Stock Market Prediction with Reuters News Archive 2007 ~ 2018

Jingying (Jane) Bi | UCID: 12174556 | Emial: jingyingb@uchicago.edu

Abstract

This paper aims at predicting asset prices. It uses both *numbers* and *text* dataset with various statistical models such as linear regression, LASSO, partial least squares (PLS), principal component analysis/regression (PCA/R), CART Tree, random forest and averaging modeling. Except SP500 historical prices, CPI, and Fama-French 3 factors, all the dataset used in this paper were constructed by the author. Most *numbers* variables are newly derived from the prices, CPI, and so on. *Text* data were scraped from Reuters online news archive. 3,279,343 Reuters news were collected, which covers since Jan 1st, 2007 to Apr 29th, 2018. The results show that random forest has the best explanatory power, while CART Tree has the best predicting power with low out-of-sample R2 (OOS R2).

Keywords: stock price, textual analysis, big data, machine learning

1. Introduction

Predicting asset prices is always one of the major research topics in finance. One way to predict asset prices is to look for key *financial* factors which could explain the variances in asset prices. For instance, CAPM describes how the systematic risk, characterizing by market excess return, correlate with asset excess returns. Afterwards, in 1992 Fama and French proposed the well-known three factors, i.e. SML, HML, and market excess return. Nowadays, hundreds of thousand factors have been discovered, but most of them are only able to work well under certain conditions.

In addition to predict stock prices by using *financial* factors, as natural language processing gets more popular, scholars started to integrate textual dataset into prediction. Information about market sentiment is extracted from the text corpus, which makes *sentiment* another factor playing a role in predicting stock prices.

In this paper, I aim to build statistical models which could predict the stock prices at a certain level of out-of-sample accuracy. To achieve this goal, I first collected and cleaned the dataset through the following steps:

(1) Scraped 3,279,343 Reuters news since Jan 1st, 2007 to May 29th, 2018 from its archive.
(2) Processed the Reuters corpus via tokenization, stop-list words removal, punctual removal, and lemmatization. Finally, the words with the least and largest frequencies were removed. This processing reduces the dimensionality of the corpus from 30,729,641 to 24,857,489.
(3) Extracted sentiment information from the Reuters corpus via using three different popular dictionaries, namely AFINN, BING, and NRC. Using various dictionaries not only provides richer information about the market sentiment, but also guarantees a more robust analysis and results.
(4) In addition to the textual dataset, I also collected data of stock & index prices (i.e. APPL, SP500TR) from *Yahoo Finance* and Fama-French 3 factors posted online.

Next, I first split the dataset into *training* and *testing* sub-data. *Training* sub-data covers 2250 rows and *testing* sub-data covers 573 rows. Then, with the *training* sub-data, I pursued two different approaches to train statistical models. In Approach I, I took the whole tokenized corpus into consideration. The corpus was pooled into LASSO. Meanwhile, I applied principal component analysis (PCA) as an alternative way of reducing its dimensionality. Then I ran principal component regressions (PCR) on the important principal components (PC). In Approach II, I only

focused on the potential factors affecting stock prices, including Fama-French 3 factors (SML, HML, and excess market return), and market sentiment variables. These factors helped train linear regression model, LASSO, partial least squares (PLS), CART tree, and random forest. After training all the aforementioned models, I tested out-of-sample (OOS) explanatory power of the models by looking at OOS R2 and OOS deviance. In short, random forest always give the best explanatory power, while CART Tree and LASSO seem to have stronger predicting power.

The rest of paper is organized as follows: Section 2 discusses data collection and data cleaning. Section 3 presents the Approach I of model training and testing processes. Section 4 displays the Approach II of model training and testing processes. Conclusion is in Section 5.

2. Dataset Collection and Cleaning

2.1 Data Collection

I collected two types of dataset, namely *numbers* and *text*. The *numbers* dataset contains information of stock & index prices, returns, Fama-French 3 factors. It was collected from Yahoo Finance and Professor Famma's website. Table 2.1 shows an example of the *numbers* dataset. Text was scraped from Reuters historical archive from Jan 1st, 2007 to Apr 30th, 2018. The scraping code was written in Python script. I ran the code on midway terminals from May 17th to June 5th, 2018. Finally, 3,279,343 of news has been collected.

Table 2.1: An example of *numbers* dataset

	DATE	SMB	HML	Mkt_RF	RF	Open	High	Low	Close	Adj_Close	Volume	simple_return
0	2007-01-04	0.24	-0.51	0.16	0.022	1416.599976	1421.839966	1408.430054	1418.339966	1418.339966	3004460000	0.001228
1	2007-01-05	-0.91	-0.33	-0.73	0.022	1418.339966	1418.339966	1405.750000	1409.709961	1409.709961	2919400000	-0.006085
2	2007-01-08	-0.07	0.08	0.24	0.022	1409.260010	1414.979980	1403.969971	1412.839966	1412.839966	2763340000	0.002220
3	2007-01-09	0.28	-0.20	0.00	0.022	1412.839966	1415.609985	1405.420044	1412.109985	1412.109985	3038380000	-0.000517
4	2007-01-10	-0.08	-0.17	0.23	0.022	1408.699951	1415.989990	1405.319946	1414.849976	1414.849976	2764660000	0.001940
5	2007-01-11	0.55	-0.29	0.74	0.022	1414.839966	1427.119995	1414.839966	1423.819946	1423.819946	2857870000	0.006340
6	2007-01-12	0.21	-0.28	0.50	0.022	1423.819946	1431.229980	1422.579956	1430.729980	1430.729980	2686480000	0.004853
7	2007-01-16	-0.26	0.06	0.00	0.022	1430.729980	1433.930054	1428.619995	1431.900024	1431.900024	2599530000	0.000818

2.2 Data Cleaning

After scraping all the news headlines, I firstly tokenized each headline and removed the words on the stop list. Then the punctuations were also removed. Afterwards, the rest of tokens

were lemmatized. These three steps reduced the dimensionality of the raw corpus form 30,729,641 to 24,857,489. Finally, I transform all the words into lower case.

2.3 Data Construction

I constructed one *numbers* dataset and three *text* datasets.

- **Numbers dataset: *Stock_and_Index.csv***

This is a table containing the Fama-Frech 3 factors (SML, HML, excess market return), stock and index prices (i.e. high, low, open, close, adjusted close prices), and excess returns. In this paper, I chose AAPL and SP500 to study how the models fit and predict the stock prices and index prices, respectively. Table 2.1 shows the structure of this table.

- **Text Dataset I: *Word_List.csv***

This is a list of alphabetically-ordered word list containing all the English words which have ever appeared in Reuters news headlines since Jan 1st, 2007 to Apr 30th, 2018. Specifically, in the cleaned data in Section 2.2, there were still some tokens not identified as English words. To make sure that I only keep meaningful tokens, I installed three English dictionaries, namely *word* and *wordnet* packages of *nltk.corpus* in python, and an online English words list. Only the tokens appearing on all three of English dictionaries could be kept on the *Word_List.csv*. This list contains 23,508 English words. Table2.2 shows an example of *Word_List.csv*.

Table 2.2: An example of *Word_List.csv*

	Word_ID	Word
0	0	aardvark
1	1	aback
2	2	abacus
3	3	abalone
4	4	abandon
5	5	abandoned
6	6	abandonment
7	7	abase

Table 2.3: Structure of *Word_Frequency.csv*

	Date	Date_ID	Word_ID	Word_Freq	Word
0	2007-01-04	1	6390	1	due
1	2007-01-04	1	15712	2	prior
2	2007-01-04	1	18576	1	signal
3	2007-01-04	1	8964	1	grain
4	2007-01-04	1	6466	1	eagle
5	2007-01-04	1	16701	4	reform
6	2007-01-04	1	8140	1	force
7	2007-01-04	1	22703	3	visit

- **Text Dataset II: *Word_Frequency.csv***

Based on the *Word_List.csv*, I counted the word frequencies on a daily basis. **Table 2.3** displays the structure of *Word_Frequency.csv.*

- **Text Dataset III: *Sentiment_Factor.csv***

This table contains the sentiment measures of daily news headlines. To enrich the sentiment understanding and guarantee a more accurate and robust results, I extracted market sentiment from headlines by applying four dictionaries, namely textBlob, AFINN, BING and NRC. Specifically, textBlob is a python package. Given a bag of words as input, it will return a polarity value indicating the sentiment. The polarity ranges from -1.0 to 1.0. Negative polarity value represents negative sentiment. AFINN is a dictionary assigning each word a sentiment score ranging from -5.0 to 5.0. BING is a dictionary categorizing each word as binary "negative" or "positive". NRC is a leading lexicon curated by National Research Council Canada, which consists of a comprehensive list of ~140,000 English words. NRC dictionary associates each word with one of ten emotions, including *anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise* and *trust*. Next, I will talk about the processing with each of dictionaries AFINN, BING, and NRC.

Dictionary 1: AFINN

For each day, I went through all the tokens, and assigned tokens positive or negative values according to AFINN dictionary, given that the tokens were on the AFINN list. Those words off AFINN were recorded as missing words. Then I sum up negative values of all negative words to get the negative score for that day. I got the positive score for that day through a similar way. Then I calculated the compounded score by summing up negative and positive scores. Finally, I normalized the both negative and positive scores by dividing the sum of negative word number and positive word number. **Table 2.4** shows an example of my AFINN analysis.

Table 2.4: An example of *AFINN* structure

	Date	Positive_score	Negative_score	Positive_words	Negative_words	Missing_words
0	2007-01-04	270	-304	straight share best help cool top peace growth...	crush warn alone cut cut crisis no regret pay ...	leafs score nine goal bruins shareholder vodaf...
1	2007-01-05	333	-368	warm successful commits boosting romance peace...	suicide cut cut warn poor kill resign dead cut...	singh move one ahead wet windy kapalua some do...
2	2007-01-08	478	-557	fresh boost help growth hope ease big big posi...	unhappy lonely lonely pressure risk collide in...	press digest washington post business jan iraq...
3	2007-01-09	302	-665	resolve top interest top fame vitamin solid st...	debt infringement drag flu miss disaster disas...	update file patent suit china give hk pandas m...
4	2007-01-10	425	-647	justice share boost chance big awards expands ...	murder poor death disaster disaster drop criti...	press digest financial times jan india pantalo...
5	2007-01-11	397	-619	share comedy peacefully marvel boost success f...	battle injury worry dead miss weakness anti no...	japan topix rise pct tech bank factbox players...

