ORIE 4741 Final Report
Cornell University
Dec 4, 2017

VIX Prediction

Hanyin Cao (hc936), Yiyang Wen (yw892), Leqi Zhao (lz469)

Abstract
Among all kinds of financial indices, the CBOE Volatility Index - VIX, is the benchmark for stock market volatility. It is constructed using the implied volatilities of a wide range of S&P 500 index options, indicating the market's expectations on 30-day volatility [1]. VIX is often referred to as the "fear index", with its value spikes during market turmoil or periods of extreme uncertainty [2]. This can be reflected on the VIX movement during the 2008 financial crisis, which means VIX is a very useful indicator for financial risk management.

In addition, these years trading VIX relevant products like VIX ETFs/ETNs is becoming one of the most profitable trading strategies and thus many hedge funds are putting much investment on its research. As a result, if we can correctly predict the VIX trend in the future, we can gain large profit from this booming market. In this project, we are going to develop efficient models to predict VIX movements and finally construct a effective trading strategy on VIX ETF.

Contents

1. Data Exploration

1.1 Data Gathering

To determine what kinds of data to use, we first decided to include the main financial market data and active options contracts data since they were directly correlated with the VIX. What about the data in other categories? We looked at the events occurred when extreme VIX values appeared. From figure 1, we found that there was a positive correlation between VIX and the unemployment rate. The On the other hand, the events like 2008 financial crisis and European sovereign debt crisis were also accompanied by the high VIX value, which told the strong connection between VIX and market motion. As a result, we were also going to include the macroeconomic and market sentiment indices in our data set.

We obtained our financial and sentiment indices through the Bloomberg terminal on campus. Our macroeconomic data were mainly obtained from https://fred.stlouisfed.org/. The website provides a wide range of economic series' historical data. Besides, we chose 50 most relevant keywords with respect to VIX and downloaded their searching volumes through Google Trends.

1.2 Data Overview

The data we are gathering includes 78 time series with different start and end dates, different frequency (daily, weekly, monthly and quarterly), and different types (continuous, discrete and nominal). More specifically, the data can be further grouped into four categories:

1. **Main market indices** include two parts of data. The first part is the financial market indicies, including S&P 500, Nasdaq, Russell 2000, Dow Jones Industrial Average, etc. These indices can reflect the domestic market trend to a large degree as they include a wide variety of securities. The second part is the active options contracts indices including ESA index, SPA index and other 11 indices. Essentially, VIX reflects the implied volatility of the market, so the price of the active options contracts may have fundamental influence on VIX.

2. **Macroeconomics data** include consumer price index for all urban consumers, crude oil prices, USD/EUR foreign exchange rate and other 24 indicators, with different frequency in daily, weekly, monthly and quar-

