Matthew Forey
April 2020
University of Texas at San Antonio

Predicting Stock Prices & Assessing Risk with CAPM

Executive Summary

To accurately predict future changes in stock prices and limit the amount of observed variance in prices, it is important to understand the extent to which certain market forces play in price evolution. Price changes in the stock market are dynamic and very difficult to predict, and they are reflective of both systemic and unsystemic risk. Systemic risks are outside forces that can't be diversified away. These include recessions, global catastrophes, or changing interest rates. Unsystemic risks are specific to each individual stock and can be diversified away with strategic investing.

The unpredictable nature of these events can yield the potential of high amounts of error in forecasts over long periods of time; therefore in this paper the goal is to test the accuracy of predicted one-week price forecasts in the market from 2011. This study found that support vector machine models will yield the most accurate one-week prediction of stock price change, followed closely by decision tree classification.

Problem

The Dow Jones Industrial Average (DJI) is a stock market index that measures stock performances of 30 public U.S. companies. It is often used as a representative sample size of the overall U.S. market because the Dow includes the prices of the 30 largest companies by capitalization in its overall value. At the end of each trading day, the closing price of each individual company is listed, and the value of the Dow is the compilation of the end-of-day closing price for each of the 30 companies.

Investing in the stock market can be a challenge because of the extent to which levels of uncertainty affect the market. The goal of this case study is to use predictive modeling to predict stock prices for the upcoming week using the previous week's data. We also want to evaluate risk levels of these same stocks. The models that will be used for testing are a simple linear model, a decision tree model, and a support vector machine model. Lastly, the level of risk for each company will be determined using a capital asset pricing model (CAPM) to describe the relationship the observed returns have with unplanned systemic risk.

Related Literature

Fama and Malkiel [1] hypothesized that all stock share price variations are driven by all publicly available information, including past information. Called the efficient market hypothesis, this theory would mean it is impossible to beat the market, much less predict exactly what will happen in the future. Still, many papers have been written that have discussed various measures of quantifying uncertainty and applied these measures to market forecasting.

One paper proposed an index developed by Baker et al. [2] which searched economic policy uncertainty based on the frequency of terms in newspaper coverage related to economic uncertainty. Rather than selecting individual companies from the same index to predict prices, Huarng et al. [3] used data from the Dow Jones and NASDAQ indices to create their own forecast of the Taiwan stock index using a multi-variable approach to improve their forecast.

Methodology

The data gathered for this case study comes from 750 weekly observations in the Dow Jones stock values of 16 variable measurements or classifications (as retrieved from research by Brown, Pelosi, and Dirska) [4]. The values represented in this set come from the stock values during the first and second financial quarters of 2011 (January - March and April – June, respectively). In addition to information about stock price for the week (open, high, low, close, trade volume), information was included for the percent change in price throughout the week, percent change in trade volume for the week, last week's trade volume, and the next week's open, close, and percent change in price. This percent change in price for the upcoming week is our dependent variable, and thus we want to find models that will maximize this value. No sampling techniques were used from the dataset.

Additional lag variables were created and added to the model to predict weekly changes in next week's price, which will be further explained in the next section. The training data for the three models used all 24 variables to predict the percent changes in next week's price. All 20 variables were fit in the linear relationship assumption, as all variables were either numeric price and volume values, or deltas (other than the company name values which were factors). The decision tree was split and run on the training data, with an optimal size tree of 21 subtrees. The support vector machine model included such parameters as cost of 0.5, gamma of 0.06, and 192 support vectors.

