Volatility Prediction
Final Report

Wanlin Li wl596, Zhiwei Zhou zz498, Meiyi Li ml2549

December 5, 2017

Abstract

This project aims to develop a deeper understanding of stock return volatility. Traditionally, people use historical volatility as an estimation for future volatility. This project hypothesizes that a better volatility estimation could be obtained through other information such as stock fundamentals and sentiment analysis information. We applied PCA to fill our missing data, and used K-fold cross validation to train our hyper-parameters. For fitting the data, we applied linear regression with Lasso, Ridge and Huber loss functions. As the result we found out that three parameters out of the 10 parameters we selected have significant predictive power over volatility.

1 Exploring Raw Data

1.1 Description

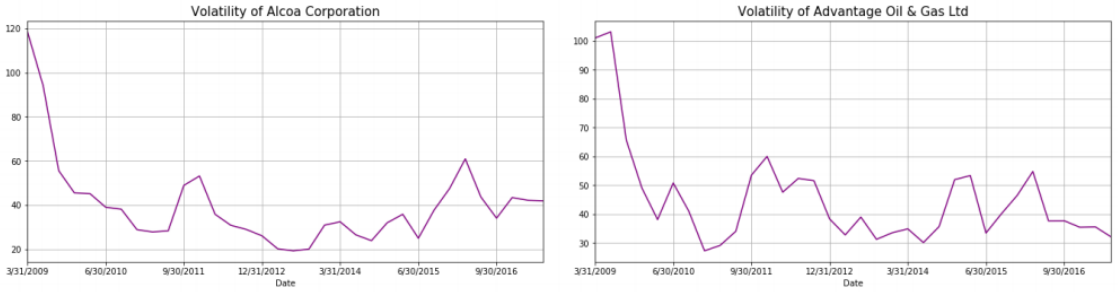

Figure 1: Volatility Data Sample

The volatility that we wish to predict is the volatility of all stocks in NYSE. (See sample in Figure 1) Our raw data come from Bloomberg, with a time span of 10 years. The data consist of one outcome variable, which is the volatility of stock daily returns over the past three months, and nine feature variables which are quarterly data of Asset Turnover, Current Ratio, Debt to Asset, Invested Capital, PE Ratio, Closing Price, Lagged Volatility, Implied Volatility, as well as Google Trend analysis of key words 'NYSE' and 'VIX'. (See variable explanation in the appendix. See sample raw data in Figure 3, Section 1.3) Total number of data point is 107,304. Since we have feature data that scale only from 0 to 1, and we have feature data that are in hundred-level, we normalized our feature data and volatility data for truthful investigation of their relationship.

1.2 Basic Exploration

Since we are looking for relationship between each feature and volatility, we have graphed the scatter plot of volatility against each feature. We have added color in the graph to represent close prices for an in-depth understanding of the dynamics. Selected examples are demonstrated in Figure 2.

There are several preliminary understandings from the graph. For the upper left graph of Volatility against PE, we could see that there is a negative relationship. Moreover, the variance of Volatility of a given PE level decreases as PE increases. So our preliminary assumption is that there is a negative correlation between volatility and PE.

