Machine Learning Engineer Nanodegree

Capstone Project

Chunyan Lei

May 26th, 2020

I. Definition

Project Overview

Quant trading is currently widely used in the finance industry. Among the quantitative strategies in the secondary market, the multi-factor strategy can be described as the earliest created but also one of the most changing investment strategies. A multi-factor model is a financial model that employs multiple factors, macroeconomic as well as fundamental and statistical, in its calculations to explain market phenomena and/or equilibrium asset prices. It can be used to analyze the returns of individual securities but also of entire portfolios. Therefore, it is a good model for stock selection and risk prediction, which means its importance in trading and investment.

The traditional multi-factor linear regression model is not stable enough sometimes. However, machine learning has a good performance in prediction and classification. The traditional multi-factor linear regression model also proves that multiple factors are indeed related to stock prices. This relationship is naturally suitable for applying machine learning to stock selection and timing, which makes the possibility of obtaining excess returns higher.

For the factor model, there are many researches of it. Below are some links of the researching papers on it. They mostly only implemented linear regression model. We will try some other machine learning models.

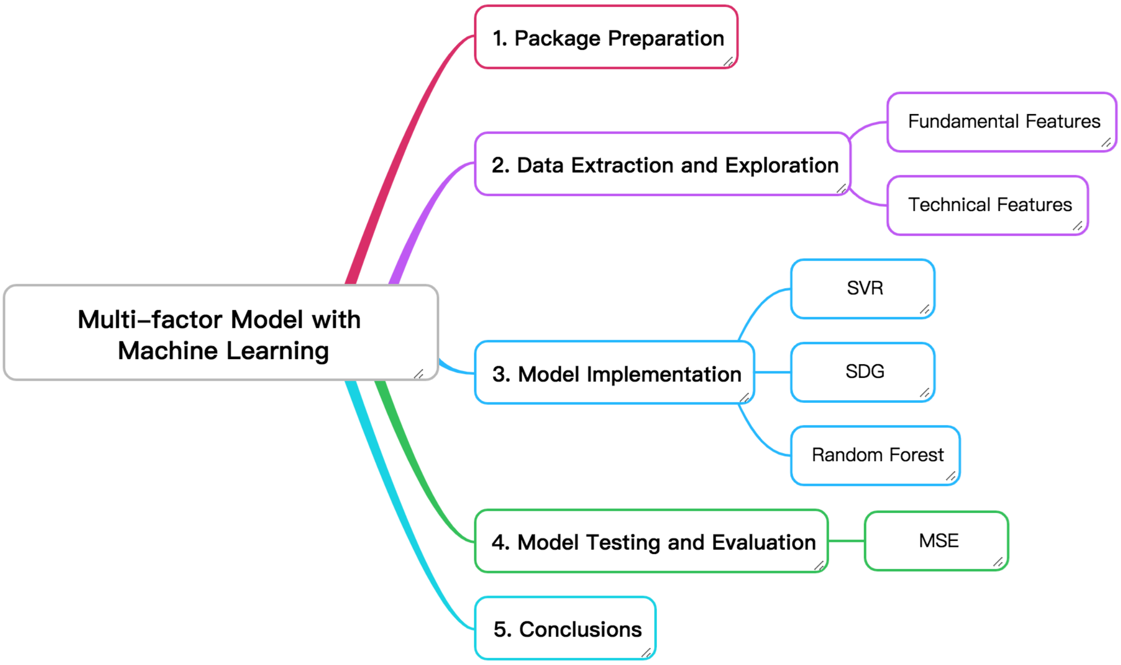
https://www.tandfonline.com/doi/abs/10.1080/13518470110071137

http://cdmd.cnki.com.cn/Article/CDMD-10056-1018059319.htm

https://onlinelibrary.wiley.com/doi/full/10.1111/j.1540-6261.1996.tb05202.x

Problem Statement

This is a regression problem. In this project, we aim to explain the stock return with factors using machine learning algorithms according to the following steps. Mining effective factors are meaningful in quantitive trading, effective factors should explain the most part of the trend of the stock return.



Metrics

This project is mainly about a statistical regression problem, we aim to explain the stock return as much as possible by multiple factors, thus, mean squared error (MSE) would be a good choice of metric, which measures the average of the squares of the errors, that is, the average squared difference between the estimated values and the actual value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

where Y is the vector of observed values of the variable being predicted, with $\hat{Y}$ being the predicted values.

We want to achieve a least MSE which means that the predicted value of the model is close enough to the actual value.