Dictionary 2: BING

For each day, I went through all the tokens, and classified tokens as *positive* or *negative* according to BING dictionary, given that the tokens were on the BING list. Those words off BING list were recorded as missing words. I counted the number of negative and positive words. I calculated the compounded score by subtracting the number of negative words from the number of positive words. Then, I normalized the compounded score by dividing the sum of the negative words number and positive words number. **Table 2.5** gives an example of my BING analysis.

Table 2.5: An example of *BING* structure

	Date	Positive_num	Negative_num	Missing_num	Positive_words	Negative_words	Missing_words
0	2007-01-04	141	222	4193	best cool blossom top peace gain best portable...	crush fall stigma crisis regret slow debt fall...	leafs score nine straight goal bruins sharehol...
1	2007-01-05	166	253	3461	warm successful intelligence lead renewed peac...	mistakenly tentative sue suicide poor kill rad...	singh move one ahead wet windy kapalua some do...
2	2007-01-08	207	340	5322	fresh boost wonder steady ease positive positi...	resistance resistance unhappy bleeds lonely fl...	press digest washington post business jan iraq...
3	2007-01-09	169	425	5185	lead top top idol fame lighter solid strong to...	debt infringement fall cheap drag miss disaste...	update resolve file patent suit china give hk ...
4	2007-01-10	198	381	5787	boost awards boost tranquil best top guarantee...	murder poor death fall grim disaster disaster ...	press digest financial times jan india pantalo...

Dictionary 3: NRC

Table 2.6: An example of *NRC* structure

	Date	Anger_num	Anticipate_num	Disgust_num	Fear_num	Joy_num	Negative_num	Positive_num	Sadness_num	Surprise_num	Trust_num
0	2007-01-04	142	0	10	61	37	120	215	5	10	56
1	2007-01-05	167	0	18	74	34	131	241	0	16	63
2	2007-01-08	226	0	25	137	41	120	325	5	24	58
3	2007-01-09	247	0	34	139	38	179	241	3	16	104
4	2007-01-10	282	0	27	133	53	156	338	3	15	124
5	2007-01-11	259	0	27	154	54	173	331	5	13	135

For each day, I went through all the tokens, and classified tokens as *anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise* or *trust* according to NRC dictionary, given that the tokens were on the NRC list. Those words off NRC list were recorded as missing words. Then I calculated the compounded numbers of words in each category. I standardized the compounded numbers by dividing total number of words (excluding missing words). **Table 2.6** gives an example of my NRC analysis.

Finally, I combined all the sentiment scores in the *Sentiment_Factor.csv* as shown in **Table 2.7**. Last but not list, the time frames of dataset *Stock_and_Index.csv, Word_Frequency.csv, and Sentiment_Factor.csv* matched with one another through variable "Date".

Table 2.7: An example of *Sentiment_Factor.csv* structure

	TextBlob	AFINN_Positive	AFINN_Negative	BING_Positive	BING_Negative	NRC_Anger	NRC_Disgust	NRC_Fear	NRC_Joy	NRC_Negative	NRC_Positive
0	0.054109	0.828	-0.933	0.388	0.612	0.216	0.015	0.093	0.056	0.183	0.328
1	0.077918	0.854	-0.944	0.396	0.604	0.224	0.024	0.099	0.046	0.176	0.324
2	0.048532	0.872	-1.016	0.378	0.622	0.235	0.026	0.143	0.043	0.125	0.338
3	0.039919	0.812	-1.788	0.285	0.715	0.247	0.034	0.139	0.038	0.179	0.241
4	0.045828	0.878	-1.337	0.342	0.658	0.249	0.024	0.118	0.047	0.138	0.299
5	0.045266	0.820	-1.279	0.365	0.635	0.225	0.023	0.134	0.047	0.150	0.288

3. Approach I: Text Corpus – Word Frequency

Firstly, I divided the dataset into *training* and *testing* sub-data. *Training* sub-data contains 2250 rows from 2007-01-04 to 2015-12-21. *Testing* sub-data contains 573 rows from 2015-12-22 to 2018-04-30. I made two approaches to training the model. The first approach only considers about text corpus, i.e. how the single word frequencies in headlines correlate with stock or index's excess return. The second approach considers about factors, including Fama-French 3 factors (i.e. SME, HML, and excess market return), and sentiment factors curated in Section 2.3. I applied LASSO, linear PCR, and LASS PCR in the first approach. The second approach was to use linear, LASSO, PLS, CART tree, random forest, and these PCR on them.

In approach one, I pooled whole corpus into models by transferring *Word_Frequency.csv* into a sparse matrix. The dimensionalities of training sparse matrix and testing sparse matrix are 2250 x 23,509 and 573 x 23,509, respectively. Due to the super large dimension of the sparse matrix, I chose models which could reduce the dimension of covariates, namely LASSO and PCA.

3.1 LASSO

Figure 3.1 displays LASSO regression of excess return on the whole corpus. LASSO only keeps 52 words which have non-zero effect on return variances. Table 3.1 shows words having the largest effect, either negative or positive, on return variances. Such words are "explicitly", "demigod", "equalization", "desensitization", "dubs", and so on. However, there is little story by merely looking at these single words. The in-sample R2 (IS R2) is 0.165 which is relatively small

and indicates a relatively weak explanatory power of non-zero coefficient words to explain the return variances.

Figure 3.1: LASSO of WordFreq on SP500 return

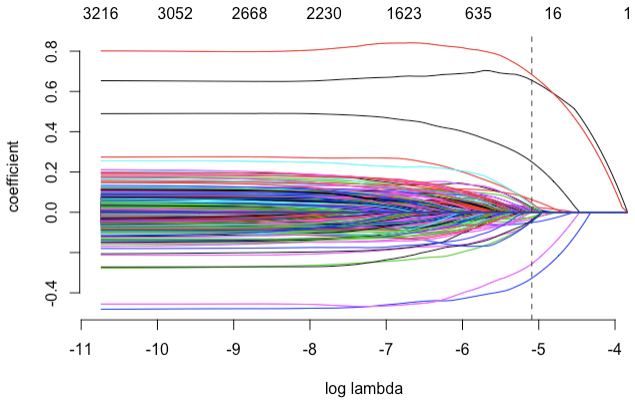
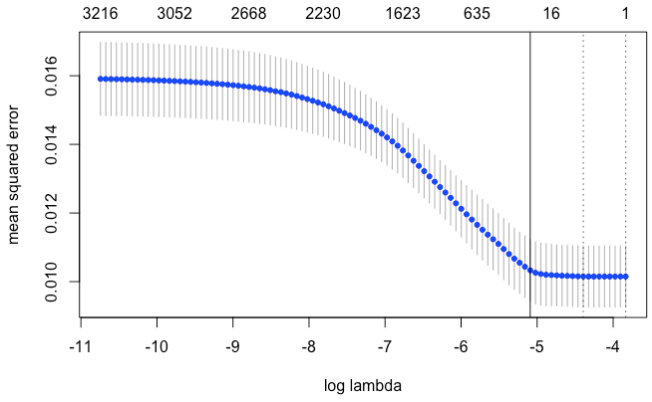


Table 3.1: Words of largest effect

	Word_ID	Word	Effect
explicitly	7339	explicitly	0.6834017
demigod	5410	demigod	0.6547955
equalization	7012	equalization	-0.3275262
desensitization	5544	desensitization	-0.2566792
dubs	6381	dubs	0.2502804
lifesaver	11758	lifesaver	0.0622516
incapable	10317	incapable	-0.0507256
stepwise	19682	stepwise	-0.0457569
trapper	21288	trapper	-0.0453814
bigamous	1980	bigamous	0.0449875
expressionism	7367	expressionism	-0.0447292
proprietorship	15881	proprietorship	-0.0387358
enamel	6792	enamel	0.0373341
tinsel	20983	tinsel	-0.0329104
pear	14662	pear	0.0260011
unimaginable	21999	unimaginable	-0.0247666
cloche	3736	cloche	0.0241748
buckskin	2678	buckskin	-0.0227421

To find the best lambda which minimizes the mean squared error (MSE), I ran the cross-validation as displayed in Figure 3.2. The dashed line on the left corresponds to the lambda minimizing MSE. The dashed line on the right corresponds to the lambda within the 1 standard error from the minimizing lambda. Moreover, the solid line on the left most corresponds to the lambda selected by AICc. Figure 3.2 tells that the three lambda candidates are close to each other. Hence both cross-validation and information criteria had similar performance in this case. I picked the lambda minimizing MSE, yielding the result of cross-validation.

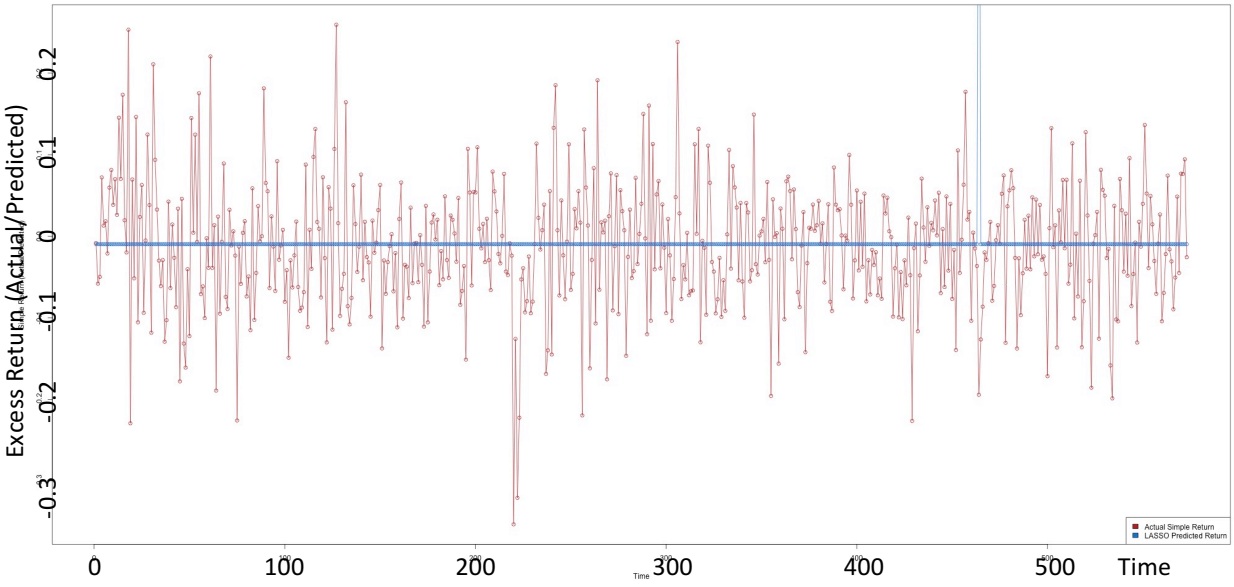
Figure 3.2: Cross-validation of LASSO



Next, I tested the training LASSO model with the *testing* dataset. Figure 3.3 shows the result. Red line is actual excess return. Blue line is the poor LASSO predicted excess return. The predicted values are too small to be seen any fluctuations. This might be due to the nature of LASSO regression, as LASSO punishes large coefficients. Hence when the coefficient values are regularized, the predicted values tend to be lower as well. Since the predicting power is so poor, there is no doubt that the out-of-sample R2 (OOS R2) is as negative as -6.123. This indicates that LSSO model performed even worse than the one merely using the mean of history excess returns to predict the future.