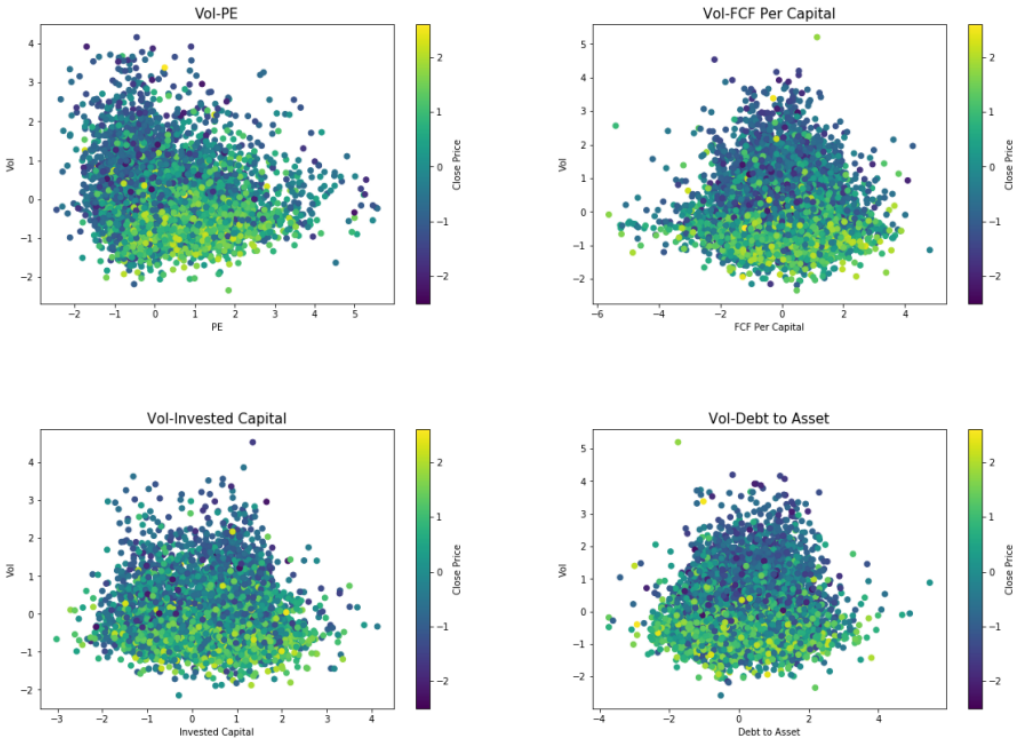

Figure 2: Scatter plot

For the lower left graph of Volatility again Invested Capital, we can still see a vague negative correlation between the two variables. However, whether the correlation relationship is truly negative, and whether the relationship is linear or non-linear can not be immediately concluded from simply looking at the graph.

For the upper right and lower right graphs, the direction of correlation could not be observed. The only information we could get from these two graphs is that Volatility data have a wider span when the feature data is closer to average value. This makes sense, because when feature data, which is fundamental data take extreme values, market is more certain about the company's future performance, therefore the variance of volatility is low when feature data take extreme values. However, the prediction power of these two features on volatility does not appear to be strong.

Finally, there is a clear relationship between closing price (represented by color) and Volatility. When the closing price is low, Volatility tends to be high. Therefore, closing price is a promising feature to predict Volatility.

1.3 Data Cleaning

There are several challenges during data cleaning. As shown in Figure 3, the PE ratio data for two different stocks are misaligned in terms of date. Therefore, our first challenge is that our data were recorded with unaligned date time and indexing. Furthermore, the PE ratio data for AA US Equity is shorter than A US Equity ('A' and 'AA' are ticker names) because some data are missing. Thus the second thing we need to over come is the abundant presense of NaN data.

1.3.1 Index Alignment

To tackle the first problem, we replaced the old indexes, which came in mixed forms of string and integer, with new self-created indexing in the form of pandas.datetime(). With the new index, we re-aligned the data with higher precision and generated a read-friendly database.