II. Analysis

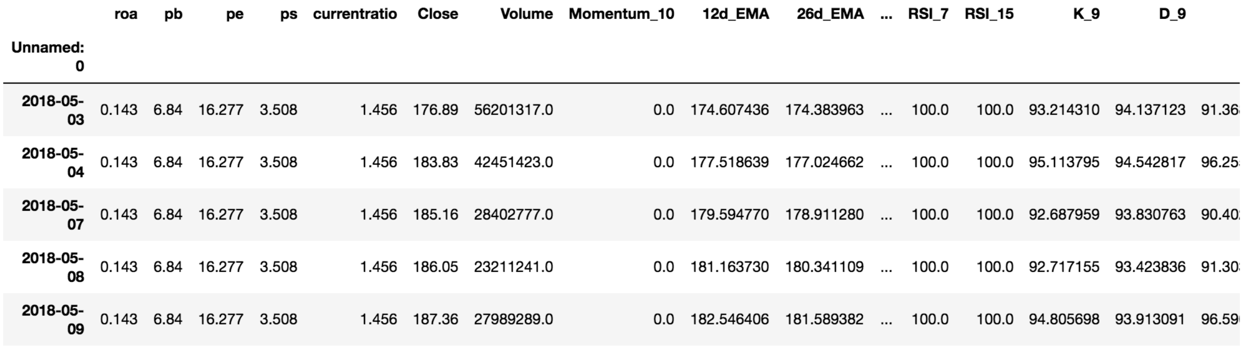
Data Exploration

The dataset for this project was obtained by Quandl Package of Python. We collect the daily data about each stock that Dow Jones index measures from May 1^{st} 2018 to May 1^{st} 2020, including P/E, P/B, P/S, RoA current ratio and the closing price, then we will calculate some technical factors including Momentum, MACD, RSI, KDJ, OBV and VR. The reason why I choose Dow Jones index stocks is that it only has 30 stocks, which will be convenient for me to do some researches and testing, due to the limited memory of my laptop. Certainly we can choose other stock pools and it is what we should do in practice.

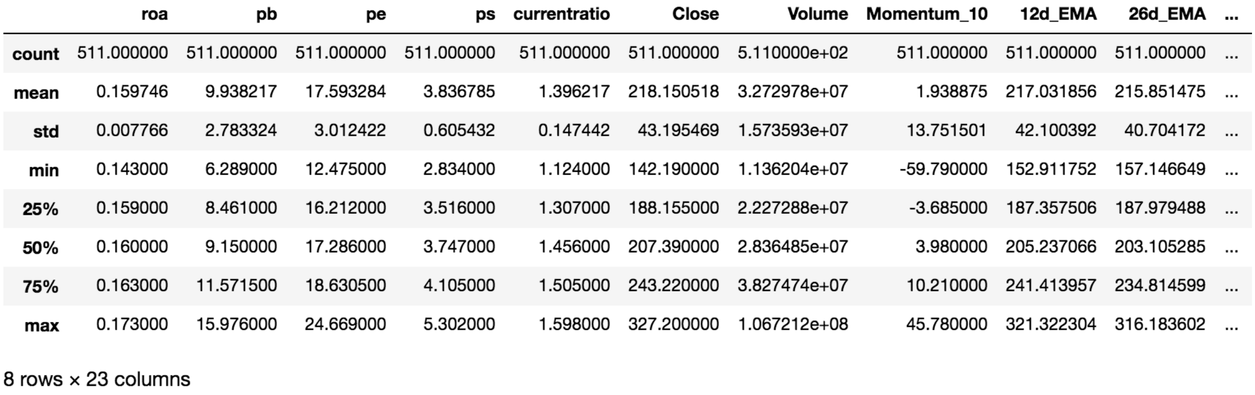
In order to use Quandl, first we need register a apikey on https://www.quandl.com/

To calculate fundamental features, we can access to ***'SHARADAR/SF1'*** dataset while calculate technical features, we need to use ***'EOD'*** dataset of Quandl. Fundamental features can be directly obtained by Quandl, while technical features need to be calculated with variables such as the closing price and volume.

Take Apple stock ('AAPL') as an example, after data preparation we get the following table.



Unnamed: 0	roa	pb	pe	ps	currentratio	Close	Volume	Momentum_10	12d_EMA	26d_EMA	...	RSI_7	RSI_15	K_9	D_9	
2018-05-03	0.143	6.84	16.277	3.508	1.456	176.89	56201317.0	0.0	174.607436	174.383963	...	100.0	100.0	93.214310	94.137123	91.36
2018-05-04	0.143	6.84	16.277	3.508	1.456	183.83	42451423.0	0.0	177.518639	177.024662	...	100.0	100.0	95.113795	94.542817	96.25
2018-05-07	0.143	6.84	16.277	3.508	1.456	185.16	28402777.0	0.0	179.594770	178.911280	...	100.0	100.0	92.687959	93.830763	90.40
2018-05-08	0.143	6.84	16.277	3.508	1.456	186.05	23211241.0	0.0	181.163730	180.341109	...	100.0	100.0	92.717155	93.423836	91.30
2018-05-09	0.143	6.84	16.277	3.508	1.456	187.36	27989289.0	0.0	182.546406	181.589382	...	100.0	100.0	94.805698	93.913091	96.59



