S&P 500 Classification Models
SS 3850 Statistical Learning
Individual Research Project

Yizhou Tang
250888541
April 13th, 2020

TABLE OF CONTENTS

0.0. Abstract

The prediction of stock prices and financial indices have always been challenging tasks due to the randomness and non-linear dependency of financial data. These properties make traditional forecasting methodologies have low performance results. The idea of solving the forecasting problem using modern machine learning models seems plausible. The objective of this project is to examine multiple classification algorithms to predict the next day return of Standard & Poor 500 index based on historical data.

Keywords: forecasting, stock index, S&P 500, machine learning, classification

1.0 Introduction

Forecasting the financial market is one of the most sought questions in the world, but also one of the biggest challenges any analyst faces. The financial market is constantly evolving, the market participants are constantly learning from the past and adapting their behaviours. The market is also extremely complex due to many non-linear relationships and interactive effects. In order to deal with such a complex system, it is crucial for quantitative analysts to go beyond the traditional techniques and embrace complex machine learning models.

With the rise of artificial intelligence, deep learning and computing power, we have more tools to analyze financial time series data than ever. The idea is to combine and bridge the gap in between financial mathematics, computer science, software engineering, economics, psychology. In this project I will go over multiple classification algorithms in an attempt to forecast the market. Specifically, SPDR S&P 500 Trust ETF (SPY) is picked as a convenient way to analyze the S&P 500. Many other related datasets will also be used, such as FX, commodity futures, and economic data (more information will be described in the next section).

1.1 Dataset Description & Motivation

This project involves multiple datasets from different sources. I decided to collect as many data as possible in order to feed our models with sufficient information to predict the S&P 500. The datasets consist of three groups.

Group 1:
This group of time series data all share similar characteristics as they are updated every business day. Very low data preprocessing techniques are needed. I choose multiple commodity futures, major foreign exchange pairs, different treasury related data in order to include the daily macro effects around the world.

Source: Quandl.

1. Gold Futures, Continuous Contract #1 (GC1) (Front Month)
2. Eurodollar Futures, Continuous Contract #1 (ED1) (Front Month)
3. Silver Futures, Continuous Contract #1 (SI1) (Front Month)
4. Crude Oil Prices: West Texas Intermediate (WTI) - Cushing, Oklahoma
5. USDCAD
6. EURUSD
7. GBPUSD
8. USDJPY
9. AUDUSD
10. NZDUSD
11. USDCHF
12. USDNOK
13. USDCNY
14. USDINR
15. Trade Weighted U.S. Dollar Index: Major Currencies
16. Trade Weighted U.S. Dollar Index: Broad
17. Effective Federal Funds Rate
18. 3-Month Treasury Bill: Secondary Market Rate
19. 5-Year Treasury Constant Maturity Rate
20. 10-Year Treasury Constant Maturity Rate
21. 30-Year Treasury Constant Maturity Rate
22. 5-year Breakeven Inflation Rate
23. 10-year Breakeven Inflation Rate
24. 5-Year Forward Inflation Expectation Rate
25. TED Spread

26. Bank Prime Loan Rate

Group 2:

While the previous data represents the external macro relationships surrounding the equity market. This group of data represents the internal interactive effects within the U.S. equity market. The VIX index (also known as the fear index) is included to measure the “mood” of the market participants. ETFs that represent different industries are included with a goal to model different market regimes.

Source: Yahoo Finance.

1. Vix index
2. Energy Select Sector SPDR Fund
3. Financial Select Sector SPDR Fund
4. Utilities Select Sector SPDR Fund
5. Industrial Select Sector SPDR Fund
6. Technology Select Sector SPDR Fund
7. Health Care Select Sector SPDR Fund
8. Consumer Discretionary Select Sector SPDR Fund
9. Consumer Staples Select Sector SPDR Fund
10. Materials Select Sector SPDR Fund

Group 3:

The previous two groups allow the models to have an idea of what is happening on a daily basis due to the high frequency of the data. However, in order to measure the financial market, it is also important to look at the lower frequency economic data. This group includes lower frequency data that are released on a monthly/quarterly basis, hence data preprocessing techniques such as forward fill will be needed. The economic indices are chosen in order to cover the different areas such as growth, inflation, employment, income and expenditure, and debt.

Source: Quandl.

1. Gross Domestic Product
2. Real Gross Domestic Product
3. Real Potential Gross Domestic Product
4. Consumer Price Index for All Urban Consumers: All Items
5. Consumer Price Index for All Urban Consumers: All Items Less Food & Energy
6. Gross Domestic Product: Implicit Price Deflator
7. St. Louis Adjusted Monetary Base
8. M1 Money Stock
9. M2 Money Stock