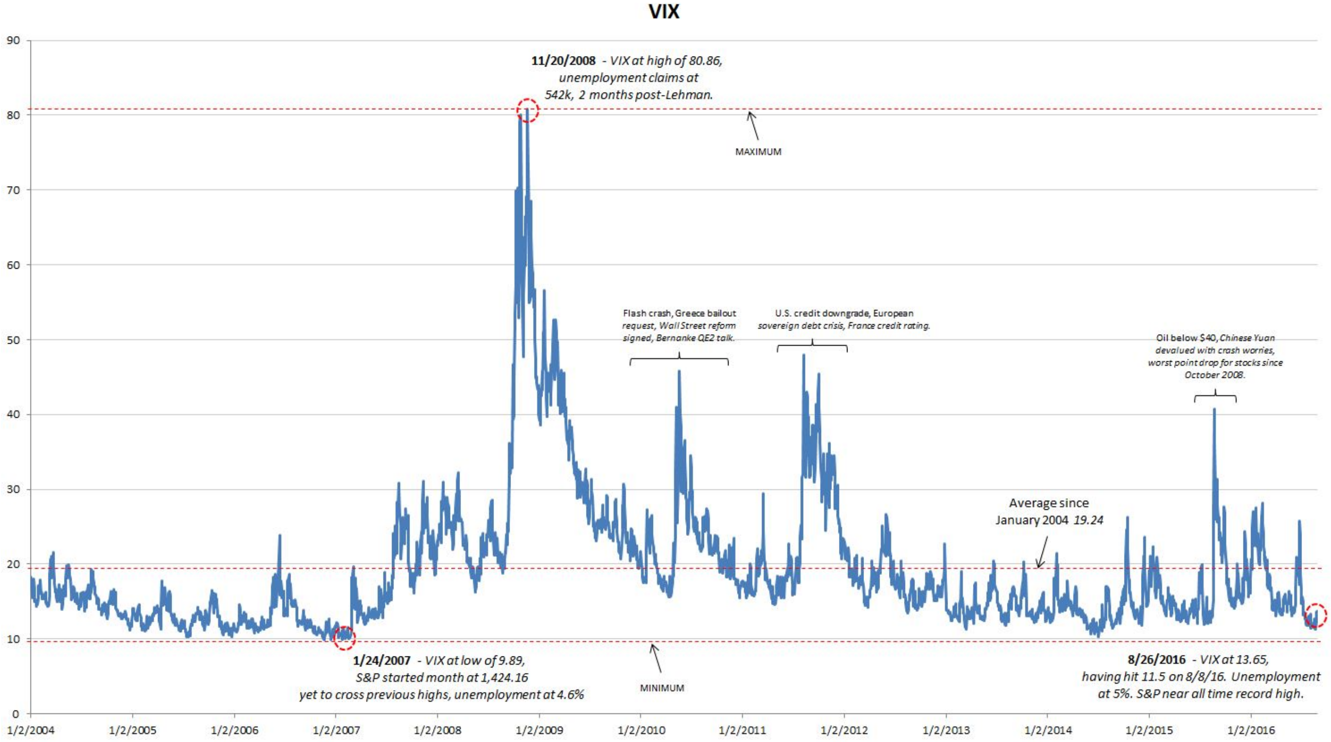

Figure 1. 2008 Financial Crisis [3]

terly. Such data reflects the general condition of the U.S. economy, subsequently influence the general expectation and further influence VIX index.

3. **Sentiment Index data** indicate the market emotion. The 23 sentiment indices were obtained from Bloomberg which could reflect the market emotion from various aspects such as ratios of different asset class, the performance of certain securities, social surveys of the investors' sentiment.

4. **Google Trends data** contain VIX related topics searching volume through Google Trends. The topics we chose including VIX, volatility, risk, stock and 21 other key words. We selected our keywords by choosing from the Google recommended VIX top related topics. We also added the topics which were thought essential to improve our prediction like unemployment rate (the weekly frequency in Google Trends was higher than the corresponding index frequency in macroeconomic section). When downloading the data, we specified the region by United States and category by Finance. The obtained data were in weekly frequency and ranging from Jan 2004 to Nov 2017.

For more detailed data description including series full name, meaning, frequency, starting and ending dates, please refer to the Appendix file in our Github folder.

1.3 Data Preprocessing

The data we were processing was quite messy, with different time range, different frequency and different ratio of missing data. Before gathering all features, we decided to preprocess our four categories of data separately. We wrote a date processing function that convert the date to standard format, like 2017-10-27. Based on the different ways we obtained our data, each group of indices were stored in various formats. Thus, we wrote respective data aggregation functions to combine indices in the same group into a single file under a consistent format and time range. Here are two problems we met during the data aggregation:

The main problem of the google trend data was that the scale of data was inconsistent. For each time interval, Google sets the most frequently searched week with a value 100, with the value of other weekly searching volumes set based on it. Since we were allowed to download 5-year data at a time, in other words, we had three time intervals for each time series, which means that we need to standardize the scale on the whole time interval that we covered. The second problem was our macro data were in different frequency: daily, weekly, monthly, quarterly, we need to find a way to combine them in daily basis.

As for the problem of data scale, we solved it by downloading the data of two adjacent time intervals with overlapped date. In this case, we can calculate the ratio of overlapped day searching volume from two time intervals to get the stan-

Table 1. Summary statistics

Group	Column	Mean	Std	Min	Med	Max
Macro Data	DAAA DGS10	6.98 6.26	2.38 2.86	3.18 1.37	6.81 6.01	13.76 15.84
Google Trend	BETA RISK	27.30 50.96	13.31 12.60	6.86 18.09	24.43 49.77	100 100
Senti- ment	CSFB ISESEQ	20.28 121.37	7.57 77.16	9.74 0	17.70 140	46.06 410
Market Index	SPX SPA	1145.50 1957.93	532.46 323.92	295.46 0	1153.59 2003.60	2575.21 2574.00

dardizing factor. By multiplying each searching volume from the second time interval with the standardizing factor, we can get consistent scale on the whole time interval. To solve the macro frequency problem, since we want our final data in the daily basis, we filled in our lower frequency data value to a corresponding range of time. It is reasonable based on the time series property.