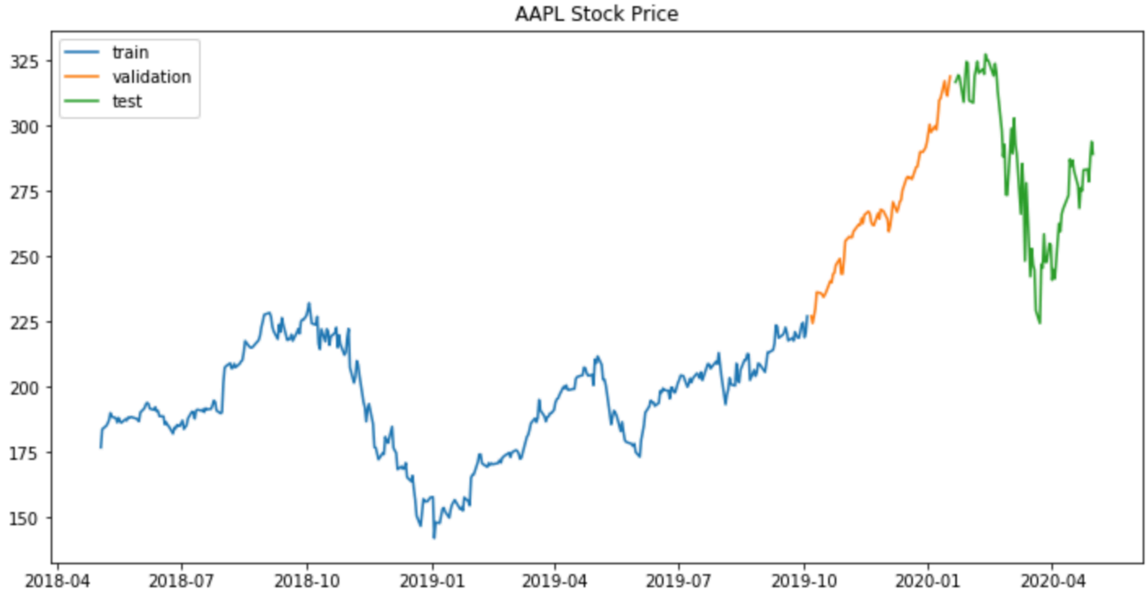
	roa	pb	pe	ps	currentratio	Close	Volume	Momentum_10	12d_EMA	26d_EMA	...
count	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	5.110000e+02	511.000000	511.000000	511.000000	...
mean	0.159746	9.938217	17.593284	3.836785	1.396217	218.150518	3.272978e+07	1.938875	217.031856	215.851475	...
std	0.007766	2.783324	3.012422	0.605432	0.147442	43.195469	1.573593e+07	13.751501	42.100392	40.704172	...
min	0.143000	6.289000	12.475000	2.834000	1.124000	142.190000	1.136204e+07	-59.790000	152.911752	157.146649	...
25%	0.159000	8.461000	16.212000	3.516000	1.307000	188.155000	2.227288e+07	-3.685000	187.357506	187.979488	...
50%	0.160000	9.150000	17.286000	3.747000	1.456000	207.390000	2.836485e+07	3.980000	205.237066	203.105285	...
75%	0.163000	11.571500	18.630500	4.105000	1.505000	243.220000	3.827474e+07	10.210000	241.413957	234.814599	...
max	0.173000	15.976000	24.669000	5.302000	1.598000	327.200000	1.067212e+08	45.780000	321.322304	316.183602	...

8 rows × 23 columns

Data Visualization

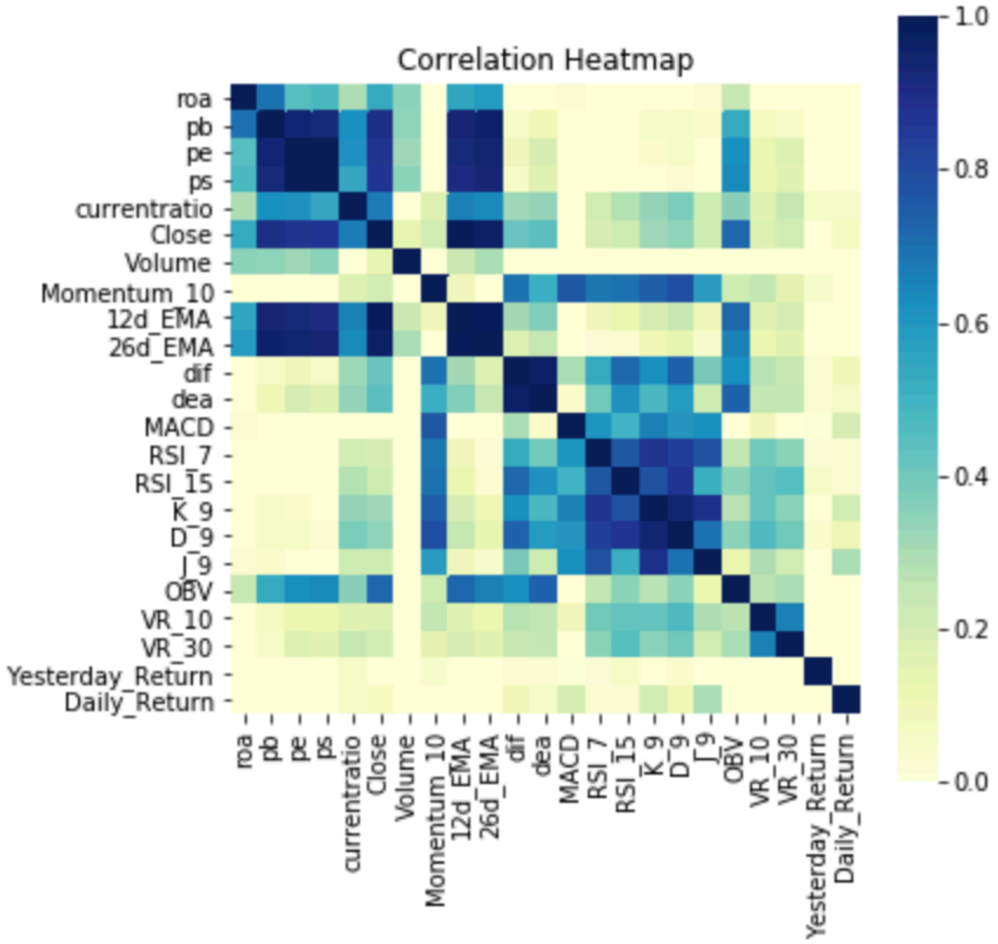
There are totally 511 pieces of data, in order to better evaluate our model, we need to split the whole dataset into 3 groups, train, validation and test data. The training set will conain 365 observations and will contain values from the begining of the data 2018-05-03 to 2019-10-04 (depicted by the blue line in the plot below). The validation set will contain 73 observations and will go on until 2020-01-17 (depicted by the yellow line below). The testing set would be the period straight after validation set, which will contain 73 observations and go on untill 2020-05-01.

```
plt.figure(figsize=(12,6))
plt.plot(df['Close'].iloc[0:int(len(df)/7*5)])
plt.plot(df['Close'].iloc[int(len(df)/7*5):int(len(df)/7*6)])
plt.plot(df['Close'].iloc[int(len(df)/7*6):])
plt.legend(['train', 'validation', 'test'])
plt.title('AAPL Stock Price')
```



