Classification

Hrishabh Khakurel

6/6/2020

Contents

4.7.10 **2**

Dataset . . . 2

(a) . . . 2

(b) . . . 6

(c) . . . 6

(d) . . . 7

(e) . . . 8

(f) . . . 11

(h) . . . 12

4.7.11 **12**

(a) . . . 13

(b) . . . 13

(c) . . . 16

(d) . . . 16

(e) . . . 17

(f) . . . 18

(g) . . . 19

4.7.13 **20**

Visualizing the correlation among the variables . . . 20

Train-Test Split . . . 21

Logistic Regression . . . 22

4.7.10

This problem uses the *Weekly* data set from *ISLR* package. It contains 1,089 weekly returns for 21 years, from the begining of 1990 to the end of 2010.

Dataset

```
weekly <- Weekly
kableExtra::kable(head(weekly, n = 10))
```

Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
1990	0.816	1.572	-3.936	-0.229	-3.484	0.1549760	-0.270	Down
1990	-0.270	0.816	1.572	-3.936	-0.229	0.1485740	-2.576	Down
1990	-2.576	-0.270	0.816	1.572	-3.936	0.1598375	3.514	Up
1990	3.514	-2.576	-0.270	0.816	1.572	0.1616300	0.712	Up
1990	0.712	3.514	-2.576	-0.270	0.816	0.1537280	1.178	Up
1990	1.178	0.712	3.514	-2.576	-0.270	0.1544440	-1.372	Down
1990	-1.372	1.178	0.712	3.514	-2.576	0.1517220	0.807	Up
1990	0.807	-1.372	1.178	0.712	3.514	0.1323100	0.041	Up
1990	0.041	0.807	-1.372	1.178	0.712	0.1439720	1.253	Up
1990	1.253	0.041	0.807	-1.372	1.178	0.1336350	-2.678	Down

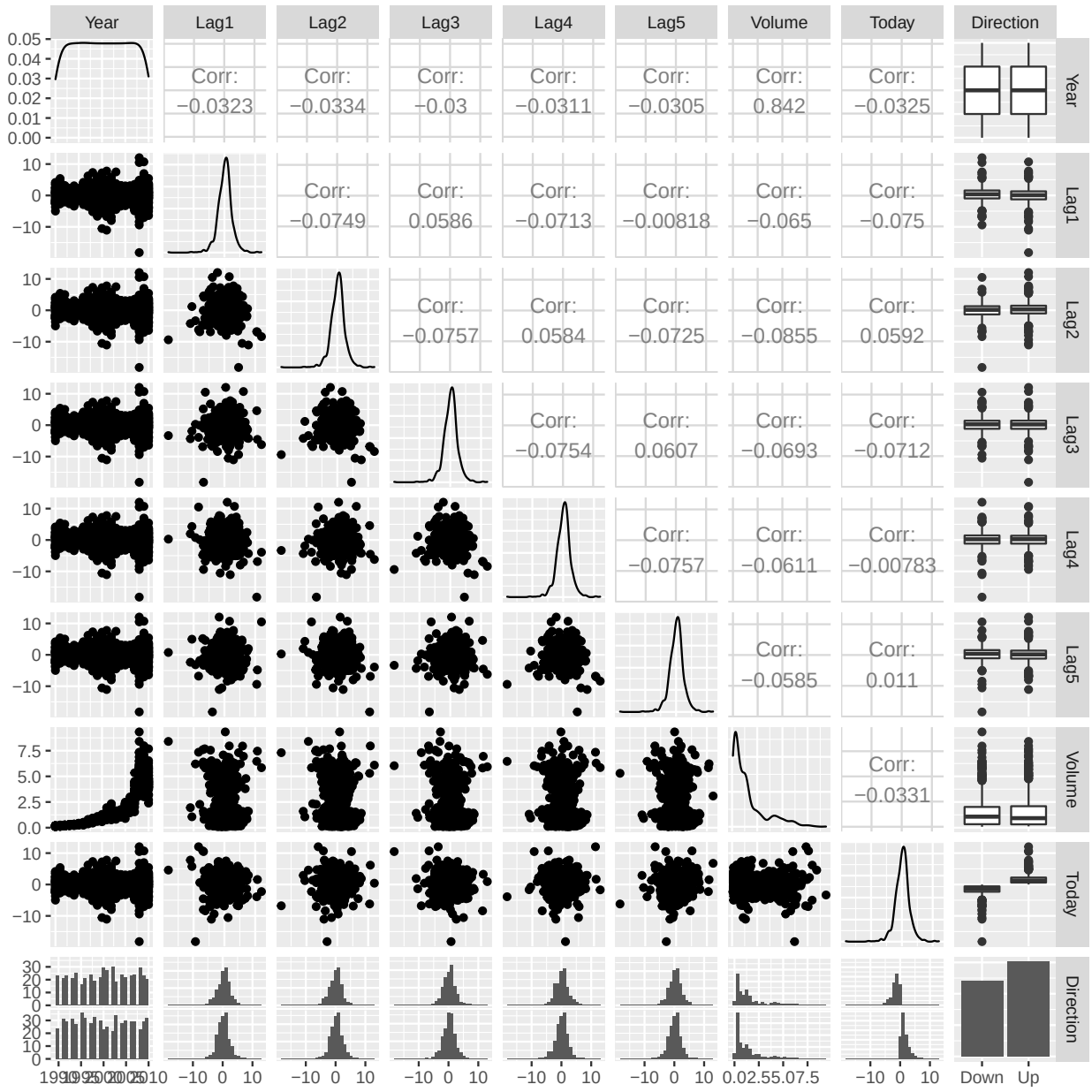
(a)

