Project Mithril: Final Report[1]

Shiyun Zou

[1]Modules to be graded: ML and Visualization

1. Executive Summary

Legend has it that many successful traders can consistently beat the market by using "technical analysis". However, main stream finance academia still holds on to efficient market hypothesis and believes that the existence of successful traders is just a matter of probability: if you have 10000 traders toss coins to decide buy or sell for 10 times, you should expect to see 10 of them make correct bets all times. In a word, it is believed that legendary traders are just made of probability and survivor bias.

The project is my first step to study whether we can build a machine that acts like a good trader, who sees the various indicators and makes correct predictions based on them.

I start with a simple linear regression model because it is a standard tool used by financial researchers. It fails unsurprisingly and I then examine the possible reasons, which is the real pupose of using a linear model. The second baseline model is a decision tree classifier, which not only takes care of the non-linearity nature of the problem, but also (to some extent) echoes the way that traders make decisions. For the advanced model, I choose to use convolutional neural network because (1) it better imitates how traders value various indicators of a moving time window; and (2) it does not rely on assumptions about underlying distributions.

The results of the three models are summarized in the table below.

Figure 1: Summary of 3 Models

Model	Type of Problem	Accuracy Rate
Linear Regression	Regression	0.4609
Decision Tree	Classification	0.4564
CNN	Classification	0.6085

2. Introduction

Modern finance theory is based on a set of assumptions that are not entirely realistic. The most important ones include (1) market is efficient and (2) returns are random. Based on these assumptions, mainstream finance academia denounces the idea of active investing. Especially, the school of so called “technical analysis”, which speculates based on studying statistical trends, is regarded nonsensical. Reading stock charts is no better than reading palms.

However, there are still various counterexamples of the successful traders from the technical analysis school who generated non-trivial excess returns over a long time period. Some of them have openly talked or even lectured about how they “read the charts”. It is worth noting that the traders referred to here are not necessarily “quantitative traders”. They are alert to various quantitative market signals, they have their unique rules to process these signals, and they calculate the odds - all in their brains, not on computers.

For this project, my goal is to build several models that behave like technical traders. The models take stock market signals that a typical technical trader examines as inputs, and generate a trading decision (buy or sell) as the output.

The first baseline model uses simple linear regression. Despite the fact that the stock time series does not support some of the linear regression assumptions, I choose to begin with this model because it is a standard tool used by financial researchers. Examining why this model fails, I give several proposals of possible causes and proceed to make respective changes in later models.

The second baseline model is a decision tree. I choose to switch to this model mainly because its decision boundary is not linear. Also, the model is somewhat similar to how a person makes decisions - some traders, from my observation, make a sequence of binary choices in deciding their trades.

The advanced model is a convolutional neural network (CNN). The problem with stock market analysis is that we do not understand everything going on in the market. The practice of using "expert knowledge" and build trading rules based on such knowledge bears little chance of bringing beneficial inductive bias. The advantage of using a neural network is that it brings minimal inductive bias and can handle complex, dynamic, or seemingly chaotic data better.

3. Literature Review

When studying the relationship between asset price and market data, mainstream finance researchers focus on the "fundamental data" rather than the "technical indicators". The most established class of models is, of course, the factor model that was originally proposed by French and Fama, with numerous variations coming along over the decades. The figure below, obtained from Ken French’s website, shows the most recent performance of the commonly studied factors.

Figure 2: Current Factor Returns of Fama-French Models

	October 2018	Last 3 Months	Last 12 Months
Fama/French 3 Research Factors			
Rm-Rf	-7.67	-4.45	5.66
SMB	-4.66	-5.89	-1.88
HML	3.41	-1.48	-9.11
Fama/French 5 Research Factors (2x3)			
Rm-Rf	-7.67	-4.45	5.66
SMB	-4.42	-6.21	-2.90
HML	3.41	-1.48	-9.11
RMW	0.77	1.29	3.52
CMA	3.46	2.28	1.59
Fama/French Research Portfolios			
Size and Book-to-Market Portfolios			
Small Value	-9.07	-9.30	2.01
Small Neutral	-11.16	-11.09	1.39
Small Growth	-13.57	-9.93	7.15
Big Value	-6.26	-6.65	-0.84
Big Neutral	-4.99	-2.94	4.81
Big Growth	-8.57	-3.06	12.23
Size and Operating Profitability Portfolios			
Small Robust	-11.64	-11.62	1.86
Small Neutral	-10.23	-10.23	0.95
Small Weak	-12.21	-9.70	5.71
Big Robust	-7.42	-1.92	13.62
Big Neutral	-6.54	-4.56	2.33
Big Weak	-8.39	-6.43	2.72
Size and Investment Portfolios			
Small Conservative	-10.50	-8.74	7.93
Small Neutral	-10.29	-8.36	3.71
Small Aggressive	-13.25	-12.13	1.05
Big Conservative	-4.75	-2.09	7.84
Big Neutral	-6.35	-4.25	3.60
Big Aggressive	-8.92	-3.27	11.54

On contrary, in finance academia, technical analysis has received very little attention. The distaste for technical analysis makes perfect sense because the technical school is based on assumptions that are contradictory to the assumptions of modern finance theory. As Prof.

Aswath Damodaran of NYU summarized, technical analysis has 4 basic assumptions.[1]

- Price is determined solely by the interaction of supply and demand.
- Both rational and irrational factors govern supply and demand.
- Stock prices tend to move in trends which persist for an appreciable length of time, regardless of minor fluctuations in the market.
- Changes in such trends are caused by changes in supply and demand. These changes can be detected in the action of the market itself.