Because we have a lot of features, we can plot the heatmap of the correlation matrix.

```
fig, ax = plt.subplots(figsize = (9,9))
sns.heatmap(df.corr(),vmin = 0, xticklabels= True, yticklabels= True,
square=True,  cmap="YlGnBu")
ax.set_title("Correlation Heatmap")
```



According to the Heatmap, we notice that most of the features have low correlation, which make it easier for regression.

Algorithms and Techniques

We will use three machine learning algorithms to explain and estimate the stock return.

1. Supported Vector Regression (SVR)

The **Support Vector Regression (SVR)** uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to **minimize error**, individualizing the hyperplane which maximizes the margin, and that part of the error is tolerated.

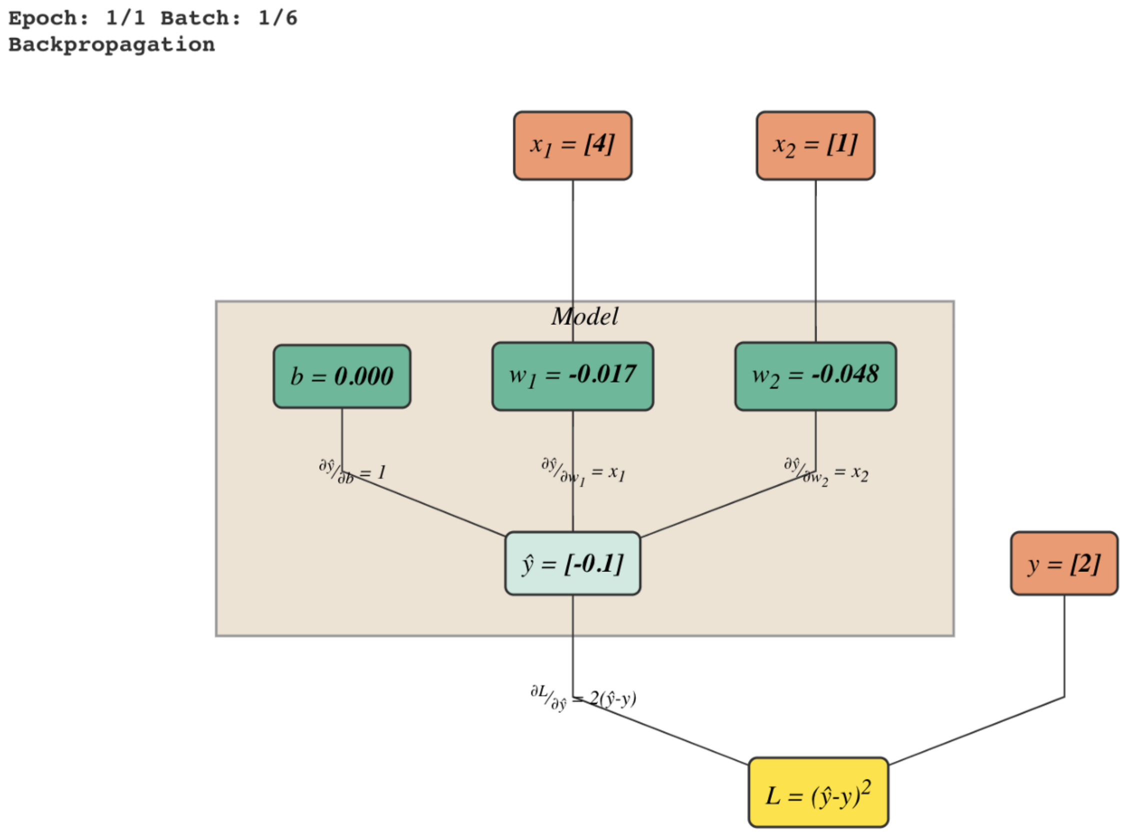
2. Stochastic Gradient Descent

Gradient Descent is the process of minimizing a function by following the **gradients of the cost function**. This involves knowing the form of the cost as well as the derivative so that from a given point you know the gradient and can move in that direction, e.g. downhill towards the minimum value. In Machine learning we can use a similar technique called stochastic gradient descent to minimize the error of a model on our training data. The way this works is that each training instance is shown to the model one at a time. The model makes a prediction for a training instance, the error is calculated and the model is **updated** in order to reduce the error for the next prediction.

