# Homework 3

Liting Hu

November 29, 2016

## Decision tree

### 1. Stock and technical indicators selection

We select PEP as our stock index. The 10 technical indicators are showed below:

| | |
|---|---|
| MACD | Moving Average Convergence Divergence |
| SMI | Stochastic Momentum Indicator |
| EMA5 | 5-period Exponential Moving Average |
| EMA15 | 15-period Exponential Moving Average |
| RSI3 | 3-period Relative Strength Index |
| RSI10 | 10-period Relative Strength Index |
| ROC | Rate Of Change |
| ADX7 | 7-period Average Directional Index |
| ADX14 | 14-period Average Directional Index |
| Volume | Volume |

Table 1: Technical indicators

```
startDate = as.Date("2012-01-01")
endDate = as.Date("2014-01-01")
getSymbols("PEP", from = startDate, to = endDate) # Get the stock symbol

# 1
MACD <- MACD(Op(PEP),fast = 12, slow = 26, signal = 9)
#Calculate a MACD with standard parameters
MACDsignal <- MACD[, 2]
#Grab just the signal line to use as our indicator.

# 2
SMI <- SMI(Op(PEP),n=13,slow=25,fast=2,signal=9)
#Stochastic Oscillator with standard parameters
SMI <- SMI[, 1]
#Grab just the oscillator to use as our indicator

# 3
EMA5 <- EMA(Op(PEP), n = 5)
#Calculate a 5-period exponential moving average (EMA)

# 4
EMA15 <- EMA(Op(PEP), n = 15)
#Calculate a 15-period exponential moving average (EMA)
```

```
# 5
RSI3 <- RSI(Op(PEP), n = 3)
#Calculate a 3-period relative strength index (RSI) off the open price

# 6
RSI10 <- RSI(Op(PEP), n = 10)
#Calculate a 10-period relative strength index (RSI) off the open price

# 7
ROC <- ROC(Op(PEP), n=1, type = c( "discrete"))
#Calculate Rate Of Change (ROC)

# 8
ADX7 <- ADX(HLC(PEP), n = 7)
ADX7 <- ADX7[, 4]
#Calculate a 7-period Average Direction Index (ADX)

# 9
ADX14 <- ADX(HLC(PEP), n = 14)
ADX14 <- ADX14[, 4]
#Calculate a 14-period Average Direction Index (ADX)

# 10
Volume <- Vo(PEP)
#Get volume
```

## 2. Construct the decision tree

```
#Then calculate the variable we are looking to predict and build our data sets.
PriceChange <- Cl(PEP) - Op(PEP)
#Calculate the difference between the close price and open price
Class <- ifelse(PriceChange > 0,"UP","DOWN")
#Create a binary classification variable, the variable we are trying to predict.
DataSet <- data.frame(MACDsignal, SMI, EMA5, EMA15, RSI3, RSI10,
                      ROC, ADX7, ADX14, Volume, Class)
#Create our data set
colnames(DataSet)<-c("MACDsignal", "SMI", "EMA5", "EMA15", "RSI3", "RSI10",
                     "ROC", "ADX7", "ADX14", "Volume", "Class")
#Name the columns
DataSet <- DataSet[-c(1:33),]
#Get rid of the data where the indicators are being calculated
TrainingSet <- DataSet[1:312,]
#Use 2/3 of the data to build the tree
TestSet <- DataSet[313:469,]
#And leave out 1/3 data to test our strategy

DecisionTree<-rpart(Class~MACDsignal + SMI + EMA5 + EMA15 + RSI3 +  RSI10 +
                    ROC + ADX7 + ADX14 + Volume, data=TrainingSet, cp=.001)

prp(DecisionTree,type=2,extra=8)
```