Data

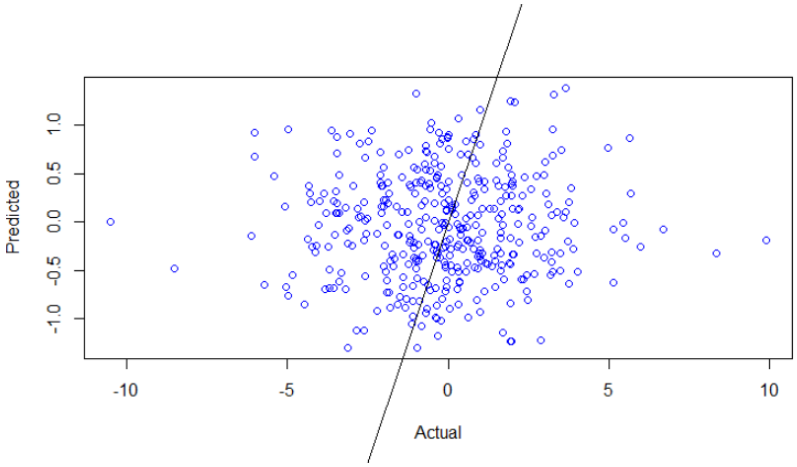
As previously mentioned, additional lag variables were created to predict weekly changes in next week's price using the values of percent change of this week's price. The functions 'mutate' and 'lag' were used to create four lag variables each for price change and volume change, with an 'n' lag of 1,2,3, and 4 weeks. No transformations needed to be applied and the values were all normally distributed, so the data was finally partitioned by first quarter values and second quarter values of the total 720 observations (after 30 NA values were removed from the original 750 rows). Checking for multicollinearity showed that the price values for the week (open, close, high, low) were heavily correlated with each other and next week's open/close/high/low, but that is expected because the average week should not see a huge fluctuation in price change.

Findings

Each of the three models were fit on data from quarter 1, and a test error rate was calculated from predictions made on the test data from quarter 2. The linear model found that a lag closing price of 3 weeks was the most statistically significant prediction variable. The model included an R-squared measurement of .37, but because so many outside forces can impact changes in stock market data this value can be acceptable in this linear model. The linear regression model had a mean square error of 11.40 difference between the prediction and test value.

The decision tree model was constructed using K-fold cross-validation to find alpha and the best sequence of subtrees. 10 significant variables out of the 24 were used in the tree construction, and the number of terminal nodes was 21 subtrees, as seen in Figure 1 in the appendix. The decision tree model had a mean square error of 7.69 difference between the prediction and test value.

The support vector machine had the best mean square error rate of 6.27, as depicted in the Figure 2 plot below. Again, because of the uncertain nature of the stock market moves, a mean-squared error rate of 6.27 could certainly be sufficiently used for monetary gain in predicting weekly price changes and for leveraging risk in stock trading.



Calculating the risk levels of each company as compared to the Dow Jones using CAPM showed that most of the companies have a Beta value of less than 1 (as seen in Figure 3 in the appendix for the first 5 companies in the Dow Jones). This means adding these companies to a portfolio will reduce the overall risk because these companies are much safer than the Dow Jones market. This makes sense because these companies are large enough where any systematic risk can be withstood better than much smaller companies.

Conclusions

The goal of this analysis was conducted to build and predict models for stock price changes in the span of one week. Support vector machine models are the best performing of the 3 models tested, having the lowest prediction error rate. I was not able to find much information about the week of May 27, 2011 and why that week may have stood out. Some research seems to indicate that that time period began a period of a slight bear market (peaking in mid-May and then declining for many months) until October of that year. In future analysis, many of the stocks could be grouped together who performed similarly in their price changes, so that their deltas are grouped and better classified for analysis using additional modeling techniques.

Citations

[1] Fama E. (1970) « Efficient Capital Markets: a Review of Theory and Empirical Work » *Journal of Finance*, Vol.25, No.2.pp. 383-417.

[2] Baker, S. R., Bloom, N., & Davis, S. J. (2016). Measuring economic policy uncertainty. *The quarterly journal of economics, 131*(4), 1593-1636.

[13] K.H. Huarng, H.K. Yu, Y.W. Hsu. A multivariate heuristic model for fuzzy. IEEE Trans. Syst. Man Cybern. B: Cybern., 37 (4) (2007), pp. 836-846

[4] Brown, M. S., Pelosi, M. & Dirska, H. (2013). Dynamic-radius Species-conserving Genetic Algorithm for the Financial Forecasting of Dow Jones Index Stocks. Machine Learning and Data Mining in Pattern Recognition, 7988, 27-41.