This procedure can be used to find the set of coefficients in a model that result in the smallest error for the model on the training data. Each iteration the coefficients, called weights (w) in machine learning language are updated using the equation:

$$w = w - \alpha\delta$$

Where w is the coefficient or weight being optimized, α is a learning rate that you must configure and gradient is the error for the model on the training data attributed to the weight.



Linear regression does provide a useful exercise for learning stochastic gradient descent which is an important algorithm used for minimizing cost functions by machine learning algorithms (the graph above is an example).

3. Random Forest Regression

Random forest is a **Supervised Learning algorithm** which uses ensemble learning method for **classification and regression**. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or **mean prediction** (regression) of the individual trees. The advantage of random forest is that it runs **efficiently** on large databases, It can **handle thousands of input variables** without variable deletion, also it has an **effective method for estimating missing data** and maintains accuracy when a large proportion of the data are missing. However, random forests have been observed to **overfit for some datasets** with noisy regression tasks.

Benchmark

As mentioned above, linear regression is a simple form to use stochstic gradient decent algorithm. We can use a linear model fitted by minimizing a regularized empirical loss with SGD as our benchmark model.

III. Methodology

Data Preprocessing

As metioned earlier, we need to calculate some fundamental features and technical features. As for fundamental features, we can access to it directly by Quandl package.

1. Fundamental Features

RoA: Return on Assets (ROA) is an indicator of how well a company utilizes its assets, by determining how profitable a company is relative to its total assets.

$$RoA = \frac{Net\ Income}{Total\ Assets}$$

P/B: The Price-To-Book (P/B) ratio measures the market's valuation of a company relative to its book value.

$$P/B\ Ratio = \frac{Market\ Price\ per\ Share}{Book\ Price\ per\ Share}$$

P/E: The price-to-earnings ratio (P/E ratio) is the ratio for valuing a company that measures its current share price relative to its per-share earnings (EPS). A high P/E ratio could mean that a company's stock is over-valued, or else that investors are expecting high growth rates in the future.

$$P/E\ Ratio = \frac{Market\ Price\ per\ Share}{Earnings\ per\ Share}$$

P/S: The price-to-sales (P/S) ratio is a valuation ratio that compares a company’s stock price to its revenues. It is an indicator of the value placed on each dollar of a company’s sales or revenues. It is a key analysis and valuation tool that shows how much investors are willing to pay per dollar of sales for a stock.

$$P/S\ Ratio = \frac{Market\ Value\ per\ Share}{Sales\ per\ Share}$$

Current Ratio: The current ratio is a liquidity ratio that measures a company's ability to pay short-term obligations or those due within one year. The current ratio is sometimes referred to as the “working capital” ratio and helps investors understand more about a company’s ability to cover its short-term debt with its current assets.

$$Current\ Ratio = \frac{Current\ Assets}{Current\ Liabilities}$$

2. Technical Features

MACD: Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA. A fast EMA responds more quickly than a slow EMA to recent changes in a stock's price. By comparing EMAs of different periods, the MACD series can indicate changes in the trend of a stock. It is claimed that the divergence series can reveal subtle shifts in the stock's trend.

$$MACD = 2 \times (DIF - DEA)$$
$$DIF = EMA(12) - EMA(26)$$
$$DEA = EMA(DIF)$$

RSI: The relative strength index (RSI) is a momentum indicator used in technical analysis that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset.

$$RSI = 100– \frac{100}{1 + \frac{Average\ of\ Upward\ Price\ Change}{Average of Downward Price Change}}$$

KDJ: KDJ indicator is a technical indicator used to analyze and predict changes in stock trends and price patterns in a traded asset. KDJ indicator is otherwise known as the random index. It is a very practical technical indicator which is most commonly used in market trend analysis of short-term stock. The calculation of KDJ is a little complicated.