In order to improve LASSO's performance, I added one more variable, i.e. previous return, to the sparse matrix. Although this lead to a larger IS R2 which was 0.252, the OOS R2 was still as negative as -4.355.

Figure 3.3: SP500 excess return prediction by LASSO
Red line is the actual excess return; blue line is the predicted value.



3.2 PCR

Since PCA collages highly-correlated variables into one component. I firstly inspected the correlation among different words. Due to the dimension issue, Figure 3.4 only displays the heat plot of the correlation matrix containing words with frequencies from 92 to 95. There are 67 words. Next, I obtained the principal components displayed in Figure 3.5. Obviously, principal component 1 (PC1) explains the largest variances.

Figure 3.4: Correlation of word occurrences
Only containing words with frequencies
from 92 to 95

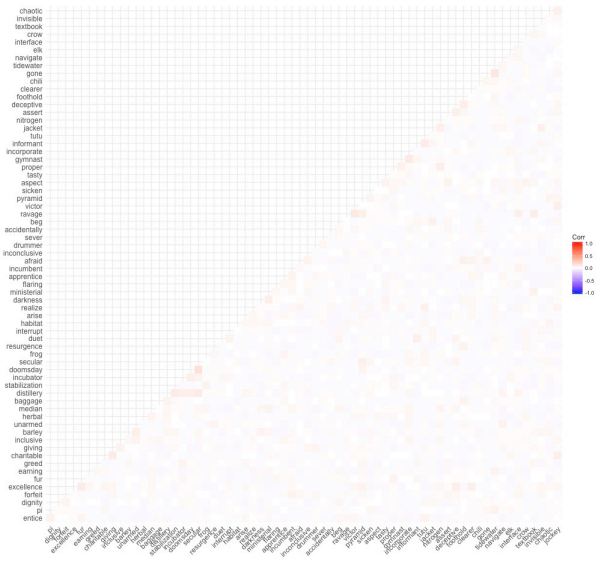
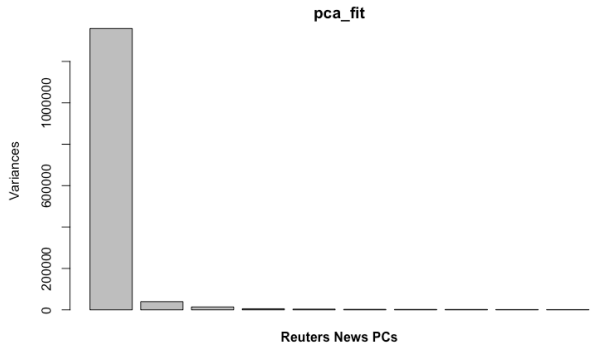
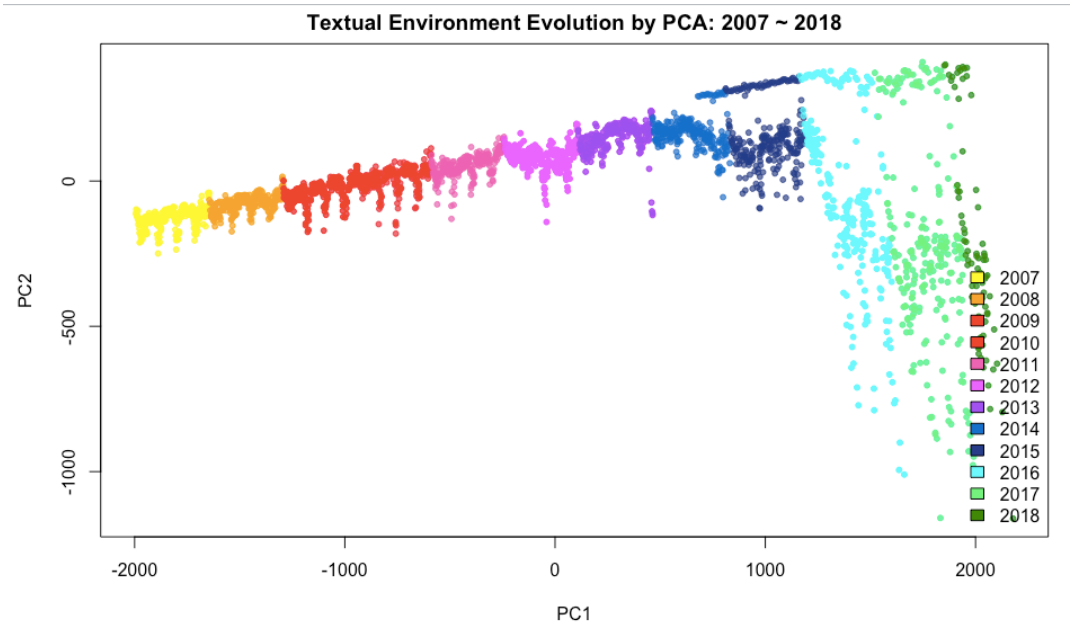


Figure 3.5 PC & variance



Next, I predicted the rotates of PCs for each word and plotted them on Figure 3.6 by year. Interestingly, there is a sequential order according to years. Both PC1 and PC2 values gradually increased over the years. Years 2006 – 2008 had a wider range of PC2 values as well. However, since principal component is like a black box containing and synthesizing the raw variables, it is hard to trace the story or intuition behind such trend.

Figure 3.6: Textual Environment Evolution by PCA: 2007 ~ 2018



As mentioned before, PCA helped reduce the dimension of dataset significantly. With the cutting dataset, I ran linear regression and LASSO regression to see whether the LASSO performance improved or not. To run PCR with enough information inherent in the dataset, I picked up the first 200 PCs. With these 200 PCs, I used AICc and BIC to find the best number of PCs for a linear regression of excess returns on factors. Figure 3.7 shows the results. Both AICc and BIC recommended that PC1 is good enough to model the linear regression. However, the performance of PC1 was poor. IS R2 was only 0.000473. Its prediction power is even worse as shown in Figure 3.8, and unsurprisingly, followed by a huge negative OOS R2 -102.679.

Figure 3.7: Select the best number of Factors in linear regression by AICc and BIC

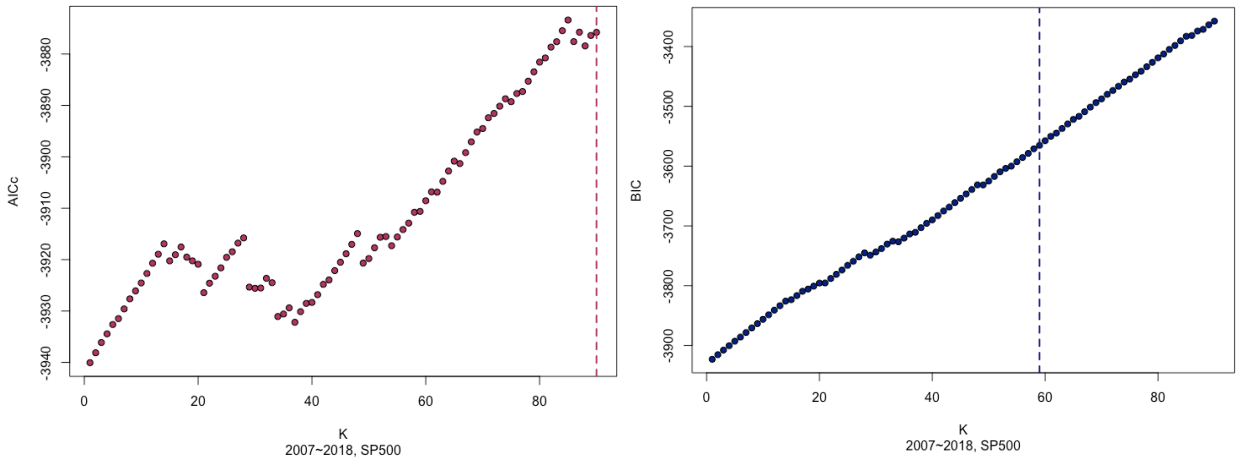
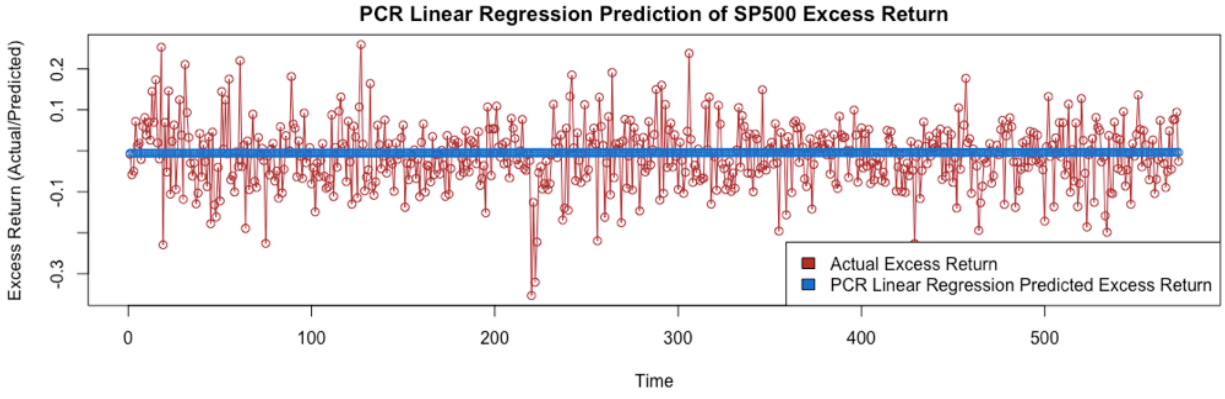


Figure 3.8: SP500 excess return prediction by PCR-linear



Then I ran LASSO on those top 200 PCs, filtering out 25 non-zero PCs and yielding the IS R2 of 0.0314. Figure 3.9 shows LASSO and cross-validation process. The predicting performance is shown in Figure 3.10, followed by OOS R2 -772.247.