1.4 Data Visualization

Distribution

First, let's have a look at the VIX return distribution in Figure 2. We see that the values are in a quite good normal shape with a long right tail. The skew value is 1.32. It implies that the VIX value is stable in a range in general, however, there are cases the VIX will grow dramatically in a very short time, showing the great fear from the investors.

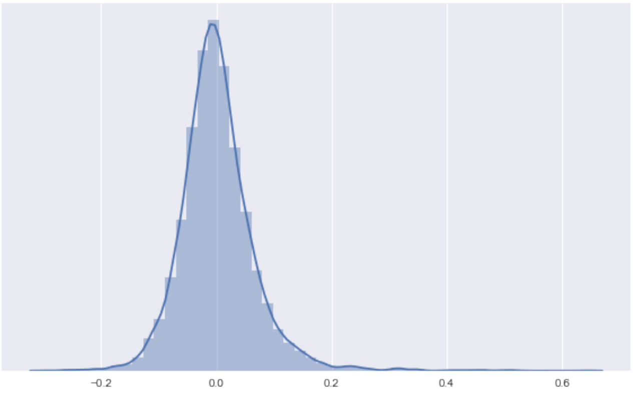

Figure 2. VIX return distribution

Next, we drew two box plot corresponding to macro economy data and sentiment data, respectively. We find two main problems from these two plots. Firstly, some features, like DAAA and DGS10, have similar distribution. This reminds us that some features may highly correlated with each other and we can drop some of the features. Secondly, from Figure 3 we can see that some sentiment data have low variance, so we should be aware of that this type of data might not offer enough information to predict the VIX index.

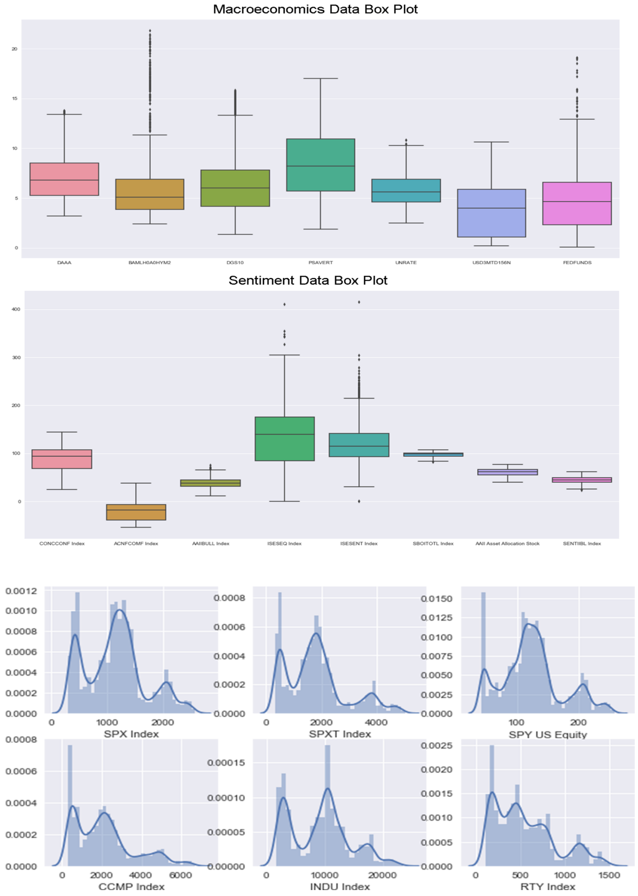

Figure 3. Distributions

The Figure 3 also indicates that the financial indices like SPX, SPXT, SPY US Equity, INDU are having the similar distributions. As a result, we can guess that some of our features should be correlated. In order to find out how strongly they are correlated, we drew the correlation plots for our financial indices, active contracts, macro indices, and Google Trends data to figure out the indices relationship within the same group.