OBV: On-balance volume (OBV) is a technical trading momentum indicator that uses volume flow to predict changes in stock price. OBV shows crowd sentiment that can predict a bullish or bearish outcome.

$$OBV = OBV_{prev} + \begin{cases} volume & if\ close > close_{prev} \\ 0 & if\ close = close_{prev} \\ -volume & if\ close < close_{prev} \end{cases}$$

VR: Volatility Ratio (VR) is technical analysis indicator used to identify price ranges and breakouts.

$$VR = \frac{TR}{TR_{prev}}$$

Momentum: Momentum is the measurement of the speed or velocity of price changes.

$$Momentum = V - V_n$$

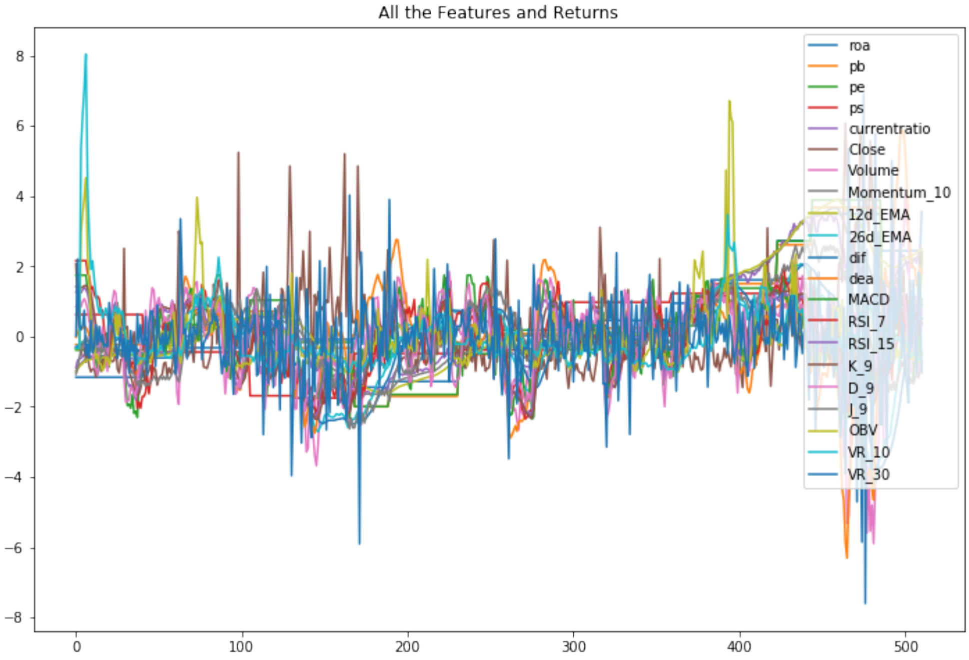
where V is the latest price and V_n is the closing price n days ago.

After calculating above features for each stock, we should save the results in csv format.

3. Data Normalization

Because the 'size' of the features varies a lot, for example, volume is really large and some features are less than 1. We need to normalize the data. Also, as we cannot predict the future, we only train the scaler in train and validation dataset, and use it to scale the test dataset.

Still take Apple ('AAPL') as an example, we can plot all the features after normalization (see the figure below)



Through the above figure, we notice that it seems that there is some relationship between those variables.

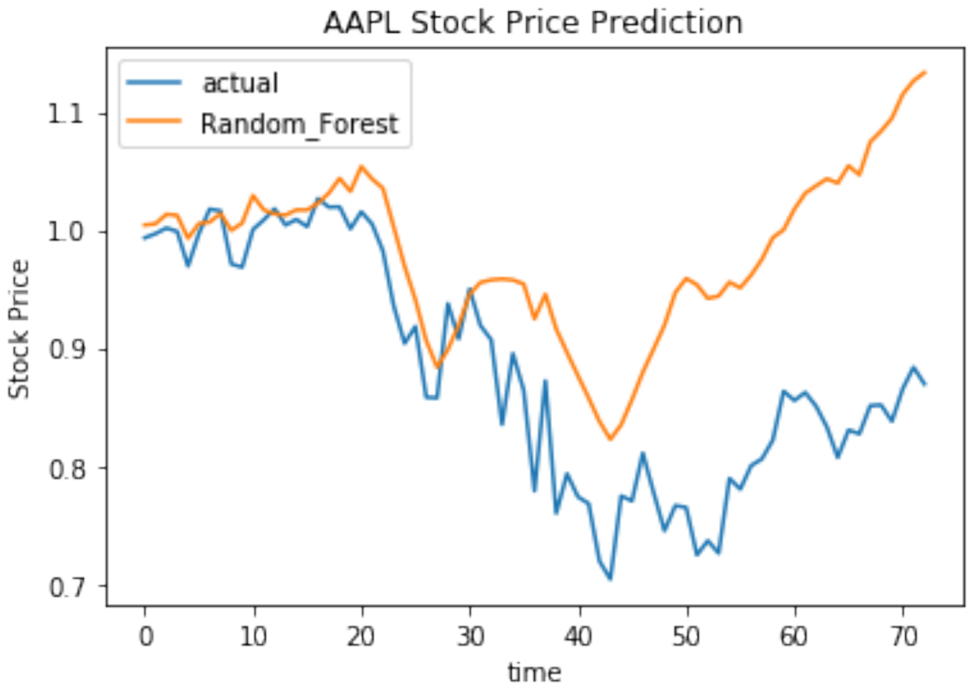
Implementation

For each stock, we first download and prepare the data as stated above. Then split the dataset into approximately 5:1:1, respectively train, validation and test set. Finally run three machine learning regressors including Supported Vector Machine Regressor with radio basis kernel and polynomial kernel, Stochastic Gradient Decent Regressor and Random Forest Regressor, and evaluate the model by calculating MSE. A good regression model explains the most part of the stock price by which we can predict the stock price for the next period with the features we calculate, that is to say it gives us a criterion to select stocks.

