Analyze stock data using machine learning approaches

Predict Direction (up or down)

Xin Guan (github.com/x-guan)

Data

```
rm(list=ls())
setwd("/Users/Guan/Xin/R/Github/ml_stock")

set.seed(443)
stock <- read.csv("stock.csv", header=T)
dim(stock)
```

```
## [1] 1250    9
```

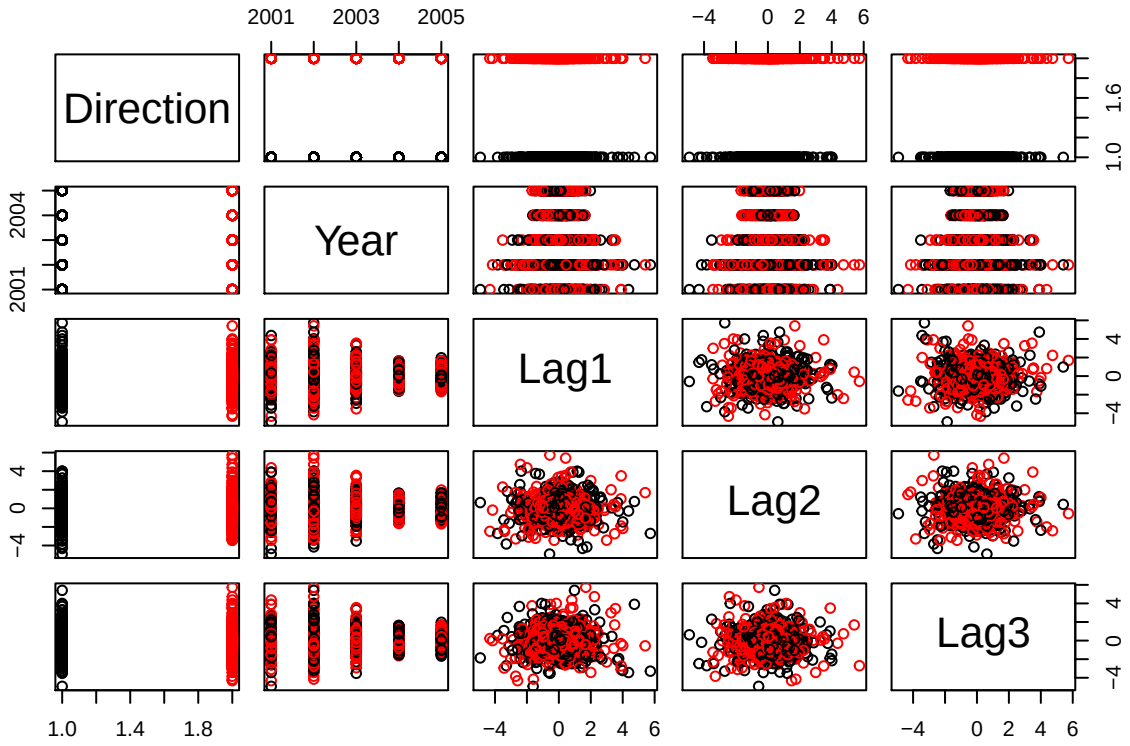
```
head(stock)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5 Volume  Today Direction
## 1 2001  0.381 -0.192 -2.624 -1.055  5.010 1.1913  0.959        Up
## 2 2001  0.959  0.381 -0.192 -2.624 -1.055 1.2965  1.032        Up
## 3 2001  1.032  0.959  0.381 -0.192 -2.624 1.4112 -0.623      Down
## 4 2001 -0.623  1.032  0.959  0.381 -0.192 1.2760  0.614        Up
## 5 2001  0.614 -0.623  1.032  0.959  0.381 1.2057  0.213        Up
## 6 2001  0.213  0.614 -0.623  1.032  0.959 1.3491  1.392        Up
```

