THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

HAROLD AND INGE MARCUS DEPARTMENT OF
INDUSTRIAL AND MANUFACTURING ENGINEERING

INSIDER TRADING PATTERNS:
A PRICE INDICATOR FOR US EQUITIES

PAUL BOEHRINGER
Spring 2019

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree in Industrial Engineering
with honors in Industrial Engineering

Reviewed and approved* by the following:

Guodong Pang
Associate Professor in The Department of Industrial and Manufacturing Engineering
Thesis Supervisor

Catherine Harmonosky
Associate Professor and Associate Department Head of
The Department of Industrial and Manufacturing Engineering
Honors Adviser

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

This paper investigates the potential of insider trading as a price indicator for US equities along with a non-stochastic relationship between insider transactions and prices. Further tests are done to see if there is a difference in the quality of information coming from insiders buying compared to selling, as the majority of the current literature tends to discredit the use of insider selling as an input in any investment strategy.

Analyzing the returns of a company's stock for a series of holding periods ranging from five days to four years after each insider buy and sell shows insiders have a significant advantage on timing regarding when to buy and sell their own stocks. After confirming a difference in the behavior of buys and sells a price forecasting model was built. Including either buying or selling information during model training increase prediction accuracy compared to a model without any insider trading information. Further, the most accurate models include both buy and sell information. However, there was no significant difference between the quality of buying and selling info.

There is also significant evidence that price formation is non-stochastic based on the financial state and insider trading patterns for a company over time. Lastly, both a linear regression and neural network were used to make pricing predictions, while the linear regression was more accurate for pricing predictions the activation function in the network models allowed for significantly more accurate timing predictions, that is when the high and low prices will occur in a quarter. The insights from this study are found using the price, insider trading, and quarterly financial histories from 3,505 companies with 3,021,444 insider transactions and 139,986 financial quarters shared between these companies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. Guodong Pang, my thesis adviser. Three years before any work on this paper began, he introduced the topic of financial engineering to me through a project on failure propagation in financial loan networks. This was the inspiration for me getting involved in this sort of research. Further, Dr. Pang provided great mentorship during the whole writing process. Without his introduction to research on universal price formation patterns in equities based on path dependent properties many of the takeaways from this study would not have occurred.

Further I would like to thank Dr. Soundar Kumara who pushed me to learn how to code. Without this initial kick in the right direction I would have never had any of the skills that made this work possible.

I would also like to thank Dr. Catherine Harmonosky for her consultation during the process and the time it took her to review this work.

Chapter 1: Introduction to Insider Trading

Insider trading is defined as an individual who has access to nonpublic information making a transaction on a publicly traded stock or security. Rules and regulation regarding insider trading are heavily dependent on the country of an exchange. This means that any analysis being done regarding equities should only include those from the same country unless stringent checks have been done to make sure that the countries have very similar laws. In the United States insider trading is allowed upon the condition that transactions do not rely on information not in the public domain.

Chapter 1.1: Tracking Insider Trades

The United States has a regulating body in charge of stock exchanges where these commodities are traded publicly. This is called the Security Exchange Commission (SEC, 2018). One of their main responsibilities is tracking insider trading; the SEC does this using various forms. The main one being a form 4. The two others are the form 3 & form 5 (SEC, 2018). A form 3 is required to declare an initial statement of beneficial ownership of securities; these are most commonly filed when directors or officers receive stocks as a form of signing bonus when joining a company. A form 5 is required to report any transactions that should have been reported on a form 4 and were not reported within the required time period. The other reason to file a form 5 instead of a form 4 is the rare case when a transaction is eligible to remain private until a deferred date. Either of the situations requiring a form 5 are rare as most companies have strict rules for their own insiders such that the company must pre-approve trades. Companies often go one step further and report transactions to the SEC for the individual executing the transaction such that the company can be sure the form is submitted properly and in a timely manner. Note, the second case where the SEC approves a deferred release of information is extremely rare. As form 3s and form 5s are more rare, the main source for insider trading information is the form 4s on record with the SEC.

There is some confusion over how insider trading is tracked and why it is allowed; this may be because the term implies an illegal action. However, the government is essentially required to allow some forms of insider trading due to the way companies compensate their employees. An example of legal insider trading would be a CEO selling their stock options off at a predetermined rate so they can diversify their portfolio; this is a common scenario. Further as the sale of stock is structured it is clear the insider is not trading based on private info. An illegal trade would be a company’s chief financial officer unloading shares before a bad financial statement release which led to a drop in the stock price. This is illegal as the company’s financials were privileged information to that individual. These are clear cut examples but applying the rules can be more complicated.

Due to the nature of inside information, regulations are extended to interactions between two companies. For example, if an Apple employee knew well in advance of the public that Apple was planning on using Samsung screens for the iPhone X this person would not be legally allowed to buy Samsung’s stock. This is because the information is privileged due to the nature of a private deal between the two companies. This becomes harder to track as the Apple employee isn’t buying their own company’s stock, though it would probably be easy for a prosecutor to argue this individual traded on that information.

To finish, a final example that many do not realize is technically illegal. Take the same insider at Apple, but this time they tell a friend about the deal with Samsung. Here the friend could buy Samsung stock and the two could share the profits. While this activity is illegal, it would likely go unnoticed. Cases such as this are almost never prosecuted as the 3rd party buyer of Samsung stock has no formal access to the information making the burden of proof significantly more difficult as the Apple insider likely knew to tell their friend by word of mouth.

Overall, when an individual makes a legal transaction on stock of a company they have privileged information regarding they are required to file a form 4. The time period required for this filing with the SEC has changed over time as filings have become easier thanks to the internet. Currently this period is within two days of a transaction (SEC, 2018).

Chapter 1.2: Spectrum of Regulation Regarding Insider Trading

As with most legal topics there is a wide range of opinions on how insider trading should be regulated. Some economists argue that insider trading could benefit markets stating that banning the practice is attempting to get rid of an even playing field. These experts believe giving equities markets a fair playing field is an impossible goal as large institutions are viewed to have huge advantages over individual investors. These institutions have people whose job is to notice abnormal insider transactions. Admittedly, removing the regulation on insider trading does have a somewhat paradoxical position if the goal is to make markets fair. An integral part of the argument for deregulation recognizes that insiders would gain an advantage; the hypothesis is their transactions would then convey more valuable information to markets. This would undoubtedly help insiders and the aforementioned institutions, but the libertarian argument states the public also benefits making it a win for all parties involved. To see this look at Henry Manne's argument.

Manne is one of the most well-known advocates for deregulating insider trading. He postulates that insider trading would prevent scandals because "if we allow insider trading there would be plenty of people in those companies who would know exactly what was going on and would sell shares accordingly" (Matthews, 2013). An anecdote where this situation comes to mind is the recent accounting scandal at Caterpillar Inc., the construction equipment enterprise. Their accounting department was fudging numbers to make the company look more appealing to investors. Under Manne's assumption, if trading on private information was allowed insiders from the accounting department would have started selling shares much sooner. When Wall Street caught on to this trend investment bankers would have sold too, potentially years earlier. It is clear that the large institutions are benefitting by getting out first. However much of the money these institutions manage comes from large pension and mutual funds which private citizens rely on for retirement.

Caterpillar's story was going to end in the stock dropping regardless of when insiders sold their shares due to knowledge of the accounting scandal. However, in the aforementioned hypothetical where insider trading is legal, the investors who made it out early would have had the additional time-value with their investment capital to allocate towards better opportunities hence promoting market efficiency. Further, if insider trading allowed markets to catch on sooner then the stock's price would have had less time to appreciate leading to less overall economic loss when the drop did occur (Yu, 2017). This real-world anecdote gives Manne's position good grounding as there is no way that insider trading would have hurt individual investors or the economy more than what actually happened. For terminology's sake, Manne is on the libertarian end of the spectrum of insider trading regulation. While Manne makes an argument rooted in overall economic benefit for society some hard-core libertarians share Manne's position for more elementary reasons. Some libertarians more simply believe the government has no right to dictate what its citizens can do with their own private assets.

The hard-core libertarians run into trouble with their argument when considering potential malicious intent of insiders. This leads to the middle of the regulatory spectrum which argues that some deregulation would be good. This stance believes that current regulations are too restrictive and therefore hinder growth. Most in this area agree that insider trading is somewhat unfair to the public however there are also benefits to allowing the practice.

This centrist position mostly pushes for companies to form their own policies for employees. With this practice the generally accepted policy is that the government should only get involved in the case of severe ethical violations (Bainbridge, 1998). Similar to libertarians, supporters of this policy believe it is self-evident that allowing insiders to inform the market will promote market efficiency. This position also states that privileged access to information offers managers a form of compensation. This information then becomes a performance incentive to all employees by making promotions more valuable. From the perspective of the corporation, allowing companies to regulate themselves in a manner they see fit also benefits the company as stock options become a much more valuable incentive for employees at no cost to the company. Current regulations also drive up operating costs as large companies have to hire legal teams and SEC filing services to be sure they are complying properly. It seems reasonable that a company should know how to prevent egregious violations better than the government as the nature of privileged information varies greatly depending on corporation. An example where the government would get involved under the centrist policies would be insiders acting against the best interest of their company, its employees, and investors simply to make a quick buck by shorting the stock and then causing it to crash. Centrists believe strict enough punishment should effectively incentivize corporations to prevent this behavior.

Finally, there is the current and strict regulation of insider trading in the United States. This policy is argued from a stance focused on “fairness” for private citizen investors. Fairness is a fuzzy topic based heavily on an individual’s perspective. What is fair to an average citizen who owns Caterpillar stock in the aforementioned case may not seem fair to a Caterpillar accountant who has their 401k heavily reliant on their own company. In this case, the accountant may be forced to hold onto the stock as they would rather lose money than go to prison for insider trading, but their retirement funds could still take a steep hit. The current regulations don’t have a solution for this employee. While the current policy is focused on fairness to the public the other stances argue that the modern approach hampers economic growth which hurts the public. Some take this a step further to argue that fairness will never exist in capital markets stating it is preposterous to think that the “average Joe” has an equal chance of beating market returns against CEOs and CFOs at a Fortune 500 (Matthews, 2013).

Current market regulations on insiders are so strict that they rarely have a chance to transact on their company’s stock. Often companies only give individuals required to file a form 4 opportunities once or twice a quarter to sell stock. These days are set in advance as far from the release of a company’s quarterly financial statements as possible. While the government does not specifically require that companies adopt such policies, many choose to do so in order to avoid the potential of massive legal costs along with the bad public relations that come from insider trading scandals.

Whatever someone believes, it is clear the insider trading is a controversial topic. The quantity of legal debate & research done makes it evident that the government, markets, and public opinion believe insiders hold some advantage. Trading on specific information may still be illegal, but it does seem plausible that insiders know when good entrance and exit opportunities are around the corner. This begs the question, does the information tracking insider transactions offer valuable insights regarding the price movements of public equities for the future? The next chapter will go deeper into public research focused on insider trading's relationship with equity pricing.

Chapter 2: Relevant Literature

There is a lot of literature analyzing potential effects of insider trading regulation. Most use philosophical arguments or are based in economic theory. These are usually published out of law schools or economics programs instead of STEM based departments. This may be partially caused by the fact that records have only been available on the internet since 2005, however they were much more difficult to access until the SEC revamped their database in 2011. While public record lacks any empirical analysis on insider trading it is clear that financial institutions have done their own analysis early on and kept it out of public circulation in order to keep an advantage in markets. From here on only data-based analysis of insider trading is featured.

Chapter 2.1: Proof Insiders Posses an Advantage Over the Public

Ryle et al. (2017) shows insiders have historically had an edge in markets. The analysis focuses on open market orders from insiders based off of a company's form 4 history. Derivatives were ignored as they can be reported in various ways on the form making collecting this information more manual. This is justified considering derivative trades represent a small proportion of insider transactions. Moreover, any shorting within a company looks unfavorable from the eyes of the SEC. In response many companies have a flat ban on employees executing options.

Ryle's analysis shows that insiders have significantly better returns when buying their own stock compared to average market returns. Furthermore, the difference in returns increased when only considering trades of company officers. Officers are positions such as the chief executive, financial, & operations officers, along with treasurer, president, VP, or someone on the company's board (Core & Guay, 2004). Finally, the type of trade made a significant difference. General purchases of common stock saw the greatest returns compared to employees executing stock options. The hypothesized explanation is that options allow employees to buy the stock at a sub-market price and are often very structured regarding the timing of executions.

Next is a study that shows insiders have previously traded on knowledge of events such as bad accounting releases (Garfinkel, 1997). This was shown by the fact that a significant drop in trades was seen in the 30 days before a bad accounting release. A later study out of the Penn State Accounting department continued this work (Ke et al., 2002). This analysis is more telling as it bolsters the hypothesis that insiders trade based on bad accounting releases. This is done by comparing two time periods. The earlier period, also analyzed by Garfinkel, had less regulations and tracking of insider trading. The second is when federal regulations were tightened to prevent insider trading with a new program which offered a reward to those who reported such activities.

During the second time period with the new program the insider behavior of trading soon before bad releases which was found in Garfinkel's paper essentially disappeared. However, the new analysis discovers the phenomenon simply happens at a much earlier date. The change in regulation led to insiders trading on the knowledge of underperforming accounting disclosures up to two years beforehand as insider sales instead increased three to nine quarters ahead of a poor disclosure during the more regulated period. This type of behavior occurred more commonly for growth companies. An interesting corollary, there is almost always an abnormally low quantity of selling two quarters or fewer before a bad financial statement release during the latter period. This adds further justification to question the difference between buying and selling information.

Chapter 2.2: Evidence of Markets Reacting to Insider Information

So far, the literature covered has shown insiders possess an edge regarding market timing for their own companies. It is also clear that insiders have historically traded on privileged information such as future accounting releases. However, an insider beating the market is not the same as the market noticing and outside investors adjusting their stake accordingly. This section provides evidence that traders react to the information of insiders transacting by citing price movement behavior that is consistent with the timing of form 4 filings.

First is a paper which covers how insider trading evolved in 2002 after the Sarbanes – Oxley Act (Brochet, 2010). Brochet has a similar methodology to the two financial disclosure studies mentioned previously in that he analyzes the effect of new regulations on insider trading by comparing the market's behavior before and after new legislation. Brochet provides evidence that markets began to react in much greater force when the SEC provided a timelier filing standard for form 4s. This was section 403 of the Sarbanes-Oxley Act (or SOX Act). Before the introduction of the SOX Act insiders could report transactions up to a month after the fact. After the new law reports must be within two days. It is further claimed that other sections of the legislation drew attention to newly available information which made markets focus in on insider trading to gain an extra edge.

