Predicting Stock Prices Based on Signed Volume

Hanchao Lei, Ye Lin, Kelsie Lu, Sabrina Peng

1. Background and Motivation

Data-based trading strategy has received lots of attention in recent years due to the development of statistical theories and machine learning techniques. These strategies are built based on publicly available data and aim to capture the signals that are usually hard to detect based only on fundamental analysis. Historic price data is one of the most common features for building a quantitative strategy. Price data is believed to reflect all available information and therefore many technical indicators are developed based on price changes such as momentum, moving-average, etc. However, there are many other features that can be easily accessed and could include additional information for price prediction. Therefore our goal in this report is to study another feature, volume, to see if we can predict future prices based on past volumes.

In order to find a feature or build a model to predict future prices, we first need to answer the fundamental question: what makes prices move? The direct answer would be: the change of supply and demand. In other words, if people want to buy more shares of AAPL, the price would go up, and vice versa. But what would make people want to buy or sell? One of the most important factors is news. People make decisions based on all the available information they obtain. If there's a good news, the price would go up. Therefore predicting future prices based on this mechanism means we need to predict future news, which is equally challenging. However, it is reasonable to assume the effect of the news on price is not momentary, but could last for some period of time. If there's good news for AAPL today, it is reasonable to predict the price going up tomorrow as a continuation of the positive effect from today, even if we cannot predict the news for tomorrow. Therefore future prices could be predicted by estimating the effect from all past news.

In this study, we use daily volume to measure the amount of information during the day. Daily volume is defined as the number of shares being traded during the day. It is a good measure of

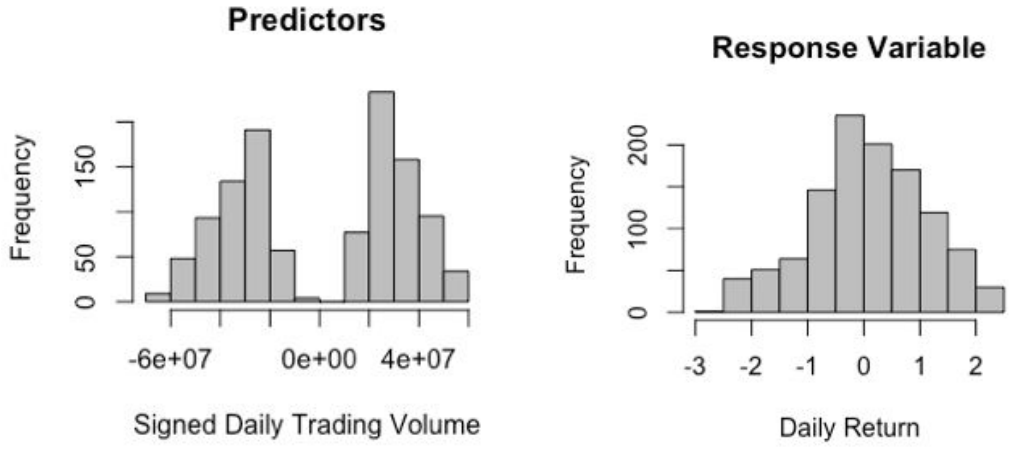
the market volatility. The more information there is, the bigger the news, we should observe a larger volume. To further identify the direction of this daily information (good or bad), we define the signed volume as follows:

$$\textit{Signed volume} = \textit{volume} * \textit{sign(closing price - opening price)}$$

If the price goes up during the day, we assume the direction of overall information/news during the day is positive, and vice versa. The problem of predicting future prices can thus be viewed as a machine learning problem where inputs are past daily signed volume and the outputs are future returns.

2. Data Summary and Illustration

The data used in this study is the daily data of AAPL, including date, open price, closing price and volume for past 1000 days. The response Y is the daily return and the predictors are 20 past signed daily volumes X1, X2, …, X20, where Xi represents the signed daily volume i days before Y. The histograms of the predictors (signed volume) and response variable (daily return) are shown below (only data ranging from the 5th percentile to the 95th percentile for both variables are plotted to get a clear view).



From the histogram on the right, we can see that daily return is clearly skewed to the right, which is expected because of the general upward trend in the stock market over the past few years.