10. Velocity of M1 Money Stock
11. Velocity of M2 Money Stock
12. Civilian Unemployment Rate
13. Natural Rate of Unemployment (Long-Term)
14. Natural Rate of Unemployment (Short-Term)
15. Civilian Labor Force Participation Rate
16. Civilian Employment-Population Ratio
17. Unemployed level
18. All Employees: Total nonfarm
19. All Employees: Manufacturing
20. Initial Claims
21. Real Median Household Income in the United States
22. Real Disposable Personal Income
23. Personal Consumption Expenditures
24. Personal Consumption Expenditures: Durable Goods
25. Personal Saving Rate
26. Real Retail and Food Services Sales
27. Disposable personal income
28. Federal Debt: Total Public Debt
29. Federal Debt: Total Public Debt as Percent of Gross Domestic Product
30. Excess Reserves of Depository Institutions
31. Commercial and Industrial Loans, All Commercial Banks
32. Industrial Production Index
33. Capacity Utilization: Total Industry
34. Housing Starts: Total: New Privately Owned Housing Units Started
35. Gross Private Domestic Investment
36. Corporate Profits After Tax (without IVA and CCAdj)
37. St. Louis Fed Financial Stress Index
38. Leading Index for the United States

2..0 Methodology

2.1 Date Preprocessing & Data Engineering

Due to the complex structures of data, multiple data preprocessing and engineering techniques were used, which will be covered in this section.

For the first two groups of data, due to the high frequency nature of the time series, additional features are derived from the original data. The motivation is that, instead of price data alone, it would be beneficial to include features such as percentage returns of prices, rolling average of returns, rolling standard deviation of returns, etc.

The additional derived metrics are:

1. Percentage return
2. Difference between price at time t and price at time t-1
3. 21 day rolling mean of returns
4. 21 day rolling standard deviation of returns
5. 21 day rolling skewness of returns
6. 21 day rolling kurtosis of returns
7. 63 day rolling mean of returns
8. 63 day rolling standard deviation of returns
9. 63 day rolling skewness of returns
10. 63 day rolling kurtosis of returns
11. 126 day rolling mean of returns
12. 126 day rolling standard deviation of returns
13. 126 day rolling skewness of returns
14. 126 day rolling kurtosis of returns
15. 252 day rolling mean of returns
16. 252 day rolling standard deviation of returns
17. 252 day rolling skewness of returns
18. 252 day rolling kurtosis of returns
19. 10 day exponential moving average of prices
20. 20 day exponential moving average of prices
21. 60 day exponential moving average of prices
22. 252 day exponential moving average of prices
23. Ratio between current price and 10 day exponential moving average
24. Ratio between current price and 20 day exponential moving average
25. Ratio between current price and 60 day exponential moving average
26. Ratio between current price and 252 day exponential moving average
27. MACD

The list above is applied to every single feature in our first two groups. The reason for not applying these calculations on the third group is due to the low frequency. However, percentage return is applied to every single feature in the third group. In order to deal with the sparse problem of the data, forward fill is used on the features of the third group as well.

With the data preprocessing methods used above, the number of features increased by multiple times. By computing these additional derived features, more meaningful information might be derived for the model, and hence make better predictions.

However, due to the large number of features, problems such as curve fitting arise, hence feature selection and dimension reduction techniques are implemented in the next section.

One additional step is the train/test split. The train set consists of all the data starting from 2000-01-01 to 2016-01-01. The test set consists of all the data from 2016-01-01 to 2020-02-20.

2.2 Feature Selection

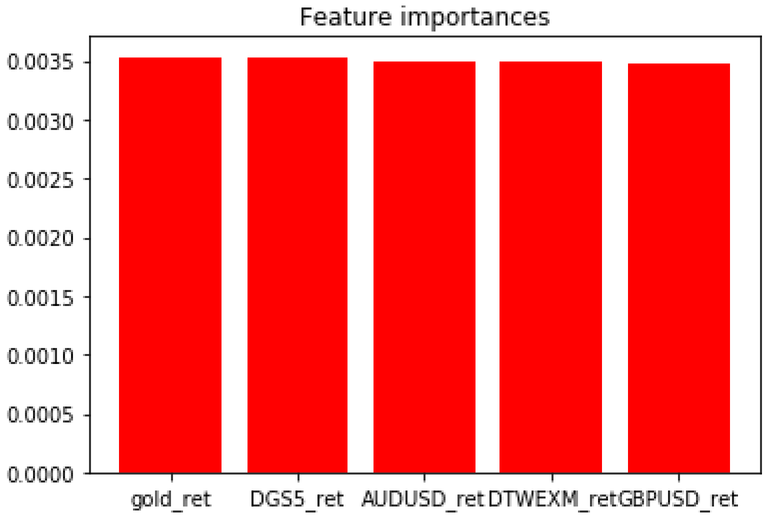
In the previous section, we addressed some of the issues with the data structures and attempted to derive meaningful features. As a result, we are now dealing with a much more comprehensive set of features, and in this section and the next, the goal is to get rid of the noises and only keep the ones that are actually meaningful to our model.

First step before conducting any feature selection algorithms is to standardize. I chose to use the standard scaler method, which standardizes each feature by its mean and standard deviation.

After standardizing our data, a model-based feature selection method is used by using scikit-learn package's SelectFromModel method. In this research I decided to use the Extra Trees Classifier as the base estimator from which the transformer is built. The reason for picking the Extra Trees Classifier instead of simple models such as Logistic Regression is because I want to be able to include non-linear relationships into my models, if a linear model is used for the feature selection process, potential complex relationships might be penalized and hence removed.

With 583 features, after using the SelectFromModel method, number of features reduced to 309. Since our base model is the Extra Trees Classifier, we can also call the feature importance attribute and analyze the top features. The top 5 features concluded by our feature selection algorithm are listed below.