The SOX Act also led to a significant decrease in the volume of insider trading, Brochet attributes this to insiders now fearing scrutiny from the SEC. On the other hand it is clear that the

SOX act led to an increase in trading on filing of Form 4's. This phenomenon can be seen by the fact that after the SOX Act average daily trading was higher upon insider trading signals than it was beforehand. Pre-SOX daily volume was 1.03% higher than expected upon a form 4 submission for an equity. Post-SOX market volumes were 12.03% above average daily volume.

Rodgers et al. (2017) is the second paper that addresses the market's reactions to insider transactions. This focuses solely on the effects of a time delay for insider information arriving to large institutions vs. being published online. This analysis compares two periods, the earlier time involves certain institutions having access to information before the majority of others in US markets. During this time the SEC had a paid subscription service called the PDS (Public Dissemination System). This sent out information including form 4 filings to subscribers. On average these customers received information about 40 seconds before it was publicly available on the SEC's website. With only 20 subscribers to this service it was clear that price, volumes, and spreads responded to these filings 30 seconds before the information was available to the public. This proved that the PDS service provided a clear advantage of the same nature insider regulation attempt to eradicate.

This advantage is important for two reasons: first, it proves that insider trading information is clearly traded on in markets and large trading firms recognize the value of this information. Second, it shows the SEC adds to the unfairness associated with insider trading which clearly makes parts of the current policy inefficient. However, credit should be given where it is due; after the results for this study were published the Chair of the SEC at the time, Mary White, found the results so conclusive that she adjusted the SEC's policy. The new policy "ensures that EDGAR filings are available to the public on the SEC website before such filings are made available to PDS subscribers". This provided further opportunity for investigation thanks to the change in how markets receive their information.

After the adjusted policy was in place the same study published in 2014 was redone. This time with data from after the PDS Service change in 2015. The new data makes it clear that large institutions were still trading on the information. The same reactions to prices, volumes, and spreads were seen compared to when the PDS service was available (Rodgers et al., 2017). However, this reaction moved to the time that the data was publicly available. The fact that the reaction moved later to the same degree that the timing of information dissemination was pushed back further confirms that large institutions trade on insider transactions.

Chapter 2.3: Buying vs. Selling Information

It should be noted that a common feature most papers that analyze insider trading share is a disregard for insiders selling. The only papers with a specific focus on selling were Garfinkle (1997) and Ke et al. (2002). Others brushed it off, for example Ryle (2017) addresses this issue by citing evidence from a short article (Roberts, 2013). A quick review shows that this article cites no analysis and instead claimed common knowledge from Wall Street. That is, selling is not as effective of a signal because top executives will often liquidate their accumulated stock options when they need to make big purchases. Other times they simply feel their wealth is too tied up in their own company and would like to diversify to mitigate some risk. A quick internet search confirms this point of view. This common knowledge from wall street is given a bit more credibility as this is the same explanation Rodger et al. (2017) use when explaining why insider sells were not included in analysis. It is notable that this paper did directly affect SEC policy on the issue. However, this idea should still be taken lightly there is still no numerical analysis cited as justification. In all, the fact that there is contradictory evidence regarding selling means the relationship of insider selling and price movement will be an area of high interest in this paper. More specifically, whether insider buying information provides different value compared to insider sells in a price prediction model.

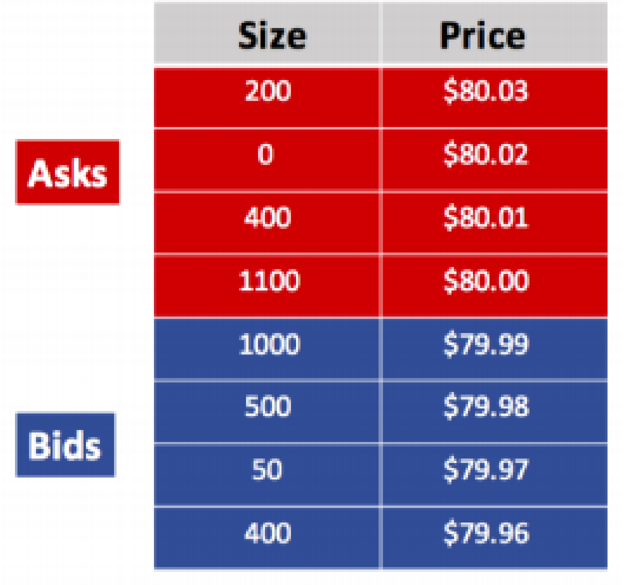
Before going further, I would like to present some anecdotal examples of why selling information will be so heavily investigated. First, in one section of Ryle's (2017) analysis, it is shown that insiders do sell at advantageous times to beat market returns. However, selling information is never incorporated into a model. This was justified by stating one would not want to purchase a stock when an insider sold. The second reason it will be investigated are historical examples of insiders dumping shares and the stock price underperforming soon after. Two recent examples are Twitter back in August of 2015 when insiders dumped $103 million before a significant price decline (Sun, 2015). More recently, Mark Zuckerberg has been observed unloading over $10 billion of his ownership in Facebook just in time to avoid a 33% dip since June of 2018 to date (Gajanan, 2017).

It should be noted that Brochet (2010) claims some concrete findings regarding selling information and price predictions. This is that stock returns are not more negative after insiders selling; it took controlling for multiple variables such as pre-planned transactions, reporting lag, litigation risk, and news to find a statistically significant correlation between selling and returns. However, this was just a side note in the study. However, the recent examples of share dumping provided offer anecdotal motivation to test selling information's value. Arguably the most significant reason for this focus is that no study found questions the value of insider purchasing information, however the value of selling information is consistently ignored with almost no empirical justification.

In summary, Garfinkel (1997) clearly shows insiders previously traded in close proximity to bad accounting statements in the late 90’s. Then in Ke et al. (2002) show a change in this behavior due to a new policy which gave monetary rewards for reporting insider trading violations. Later on Brochet (2010) shows that this behavior did not disappear, instead selling was pushed further forward before the date of bad news due to the Sarbanes-Oxley Act. This continuous trend of insiders pushing their trades forward before bad news is released leads investors to live in ignorance regarding bad practices of the companies they are holding for even longer. Instead current legislature stops markets from reacting as they should because insiders are specifically prohibited from selling ahead of bad info. This has compounded over time leading to longer periods of investors having capital tied to bad stocks. Once bad news surfaces the public already lost time value of that money.

Chapter 2.4: Universal Trends and Path Dependence in Price Formation

The final relevant topic is focused on building a pricing model of non-stochastic nature. Cont et al. (2018) set out to predict the next tick in price of any U.S. equity, be it up or down, based on the state of that company’s order book. Figure 1 shows an example order book to more clearly illustrate the concept.



	Size	Price
Asks	200	$80.03
	0	$80.02
	400	$80.01
	1100	$80.00
Bids	1000	$79.99
	500	$79.98
	50	$79.97
	400	$79.96

Figure 1. Example Order Book

The order book represents a snapshot of the supply and demand for a stock at a current point in time. The ‘ask’ rows in red show the current quantity of orders to sell a stock at a specific asking price. The bids in blue show the quantity of orders to sell a stock at a certain price. As new orders come in they are executed to who is next in line in the order book for a certain price.

In this example the spread is 1¢, this is the difference between the lowest sell (ask) price and the highest buy (bid) price. If an order for 500 stocks at $80.00 was placed then the corresponding ask quantity for that price would go down to 600 as the order would be executed immediately. Knowing the state of a company’s order book over time can inform investors of the direction a stock is going. This study shows there is a universal price formation mechanism that applies to all stocks based on the structure of an order book.

During this research models were trained to predict the next price movement given the current and a prior states of an equity’s order book. First using company specific data meaning a company's predictions were only based on that company's past order book history. This method yielded a prediction accuracy ranging from 65% - 75% depending on the company. The single company model had a mean overall accuracy of about 70%. Afterwards a universal model was trained; meaning it used data from all stocks for training with no consideration for what company data came from or what company it was making price predictions for either. This combined model out preformed others which were trained and tested using individual company or sector specific data. On average, the models trained universally were over 2% more accurate than company, sector specific, or any other type of dataset tried. This was true for testing sets from models that were trained with totally unrelated data. For example, testing with companies and time periods not included in the training set. The size of the testing sets from this paper are so large that these sorts of differences in performance indicate a universal behavior. This universal behavior was the main focus for this study; however, in this paper the same concept will be applied as it allows for much larger sets to train a pricing model. This is useful because insider trading data is less abundant as transactions occur less than order book structure changes which can vary hundreds of time per second. The second concept that will be used from this study is the path dependence property of price formation.

Path dependent price formation means that prices do not move based only on a company's current state. This is referred to as non-Markovian in statistics. This order book study makes it amply clear that using more order book states going back in time to make one prediction going forward leads to consistently more accurate predictions. This concept will be used such that the insider trading status for multiple quarters back will be used to make a prediction looking one quarter forward. The fact that the time frames change so much during the insider trading analysis will require a slightly different approach; this will be covered during the analysis sections.

The final area of interest that will be used as inspiration in this research is that deep learning models outperformed regressions by 5% - 10%. Neural networks for deep learning have been researched and implemented commonly in many applications regarding image and language processing. However, more recently their applications have been further reaching. The boosted performance that Cont et al. found in their research shows potential benefits of using a neural networks in a finance application where regressions would typically be the go to option.

Chapter 3: Data Gathering and Processing

This chapter describes the data collection and wrangling process. This is done in hopes of providing transparency for how the results were found such that they can be replicated if desired. It will start off with describing how the list of companies was found for analysis, after it will cover collection for the price history and company information necessary to get the insider trading history. Next, it will cover the process used to efficiently gather quarterly financial statements from a Bloomberg terminal. Finally it describes how all of this is put together into a comprehensive dataset for the models that will be built in chapter 4.

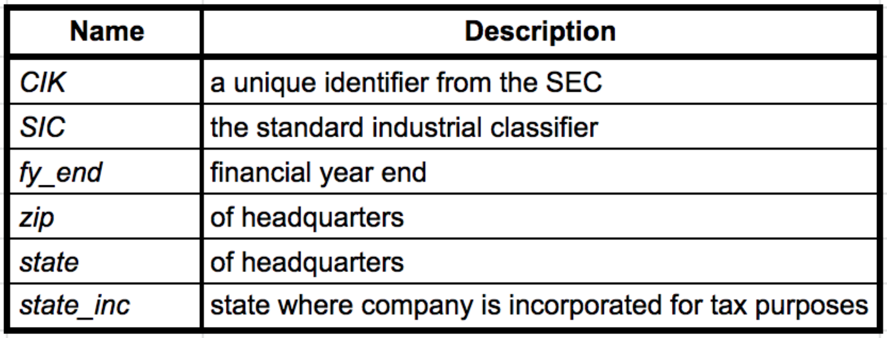
3.1: Creating a List of Companies for Consideration

To start off, a list of companies to be considered was created. This was done by going to zacks.com and running a screen for all companies they had on record which have ever been publicly traded in the US (Zacks, 2018). Zack's screen returns the following three fields: (1) Company Name, (2) Ticker, & (3) Market Cap. This query contained information for 7,787 companies. This is saved to a comma separated value (CSV) file. Now the list of tickers obtained will be used to gather more data about each company. This will then help with scraping the insider trading history for these companies. Many steps of data collection will be covered; after each step companies are thrown off the list when data is not available from that part of the collection process. When this happens, it will be clearly noted along with a summary of the relevant impacts on the dataset.

3.2: Getting Additional Company Heuristics & Price History

The next step is to get more company information and a price history for these companies. This is done in R using the *finreportr & BatchGetSymbols* packages (Seward, 2016 & Marcelo, 2018). First, *finreportr* is used to gather more heuristics about each company which will be used to scrape insider trading information. These are shown in Table 1.

Table 1. Information Collected on Each Company



Name	Description
CIK	a unique identifier from the SEC
SIC	the standard industrial classifier
fy_end	financial year end
zip	of headquarters
state	of headquarters
state_inc	state where company is incorporated for tax purposes

The package used, *finreportr*, gets this information directly from the SEC's website. For this reason, companies which *finreportr* returns no results are removed from consideration. This brings the list of 7,787 companies gathered from zacks.com down to 5,423 companies.

Next, *BatchGetSymbols* was used to retrieve the price history for each company. This price history is saved to a CSV with columns for the daily open, high, low & closing price along with daily volume. It is important to note that these price histories are adjusted for stock splits. The price history CSVs are kept in a folder for later; each is named according to its company's ticker. Note that the code used to collect this data can be found as Appendix Item A.

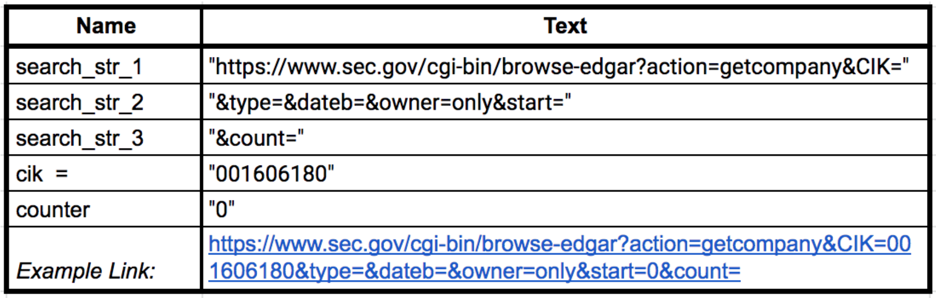
3.3: Scraping Insider Trades

Next the insider trading information for all companies had to be retrieved. This is by far the longest process taking about 5 to 10 minutes for each company. Because of the time investment price history CSVs from chapter 3.2 were checked to see if any companies could be removed. There was one company that needed to be removed as no price history was available leading to have 5,422 companies remaining for analysis. This one was removed because it was the phantom stock ticker for a company that somehow made it this far in the collection process.

Before going further, a quick divergence on phantom stocks as this will become more relevant later on. Phantom stock is an internal tool large companies have started to use as compensation. However, it cannot be publicly traded and insider rules actually do not apply to these shares. This will be discussed more later on.

With a sufficient list of companies compiled a scraper was built to retrieve all the insider trading histories for the list. This is done through the SEC's website by picking apart the structure of their EDGAR Database (Electronic Data Gathering, Analysis, and Retrieval). First it was found that the search results of a company's insider trading forms can be generated using the following formula for a search result link: *search_str_1 + cik + search_str_2 + str(counter*100) + search_str_3 + str((counter+1) *100)*. This link contains the accession numbers which is the first step toward getting insider transaction histories. An example of how one of these search result links are generated is provided in Table 2.