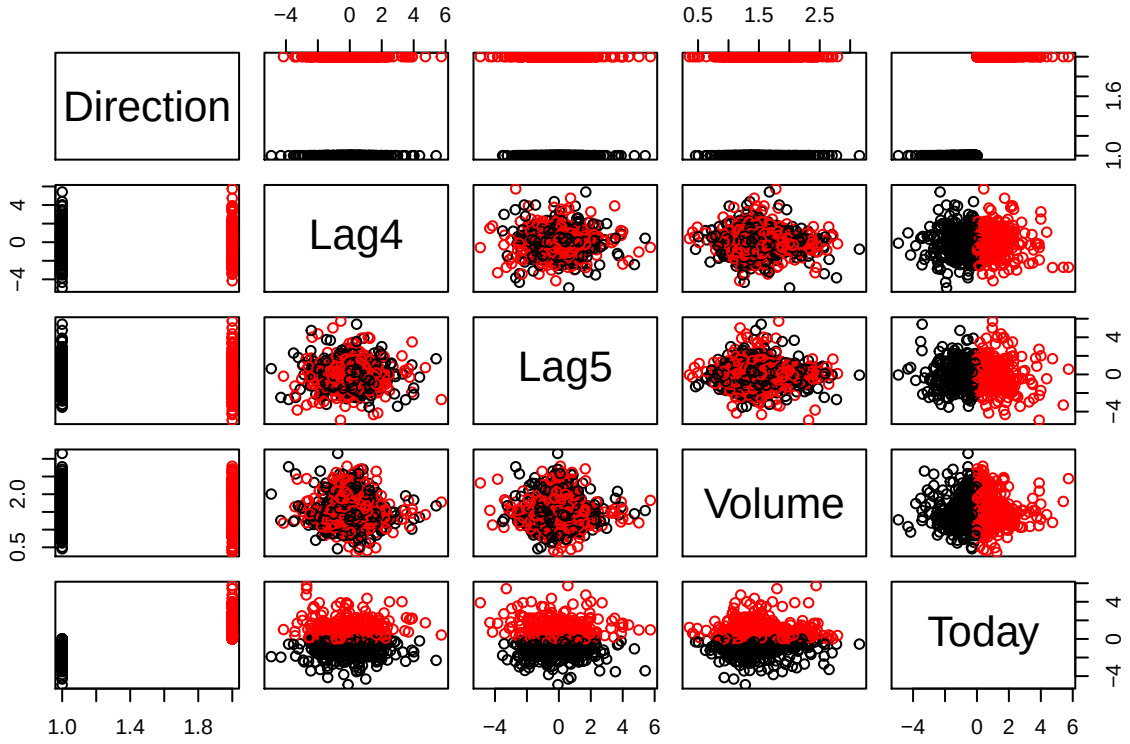
```
# summary
summary(stock)
```

```
##       Year           Lag1                Lag2          
##  Min.   :2001   Min.   :-4.922000   Min.   :-4.922000  
##  1st Qu.:2002   1st Qu.:-0.639500   1st Qu.:-0.639500  
##  Median :2003   Median : 0.039000   Median : 0.039000  
##  Mean   :2003   Mean   : 0.003834   Mean   : 0.003919  
##  3rd Qu.:2004   3rd Qu.: 0.596750   3rd Qu.: 0.596750  
##  Max.   :2005   Max.   : 5.733000   Max.   : 5.733000  
##       Lag3                Lag4                Lag5          
##  Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.92200  
##  1st Qu.:-0.640000   1st Qu.:-0.640000   1st Qu.:-0.64000  
##  Median : 0.038500   Median : 0.038500   Median : 0.03850  
##  Mean   : 0.001716   Mean   : 0.001636   Mean   : 0.00561  
##  3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.59700  
##  Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.73300  
##      Volume            Today           Direction  
##  Min.   :0.3561   Min.   :-4.922000   Down:602  
##  1st Qu.:1.2574   1st Qu.:-0.639500   Up  :648  
##  Median :1.4229   Median : 0.038500             
##  Mean   :1.4783   Mean   : 0.003138             
##  3rd Qu.:1.6417   3rd Qu.: 0.596750             
##  Max.   :3.1525   Max.   : 5.733000
```

```
# check linearity
pairs(stock[,c(9,1:4)], col=stock$Direction)
```




```
pairs(stock[,c(9,5:8)], col=stock$Direction)
```




```
# select training set
train <- stock$Year<2005
```

