Anna Lu

Jinbo Wu

Yuanrong Li

Yijun Xu

STAT-154, Fall-2018

Final Report

Machine Learning and Apple Stock Return Prediction

Introduction

Apple has itself expanded rapidly in a recent decade and many of us can not live without Apple's products because they are already a part of our daily life. With such a great development, we are also interested in how well does Apple perform in the stock market. It might be more intuitively for us to see the return from the stock price. However, predicting the price of that day will be affected by many factors: financial policies, decisions made by the company, international relations, etc. We have to achieve our purpose from another direction.

Our project is aiming at predicting the return of the Apple Inc,. stock by the means of classifying the return is positive or negative and estimate the return amount based on the 10 years stock data of Apple Inc,.. In our project, we applied multiple statistical models on the dataset: multiple logistic regression, classification tree, linear regression, OLS, splines, ridge regression and LASSO. But only multiple logistic regression, ridge regression and LASSO can be used to achieve our purpose. In the process of multiple logistic regression, we generate the useful predictors for classifying an observation into the group of going up next day or going down next

day and found that only few of the predictors in that dataset were significant. Next, trying to find the plausible pattern in our predicting process and see if the pattern fits. By comparing the final estimated error rate, we chose the pattern with a smaller error rate to be the plausible model.

The most challenging part of this project was that the data is too flexible and hard to decide the best pattern at the first sight because we had to do the data cleaning and generate new columns based on the existing columns, then the dataset was useful for the project. Basically, it was not possible to decide if the return outputs have linear relationship with the predictors or not. Therefore, during the exploring into the dataset, multiple trials on different statistical models were needed.

Data Description and Regressors Generation

Data is obtained from *finance.yahoo.com*, and the URL is shown below (https://finance.yahoo.com/quote/AAPL/history?p=AAPL).

The raw dataset only contains the open, close, high, low price, and the volume of 2514 days from Nov. 2008 to Nov. 2018. The sample is big enough to be generated. Instead of using the raw dataset we did some data manipulation on the dataset to generate a valid regression model. We generate two response variables, one is Return_t, which is the return of a specific day. Another is Return_t_pos, and it equals to 1 if Return_t is positive and 0 otherwise. In generating the models, only using the returns of one or two previous days is not strong enough to explain the response variable. Therefore, we are going to use as much information of the last three days as we can manipulate. Except for using each single previous days' returns, we go deeper and try to see the

trend of the last few days. Therefore, we generate several cumulative returns as well as two dummy variables showing the overall direction of price movement.

Here is the major variable information:

Volume: total trading volume of yesterday.

ln_vol: log form of volume. (note: ln(volume))

Vol_mi: volume in million.

Return_t: the return of that day(day4). (note: (close-open)/open*100, in percentage.)

Return_t_pos: whether the return is positive or not. 1 is positive and 0 otherwise.

Return_t_1: return of yesterday(day3).

Return_t_2: return of the day before yesterday(day2).

Cum_return_3: cumulative return from day1 open to day3 close.

Cum_return_2:cumulative return day1 open to day2 close.

Cum_return_1: return of day1.

Cum_inc: whether the return is continuous to increase in the last three days(day1, day2, day3). 1 is yes and 0 otherwise.

Cum_dec:whether the return is continuous to decrease in the last three days. 1 is yes and 0 otherwise.