There are two goals in this study. First goal is to identify the effect window of news on prices. For example, if the signed volume during the day is positive, how many days it will keep affecting the future prices to go up/down, which is a variable selection problem. Once we have identified which predictors to include in our model, our second goal is to model and predict Y, the future daily returns. To achieve this goal, we model Y as continuous response (% change) with linear models and as binary response with logistic regression, svm and random forest.

For model training and testing, we use first 700 observations as training set and the rest 300 observations as testing set.

3. Methodology

We first use a linear model to perform variable selection and prediction for the continuous response case. We then use classification methods for the binary response case.

3.1 "Forward" Selection Procedure

To identify the effect window and selector predictors, we develop a new "forward" selection procedure. Unlike the original forward selection where we keep adding predictors into the model based on their importance, we add predictors one by one based on the date. Starting from the model with no predictors, we add X1, X2, …, X20 sequentially and find the best model based on AIC.

3.2 Prediction Performance Evaluation

For the binary cases, we calculate accuracy, sensitivity and specificity from the two way table to evaluate the prediction performance.

For the continuous case, we adopt 3 different measures: MSE, out-of-sample correlation, and sign correlation. Out-of-sample correlation is defined as the correlation between predicted values

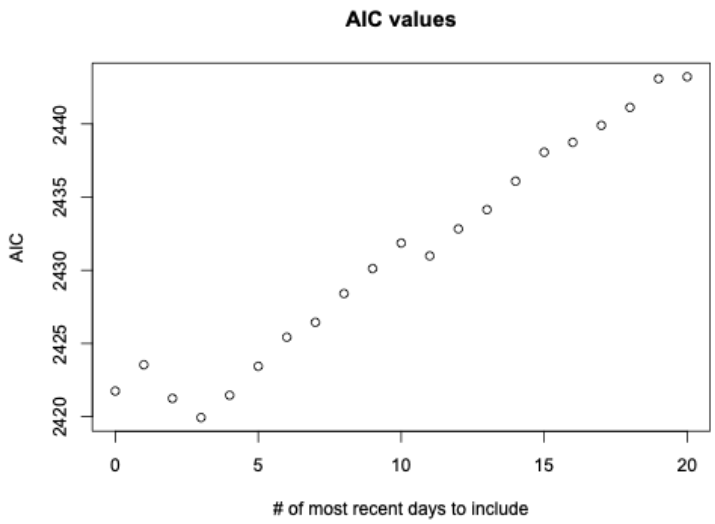
and the true values. Sign correlation is to look at whether the sign of the prediction is correct based on the two-way table.

3.3 Classification Methods

For the classification problems, we apply Logistic Regression, SVM and Random Forest to predict the sign of future return.

4. Results

4.1 Identify the Effect Window through "Forward" Variable Selection



Using the "forward" variable selection, we generate the plot above. The y-axis represents the value of AIC and the x-axis represents the number of most recent days to include in the model. As we can see, the AIC value is the lowest when we include 3 most recent days.

Therefore, we conclude that the best model is Y~X1+X2+X3 with past 3 signed volumes included. It is reasonable to believe that generally the effect window of the daily news is 3 days. In other words, the overall information flowing during the day as a whole will keep affecting the future prices for 3 days. So we only use 3 predictors (volumes for past 3 days) for price prediction in the rest of the project.

4.2 Continuous Response

The model selected using linear model and predictors as above are shown below:

```
Coefficients:
     Estimate Std. Error t value Pr(>|t|)
X1 -6.923e-10  1.431e-09  -0.484   0.6287
X2 -2.767e-09  1.425e-09  -1.942   0.0525 .
X3  2.751e-09  1.428e-09   1.926   0.0545 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.36 on 697 degrees of freedom
Multiple R-squared:  0.01081, Adjusted R-squared:  0.006549
F-statistic: 2.538 on 3 and 697 DF,  p-value: 0.05558
```

In order to evaluate the prediction performance based on the testing set, we calculate MSE (1.94), out-of-sample correlation (-0.03). We also generate a two-way table of sign prediction (shown below). Accuracy, sensitivity and specificity are calculated.

	Prediction		
True		False	True
	False	**71**	**65**
	True	**73**	**90**

Accuracy	sensitivity	specificity
53.8%	55.2%	52.2%

The results show that although out-of-sample correlation is negative, the model has a relatively robust performance for sign prediction, which is slightly better than a random guess.