Pairs Plot

```
GGally::ggpairs(weekly, progress =F)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

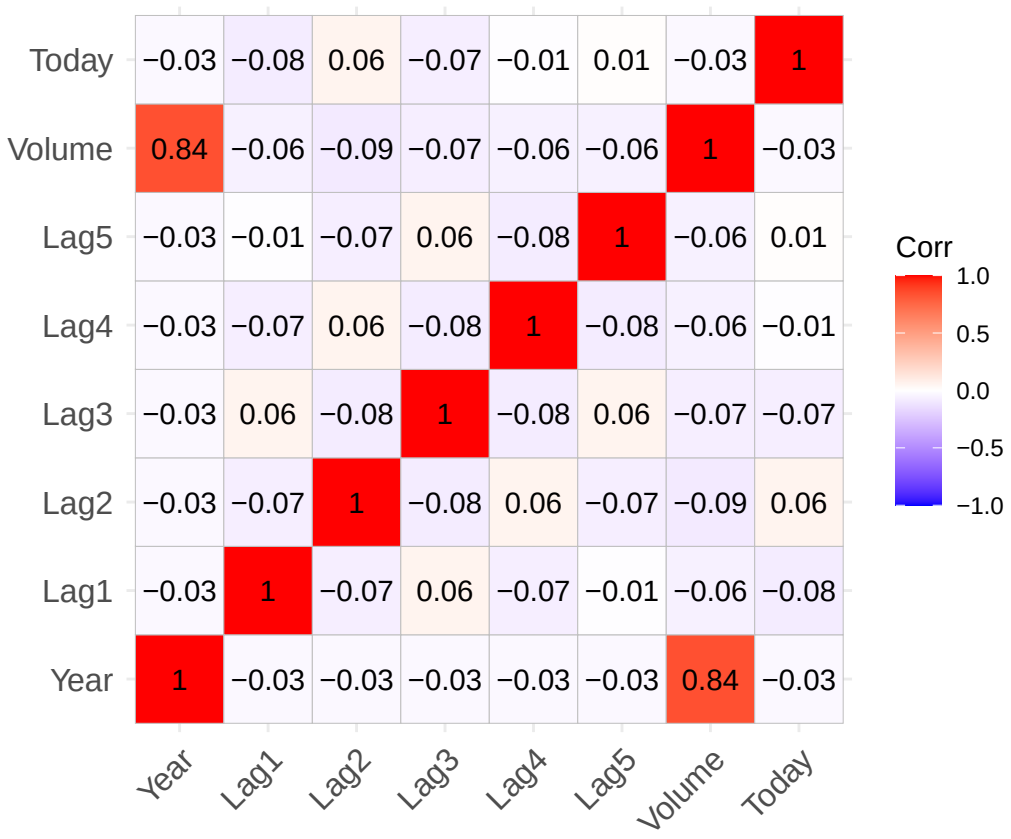
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see that most of the variables donot have much relationship, but the Volume and Year variables have a significant relationship.

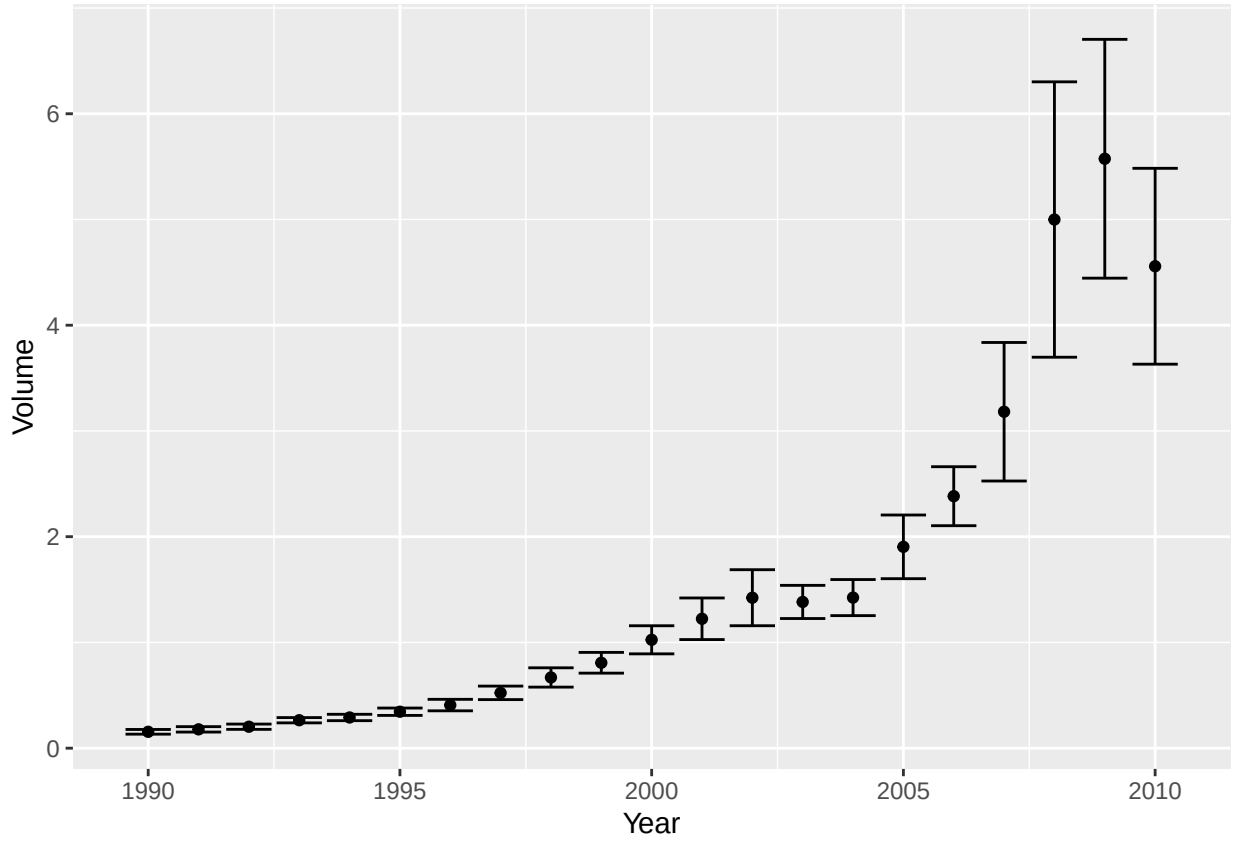
Correlation Heat Map

```
corr <- cor(weekly[-9])
ggcorrplot::ggcorrplot(corr, lab =T)
```



As seen from the pairs plot, the correlation heat map also shows that there is a significant correlation between the Volume and the Year. We will now visualize this relationship.