As for our Apple ('AAPL') example, we get the following MSE respectively.

Python Package	Regressor	Validation MSE
sklearn	svm (radio basis)	0.322423
sklearn	svm (polynomial)	1.346406
sklearn.linear_model	SGDRegressor	0.332687
sklearn.ensemble	RandomForestRegressor	0.271519

According to the above table, in this case, **Random Forest Regression** gives us the least Validation MSE. Apply this regression model to the test data, we get MSE as 0.0012855454101981022.



Blue line is the actual value while yellow line is the random forest value. From above, we notice that Random Forest Model performs great in this case, it captures the main changes in stock price. We will repeat the above steps for all the stocks in Dow Jones.

Refinement

Originally, I just used Stochastic Gradient Descent Regression and Supported Vector Machine Regression models, after testing, I added Random Forest Regression model in the model set, and finally found it did a great job at least in the AAPL example.

IV. Results

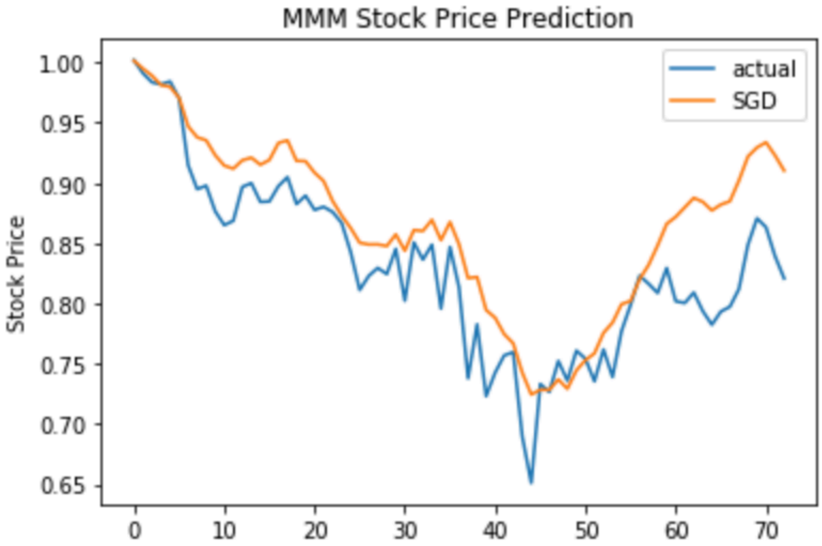
Model Evaluation and Validation

Apply the above procedure to all the stocks in our stock pool, we find that Stochastic Gradient Descent Regression (SGD) and Supported Vector Machine Regression (SVM with rbf kernel) performs better in most cases.

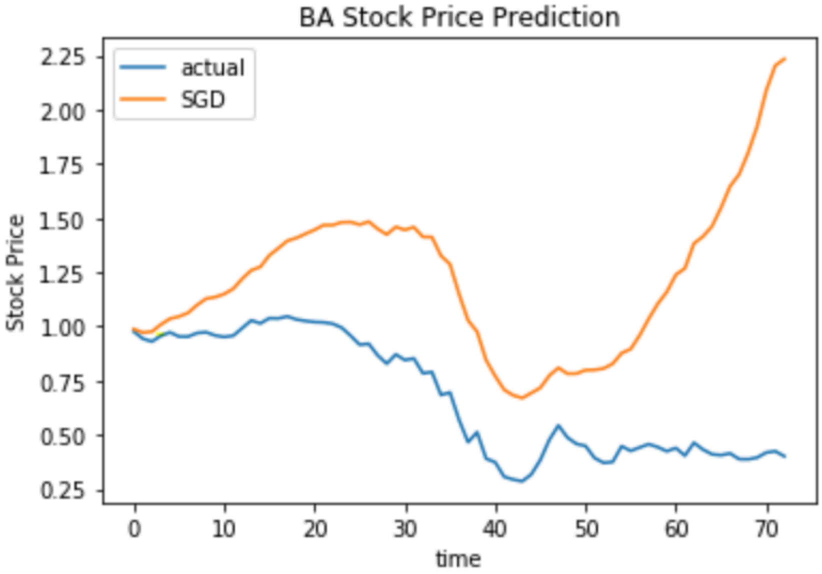
Justification

Below are some other examples of our model.

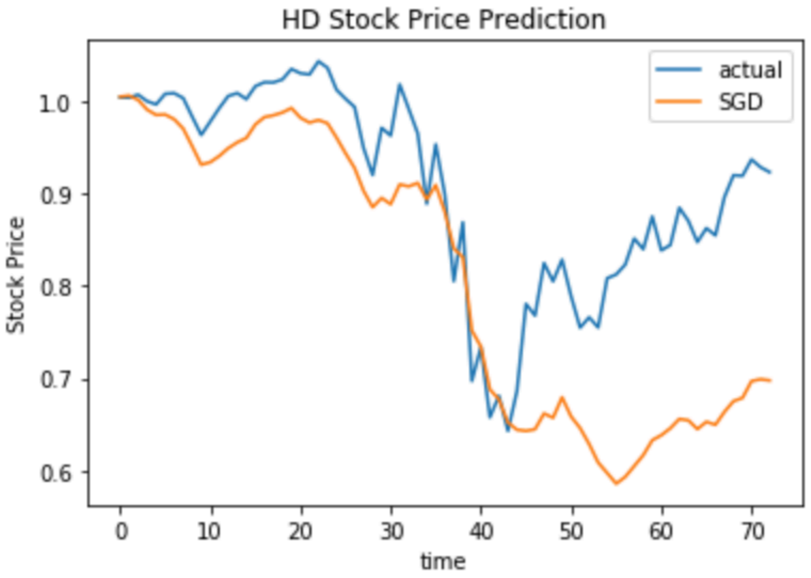
```
MMM
Best Model: SGD
Min MSE is 0.0007262315440505807
```