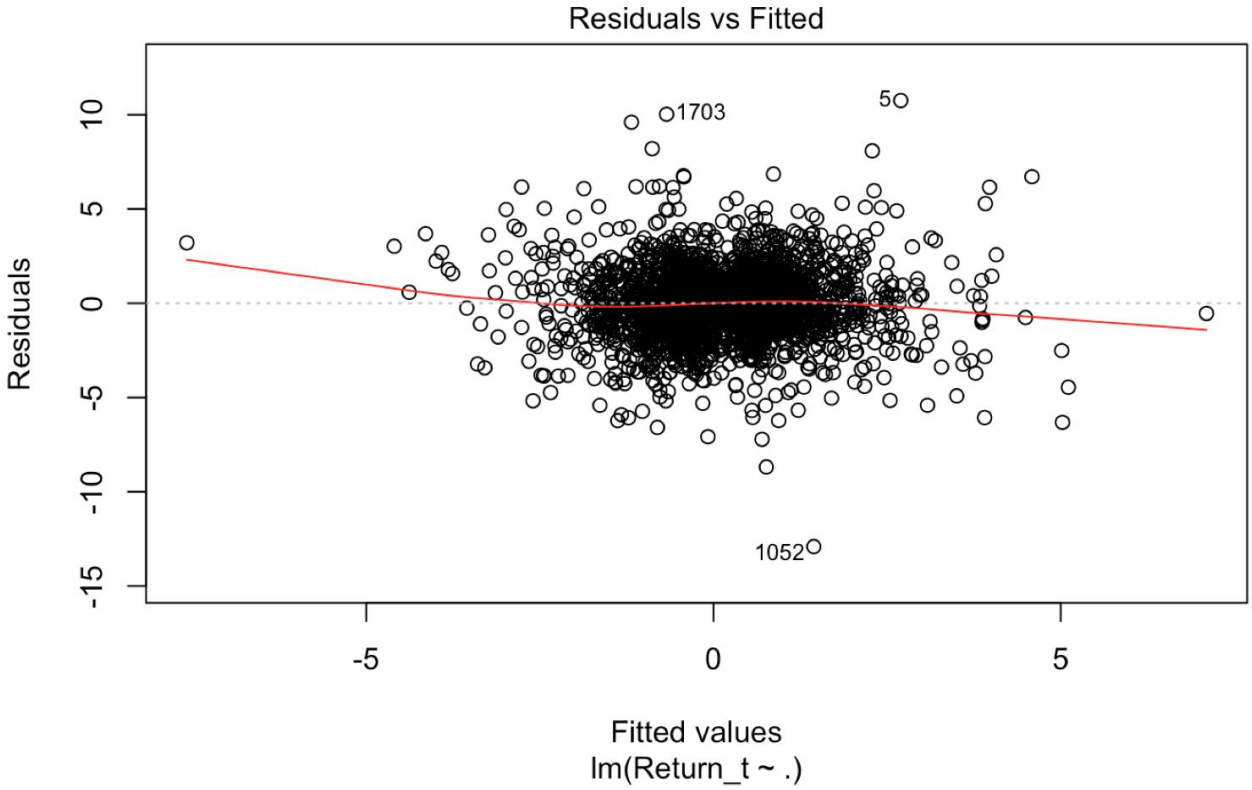
```
    Date               Open             High              Low              Close          Adj.Close          Volume
Length:2514        Min.   : 11.34   Min.   : 11.71   Min.   : 11.17   Min.   : 11.17   Min.   :  7.481   Min.   : 11475900
Class :character   1st Qu.: 49.85   1st Qu.: 50.27   1st Qu.: 49.52   1st Qu.: 49.93   1st Qu.: 33.433   1st Qu.: 37194275
Mode  :character   Median : 84.34   Median : 84.97   Median : 83.34   Median : 84.16   Median : 65.817   Median : 72733900
                   Mean   : 89.92   Mean   : 90.71   Mean   : 89.09   Mean   : 89.92   Mean   : 79.795   Mean   : 88278444
                   3rd Qu.:117.49   3rd Qu.:118.20   3rd Qu.:116.48   3rd Qu.:117.44   3rd Qu.:112.301   3rd Qu.:119130200
                   Max.   :230.78   Max.   :233.47   Max.   :229.78   Max.   :232.07   Max.   :231.263   Max.   :470249500
    ln_vol           Vol_mi          Return_t          Return_t_pos      Return_t_1        Return_t_2        Cum_return3
Min.   :16.26   Min.   : 11.48   Min.   :-11.4601   Min.   :0.0000   Min.   :-11.4601   Min.   :-11.4601   Min.   :-12.8191
1st Qu.:17.43   1st Qu.: 37.19   1st Qu.: -1.0362   1st Qu.:0.0000   1st Qu.: -1.0385   1st Qu.: -1.0385   1st Qu.: -1.5407
Median :18.10   Median : 72.73   Median :  0.1686   Median :1.0000   Median :  0.1678   Median :  0.1678   Median :  0.4231
Mean   :18.04   Mean   : 88.28   Mean   :  0.1459   Mean   :0.5394   Mean   :  0.1435   Mean   :  0.1401   Mean   :  0.3888
3rd Qu.:18.60   3rd Qu.:119.13   3rd Qu.:  1.3415   3rd Qu.:1.0000   3rd Qu.:  1.3392   3rd Qu.:  1.3392   3rd Qu.:  2.2971
Max.   :19.97   Max.   :470.25   Max.   : 13.4505   Max.   :1.0000   Max.   : 13.4505   Max.   : 13.4505   Max.   : 19.0260
 Cum_return2       Cum_return1          Cum_inc          Cum_dec
Min.   :-13.5473   Min.   :-11.4601   Min.   :0.0000   Min.   :0.0000
1st Qu.: -1.3105   1st Qu.: -1.0385   1st Qu.:0.0000   1st Qu.:0.0000
Median :  0.2526   Median :  0.1678   Median :0.0000   Median :0.0000
Mean   :  0.2628   Mean   :  0.1396   Mean   :0.2816   Mean   :0.2235
3rd Qu.:  1.8354   3rd Qu.:  1.3392   3rd Qu.:1.0000   3rd Qu.:0.0000
Max.   : 15.6214   Max.   : 13.4505   Max.   :1.0000   Max.   :1.0000
```