```
ggplot(data = weekly, aes (x = Year, y = Volume))+
  stat_summary(fun.y = mean, geom = 'point') +
  stat_summary(fun.data = mean_sdl, fun.args = list(mult = 1),geom = 'errorbar')
```



As, we can see the Volume traded is increases with the year.

Some numerical summaries:

```
mn <- vector()
med <-  vector()
maximum <-  vector()
minimum <-  vector()
std_dev <-  vector()

for (i in 2:8){
  mn[i-1] <- round(mean(weekly[,i]),3)
  med[i-1] <- round(median(weekly[,i]),3)
  maximum[i-1] <- round(max(weekly[,i]),3)
  minimum[i-1] <- round(min(weekly[,i]),3)
  std_dev[i-1] <- round(sqrt(var(weekly[,i])),3)
}
var_names <- colnames(weekly)[-c(1,9)]
summary_stat <- data.frame(Variables = var_names,Mean = mn, Median = med,
                           Maximum = maximum, Minimum = minimum, Std_deviation = std_dev)
kableExtra::kable(summary_stat)
```

Variables	Mean	Median	Maximum	Minimum	Std_deviation
Lag1	0.151	0.241	12.026	-18.195	2.357
Lag2	0.151	0.241	12.026	-18.195	2.357
Lag3	0.147	0.241	12.026	-18.195	2.361
Lag4	0.146	0.238	12.026	-18.195	2.360
Lag5	0.140	0.234	12.026	-18.195	2.361
Volume	1.575	1.003	9.328	0.087	1.687
Today	0.150	0.241	12.026	-18.195	2.357

(b)

In this section, we are going to perform a logistic regression.

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

```
##
## 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
```

Looking at the above regression summary we can see that only the Lag 2 variable is significant. The others have high p-values.

(c)

Using the above logistic model we are going to predict the Direction and compute the confusion matrix.

```r
glm_probs <- predict(glm_fit, type = 'response')
glm_pred <- rep("Up",nrow(weekly))
glm_pred[glm_probs < 0.5] = 'Down'
table(glm_pred,weekly$Direction)
```

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

```r
Metrics <- c('Sensitivity','Specificity','Precision','Accuracy')
Values1 <- c(round(((557/(48+557))*100),2),round(((54/(54+430))*100),2),
             round(((557/(430+557))*100),2),
             round((((557+54)/nrow(weekly))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values1))
```

Metrics	Values
Sensitivity	92.07
Specificity	11.16
Precision	56.43
Accuracy	56.11

As we see the accuracy is not very high, it is only equal to 56.11%, which is slightly better than random guessing. THe precision also arounf the same limit.

(d)

We saw that only Lag2 was significant. So, in this model we will use only Lag2 to predict direction. We are also going to split the data into training and test set.

Train-Test split

```r
test <- weekly$Year > 2008
test_data <- weekly[test,]
train_data <- weekly[!test,]
```

```r
glm_fit2 <- glm(Direction ~ Lag2, data = train_data, family = binomial)
summary(glm_fit2)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = train_data)
##
## 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
```

Using the above model to predict results for the test set.

```
glm_probs2 <- predict(glm_fit2,newdata = test_data)
glm_pred2 <- rep("Up",nrow(test_data))
glm_pred2[glm_probs2 < 0.5] = 'Down'
table(glm_pred2, test_data$Direction)
```

```
##
## glm_pred2 Down Up
##      Down   41 56
##      Up      2  5
```

```
Values2 <- c(round(((5/(56+5))*100),2),round(((41/(41+2))*100),2),round(((5/(5+2))*100),2),
             round((((41+5)/nrow(test_data))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values2))
```

Metrics	Values
Sensitivity	8.20
Specificity	95.35
Precision	71.43
Accuracy	44.23

In this case the accuracy is lower than random guessing. But the precision is okay. This means that when the model predicts "Up" Direction it is correct 71.43% of the time.

(e)

In this section we are going to use LDA classifier.