Correlation Analysis

In order to make the feature relationship more visible, we changed colour range scale for each data group, which is specified in the sub-plot title. It is obvious to notice that the financial market group data are highly correlated. Furthermore, based on our calculation for the active contracts indices, all features except STEA have the correlation of 1. So besides keeping STEA, we eliminated other redundant features and only remained ESA index as it has the most data points. On the other hand, the correlation plots for Macroeconomics and Google Trends data are more complex. It tells we might be able to use them to catch different information on VIX. In the next section, we will see how Google Trends data is connected with the VIX movement.

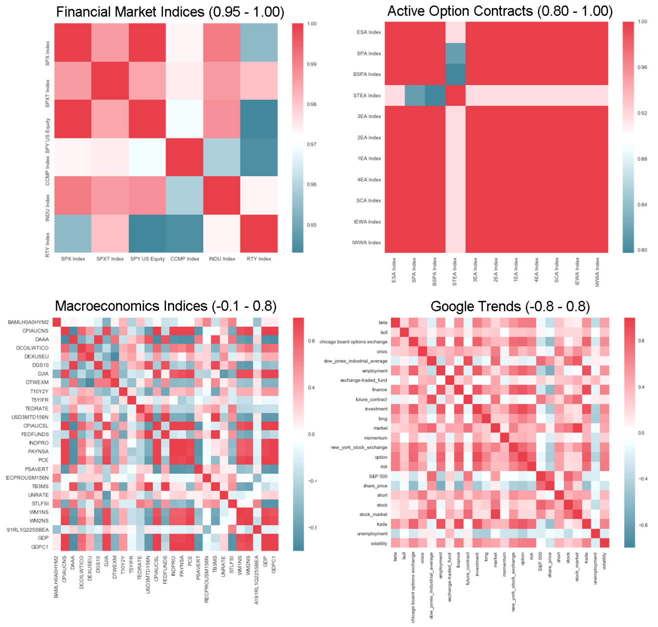

Figure 4. Correlation Analysis

Google Trends

From the google trend time series data, we noticed that the changes of searching volume of some key words are a little bit advanced comparing to the trend of VIX index. This phenomenon gave us confident that the google trend data are efficient to predict the VIX index.

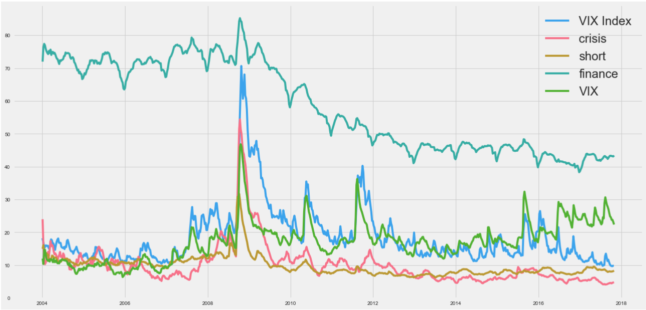

Figure 5. Google Trends

1.5 Data Cleaning

Since we are predicting the VIX movement, we are more interested on the change of VIX other than the VIX numerical value. As a result, we realized that it made more sense to change all feature values into the change rate form, that is, the new value of feature n in observation m is:

$$f_{mn}^* = \frac{f_{mn} - f_{mn-1}}{f_{mn-1}}$$

Our VIX data is range from 1/2/1990 to 10/20/2017, the first step we did was to remove earlier dates data on all features to ensure they had the same start date as VIX. Next, as different features are having different time ranges, we are interested in those features' missing value ratios. Figure 6 shows the top ranked 6 features with the highest missing values ratio. We decided to drop the STEA Index and SPA Index since their missing percentage were larger than 80 percents.

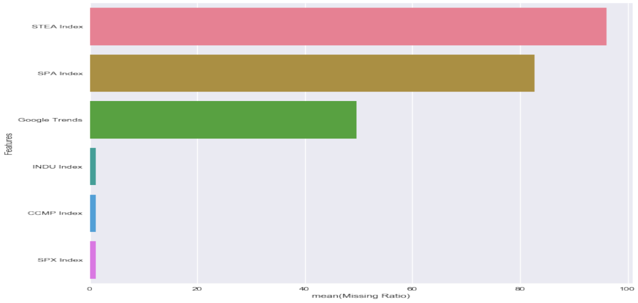

Figure 6. Missing Ratio

To handle the other missing values in various features, we chose to fill in numbers for these spots using two methods based on the models we run. When we are doing the linear model, we set the missing value to 0, equivalently, a zero return rate. When we are running our tree model, we set the missing value to a large number where the tree model will recognize the large number as a missing value.

