Assignment35

John Bute

2024-11-17

R Markdown

Apply boosting, bagging, and random forests to the Weekly dataset. Run each algorithm 100 times to average out variability in performance.

Load Data and Libraries

```
Weekly <- read.csv("C:/Users/johnb/Desktop/Machine Learning/data/Weekly.csv")
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.4.2

##
## Attaching package: 'ISLR'

## The following object is masked _by_ '.GlobalEnv':
##
##     Weekly

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.4.2

## randomForest 4.7-1.2

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

library(gbm)

## Warning: package 'gbm' was built under R version 4.4.2

## Loaded gbm 2.2.2

## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.com/gbm-
developers/gbm3

library(caret)

## Warning: package 'caret' was built under R version 4.4.2

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin

## Loading required package: lattice

set.seed(1)
index <- sample(1:nrow(Weekly), 0.7 * nrow(Weekly))
train_data <- Weekly[index, ]
test_data <- Weekly[-index, ]
predictors <- c("Lag1", "Lag2", "Lag3", "Lag4", "Lag5", "Volume")

train_data$Direction <- ifelse(train_data$Direction == "Up", 1, 0)
test_data$Direction <- ifelse(test_data$Direction == "Up", 1, 0)
```

```r
logistic_errors <- numeric(100)

for (i in 1:100) {
  set.seed(i)
  logistic <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                  data = train_data, family = "binomial")
  log_probs <- predict(logistic, newdata = test_data, type = "response")
  log_preds <- ifelse(log_probs > 0.5, 1, 0)
  logistic_errors[i] <- mean(log_preds != test_data$Direction)
}

logistic_avg_error <- mean(logistic_errors)
cat("Average Logistic Regression Error Rate:", logistic_avg_error, "\n")

## Average Logistic Regression Error Rate: 0.4159021
```

On Average, our model makes an error 41% of the time, which might be acceptable, considering that the top traders usually aim for a precision rate of about 60% success rate.

```r
bagging_errors <- numeric(100)

for (i in 1:100) {
  set.seed(i)
  bagging <- randomForest(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                          data = train_data, mtry = length(predictors))
  bagging_preds <- predict(bagging, newdata = test_data)
  bagging_preds <- ifelse(bagging_preds > 0.5, 1, 0)
  bagging_errors[i] <- mean(bagging_preds != test_data$Direction)
}

bagging_avg_error <- mean(bagging_errors)
cat("Average Bagging Error Rate:", bagging_avg_error, "\n")

## Average Bagging Error Rate: 0.4608869
```

It seems that we did worse, with our model being erronuous with a 46% error rate. We were probably better off with our logisitc regression model.

```r
rf_errors <- numeric(100)

for (i in 1:100) {
  set.seed(i)
  rf <- randomForest(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                     data = train_data, mtry = 3)
  rf_preds <- predict(rf, newdata = test_data)
  rf_preds <- ifelse(rf_preds > 0.5, 1, 0)
  rf_errors[i] <- mean(rf_preds != test_data$Direction)
}

rf_avg_error <- mean(rf_errors)
cat("Average Random Forest Error Rate:", rf_avg_error, "\n")

## Average Random Forest Error Rate: 0.464893
```

Our Random Forest model performed similarly to our bagging model.

```r
boosting_errors <- numeric(100)

for (i in 1:100) {
  set.seed(i)
```

```
  boosting <- gbm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                  data = train_data, distribution = "bernoulli", n.trees =
1000,
                  interaction.depth = 4, shrinkage = 0.01, verbose = FALSE)
  boost_probs <- predict(boosting, newdata = test_data, n.trees = 1000, type
= "response")
  boost_preds <- ifelse(boost_probs > 0.5, 1, 0)
  boosting_errors[i] <- mean(boost_preds != test_data$Direction)
}

boosting_avg_error <- mean(boosting_errors)
cat("Average Boosting Error Rate:", boosting_avg_error, "\n")

## Average Boosting Error Rate: 0.4794801
```

Even Worse :(

```
results <- data.frame(
  Model = c("Logistic Regression", "Bagging", "Random Forest", "Boosting"),
  Average_Error_Rate = c(logistic_avg_error, bagging_avg_error, rf_avg_error,
boosting_avg_error)
)

print(results)

##                 Model Average_Error_Rate
## 1 Logistic Regression          0.4159021
## 2             Bagging          0.4608869
## 3       Random Forest          0.4648930
## 4            Boosting          0.4794801
```

From the results, we can see that the logistic regression model performed the best out of all our options. However, I believe that no model had a good enough error rate to make me comfortable using it. The fact that there are super computers out there trying to find the smallest margins to make money off the market tells me that a simple model that I can program on my computer won't make me a millionaire any time soon.