 56%. Out of 61 UPs, it correctly predicted 56 times and out of 43 Downs , it predicted 9 times correctly

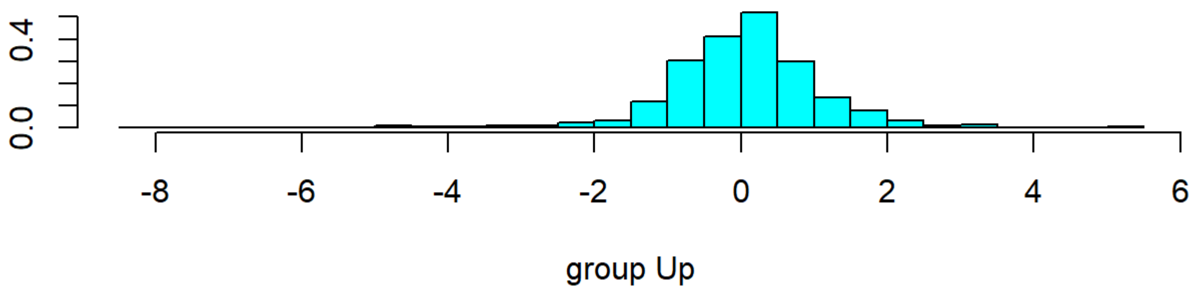
E.LDA

```
lda_wkly <- lda(Direction ~ Lag2, data = Weekly, subset = train)
lda_wkly
```

```
## Call:
## lda(Direction ~ Lag2, data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
##
## Coefficients of linear discriminants:
##            LD1
## Lag2 0.4414162
```

```
plot(lda_wkly)
```






```
lda_probs <- predict(lda_wkly, Weekly_test)
table(lda_probs$class, Direction_test)
```

```
##       Direction_test
##        Down Up
##   Down    9  5
##   Up     34 56
```

```
mean(lda_probs$class == Direction_test)
```

```
## [1] 0.625
```

Again, LDA is performing same as logistic regression.

F.QDA

```
qda_wkly <- qda(Direction ~ Lag2, data = Weekly, subset = train)
qda_wkly
```

```
## Call:
## qda(Direction ~ Lag2, data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
```

```
qda_pred <- predict(qda_wkly, Weekly_test)
table(qda_pred$class, Direction_test)
```

```
##       Direction_test
##        Down Up
##   Down    0  0
##   Up     43 61
```

```
mean(qda_pred$class == Direction_test)
```

```
## [1] 0.5865385
```

QDA is actually performing worst than both LDA and logistic regression.

G.KNN with k=1

```
train_X <- as.matrix(Weekly$Lag2[train])
test_X <- as.matrix(Weekly$Lag2[!train])

set.seed(1)
knn_pred <- knn(train_X, test_X, Direction_train, k = 1)
table(knn_pred, Direction_test)
```

```
##         Direction_test
## knn_pred Down Up
##     Down   21 30
##     Up     22 31
```

```
mean(knn_pred == Direction_test)
```

```
## [1] 0.5
```

Actually KNN is worst of all the other models

H. Clearly Logistic and LDA are almost equally accurate. QDA acting little bad and KNN being worst. Clearly KNN and QDA are producing more test errors because of overfitting indicating the relation between probability of direction and lag2 predictor is more of linear.

I.Let’s first see logistic models

```
logistic_wkly3 <- glm(Direction ~ Lag2:Lag1,
                     data = Weekly_train,
                     family = binomial)
summary(logistic_wkly3)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2:Lag1, family = binomial, data = Weekly_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.368  -1.269   1.077   1.089   1.353
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.21333    0.06421   3.322 0.000893 ***
## Lag2:Lag1    0.00717    0.00697   1.029 0.303649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1353.6  on 983  degrees of freedom
## AIC: 1357.6
##
## Number of Fisher Scoring iterations: 4
```

```
logistic_probs3 <- predict(logistic_wkly3, Weekly_test, type = "response")
logistic_pred3 = rep("Down", length(Direction_test))
logistic_pred3[logistic_probs3 > 0.5] <- "Up"
table(logistic_pred3, Direction_test)
```

```
##               Direction_test
## logistic_pred3 Down Up
##           Down    1  1
##           Up     42 60
```

```
mean(logistic_pred3 == Direction_test)
```

```
## [1] 0.5865385
```