When prominent finance researchers look into the topic of technical analysis - if they ever care to - they tend not to use the black box tools offered by neural networks and their peers. For example, when Prof. Andrew Lo conducted a in-depth research on technical indicators in 2000[2], the major tool he used was kernel regression. Interestingly though, he did suggest at the end of the paper that the new statistical learning theories, which already had successful applications in handwriting or face recognition, may bring more insights into the analysis of technical indicators.

In fact, since as early as the 1990s, computer scientists have already started trying out various machine learning techniques on the stock market data. For instance, in 1997, a paper by Prof. Ramon Lawrence [3] at University of British Columbia proposed a neural network to predict the stock price of South Africa. The model generates superior performance and is seen as being adequate to rebut the efficient market hypothesis. The model uses 63 indicators which can be divided into 7 categories, which are shown in the figure below. Put together, these indicators can tell a story of the stock's fundamental strength, the statistical patterns of the stock price, and the macro economy condition worldwide.

Figure 3: Indicators in Lawrence[1997]

1. fundamental(3) - volume, yield, price/earnings
2. technical(17) - moving averages, volume trends, etc.
3. JSE indices(20) - market indices for various sectors: gold, metals, etc.
4. international indices(9) - DJIA, etc.
5. gold price/foreign exchange rates(3)
6. interest rates(4)
7. economic statists(7) - exports, imports, etc.

[1] http://people.stern.nyu.edu/adamodar/pdfiles/invphiloh/techanal.pdf

[2] ANDREW W. LO, HARRY MAMAYSKY, AND JIANG WANG; Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation

[3] Ramon Lawrence, Using Neural Networks to Forecast Stock Market Prices

A review of literature published in 2010 by Dase R.K. and Pawar D.D.[4] shows that researchers started with artificial neural network to predict financial markets, and later used models specifically tailored for time series data analysis, including the time delay neural network, multi-branch neural network, etc. In the early 2000s, researchers started to combine traditional models used in financial time series analysis (such as GARCH) with neural networks. I think this combination to some extent bridges the gap between standard financial research and novel computer techniques. But due to the limited time available for the project, I am mostly relying on machine learning techniques and little financial context.

In the most recent years, computer scientists also compared stock price prediction with image detection. In the past year along, there were several papers on using convolutional neural networks to predict stock price. For instance, in Sreelekshmy Selvin[2017][5], it is demonstrated that CNN's archetecture is especially good for capturing regime shifts in stock market, and can outperform other deep learning models under certain conditions. Intuitively, this makes sense because the "sliding window" scheme is analogous to how traders process information in their head: scanning various data series in a specific, rolling time window to recognize price patterns. Therefore, I choose to use CNN for the advanced model.

[4] Dase R.K. and Pawar D.D., Application of Artificial Neural Network for stock market predictions: A review of literature

[5] Sreelekshmy Selvin, Stock price prediction using LSTM, RNN and CNN-sliding window model

4. Data

4.1 Time Series Data Gathering

The raw data mainly comes from two sources: CRSP and Bloomberg. For the project, I am choosing iShares SP 500 ETF (ticker: IVV) as the tradable security. It is an ETF established by BlackRock, with the purpose to track the performance of the SP 500 index. The time series of IVV is obtained from CRSP, with a daily frequency and a range from May 2000 to June 2016. All the technical indicators are calculated from the IVV time series.

As discussed in the Literature Review, besides the technical indicators, I also add a few economic indicators to the advanced model, to test whether the incorporation of additional, fundamental indicators can improve the model. The table below summarizes the complete dataset composition.

Figure 4: Data Source Summary

Time Series Name	Description	Reason for Inclusion	Source
IVV	iShares S&P 500 ETF The dataset includes price (open, high, low, close) and volume series.	It is the chosen security to be traded.	CRSP
Brent Crude Oil	1st month futures price	Crude oil price contains information of world macro economy, geopolitical tensions, inflation, which can potentially affect the market's expectation of future stock performance.	Bloomberg terminal
Gold	1st month futures price	Same as oil.	Bloomberg terminal
CAPE	Shiller PE	Shiller PE is an important indicator of how expensive the stocks are.	R. Shiller website
VIX	CBOE VIX index	It implies market's expectation of volatility.	CRSP
UST	1yr, 2yr, 5yr, 10yr YTM of UST	The absolute level and curve shape of treasury yields indicate market liquidity, macro conditions and monetary policies, which may affect the market's expectation of future stock performance.	CRSP

4.2 Technical Indicator Calculation

Among the many popular technical indicators, I choose the following ones which - based on literature review - are more frequently used. It is worth noting that different practitioners may have their own way of calculating such indicators. There is not a universal standard of how each of them should be calculated. Therefore I take the discretion to specify the formula for each indicator.

Figure 5: Formula for Technical Indicators

Indicator Name	Formula	Purpose of the Indicator
Moving Average	simple arithmetic average of the price for the past 50-day, 100-day, 200-day windows	Traders typically compare current prices with MAs. They also look at the angle of the MA lines.
MACD: moving average convergence divergence	EMA(12-day) - EMA(26-day), where EMA means the exponential moving average of certain given period of time	It is used in trend-following strategies, often seen as a signal for buy or sell.
RSI: relative strength index	RSI = 100 - 100/(1 + RS), where RS = ema gain (14-day) / ema loss (14-day)	It calculates the ratio of a security's recent gains to its recent loss. It is interpreted as a signal for over-buying or over-selling.
SO: stochastic oscillator	SO = 100(C - L14)/(H14 - L14), where C is today's close price, L14 is the high of the last 14 days, and L14 is the low of the last 14 days	It measures the current price relative to the price range over a certain period of time. It also indicates over-buying or over-selling.