Since open, high, low, and close. would be known until the end of the day, these variables are excluded in all regressions that we generated. In original data set, variable volume is the volume of that day, so we lag the value of volume and use previous volume(or vol_mi) as today's predictor. The reason why we want cumulated return is because we want to offset the fluctuations between several days and see the price trend in a bigger picture. We hope these regressors will be useful in explaining today's return.

Model Specification

In this project we will make two main predictions, one is the direction of return, the other is the value of return. In order to address these two questions, we need to find a plausible classification model as well as a regression model to fit data. For classification models, we have Logistic Regression, Linear Discriminant Analysis and Classification Tree. For regression models, we have Multiple Linear Regression with OLS estimates, Ridge/Lasso, and Regression Tree. However, before working on each model assumptions and selecting models, we have to make two things clear: patterns of data and bias-variance tradeoff.

Too see the pattern of data, we fit a multiple linear regression on Return_t with all regressors, and plot the "residual-fitted value" graph:



From this graph, we see most of residuals fall between -5 and 5 and evenly distributed around 0. It means our data has a linear pattern, and there is no significant boundaries to set data points into different groups. With this information, we conclude that Tree-related method is not plausible, because they require non-linear model and clear decision boundaries between data points.

Speaking of Bias-variance tradeoff, we know it is the main difference between OLS estimates and Ridge/Lasso estimates. OLS gives the unbiased estimates, but larger variance in coefficients. Ridge and Lasso can reduce variance significantly by applying penalty term, but will result in biased estimates. Since we are studying stock return, a slightly more uncertainty in prediction may bring a very different consequence. We hope the variance of estimates to be as small as possible, despite of sacrificing some accuracy in coefficients. Therefore, we prefer Ridge/Lasso estimates over OLS estimates.

Finally, we decided to use Logistic Regression to predict the direction of Return_t, and use Multiple Linear regression with Ridge/Lasso estimates to estimate the value of Return_t.

Predicting Direction of Return: Logistic Regression

Before we started working on this part, we needed to assume what the multiple logistic regression could bring us. Logistic regression model is for predicting the percentages of an observation of the belonging class and it does not require the linearity of the data. In our project, we set the the classes to return of the next day of this observation in positive and return of the next day of this observation in negative. In the output of the regression, if the estimated

percentages of an observation is greater than 50%, then the observation was considered into the positive class, negative otherwise. We mainly focus on the estimated error rate of the trained model on the testset because by then we would know if such a classification method is plausible for the dataset or not.

In order to do this step, we chose only the vol_mi, return_t, return_t_1, return_pos, return_t_2, cum_return3, cum_return2, cum_inc and cum_dec because the price columns are for calculating the price and they are not useful in the classification. Then, we apply the logistic model to the dataset in R with the glm() function. The graph below is the output.

```
Call:
glm(formula = Return_t_pos ~ Vol_mi + Return_t_1 + Return_t_2 +
    Cum_return3 + Cum_return2 + Cum_return1 + Cum_inc, family = binomial,
    data = Apple_classification)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2588  -1.0021   0.4599   0.9441   2.5187

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.0044964  0.0818989  -0.055  0.95622
Vol_mi       0.0001355  0.0007712   0.176  0.86048
Return_t_1  -0.1570053  0.0543634  -2.888  0.00388 **
Return_t_2  -0.0523295  0.0416724  -1.256  0.20921
Cum_return3  0.8079206  0.0677755  11.921  < 2e-16 ***
Cum_return2 -0.7454439  0.0652333 -11.427  < 2e-16 ***
Cum_return1 -0.0128172  0.0580243  -0.221  0.82517
Cum_inc      0.3842673  0.1273479   3.017  0.00255 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We are focusing on the significance of the predictors in the multiple logistic model. These significant predictors contains more information of the dataset than the non-significant

predictors. The predictors in the summary of the logistic model with *s in the end implied the significance of the predictors. return_t_1, Cum_return3, Cum_return2 and Cum_inc are predictors with the most *s. They are the most significant predictors in the model. But we still have to run backward and forward selection to see if there are any differences.

In the forward selection, the significant predictors are return_t_1, return_t_2, cum_return3, cum_return2, cum_inc and vol_mi. In the backward selection, the significant predictors are vol_mi, return_t_1, cum_return3, cum_return2 and cum_inc.