```
lda_fit <- lda(Direction ~ Lag2, data = train_data)
lda_fit
```

```
## Call:
## lda(Direction ~ Lag2, data = train_data)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
```

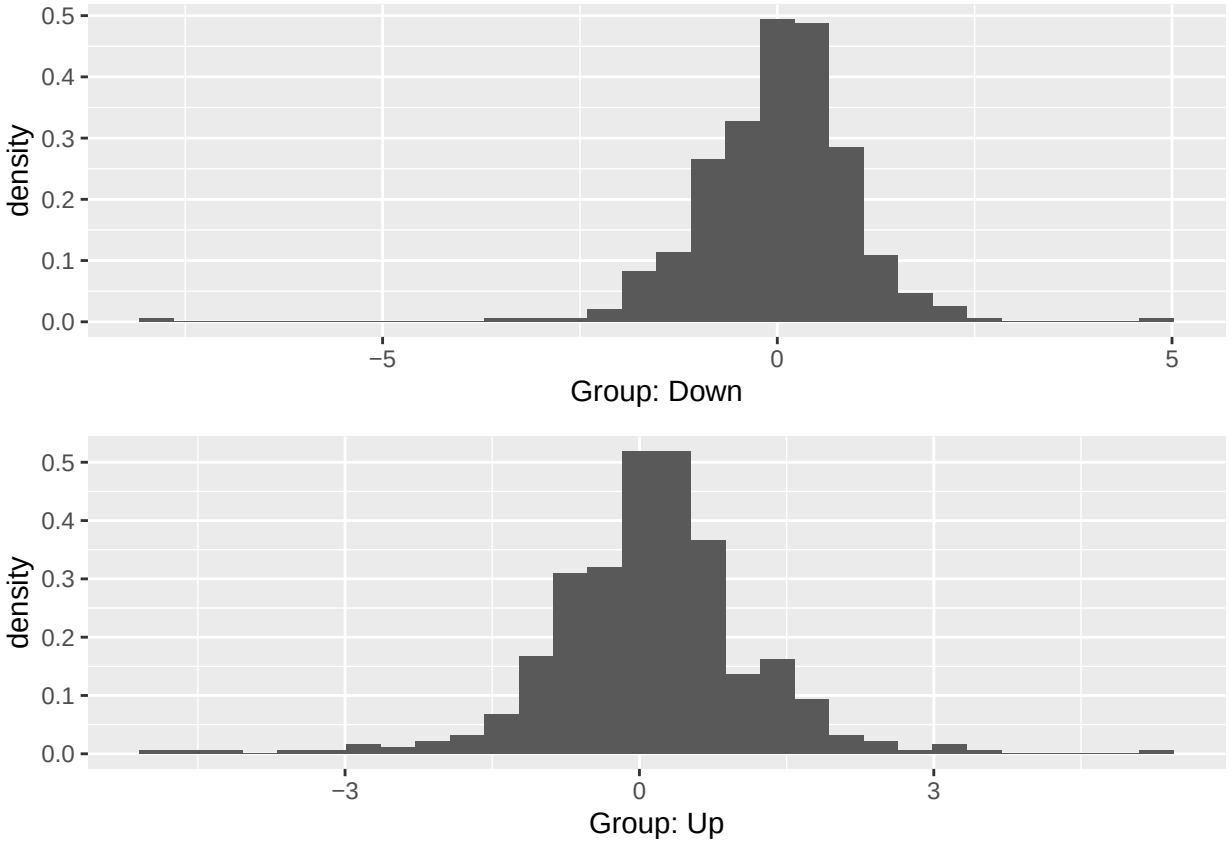
```
## Up    0.26036581
##
## Coefficients of linear discriminants:
##            LD1
## Lag2 0.4414162
```

From the summary above, we can see that the prior probability for market going down is 44.77% and market going up is 55.22%. We also see that if the market goes up, then there is a tendency for the Lag2 to be positive, wheras if the market goes down then there is a tendency for Lag2 to be negative. The coefficient of linear discriminants is used to calculate the decision rule. If (0.4414 * Lag2) is large, then that observation is calssified as "Up" else it is classified as "Down".

Visualizing the Classifier

```
group1 <- train_data[train_data$Direction == "Up",]$Lag2
group2 <-train_data[train_data$Direction == "Down",]$Lag2
prob1 <- 0.4414162 * group1
prob2 <- 0.4414162 * group2
plot1 <- ggplot(data.frame(prob1), aes(prob1)) +
  geom_histogram(aes(y = ..density..)) +
  xlab("Group: Up")
plot2 <- ggplot(data.frame(prob2), aes(prob2)) +
  geom_histogram(aes(y = ..density..)) +
  xlab("Group: Down")
gridExtra::grid.arrange(plot2, plot1, ncol =1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Predicting the results

```
pred_lda <- predict(lda_fit,test_data)
table(pred_lda$class,test_data$Direction)
```

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

Metrics

```
Values3 <- c(round(((56/(56+5))*100),2),round(((9/(34+9))*100),2),round(((56/(56+34))*100),2),
             round((((56+9)/nrow(test_data))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values3))
```

Metrics	Values
Sensitivity	91.80
Specificity	20.93
Precision	62.22
Accuracy	62.50

We have increased the accuracy to 62.50%. But the precision has dropped significantly. The sensitivity of the model is very high.

(f)

In this section we are going to use QDA classifier.

```
qda_fit <- qda(Direction ~ Lag2, data = train_data)
qda_fit
```

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

The interpretation of the Prio probabilities and the group means are the same as in answer (e).

Predicting the results

```
pred_qda <- predict(qda_fit,newdata = data.frame(Lag2 = test_data$Lag2))
table(pred_qda$class,test_data$Direction)
```

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

Metrics

```
Values4 <- c(round(((61/(61))*100),2),0,round(((61/(61+43))*100),2),
             round((((61)/nrow(test_data))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values4))
```

Metrics	Values
Sensitivity	100.00
Specificity	0.00
Precision	58.65
Accuracy	58.65

This model predicts only "Up" direction. Thus it is not a good idea to use this model.

(h)

First, we need to prepare the inputs.

```
train_X <- data.frame(weekly$Lag2[!test])
test_X <- data.frame(weekly$Lag2[test])
train_direction <- weekly$Direction[!test]
```

Prediction

```
set.seed(1)
knn_pred <- knn(train_X,test_X, train_direction, k = 1)
# Confusion Matrix
table(knn_pred, test_data$Direction)
```

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

Metrics

```
Values5 <- c(round(((31/(61))*100),2),round(((21/(21 + 22))*100),2),round(((31/(31+22))*100),2),
             round((((31+21)/nrow(test_data))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values5))
```

Metrics	Values
Sensitivity	50.82
Specificity	48.84
Precision	58.49
Accuracy	50.00

The accuracy is lower than before and is equal to random guessing. So, this model might not be the best.

(h)

From the above metrics and analysis we can see that the best classifier for this data is LDA.

4.7.11

In this problem we will be using the *Auto* dataset to predict whether a given car gets high or low gas mileage based on the Auto data set.

Loading the Dataset

```
auto <- Auto
kableExtra::kable(head(auto, n = 10))
```

mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
18	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu
15	8	350	165	3693	11.5	70	1	buick skylark 320
18	8	318	150	3436	11.0	70	1	plymouth satellite
16	8	304	150	3433	12.0	70	1	amc rebel sst
17	8	302	140	3449	10.5	70	1	ford torino
15	8	429	198	4341	10.0	70	1	ford galaxie 500
14	8	454	220	4354	9.0	70	1	chevrolet impala
14	8	440	215	4312	8.5	70	1	plymouth fury iii
14	8	455	225	4425	10.0	70	1	pontiac catalina
15	8	390	190	3850	8.5	70	1	amc ambassador dpl

(a)

Creating the classifying variable.