4.3 Data Preparation

The final dataset for the baseline model consists of 2 parts, the original times series (price and volume of IVV) and the technical indicators which are calculated from the original times series. For the advanced models, the fundamental time series (namely, VIX, UST, CAPE, oil, gold) are merged onto the baseline dataset, on the same date.

A snippet of the baseline dataset is posted below. Note that the price column here is not normalized yet. I leave that column intact for now, in order to calculate returns over different time windows for different models later.

Figure 6: Dataset Snippet

date	low	high	close	volume	open	macd	rsi	50d_avg	100d_avg	200d_avg	so
20010307	-0.439921	-0.455396	126.55	-0.669027	-0.438876	-1.713257	-0.062340	-0.328303	-0.265805	-0.107318	-0.765715
20010308	-0.429719	-0.447085	126.92	-1.015534	-0.439084	-1.572642	0.318968	-0.330083	-0.267201	-0.108762	-0.303249
20010309	-0.494888	-0.480540	123.47	-0.244510	-0.468421	-1.636192	-0.013111	-0.333621	-0.270195	-0.110353	-1.609645
20010312	-0.609195	-0.550982	117.78	-0.285677	-0.534586	-1.982363	-0.009977	-0.340270	-0.274561	-0.112813	-1.609645
20010313	-0.615233	-0.596905	119.72	-1.009871	-0.601999	-2.125731	-0.012221	-0.346038	-0.277851	-0.114912	-1.036723

Original time series from CRSP

Calculated technical indicators

To clean time series of stock prices, the most difficult thing is to make sure that all the events such as dividends, split or reverse split, inclusion or exclusion from index etc. are taken care of in a consistent manner. CRSP has already done the housekeeping work regarding these events. The dataset from Bloomberg only concerns times series of commodity futures, which do not have such issues. Therefore, the remaining cleaning work mainly includes (1) check for NAs and decide whether to drop or interpolate the missing data; (2) check for numbers that do not make economic sense, for example, negative prices; (3) normalize all the independent variables, because they come in very different scales.

Due to the inclusion of the 200-day average series, the first 200 rows of the $200d\_avg$ column now contains NA. After dropping these first 200 rows, the remaining dataset has 4357 rows. (Deleting these rows only shrinks the size by 4%) Since the time series obtained from Bloomberg terminal had missing data, the merged dataset for advanced models contains 260 empty entries. Because the 260 entries sum up to only 0.037% of the total 4357 * 16 entries, either choice of dropping or interpolating the missing data is reasonable. In order to keep the date sequence intact, I choose to interpolate the missing data, using simple linear interpolation. One final step is to make sure that the date ranges of X and Y are correct. Since the model is trying to make predictions of future stock returns, the technical indicators calculated at the market close on day T are used to predict the return on day T+n, where n differs in different models. It does not make economic sense if the model is using the indicators from day T to predict the return on day T as well.

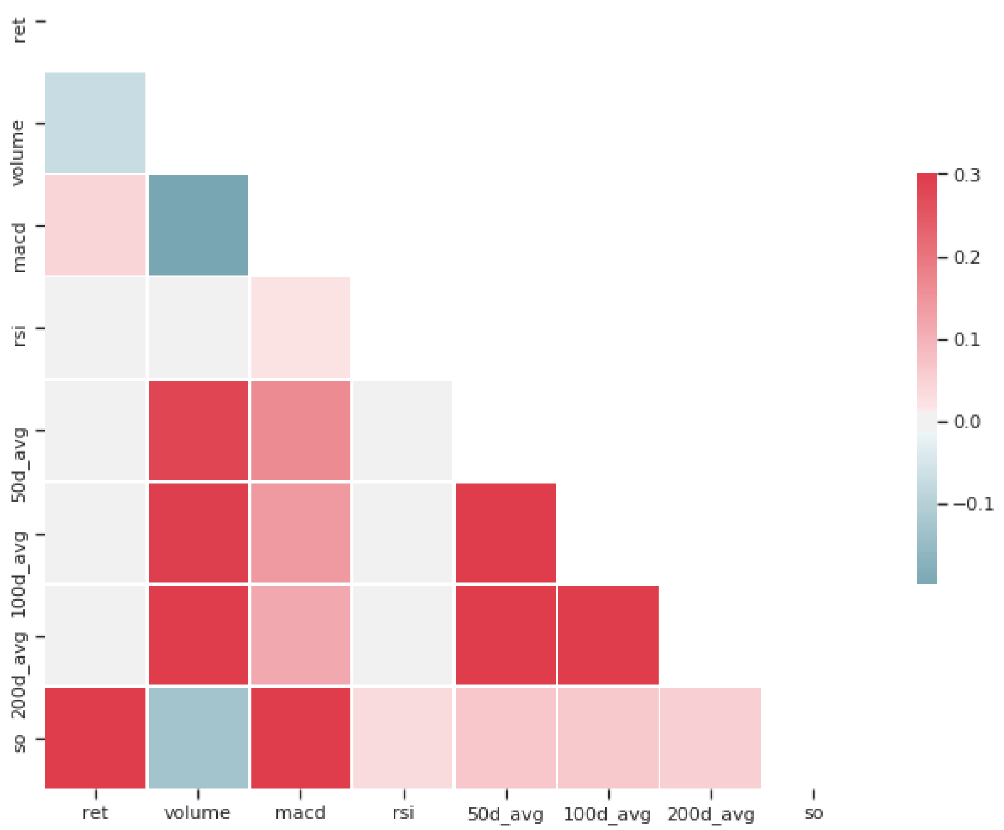
4.4 Exploratory Data Analysis

For the EDA part, my major focus is to test the following features of the time series. Some sample plots are shown below, and the rest of the results will be discussed with greater detail in Section 5.1.