Figure 3.9: LASSO and cross-validation process

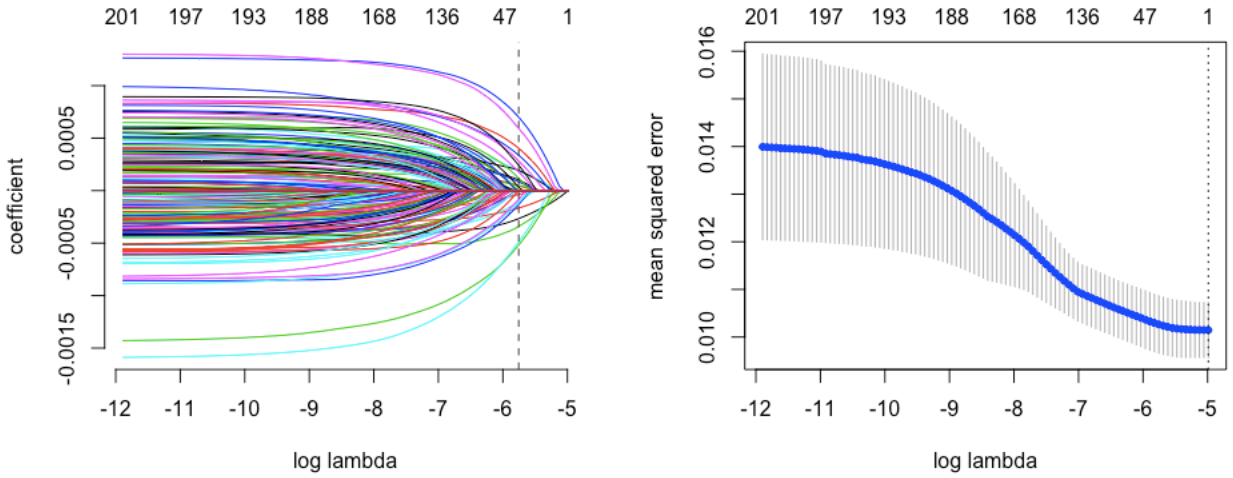
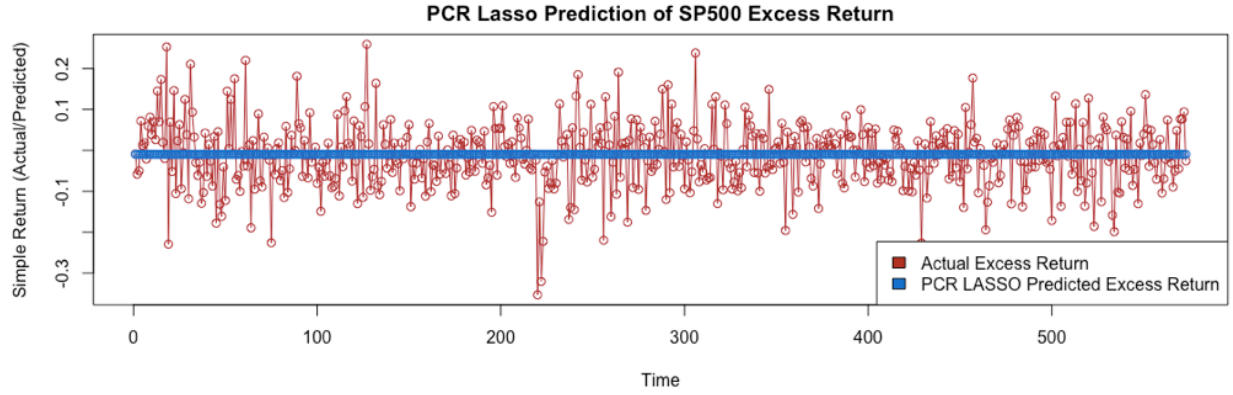


Figure 3.10: SP500 excess return prediction by PCR-lasso



3.3 Summary

In short, I chose LASSO, PCR of linear and PCR of LASSO to predict the SP500 excess return by using the sparse matrix of words from Reuters News headlines. None of these models did well in either fitting or predicting excess returns. The reasons might be that merely using raw tokens from headlines was costly and potentially lost much information. After all, single words carried so little useful information in fitting or predicting excess returns. No need to mention that there were thousands of hundreds single words mixing in one dataset and many of them may make more noise than contribution to any explanatory or predicting power of the models.

Therefore, I resorted to a wiser way to extract and compound the information from the tokens. I summarized the information as different sentiment factors which were used in combine with Fama-French 3 factors. More detailed discussion is in subsection 3.2.

4. Approach II: A Set of Factors

In this subsection, I will focus on the factors, including Fama-French 3 factors (SMB, HML, and market excess return) and sentiment variables. There are totally nineteen variables, namely SMB, HML, Mkr_RF, TextBlob, AFFIN_Positive, AFIINN_Negative, BING_Positive, BING_Negative, NRC_Anger, NRC_Disgust, NRC_Fear, NRC_Joy, NRC_Negative, NRC_Positive, NRC_Sadness, NRC_Surprise, NRC_Trust and two time trending variables Year and Month. Since the dimension is handful, I applied various models to predict the SP500 excess return. Table 4.1 show the structure of all independent variables.

Table 4.1: Key factors

	SMB <dbl>	HML <dbl>	Mkt_RF <dbl>	TextBlob <dbl>	AFINN_Positive <dbl>	AFINN_Negative <dbl>	BING_Positive <dbl>
1	0.24	-0.51	0.16	0.054109278	0.828	-0.933	0.388
2	-0.91	-0.33	-0.73	0.077918201	0.854	-0.944	0.396
3	-0.07	0.08	0.24	0.048532070	0.872	-1.016	0.378
4	0.28	-0.20	0.00	0.039919058	0.812	-1.788	0.285
5	-0.08	-0.17	0.23	0.045828309	0.878	-1.337	0.342
6	0.55	-0.29	0.74	0.045266138	0.820	-1.279	0.365
7	0.21	-0.28	0.50	0.036916285	0.825	-1.323	0.352
8	-0.26	0.06	0.00	0.028434526	0.833	-1.608	0.313
9	-0.21	-0.05	-0.14	0.068254312	0.849	-1.201	0.357
10	-1.07	0.53	-0.48	0.054434921	0.837	-1.008	0.368

BING_Negative <dbl>	NRC_Anger <dbl>	NRC_Disgust <dbl>	NRC_Fear <dbl>	NRC_Joy <dbl>	NRC_Negative <dbl>	NRC_Positive <dbl>
0.612	0.216	0.015	0.093	0.056	0.183	0.328
0.604	0.224	0.024	0.099	0.046	0.176	0.324
0.622	0.235	0.026	0.143	0.043	0.125	0.338
0.715	0.247	0.034	0.139	0.038	0.179	0.241
0.658	0.249	0.024	0.118	0.047	0.138	0.299
0.635	0.225	0.023	0.134	0.047	0.150	0.288
0.648	0.241	0.031	0.140	0.061	0.132	0.298
0.687	0.249	0.022	0.160	0.041	0.146	0.260
0.643	0.234	0.027	0.125	0.035	0.149	0.302
0.632	0.208	0.025	0.107	0.051	0.152	0.324

NRC_Sadness <dbl>	NRC_Surprise <dbl>	NRC_Trust <dbl>	Year <int>	Month <int>
0.008	0.015	0.085	2007	1
0.000	0.022	0.085	2007	1
0.005	0.025	0.060	2007	1
0.003	0.016	0.104	2007	1
0.003	0.013	0.110	2007	1
0.004	0.011	0.117	2007	1
0.008	0.013	0.076	2007	1
0.007	0.004	0.111	2007	1
0.002	0.018	0.107	2007	1
0.004	0.019	0.109	2007	1

4.1 Linear Regression

I started from linear regression by regressing excess returns on the nineteen factors. Table 4.2 shows the regression results.

Table 4.2: Linear regression results of excess returns on key factors

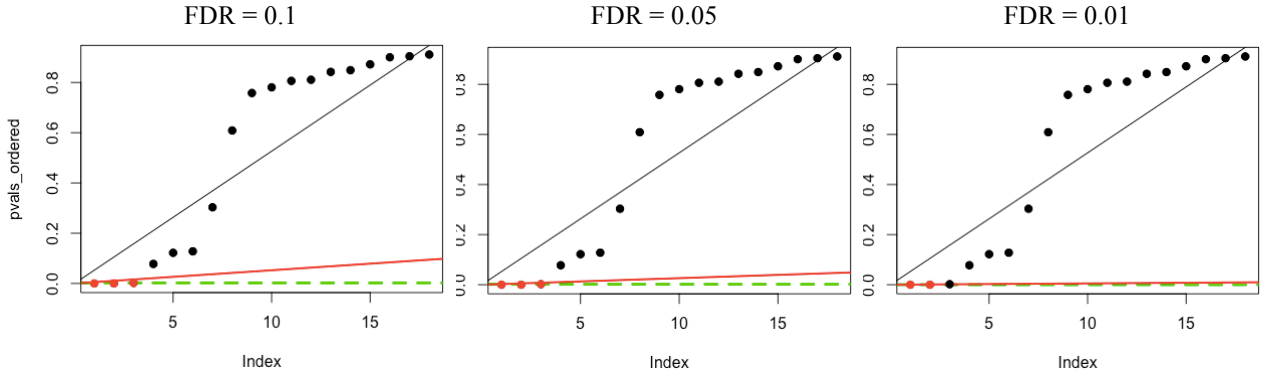
	Dependent variable:		
	Y_1_SP500_training		
	Fama-Frech 3 Factors	Full Model	Cut Model
	(1)	(2)	(3)
SMB	-0.138***	-0.138***	-0.138***
	(0.002)	(0.002)	(0.002)
HML	-0.003*	-0.003*	-0.003*
	(0.002)	(0.002)	(0.002)
Mkt_RF	0.006***	0.008***	0.007***
	(0.001)	(0.001)	(0.001)
TextBlob		0.090	
		(0.087)	
AFINN_Positive		0.010	
		(0.032)	
AFINN_Negative		-0.001	
		(0.009)	
BING_Positive		-0.167***	-0.108***
		(0.054)	(0.031)
BING_Negative			
NRC_Anger		0.336	
		(1.404)	
NRC_Disgust		0.177	
		(1.412)	
NRC_Fear		0.268	
		(1.407)	
NRC_Joy		0.391	
		(1.404)	
NRC_Negative		0.227	
		(1.407)	
NRC_Positive		0.345	
		(1.405)	
NRC_Sadness		0.749	
		(1.464)	
NRC_Surprise		0.158	
		(1.420)	
NRC_Trust		0.280	
		(1.404)	
Year		0.001	
		(0.001)	
Month		0.001	
		(0.0004)	
Constant	-0.009***	-2.225	0.031***
	(0.001)	(1.903)	(0.012)
Observations	2,250	2,250	2,250
Log Likelihood	3,210.067	3,223.035	3,216.084
Akaike Inf. Crit.	-6,412.134	-6,408.069	-6,422.168
Note:		*p<0.1; **p<0.05; ***p<0.01	