Table 2. Example Link Formation for Scraping Accession Numbers



Name	Text
search_str_1	"https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK="
search_str_2	"&type=&dateb=&owner=only&start="
search_str_3	"&count="
cik =	"001606180"
counter	"0"
Example Link:	https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK=001606180&type=&dateb=&owner=only&start=0&count=

The example link provided in a copy paste format is:

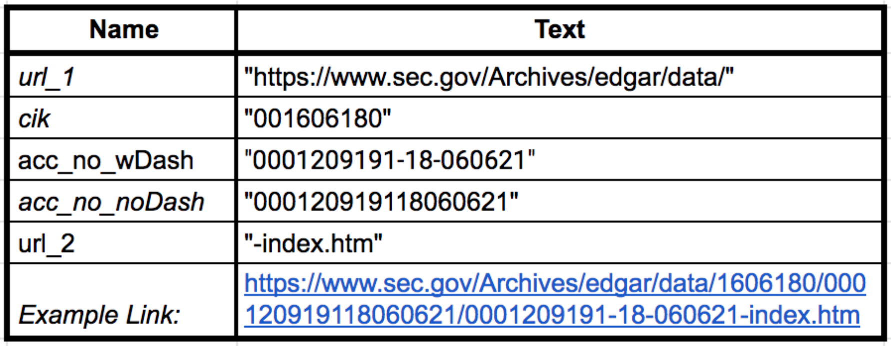
- https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK=001606180&type=&dateb=&owner=only&start=0&count=

Investigating this link will show an arbitrary search result for the company with ticker AAC which was output during the scraping process; AAC happens to have the CIK '001606180' listed in the table. While describing the scraping process an example link stemming from this AAC result will be provided all the way to one specific form 4 of information. This is to help any reader that would like to follow the path a computer took to scrape the info.

This search result link is generated using a loop in python for each company using their unique CIK identifier along with a counter. The counter is because the SEC website only allows the 100 most recent results for a company on one page. Each loop iterates through the next 100 results for a company by incrementing the counter by 1. The only piece of information needed from the links generated in this step is the accession number of each form 4. The accession number is a unique identification number the SEC has for all documents in their system. Numbers are assigned to documents in increasing integer order according to when the SEC received the filing. This was gathered for each of the results of each page using a python package *Beautifulsoup4* (Richardson, 2018).

Now, the document name of each form 4 is needed as another piece of information to access information on a form 4 submission. To access the document name another group of pages has to be accessed, one for each form 4 accession number found or over 3.8 million individual links. The link to access the document names page uses the following format: *submission_link = url_1 + cik + '/' + acc_no_noDash + '/' + acc_no_wDash + url*_2. Table 3 shows an example of what these parameters look like and how they go together.

Table 3. Example Link Formation for Scraping Document Names



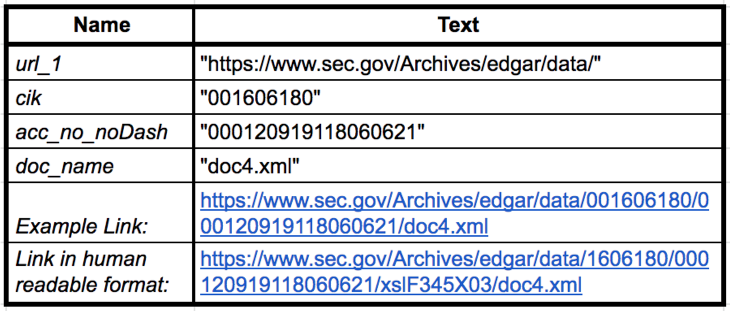
Name	Text
url_1	"https://www.sec.gov/Archives/edgar/data/"
cik	"001606180"
acc_no_wDash	"0001209191-18-060621"
acc_no_noDash	"000120919118060621"
url_2	"-index.htm"
Example Link:	https://www.sec.gov/Archives/edgar/data/1606180/000120919118060621/0001209191-18-060621-index.htm

The example link from Table 3 is provided in a clickable format:

- https://www.sec.gov/Archives/edgar/data/1606180/000120919118060621/0001209191-18-060621-index.htm

This is the link for the first form 4 filing from the last results link showing examples for the company with ticker AAC. The specific document is the first result on that page and matches the given accession number. As mentioned, what is desired on this page is the document name; again *Beautifulsoup4* was used to scrape this information. Now the actual form 4 information can be collected since all accession numbers and corresponding document names have been collected for each CIK. The document name is important because the submitter of the form chooses the name of the file. This name is included in the link that contains the information on the form 4 so the form cannot be accessed without the name. The formula to access a specific form has the following format: *formula form_url* = *url_1* + *cik* + '/' + *acc_no_noDash* + '/' + *doc_name*. Table 4 shows an example for each parameter.

Table 4. Example Link Formation for Scraping Document Information



Name	Text
url_1	"https://www.sec.gov/Archives/edgar/data/"
cik	"001606180"
acc_no_noDash	"000120919118060621"
doc_name	"doc4.xml"
Example Link:	https://www.sec.gov/Archives/edgar/data/001606180/000120919118060621/doc4.xml
Link in human readable format:	https://www.sec.gov/Archives/edgar/data/1606180/000120919118060621/xslF345X03/doc4.xml

The link in Table 4 goes to the following form:

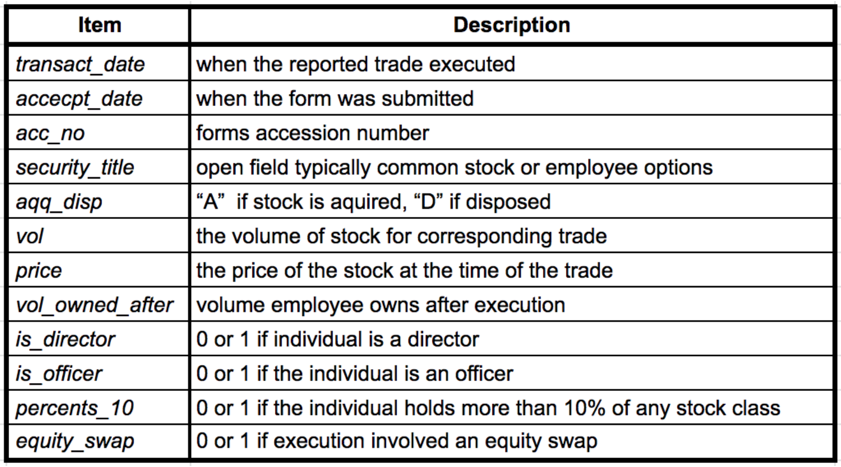
- https://www.sec.gov/Archives/edgar/data/001606180/000120919118060621/doc4.xml

However, this is in eXtensible Business Reporting Language or XBRL. The SEC also provides the information in a more human friendly format, this is shown at the following page:

- https://www.sec.gov/Archives/edgar/data/1606180/000120919118060621/xslF345X03/doc4.xml

The example links above are for the same document that has been traced through Table 1 and Table 2. Information was scrapped from 3.8 million other pages just like this example. The 12 fields collected from each submission are shown in Table 5 along with a description.

Table 5. Information Collected for Each Insider Transaction



Item	Description
transact_date	when the reported trade executed
accecpt_date	when the form was submitted
acc_no	forms accession number
security_title	open field typically common stock or employee options
aqq_disp	"A" if stock is aquired, "D" if disposed
vol	the volume of stock for corresponding trade
price	the price of the stock at the time of the trade
vol_owned_after	volume employee owns after execution
is_director	0 or 1 if individual is a director
is_officer	0 or 1 if the individual is an officer
percents_10	0 or 1 if the individual holds more than 10% of any stock class
equity_swap	0 or 1 if execution involved an equity swap

Most of the fields on the form are straight forward however a closer inspection of the data gathered was required to determine how it would be best processed for analysis. This revealed some important characteristics for the price and security title fields which are noted:

(1) Price: The first field that requires some special treatment is the price reported on a form 4. This price does not necessarily indicate the market value of a stock at that time in any way. This is for two reasons. First, some of these trades are recorded for stock options which the employee is granted for working at a company for a period of time. Second, stock splits are a common issue when doing analysis of equities. A stock split is when a company increases the number of outstanding shares by some multiple in hopes of making their shares more accessible to anyone. When a stock does a 2 for 1 split at a price of $100 dollars anyone who holds a share will receive an additional share. Both will be priced at $50 instead of $100. No value is lost however not accounting for these events could throw off results dramatically. Clearly prices reported at the time of a purchase will not be adjusted for future splits. In summary, options and stock splits are two reasons why the form 4 price will not be used much for analysis. Instead the transaction date will be paired up with the price histories retrieved earlier as this data is adjusted

for splits. A negative side effect of using these price histories is that it becomes impossible to do any analysis that compares the price an employee transacted at compared to the market price at the time.

(2) Security_title: This is one of the only open-ended fields on the form. However, it is important to approach correctly as the *security_title* determines which trades to include in the analysis. Of the 3,827,163 transactions recorded only 3,021,444 were kept; the majority of these are discarded due to the field not fitting the right criteria. An example of this would be phantom stock. As discussed earlier this is a relatively new concept large companies have started using as an employee incentive. This equity emulates the price of a company's stock. Employees are granted phantom shares in the same way they are typical options. The difference is that an employee may choose to "cash out" on these shares at any time past an initial expiration date which is set by the company. In fact, because phantom stocks are not publicly traded their insider trading is only monitored and not regulated. This benefits the holder such that an insider can sell them back to the company if the insider knows negative and privileged information that may be going public soon. In this case the company is forced to repurchase the shares at the current high price as long as the phantom contracts are past their expiration date. Due to the nature of phantom stocks they are likely a better price predictor than normal insider transactions, however their correlation is likely very different from the more controlled common stock as they can be traded based on privileged information. Analysis of these equities would likely yield interesting results but they have only recently gained popularity so there is not much data on them. In total trades were reported with 16,028 different *security_title(s)*, however this list is case sensitive. Many individuals who submitted their forms would provide entries such as "AAPL common

stock" or "Aapl Common Stock" which are counted separately. For this reason, all security titles were converted to lower case and only trades with the exact words "common stock" were included. A random sample of trades selected using this filter was spot checked and the spot-checked sample yielded only appropriate transactions to include for analysis.

The last issue taken care of in this section was to remove any companies without any trading history. This leaves 4,469 companies for analysis. Note that the code for this portion of the project can be found as Appendix Item B.

3.4: Quarterly Financial Statement Scraping

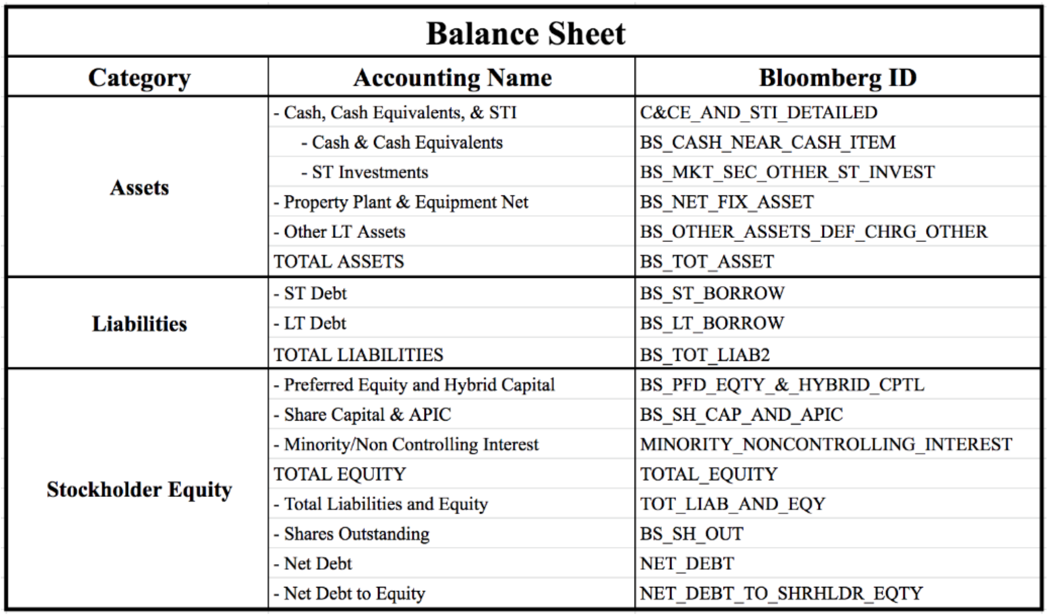
Gathering quarterly financial state data was done using an R package called *Rblpapi* (Armstrong & Eddelbuettel, 2018). *Rblpapi* is a package that interfaces with a Bloomberg Terminal and allows for data queries to be pulled based on a Bloomberg ID of an item. Some fields appeared undesirable from a spot judgment on the Bloomberg Terminal and were not included. An example of this would be number of employees as this was consistently empty. Other dropped fields had non-numeric values. Tables that contain all fields initially investigated are presented as Appendix Items C, D, & E.

A script was made that loops through all companies to retrieve common portions of their quarterly balance sheet, income statement, and cash flow statement. This was done for each company starting at the current date then going back in time until no data is available. This gave the earliest date that data is available for a company on Bloomberg.

With the earliest available date, a much more efficient query can be done which calls all fields of the three financial statements for each company from its earliest date to today. The quarterly history of each company is then saved to its own CSV file named according to its ticker. A total of 120 data fields were scraped.

The next step is to check the quality of data for all the fields collected. This was done by counting the not-a-number (NaN) values for each field. NaNs were tallied up and any field missing more than 10% on average was removed from every company's quarterly financial history. This left 38 fields for the financial statements. Tables 6, 7, & 8 show the 38 items to be included in the analysis based on the accounting statement the field is from.