- correlation between different features (i.e. the technical indicators) and the stock return, which is summarized using a correlation matrix
- multicollinearity between the features, which is shown in a pairplot
- normality of each time series, which is shown by a qq-plot
- autocorrelation of return series

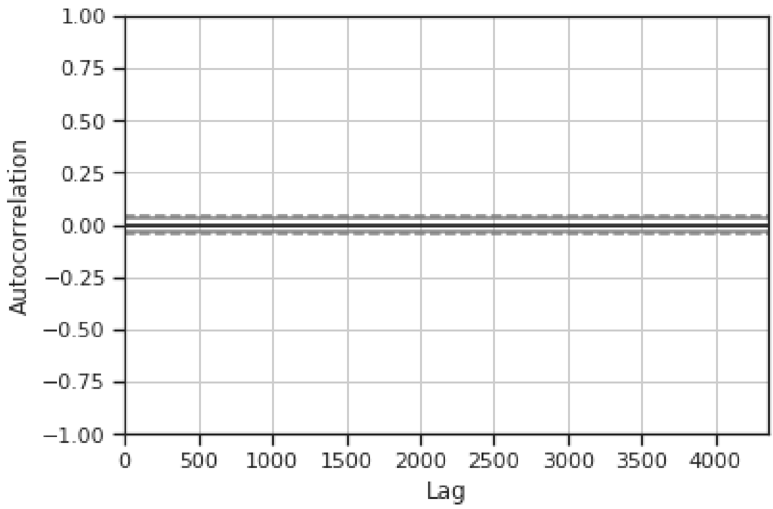
The correlation heatmap (as shown below) seems to indicate that the IVV return series does not have high correlation with any of the independent variables.

Figure 7: Sample EDA: Correlation Heatmap



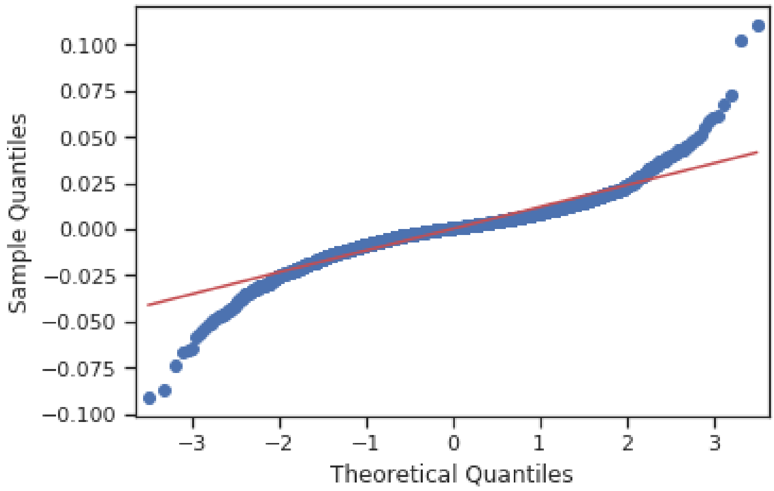
The autocorrelation of the return series shows that the IVV returns have no autocorrelation at the daily frequency level. This is not surprising and is in line with the basic assumptions of efficient market hypothesis. Moreover, the heatmap shows that there exists certain degree of multicollinearity between the independent variables.

Figure 8: Sample EDA: Autocorrelation of IVV Return Series



Contrary to the assumptions in modern finance theory, the returns of IVV are not normally distributed. The qq plot posted below and imples that the return distribution of IVV (between May 2000 and June 2016) has strong fat tail. Indeed, the Kurtosis of the distribution is as high as 8.6.

Figure 9: qq plot: Entire IVV Return Series



5. Baseline Model: Why Linear Regression Does Not Work

The initial exploration of the dataset casts some doubt on whether we should use linear regression model because some of its assumptions do not hold for the dataset. However, I choose to start with linear model because I want to study why this model does not work. More importantly, the linear regression model is one of the most commonly used model in financial time series analysis. To what extent does the model make sense? How can we improve from here?

5.1 Model Setup

There are 7 independent variables and 1 dependent variable. The 7 independent variables include the 6 technical indicators and the trading volume. The complete time series has 4356 days of entries, and is splitted into a training set of 3000 rows, and a remaining testing set of 1356 rows.

Figure 10: Model Setup

Independent Variables (X)	Dependent Variable (y)
MACD indicator on day T	Return of IVV on day T+1
RSI indicator on day T	
50-day-simple moving average, calculated on day T	
100-day-simple moving average, calculated on day T	
200-day-simple moving average, calculated on day T	
SO indicator	
Volume (total number of shares of IVV traded on day T)	

5.2 Results

Unsurprisingly, the result is as complete failure in both the training set and the testing set. The accuracy rate for the training set is as low as 0.46 - it is even worse than tossing a coin. (The prediction isd considered correct if the prediction and the actual number are in the same direction.)