Appendix

Figure 1

Decision Tree

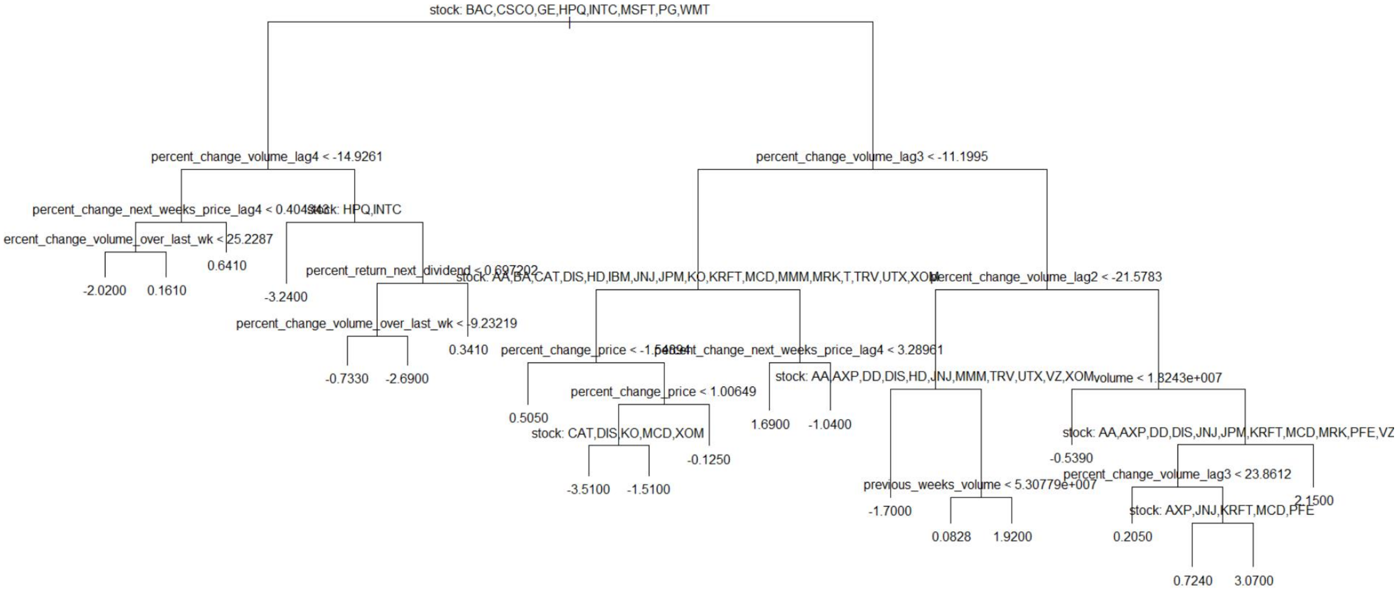

stock: BAC,CSCO,GE,HPQ,INTC,MSFT,PG,WMT
percent_change_volume_lag4 < -14.9261
percent_change_volume_lag3 < -11.1995
percent_change_next_weeks_price_lag4 < 0.404843
stock: HPQ,INTC
ercent_change_volume_over_last_wk < 25.2287
0.6410
-2.0200
0.1610
-3.2400
percent_return_next_dividend < 0.697202
stock: AA,BA,CAT,DIS,HD,IBM,JNJ,JPM,KO,KRFT,MCD,MMM,MRK,T,TRV,UTX,XOM
percent_change_volume_lag2 < -21.5783
percent_change_volume_over_last_wk < -9.23219
0.3410
-0.7330
-2.6900
percent_change_price < -1.54894
percent_change_next_weeks_price_lag4 < 3.28961
stock: AA,AXP,DD,DIS,HD,JNJ,MMM,TRV,UTX,VZ,XOM
volume < 1.8243e+007
percent_change_price < 1.00649
0.5050
stock: CAT,DIS,KO,MCD,XOM
-0.1250
-3.5100
-1.5100
1.6900
-1.0400
stock: AA,AXP,DD,DIS,JNJ,JPM,KRFT,MCD,MRK,PFE,VZ
-0.5390
percent_change_volume_lag3 < 23.8612
previous_weeks_volume < 5.30779e+007
-1.7000
0.0828
1.9200
2.1500
stock: AXP,JNJ,KRFT,MCD,PFE
0.2050
0.7240
3.0700