A US Equity			AA US Equity	
Date	PE_RATIO		Date	PE_RATIO
3/31/2009	8.1677		3/31/2009	65.3614
6/30/2009	16.9562		6/30/2009	92.9664
9/30/2009	40.0473		6/30/2010	30.1387
12/31/2009	32.2849		9/30/2010	32.9657
3/31/2010	23.8146		12/31/2010	24.9358
6/30/2010	10.172		3/31/2011	18.7972
9/30/2010	9.9363		6/30/2011	13.7391
12/31/2010	11.1039		9/30/2011	7.4223
3/31/2011	10.9535		12/30/2011	9.2189
6/30/2011	13.2875		3/30/2012	14.3762
9/30/2011	7.7628		6/29/2012	25.387
12/30/2011	8.4473		9/28/2012	689.0159
3/30/2012	10.3927		12/31/2012	57.0775
6/29/2012	9.6592		3/28/2013	44.6684
9/28/2012	11.1765		6/28/2013	37.6359
12/31/2012	12.6985		9/30/2013	17.1503
3/28/2013	13.7932		12/31/2013	23.6453
6/28/2013	15.0227		3/31/2014	34.099
9/30/2013	15.3921		6/30/2014	27.2816
12/31/2013	19.0446		9/30/2014	20.6336
3/31/2014	22.9077		12/31/2014	13.5565
6/30/2014	27.4998		3/31/2015	8.9699
9/30/2014	38.7684		6/30/2015	7.6992
12/31/2014	38.4372		9/30/2015	8.3498
3/31/2015	36.2145		12/31/2015	14.7318
6/30/2015	31.1853		3/31/2016	441.5141
9/30/2015	23.5085		6/30/2016	419.3923
12/31/2015	27.6865		12/30/2016	83.6825
3/31/2016	26.049		3/31/2017	21.8381
6/30/2016	28.3274		6/30/2017	12.7006
9/30/2016	29.1689		9/29/2017	17.9731
12/30/2016	25.8774			
3/31/2017	27.3325			
6/30/2017	31.3783			
9/29/2017	32.6412			

Figure 3: Sample Data

1.3.2 PCA: Filling in Missing Data

For the second problem, we investigated all the features for each stock. For any missing values in our data matrix, we used PCA model to auto-fill in the data that's missing. The PCA loss function is stated as follows:

$$min||Y - XW||^2 \tag{1}$$

Y	Data in matrix form of one single feature (rows: stocks, columns: date time)
X	Example vector
W	Feature vector

After the analysis, we multiplied X and W to generate the data for each single feature with all missing data filled. We applied PCA separately on data of each feature before flattening the data. (Flattening here means concatenating the data together to become the feature matrix.) The reason behind is that if we did PCA analysis after flattening the data, the filled data for one specific stock will include information from other stocks as well, which is not wanted since a specific stock's feature data is only most correlated with other data under that feature for this stock.

1.4 Data Formatting

We formated our sample data by first normalizing our dependent variable and independent variable for comparability since some feature data scale in 0 to 1 while some other feature data scale in 100s. Second we concatenated the data matrix of each feature together. After concatenation, our finalized sample data would be normalized volatility data for dependent variable. The columns for feature data matrix would be the 10 features that we selected, and the rows would be repetitive time. See Figure 4 for finalized sample data.

2 Preliminary Regression and Analysis

In this section, we made the temporary assumption that our outcome variable, which is Volatility, is in linear relationship feature variables. We divided our data into training and test set with ratio 2:1.

	CR	Imp_Vol	AT	DA	NYSE_Tr	PE	Close	Lag_Vol	VIX_Tr	IC	Vol
0	2.623300	0.000	0.446925	32.755500	124	8.167700	11.176800	60.979000	123	4662.000000	50.719
1	1.522200	-13.615	0.567539	27.646800	103	65.361400	25.217400	118.418000	115	23294.000000	94.063
2	0.723789	0.000	1.042106	43.192916	85	1.574106	15.797546	198.286701	108	247.649035	0.000
3	4.667500	4.920	1.181921	6.399000	87	14.617500	17.746700	61.122000	104	882.231000	50.914
4	1.260500	1.933	1.597668	9.597500	77	14.414000	33.650000	52.466000	106	1473.660000	40.035
5	0.201298	0.000	0.165850	59.900195	80	116.046587	12.560233	66.923478	142	106.277856	0.000
6	0.629700	16.081	0.112617	35.916700	67	6.391271	4.210000	100.954000	93	2094.754000	103.095
7	1.996639	-2.527	4.157729	0.000000	78	7.251200	20.790000	113.330000	90	1596.763000	86.887

Figure 4: Finalized Sample Data

2.1 Linear Regression with One Feature

We first regressed Volatility over each single feature separately. The motivation is to find out whether any feature has strong enough explanatory only by itself. An example of Volatility regressing against Price Earning Ratio is shown.

$$Vol = a * Feature + b \tag{2}$$

Vol : Normalized Volatility
Feature: We regress Volatility against each of the feature variable in normalized form.