Feature ranking and their importances:
1. feature gold_ret (0.003529)
2. feature DGS5_ret (0.003527)
3. feature AUDUSD_ret (0.003489)
4. feature DTWEXM_ret (0.003484)
5. feature GBPUSD_ret (0.003472)



As we can see, gold_ret, which is the continuous gold future's current day return, is concluded to be the most important feature in our data set. This is not surprising as gold is one of the most traded assets in the world and it is known as a "safe haven asset" by the practitioners.

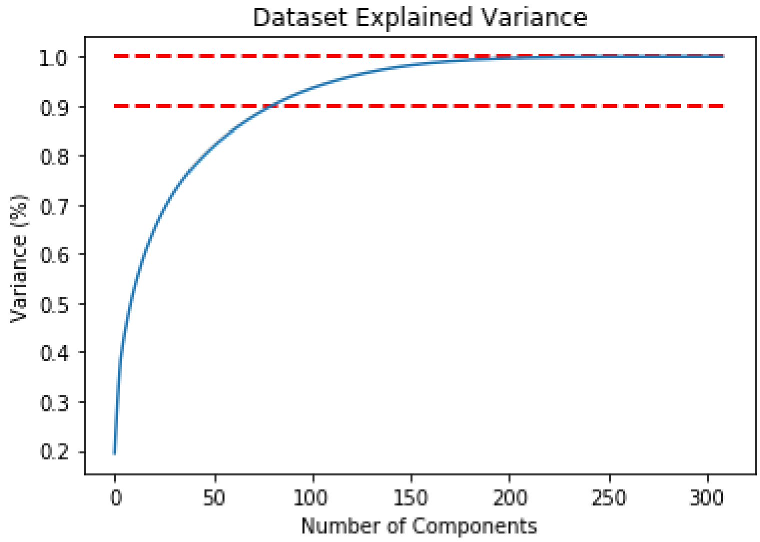
The second most important feature is DGS5 return, DGS5 is the notation used for 5-Year Treasury Constant Maturity Rate, hence the daily return of the 5-year treasury rate has a strong implication toward S&P 500's next day return.

The third, fourth, and fifth most important features in our train set are all FX related features. AUDUSD_ret representing the daily return of the AUDUSD exchange rate; DTWEXM_ret representing the daily return of Trade Weighted U.S. Dollar Index: Major Currencies; GBPUSD_ret representing the daily return of the GBPUSD exchange rate. It is not surprising that so many exchange rates are ranked highly. Exchange rates are directly related to the macro developments around the world and particularly the international trades with the US.

2.3 Principal Component Analysis (PCA)

In section 2.2, the number of features were reduced from 583 to 309, using a tree-based feature selection method. In this section, I will attempt to use a popular dimension reduction method, Principal Component Analysis (PCA) to further reduce the number of features. PCA allows us to use a smaller set of features to represent our total set of features. While the algorithm is very effective at dimension reduction, it is also important to decide the number of components in order to preserve the most amount of information.

The results of PCA is attached below:



With a simple search algorithm, it is found that 171 components can explain more than 99% of the total variance.

2.4 Classification Models

Now that the features went through standardization, feature selection, and dimension reduction. With the 171 components, it is time to train the models.

Below is the list of the classification models that will be tested:

1. Logistic Regression
2. Linear Discriminant Analysis
3. Quadratic Discriminant Analysis
4. Decision Tree
5. Extra Trees Classifier
6. Random Forest
7. Ridge Classifier
8. K Nearest Neighbors Classifier
9. Support Vector Classifier

The reason for picking classifiers above is to cover as many different classification methodologies as possible. Logistic regression and ridge classifier fall under linear models; LDA and QDA fall under discriminant analysis models; Random forest classifier and extra trees classifier fall under ensemble models; K Nearest Neighbors Classifier representing the nearest neighbors algorithms; Support vector classifier representing the support vector machine family of algorithms.

3.0 Results

We can analyze the performance of each model based on metrics such as accuracy, precision, recall, and F1 score.

Accuracy:

	Train_Accuracy	Test_Accuracy
Logistic Regression	0.590689	0.533531
LDA	0.590435	0.5286
QDA	0.750445	0.504931
Decision Tree	1	0.482249
ETC	1	0.482249
Random Forest	0.987535	0.499014
Ridge	0.590689	0.529586
KNN	0.690918	0.513807
SVC	0.838972	0.527613

We can see that the logistic regression performed the best in out of sample testing in terms of accuracy, with a score of 0.533531.

	Train_Precision	Test_Precision
Logistic Regression	0.599582	0.556818
LDA	0.599165	0.552743
QDA	0.772997	0.547486
Decision Tree	1	0.524911

ETC	1	0.534314
Random Forest	0.994655	0.56447
Ridge	0.599332	0.553371
KNN	0.701389	0.543988
SVC	0.814286	0.549072

We can see that the random forest performed the best in out of sample testing in terms of precision, with a score of 0.56447.