Figure 2

SVM Mean Squared Error

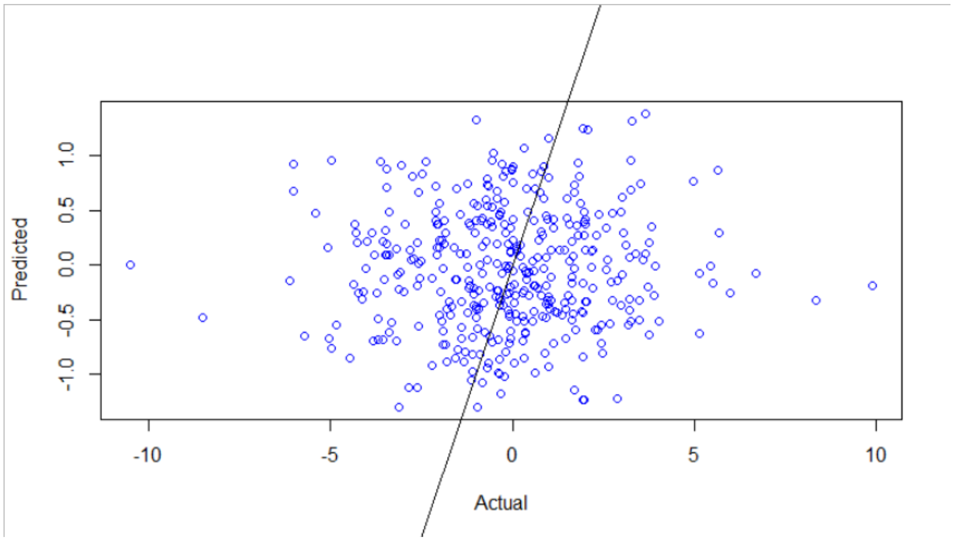
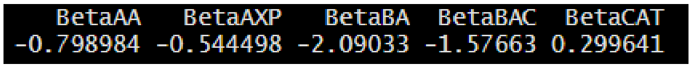


Figure 3

Risk Levels (beta) of Companies AA, AXP, BA, BAC, CAT



```
   BetaAA    BetaAXP    BetaBA  BetaBAC  BetaCAT
-0.798984 -0.544498 -2.09033 -1.57663 0.299641
```