As the figures below show, in both the training set and the testing set, the model underestimates the magnitude of daily growth rate in either long or short direction. Basically it is not responding to the market signals. Moreover, since the cutoff of the training set happens to lie soon after the end of the 2008 financial crisis, the testing set is in a bull market, whose underlying distribution may be very different from that of the training

Figure 11: Scatter Plot: Actual vs Predicted

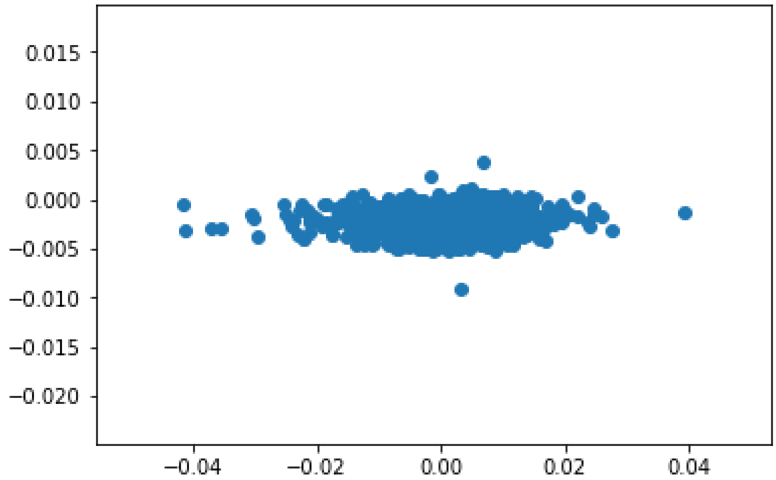
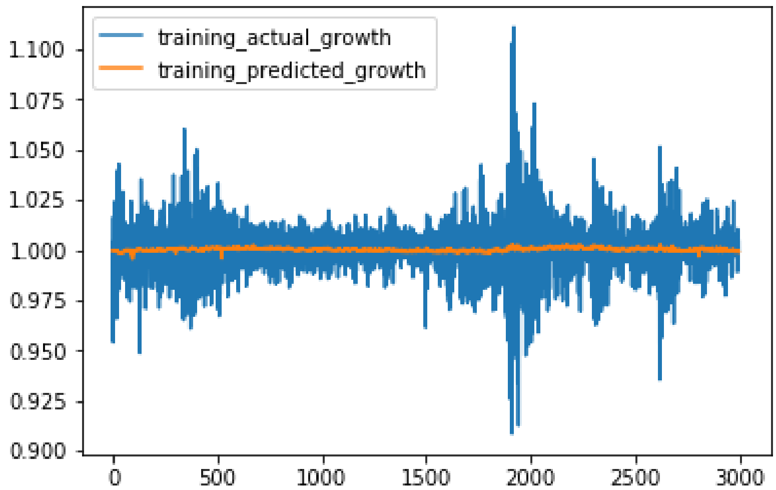
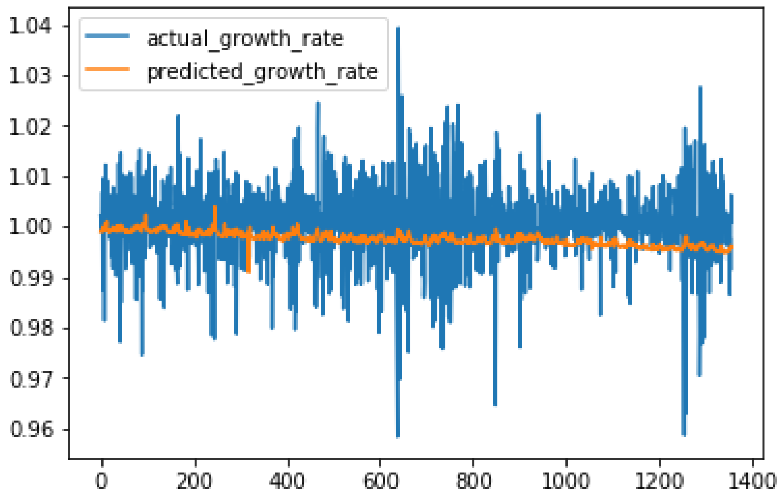


Figure 12: Actual Rate vs Predicted Rate: Training Sample



set. As a result, the same model systematically underestimate the positive gains in the testing set. If a portfolio is built on this model, the investor can lose all the money because essentially, the model is suggesting the manager to short what turns out to be a long bull market.

Figure 13: Actual Rate vs Predicted Rate: Testing Sample



5.3 Why it fails

The fact that the model behaves much worse in the testing set in fact addresses an important issue in financial time series analysis: the regime shift. Although almost every student in finance has been taught to calculate the mean and variance of the return time series, and the assumption that returns can be modeled by normal distribution, the fact is that the returns do not come from the same distribution. Calculating the mean and variance of 20 years' return is a convenient way to gain some initial insights into the market, but is far from enough to build some solid understanding of the market.

Specifically for the dataset at hand, the distributions of the training sample and the testing sample are visibly different. The Kurtosis for the training period is as high as 7.41 while the Kurtosis for the testing sample is 3.15.