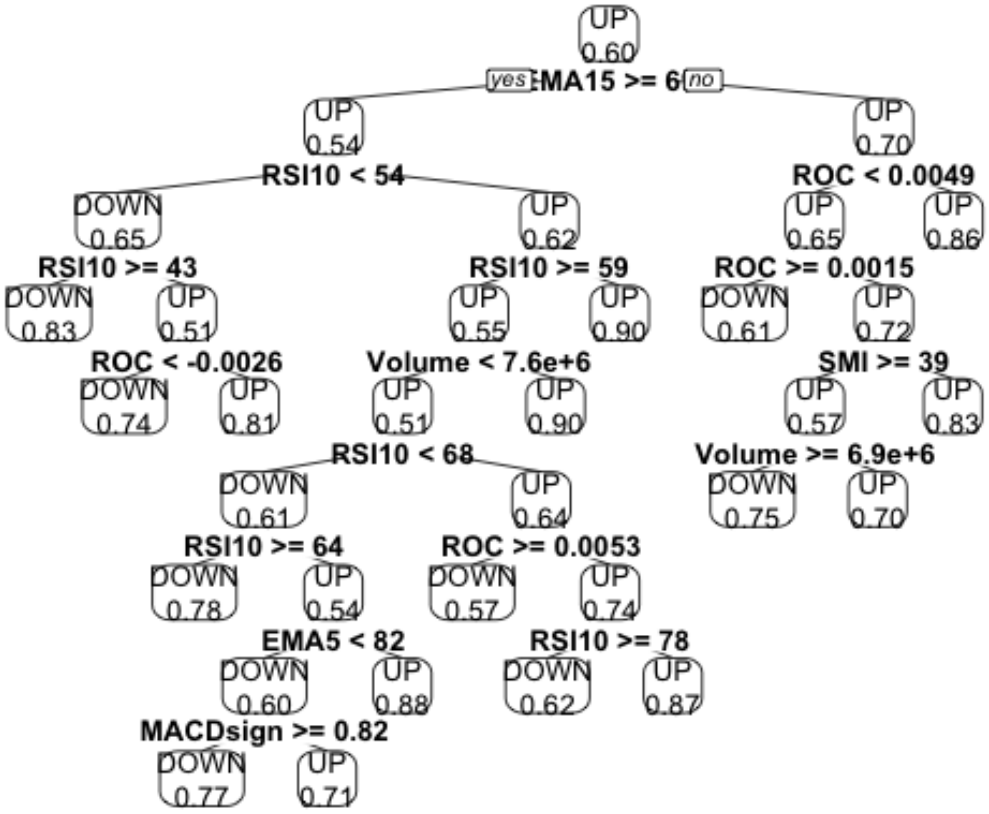


Figure 1: Decision tree

## 3. Prune the tree

```
printcp(DecisionTree)
#shows the minimal cp for each trees of each size.
plotcp(DecisionTree,upper="splits")
#plots the average geometric mean for trees of each size.

PrunedDecisionTree <- prune(DecisionTree, cp = 0.039683)
prp(PrunedDecisionTree, type=2, extra=8)
```

```
Classification tree:
rpart(formula = Class ~ MACDsignal + SMI + EMA5 + EMA15 + RSI3 +
    RSI10 + ROC + ADX7 + ADX14 + Volume, data = TrainingSet,
    cp = 0.001)

Variables actually used in tree construction:
[1] EMA15      EMA5       MACDsignal ROC        RSI10      SMI        Volume

Root node error: 126/312 = 0.40385

n= 312

        CP nsplit rel error  xerror     xstd
1 0.075397      0   1.00000 1.00000 0.068785
2 0.039683      2   0.84921 0.91270 0.067629
3 0.029101      4   0.76984 0.99206 0.068695
4 0.023810      7   0.68254 1.07937 0.069515
5 0.015873     10   0.61111 1.10317 0.069676
6 0.001000     16   0.51587 1.14286 0.069886
```

Figure 2: Classification tree

Selecting the complexity parameter (cp) that has the lowest cross-validated error (xerror), which is 0.039683, to prune the tree:

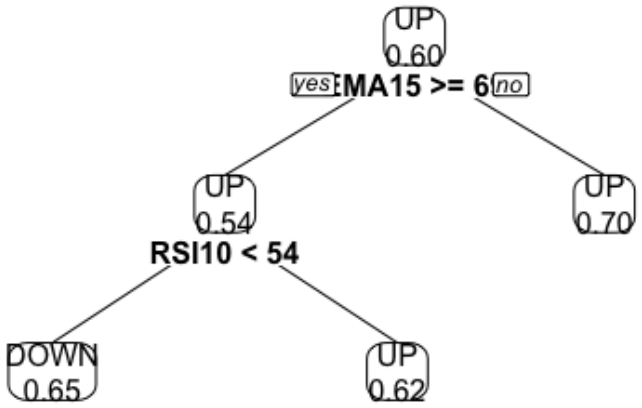


Figure 3: Pruned decision tree

## 4. Confusion matrix

```
table(predict(PrunedDecisionTree,TestSet,type="class"),TestSet[, 11],dnn=list('predicted','actual'
```

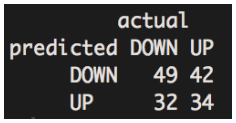


Figure 4: Confusion matrix

The accuracy of prediction is

$$\frac{49+34}{49+42+32+34} = 0.5286624$$

## 5. Alternative models

Considering logistic regression, linear discriminant analysis and quadratic discriminant analysis used in HW2, we get predictions as follows:

```
# Logistic Regression
glm.fit <- glm(Class~MACDsignal + SMI + EMA5 + EMA15 + RSI3 +  RSI10 +
                   ROC + ADX7 + ADX14 + Volume, data=TrainingSet, family=binomial)

glm.probs <- predict(glm.fit, TestSet, type ="response")

glm.pred=rep("DOWN", 157)
glm.pred[glm.probs >.5]="UP"

table(glm.pred, Class[313:469,])
mean(glm.pred == Class[313:469,])
# 0.4585987

# Linear Discriminant Analysis
lda.fit <- lda(Class~MACDsignal + SMI + EMA5 + EMA15 + RSI3 +  RSI10 +
                   ROC + ADX7 + ADX14 + Volume, data=TrainingSet)
lda.fit
plot(lda.fit)

lda.pred <- predict(lda.fit, TestSet)
lda.class <- lda.pred$class
table(lda.class, Class[313:469,])
# 0.4585987

# Quadratic Discriminant Analysis
qda.fit <- qda(Class~MACDsignal + SMI + EMA5 + EMA15 + RSI3 +  RSI10 +
                   ROC + ADX7 + ADX14 + Volume, data=TrainingSet)
qda.fit

qda.probs <- predict(qda.fit, TestSet)
qda.class <- qda.probs$class
table(qda.class, Class[313:469,])
```

```
mean(qda.class == Class[313:469,])
# 0.4713376
```

| glm.pred | DOWN | UP |
|---|---|---|
| DOWN | 72 | 83 |
| UP | 2 | 0 |

Figure 5: Logistic regression confusion matrix

The accuracy of prediction of logistic regression is 0.4585987.

| lda.class | DOWN | UP |
|---|---|---|
| DOWN | 72 | 83 |
| UP | 2 | 0 |

Figure 6: Linear discriminant analysis Confusion matrix

The accuracy of prediction of linear discriminant analysis is 0.4585987.

| qda.class | DOWN | UP |
|---|---|---|
| DOWN | 51 | 60 |
| UP | 23 | 23 |

Figure 7: Quadratic discriminant analysis confusion matrix

The accuracy of prediction of quadratic discriminant analysis is 0.4713376.
None of these methods is more efficient than decision tree.