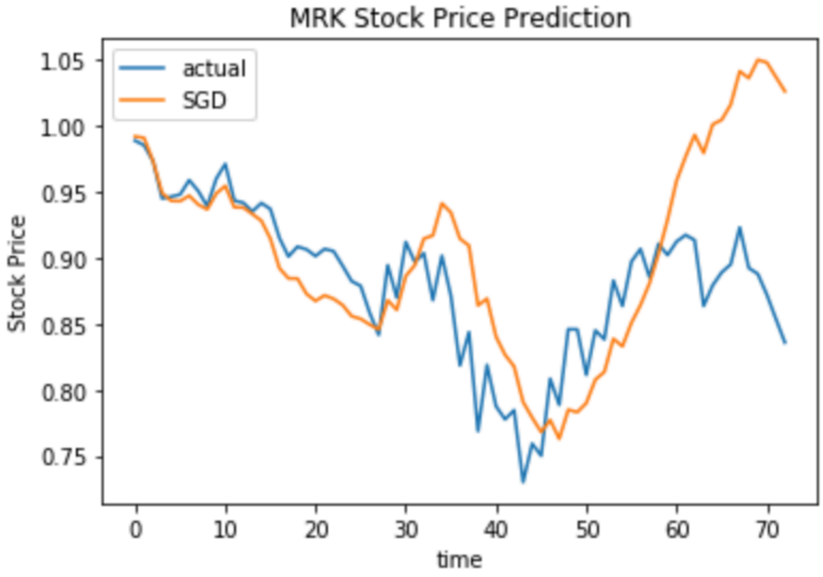
```
BA
Best Model: SGD
Min MSE is 0.0039096756292357655
```



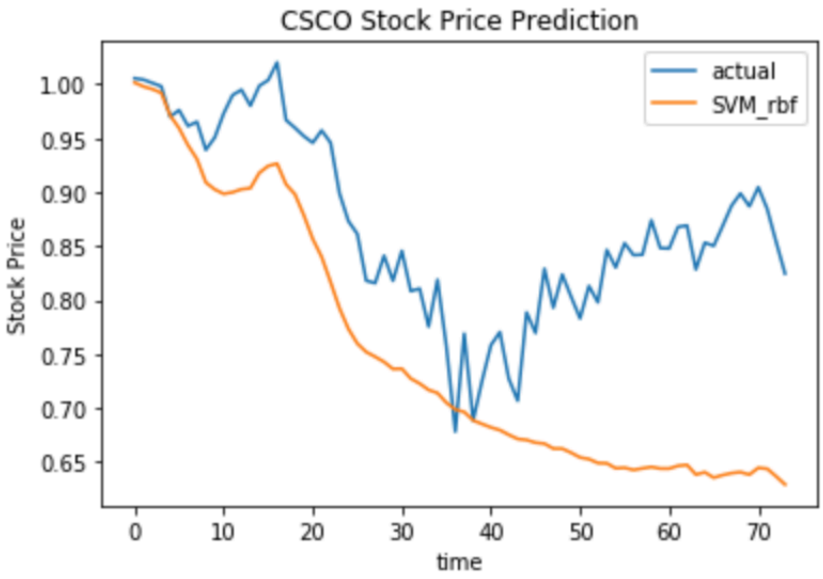
```
HD
Best Model: SGD
Min MSE is 0.001267861493004799
```



```
MRK
Best Model: SGD
Min MSE is 0.0005216105287110476
```



```
CSCO
Best Model: SVM_rbf
Min MSE is 0.0013908569591495573
```



According to the above prediction figures, we can see that our model successfully captures the main trend and changes in the stock price.

V. Conclusion

Visualization

I've already talked about it in the last section, please see the above figures

Reflection

According to the above graphs, we find that in most cases, our model performs well and has a quite small MSE, which means that our model can capture the main changes and trends in the stock. Especially, linear regression model with SGD performs best in the most cases. It tells us sometimes SIMPLE is BEST. We can use the simplest linear regression model to fit the data, and achieve satisfying results.

This method gives us some meaningful suggestions on how to select stocks. We can do the regression on a regular basis such as once every ten days, update our predictions on stock returns and select stocks with higher potential returns.

Improvement

There are still some aspects that we can consider, We can use NLP or try dimension reduction techniques to improve the algorithm. As for the factors, we can do more researches to mine more effective factors to make a better prediction.

Finally, financial market is changing rapidly everyday, factor may not be effective for a long time, since every quant will use it and then it will be ineffective eventually. Thus, we need to adjust our factors and models dynamically and regularly to maintain the whole system.