	Train_Recall	Test_Recall
Logistic Regression	0.68729	0.708861
LDA	0.688249	0.710669
QDA	0.74964	0.531646
Decision Tree	1	0.533454
ETC	1	0.394213
Random Forest	0.981775	0.356239
Ridge	0.688729	0.712477
KNN	0.726619	0.670886
SVC	0.902158	0.748644

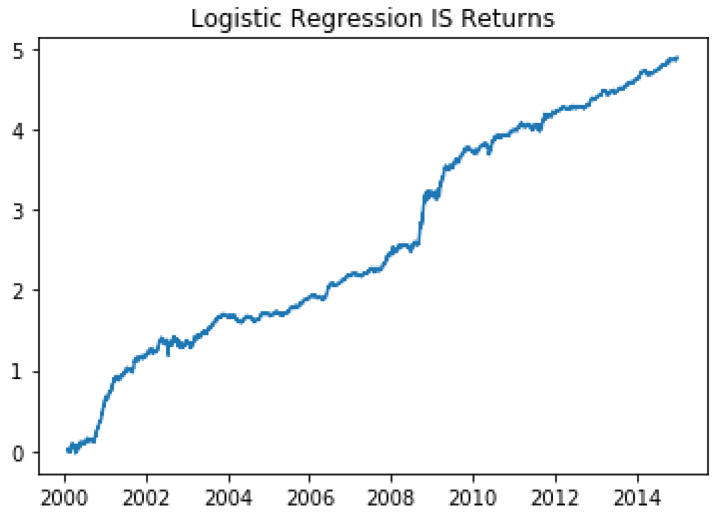
We can see that the support vector classifier performed the best in out of sample testing in terms of recall, with a score of 0.748644.

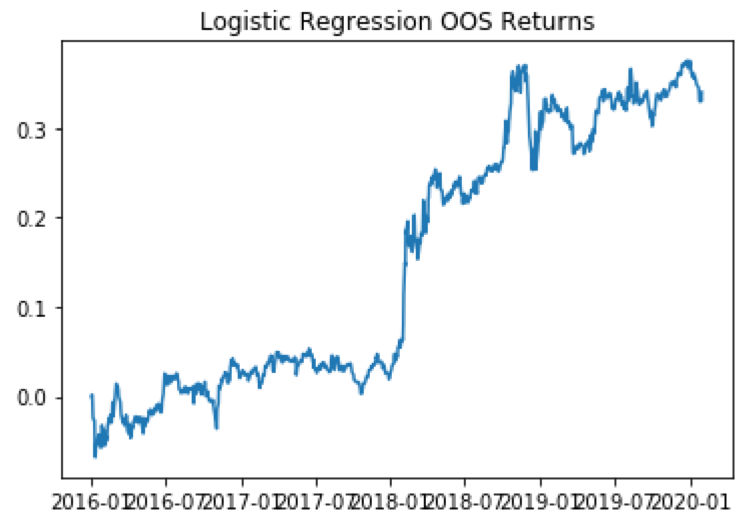
	Train_F1_Score	Test_F1_Score
Logistic Regressi	0.586209	0.51587

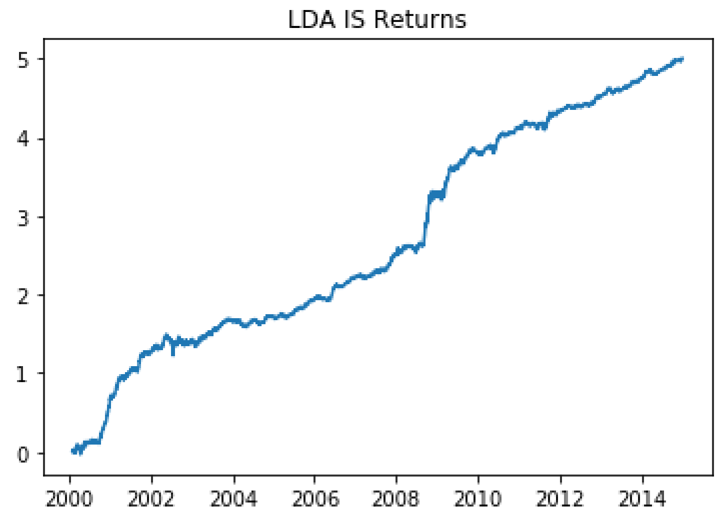
on		
LDA	0.585835	0.509318
QDA	0.750625	0.50552
Decision Tree	1	0.471583
ETC	1	0.486147
Random Forest	0.987795	0.459485
Ridge	0.586068	0.510135
KNN	0.690444	0.499654
SVC	0.837869	0.498091

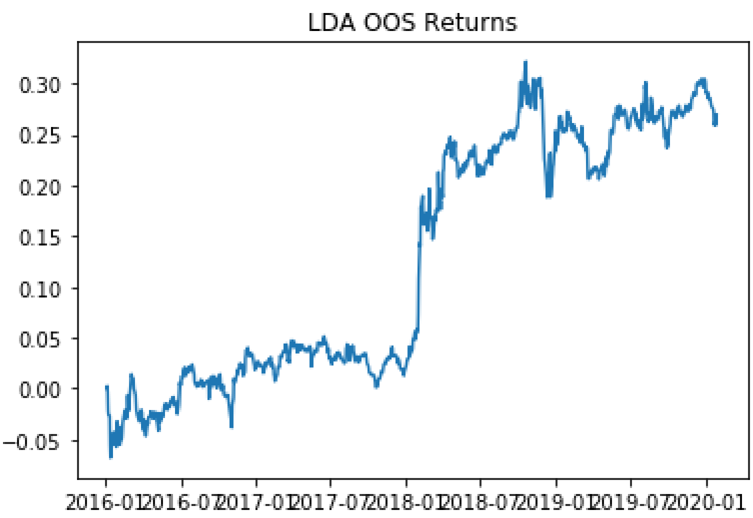
We can see that the logistic regression performed the best in out of sample testing in terms of F1 score, with a score of 0.51587.