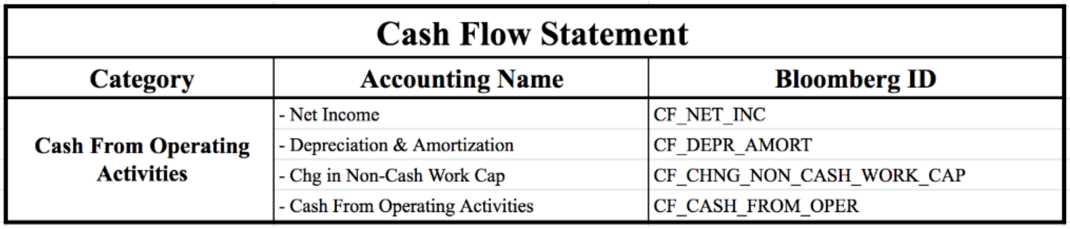
Table 6. Balance Sheet Items Included in Analysis



Balance Sheet		
Category	**Accounting Name**	**Bloomberg ID**
Assets	- Cash, Cash Equivalents, & STI	C&CE_AND_STI_DETAILED
	- Cash & Cash Equivalents	BS_CASH_NEAR_CASH_ITEM
	- ST Investments	BS_MKT_SEC_OTHER_ST_INVEST
	- Property Plant & Equipment Net	BS_NET_FIX_ASSET
	- Other LT Assets	BS_OTHER_ASSETS_DEF_CHRG_OTHER
	TOTAL ASSETS	BS_TOT_ASSET
Liabilities	- ST Debt	BS_ST_BORROW
	- LT Debt	BS_LT_BORROW
	TOTAL LIABILITIES	BS_TOT_LIAB2
Stockholder Equity	- Preferred Equity and Hybrid Capital	BS_PFD_EQTY_&_HYBRID_CPTL
	- Share Capital & APIC	BS_SH_CAP_AND_APIC
	- Minority/Non Controlling Interest	MINORITY_NONCONTROLLING_INTEREST
	TOTAL EQUITY	TOTAL_EQUITY
	- Total Liabilities and Equity	TOT_LIAB_AND_EQY
	- Shares Outstanding	BS_SH_OUT
	- Net Debt	NET_DEBT
	- Net Debt to Equity	NET_DEBT_TO_SHRHLDR_EQTY

Originally there were 49 items from the balance sheet, after filtering out fields that were consistently empty there are only 17 remaining fields.

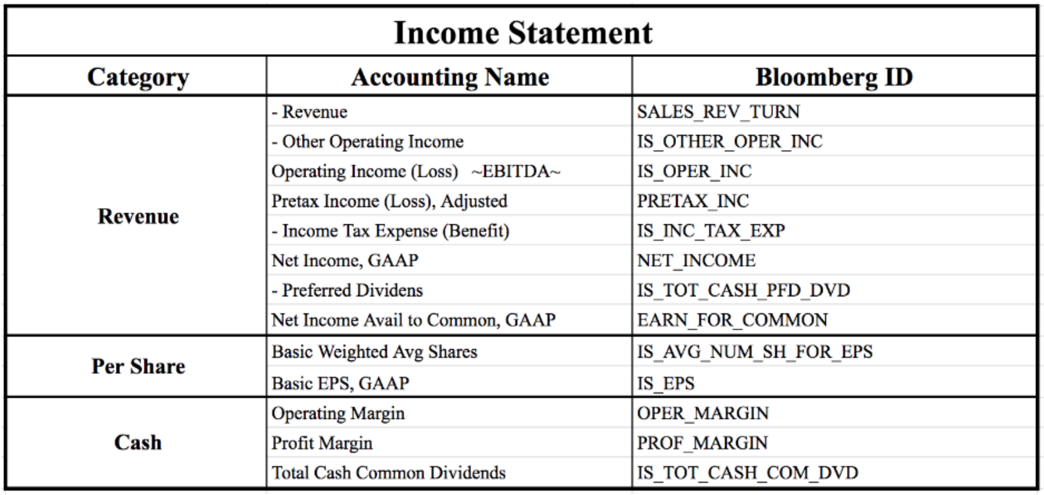
Table 7. Cash Flow Statement Items Included in Analysis



Cash Flow Statement		
Category	**Accounting Name**	**Bloomberg ID**
Cash From Operating Activities	- Net Income	CF_NET_INC
	- Depreciation & Amortization	CF_DEPR_AMORT
	- Chg in Non-Cash Work Cap	CF_CHNG_NON_CASH_WORK_CAP
	- Cash From Operating Activities	CF_CASH_FROM_OPER

The cash flow statement started with 39 items. After filtering out those with more than 10% missing only 4 remain.

Table 8. Income Statement Items Included in Analysis



Income Statement		
Category	**Accounting Name**	**Bloomberg ID**
Revenue	- Revenue	SALES_REV_TURN
	- Other Operating Income	IS_OTHER_OPER_INC
	Operating Income (Loss) ~EBITDA~	IS_OPER_INC
	Pretax Income (Loss), Adjusted	PRETAX_INC
	- Income Tax Expense (Benefit)	IS_INC_TAX_EXP
	Net Income, GAAP	NET_INCOME
	- Preferred Dividens	IS_TOT_CASH_PFD_DVD
	Net Income Avail to Common, GAAP	EARN_FOR_COMMON
Per Share	Basic Weighted Avg Shares	IS_AVG_NUM_SH_FOR_EPS
	Basic EPS, GAAP	IS_EPS
Cash	Operating Margin	OPER_MARGIN
	Profit Margin	PROF_MARGIN
	Total Cash Common Dividends	IS_TOT_CASH_COM_DVD

The income statement had 30 fields to start, after the filter there were 13 remaining. Removing these bad columns helped to bring the number of missing values in the data set down to about 2.15% on average for each column.

There were two main causes contributing to the rest of the missing values. First, for companies that had histories going far back, certain fields had a tendency to not be reported in the 1980's leading to many NaN's at the beginning of that company's set. The second issue was having a whole column NaN. Removing companies that are missing more than 0.5% of their data from the model training data set would bring the company list from 3,877 to 3,523. It was believed that keeping these companies in the analysis and imputing the missing values would provide better results for the model compared to discarding the company’s whole history. Further when a certain field was not reported it seemed a reasonable assumption that this field is not relevant to the company’s financial health otherwise it would have been reported.

Under these assumptions companies that had a whole column empty were given values of 0. This should not be much of an issue as the model that will be built using this data set is focused on the relative change of these values from one quarter to the next. These zeros were given because the model requires all companies have even input dimensions and values for every entry. This approach means the model will see no change in these values just as a human doing fundamental analysis on the company.

The other missing data points that were spread out sporadically through the set were handled. Initially data was imputed by taking the previous value, however analysis showed models yielded better results using a value bisection instead. Bisection was implemented such that if there is a missing value this will be the average of the values before and after. The only exceptions were the first or last values missing. If the first is missing then it was set to the next entry; this was in order to input no change in that field into the model. In a similar manner, if the last value was missing it was set to the one before it. Again, bisection was used for imputation to minimize drastic jumps in change between two quarters’ values. Finally, the code that was used to scrape the quarterly financial statements is provided as Appendix Item F.

3.5: Combining Financial Statements with Form 4 and Price History

The final process was data wrangling to combine all of the information gathered in a nice format for the model. This was done by combining the quarterly financial statements, form 4 histories, and pricing information into one CSV which the model could be trained on.

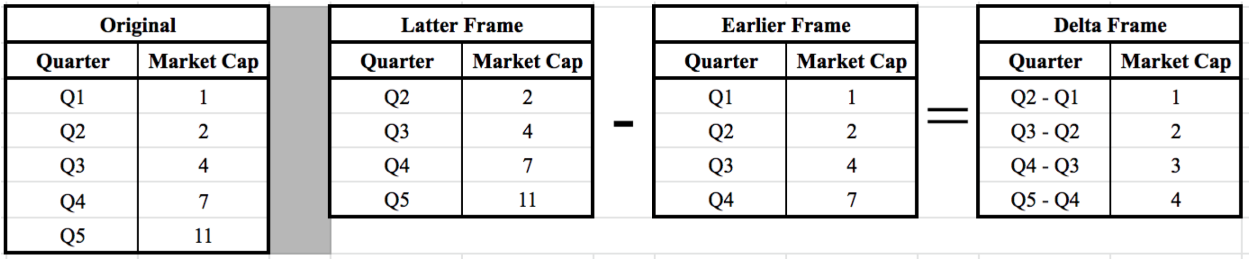
Using Python, a loop was run which executed the following process for each company. First, the separate 3 CSV histories for the company (1) insider trading, (2) price, and (3) quarterly financial statements were imported into pandas DataFrame objects. The earliest and latest dates were found for the company's three DataFrames so that each company's information all started at the same maximum earliest date. Except for the price history which was allowed to have data beyond that of the insider trading and quarterly statement histories. This is to avoid issues with not being able to pair a quarterly financial statement to information from a form 4 as some trades came from before the accounting statements were available.

Then these three DataFrames are passed to a function which loops through the company's financial statement for each quarter. This function finds the start and end dates of each quarter then uses these dates to return some heuristics from each quarter. The specific information along with a description is shown in Table 9.

Table 9. Heuristics Added to Quarterly Financial Statement History

Name	Description
buy_count	number of purchases during quarter
buy_vol	volume of stock purchased during quarter
buy_dollars	dollars worth of stock purchased during quarter
sell_count	number of sells during quarter
sell_vol	volume of stock sold during quarter
sell_dollars	dollars worth of stock sold during quarter
delta_vol	total change in volume of shares owned
price_open	starting price of the quarter
price_low	lowest price through the whole quarter
price_high	highest price through the whole quarter
price_close	ending price of the quarter.
index_L_of_L	days into quarter the lowest of daily low prices occured
index_H_of_L	days into quarter the highest of daily low prices occured
index_L_of_H	days into quarter the lowest of daily high prices occured
index_H_of_H	days into quarter the highest of daily high prices occured
span_LL_to_HL	days between lowest of lows and highest of lows
span_LL_to_LH	days between lowest of lows and lowest of highs
span_LL_to_HH	days between lowest of lows and highest of highs
span_HL_to_LH	days between highest of lows and lowest of highs
span_HL_to_HH	days between highest of lows and highest of highs

These quarterly heuristics were added to the quarterly statement histories in additional columns. From here changes in these values from quarter to quarter are calculated. This is done by taking two copies of the DataFrames, one with the first row not included and another with the last row missing. Then the difference between the two was found giving a DataFrame that represents the discrete derivative for all values between quarters. To better illustrate this methodology Figure 2 shows how this would be done for a simplified DataFrame which contains only the market cap.



Original	
Quarter	Market Cap
Q1	1
Q2	2
Q3	4
Q4	7
Q5	11

Latter Frame	
Quarter	Market Cap
Q2	2
Q3	4
Q4	7
Q5	11

-

Earlier Frame	
Quarter	Market Cap
Q1	1
Q2	2
Q3	4
Q4	7

=

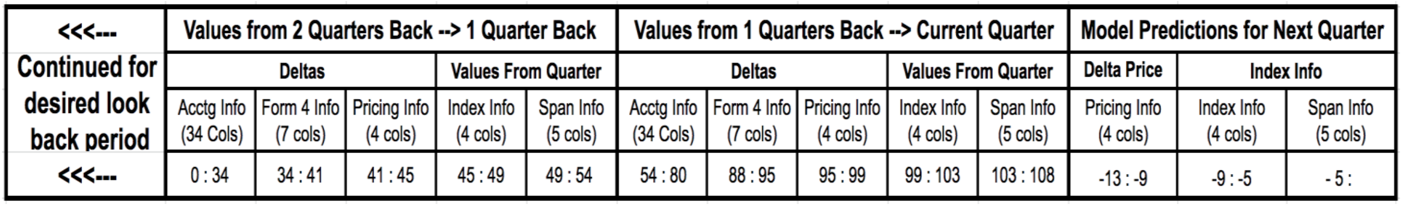
Delta Frame	
Quarter	Market Cap
Q2 - Q1	1
Q3 - Q2	2
Q4 - Q3	3
Q5 - Q4	4

Figure 2. Methodology for Calculating Value of Quarterly Changes

This delta DataFrame is combined with some information of which the actual quarter values are desired instead off the discrete derivative over that quarter. While most inputs to the model for predicting price movements will describe how their values have changed over time, some features are more valuable if their actual values are fed to the model instead of the change from last quarter.

Another way of looking at this is to consider the goal of the model. One prediction will be the day low and high prices occur along with the time span between the two. It seems intuitive that this behavior will relate to how markets react to changes in the financial performance of the stock not when the low occurred last quarter. For this reason the model should be trained with the actual value instead of the change from last quarter. From a philosophical point of view this is similar to saying companies going through a certain pattern of financial states will exhibit the same price formation patterns. For example, if a company that has been doing well puts out a terrible quarterly release it is likely that markets will overreact putting the quarterly low towards the beginning of the quarter. If the model recognizes this decline in performance one would hope it will know to put the low towards the beginning of the period regardless of when the last low occurred.

Finally, this DataFrame which represents the quarterly changes in values must be reshaped to allow the model to consider multiple quarters going back in time. To understand how this was done Figure 3 is provided to depict the structure of the path dependent dataset that will be input into the model.



<<<---	Values from 2 Quarters Back --> 1 Quarter Back					Values from 1 Quarters Back --> Current Quarter					Model Predictions for Next Quarter		
Continued for desired look back period	Deltas			Values From Quarter		Deltas			Values From Quarter		Delta Price	Index Info	
	Acctg Info (34 Cols)	Form 4 Info (7 cols)	Pricing Info (4 cols)	Index Info (4 cols)	Span Info (5 cols)	Acctg Info (34 Cols)	Form 4 Info (7 cols)	Pricing Info (4 cols)	Index Info (4 cols)	Span Info (5 cols)	Pricing Info (4 cols)	Index Info (4 cols)	Span Info (5 cols)
<<<---	0 : 34	34 : 41	41 : 45	45 : 49	49 : 54	54 : 80	88 : 95	95 : 99	99 : 103	103 : 108	-13 : -9	-9 : -5	- 5 :

Figure 3. Path Dependent DataFrame Structure

The columns under the delta categories are the pieces of information which the discrete derivative is entered. This is what was calculated using the delta method show in Figure 2. The "Values from Quarter" columns are features with time series-based information that has the exact value from the prior quarter (not the discrete derivative).

This illustration specifically pictures a two-period delta DataFrame. There are three main parts of this dataset: (1) value changes from two quarters back to one quarter back and time series information from that duration, (2) changes from one quarter back to the "current" quarter and time series info, and (3) the target values that the model will predict. Note, in this context the current quarter does not actually mean right now, instead it is relative to the target. One example of this would take the current quarter to be the last quarter in 2010 for a company, the two deltas would be from the two prior quarters and the targets are the values for the first quarter of 2011. These predicted targets are unknown from the model input's perspective.

Next, the path based DataFrame is normalized such that each column has a mean of 0 and a variance of 1. This is done so that the data from each company be complied into a larger DataFrame. Doing this process before compiling the data guarantees that all changes are relative to their company. More explanation of why this was done will be covered next chapter. What is important to understand is that this was the process for one company. Once one company is processed then it may be compiled with other processed companies. This yields the final multiquarter delta data set with mean zero and variance one which is important for machine learning. An explanation for the order of operations on the transformation for the dataset will be provided in the analysis section.

Chapter 4: Analysis

This chapter will go into various forms of exploration and analysis. This will start off by analyzing how prices for stocks tend to move after each individual insider transaction. After this a comparison for price's behavior based on all buys compared to all sells will be compared; this is to determine if there is a categorical difference between the two. Finally a model will be built off of the path back delta set which will predict what the low, high, and closing prices for the next quarter will be along with when the low and high prices will occur.