R Code:

```
library(tseries)
library(quantmod)
library(libridate)
library(dplyr)
library(MASS)
library(tree)
library(quantmod)
library(tseries)
library(e1071)

# read data
DowData <- read.delim("C:/Users/matt4/Downloads/dow_jones_index.data", header=TRUE, sep=",")
NamesData <- read.delim("C:/Users/matt4/Downloads/dow_jones_index.names", header=TRUE, sep=",")



# Remove any missing values if needed
DowData = na.omit(DowData)
#went from 750 obs to 720

# See what the data classes are
str(DowData)

DowData <- as.data.frame(DowData)

#DowData$date <- mdy(as.character(DowData$date))

# Change columns 4-7 to numeric values keeping decimals
```

```
DowData$open <- as.character(gsub("\\$", "", DowData$open))
DowData$open <- as.numeric(DowData$open, options(digits = 6))
DowData$high <- as.character(gsub("\\$", "", DowData$high))
DowData$high <- as.numeric(DowData$high, options(digits = 6))
DowData$low <- as.character(gsub("\\$", "", DowData$low))
DowData$low <- as.numeric(DowData$low, options(digits = 6))
DowData$close <- as.character(gsub("\\$", "", DowData$close))
DowData$close <- as.numeric(DowData$close, options(digits = 6))



# Change columns 12 & 13 to numerics
DowData$next_weeks_open <- as.character(gsub("\\$", "", DowData$next_weeks_open))
DowData$next_weeks_open <- as.numeric(DowData$next_weeks_open, options(digits = 6))

DowData$next_weeks_close <- as.character(gsub("\\$", "", DowData$next_weeks_open))
DowData$next_weeks_close <- as.numeric(DowData$next_weeks_close, options(digits = 6))

#Get lags for 1 week biweekly and monthly



# library(dplyr)
# DowData <-
#   DowData %>%
#   group_by(PreviousClose) %>%
#   mutate(lag.value = dplyr::lag(value, n = 1, default = NA))

DowData <- DowData %>%
  group_by(stock) %>%
  mutate(percent_change_next_weeks_price_lag1 = lag(percent_change_price, n = 1, default = NA))
```

```
DowData <- DowData %>%
  group_by(stock) %>%
  mutate(percent_change_next_weeks_price_lag2 = lag(percent_change_price, n = 2,
default = NA))


DowData <- DowData %>%
  group_by(stock) %>%
  mutate(percent_change_next_weeks_price_lag3 = lag(percent_change_price, n = 3,
default = NA))


DowData <- DowData %>%
  group_by(stock) %>%
  mutate(percent_change_next_weeks_price_lag4 = lag(percent_change_price, n = 4,
default = NA))


DowData <- DowData %>%
  group_by(stock) %>%
  mutate(percent_change_volume_lag1 = lag(percent_change_volume_over_last_wk, n = 1,
default = NA))


DowData <- DowData %>%
  group_by(stock) %>%
  mutate(percent_change_volume_lag2 = lag(percent_change_volume_over_last_wk, n = 2,
default = NA))


DowData <- DowData %>%
  group_by(stock) %>%
  mutate(percent_change_volume_lag3 = lag(percent_change_volume_over_last_wk, n = 3,
default = NA))


DowData <- DowData %>%
  group_by(stock) %>%
  mutate(percent_change_volume_lag4 = lag(percent_change_volume_over_last_wk, n = 4,
default = NA))



# Check variable correlation
corr_data <- DowData[,4:16]
corr_matrix <- cor(corr_data)
corrplot(corr_matrix, type = "upper")




# Split into training and testing
train <- which(DowData$quarter == 1)
dataTrain <- DowData[train,]
dataTest <- DowData[-train,]



# Set the formula
formula <- (percent_change_next_weeks_price ~ stock + open + close + high + low +
volume + percent_change_price +
            percent_change_volume_over_last_wk + previous_weeks_volume +
days_to_next_dividend + percent_return_next_dividend +
            percent_change_next_weeks_price_lag1 + percent_change_next_weeks_price_lag2
+ percent_change_next_weeks_price_lag3 +
            percent_change_next_weeks_price_lag4 + percent_change_volume_lag1 +
percent_change_volume_lag2 +
            percent_change_volume_lag3 + percent_change_volume_lag4)

# Linear Model
lmModel <- lm(formula, data = dataTrain)
```

```