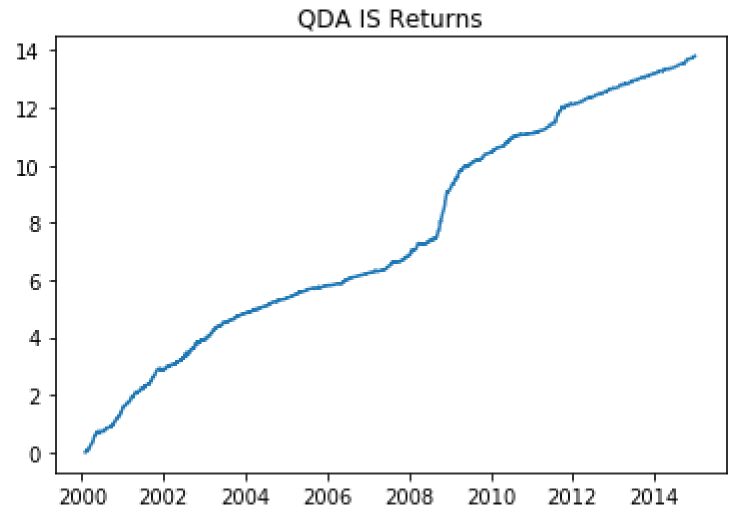
I also performed a backtest of each strategy in the train set and test set. The results are attached below:

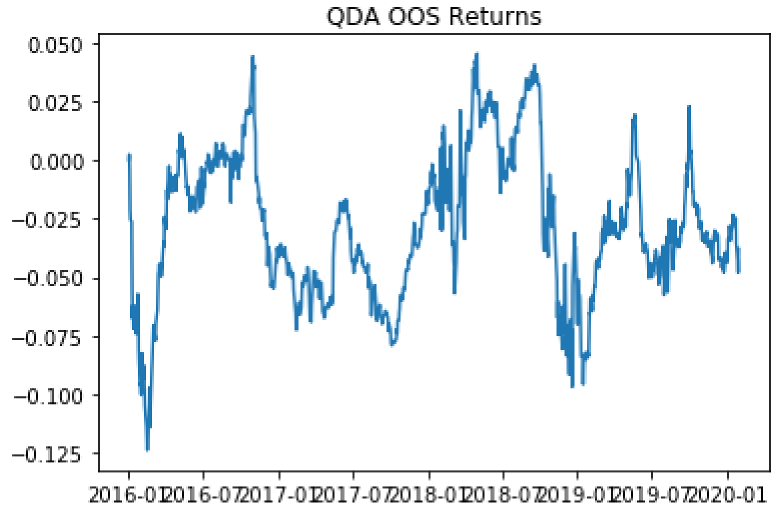


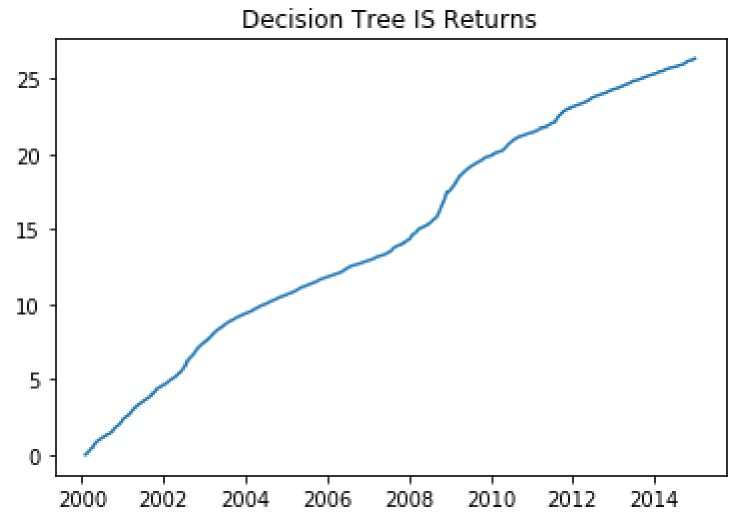

Logistic Regression OOS Returns
0.3
0.2
0.1
0.0
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01