In column (1), only Fama-French 3 factors were considered. All of them were statistically significant, especially SMB not only significant but also imposing a relatively large effect on the

variance of excess returns. Column (1) regression has IS R2 0.667. Next, I added the sentiment factors onto the base Fama-Frech 3 factor model. Column (2) shows that BING_Positive is statistically significant and of a large effect. It has negative coefficient -0.167, which means that when the percentage of positive words increased by 1 unit, excess returns decreased by 0.167 percent. However, it is still hard to build up the negative correlation between BING_Positive and excess returns. This is because these positive words were not necessarily associate with positive situations in financial market. Instead, such positive words reflect a more broad "positive" sentiment or merely people's positive reviews and opinions. Column (2) has IS R2 0.671 which is slightly higher that column (1) base model. Since column (2) shows that only four variables are statistically significant, I re-ran the regression exclusively onto these four variables, as shown in column (3). The IS R2 is 0.669

Next, to restrict the false discovery rate (FDR) within a certain range, I conducted FDR analysis to filter out those "truly" significant variables. Figure 4.1 displays the FDR results corresponding to different FDR levels, namely 0.1, 0.05, and 0.01. For FDR at level q= 0.01, the number of tests that are significant is 2. The p-value cutoff for FDR at level q= 0.01 is 1.499e-12. Although FDR test only filters out two important variables, I would still keep all nineteen variables to get a complete understanding about each variable's performance.

Figure 4.1: FDR of linear regression



I further tested the predictions of both full model and cut model. Both predictions seem to be much better than those in subsection 3.1. Figure 4.2 displays the actual and predicted excess returns of the cut model, with an OOS R2 0.564. In contrast, the full model prediction only has OOS R2 0.559, as shown in Figure 4.3. Hence, removing the redundant variables helps increase the predicting power in this case.

Figure 4.2: SP500 excess return prediction by linear regression (only FF3)

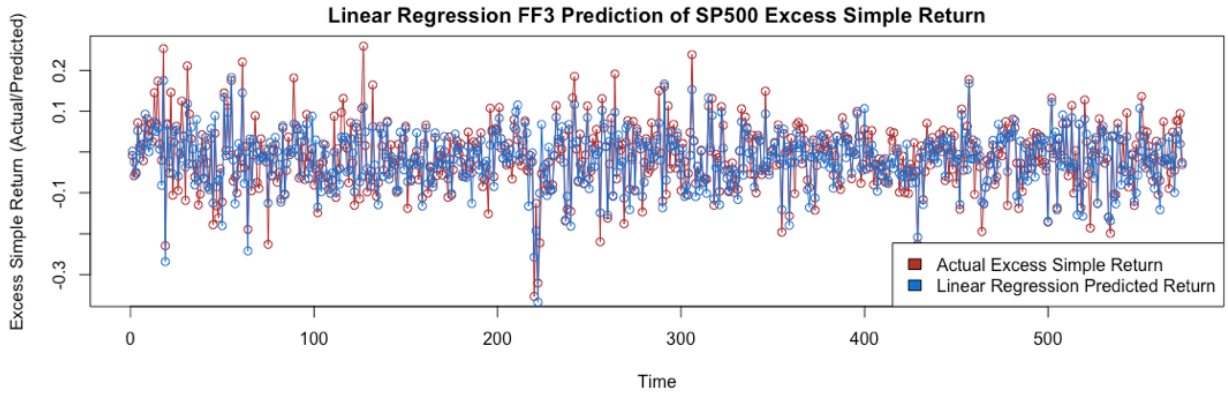
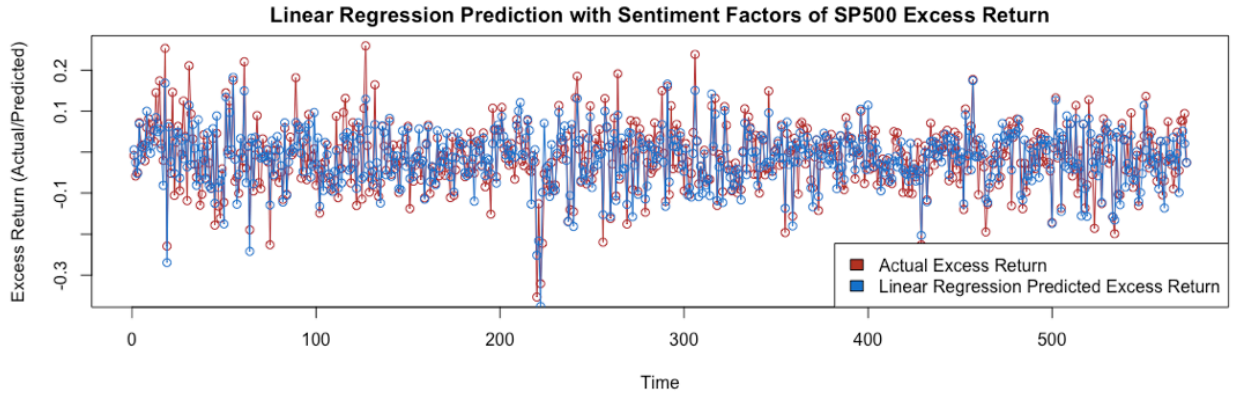


Figure 4.3: SP500 excess return prediction by linear regression (+ sentiment)



4.2 LASSO

With all nineteen variables, I ran a LASSO model as shown in Figure 4.4. The fitting has a good IS R2 0.670. LASSO model selected 10 non-zero betas, namely NRC_Sadness, SMB, BING_Positive, NRC_Negative, NRC_Disgust, NRC_Anger, Mkt_RF, HML, Year and Month.

Figure 4.4L LASSO regression

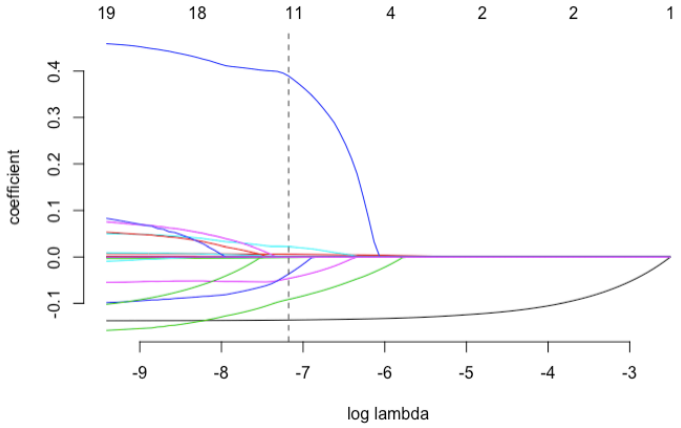


Table 4.3: Regression oefficients

	Effect
intercept	-0.4658557
NRC_Sadness	0.3894386
SMB	-0.1358852
BING_Positive	-0.0916134
NRC_Negative	-0.0464801
NRC_Disgust	-0.0367608
NRC_Anger	0.0219533
Mkt_RF	0.0060100
HML	-0.0011739
Year	0.0002442
Month	0.0002426

From Table 4.3, we see that NRC_Sadness and SMB has the largest positive and negative effects on the variances of excess return. Interestingly, similar to the previous counter-intuition where more negative words brought higher excess return, here, keep everything same, when the percentage of words with sad sentiment increased by 1 unit, there is 0.38 percentage increase in excess returns. However, there is no guarantee that all these 10 non-zero betas are statistically significant.

Figure 4.5: Cross-validation and LASSO model

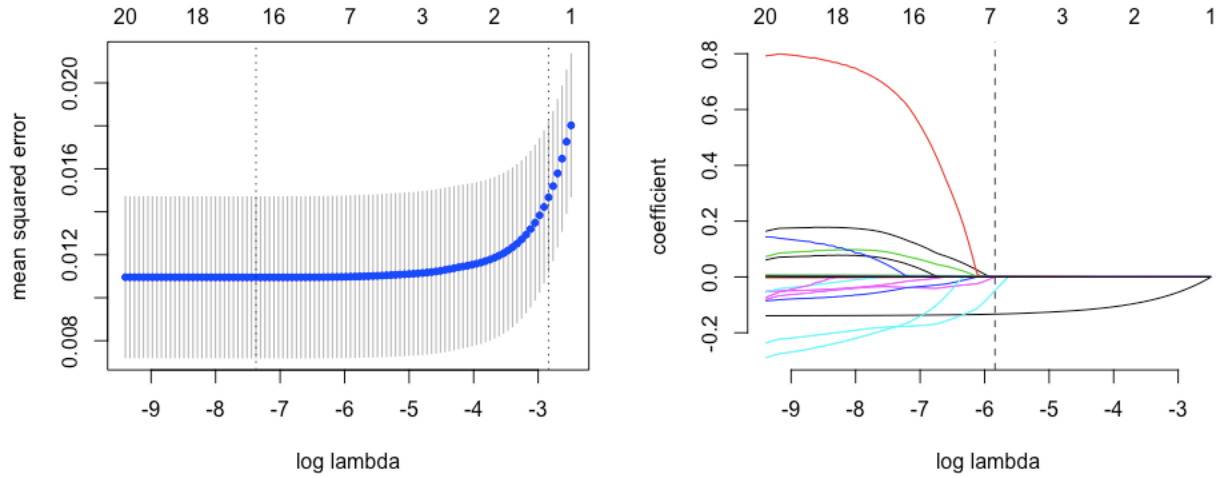
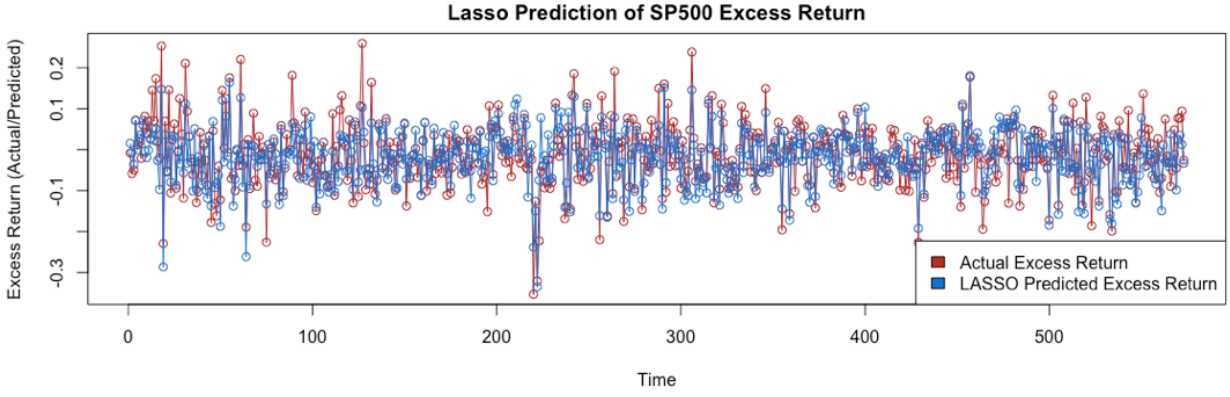