Let’s try 1 more time with lag 1,2 and 3

```
logistic_wkly4 <- glm(Direction ~ Lag3+Lag2+Lag1,
                     data = Weekly_train,
                     family = binomial)
summary(logistic_wkly4)
```

```
##
## Call:
## glm(formula = Direction ~ Lag3 + Lag2 + Lag1, family = binomial,
##     data = Weekly_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.638  -1.255   1.000   1.088   1.510
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.212334   0.064698   3.282  0.00103 **
## Lag3        -0.008887   0.028830  -0.308  0.75788
## Lag2         0.053092   0.029128   1.823  0.06834 .
## Lag1        -0.053680   0.028924  -1.856  0.06347 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1346.9  on 981  degrees of freedom
## AIC: 1354.9
##
## Number of Fisher Scoring iterations: 4
```

```
logistic_probs4 <- predict(logistic_wkly4, Weekly_test, type = "response")
logistic_pred4 = rep("Down", length(Direction_test))
logistic_pred4[logistic_probs3 > 0.5] <- "Up"
table(logistic_pred4, Direction_test)
```

```
##                Direction_test
## logistic_pred4 Down Up
##           Down    1  1
##           Up     42 60
```

```
mean(logistic_pred4 == Direction_test)
```

```
## [1] 0.5865385
```

Clearly lag3 shouldn’t be used a predictor at all.

Let’s try once again

```
logistic_wkly5 <- glm(Direction ~ Lag4+Lag3+Lag2+Lag1,
                     data = Weekly_train,
                     family = binomial)
summary(logistic_wkly5)
```

```
##
## Call:
## glm(formula = Direction ~ Lag4 + Lag3 + Lag2 + Lag1, family = binomial,
##     data = Weekly_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6093  -1.2529   0.9959   1.0884   1.4730
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.21562    0.06488   3.323 0.000889 ***
## Lag4        -0.02195    0.02881  -0.762 0.446049
## Lag3        -0.01095    0.02909  -0.376 0.706655
## Lag2         0.05448    0.02919   1.866 0.061989 .
## Lag1        -0.05494    0.02896  -1.897 0.057849 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1346.3  on 980  degrees of freedom
## AIC: 1356.3
##
## Number of Fisher Scoring iterations: 4
```

```
logistic_probs5 <- predict(logistic_wkly5, Weekly_test, type = "response")
logistic_pred5 = rep("Down", length(Direction_test))
logistic_pred5[logistic_probs5 > 0.5] <- "Up"
table(logistic_pred5, Direction_test)
```

```
##               Direction_test
## logistic_pred5 Down Up
##           Down    8  8
##           Up     35 53
```

```
mean(logistic_pred5 == Direction_test)
```

```
## [1] 0.5865385
```

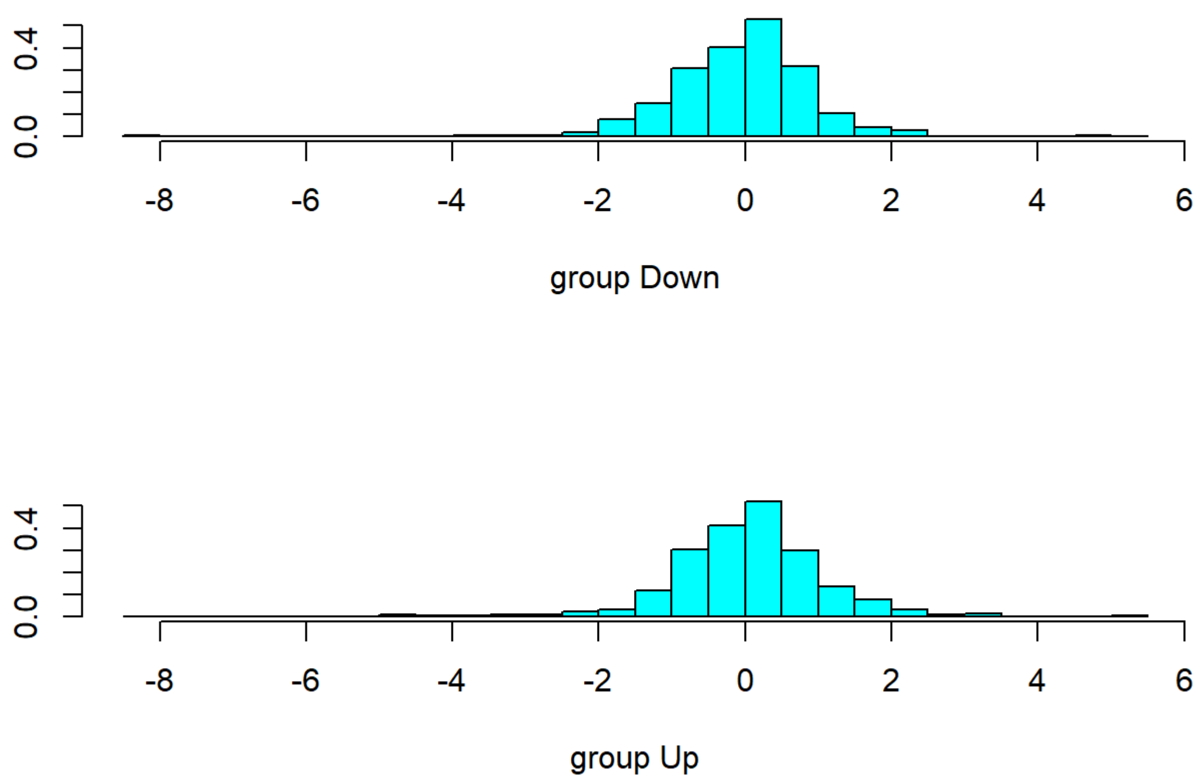
Lag4 is also not a good choice of variable.