1.6 Feature Engineering

Weekday and Month Features

We mainly did three things in this part. The first thing was that we thought the month and weekday may have influence on the VIX index, because at some specific days and festivals the stock exchange may suspend trading, changing investors' expectation on the risk.

One-hot Encoding

Based on this understanding, we use one-hot encoding method to generate variables indicating whether that day is certain weekday and is in certain month. After doing that, we can involve the month and weekday in the model.

Exponential Weighted Moving Average

Secondly, in order to involve historical information in the model and under the assumption that the historical data will be exponentially decay in the future, we conducted the Exponential Weighted Moving Average (EWMA) on each features and involved the new features in the model. We chose $\lambda = 0.8$. The formulas are as follows:

$$\bar{x_t} = w_t x_t + w_{t-1} x_{t-1} + w_{t-2} x_{t-2}$$

$$w_t = \frac{1-\lambda}{1-\lambda^3} \times 1, \quad w_{t-1} = \frac{1-\lambda}{1-\lambda^3} \times \lambda, \quad w_{t-2} = \frac{1-\lambda}{1-\lambda^3} \times \lambda^2$$

Lagged VIX Return Feature

Consistent with the idea of involving historical data, we also involved the return rate of VIX index on last day to improve the predicting accuracy of the model.

1.7 Feature Selection

After forming all the features we need, the next step is feature selection. We calculated the variance of each features and dropped the features with low variance. At this step, we dropped PAYNSA.

Besides, based on the current features, we trained a random forest model and ranked the features according to the attribute importance. We decided to select the features with attribute importance larger than 0.01.

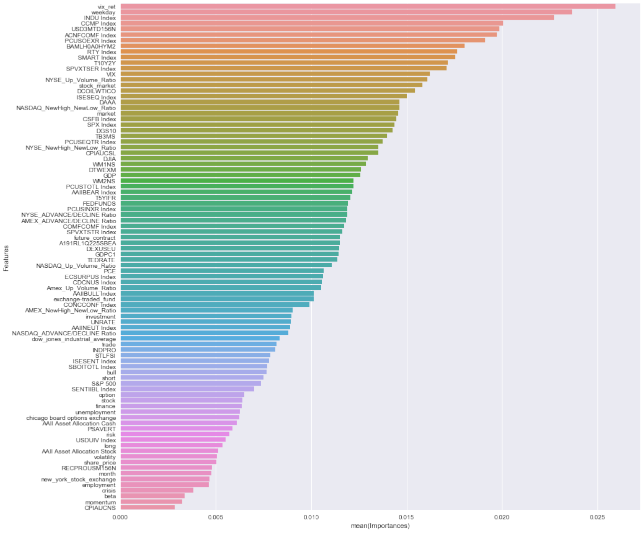

Figure 7. Feature Importance

2. Model Selection

2.1 Linear Models

In this part, we tried linear model with different loss functions and regularizers, and utilized validation to optimize parameters for each model. The first step is dividing the data set into training set and testing set. Because our data is time series data and we cannot use future data to predict the history, the cross validation is of no use in this scenario. We divided our data into three groups, 70% for training data, 15% for validation data, 15% for testing data according to the time order from the earliest to the latest.

The very first model we used was the most basic **linear regression** model with no regularizer.

$$\text{minimize} \quad \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

The model result can be served as a benchmark to evaluate on how other loss functions and regularization functions perform.

The second model we tried was **ridge regression**:

$$\text{minimize} \quad \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \sum_{i=1}^{n} {w_i}^2$$

Because we have many features, the ridge could help us prevent the problem of overfitting. Besides, when given very different input data, the ridge regression can prevent some ”crazy” prediction. By plotting out the MSE value based on different λ, we found the optimal $\lambda = 0.1$.