lmModel
summary(lmModel)

# Find significant predictors
lmstepData = na.omit(dataTrain)
steplmModel <- stepAIC(lm(formula, data = lmstepData), direction = "both")

steplmModel
summary(steplmModel)

predLM = predict(lmModel, dataTest)

plot(dataTest$percent_change_price, predLM, main = "Linear Model Actual vs Predicted",
    xlab = "Actual", ylab = "Predicted", col = 4)
abline(0,1)

mean((predLM - dataTest$percent_change_price)^2)



# Decision Tree
treeModel <- tree(formula,
             data = dataTrain, subset = train)

crossValTree = cv.tree(treeModel)
#best size = 21
optimalSize = which.min(crossValTree$size)

treeModel = prune.tree(treeModel, best = optimalSize)
```

```
plot(treeModel)

text(treeModel, pretty = 0)



#Predictions on test set

treePreds = predict(treeModel, dataTest)

plot(treePreds, dataTest$percent_change_price, xlab = "Prediction", ylab = "Percent Change in Price")

abline(0,1)


mean((treePreds - dataTest$percent_change_price)^2)





# SVM Model

tuned = tune.svm(formula,

                 data = dataTrain,

                 gamma = seq(0.01, 0.1, by = 0.01),

                 cost = seq(.1, 1, by = .1))


svmModel = svm(formula,

               data = dataTrain,

               gamma = tuned$best.parameters$gamma,

               cost = tuned$best.parameters$cost)


svmModel

summary(svmModel)


#Prediction on test set
```

```
svmpredict = predict(svmModel, dataTest, type = "response")


plot(dataTest$percent_change_price, svmpredict, xlab = "Actual", ylab = "Predicted", col
= 4)

abline(svmpredict)


# error:

mean((svmpredict - dataTest$percent_change_price)^2)








###################### CAPM #################

# Compute the returns and remove any missing values.

ReturnDow = (Delt(DowData[,7]))


DowData$ReturnDow=ReturnDow


DowData = na.omit(DowData)


colnames(DowData$ReturnDow) = "Dow Jones Weekly Returns"

head(DowData$ReturnDow)




#See how the data looks

boxplot(DowData$ReturnDow, main="Expected Return", xlab="Dow Jones",
ylab="Return")
```

```
# Compute mean and stdev for the returns.
DataMean=apply(DowData$ReturnDow, 2, mean)
DataSD=apply(DowData$ReturnDow, 2, sd)
# Take a look at the means and standard deviations.
cbind(DataMean,DataSD)


# According to the CAPM formula, we will first get the beta of each stock by
# regressions



########## Stock = AA

AAdata <- subset(DowData, stock == "AA",
            select=c(stock, close, ReturnDow))

lm.Dow.AA<- lm(AAdata$close
        ~ ReturnDow, data = as.data.frame(AAdata))
summary(lm.Dow.AA)
BetaAA <- summary(lm.Dow.AA)$coefficients[2, 1]
BetaAA

########## Stock = AXP

AXPdata <- subset(DowData, stock == "AXP",
           select=c(stock, close, ReturnDow))

lm.Dow.AXP<- lm(AXPdata$close
            ~ ReturnDow, data = as.data.frame(AXPdata))
summary(lm.Dow.AXP)
BetaAXP <- summary(lm.Dow.AXP)$coefficients[2, 1]
BetaAXP

########## Stock = BA

BAdata <- subset(DowData, stock == "BA",
              select=c(stock, close, ReturnDow))

lm.Dow.BA<- lm(BAdata$close
            ~ ReturnDow, data = as.data.frame(BAdata))
summary(lm.Dow.BA)
BetaBA <- summary(lm.Dow.BA)$coefficients[2, 1]
BetaBA

########## Stock = BAC

BACdata <- subset(DowData, stock == "BAC",
              select=c(stock, close, ReturnDow))

lm.Dow.BAC<- lm(BACdata$close
            ~ ReturnDow, data = as.data.frame(BACdata))
summary(lm.Dow.BAC)
BetaBAC <- summary(lm.Dow.BAC)$coefficients[2, 1]
BetaBAC


########## Stock = CAT
```

```
CATdata <- subset(DowData, stock == "CAT",
               select=c(stock, close, ReturnDow))

lm.Dow.CAT<- lm(CATdata$close
             ~ ReturnDow, data = as.data.frame(CATdata))
summary(lm.Dow.CAT)
BetaCAT <- summary(lm.Dow.CAT)$coefficients[2, 1]
BetaCAT


data.table(BetaAA, BetaAXP, BetaBA, BetaBAC, BetaCAT)
```