```
Step:  AIC=2933.87
Return_t_pos ~ Return_t_1 + Return_t_2 + Cum_return3 + Cum_return2 +
    Cum_inc

              Df Deviance    AIC
<none>             2921.9 2933.9
+ Cum_return1  1   2921.8 2935.8
+ Vol_mi       1   2921.8 2935.8
```

(Forward selection)

```
Call:  glm(formula = Return_t_pos ~ Return_t_1 + Cum_return3 + Cum_return2 +
    Cum_inc, family = binomial, data = Apple_classification)

Coefficients:
(Intercept)   Return_t_1  Cum_return3  Cum_return2      Cum_inc
    0.01087     -0.16724      0.81932     -0.79288      0.36757

Degrees of Freedom: 2513 Total (i.e. Null);  2509 Residual
Null Deviance:      3470
Residual Deviance: 2924         AIC: 2934
```

(Backward selection)

After that, we divided the dataset into training set and testset. Take 75% of the data observations into training set and the other 25% as the testset.

```r
#set up the training set and test set
set.seed(1)
train=sample(nrow(apple), nrow(apple)*0.75)
trainset_=apple[train,]
testset_=apple[-train,]
```

After that, we apply two models from forward selection and backward selection to the training set to train the model.

```
Call:
glm(formula = Return_t_pos ~ Return_t_1 + Return_t_2 + Cum_return3 +
    Cum_return2 + Cum_inc, family = binomial, data = trainset_)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2300  -0.9961   0.4330   0.9475   2.5155

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.04184    0.06346  -0.659 0.509704
Return_t_1  -0.18792    0.05075  -3.703 0.000213 ***
Return_t_2  -0.02307    0.04137  -0.558 0.577093
Cum_return3  0.83237    0.06787  12.264  < 2e-16 ***
Cum_return2 -0.79355    0.07224 -10.986  < 2e-16 ***
Cum_inc      0.46916    0.14681   3.196 0.001395 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

(This is the output summary of the trained model from forward selection.)

```
Call:
glm(formula = Return_t_pos ~ Return_t_1 + Cum_return3 + Cum_return2 +
    Cum_inc, family = binomial, data = trainset_)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2398  -0.9985   0.4332   0.9439   2.5269

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.03911    0.06326  -0.618  0.53645
Return_t_1  -0.19631    0.04851  -4.047  5.2e-05 ***
Cum_return3  0.84200    0.06574  12.807  < 2e-16 ***
Cum_return2 -0.81472    0.06168 -13.210  < 2e-16 ***
Cum_inc      0.45801    0.14544   3.149  0.00164 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(This is the output summary of the trained backward selection model.)

The commons between two models are they both take return_t_1, cum_return3 and cum_return2 to be the significant models. In order to better understand which model is better, the models had to be applied to testset to compare their estimated error rate.

Below are the results from applying two models to testset.

```
#Apply the F-selection logistic model on the test set.
```{r}
glm.probs_F=predict(glm.train2, testset_, type="response")
glm.pred=rep(0, nrow(testset_))
glm.pred[glm.probs_F>0.5]=1
table(glm.pred, testset_$Return_t_pos)
1-mean(as.integer(glm.pred)==as.integer(testset_$Return_t_pos))
```
```

```
glm.pred   0   1
       0 185 100
       1  94 250
[1] 0.3084261
```

```
#Apply the B-Selection logistic model on the test set.
```{r}
glm.probs=predict(glm.train1, testset_, type="response")
glm.pred=rep(0, nrow(testset_))
glm.pred[glm.probs>0.5]=1
table(glm.pred, testset_$Return_t_pos)
1-mean(as.integer(glm.pred)==as.integer(testset_$Return_t_pos))
```
```