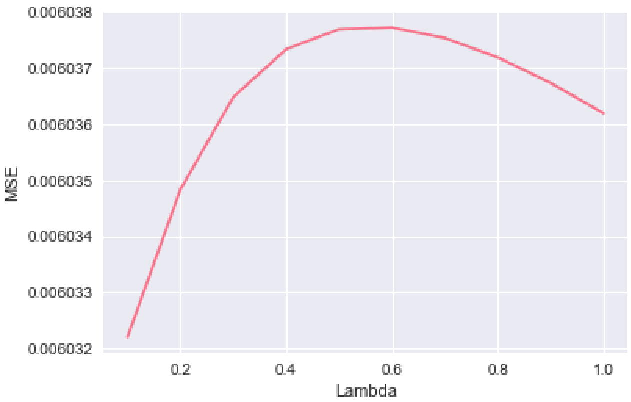

Figure 8. Select λ for Ridge Regression

We also tried **lasso regression**:

$$\text{minimize} \quad \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \sum_{i=1}^{n} |w_i|$$

The lasso regularizer usually tends to give a more sparse result. We can use this property to select an appropriate subset of features for the prediction, which increases the model accuracy as well as interpretability. Similarly, we found our optimal $\lambda = 0.001$ under the validation data set.

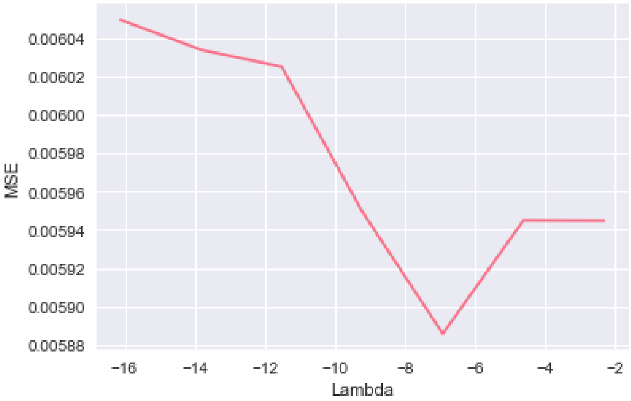

Figure 9. Select λ for Lasso Regression

The fourth linear model we used was the **huber regression**:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \mathbf{huber}(y_i - w^T x_i)$$

$$\mathbf{huber}(z) = \begin{cases} \frac{1}{2}z^2 & |z| \leq \varepsilon \\ |z| - \frac{1}{2} & |z| \geq \varepsilon \end{cases}$$

The huber loss is calculated as a combination of squared loss and absolute loss. The function is sensitive to small error while robust to large error to avoid the domination of outliers. We believe such loss function should have a good performance. The following plot shows how we choose the ε, λ values (x-axis for λ, y-axis for ε). After calculating the MSE with our validation dataset, we selected $\varepsilon = 1, \lambda = 0.001$.

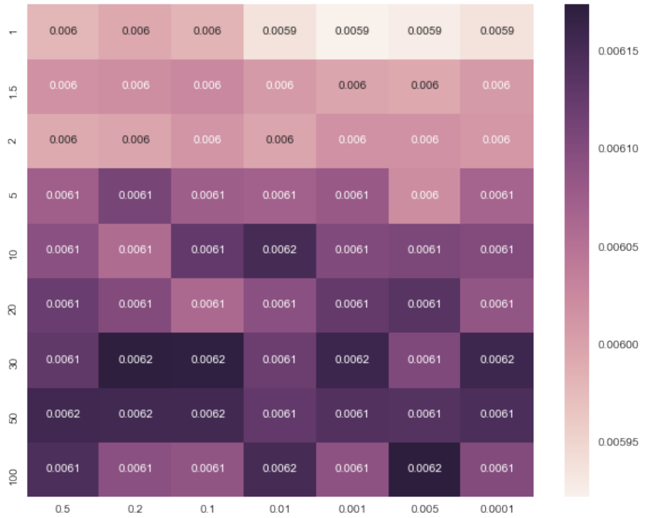

Figure 10. Select parameters for huber regression

2.2 Tree Models

In this project we also utilized random forest model to conduct prediction. Random forest is consist of a multitude of decision trees with boostrap samples from a dataset. In addition, it also randomly selects certain number of features for each tree. After the training process, when given an input data, each trees will give a prediction. Finally, the mean prediction of all decision trees is regarded as the output of the random forest model.