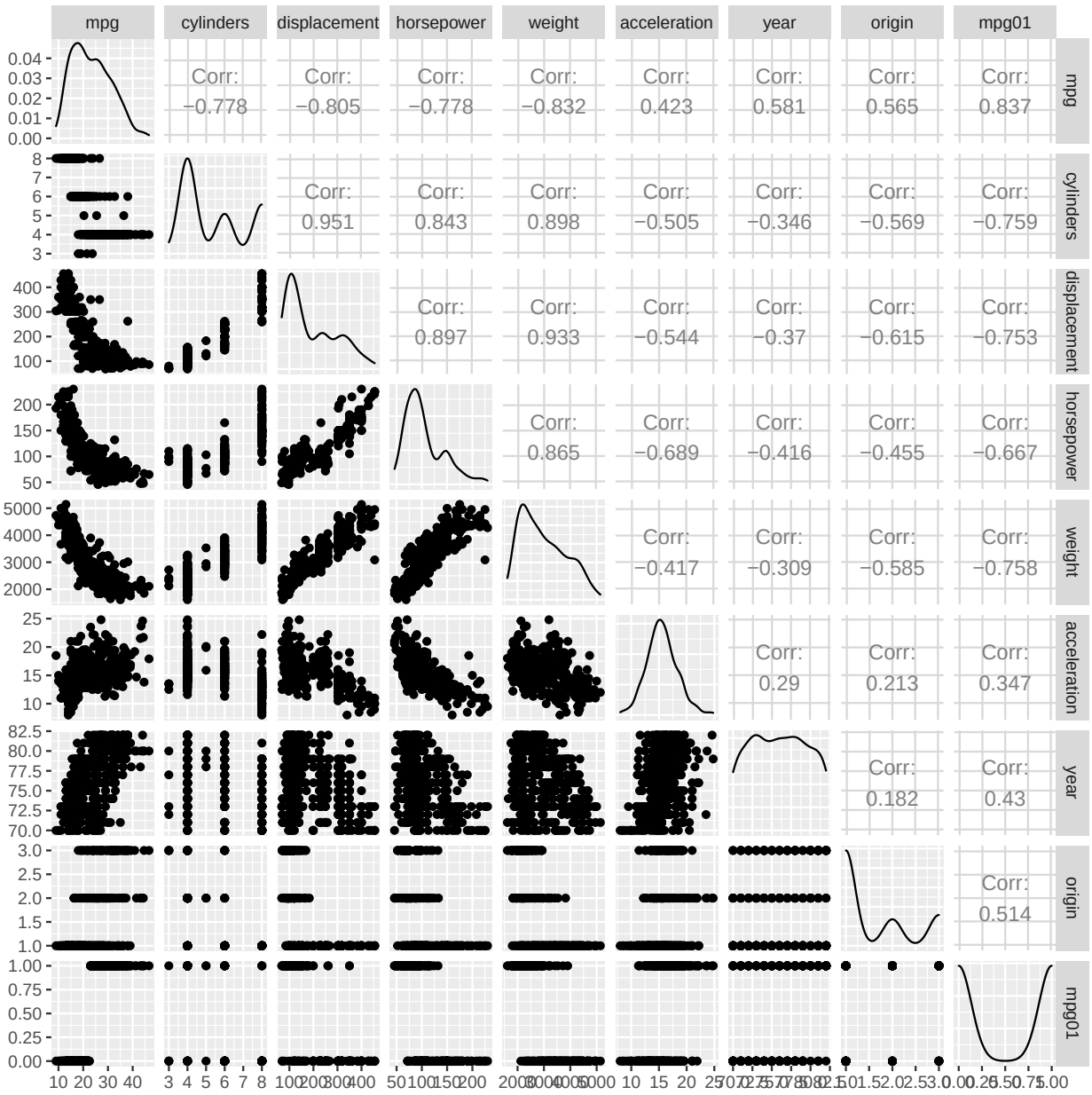
```
mpg01 <- data.frame(rep(1,nrow(auto)))
mpg01[auto$mpg < median(auto$mpg),] = 0
auto <- cbind(auto, mpg01)
col_names <- colnames(auto)
col_names[10] = 'mpg01'
colnames(auto) <- col_names
kableExtra::kable(head(auto, n = 5))
```

mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name	mpg01
18	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu	0
15	8	350	165	3693	11.5	70	1	buick skylark 320	0
18	8	318	150	3436	11.0	70	1	plymouth satellite	0
16	8	304	150	3433	12.0	70	1	amc rebel sst	0
17	8	302	140	3449	10.5	70	1	ford torino	0

(b)

Pairs Plot

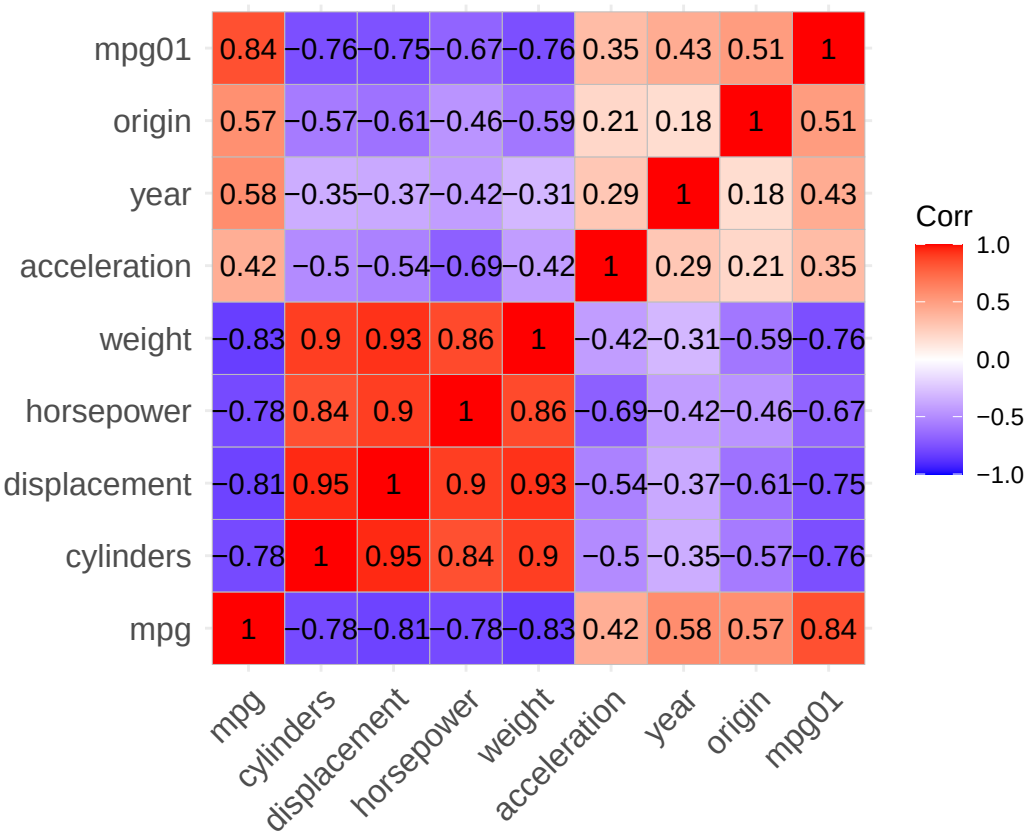
```
GGally::ggpairs(auto[-9], progress = F)
```