Figure 4.5 displays the cross-validation results and the LASSO model with the lambda minimizing the MSE. With this training model, I used the testing dataset to predict the excess returns and compared the predicted values with the actual values in Figure 4.6. The OOS R2 is 0.518.

Figure 4.6: SP500 excess return prediction by LASSO



4.3 PLS

When I fitted the dataset with PLS model, I looped the fitting process with differ number of PLS desired directions. I selected the best K with largest IS R2 and adjusted R2, as shown in Figure 4.7. Both criteria agreed with K = 4 as ideal choice. Therefore, I trained the PLS model with K = 4 and the result is displayed in Figure 4.8. By the time of completing the calculation of forth direction, the correlation between PLS fitted values and actual observations reached as high as 0.82. Moreover, the IS R2 is 0.509, indicating a fairly satisfying performance. To test the model's predicting power, I predict excess returns on the *testing* dataset. The result is in Figure 4.9. PLS has OOS R2 to be 0.566.

Figure 4.7: Find the best K for PLS regression by R2 and Adjusted R2

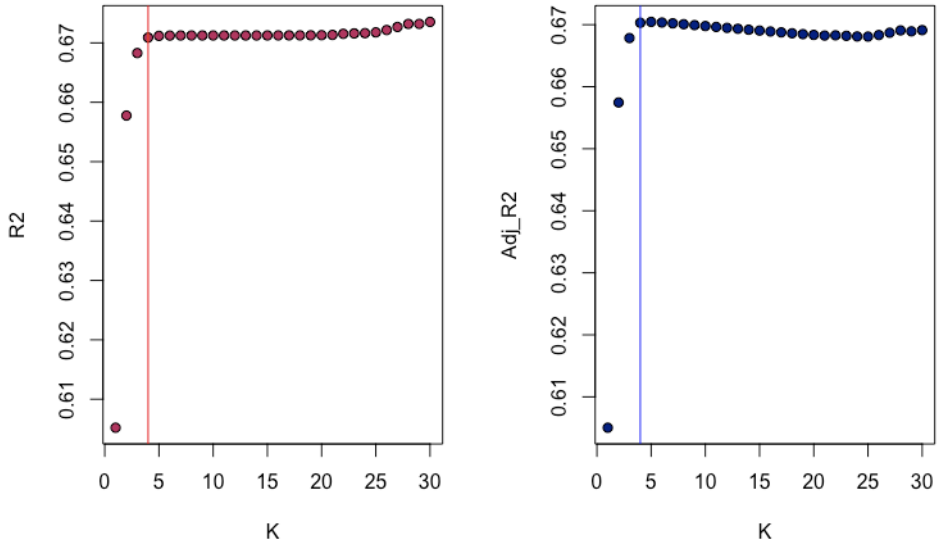


Figure 4.8: Correlations of PLS predicted values and responses

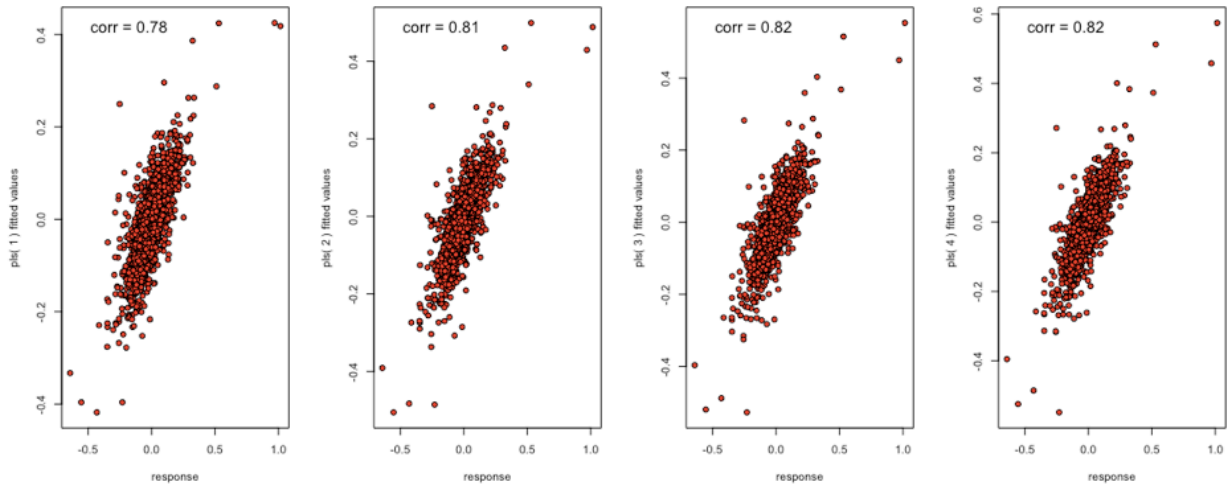
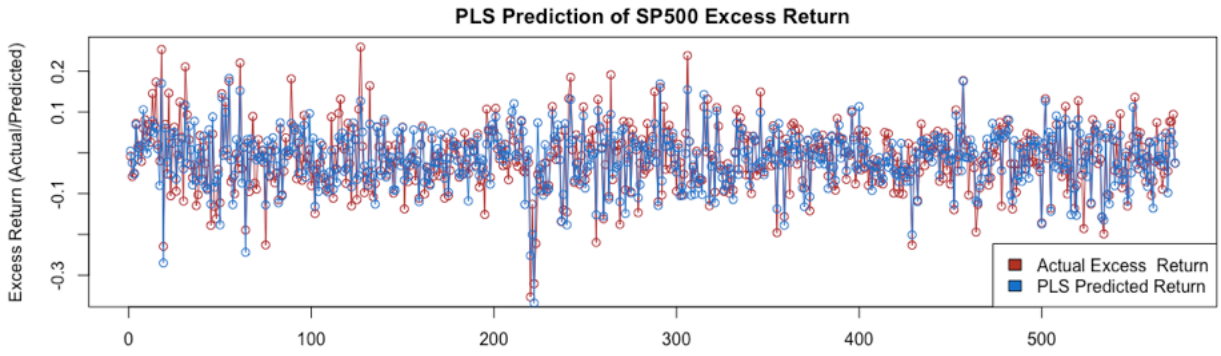


Figure 4.9: SP500 excess return prediction by PLS



4.4 CART Tree and Random Forest

I continued my model fitting and testing with CART Tree. The fitted tree is displayed in Figure 4.10. Surprisingly, CART tree exclusively used SMB as the cutoff variable at each branch level. The IS R2 is 0.471. Then, the predicted values were compared with the actual observed values in Figure 4.11. the OOR2 is 0.666.

Figure 4.10: CART Tree

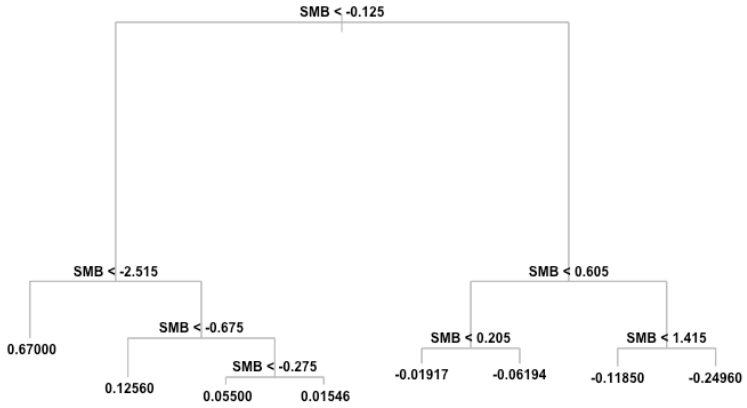
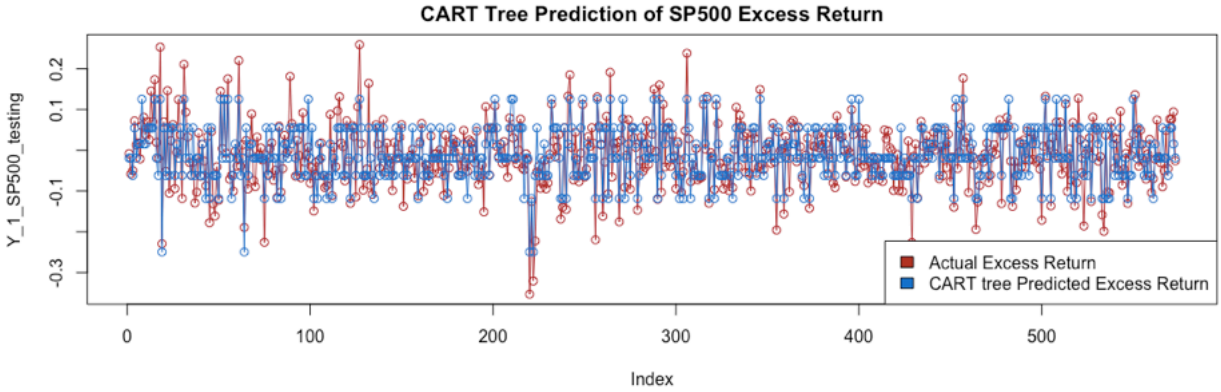


Figure 4.11: SP500 excess return prediction by CART Tree



Besides using tree model, random forest was applied as well. I plotted random forest's variable importance, as shown in Figure 4.12. The result is consistent with previous findings. For example, SMB, Mkt_RF always has a significant impact on the variance of excess returns. BING_Negative is another key variable. However, in this case, the importance of HML dwindled. This fitted random forest model has IS R2 0.722. I also used the fitted random forest model to predict excess returns in the *testing* dataset. The result is in Figure 4.13. and the OOS R2 is only 0.361.