Let’s try LDA now

```
lda_wkly2 <- lda(Direction ~ Lag2:Lag1,
                 data = Weekly,
                 subset = train)
lda_wkly2
```

```
## Call:
## lda(Direction ~ Lag2:Lag1, data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##       Lag2:Lag1
## Down -0.8014495
## Up   -0.1393632
##
## Coefficients of linear discriminants:
##                 LD1
## Lag2:Lag1 0.1013404
```

```
plot(lda_wkly)
```



```
lda_probs2 <- predict(lda_wkly2, Weekly_test)
table(lda_probs2$class, Direction_test)
```

```
##       Direction_test
##        Down Up
##   Down    0  1
##   Up     43 60
```

```
mean(lda_probs2$class == Direction_test)
```

```
## [1] 0.5769231
```

Different QDA model with transformation

```
qda_wkly2 <- qda(Direction ~ Lag2 + sqrt(abs(Lag2)),
                 data = Weekly,
                 subset = train)
qda_wkly2
```

```
## Call:
## qda(Direction ~ Lag2 + sqrt(abs(Lag2)), data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2 sqrt(abs(Lag2))
## Down -0.03568254        1.140078
## Up    0.26036581        1.169635
```

```
qda_pred2 <- predict(qda_wkly2, Weekly_test)
table(qda_pred2$class, Direction_test)
```

```
##       Direction_test
##        Down Up
##   Down   12 13
##   Up     31 48
```

```
mean(qda_pred2$class == Direction_test)
```

```
## [1] 0.5769231
```

Not improving the performance at all

Different KNN model

```
set.seed(1)
knn_pred3 <- knn(train_X, test_X, Direction_train, k = 3)
table(knn_pred3, Direction_test)
```

```
##          Direction_test
## knn_pred3 Down Up
##      Down   16 20
##      Up     27 41
```

```
mean(knn_pred3 == Direction_test)
```

```
## [1] 0.5480769
```

Let’s change K=20

```
set.seed(1)
knn_pred4 <- knn(train_X, test_X, Direction_train, k = 20)
table(knn_pred4, Direction_test)
```

```
##          Direction_test
## knn_pred4 Down Up
##      Down   21 21
##      Up     22 40
```

```
mean(knn_pred4 == Direction_test)
```

```
## [1] 0.5865385
```

performance increased a bit

Let’s try with K=50

```
set.seed(1)
knn_pred5 <- knn(train_X, test_X, Direction_train, k = 50)
table(knn_pred5, Direction_test)
```

```
##          Direction_test
## knn_pred5 Down Up
##      Down   20 23
##      Up     23 38
```

```
mean(knn_pred5 == Direction_test)
```

```
## [1] 0.5576923
```

Performance decreased as K increased from 20 to 50. Let’s try 10 once

```
set.seed(1)
knn_pred6 <- knn(train_X, test_X, Direction_train, k = 10)
table(knn_pred6, Direction_test)
```

```
##          Direction_test
## knn_pred6 Down Up
##      Down   17 21
##      Up     26 40
```

```
mean(knn_pred6 == Direction_test)
```

```
## [1] 0.5480769
```

```
set.seed(1)
knn_pred7 <- knn(train_X, test_X, Direction_train, k = 30)
table(knn_pred7, Direction_test)
```

```
##          Direction_test
## knn_pred7 Down Up
##      Down   20 24
##      Up     23 37
```

```
mean(knn_pred7 == Direction_test)
```

```
## [1] 0.5480769
```

```
set.seed(1)
knn_pred8 <- knn(train_X, test_X, Direction_train, k = 25)
table(knn_pred8, Direction_test)
```

```
##          Direction_test
## knn_pred8 Down Up
##      Down   19 25
##      Up     24 36
```

```
mean(knn_pred8 == Direction_test)
```

```
## [1] 0.5288462
```

So, it seems K =20 seems to be best producing accuracy among all.