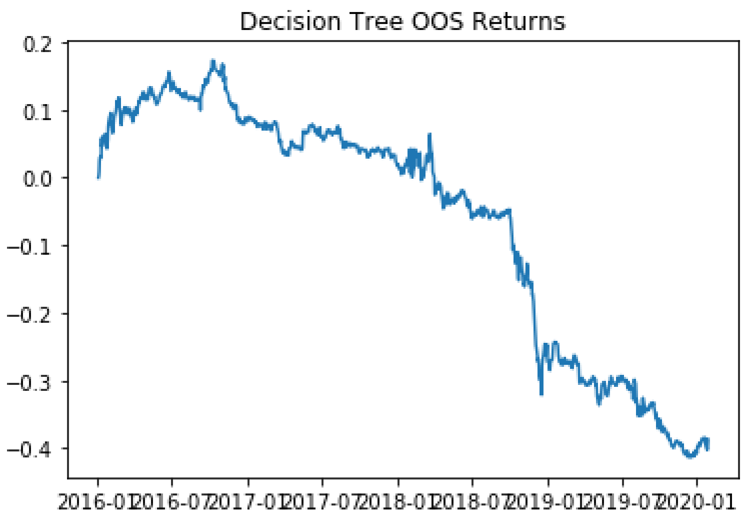

LDA IS Returns
5
4
3
2
1
0
2000
2002
2004
2006
2008
2010
2012
2014

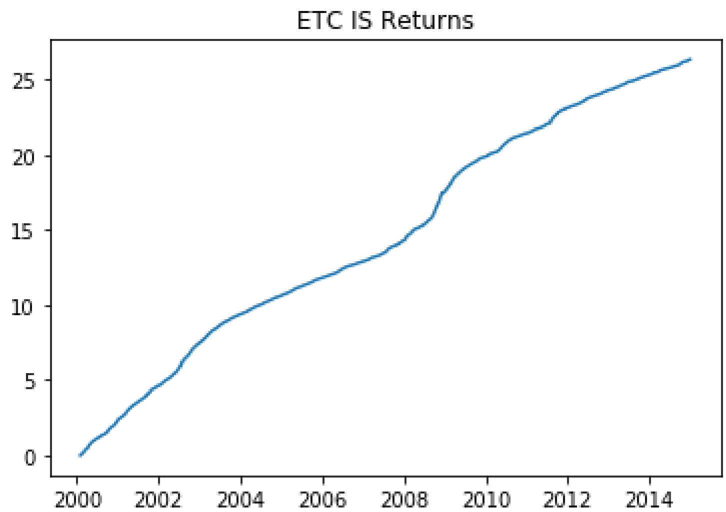

LDA OOS Returns
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01

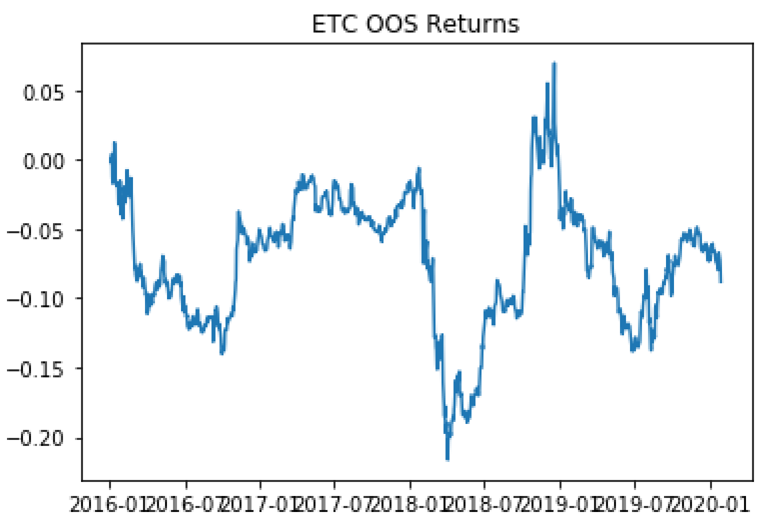

QDA IS Returns
14
12
10
8
6
4
2
0
2000
2002
2004
2006
2008
2010
2012
2014

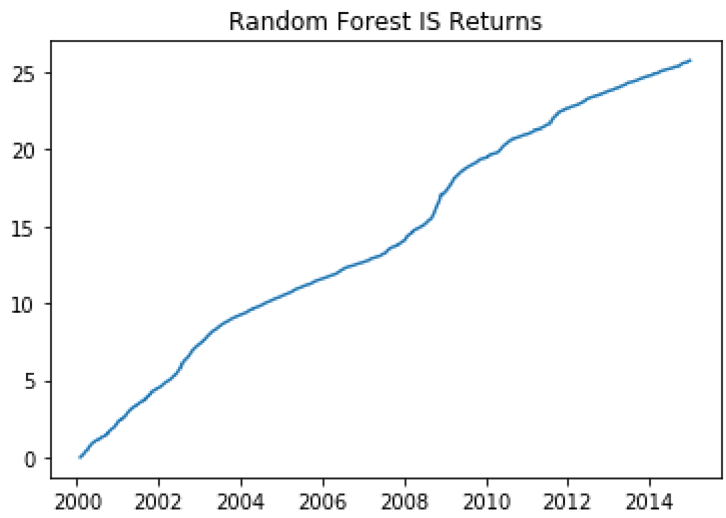

QDA OOS Returns
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01

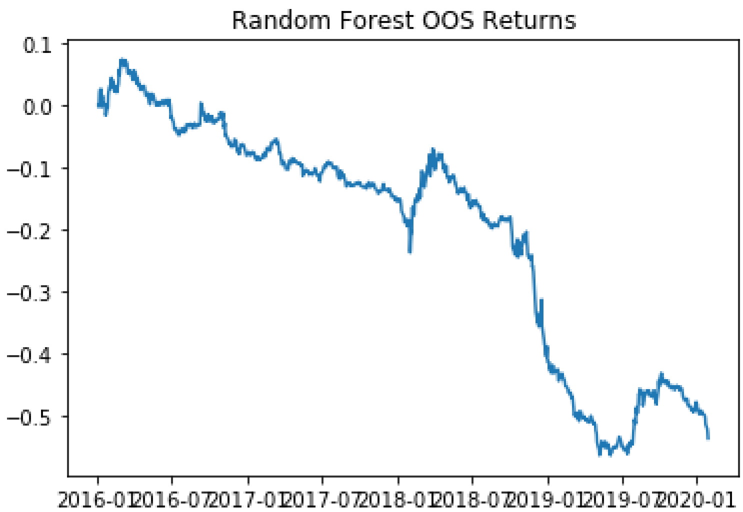