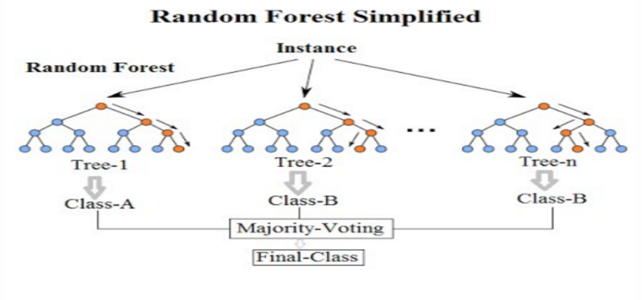

Figure 11. Random Forest Explanation [4]

We used validation to optimize the parameters. Firstly, we found the optimal tree number 900. Next, we chose to select 20% of the features for each tree which minimize the MSE. Based on these, we trained a optimal tree depth of 20.

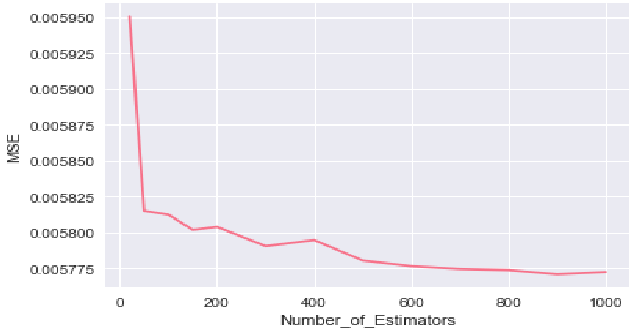

Figure 12. Tree Selection

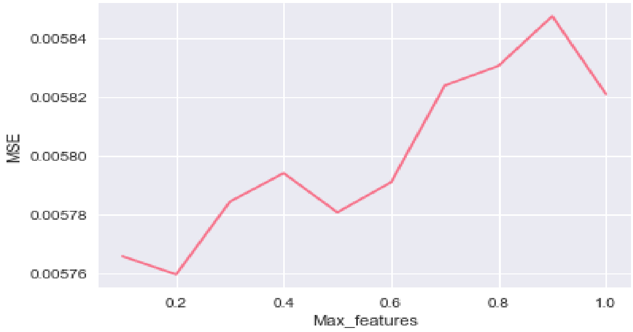

Figure 13. Feature Selection

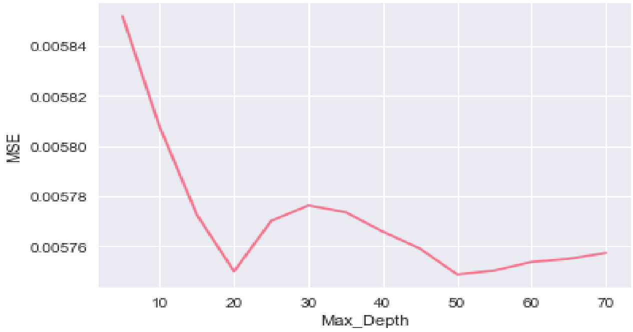

Figure 14. Depth Selection

3. Results

The result of the model can be seen in the Table 2. On the testing set, the random forest model has the lowest MSE. It is not surprising because random forest is better than other models to describe the non-linear relation between the input and output data. Clearly, the relation between features that we selected and the VIX return rate might be non-linear, so the random forest has the best prediction performance.

Table 2. Model Result

	MSE
Random Forest (n estimators=900, max features=0.2, max depth=20)	0.00575
Linear Regression	0.00605
Lasso(lambda=0.001)	0.00589
Ridge(lambda=0.1)	0.00603
Huber(eplison=1, lambda = 0.001)	0.00594

4. Trading Strategy

Suppose we could trade the VIX Index directly, we can construct a simple trading strategy which are as follows:

Table 3. Trading Strategy

Prediction Value	Position
Larger than 0.005	1
Between -0.005 and 0.005	0
Smaller than -0.005	-1

If the the prediction value for tomorrow's return is larger than 0.005, we will adjust our position to positive, that is, long the VIX Index. Similarly if the prediction value for tomorrow's return is smaller than 0.005, we will short the VIX Index. And if the prediction value lies in [-0.005,0.005], we will set our position to zero.

In the back-test program, we use 0.1% as the transaction cost and 0.1% as the impact cost. From the plot we find the return of this strategy is really high. It seems that we could earn large amount of money, doesn't it? The answer is 'NO', because actually we can not trade the VIX Index directly, we could only trade some derivatives of VIX Index, such as the ETF Futures.