Figure 4.12: Random forest variable importance

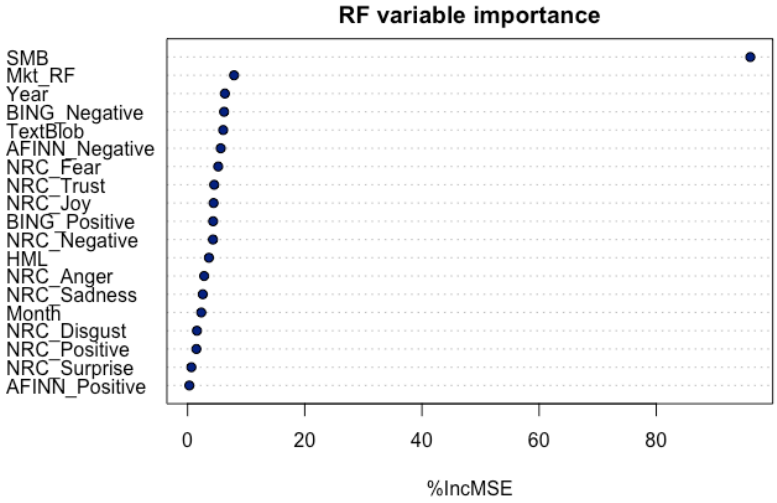
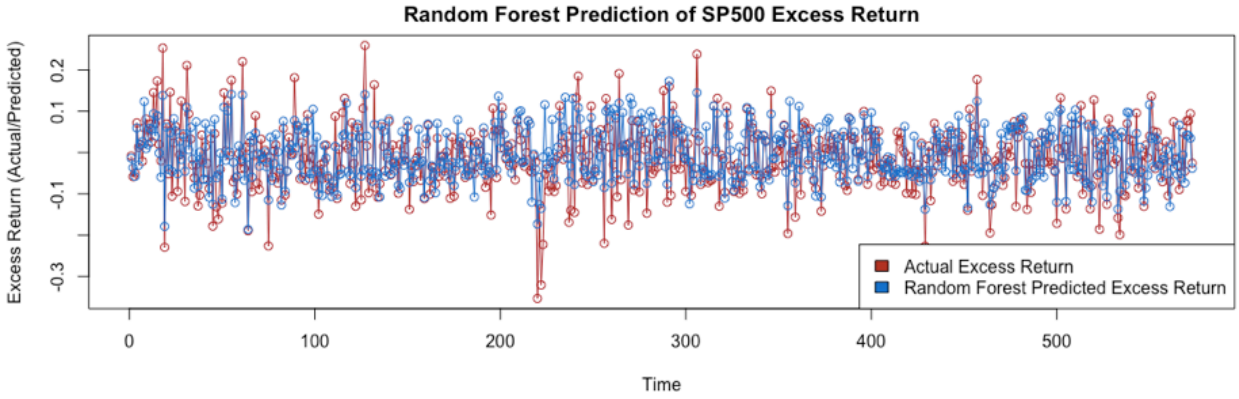


Figure 4.13: SP500 excess return prediction by Random Forest

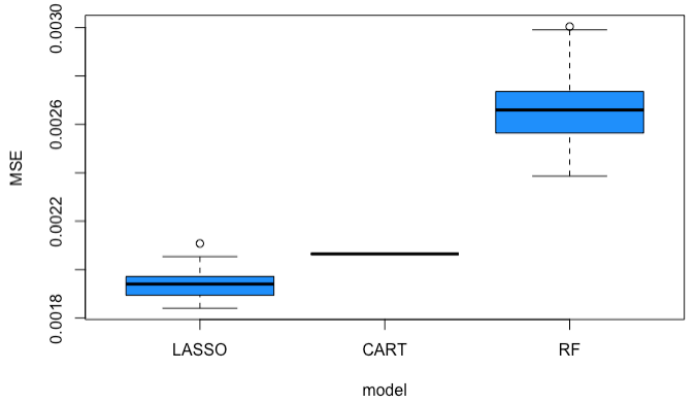


4.5 Average Modeling

My Approach II predicts excess returns by building up models over key factors. The models applied were linear regression, LASSO, PLS, Cart Tree and random forest. In addition to trees and forests, average modeling serves another way to improve the predicting performance. Hence, now

I check the MSE of LASSO, Cart Tree and random forests running these models 50 rounds. Specifically, instead of cutting t*esting* and dwddwFor each round, I calculated the OOS MSE. Afterwards, I collected 50 MSEs for each model and drew the box plot as shown in Figure 3.24. Obviously, LASSO outperformed CART Tree and random forest. LASSO model has smallest MSEs within a narrowed range. Since there is not much overlapping of MSEs in different models, LASSO is always the best choice. And average modeling is unnecessarily as satisfying as factors.

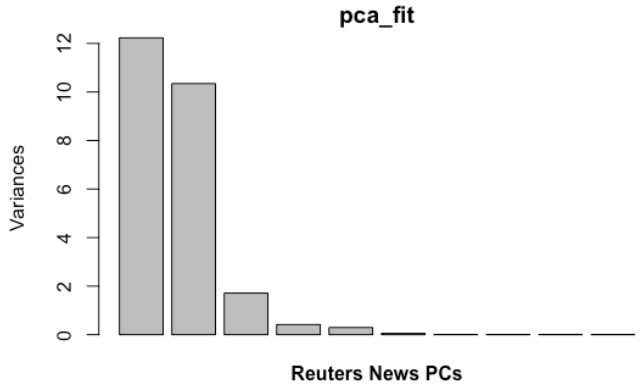
Figure 4.14: MSE of LASSO, CART, and Random Forest



4.6 PCA

After going through all such models as linear regression, LASSO, PLS, CART Tree, random forest and average modeling, I finally look at PCA. Figure 4.15 tells about how much variance of dataset captured by each PC. PC1, PC2, and PC3 have the most significance in summarizing the information.

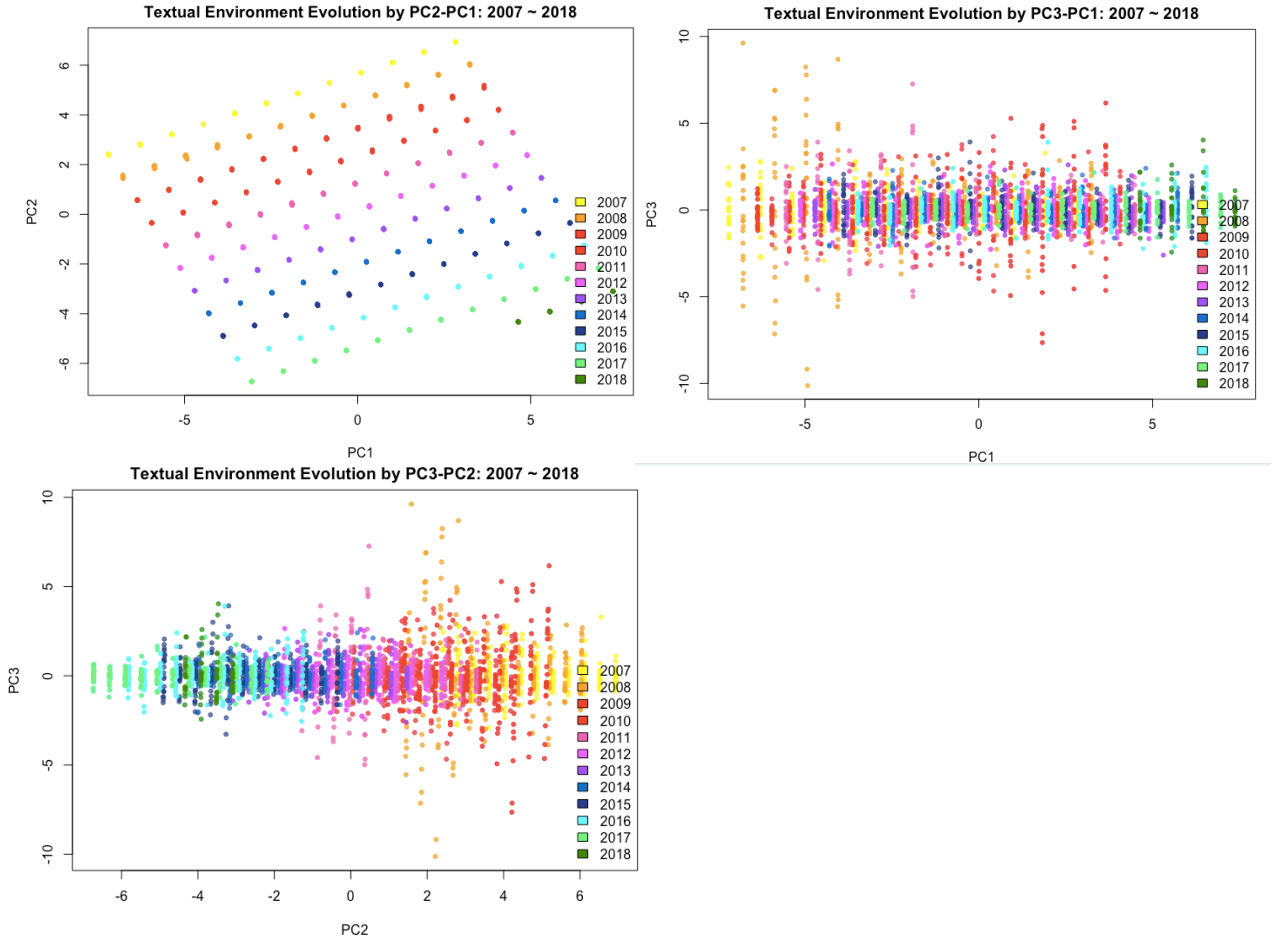
Figure 4.15: PC & variance



Next, I predicted the rotation of PCs for each variable and plotted them by years. For instance, Figure 4.16 displays the PC1-PC2 coordinate. The grid-liked scatter plots location is interesting.