```
glm.pred   0   1
       0 185  99
       1  94 251
[1] 0.3068362
```

The estimated error rate from the backward selection is slightly smaller than the result from the forward selection. We have reasons to believe that the backward selection can be a better model for predicting the sign of the stock return. Based on the estimated result, the return_t_1,

cum_return3, cum_return2 and cum_inc are significant predictors for classifying observations in the data set. If the result from the prediction is classified into the positive, we can say that the return of the day after the date of the observation is positive. In other words, it is a good timing for making investment in the Apple Inc,. stock.

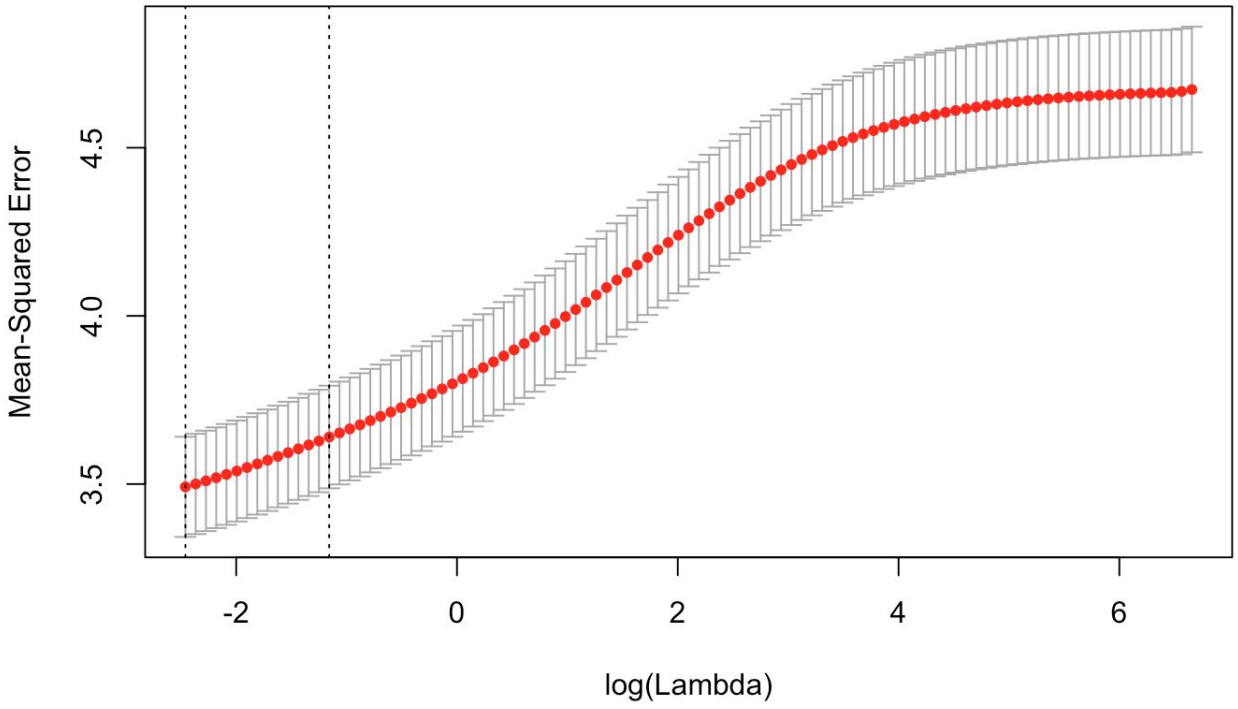
Predicting Value of Return: Ridge/Lasso Regression

To apply Ridge and Lasso estimates, we need to first check the linear regression assumptions. The Multiple Linear Regression requires that the response variable and explanatory variables have a linear relationship, and the error term follows normal(0,1) distribution. We already proved previously that Return_t is linearly correlated with other regressors, and based on the "residual-fitted value plot", we can also conclude that the error has a constant variance and mean equals to 0. Therefore, the assumptions of Multiple Linear Regression are met. Since we care variance more than bias, we want to fit a Ridge/Lasso multiple linear regression rather than OLS estimates.

First we want to fit a Ridge regression using all 8 predictors, Vol_mi, Return_t_1, Return_t_2, Cum_return1, Cum_return2, Cum_return3, Cum_inc and Cum_dec. We randomly split 75% of observations into a training set and the rest into a test set in order to estimate the test error. Next, we fitted a ridge regression model on the training set, and randomly set lambda=1 to get the general regression model. The result is as following:

```
  (Intercept)        Vol_mi    Return_t_1    Return_t_2   Cum_return3   Cum_return2   Cum_return1
0.0332906398  0.0005284665  0.1583026828 -0.1125464523  0.1101924163 -0.0970645779  0.0202283073
     Cum_inc       Cum_dec
0.4337517036 -0.3731398661
[1] "MSE for Ridge when lambda=1:  4.05822446345859"
```

From the above result, we see none of the coefficients are zero, which means Ridge regression keeps all regressors. The MSE is 4.06, and it is quite big. Therefore, we want to adjust the weight of penalty by selecting the best lambda that minimize the MSE. We then applied cross validation to find the best lambda for the model.



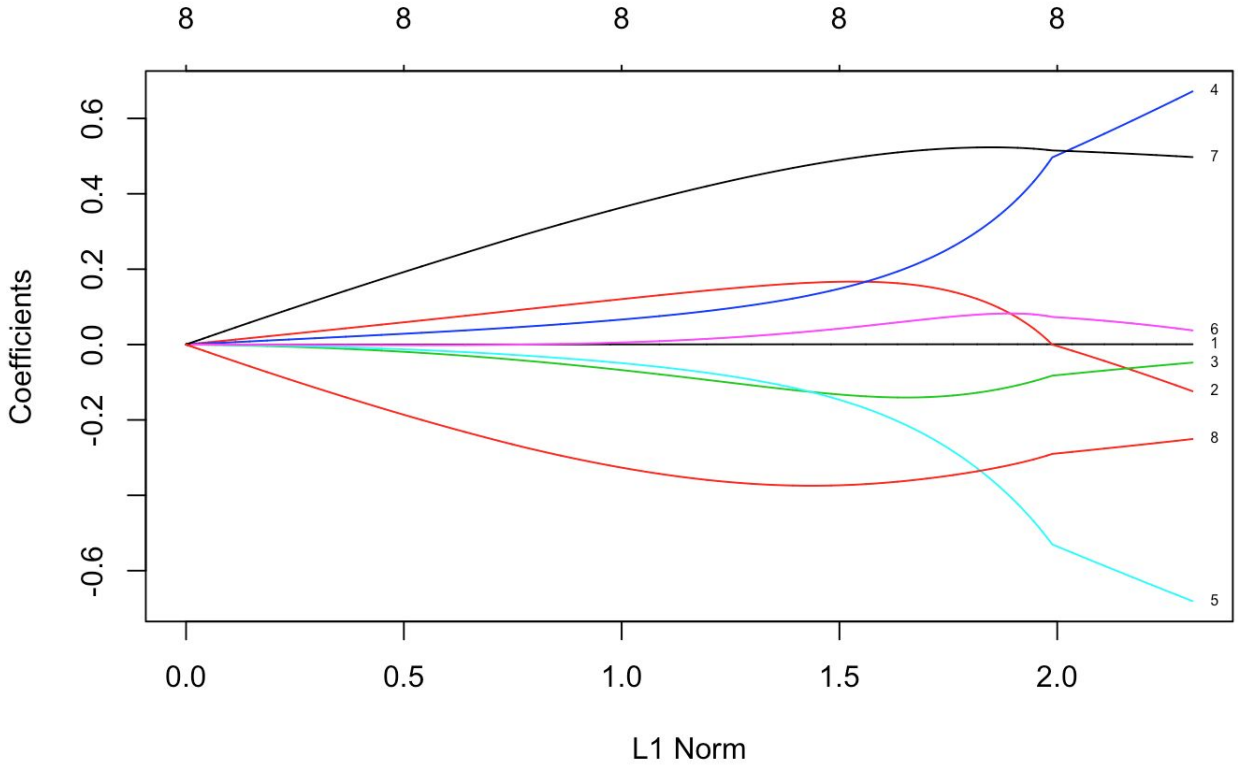
We can tell from the above graph, as lambda decreases, the mean-square error increases. Therefore, the best lambda will be very small and close to 0.

```
[1] "bestlam for Ridge:  0.0854238116298497"
[1] "MSE for best lambda:  3.72674563538809"
```

From the output, the best lambda is about 0.085, and the MSE corresponding to the best lambda is 3.73. It is reduced significantly. The estimate coefficients when lambda is 0.085 are shown below:

```
  (Intercept)        Vol_mi    Return_t_1    Return_t_2   Cum_return3   Cum_return2   Cum_return1
-0.0266181557  0.0007008581  0.0663896793 -0.1140392838  0.4109834897 -0.4262565116  0.0621062172
     Cum_inc       Cum_dec
 0.4499904110 -0.2960737228
```

We are also interested in whether Ridge regression can perform variable selection. To see intuitively, we plot the coefficients of each predictor as lambda decreases from 10^10 to 10^(-2).

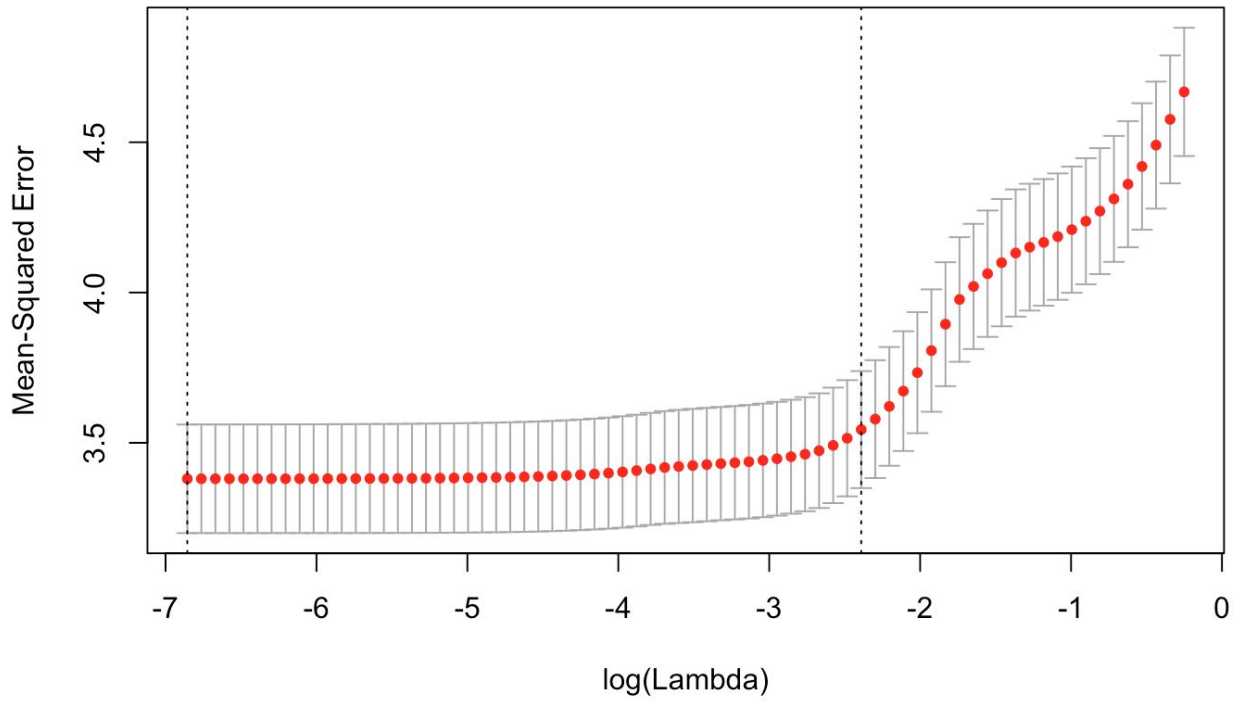


When we look at this graph from right to left, we can see as lambda increases, all coefficients are reduced to zero at the same time. It means we will either have all zero coefficients, or all non-zero coefficients. Therefore, Ridge regression cannot perform variable selection.

Secondly, we try to fit LASSO regression and see if there is any difference. By using the same predictors, training set, test set and lambda equals to 1 as before, we get the regular LASSO regression coefficients and MSE:

```
(Intercept)      Vol_mi  Return_t_1  Return_t_2 Cum_return3 Cum_return2 Cum_return1     Cum_inc
  0.1458697   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
    Cum_dec
  0.0000000
[1] "MSE for LASSO:  4.86799916069913"
```

From the above result, all regressors are zero, and the MSE is very big. One possible reason is that the lambda we selected is too large. To fix this issue, we apply cross validation method to find the best lambda:



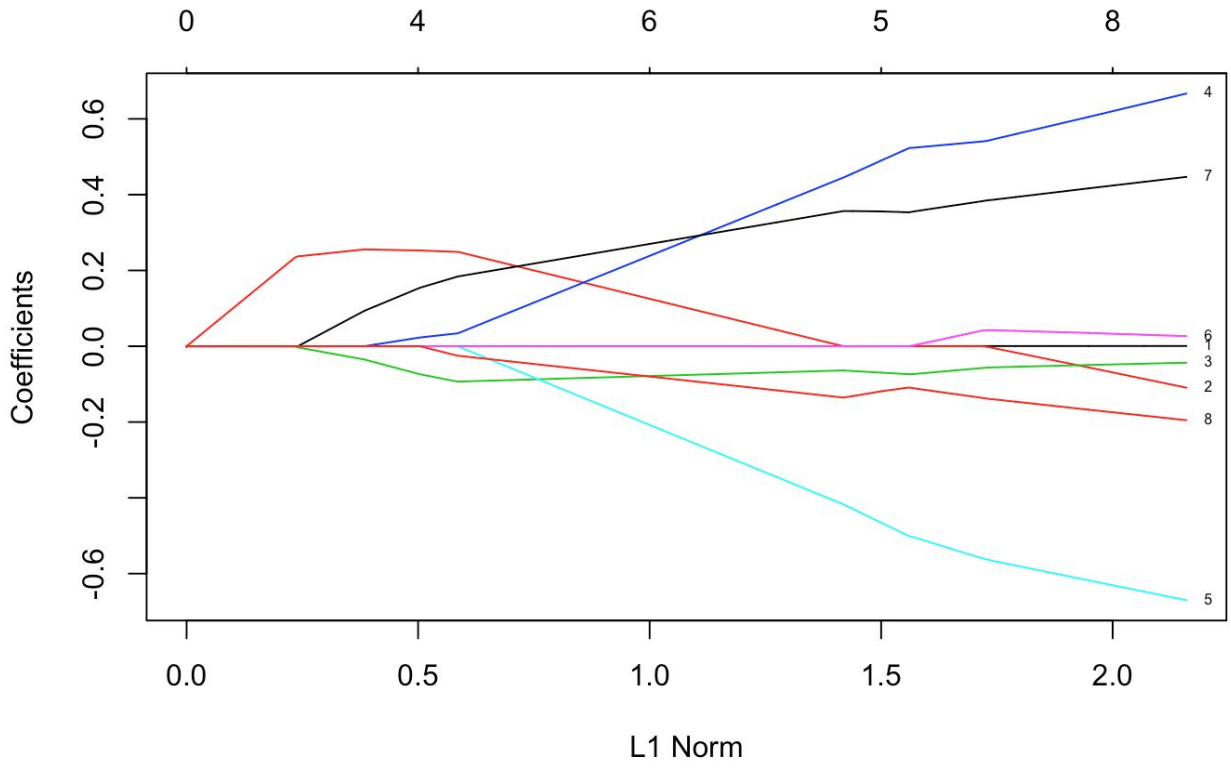
Similar to Ridge result, our mean-squared error increases as lambda decreases. However, in Lasso, the coefficients are more sensitive to the change of penalty weight.