4.3 Binary Response

Now we focus on predicting the sign of future return. That is, whether the future price would go up or down. In this case, the response value is binary.

4.3.1 Logistic Regression

Accuracy	sensitivity	specificity
49.5%	49.1%	50.0%

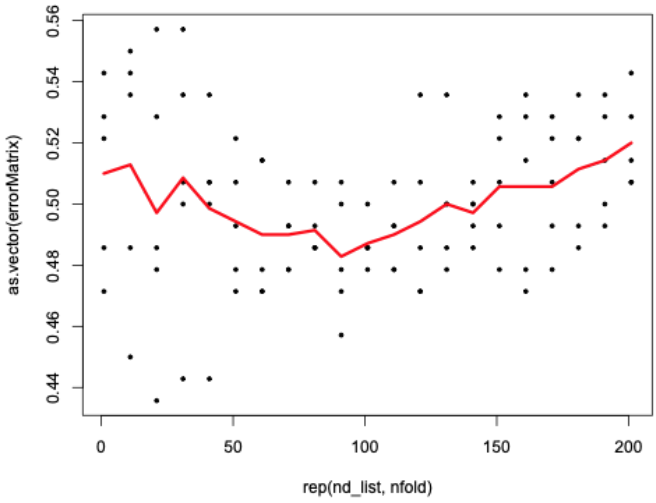
A Logistic Regression Model is fitted on the training set using predictors X1, X2 and X3. Accuracy, sensitivity, and specificity are calculated based on the predictions of the signs using the test data.

4.3.2 SVM

Accuracy	sensitivity	specificity
53.2%	71.8%	30.9%

A SVM model, which uses X1, X2 and X3 as predictors, is fitted on the training set. We calculate Accuracy, sensitivity, and specificity (shown above) to evaluate the predictions on the testing set.

4.3.3 Random Forest



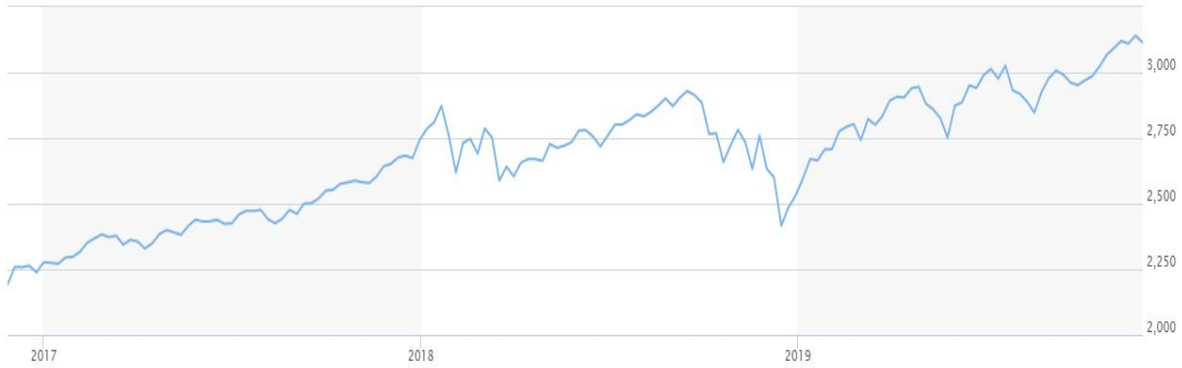
For the parameters of the random forest model, we use ntree, the number of trees, equal to 500 and mtry, the number of variables used at each split, equal to 3 (since we want to include all three predictors X1, X2, and X3 at each split). To select the best node size, a 5-fold Cross Validation is performed on the training set. In the above plot, the y-axis shows the cross validation Mean Squared Error based on the training data while the x-axis represents the node size used in the Random Forest Model. Based on the plot, we can tell that the Mean Squared Error is the lowest when the node size is equal to 91. Therefore, 91 is chosen as the optimal node size for our model.

A Random Forest model with a node size of 91 is fitted on the training data. Accuracy, sensitivity, and specificity, which are calculated based on the predictions on the testing data, are shown below.