4.1: A Visual Analysis of Prices Immediately Following Trades

The first method that was tried intended to see any unexpected qualitative relationship between insider trading and price movement with the main focus being a visual analysis to determine if insider trades show potential as a short-term price signal. First, the form 4 and the price history CSVs are called for a company.

The form 4 and price histories are cleaned for each company such that neither one starts before the other. The price history is allowed to go on further than the insider trading history, but not vice versa. This is so price movement can be seen for as long as possible after the last trade on record allowing for more trades to be included in the graphics generated as the price history after the time of a trade is needed. Next, the data is processed for the graphic.

First the transaction type, i.e. buy or sell, was noted. Then the day of a transaction was paired up with the price history and the prices for this day and the 30 days after were put into a list. This list of prices was then normalized such that the price on the day of the trade was 1, then all others in the list were adjusted by the same factor. This linear transformation allows the price movement after all transactions to be viewed evenly. It would be an unfair to look at the price movements of a stock when it's price was $40 on the day of a transaction and then compare these movements to another insider transaction years later when the price was $400. Movements in the latter transaction would clearly be much larger. Finally, a function took this normalized list of prices and added it to a plot. Once all trades for the company were processed the plot was complete. The result of this method for Apple whose ticker is AAPL is shown in Figure 4.

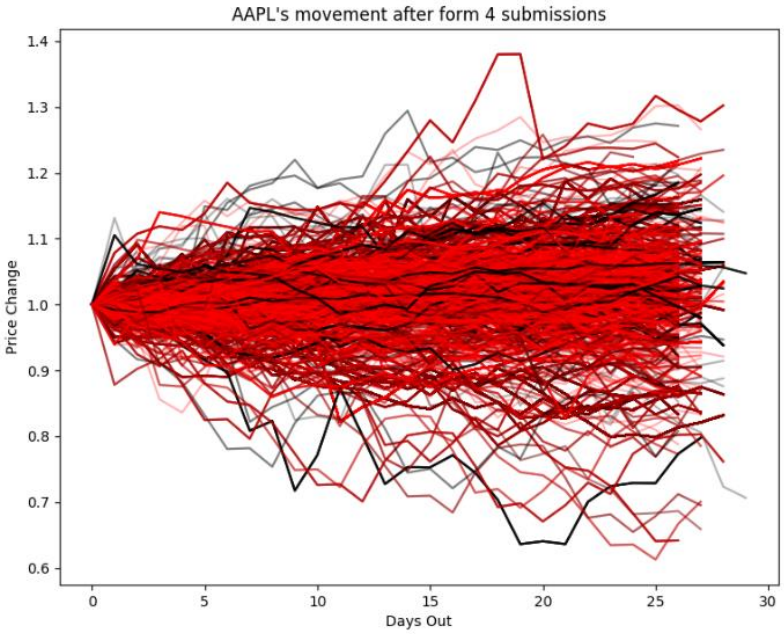


Figure 4. AAPL price movement after each insider purchase

In this graph, black lines represent price movements after a buy and red represents post selling movements. One of these graphs was created for each company.[1]

The results for each company were visually inspected. Generally one could not tell the difference from one graph to another besides the fact that some smaller companies clearly had a significant advantage for buyers. This can be seen by the fact that the black lines are generally

[1] Access to the whole folder of these images will be happily provided upon request. My email is included in the academic vita section at the end of the paper.

climb above the red lines. This indicates that the price of the stock after buys raises more when insiders buy and falls upon insiders selling. This however does not seem like it is worth much investigation as these examples were few and far between making it hard to train a computer to predict this behavior.

Note, there were no firm numbers yielded based on this work. Further, I make no claims in this section on the difference between buying and selling behavior as none can be seen. This is only an exploratory activity that was included for two reasons. First, these graphs help to illustrates the way that prices are normalized to the day of the transaction. This concept will be relied on heavily through the rest of the paper. Second, this process provided inspiration which impacted the approaches taken later on in the paper and were included in the hopes of showing the reader this thought process along with providing potential inspiration for future research.

A final note, this step also calculated other features of the trades such as a simple linear regression for each transaction. These features were then separated based on buying and selling transactions to see if factors such as the expected slope and intercept of the price varied based on different trade heuristics of the insider transaction. These results led to the inspiration for the next section, Chapter 4.2.

4.2: Comparing Returns of Buys and Sells Over Various Time Frames

One of the main goals outlined in this paper is to determine whether the price of a company's stock acts differently based on the buying and selling trends for a company. To gain a better understanding each trade was looked at in the same method in the previous section where the price at time of transaction was set to one. The difference in this section is the only change of interest is what the price change was from the time of purchase to one later point in time. Again, the best way to understand this methodology is to see a visualization. Figure 5 shows the average return of a stock after a buy and sell after various time periods.

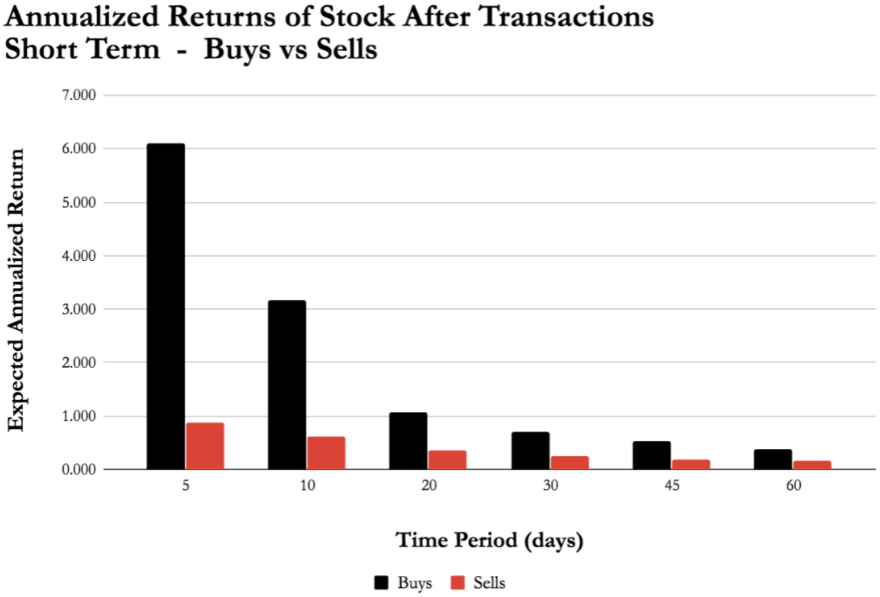


Figure 5. Average Annualized Return of Equities Based on Insider Transactions for Various Short Time Periods

This chart can be viewed as analyzing two different investment strategies. One for insider buys and another for sells. Both take the same approach; every time an insider makes a transaction you buy the stock and then hold it for a given period of time. Note with this you are buying the stock whether the insider buys or sells. Every transaction is weighted equally in this strategy. Further, the average annualized return of these strategies is shown, not the average return from one trade. To see why consider the 30-day holding period as an example. If you buy a stock upon an insider buy and hold it for 30 days the average return of this transaction would be 1.04507 relative to 1, or a 4.5% increase in value. However, this is a one-month holding period, so the annual return from this investment would be $1.045^{12\,=}$ 1.710, or 71% for the year. The annualized transformation is provided so all time periods compared fairly.

Figure 5 only shows the expected returns for shorter periods of time. The shorter periods have significantly higher returns and overshadow the returns of longer periods if graphed together. Due to this they were separated and Figure 6 shows the same comparison for the longer duration strategies.

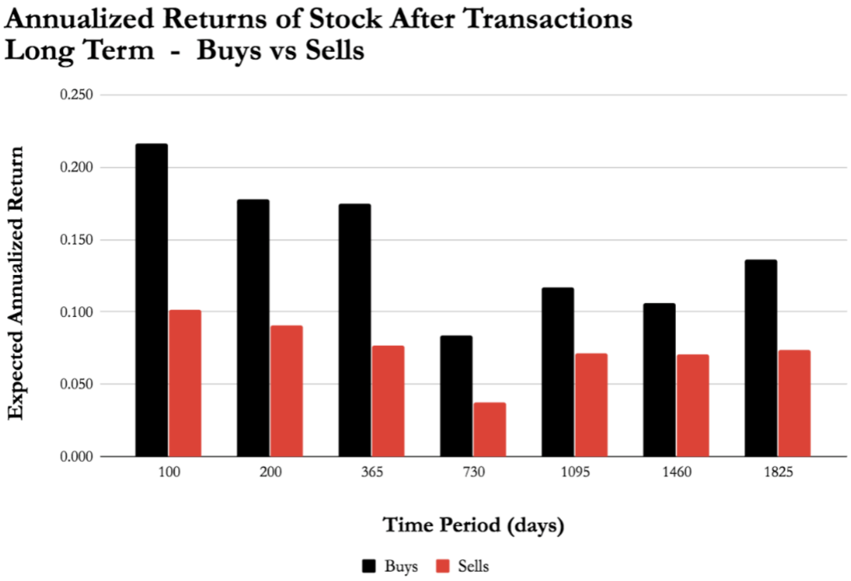
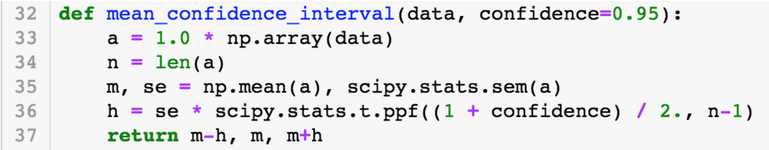


Figure 6. Average Annualized Return of Equities Based on Insider Transactions for Various Long Time Periods

There are over 3 million transactions split between insider buys and sells that contributed to generating the expected returns from these strategies. Admittedly the variability for any given transaction is extremely high, however the large sample size for both do allow for relatively small confidence intervals to be built which contain the expected mean return from each strategy. This confidence interval was calculated by feeding a vector of a returns into a function shown in Figure 7.



```
def mean_confidence_interval(data, confidence=0.95):
    a = 1.0 * np.array(data)
    n = len(a)
    m, se = np.mean(a), scipy.stats.sem(a)
    h = se * scipy.stats.t.ppf((1 + confidence) / 2., n-1)
    return m-h, m, m+h
```

Figure 7. Code to Calculate Confidence Intervals for Expected Returns

The returns are the *data* input of the function and are again relative to a purchase price of 1. The buy-based strategy had 1.4 million transactions included to build the confidence intervals. The sell-based strategy had 1.8 million transactions. This is *n* in the function above. The output of the function yielded lower and upper ends of the confidence interval along with the prediction of the true mean. The annualized returns are calculated based on the true mean estimation. These results are shown in Table 10.

Table 10. Confidence Intervals of Buy & Sell Strategies with Annualized Returns

Time Period (days)	Movement After Sell			Movement After Buy			Mean Annualized Return	
	95% Lower	Mean	95% Upper	95% Lower	Mean	95% Upper	Buys	Sells
1	0.00277	0.00785	0.01292	0.01069	0.02008	0.02948	1417.843	16.330
5	0.00394	0.00868	0.01342	0.01854	0.02722	0.03589	6.101	0.880
10	0.00825	0.01338	0.01852	0.02967	0.03989	0.05011	3.169	0.625
20	0.01314	0.01735	0.02156	0.03359	0.04084	0.04810	1.076	0.369
30	0.01464	0.01852	0.02240	0.03812	0.04507	0.05202	0.710	0.250
45	0.01853	0.02204	0.02555	0.04681	0.05357	0.06034	0.527	0.193
60	0.02221	0.02559	0.02898	0.04958	0.05561	0.06164	0.390	0.166
100	0.02433	0.02678	0.02923	0.05075	0.05514	0.05954	0.216	0.101
200	0.04459	0.04863	0.05267	0.08897	0.09373	0.09850	0.178	0.091
365	0.07355	0.07643	0.07931	0.15751	0.17507	0.19264	0.175	0.076
730	0.07355	0.07643	0.07931	0.15751	0.17507	0.19264	0.084	0.038
1095	0.22650	0.22972	0.23294	0.35359	0.39226	0.43093	0.117	0.071
1460	0.31103	0.31457	0.31811	0.48830	0.49660	0.50490	0.106	0.071
1825	0.42136	0.42662	0.43188	0.59010	0.89495	1.19980	0.136	0.074

This table shows that the only time period analyzed which had any overlap for expected returns is a one day holding period. The upper end of the confidence interval for the selling strategy is slightly greater than the lower limit of the insider buy strategy. Due to this overlap a one day holding period is the only one without a statistically significant difference in expected returns. The clear difference between holding on buys vs. sells indicates that insiders have some edge regarding knowing if their company is undervalued.

Looking at the results some other interesting patterns can be found. First, the width of confidence intervals relative to the mean return shrinks significantly as the time period increases. This is seen by the fact that the shorter time periods have wide intervals, the longer time a period being considered the more certain the true average return estimation tends to be. What is interesting is that this behavior leads to a steady increase between the spread of the confidence intervals for both the separation between their tails and center of the distribution. This pattern continues up until a holding period of 1460 days (4 years) then at 1825 (5 years) the information seems to lose value as the returns take a large step towards each other.

Note, that there are some issues that should be addressed with the analysis in this section, especially when using the metaphor of a real-life trading strategy as was given to start. First off, these strategies do not consider any transaction fees or taxes. Fees would be especially significant for the shorter-term strategies as the actual returns are relatively small; this implies large amounts of capital would be needed to outweigh these fees to get anywhere near theoretical returns. Secondly the annualized returns metric operates under the assumption that all money is constantly invested and is evenly distributed to each stock. Investing a whole pool of money evenly while making investments at different times is impossible as the rates of insider transactions varies so some proportion of capital would need to be reserved incase an unexpectedly high rate of purchasing suddenly occurs. Nevertheless, the most important part of this analysis is that both strategies were compared on a level playing field meaning there is justification stating that equity prices move differently after insiders buy compared to when they sell. As a final note, the code used to create the results for this section can be found as Appendix Item G.

4.3: Predicting Price Movements with the Financial State & Form 4 Data Set

The last part of this chapter aims at incorporating all the data wrangling and findings thus far into a comprehensive model which forecasts price movements. After a base model is built and its accuracy has been measured various features will be removed from the model to see how much accuracy, if any, is lost due to the removal of that information.