Decision Tree IS Returns
25
20
15
10
5
0
2000
2002
2004
2006
2008
2010
2012
2014

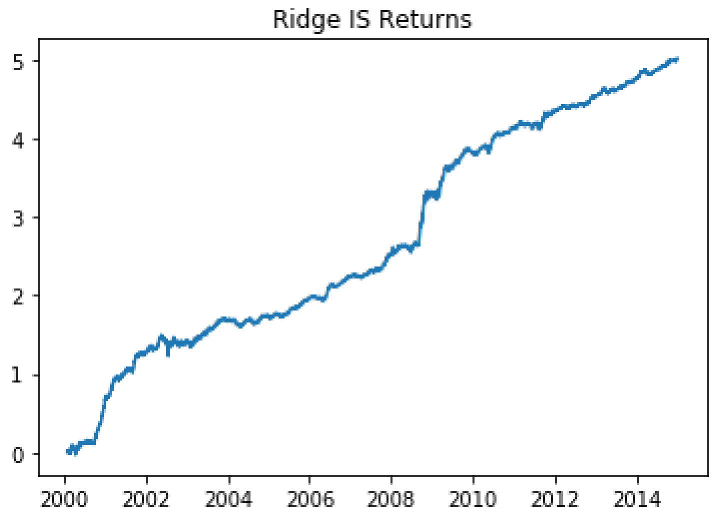

Decision Tree OOS Returns
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01

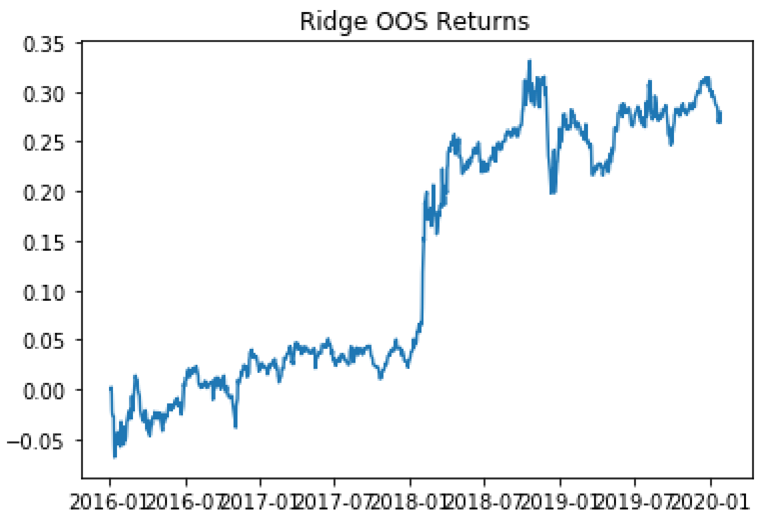

ETC IS Returns
25
20
15
10
5
0
2000
2002
2004
2006
2008
2010
2012
2014

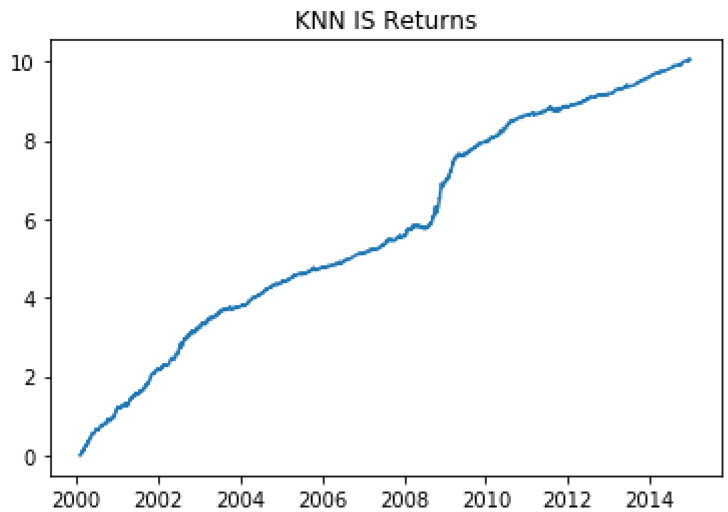

ETC OOS Returns
0.05
0.00
−0.05
−0.10
−0.15
−0.20
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01