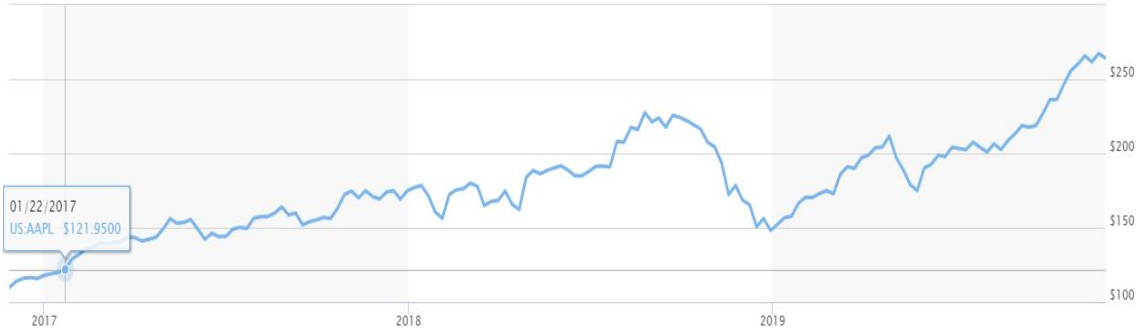
Accuracy	sensitivity	specificity
55.6%	78.2%	27.9%

5. Conclusion and Discussion

Future stock prices are difficult to predict. The testing results from our linear, support vector machine, and random forest models show accuracy slightly above 50%, which means the models perform slightly better than random guess on our test data. This shows that volume has some small predictive power in our case. Comparing the different methods we have tried, the random forest model has the highest accuracy among the three but very unbalanced sensitivity and specificity. This could be attributed to the general upward trend in the stock market in the past three years. To put our results into context, we looked at the prices of S&P 500 index over the past three years:



The above picture shows the macroeconomic trend that U.S. stock market has experienced a bull period in the past three years. We see a similar trend in the price of AAPL (which is also visible in our right-skewed daily return shown in the data illustration section):



Therefore, a limitation of our research is that our model could be predicting future price to go up the majority of the time. That is, the larger than 50% accuracy could mean that the random forest model is only capturing the general upward trend; when it comes to capture news or utilize

information in the trading volume, the model's predictive power is limited. This explains why we see a high sensitivity and a low specificity.

In contrast, our linear model has a more balanced sensitivity and specificity, both slightly above 50%. Our variable selection using linear model also shows a lower AIC with X1, X2, and X3. That being said, this encouraging result might not be very robust since our result is dependent on the data we use and might not hold for other stocks.

For future work, even though "news" is one of our justifications of why volume can be predictive of future returns, there are other measures more directly related to the news, such as some news index developed using text mining. If we can access and utilize those information, we can directly measure the news instead of using volume for rough approximation.

Since our work is more related to the volume-based strategies, another possible direction is to explore further of these volume vs. return related theories. One future direction is to use the original volume (with no signs) for price prediction because studies have shown that the volume usually has a different pattern before prices going up compared to that before prices going down. We did some preliminary analysis of this direction and the performance of the model is similar to the model with signed volume (For random forest, accuracy 51.8%, sensitivity 65.4% and specificity 40.5%). Another more popular and well studied volume-related theory is called Market Impact Models, where it is believed that a large order will have a continuing effect on future prices. For this direction, net volumes (buy orders minus sell orders) were studied and the effect of prices are estimated, which is very similar to our approach. (However, this direction requires access of order book data and is usually studied in financial institutes.)

References:

Almgren, Robert, and Neil Chriss. "Optimal execution of portfolio transactions." Journal of Risk 3 (2001): 5-40.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media, 2009.

Data Sources:

AAPL: Yahoo finance (https://finance.yahoo.com/quote/AAPL/history?p=AAPL)