Figure 5 shows the predicted Volatility against real Volatility of test set. As can be seen from the graph, data scattered focusing on the red line, which is prediction equals to real value. Therefore, we concluded that PE has explanatory power over Volatility, and this feature is worth keep exploring.

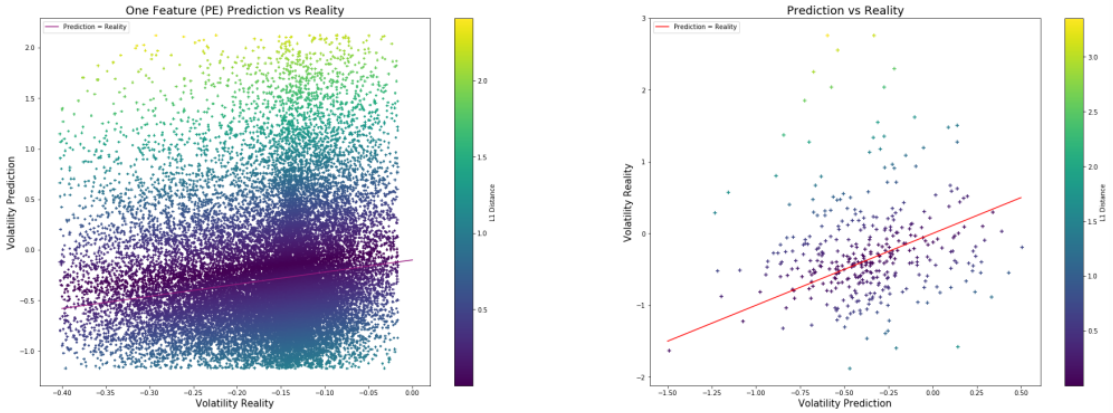

Figure 5: Prediction vs. Reality for Regression Using Price Earning Ratio.

2.2 The Correlation Matrix

Before we first fit data with our model, we wanted to know whether there is significant correlation between features, in other words, any possibilities of multicollinearity. We investigated the correlation relationship using covariance matrix. See Figure 6 for correlation matrix.

As seen from the graph, most correlation terms are rather insignificant besides the correlation of VIX_Tr and NYSE_Tr. The correlation between these two features are over 0.6. In this case, we are expecting these two features either to be not important to volatility prediction at the same time, or that one of the features will be dropped when we are using Lasso regression.

3 Fitting the Data Using Machine Learning Method

In this section, we ran linear regressions with different loss functions and select the model that fits best, and then test the selected model on test data. We will first use K-fold cross validation to select the hyper-

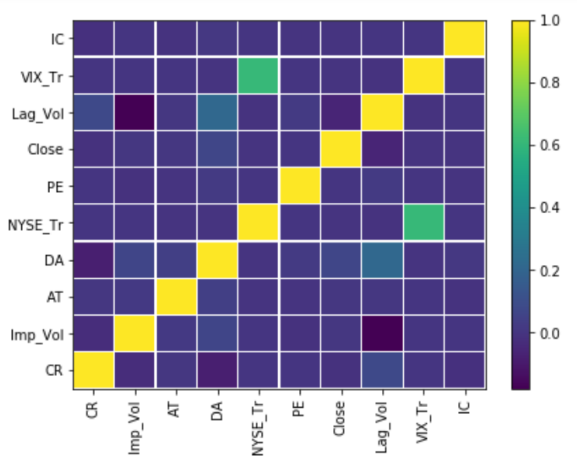

Figure 6: Correlation Matrix

parameter, and then we fit data with Lasso, Ridge and Huber. will use cross validation to select

3.1 The Linear Regression

We have chosen linear regression to fit our data. The equation is shown below. We are going to use loss functions of Lasso, Ridge, and Huber regressions to fit the data.