Except acceleration, year and origin all the other variables have a stron relationship with the mpg01 variable. We will confirm this using the correlation heat map.

Correlation Heat Map

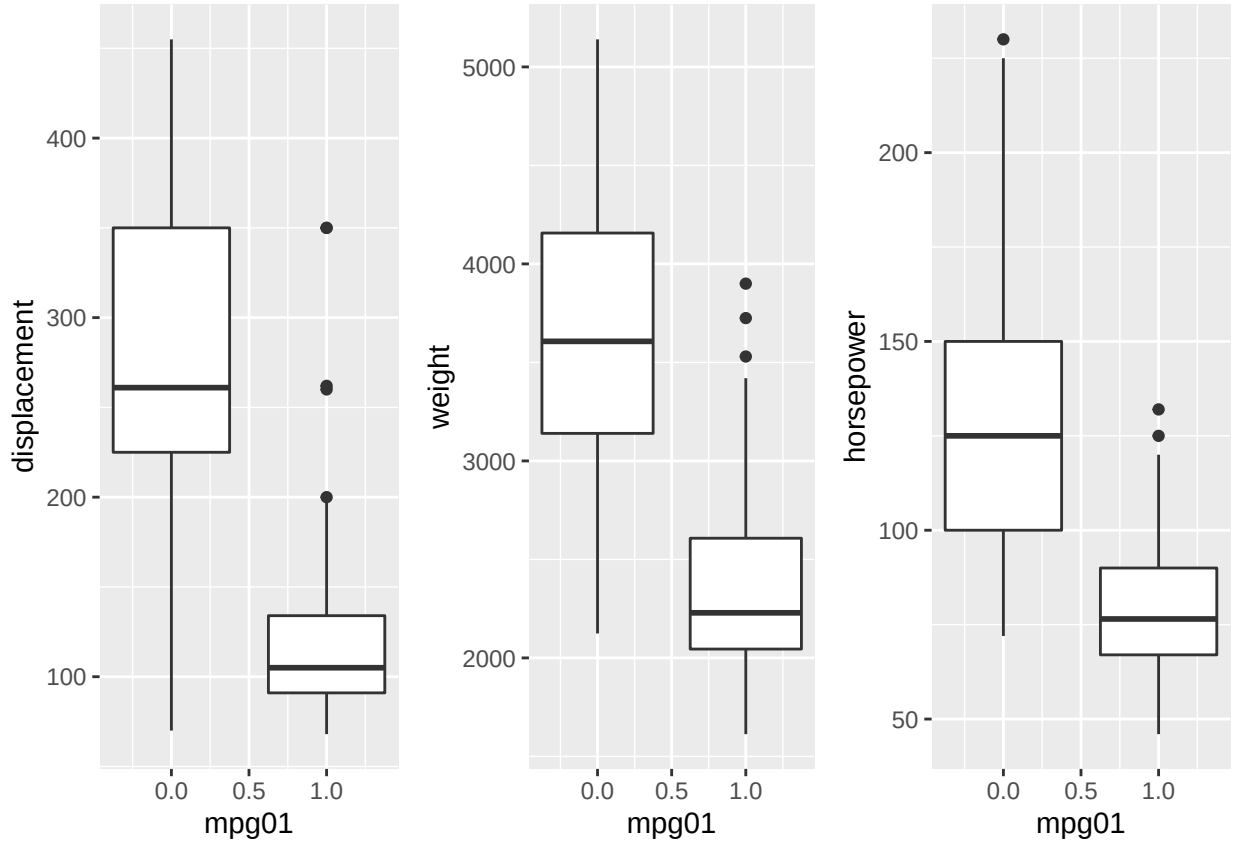
```
corr2 <- cor(auto[-9])
ggcorrplot::ggcorrplot(corr2, lab = T)
```



The correlation plot shows strong relationsips between some of the variables. cylinders, displacement, horsepower and weight have the strongest association.

We will check some boxplots to see how the variables are affecting the mpg01 variable.

```
bp1 <- ggplot(data = data.frame(auto[-9]), aes(x = mpg01, y = displacement, group = mpg01)) +
  geom_boxplot()
bp2 <-  ggplot(data = data.frame(auto[-9]), aes(x = mpg01, y = weight, group = mpg01)) +
  geom_boxplot()
bp3 <-  ggplot(data = data.frame(auto[-9]), aes(x = mpg01, y = horsepower, group = mpg01)) +
  geom_boxplot()
gridExtra::grid.arrange(bp1,bp2,bp3, ncol = 3)
```



So, cars with higher mpg have lower displacement. Similarly, these cars also have lower weight and these cars also have lower horsepower.