Two data sets were created; one with five and the other with eight quarters of information going back in time. As a reminder, the structure of this data set is shown in Figure 3 from Chapter 3.3 that covered how these data sets are created. The reason two different time period data sets are used is because for each desired extra quarter back in time, an additional row is lost for each company. This loss adds up as there are a significant amount of companies. This is due to the fact that looking more time periods back simply requires more data. To see this consider making a delta from one quarter to the next and the associated target; this requires three quarters. Two quarters for the delta and one for the targets. To make a two-path delta, four quarters are needed. The 8-quarter delta had 120,855 total entries and the 5-quarter delta had 130,926 entries. Hopefully these additional entries can provide a slight performance boost.

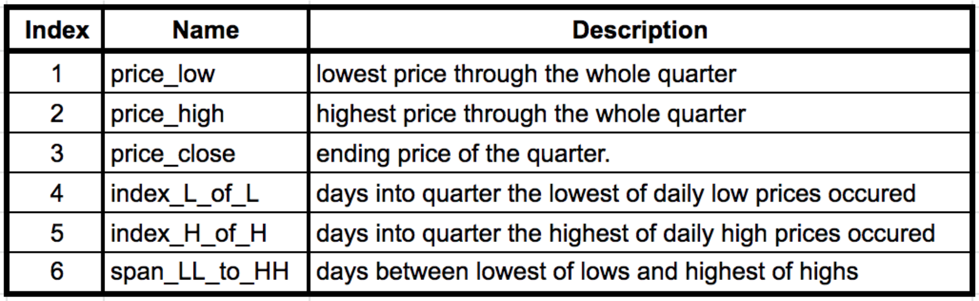
A final note on the values of the data themselves; it is best practice in most machine learning methods to input all data such that it has a mean zero and variance of one (Guido 2016). The fact that this was done was covered earlier. However, the order of operations in this process is very important and before going further a quick explanation on this. When training a model using a data set some features (or columns) will have significantly different univariate qualities. For example, two features in these sets are (1) quarterly earnings and (2) number of insider transactions in a month. Earnings will consistently be a higher number, by scaling each feature to fit a normal distribution it makes changes to the model have equal effects as the scale of the features are similar. However, the data sets could have been processed to have such qualities after the entries for all companies were compiled on top of each other. Instead, each company's history was given this property individually and they were then complied. This is because all changes from quarter to quarter should be viewed with their relative magnitude. That is, if a small company that initially went through an initial public offering sees its earnings increase by a few million in a quarter this shows a good trend for the company and one would expect the stock to react accordingly. Though if you take an Apple or Microsoft and give them the same bump in profit margin it will go unnoticed. It is likely that this sort of earnings increase would lag behind the inflation rate for a company that large and actually reflect poorly on the stock. In conclusion the quarterly changes were scaled relative to the magnitude of their own company such that these two scenarios are differentiated.

Next, some notes on the actual models themselves. There will be two different types of models trained and tested on the quarter delta data set built in Chapter 4.3. The first is a simple linear regression with the following equation in matrix form:

$$(1)\ \hat{Y}_i\ = bX_i + a$$

$\hat{Y}_i$ represents the target vector of dimension *(6, 1)* being predicted by the model. The targets, a description, and their indices are shown in Table 11.

Table 11. Target Variables Predicted by Model



Index	Name	Description
1	price_low	lowest price through the whole quarter
2	price_high	highest price through the whole quarter
3	price_close	ending price of the quarter.
4	index_L_of_L	days into quarter the lowest of daily low prices occured
5	index_H_of_H	days into quarter the highest of daily high prices occured
6	span_LL_to_HH	days between lowest of lows and highest of highs

In the linear model, X_i is the input or feature vector for the model; this has dimension *(n, 1)*. The value of n is then number of features being used to train the set. Due to the nature of this experiment X_i does not have a fixed size as certain tests fundamentally change X_i. For example, when the 8 path back delta DataFrame is used X_i will have 432 features where the 5 path back will have 270 features input as there are 54 from each quarter. Further, this dimension n will vary based on what information is removed when testing the value of buy vs. selling information as the removed features are not fed into the model. This naturally has a down stream effects on the size of b, the weight matrix of dimensions which has dimensions *(6, n)* which is multiplied by the quarter change values from $X_{i.}$ Next is the adjustment constant, a, which is simply a constant that shifts the predictions vertically; as there are 6 predictions being made the adjustment constant also has dimensions *(6, 1)* to match $\hat{Y}_i$. Finally the subscript, i in $\hat{Y}_i$ & X_i indicates the row index of the data set.

The way both the models in this paper are trained is similar. Both aim to minimize a loss function which is proportional to error. In the case of the linear model the loss function is

$$(1)\quad \min\left(\sum_i |\hat{Y}_i - Y_i|^2\right)$$

This function minimizes the sum of the squares of the estimates' errors by adjusting the weight matrix b and the verticle shift constant a. This is done in the linear regression model by calculating the gradient of the loss function with respect to the weight matrix b and moving in the direction that indicates will lower loss the most. Figure 8 provides a good illustration of a gradient descent algorithm in two dimensions (Hutchinson, 2016).

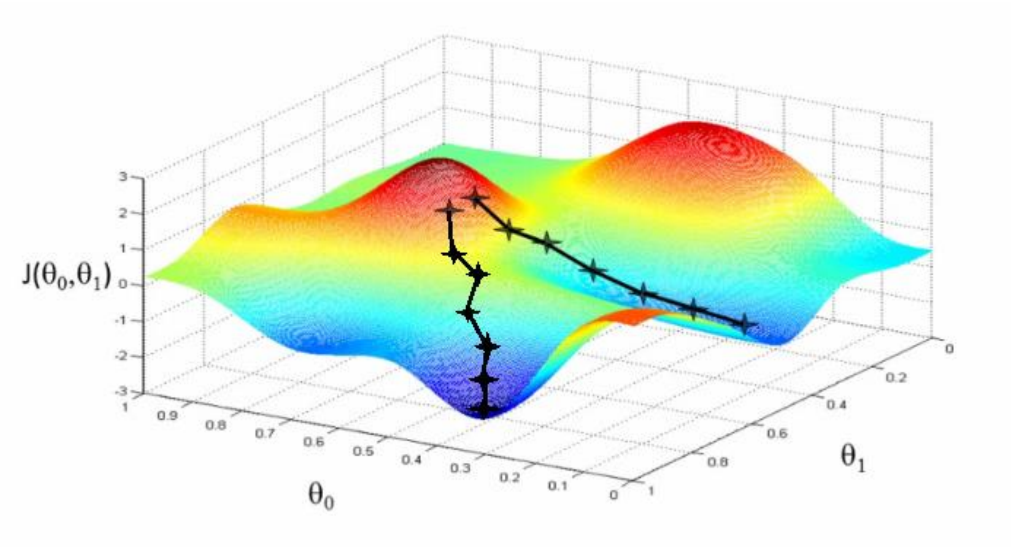


Figure 8. Illustration of Gradient Descent

Figure 8 shows a gradient descent with a 2 dimensional weight matrix. This simplification is used because gradient descent cannot be easily depicted in higher dimensions. In this example θ_1 & θ_2 are the two weights in the matrix and $J(\theta_1,\ \theta_2)$ represents the loss based on the values θ_1 & θ_2 take. Each of the dots represents a different iteration of the model during training. The two dots at the top in the red area represent the weight matrix being randomly initiated in different places. When initialized randomly the value of the model's loss is high. After each iteration of finding the gradient, the algorithm changes the weights of the matrix to go in the direction that will lower loss the most. The product of the gradient and a model parameter called the learning rate determines the magnitude weights are changed by; in this paper the most effective learning rate was found to be 0.0001. It is important to note that initializing the weights differently can lead to different termination points for the algorithm; this is shown by the two paths shown in Figure 8.

The MLP network has a similar approach to the linear regression method, in that the same sort of loss function based learning is done. However, the MLP network doesn't have one weight matrix. Instead there are many, one for each node in every layer of the network. This leads to gradients becoming much more difficult to compute. The method used to adjust all the weights in a neural network is called back propagation. Again this considers the predictions output by the network and considers how all the weights contributed to this value going back to the beginning inputs. Each constant's contribution to the output is noted and if the output was too high or low constants are adjusted accordingly. The structure of the MLP network is shown in Figure 9.

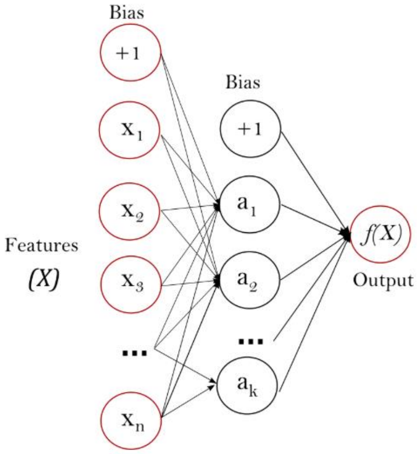


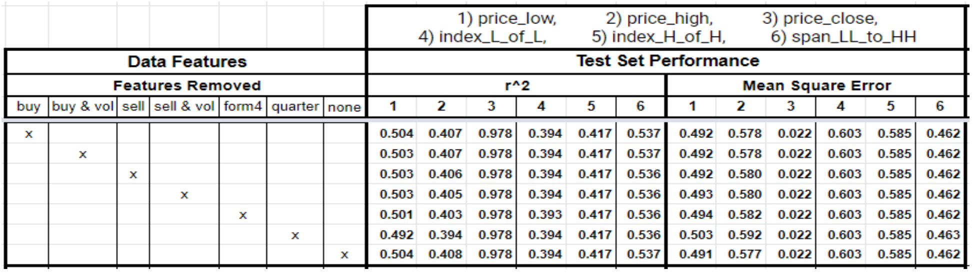
Figure 9. Multi-Layer Perceptron Neural Network Structure

The MLP network also has the same input and output features meaning the main difference from the regression is the degree of complexity in adjusting weights. The network shown in Figure 9 has one layer with k nodes. This is represented by the a column of the nodes. Every a_i contains the same dimension weight matric as b in the linear regression and is trained in an extremely similar method; the slight nuisances are outside the scope of this paper.

The last difference for the MLP network is the existence of an activation function for each node. The activation function decides whether a node will contribute to a prediction. This is key as any combination of the a_i nodes may be active depending on what the features of the input are. This gives the MLP models nonlinearities in the prediction space that allows it to treat two inputs very differently, a feature regressions lack as the output is always some combination of linear transformations of the input. This paper uses the ReLu function which is defined as $\max(0, f(a_i))$. If the value of the function does not surpass 0 the value of that node is not passed on in the training. This is the mean reason why back propagation for the neural network is more complicated than regressions. If more information on the specific network is of interest see the Sci-kit Learn documentation (Pedregosa, 2011).

Now, the results for the models; the first thing to consider is whether the insider trading information's presence in the model adds accuracy. To see this results from the eight-quarter delta data set from the linear model are shown in Table 12.

Table 12. Results of the 8 Path Back Quarter Delta Data Set for the Linear Regression Model



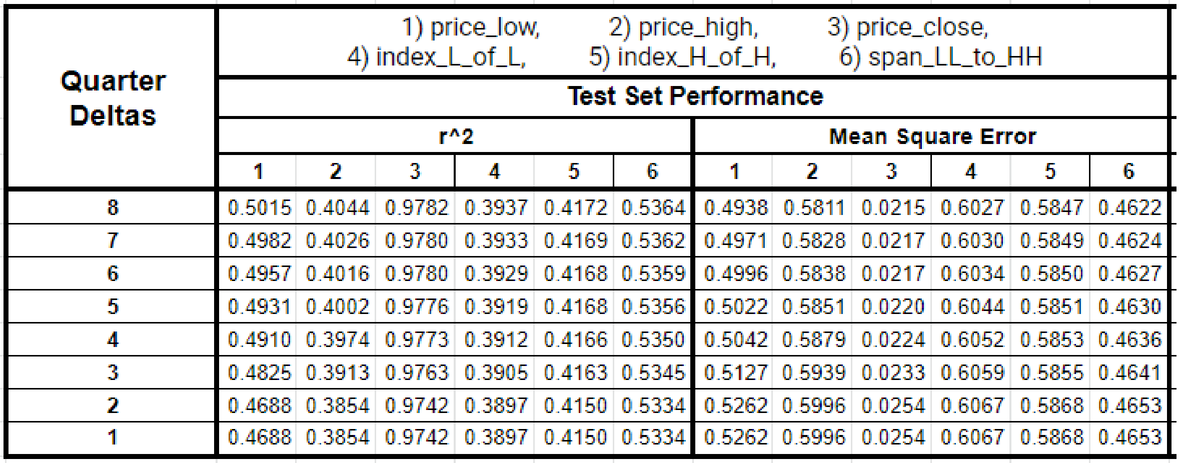
							1) price_low, 4) index_L_of_L,				2) price_high, 5) index_H_of_H,			3) price_close, 6) span_LL_to_HH				
Data Features							Test Set Performance											
Features Removed							r^2						Mean Square Error					
buy	buy & vol	sell	sell & vol	form4	quarter	none	1	2	3	4	5	6	1	2	3	4	5	6
x							0.504	0.407	0.978	0.394	0.417	0.537	0.492	0.578	0.022	0.603	0.585	0.462
	x						0.503	0.407	0.978	0.394	0.417	0.537	0.492	0.578	0.022	0.603	0.585	0.462
		x					0.503	0.406	0.978	0.394	0.417	0.536	0.492	0.580	0.022	0.603	0.585	0.462
			x				0.503	0.405	0.978	0.394	0.417	0.536	0.493	0.580	0.022	0.603	0.585	0.462
				x			0.501	0.403	0.978	0.393	0.417	0.536	0.494	0.582	0.022	0.603	0.585	0.462
					x		0.492	0.394	0.978	0.394	0.417	0.536	0.503	0.592	0.022	0.603	0.585	0.463
						x	0.504	0.408	0.978	0.394	0.417	0.537	0.491	0.577	0.022	0.603	0.585	0.462

This table shows the r^2 value and mean square error for each of the 6 targets on a linear regression. The results for each row are based on which data is removed from the input into the model indicated by an “x” in the data features columns. The results are all very close, the only differences seen from removing various pieces of information are for the low and high prices for the quarters. It is clear the regression is more accurate when all types of insider trading information are present. Following this trend, removing either the buy or the sell information is still more accurate than removing the form 4 altogether. Notably there is no practical difference which of the buying or selling information is not considered. Finally removing the accounting information leads to the most significant decline in accuracy, though it makes surprisingly little difference as the range of r^2 values is from 0.492 to 0.504. The range of the mean squared errors is just as small as the r^2 range.