$$Vol = w_1*AT+w_2*CR+w_3*DA+w_4*IC+w_5*PE+w_6*Close+w_7*Lag\_Vol+w_8*Imp\_Vol+w_9*VIX\_Tr+W_{10}*NYSE\_Tr- \tag{3}$$

Vol	Normalized Volatility
AT	Normalized Asset Turnover
CR	Normalized Current Ratio
DA	Normalized Debt to Asset
IC	Normalized Invested Capital
PE	Normalized Price to Earning Ratio
Close	Normalized Closing Price
Lag_Vol	Normalized Lagged Volatility
Imp_Vol	Normalized Implied Volatility
VIX_Tr	Normalized VIX Trend
NYSE_Tr	Normalized NYSE Trend
Constant	constant (intercept) term

3.2 Lasso

For Lasso, the regularizor is l1 norm. The loss function is shown below:

$$min\frac{1}{n}\sum(y_i - \widetilde{y})^2 + \lambda\sum|w|_{l1} \tag{4}$$

3.2.1 K-Fold Cross Validation

First of all, we split the data into 2:1 training set and test set. We decide the value of λ using K-fold cross validation on the training set. Figure 7 (left) is a plot of mean squared error of the resulting model as a function of λ. We selected our λ which is the value that achieves the lowest error.

3.3 Ridge

For Ridge, the regularizor is l2 norm. The loss function is shown below:

$$min\frac{1}{n}\sum(y_i - \widetilde{y})^2 + \lambda\sum|w|_{l2} \tag{5}$$

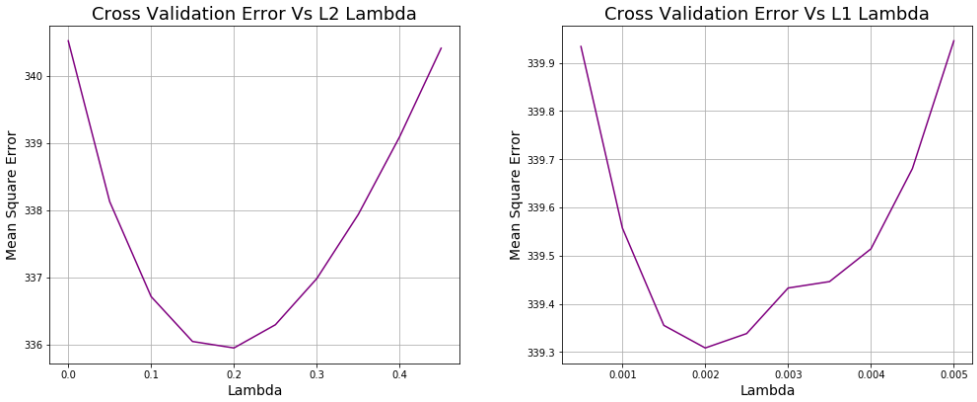

Figure 7: Error as a Function of λ: Left for Lasso and right for Ridge

3.3.1 K-Fold Cross Validation

Same procedure as in Lasso. We split the data into 2:1 training set and test set. We decide the value of λ using K-fold cross validation on the training set. Figure 7 (right) is a plot of mean squared error of the resulting model as a function of λ. We selected our λ which is the value that achieves the lowest error.

3.4 Huber

For Huber, the regularizor is l2 norm. We applied gradient descent to solve the optimization question. The loss function is shown below:

$$min \frac{1}{n} \sum huber(y_i - \widetilde{y})^2 + \lambda \sum |w|_{l2} \tag{6}$$

$$huber(z) = \begin{cases} \frac{1}{2}z^2, & \text{if } |z| \leq \epsilon \\ |z| - \frac{1}{2}, & \text{if } |z| > \epsilon \end{cases} \tag{7}$$

3.4.1 K-Fold Cross Validation

We conducted K-fold cross validation for selecting both ϵ as well as λ. First of all we set the regularization term to zero, and select value of ϵ at the point when error term is the lowest. And then we select λ with the decided ϵ value, at the point where the error term is the lowest. Figure 8 shows the relationship of error and ϵ as well as λ.