GLM

```
# fit the glm model
fit_glm <- glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=stock,
               family=binomial, subset=train)
fit_glm
```

```
##
## Call:  glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = stock, subset = train)
##
## Coefficients:
## (Intercept)         Lag1         Lag2         Lag3         Lag4
##    0.191213    -0.054178    -0.045805     0.007200     0.006441
##        Lag5       Volume
##   -0.004223    -0.116257
##
## Degrees of Freedom: 997 Total (i.e. Null);  991 Residual
## Null Deviance:       1383
## Residual Deviance: 1381  AIC: 1395
```

```
# predict
prob_glm <- predict(fit_glm, newdata=stock[!train,], type="response")
prob_glm[1:5]
```

```
##       999      1000      1001      1002      1003
## 0.5282195 0.5156688 0.5226521 0.5138543 0.4983345
```

```
pred_glm <- ifelse(prob_glm>0.5,"Up","Down")
head(pred_glm)
```

```
##    999   1000   1001   1002   1003   1004
##   "Up"   "Up"   "Up"   "Up" "Down"   "Up"
```

```
# resutls
table(pred_glm, stock$Direction[!train])
```

```
##
## pred_glm Down Up
##     Down   77 97
##     Up     34 44
```

```
mean(pred_glm==stock$Direction[!train])
```

```
## [1] 0.4801587
```

Linear discriminant analysis

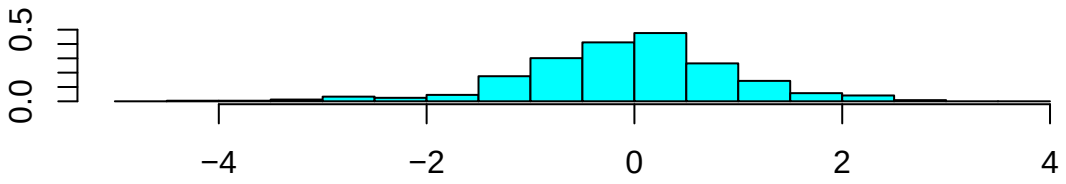
```
library(MASS)

# fit a lda model
fit_lda <- lda(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=stock, subset=train)
fit_lda
```

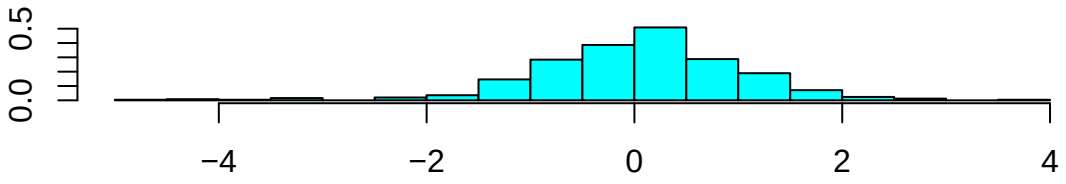
```
## Call:
## lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = stock,
```

```
##     subset = train)
##
## Prior probabilities of groups:
##     Down       Up
## 0.491984 0.508016
##
## Group means:
##             Lag1        Lag2         Lag3         Lag4          Lag5
## Down  0.04279022  0.03389409 -0.009806517 -0.010598778  0.0043665988
## Up   -0.03954635 -0.03132544  0.005834320  0.003110454 -0.0006508876
##        Volume
## Down 1.371843
## Up   1.363210
##
## Coefficients of linear discriminants:
##                LD1
## Lag1   -0.58081056
## Lag2   -0.49111007
## Lag3    0.07707664
## Lag4    0.06904095
## Lag5   -0.04549853
## Volume -1.24678716
```

```
plot(fit_lda)
```



group Down



group Up

```
# predict
pred_lda <- predict(fit_lda, stock[!train,])
pred_lda <- as.data.frame(pred_lda)
head(pred_lda)
```

```
##      class posterior.Down posterior.Up        LD1
## 999     Up      0.4718257    0.5281743  0.8675751
## 1000    Up      0.4843519    0.5156481  0.3277204
```

```
## 1001    Up      0.4773870    0.5226130  0.6277467
## 1002    Up      0.4861559    0.5138441  0.2500574
## 1003  Down      0.5016259    0.4983741 -0.4155006
## 1004    Up      0.4988759    0.5011241 -0.2972177
```

```
# results
table(pred_lda$class, stock$Direction[!train])
```

```
##
##        Down Up
##   Down   77 97
##   Up     34 44
```

```
mean(pred_lda$class==stock$Direction[!train])
```

```
## [1] 0.4801587
```

K-nearest neighbors

```
library(class)

# fit the knn model
Xlag <- with(stock, cbind(Lag1, Lag2, Lag3, Lag4, Lag5, Volume))
perd_knn <- knn(Xlag[train,], Xlag[!train,], stock$Direction[train], k=1)    # k=1 ???

# results
table(perd_knn, stock$Direction[!train])
```

```
##
## perd_knn Down Up
##     Down   50 62
##     Up     61 79
```

```
mean(perd_knn==stock$Direction[!train])
```

```
## [1] 0.5119048
```