Figure 14: Return Distribution: Training Sample

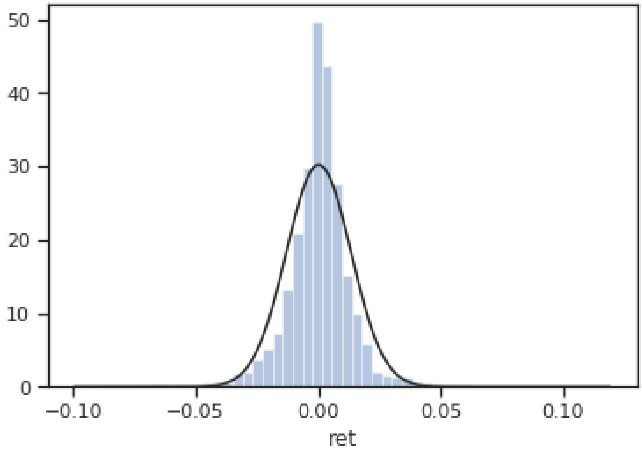
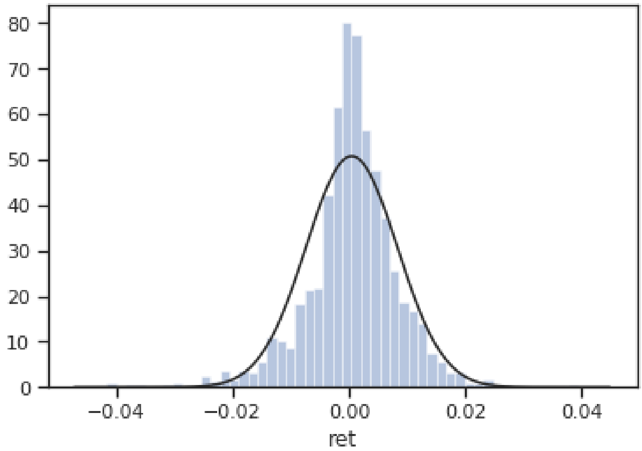
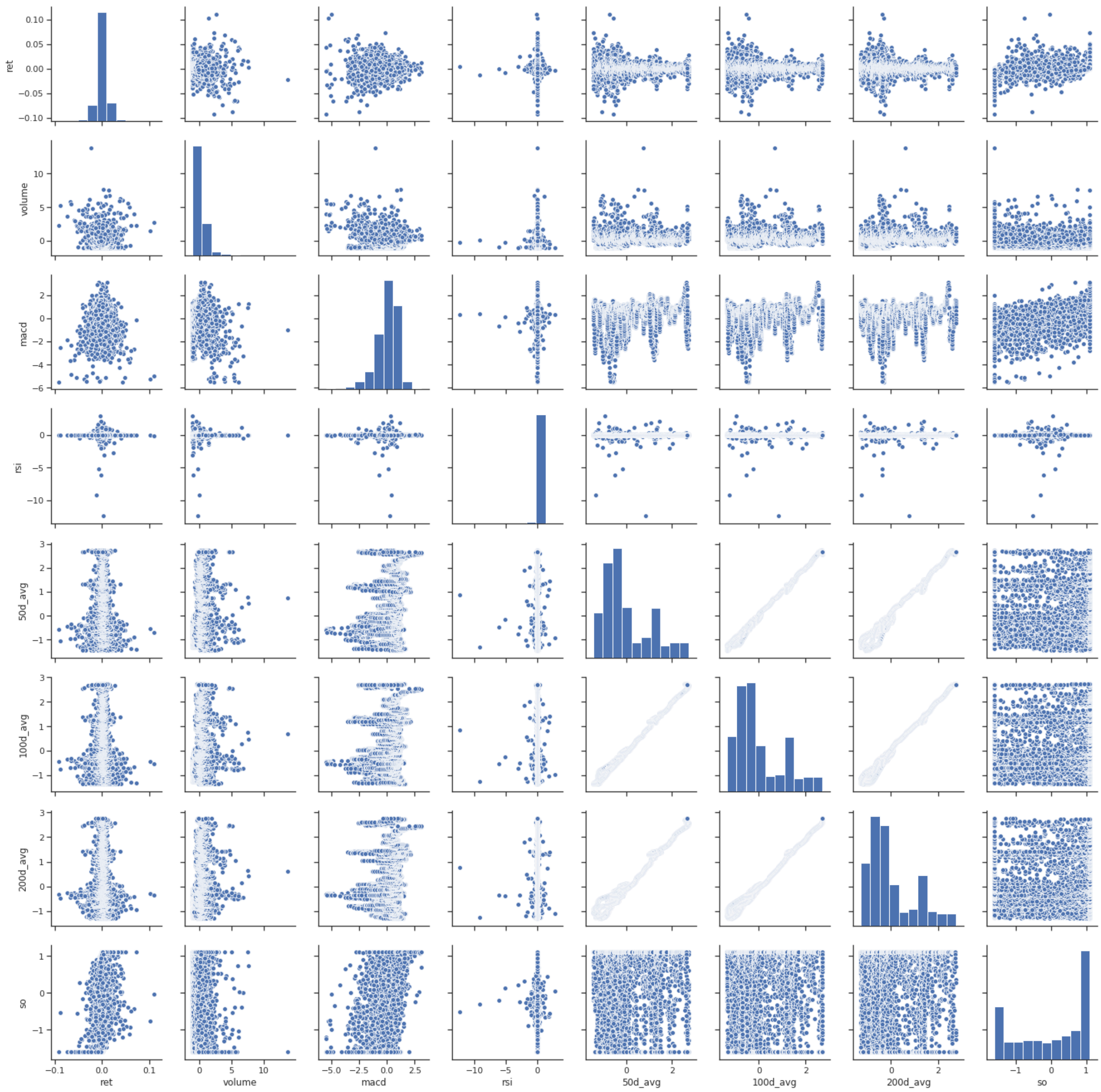


Figure 15: Return Distribution: Testing Sample



There are other assumptions that are not supported by the dataset. To begin with, the pairplot below shows that there does not exist a linear relationship between any of the independent variable and the dependent variable. Therefore, for my second baseline model, I will need to try non-linear models.