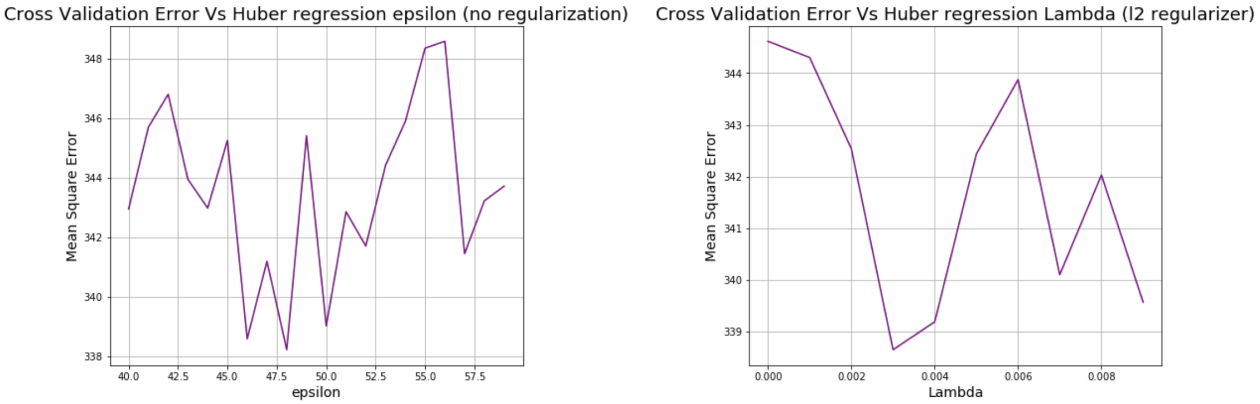

Figure 8: The relationship between error vs. ϵ and error vs. λ

3.5 Model Selection

We fitted both Lasso, Ridge and Huber regression on the validation data, and calculated the mean squared error for both regression. And we selected the model with lower mean squared error to be the model that

we will use for testing.

Below is the chart for the parameters as result of fitting.

Parameters	Lasso	Ridge	Huber
AT	0.0123	0.00735	0.579
CR	-0.188	-0.0563	-0.776
DA	0.0522	0.0557	0.962
IC	5.82e-08	5.59e-08	0.766
PE	0.000436	0.000356	1.61
Close	0.0518	0.0302	2.6
Lag_Vol	0.413	0.264	11.5
Imp_Vol	-0.243	-0.187	-3.19
VIX_Tr	0.000123	0.000432	-0.503
NYSE_Tr	0.00292	0.00113	-0.126
Constant	5.167	7.484	21.406

Below is the comparison of mean squared error (MSE) of both Lasso and Ridge.

	Lasso	Ridge	Huber
MSE	324.331	334.574	328.164

Judging from the mean squared error from the training result, we should use Lasso regression as the model to proceed to test the data.

3.6 The Testing Result

We applied Lasso to the testing data, and obtained the mean squared error of 316.331. Judging from that this number didn't increase too much from the mean squared error obtained in training set by Lasso, our model doesn't have the problem of over fitting.

Figure 9 shows the plot of predicted volatility against real volatility. There is a clear linear relationship between prediction and real values.