Random forest

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
# fit a random forest model
fit_forest <- randomForest(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
                           data=stock, importance=T, subset=train, mtry=4, ntree=1000)

# select variables ???
importance(fit_forest)
```

```
##              Down         Up MeanDecreaseAccuracy MeanDecreaseGini
## Lag1    0.6432854  0.8640336            1.0858712         87.71763
## Lag2   -2.7463633  5.0202885            1.4895598         80.94621
```

```
## Lag3   -4.4924580  3.2757589           -0.6409602         81.08298
## Lag4   -3.3356269 -1.7554056           -3.4860262         81.11855
## Lag5   -1.6951240 -1.2702636           -2.1710339         82.61853
## Volume -0.4765504 -2.7619688           -2.3484984         84.88452
```

```
# predict
pred_forest <- predict(fit_forest, stock[!train,])

# results
table(pred_forest, stock$Direction[!train])
```

```
##
## pred_forest Down Up
##        Down   50 69
##        Up     61 72
```

```
mean(pred_forest==stock$Direction[!train])
```

```
## [1] 0.484127
```

Boosting

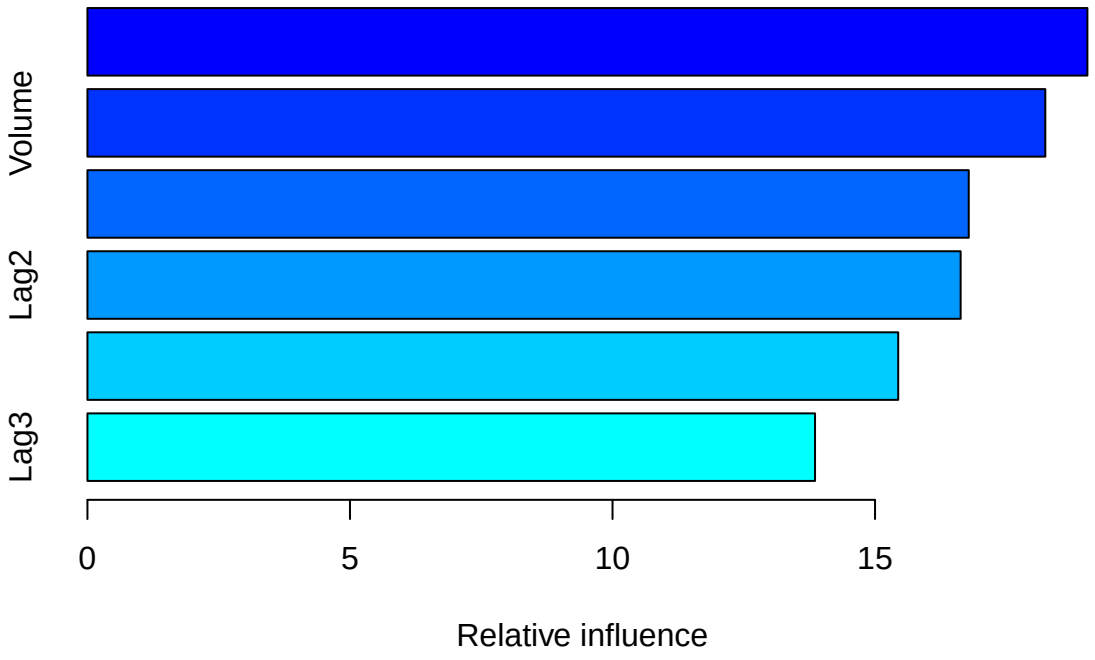
```
library(gbm)
```

```
## Loaded gbm 2.1.4
```

```
# fit a boosting model
fit_boost <- gbm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=stock[train,],
                 distribution="multinomial", n.trees=1000, shrinkage=0.01, interaction.depth=4)
fit_boost
```

```
## gbm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, distribution = "multinomial", data = stock[train,
##     ], n.trees = 1000, interaction.depth = 4, shrinkage = 0.01)
## A gradient boosted model with multinomial loss function.
## 1000 iterations were performed.
## There were 6 predictors of which 6 had non-zero influence.
```

```
# select variables
summary(fit_boost)
```




```
##           var  rel.inf
## Lag1     Lag1 19.04414
## Volume Volume 18.24257
## Lag4     Lag4 16.78460
## Lag2     Lag2 16.63036
## Lag5     Lag5 15.44133
## Lag3     Lag3 13.85700
```

```
# predict
prob_boost <- predict(fit_boost, stock[!train,], n.trees=1000, type="response")
prob_boost[1:5]
```