(c)

We are going to split the data into train and test set.

```
train <- 1:314
train_set2 <- auto[train,-9]
test_set2 <- auto[-train,-9]
```

(d)

```
lda_fit2 <- lda(mpg01 ~ cylinders + displacement + horsepower + weight,
                data = train_set2)
lda_fit2
```

```
## Call:
## lda(mpg01 ~ cylinders + displacement + horsepower + weight, data = train_set2)
##
## Prior probabilities of groups:
##         0         1
## 0.6082803 0.3917197
```

```
##
## Group means:
##   cylinders displacement horsepower   weight
## 0  6.785340     274.4817  130.97906 3630.592
## 1  4.138211     112.1504   78.60976 2282.756
##
## Coefficients of linear discriminants:
##                         LD1
## cylinders    -0.4166610446
## displacement -0.0007017588
## horsepower    0.0062536470
## weight       -0.0011423654
```

The prior probability shows that 60% of the cars have mpg01 = 0 and the rest have mpg01 = 1. The group means show that the cars with mpg01 = 0 have higher no. of cylinders, greater displacement, horsepower and weight.

Predictions

```
lda_pred <- predict(lda_fit2,test_set2)$class
table(lda_pred, test_set2$mpg01)
```

```
##
## lda_pred  0  1
##        0  5 11
##        1  0 62
```

Metrics

```
Values6 <- c(round(((62/(62 + 11))*100),2),0,round(((62/(62))*100),2),
             round((((62 + 5)/nrow(test_set2))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values6))
```

Metrics	Values
Sensitivity	84.93
Specificity	0.00
Precision	100.00
Accuracy	85.90

This model has fairly high Accuracy. And the error is equal to 14.10%. However, this model doesn't predict mpg = 0 at all. So it may not be beneficial to use this model.

(e)

```
qda_fit2 <- qda(mpg01 ~ cylinders + displacement + horsepower + weight,
                data = train_set2)
qda_fit2
```

```
## Call:
## qda(mpg01 ~ cylinders + displacement + horsepower + weight, data = train_set2)
##
```

```
## Prior probabilities of groups:
##         0         1
## 0.6082803 0.3917197
##
## Group means:
##   cylinders displacement horsepower   weight
## 0  6.785340     274.4817  130.97906 3630.592
## 1  4.138211     112.1504   78.60976 2282.756
```

The interpretation is the same as before.

Predictions

```
qda_pred <- predict(qda_fit2,test_set2)$class
table(qda_pred, test_set2$mpg01)
```

```
##
## qda_pred  0  1
##        0  5 11
##        1  0 62
```

Metrics

```
Values7 <- c(round(((62/(62 + 11))*100),2),0,round(((62/(62))*100),2),
             round((((62 + 5)/nrow(test_set2))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values7))
```

Metrics	Values
Sensitivity	84.93
Specificity	0.00
Precision	100.00
Accuracy	85.90

The results are same as before. Hence, this model may not be benificial.

(f)

We will create a logistic regression classifier.

```
glm_fit3 <- glm(mpg01 ~ cylinders + displacement + horsepower + weight,
                data = train_set2, family = binomial )
summary(glm_fit3)
```

```
##
## Call:
## glm(formula = mpg01 ~ cylinders + displacement + horsepower +
##     weight, family = binomial, data = train_set2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2682  -0.2527  -0.0138   0.3114   3.6121
##
```

```
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  12.5287730  2.1401259   5.854 4.79e-09 ***
## cylinders    -0.1557830  0.4607980  -0.338  0.73531
## displacement -0.0097865  0.0110885  -0.883  0.37747
## horsepower   -0.0479934  0.0175872  -2.729  0.00635 **
## weight       -0.0021487  0.0008512  -2.524  0.01159 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 420.45  on 313  degrees of freedom
## Residual deviance: 152.15  on 309  degrees of freedom
## AIC: 162.15
##
## Number of Fisher Scoring iterations: 7
```

It seems that cylinders and displacement are not statistically significant.

Predictions

```
glm_probs3 <- predict(glm_fit3, test_set2)
glm_pred3 <- rep(1,nrow(test_set2))
glm_pred3[glm_probs3 < 0.5] = 0
table(glm_pred3, test_set2$mpg01)
```

```
##
## glm_pred3  0  1
##         0  5 20
##         1  0 53
```

Metrics

```
Values8 <- c(round(((53/(53 + 20))*100),2),0,round(((53/(53))*100),2),
             round((((53 + 5)/nrow(test_set2))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values8))
```

Metrics	Values
Sensitivity	72.60
Specificity	0.00
Precision	100.00
Accuracy	74.36

The accuracy is lower than before. The test error is 25.64%.

(g)

Preparing the data for knn

```
train_X2 <- cbind(auto$cylinders,auto$displacement,auto$horsepower,auto$weight)[train,]
test_X2 <- cbind(auto$cylinders,auto$displacement,auto$horsepower,auto$weight)[-train,]
test_Y2 <- auto$mpg01[train]
```

Predictions

```
set.seed(1)
knn_pred2 <- knn(train_X2, test_X2, test_Y2, k = 3)
table(knn_pred2,test_set2$mpg01)
```

```
##
## knn_pred2  0  1
##         0  5 19
##         1  0 54
```

Metrics

```
Values9 <- c(round(((54/(54 + 19))*100),2),0,round(((54/(54))*100),2),
             round((((54 + 5)/nrow(test_set2))*100),2))
kableExtra::kable(data.frame(Metrics,Values= Values9))
```

Metrics	Values
Sensitivity	73.97
Specificity	0.00
Precision	100.00
Accuracy	75.64

The test error is 24.36% which is higher than for the LDA and QDA.

4.7.13

We will be using the boston dataset for this exercise.