Note, Table 12 only shows the results for the linear regression models trained with the 8 path delta data set. This same information for models trained with 1 thru 8 path data sets can be seen in Appendix Item H for the linear regression. The results in the appendix confirm that the aforementioned trends hold; that is models that include no insider information do worse than those with some. The models with all the information consistently do best. However, when looking at a difference between the quality of information from insider buys and sells the advantage actually switches back and forth. This trend is also true for the neural network models whose results can be seen in Appendix Item I, though the variation is much greater depending on network structure.

Regarding path dependence, the linear model has very telling results regarding. Table 13 shows the average results of the model when trained with different look pack periods. It should be noted that these results are the average of all models trained for each time period back with various parameters of the network missing such as buys and sells. While this may misrepresent the potential prediction accuracy, all metrics for the various look back periods are generated using the same method so the insights regarding path dependence are still credible.

Table 13. Prediction Metrics for Linear Regression Models Based on Look Back Period

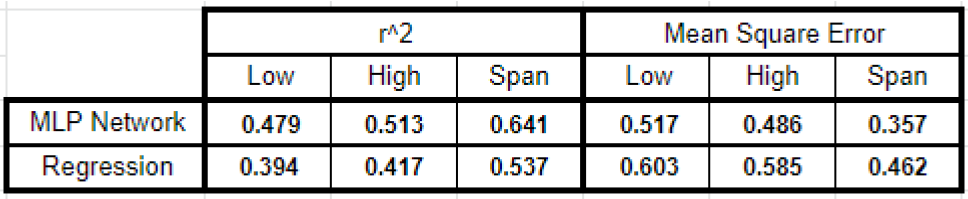


Quarter Deltas	1) price_low, 2) price_high, 3) price_close, 4) index_L_of_L, 5) index_H_of_H, 6) span_LL_to_HH											
	Test Set Performance											
	r^2						Mean Square Error					
	1	2	3	4	5	6	1	2	3	4	5	6
8	0.5015	0.4044	0.9782	0.3937	0.4172	0.5364	0.4938	0.5811	0.0215	0.6027	0.5847	0.4622
7	0.4982	0.4026	0.9780	0.3933	0.4169	0.5362	0.4971	0.5828	0.0217	0.6030	0.5849	0.4624
6	0.4957	0.4016	0.9780	0.3929	0.4168	0.5359	0.4996	0.5838	0.0217	0.6034	0.5850	0.4627
5	0.4931	0.4002	0.9776	0.3919	0.4168	0.5356	0.5022	0.5851	0.0220	0.6044	0.5851	0.4630
4	0.4910	0.3974	0.9773	0.3912	0.4166	0.5350	0.5042	0.5879	0.0224	0.6052	0.5853	0.4636
3	0.4825	0.3913	0.9763	0.3905	0.4163	0.5345	0.5127	0.5939	0.0233	0.6059	0.5855	0.4641
2	0.4688	0.3854	0.9742	0.3897	0.4150	0.5334	0.5262	0.5996	0.0254	0.6067	0.5868	0.4653
1	0.4688	0.3854	0.9742	0.3897	0.4150	0.5334	0.5262	0.5996	0.0254	0.6067	0.5868	0.4653

The main take away from Table 13 is that more time periods included leads to more accurate predictions regardless of what target or metric is chosen. Further, it is clear that these models are not terribly over trained. This is because the performance on the training set would be significantly better than that of the test set; checking the appendix (Item H) will reveal that the train and test set accuracy is usually within 1% of each other.

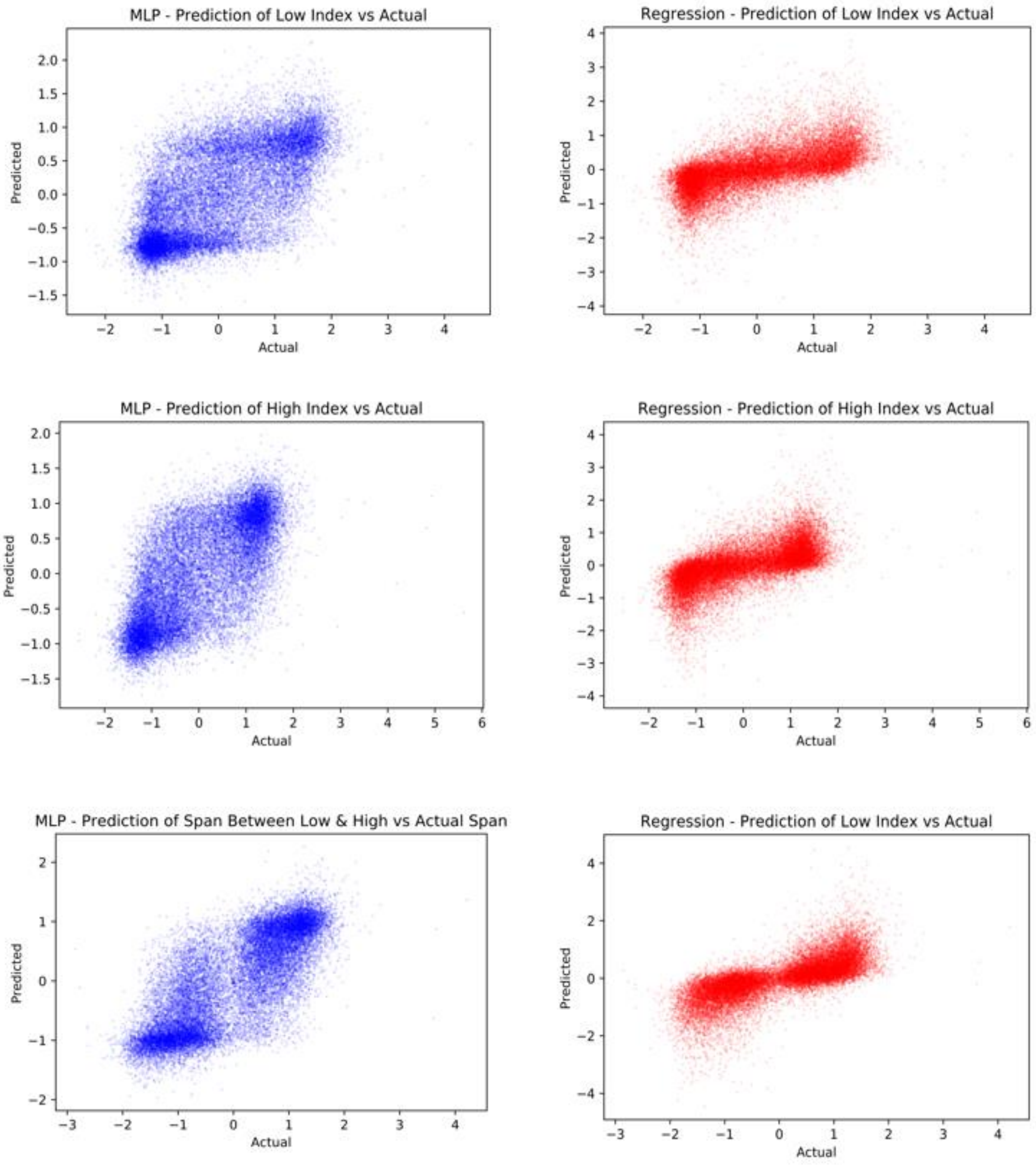
The next notable result is the accuracy of price predictions from the MLP neural networks are actually worse than the linear regression predictions. This topic is tricky as the results from the MLP vary significantly on layer structure; only the most optimized network structures came close to regression's performance. The more interesting result is that the neural networks have consistently and significantly better results for the time series predictions. The target variables of focus will be how far into the quarter the lowest & highest prices of the quarter occur along with the time span between the two. The metrics for the top performing MLP and Regression models regarding time series predictions are shown in Table 14.

Table 14. MLP vs. Regression Performance Metrics for Time Based Predictions



	r^2			Mean Square Error		
	Low	High	Span	Low	High	Span
MLP Network	0.479	0.513	0.641	0.517	0.486	0.357
Regression	0.394	0.417	0.537	0.603	0.585	0.462

This surprising difference warrants further investigation so the predicted values were plotted against the actual values for the high and low indexes along with the time span between the low and high price. This is shown in Figure 10.


MLP - Prediction of Low Index vs Actual

Regression - Prediction of Low Index vs Actual

MLP - Prediction of High Index vs Actual

Regression - Prediction of High Index vs Actual

MLP - Prediction of Span Between Low & High vs Actual Span

Regression - Prediction of Low Index vs Actual

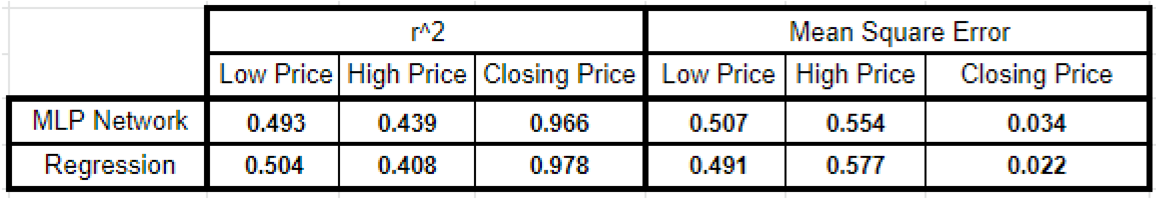
Figure 10. Plots of Predicted vs. Actual Time Based Targets

These graphs all plot the actual value for a quarter on the x-axis and the corresponding predicted value on the y-axis. The MLP Model results are shown in blue and the linear regression is red. An initial inspection seems to indicate the regression fits the data better as the regression graphs look like a straight line through the data would minimize the sum of the distances to the line. As a reminder this is essentially what r^2 represents, though r^2 uses the distances squared.

There are two reasons the MLP results may look worse. First, the axis scale for MLP predictions is different as they fit into a smaller range. Second, the dots on the graph are highly transparent meaning the solid clusters have a very dense collection of points; notably these clusters are the areas predictions are nearest actual values. A closer look may reveal many dots scattered around the edges that went unnoticed due to the combination of their sparsity and transparency.

The qualitative difference in performance between the models now brings the price predictions back into question. Initially, the small difference in price predictions did not seem worth further investigation as the performance of the regression model compared to the MLP was pretty even. After seeing the differences for the time-based targets, it seems like a good idea to investigate the qualitative differences between the price predictions in the same manner. The relevant performance metrics for the price targets for these models are shown in Table 15.

Table 15. Network vs. MLP Performance Metrics for Price Predictions



	r^2			Mean Square Error		
	Low Price	High Price	Closing Price	Low Price	High Price	Closing Price
MLP Network	0.493	0.439	0.966	0.507	0.554	0.034
Regression	0.504	0.408	0.978	0.491	0.577	0.022

Each of the models shown are the top performers for their category. The highest performer was selected by taking the best overall r^2 value for each of the models created. The regression results are from the 8 path back data set with none of the features removed. The MLP network shown is based on the 5 path back dataset, ironically with only the selling information included.

The same actual vs. predicted target value plots were made for prices, these are shown for the quarterly low and high prices in Figure 11.

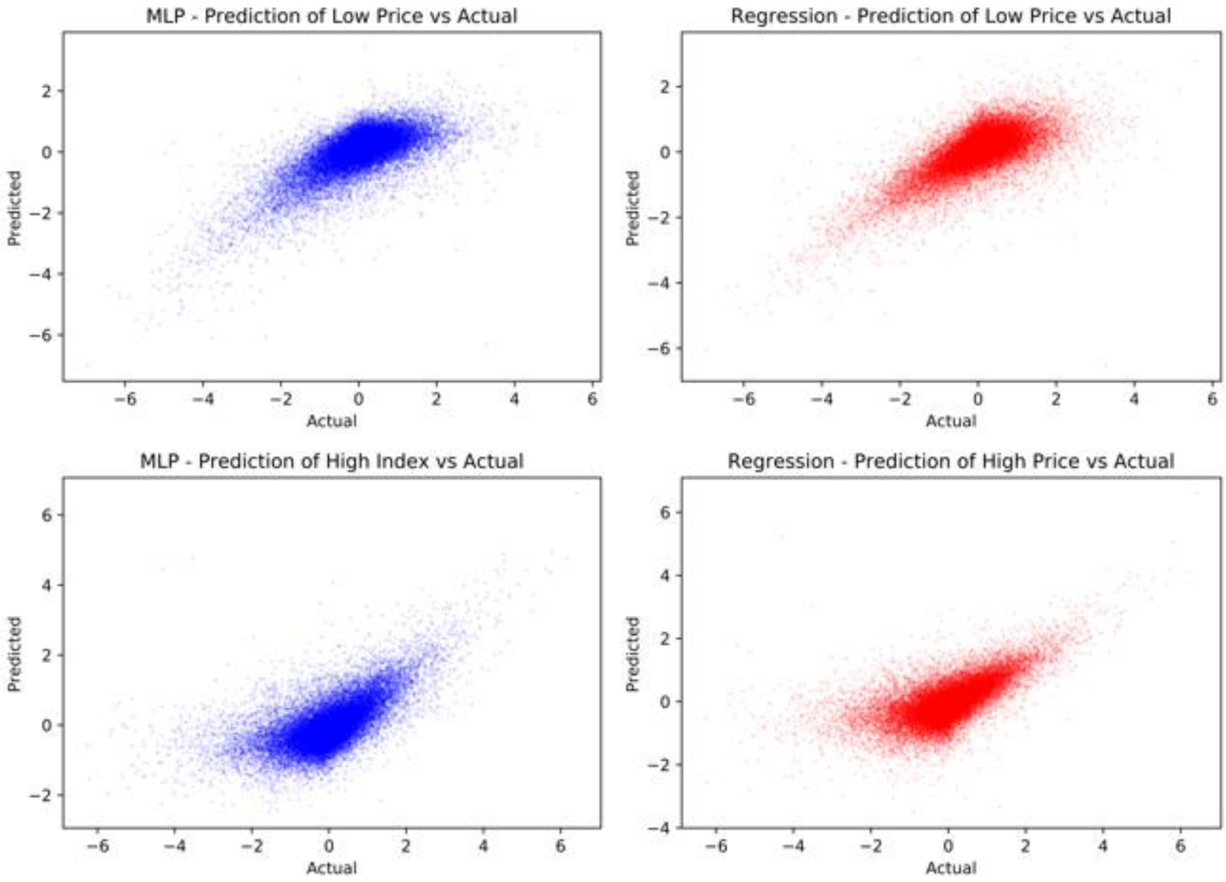
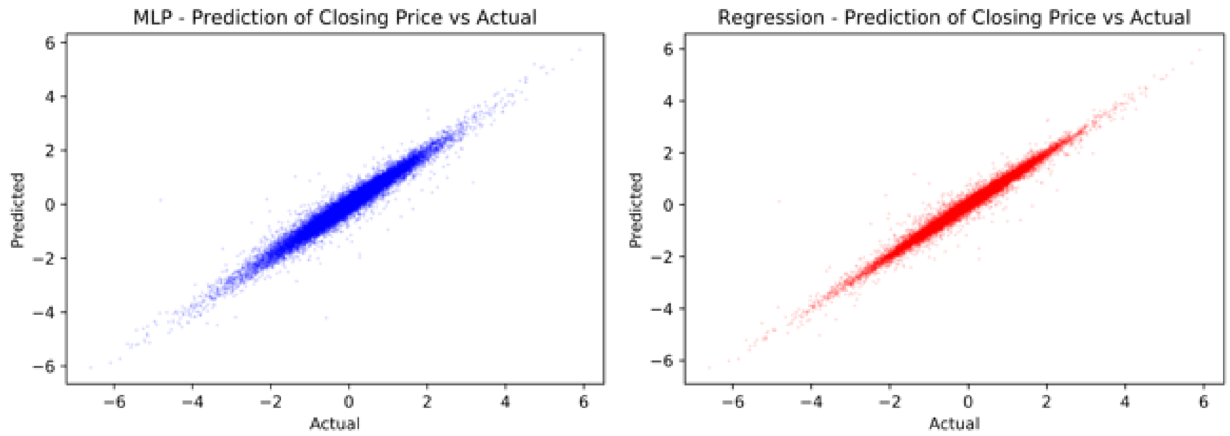


Figure 11. Plots of Predicted High & Low Price Changes vs. Actual

Unlike the time-based targets, the behavior between the two models is not different for price predictions. Though there are some other interesting observations. Not only do both models share the same characteristics but the low and high price show a mirrored geometry in their graphs. The predicted low prices generally lie above the line $y = \frac{2}{3}x - 2$ and the predicted high prices lie below the line $y = \frac{2}{3}x + 1$. Mirrored about these lines is a clear triangle feature for both metrics, a feature what would have high utility building an investment strategy off of these predictions. As the behavior between these two models is similar for high and low price predictions there is noting further to investigate.