```
## [1] 0.2928150 0.3617565 0.2710510 0.3423024 0.3792272
```

```
pred_boost <- ifelse(prob_boost[,2,]>0.5, "Up", "Down")
head(pred_boost)
```

```
## [1] "Up" "Up" "Up" "Up" "Up" "Up"
```

```
# results
table(pred_boost, stock$Direction[!train])
```

```
##
## pred_boost Down Up
##       Down   59 85
##       Up     52 56
```

```
mean(pred_boost==stock$Direction[!train])
```

```
## [1] 0.4563492
```

SVM

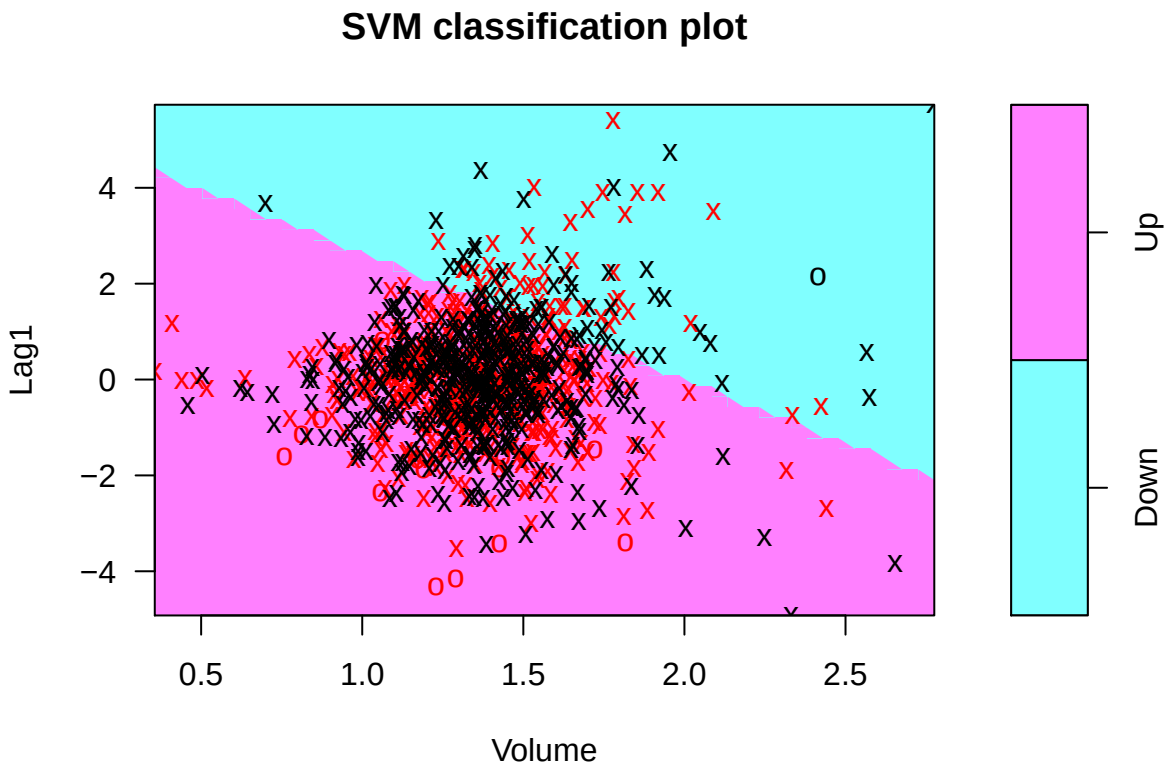
```
library(e1071)
```

```r
# fit the svm model
fit_svm <- svm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=stock,
               subset=train, kernel="linear", cost=10, scale=F)   # linear
# fit_svm <- svm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=stock,
#                subset=train, kernel="radial", cost=10, scale=F)   # non-linear

fit_svm
```

```
##
## Call:
## svm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, data = stock, kernel = "linear", cost = 10, subset = train,
##     scale = F)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##       gamma:  0.1666667
##
## Number of Support Vectors:  980
```

```r
plot(fit_svm, stock[train,], Lag1~Volume)
```




```r
# predict
pred_svm <- predict(fit_svm, stock[!train,])

# results
table(pred_svm, stock$Direction[!train])
```

```
##
## pred_svm Down Up
##     Down   43 47
##     Up     68 94
```

```
mean(pred_svm==stock$Direction[!train])
```

```
## [1] 0.5436508
```