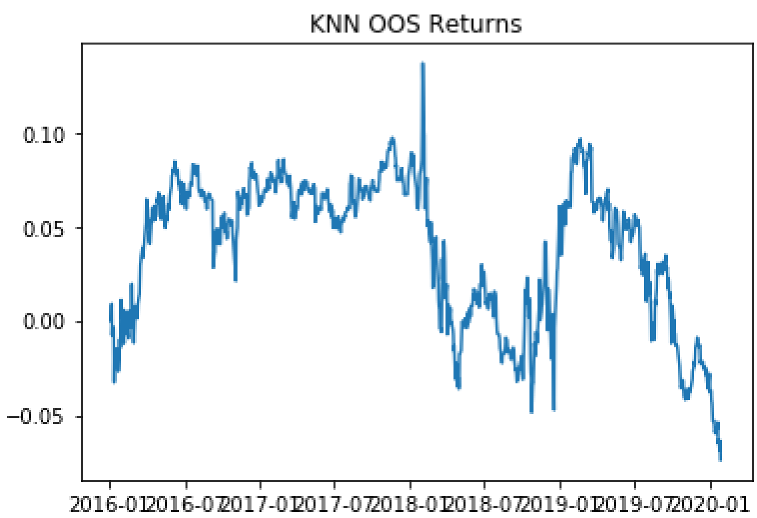

Random Forest IS Returns
25
20
15
10
5
0
2000
2002
2004
2006
2008
2010
2012
2014

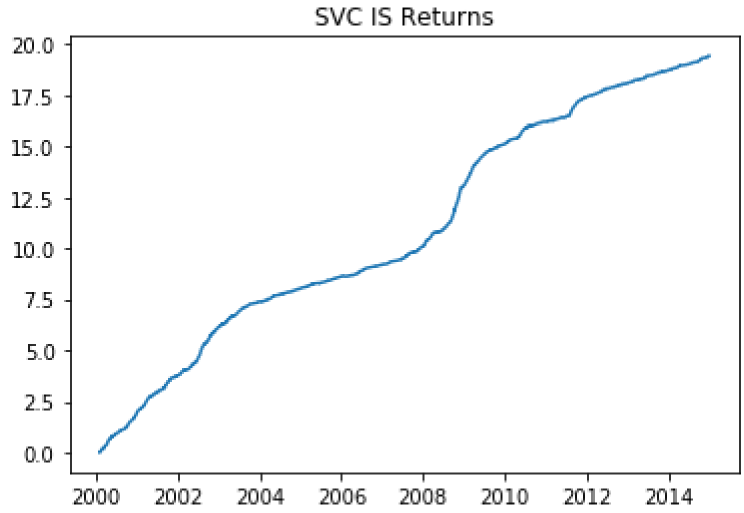

Random Forest OOS Returns
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01

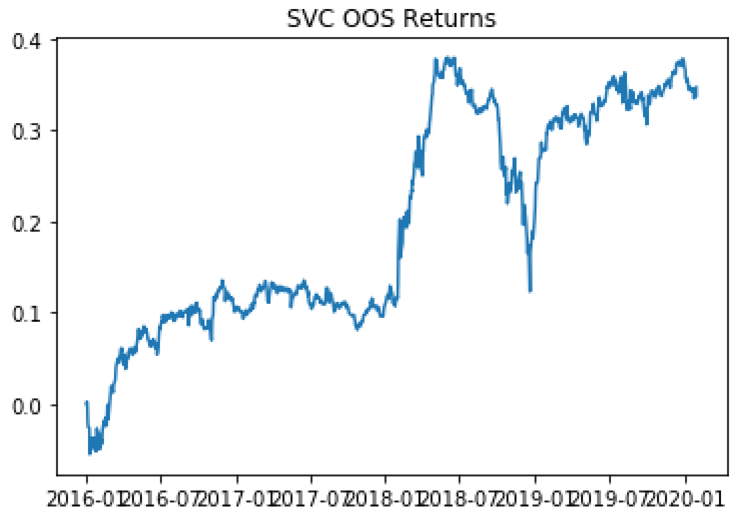

Ridge IS Returns
5
4
3
2
1
0
2000
2002
2004
2006
2008
2010
2012
2014


Ridge OOS Returns
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01


KNN IS Returns
10
8
6
4
2
0
2000
2002
2004
2006
2008
2010
2012
2014


KNN OOS Returns
0.10
0.05
0.00
−0.05
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01


SVC IS Returns
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2000
2002
2004
2006
2008
2010
2012
2014


SVC OOS Returns
0.4
0.3
0.2
0.1
0.0
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01

4.0 Discussion on Performance and Future Improvements

Interestingly, we can notice that complex models tend to perform not so well in out of sample, while simple models such as logistic regression were able to generate positive returns.

Python packages I learned:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import math
import sys
import seaborn as sns

import quandl

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.feature_selection import SelectFromModel

from sklearn import metrics


# Classifiers
from sklearn.linear_model import LogisticRegression,RidgeClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis,
QuadraticDiscriminantAnalysis
from sklearn.ensemble import RandomForestClassifier,ExtraTreesClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier,ExtraTreeClassifier
```