Figure 16: Pairplots Between All Variables



Furthermore, the residuals of the linear model, for both the training and the testing samples, are not normally distributed, as indicated by the qq plot below.

Figure 17: QQ Plot for Residuals: Training Set

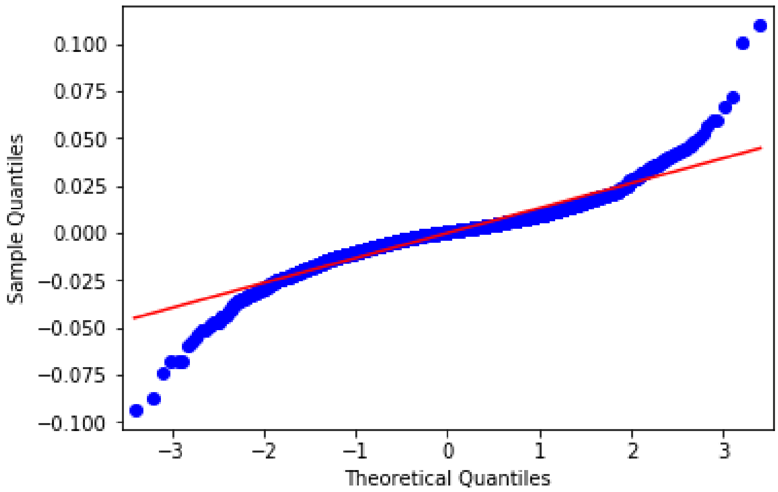
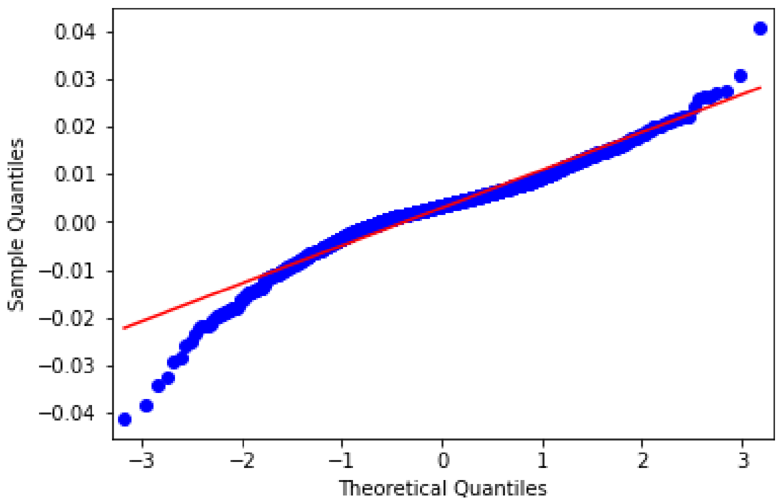


Figure 18: QQ Plot for Residuals: Testing Set



To sum up, linear regression is not suitable for the dataset at hand, due to the fact that many of the crucial assumptions are not met.

5.4 How can it be improved

Based on these findings, I make the following changes in the rest of the models.

- Use a non-linear model: since there is no linear relationship between these indicators and the return series, future models need to be not based on the linear assumptions.

- Change from a regression problem to a classification problem: the non-existence of autocorrelation and linearity suggests that maybe classification is more suitable. Intuitively, predicting the exact number is more difficult than predicting the direction.
- Change the dependent variable: it might be more feasible to predict the return over a period of time, for example, a week or a month, than the return of a single day.
- Most importantly: use a non-parametric model that does not assume the returns come from any particular distribution or that the returns come from the same distribution over time.

6. Baseline Model 2: Decition Tree is No Better

6.1 Model Setup

The set of independent variables is exactly the same as that of the linear regression. The only change for the decition tree (DT) model is that I change the problem from regression to classfication. On day T, instead of trying to predict the rate of return on day T+1, the model now tries to predict whether the return is positive (labeled as +1) or negative (labeld as -1). As discussed above, the task of predicting the exact magnitude of price change is mission impossible.

Figure 19: Model Setup: Decision Tree

Independent Variables (X)	Dependent Variable (y)
MACD indicator on day T	2 Classes: 1: if return of day T+1 is positive -1: if return of day T+1 is negative
RSI indicator on day T	
50-day-simple moving average, calculated on day T	
100-day-simple moving average, calculated on day T	
200-day-simple moving average, calculated on day T	
SO indicator, calculated on day T	
Volume (total number of shares of IVV traded on day T)	

6.2 Results

Without any tuning, the default decision tree classifier generates an accuracy result of 0.4564, which is slightly lower than the accuracy rate of the linear model. As can be interpreted from the confusion matrix below, the DT model is also "breaish" in that it makes more "sell" predictions in the testing set. This is the same as the linear model.

Figure 20: Confusion Matrix

	Predicted: Sell	Predicted: Buy
Actual: Sell	0.3503	0.1018
Actual: Buy	0.4366	0.1114

6.3 What Can Be Improved

One of my initial arguments for using the DT model is that its decision boundary is non-linear. Before seeing the results, I have the expectation that DT will be at least as good as linear regression. The results have left me with more questions than answers, to which my current knowledge base is not adequate to solve. Some thoughts and questions are listed below.