```
[1] "bestlam for LASSO:  0.00105314467392201"
[1] "MSE for best lambda:  3.63086733647149"
```

Finally, our best lambda for Lasso is 0.001. Although it is very close to 0, it is still different from OLS estimates in a way to reduce coefficient variance. The following result is the coefficient for best lambda:

```
  (Intercept)        Vol_mi     Return_t_1     Return_t_2    Cum_return3    Cum_return2    Cum_return1
-0.0305485854  0.0005079446 -0.0979590693 -0.0617392380  0.6797510214 -0.6545306137  0.0027637246
      Cum_inc       Cum_dec
 0.3474065102 -0.1621877271
```

In order to check if Lasso can perform variable selection and select the most important regressors, we plot each variable coefficients versus different lambda from 10^10 to 10^(-2):



As we move from right to left, lambda is increasing, and coefficients are shrink to zero one after another. The first one reduced to zero is Cum_return2 which is shown in plot in light blue. It means Cum_return2 is not very important in explaining return_t comparing to other regressors. From the graph, we see the three most important regressors are Return_t_1(red), Return_t_2(green), and Cum_inc(black). The most important variable that is the last to be 0 is Return_t_1. Therefore, Lasso does perform variable selection.

Comparing to Ridge, Lasso has its advantages. Ridge treats each variable equally importantly, but Lasso enables us to select a subset of variable that is most important to our model. If some of our regressors are not helpful in explaining the behavior of future return, we can perform Lasso to make a variable selection. Therefore, we conclude Lasso is the most plausible model to predict the value of future return.

Conclusion

Stock investment is one of the hottest financial investments at present, so hundreds of people are researching stock prediction model. Because the return of stocks are determined by so many different factors, it is difficult for people to get a highly accurate stock prediction model. Therefore, we want to try if it's possible to predict future stock return solely on the previous price movements. After examining several models assumptions, we finally find that Logistic regression model, Ridge and LASSO regression model are plausible to help decide the investment trend. More clearly, Logistic regression model is for predicting the positive or negative sign of return, while Ridge and LASSO regression model are useful to predict the return rate of the stock. Our purpose of this project is to build up a great model to predict return of Apple stock, including generating useful indicators, exploring patterns of return, examining variable importance and finally improving trading accuracy. According to the results of previous analysis, we have the Logistic regression model with Return_t_1, Cum_return3, Cum_return2 and Cum_inc being significant predictors. What's more, that model is selected by backward selection with 30.68% estimated error rate, which is smaller than the model selected by forward

selection. Also, we have the LASSO regression model with Vol_mi, return_t_1, Return_t_2, Cum_return3, Cum_return2, Cum_inc and Cum_dec being predictors. Since LASSO can perform variable selection, we finally conclude that Return_t_1, Return_t_2, and Cum_inc are the most important variables to predict the value of return.

Each member's contribution to this project:

Anna Lu: "Ridge/Lasso Regression"/ "Conclusion"

Jinbo Wu: "Introduction" / "Logistic Regression"

Yuanrong Li: "Model Specification"/ "Ridge/Lasso Regression"

Yijun Xu: "Data Description" / "Logistic Regression"