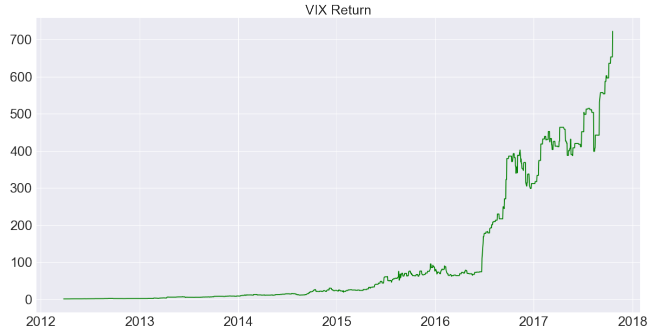

Figure 15. VIX Return

As a result, in this project we use one of most active ETF called VIXY as the real product to test our strategy. This ETF starts on 2013-10-28, so we re-split the training data from 2013-10-28 to 2015-06-28 and the testing data from 2015-06-29 to 2017-10-10. What's more, the target now is the return of VIXY instead of the return of VIX.

We still use our best model (random forest) to make prediction and apply the strategy described in Table 3. And the performance of the strategy on VIXY is shown in Figure 16 and 17.

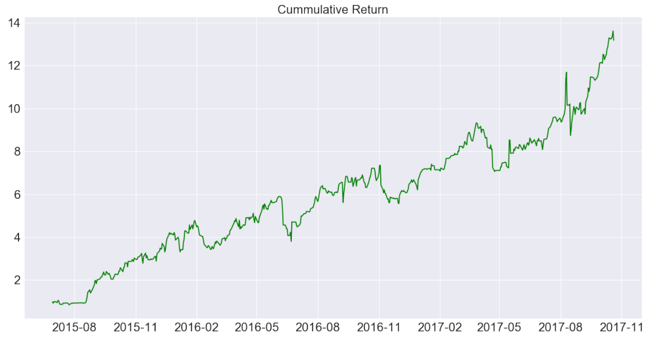

Figure 16. Cumulative Return

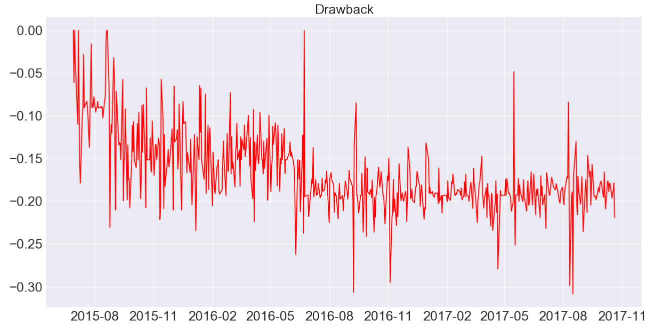

Figure 17. Drawdown

The annual return of this strategy is 214%, and the max drawdown is approximately 30%.

5. Conclusion

In this project, we utilized various data exploratory methods, data processing tools, as well as model fitting techniques introduced in class. For our linear regression models, we found our best linear model by fitting the data under different loss functions and regularizers, accompanied with the training of λ.

Our optimal model was trained with random forest. We tested this model performance by trading the ETF VIXY based on our prediction, with achieving a annual return of 214%, max 30% drawdown. Intuitively, we can try on more complex trading strategies based on our VIX prediction in the future to earn larger profits. In this project, we are limited to obtain the most complete and accurate data. There are some interesting sentiment features whose data cannot be obtained with our current available tools. Besides, if we successfully get the daily data of Google Trends, we could further improve the prediction ability of our model.

Despite some drawbacks, the methods of processing data and the models in our project can serve as a starting point to go forward. In this project, we successfully prove that predicting VIX index is possible and is helpful for developing profitable trading strategy. We believe that financial market practitioners can develop more accurate and feasible trading strategies based on our project.

References

[1] Investopedia. Vix - cboe volatility index, 2015. [Online; accessed 4-December-2017].

[2] Kiran Manda. Stock market volatility during the 2008 financial crisis, 2010. [Online; accessed 4-December-2017].

[3] Cobe. Cobe - vix options futures, 2017. [Online; accessed 4-December-2017].

[4] Venkata Jagannath. Random forest template for tibco spotfire® - wiki page, 2017. [Online; accessed 4-December-2017].