- Retrospectively speaking, if we go back to the pairplot, we can see that the data pairs have neither clear linear relationships nor clear-cut clusters. Although DT model does not need the linearity assumption, it is still tangled by the fact that the clustering of data (if any) does not have clear edges.
- Although the DT model is non-parametric, it is still affected by the fact that the testing sample distribution is drastically different than that of the training sample distribution. In this case, DT seems not any better than linear regression in terms of picking up regime shifts.

So far, I have tried (but failed) to mitigate the (1) non-linearity and the (2) distribution change issues, by swithching from the linear regression model into the decition tree model. But neither model has examined the interaction between the independent variables. Moreover, they differ from a real human trader in terms of how far back into history to look for signals. Unlike the models, human traders would mostly like use the information generated from only N days back in time, rather than from 20 years ago.

Therefore, for the final model to be studied, new changes should include (1) allowing interaction between independent variables and (2) adding moving windows of past time series.

7. Advanced Model: is CNN the key?

7.1 Structure of the Neural Network

The neural network has 2 convolutional layers, followed by two fully-connected hidden layers. After each convolutional layer, we apply max-pooling with a kernel size of 2 by 2. A detailed description of the structures is attached below.

Figure 21: Structure of the Neural Network

```
Net(
  (conv1): Conv2d(1, 2, kernel_size=(5, 5), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(2, 2, kernel_size=(5, 5), stride=(1, 1))
  (fc1): Linear(in_features=24, out_features=20, bias=True)
  (fc2): Linear(in_features=20, out_features=20, bias=True)
  (fc3): Linear(in_features=20, out_features=2, bias=True)
)
```

7.2 Data

New independent variables are added into the advanced model, which are the oil price series, gold price series, the treasury yields series and the VIX (a market volatility/fear index) series. The addition is a very high level proxy of the overall market conditions, which human traders will certainly take into consideration.

One major change on the dependent variable is that, instead of predicting daily returns, on any day T, the CNN model now predicts the return of T+14 over T+1.

The time series is divided into 3 parts: training, development and testing, each with a row number of 3000, 600, 700.

7.3 Results

With the initial data setup, the accuracy on the development set exhibits a strange pattern: it hits the highest score after the first epoch at 59.3% and decreases afterwards. Upon closer examination, I find the model always predicts "buy" (class 1). In fact, it behaves like a majority baseline.

I suspect that some input features do not provide reliable signal for the prediction. In the attempt to try fix the model, I reexamine the indicators more closely, and decide to remove the ones which have wierd-looking distributions.

7.4 Reflections and Modifications

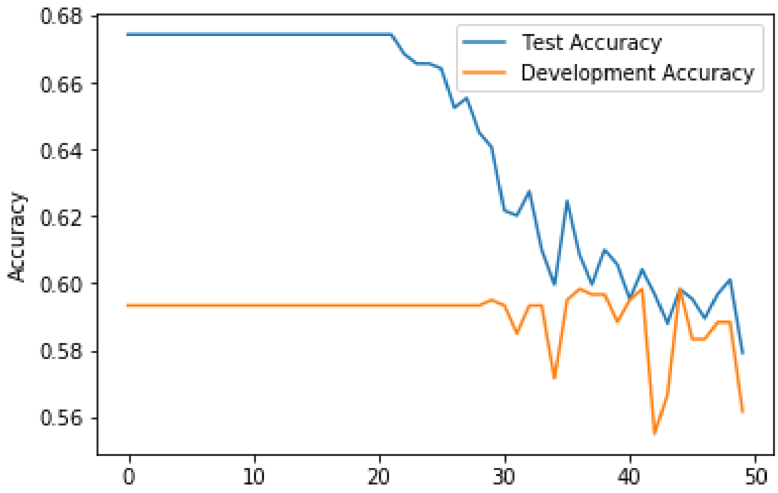
In light of this, I proceeded to remove ust1 (1 yr Treasury yield), ust5 (5 yr Treasury yield) and ust10 (10 yr Treasury yield). I give the new dataset a try and the model now behaves as expected, it finally starts to learn. The best accuracy rate of the development set is 0.5983, associating with an accuracy rate of 0.6085 in the test set.

However, it appears that the accuracy on the devleopment set does not correlate with the accuracy on the test set.

Figure 22: Standard Deviation of Independent Variables

```
macd          1.003617
rsi           0.299232
50d_avg       1.016643
100d_avg      1.011311
200d_avg      1.000113
so            0.996776
oil           0.913153
gold          1.074721
vix           1.133866
ust1          0.837241
ust2          0.842708
ust5          0.863660
ust10         0.896431
ust5vs2       0.940923
ust10vs2      0.916028
dtype: float64
```

Figure 23: Accuracy Rate: Development vs Test



This seems to suggest that (1) test and development sets come from different distributions (even though they both come from the same bull market), (2) dataset size may be too small, (2) the input features do not provide reliable signal. In other words, noise-to-signal ratio is too high.

8. Future Work

During the research process, some questions arise and remain to be answerd.

- Why a simple DT model is even worse than a simple linear regression model? How can the DT model be improved?
- Try different technical indicators.
- Filter the noise of the time series.
- Tune the architecture of the CNN to find the best hyper-parameter setting.

9. Links to Inveractive Visualization

Distribution and Fit

Candlestick

Scatter Plot

Scatter Plot