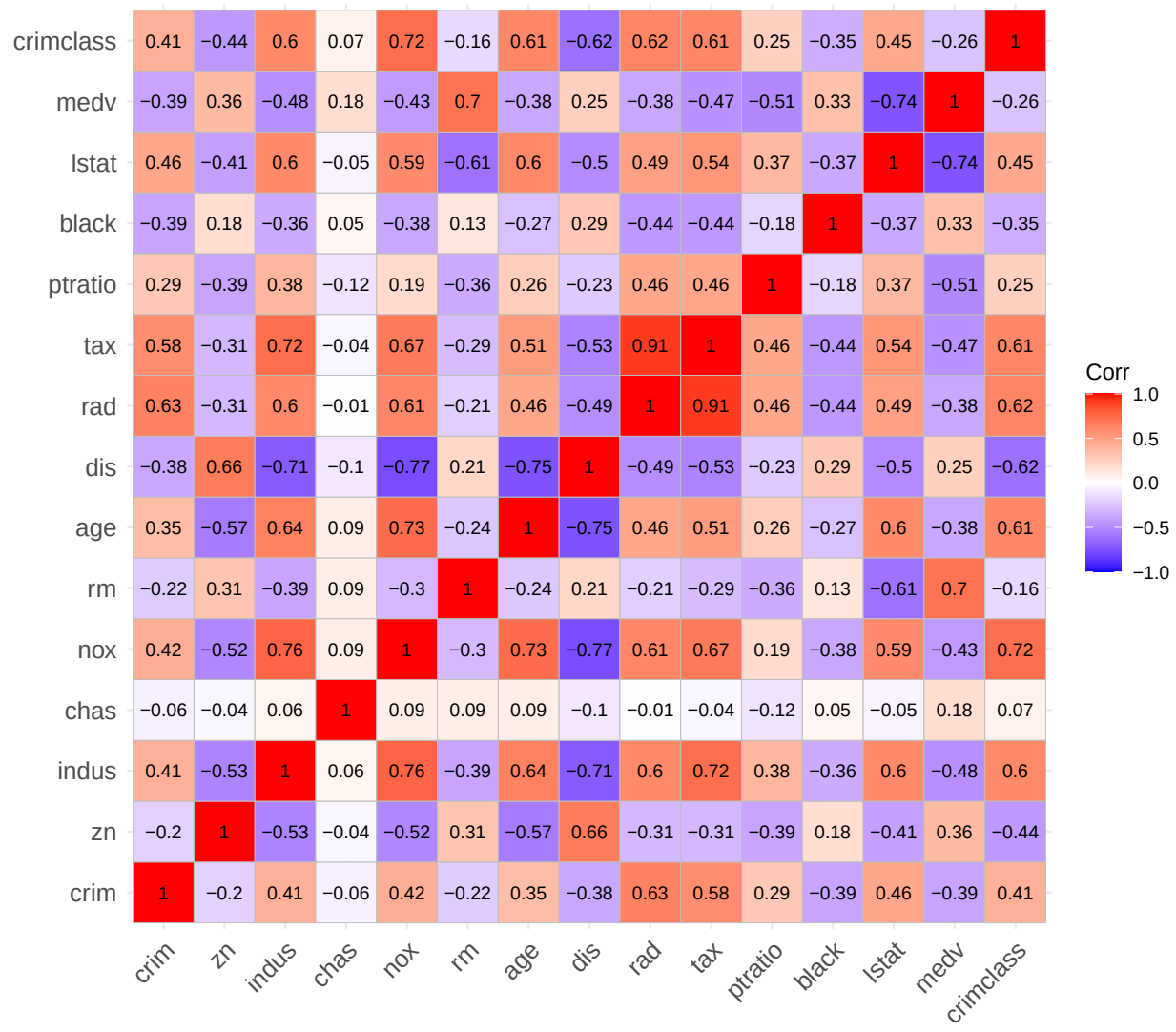
Loading the Data

```
boston <- Boston
# creating a new classifying variable
crimclass <- data.frame(rep(1,nrow(boston)))
crimclass[boston$crim < median(boston$crim),] = 0
boston <- data.frame(boston,crimclass)
col_names2 <- colnames(boston)
col_names2[15] = 'crimclass'
colnames(boston) <- col_names2
kableExtra::kable(head(boston))
```

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	crimclass
0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0	0
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6	0
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	0
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4	0
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2	0
0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7	0

Visualizing the correlation among the variables

```
corr3 <- cor(boston)
ggcorrplot::ggcorrplot(corr3, lab =TRUE, lab_size = 3)
```



Train-Test Split

```
train2 <- 1:406
train_boston <- boston[train2,]
test_boston <- boston[!train2,]
```

For this model we will be using logistic regression in the begining.

Logistic Regression

```
glm_fit4 <- glm(crimclass ~ . , data = train_boston, family = binomial )
summary(glm_fit4)
```

```
##
## Call:
## glm(formula = crimclass ~ ., family = binomial, data = train_boston)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.811e-03  -2.000e-08  -2.000e-08   2.000e-08   1.398e-03
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.072e+02  3.148e+04  -0.016    0.987
## crim         8.080e+02  1.675e+04   0.048    0.962
## zn           9.409e-01  1.099e+02   0.009    0.993
## indus       -2.841e+00  4.455e+02  -0.006    0.995
## chas        -5.734e+01  4.971e+03  -0.012    0.991
## nox          3.264e+02  5.156e+04   0.006    0.995
## rm          -1.698e+01  2.095e+03  -0.008    0.994
## age          6.863e-03  6.124e+01   0.000    1.000
## dis         -1.397e+01  1.050e+03  -0.013    0.989
## rad         -3.793e+00  2.123e+03  -0.002    0.999
## tax         -2.785e-01  5.712e+01  -0.005    0.996
## ptratio      1.742e+01  1.205e+03   0.014    0.988
## black        3.828e-02  1.494e+01   0.003    0.998
## lstat       -1.516e+00  2.309e+02  -0.007    0.995
## medv         4.329e+00  2.659e+02   0.016    0.987
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5.5071e+02  on 405  degrees of freedom
## Residual deviance: 9.3572e-06  on 391  degrees of freedom
## AIC: 30
##
## Number of Fisher Scoring iterations: 25
```

This shows that none of the variables are significant.

Predictions

```
glm_probs4 <- predict(glm_fit4, test_boston)
glm_preds4 <- rep(1,nrow(test_boston))
glm_preds4[glm_probs4 < 0.5] = 0
table(glm_preds4, test_boston$crimclass)
```

```
## < table of extent 0 x 0 >
```

This model is not good.