S&P 500: https://www.marketwatch.com/investing/index/spx/charts

R Code:

```
library(e1071)
library(ipred)
library(rpart)
library(randomForest)

data=read.csv('AAPL.csv')
data=data[,c('Date','Open','Close','Volume')]

price=data$Close
change=(price[-1]-price[-length(price)])/price[-length(price)]
data$return=c(change,NA)*100

## assign direction (+/-) to volume

data$Volume=sign(data$Close-data$Open)*data$Volume

###### format data: 20 past volumes (x1-x20) vs return (y)
```

```
ap=data.frame(matrix(NA,ncol=21,nrow=dim(data)[1]-21))
colnames(ap)[dim(ap)[2]]='y'


for (i in 1:(dim(data)[1]-20)){
        ap[i,'y']=data[i+20,'return']
        ap[i,20:1]=data[i:(i+19),'Volume']
}

##################
################## EDA
##################

vol=data$Volume
hist(vol[vol<quantile(vol, c(0.95)) & vol>quantile(vol,c(0.05))],
    main="Predictors", xlab='Signed Daily Trading Volume',
    col='grey')
ret=na.omit(data$return)
hist(ret[ret<quantile(ret, c(0.95)) & ret>quantile(ret, c(0.05))],
    main='Response Variable', xlab='Daily Return',
    col='grey')

################################################## modeling

##########   get training set and testing set

ap=ap[240:1239,]       # only use most recent 1000 observations
index=1:700
```

```
train.c=ap[index,]     # training set for continuous responses
test.c=ap[-index,]



##############################      continuous response


### variable selection ("forward" selection)

aic=rep(NA,21)
aic[1]=AIC(lm(y~1, data=train.c))
for (i in 1:20){
        aic[i+1]=AIC(lm(y~., data=train.c[,c(1:i,21)]))
}

AIC(lm(y~1, data=train.c))
AIC(lm(y~X1, data=train.c))
AIC(lm(y~X1+X2, data=train.c))
AIC(lm(y~X1+X2+X3, data=train.c))
AIC(lm(y~X1+X2+X3+X4, data=train.c))
AIC(lm(y~X1+X2+X3+X4+X5, data=train.c))
AIC(lm(y~X1+X2+X3+X4+X5+X6, data=train.c))


plot( 0:20,aic, main='AIC values', xlab='# of most recent days to include', ylab='AIC')

# conclusion: past 3 days have predictive power on tomorrow's return
```

```
###### models and predictions


model.lr=lm(y~X1+X2+X3-1, data=train.c)  #no intercept
summary(model.lr)

pred.lr=predict(model.lr, test.c[,1:20])
cor(pred.lr, test.c$y, use='pairwise.complete.obs')

## sign

t=table(test.c$y[-300]>0,pred.lr[-300]>0)

# accuracy
(t[1]+t[4])/sum(t)

#sensitivity
t[4]/(t[2]+t[4])

#specificity
t[1]/(t[1]+t[3])

## MSE

mean(c(pred.lr[-300]-test.c$y[-300])^2)^0.5

######################
###################### binary response
##########################
```

```
train.b=train.c
train.b$y=as.numeric(train.b$y>0)

test.b=test.c
test.b$y=as.numeric(test.b$y>0)


################# logistic

model.log=glm(y~X1+X2+X3-1,data=train.b, family=binomial)
summary(model.log)
pred.log=as.numeric(predict(model.log, test.b[,1:3], type='response')>0.5)
t=table(test.b$y, pred.log)
t



################## svm
model.svm=svm(y~X1+X2+X3-1, data=train.b, type='C-classification')
pred.svm=predict(model.svm, test.b)
t=table(test.b$y, pred.svm)
t


################## random forest

rf.fit = randomForest(train.b[,1:3], as.factor(train.b$y), ntree = 500, mtry = 3, nodesize = 2)
```

```
nfold = 5
infold = sample(rep(1:nfold, length.out=length(train.b$y)))



nd_list=seq(1,201,by=10)

K = length(nd_list)
errorMatrix = matrix(NA, K, nfold)
for (l in 1:nfold)
{
        for (k in 1:K){
                rf.fit = randomForest(train.b[infold!=l,1:3], as.factor(train.b$y[infold!=l]), ntree =
500, mtry = 3, nodesize = nd_list[k])
                pred.rf=predict(rf.fit, train.b[infold==l,1:3])
                errorMatrix[k, l] = mean((as.numeric(pred.rf)-1 - as.numeric(train.b$y)[infold ==
l])^2)
        }
}

plot(rep(nd_list, nfold), as.vector(errorMatrix), pch = 19, cex = 0.5)
points(nd_list, apply(errorMatrix, 1, mean), col = "red", pch = 19, type = "l", lwd = 3)
which.min(apply(errorMatrix, 1, mean))



rf.fit = randomForest(train.b[,1:3], as.factor(train.b$y), ntree = 500, mtry = 3, nodesize = 91)
pred.rf=predict(rf.fit, test.b[,1:3])
t=table(test.b$y, pred.rf)
```