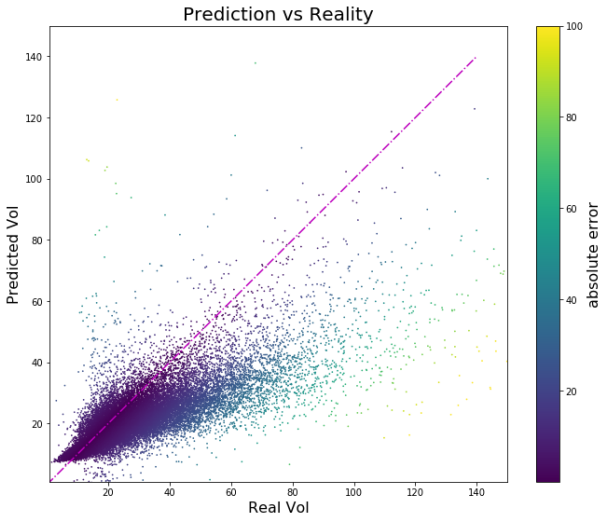

Figure 9: Predicted Volatility vs. Real Volatility

4 Analysis

4.1 Lasso vs. Ridge vs. Huber

Comparing the parameter results from Lasso and Ridge, we can see that for same feature, the parameter achieved by Lasso and Ridge is very close to each other, which adds confidence to the regression result.

For the parameters obtained from Huber loss function, all the parameters have absolute values that are much greater than those from Lasso and Ridge. In fact, all parameters are around 20 to 100 multiples of those in Lasso and Ridge. However, the relative relationships between parameters in terms of scale and direction stayed the same. Our explanation is that the lambda in the regularization term in Huber regression took a value of 0.003, very close to 0. In this case, parameters were allowed to take much larger absolute values while keeping their relative relationships.

4.2 The Parameters

According to the parameters achieved by Lasso, the rather significant contributors to volatility are Lag_Vol, Imp_Vol and CR, in order of significance.

For Lag_Vol, which is the volatility of last quarter, the positive parameter means that volatility data are actually autocorrelated, and volatility of the next quarter is significantly related with volatility from last quarter.

For Imp_Vol, which is the implied short-term (10 days) volatility calculated through Black_Scholes Model, the negative parameter signals that when the short-term volatility is high, the quarter volatility will be lower, which shows the mean-reversion mechanism of the market.

For CR, the current ratio, the negative parameter means that when the company is in healthy financial situation, the volatility is low. When the company is close to in-solvency, the volatility is high.

At the same time, the low parameters for VIX_Tr and NYSE_Tr validates what we have predicted in the correlation matrix section: since their correlation is so high, it is possible that they could be simultaneously insignificant in their parameters.

5 Conclusion

We have applied PCA to fill our missing data, and used K-fold cross validation, as well as using linear regression with Lasso, Ridge and Huber loss functions to fit our data. Our model has achieved a test set mean squared error of 316.331, which is very close to mean squared error from training set. This signals that we did not over fit.

The parameter results are very intuitive: for positive parameter of lagged volatility, it signals that volatilities are actually autoregressive in a short term. For negative parameter of implied short-term volatility, it signals that when short term volatility is high, the long term volatility should be lower, reflecting the mean-reversion mechanism of the market. For negative parameter of current ratio, it implies that when the company has enough asset, the volatility is low. When the company's solvency ability is low, the volatility is high.

6 Appendix

6.1 Variable explanation

Volatility (Vol)	Measures the risk associated with specific stock. It is the standard deviation of return of stock prices.
Asset Turnover (AT)	This is a measurement of how efficiently the company is using it's asset to generate income. Calculated by Net sales revenue/Average total assets.
Current Ratio (CR)	This is a measurement of the company's solvency. Calculated by Current Asset/Current Liability.
Debt to Asset (DA)	This measures a company's leverage over debt. It is calculated by Total Debt/Total Assets.
Invested Capital (IC)	It is the capital raised by issuing bond or equities.
Price to Earning Ratio (PE)	Measures a company's value or earning potential. Calculated by Market Value per share/Earnings per share.
Close (Close)	Closing price of stock.
Lagged Volatility (Lag_Vol)	The volatility of the same stock from last quarter.
Implied Volatility (Imp_Vol)	This measures the short-term implied volatility of the next quarter using put price by Black-Scholes formula. It is calculated by implied volatility of 90 days put minus implied volatility of 10 days put.
VIX Trend (VIX_Tr)	This is a Google-Trend measurement of how hot the 'VIX' key word is being searched. VIX is an index measuring market volatility over the future 30 days.
NYSE Trend (NYSE_Tr)	This is a Google-Trend measurement of how hot the 'NYSE' key word is being searched.