It tells that, although PC2 is an important component, it deals nothing within each year. Instead, PC2 tends to reflect more information variance over different years. To further discover principal components' roles in explaining the information variance, I also plotted on PC1-PC3 and PC2-PC3 coordinates. PC1-PC3 plot tells that within the same year, PC3 values tend to be stable, but PC1 changes in a wide range. Across different years, PC1 varies and PC3 also varies. Additionally, in the PC2-PC3 plot, there is only PC2 changes across different years and PC3 changes within the same year. To summarize, PC1 captures the variance of information which changes both within one year and across different years. PC2 only captures information variances across different years, while PC3 only captures information variances within one year.

Figure 4.16: Textual environment evolution by PCs: 2007 ~ 2008



Next, I selected the top 10 PCs and ran PCR of linear regression, LASSO, CART Tree, and random forest. I hope to see whether the performance could be improved in the hybrid PCR with other models.

4.6.1 PCR-linear

Table 4.4 displays the PCR linear regression table. Interestingly, although PC1, 2, 3 capture most amount of information variance, they are not all statistically significant. Instead, PC3 imposes a significant effect on excess returns. There are two columns in Table 4.4, corresponding to a cut model and a full model. The IS R2 of cut and full models are 0.00332 and 0.671, respectively. And the OOS R2 are -95.137 and 0.563, respectively. Therefore, the full model not only has a great explanatory power but also fairly well potential in predicting excess returns. In contrast, the cut model performs even worse than the mean of historical data, because its OOS R2 is negative.

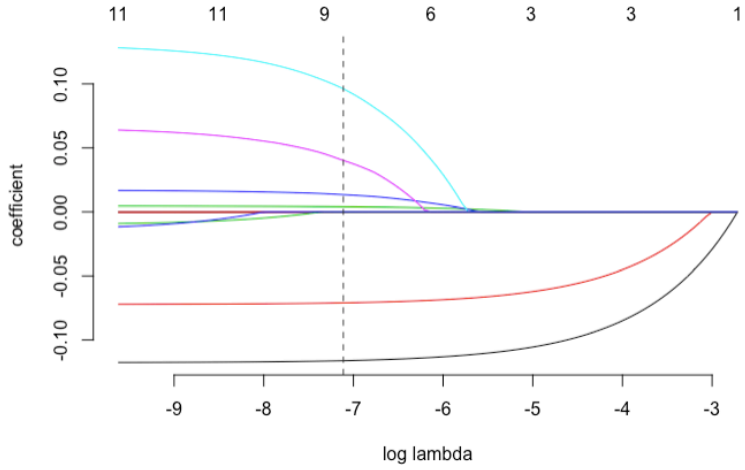
Table 4.4: Linear regression results of excess returns on PCs

	Dependent variable:	
	Y_1_SP500_training	
	Cut Model	Full Model
	(1)	(2)
PC1	-0.0001	-0.0002
	(0.001)	(0.0004)
PC2	-0.001	-0.0002
	(0.001)	(0.001)
PC3	0.004**	0.005***
	(0.002)	(0.001)
PC4		-0.072***
		(0.002)
PC5		-0.118***
		(0.002)
PC6		0.017***
		(0.006)
PC7		-0.010
		(0.023)
PC8		0.066*
		(0.038)
PC9		0.131***
		(0.048)
PC10		-0.015
		(0.049)
Constant	-0.009***	-0.010***
	(0.002)	(0.002)
Observations	2,250	2,250
Log Likelihood	1,975.236	3,221.066
Akaike Inf. Crit.	-3,942.471	-6,420.133
Note:	*p<0.1; **p<0.05; ***p<0.01	

4.6.2 PCR-LASSO

Running LASSO on the top 10 PCs filtered out 8 non-zero coefficients, as shown in Figure 4.17. Meanwhile, the IS R2 and OOS R2 are respectively 0.670 and 0.556.

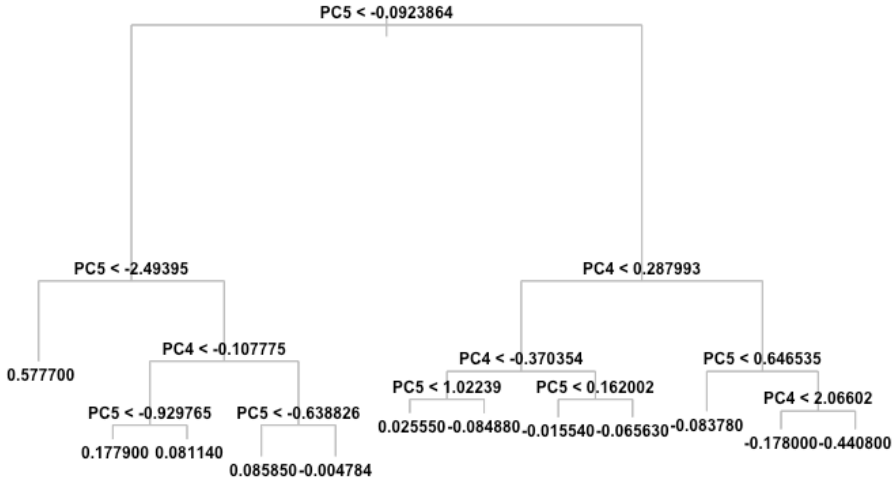
Table 4.17: PCR-LASSO



4.6.3 PCR-CART Tree

The CART Tree of the top 10 PCs is displayed in Figure 4.18. Both explanatory power and predicting power are less satisfying, being 0.338 and 0.229 respectively.

Figure 4.18: PCR-CART Tree

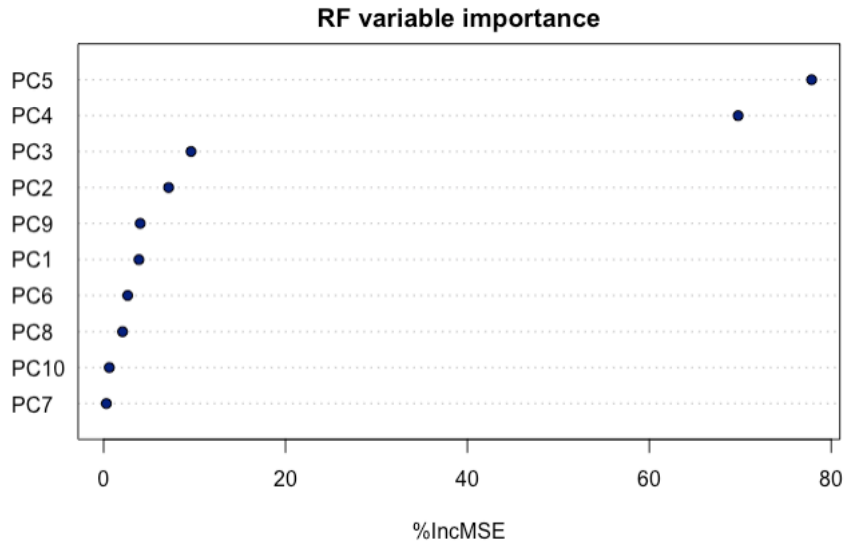


4.6.4 PCR-Random Forest

Random forest variable importance is plotted in Figure 4.19. PC5 and PC4 are two most important, although PC1, 2, 3 capture most amount of information variance. Moreover, the

importance of PC 5 and 4 is consistent with the tree structure in which PC 5 and 4 play the role as cutting points. Moreover, the IS R2 and OOS R2 are respectively 0.697 and 0.301. So, although PCR random forest has great explanatory power, its predicting power is still under-performed.

Figure 4.19: Random forest variable importance



5. Conclusion

This paper aims at finding a model having strong predicting power of excess returns. I used both *numbers* and *text* dataset, and resorted to multiple statistical models, such as linear regression, LASSO, PLS, CART Tree, Random Forest, PCA and a series of PCR. I find that the quality of the model greatly relies on the quality of the training dataset. For instance, in Section 3 Approach I, I directly used the word frequency sparse matrix to run the regressions. The results were poor and even worse than the mean value of training data. I think the reason of such poor performance is that huge dimensional sparse matrix carries information in an inefficient way. Although the information is abundant, each little piece of information is designed to be carried by a single word. In that sense, it is hard to fully synthesize the information and use it for training models. In contrast, in Section 4 Approach II, I used sentiment factors which are derived from the word frequency matrix. This synthesizing step greatly improves the "concentration" of information and hence the quality of the dataset. There are indeed a couple of satisfying models which give both high IS R2 and OOS R2.

Table 5.1 gives a summary of IS R2 and OOS R2 of all the models built in both Sections 3 and 4. IS R2 mainly reflects model's explanatory power, while OOS R2 indicates the model's predicting power. Given the goal of this paper is to predict excess returns, I will focus more on OOS R2 while evaluating the performance of each model.

Table 5.1: Summary of R2 of all models

	Section 3: Approach I			Section 4: Approach II	
	IS R2	OOS R2		IS R2	OOS R2
Linear Regression			FF3	0.667	
			Full	0.671	0.559
			Cut	0.669	0.564
LASSO	0.165	-6.23		0.670	0.581
PLS				0.509	0.566
CART Tree				0.471	0.666
Random Forest				0.722	0.361
PCR – Linear	0.000473	-102.679	Cut	0.00332	-95.137
			Full	0.671	0.563
PCR – LASSO	0.0314	-772.247		0.670	0.556
PCR – CART Tree				0.338	0.229
PCR – RF				0.697	0.301

Table 5.1 tells that the random forest model has the strongest explanatory power, which is followed by the PCR of random forest. Meanwhile, CART Tree model of key factors on excess returns yields the highest OOS R2 0.666, which is a relatively satisfying predicting power. It is followed by LASSO with 0.581 OOS R2.

As for the future work, I hope to import the daily most-up-to-date date, and pour it into the CART Tree model or LASSO model to predict the next-day excess returns. A good prediction allows me to make more sensible and wisdom trading decisions. Last but not least, although I have one subsection about average modeling, it is not fully complete. So, I hope to create a boxplot including all models MSE distributions, and seek for any opportunity to do average modeling.