The closing prices are shown next in Figure 12, these were separated from the low and high prediction graphs due to the fact that the lows and highs exhibited such similar behavior while the closes basically fit a straight line as is shown in Figure 12.


MLP - Prediction of Closing Price vs Actual

Regression - Prediction of Closing Price vs Actual

Figure 12. Plots of Predicted Closing Price Changes vs. Actual

Initially both the models' accuracy for predicting how the closing prices will change from quarter to quarter appears quite remarkable, especially when looking at these graphs. However, it is important to remember that all the actual values are transformed. This transformation thins down the distribution dramatically raising the kurtosis. This is property seems to be then carried over to the predictions for the metrics. When these actual values and predictions are put back through the same non-linear transformation functions these graphs exhibit a much more circular center making the relationship look much less impressive. Though it is clear that the model does a good job prediction the general direction that closing prices will go from quarter to quarter.

Chapter 5: Conclusion

The data used for the analysis in this study was the price history, quarterly financial statements, and insider transactions from form 4 filings from the SEC's EDGAR database. The first significant analysis was in Chapter 4.1 which involved plotting price movements after every inside trade for each company in the analysis. This yielded no numerical results but was useful for describing methodologies that are crucial to the rest of the analysis.

Next, Chapter 4.2 builds confidence intervals comparing buys and sells to determine if insiders have an advantage regarding timing in the markets for their companies. The results were clear as the returns based on insiders buying reached up to 600% of average the returns from a strategy based solely on insider selling. Further, the confidence intervals had a significant and increasing spread as the holding periods considered increased from 5 days to 4 years. This established a difference between behaviors giving validation to an investigation of incorporating insider buying vs. selling information into a model. This hypothesis is that insider buys may indicate a good time to buy a stock, but selling information could be more useful by preventing purchasing a stock before a disastrous price decline.

Once a difference between the behavior of buying and selling was confirmed Chapter 4.3 takes the next step and builds a model based on all of this information. The main goal being to determine if there was a difference between the quality of buying and selling info. Two different model types were used for this. One was a linear regression and the other a multi-layer perceptron (MLP) neural network. Both models were trained to do the same thing; take the quarter delta path back data set as an input and predict the change in low, high, and closing prices relative to these metrics in the last quarter. Timing predictions of when the high and low prices would occur is also included in the predictions.

The results from the linear regression and the MLP model were compared. The regression consistently out preformed the MLP model with regard to price movement predictions. Only the best structured MLP network came close the performance of regressions for price predictions. On the other hand accuracy was reversed for timing predictions; that is the MLP was consistently more accurate than the regression model for determining when these prices would occur in the quarter. This indicates that it would be best to use a combination of the two models to gain any insight on markets. The cause for the boosted performance of timing predictions for the MLP network was hypothesized to be the non-linearities produced in the prediction space thanks to the activation function on each node.

One of the main goals initially set out was to determine a difference between the quality of buying and selling information. No significant difference was found, though it was clear that adding either type of information to the model increased accuracy. This is seen by the fact that the models trained with only the quarterly financial information were the least accurate compared to any models with some combination of insider trading information included. More notably, the most accurate models consistently included all the insider trading information; this shows that buying and selling information provide different insights, a clear justification for using both. The final significant insight is that price formation based is clearly path dependent. This is shown by the fact that when models considered more information going back to make one prediction these predictions become more and more accurate.

During the data collection process one promising area for future research came up; investigating the relationship of phantom stock with stock price movements. Phantom stocks are a relatively new tool corporations use to compensate their employees. The timing of transactions on phantom stocks is not regulated but only tracked as phantom stocks are contracts that only a company may purchase back from an employee. An employee can execute this option any time after a certain period specified by the contract. This contract is always worth the current market value of a company's public stock. This provides an advantage to the employee as they can sell before bad releases of information without fear of any punitive action from the SEC. As transactions are still reported they may provide more immediate value to models as insiders have a lot to gain and nothing to lose by selling these equities based on privileged information that will affect the company in the near future. The one downside to this potential research is that phantom stocks are a new concept in the corporate world so there is not much data available. However over time this data will accumulate allowing analysis to find new insights from insiders' behavior.

Appendix Item A

Company Information and Price History Data Collection Code

```
#install.packages('finreportr')
install.packages('BatchGetSymbols')
library(finreportr)
library(BatchGetSymbols)


  ### READING CSV FILE AND GENERATING DATAFRAME
stocks = read.csv("1-1  -  stocks_v0.csv")
stocks['Ticker']
stocks['cik']          <-  as.vector(rep(0, nrow(stocks)))
stocks['sic']          <-  as.vector(rep(0, nrow(stocks)))
stocks['fy_end']       <-  as.vector(rep(0, nrow(stocks)))
stocks['zip']          <-  as.vector(rep(0, nrow(stocks)))
stocks['state']        <-  as.vector(rep(0, nrow(stocks)))
stocks['state_inc']    <-  as.vector(rep(0, nrow(stocks)))

  ### GET COMPANY INFO FOR ALL TICKERS IN ZACHS.COM STOCK SCREEN
for (index in 1:nrow(stocks)){
  tryCatch({
    company <- CompanyInfo(stocks[index, 'Ticker'])
    stocks[index, 'Company.Name'] <- sapply(stocks[index, 'Company.Name'], tolower)
    stocks[index, 'cik']         <- company['CIK']
    stocks[index, 'sic']         <- company['SIC']
    stocks[index, 'fy_end']       <- company['FY.end']
    stocks[index, 'state']       <- company['state']
    stocks[index, 'state_inc']    <- company['state.inc']
      # take a nap (adjust accordingly to connection speed)
    Sys.sleep(0.5)
  }, error = function(e) {})
}

  ### WRITE Stocks_v1: NEW LIST OF STOCKS WITH RETRIEVABLE COMPANY INFO (ie: CIK, SIC...)
stocks_v1 <- subset(stocks, cik!="0")
write.csv(file='1-2  -  stocks_v1', x=stocks_v1)

  ### CHANGE DIRECTORY TO WHERE THE CSVs of HISTORICAL PRICE DATA WILL BE SAVED
# stocks_v1 <- read.csv("1-2  -  stocks_v1.csv")
setwd("/Users/Paul1/Google Drive/Monkey/Algo/v2/1-2  -  Price CSVs")


### GET PRICE HISTORY:   LOOP THROUGH FOR ALL TICKERS
tickers = as.vector(stocks_v1['Ticker'])
```

```
#typeof(tickers)
for (index in 1:nrow(stocks_v1)){
 stock   <- tickers[index,1]

 ### RESET ALL LISTS FOR DF COLLECTION
 dates   <- c()
 opens   <- c()
 highs   <- c()
 lows    <- c()
 closes  <- c()
 volumes <- c()

 ### LOOPS THROUGH DAYS STARTING AT PRESENT GOING BACK IN INTERVALS OF 1000 TO GET
  # HISTORICAL TRADING DATA AND POPULATES LISTS BY APPENDING ACCORDINGLY
 for(interval in seq(from = 1000, to = 15000, by = 1000)){
  tryCatch({
   df      <- BatchGetSymbols(stock, first.date=Sys.Date()-interval, last.date=Sys.Date()-interval+1000,
                              do.complete.data = TRUE, do.cache=FALSE, thresh.bad.data=0.1)
   dates   <- append(dates,   df$df.tickers$ref.date)
   opens   <- append(opens,   df$df.tickers$price.open)
   highs   <- append(highs,   df$df.tickers$price.high)
   lows    <- append(lows,    df$df.tickers$price.low)
   closes  <- append(closes,  df$df.tickers$price.close)
   volumes <- append(volumes, df$df.tickers$volume)
   print(stock)
   print(index)
   print(interval)
   if (nrow(df$df.tickers)==0) break
  }, error = function(e) {})
 }

 ### SAVE DATAFRAME TO CSV FILE THEN IN WORKSPACE
 price_df  = data.frame(dates, opens, highs, lows, closes, volumes)
 write.csv(file=paste(toString(stock),'.csv', sep=''), x=price_df)
 assign(toString(stock), price_df)
}
```

Appendix Item B

Insider Trading Scraping Script

```
import os
import pandas as pd
import time
import pickle
from bs4 import BeautifulSoup
import urllib.request
from datetime import datetime
from pandas.tseries import offsets
from datetime import date
import sys
import csv
import builtins

NO RUN: Creates stocks_v3 pickle... o
### READING Stocks_v1 (LIST OF TICKERS AND COMPANY DATA)
# path = "/Users/Paul1/Google Drive/Monkey/Algo/v2"
path = "C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2"
os.chdir(path)
stocks_v1 = pd.read_csv('1-2  -  stocks_v1.csv')

### MAKING THE FILE LIST
# path = "/Users/Paul1/Google Drive/Monkey/Algo/v2/1-2  -  Price CSVs"
path = "C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2/1-2  -  Price CSVs"
filelist = os.listdir(path)

### IDENTIFY TICKERS WITH BAD FILE HISTORY
stocks_v2_list = []
for file in filelist:
    if '.' not in file[:-4]:
        stocks_v2_list.append(file[:-4])

### REMOVING ALL INVALID ROWS IMPORTED
stocks_v2 = stocks_v1[stocks_v1.Ticker.isin(stocks_v2_list)]
# stocks_v1.head()

### REMOVING UNNECESSARY COLUMN SHOWN IN STOCKS_V2 ABOVE ... THEN REINDEX
# del stocks_v1['Unnamed: 0']
stocks_v2 = stocks_v2.reset_index()

### FORM LIST OF CIK NUMBERS WITH CORRECT STRING LENGTH AS A STRING (TO WORK WITH URLs)
cik_str = []
for index in range(len(stocks_v2)):
    cik_string = str(stocks_v2.cik[index])
    cik_zeros_added = 9 - len(cik_string)
    cik_str.append('0'*cik_zeros_added + cik_string)

### APPEND CIK_STR TO DATAFRAME
stocks_v2['cik_str'] = cik_str

### WRITING PICKLE of stocks_v2
# path = '/Users/Paul1/Google Drive/Monkey/Algo/v2'
path = 'C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2'
os.chdir(path)

with open('stocks_v2.pickle', 'wb') as handle:
    pickle.dump(stocks_v2, handle, protocol=pickle.HIGHEST_PROTOCOL)

### FINDS ALL Acc-No's and Filing_Dates FOR: CIK
def get_form4_accnos(cik):
    start = time.time()
    ### LISTS TO FILL
    acc_no_list    = []
    for counter in range(100): ### counter loops through 100 acc_no's in one page then moves to next
        try:
            ### DECLARATIONS TO GENERATE LINKS
            search_str_1 = 'https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK='
            search_str_2 = '&type=&dateb=&owner=only&start='
            search_str_3 = '&count='
            search_result_link       = search_str_1 + cik + search_str_2 + str(counter*100) + search_str_3 +
str((counter+1)*100)

            ### FIND TABLE BY PARSING PAGE
            with urllib.request.urlopen(search_result_link) as response:
                results_html = response.read()
            soup_results  = BeautifulSoup(results_html, 'lxml')

            results_table = soup_results.find_all('table', {'class':'tableFile2'})
            results_rows  = results_table[0].find_all('tr')

            ### PARSE INTO ROWS

            for results_row in results_rows:
                col_entries = results_row.find_all('td')
                try: ### WORK AROUND FOR HEADER ROW HAVING 'th' TAG WHICH RETURNS A LIST LEN=0 when
searching for 'td'
                    if col_entries[0].string == '4':  ### FORM CHECK
                        acc_no_list.append(col_entries[2].contents[2][8:28])

                except IndexError:
                    pass
        except:
            break
    print("   done getting acc_#'s ... total time taken: " + str(time.time() - start))

    return acc_no_list
### RETURNS (XML) form4_names_list  and datetime_list FOR cik_str
def get_form4names_datetime_lists(cik):
    acc_no_list = get_form4_accnos(cik)
    ### LIST THAT WILL BE RETURNED WITH DOC NAMES
    acc_no_list_returned = []
    form4_names_list     = []
    form4_datetime_list  = []

    ### LOOP THROUGH ACC_NOs FOR COMPANY
    for counters, acc_no in enumerate(acc_no_list):
        try:
            if counters % 100 == 0:
                print('working on names and datetimes for cik: ' +cik+ "    working on doc #:   " +str(counters)+ '   out of 
'+str(len(acc_no_list)))


            ### GENERATE STRINGS FOR SUBMISSION INDEX LINK
            url_1 = 'https://www.sec.gov/Archives/edgar/data/'
            url_2 = '-index.htm'
            acc_no_wDash  = acc_no                  ### Acc_No (with dash for url generation)
            acc_no_noDash = acc_no.replace('-', '')  ### Acc_No (no dashes for url generation)
            submission_link = url_1 + cik + '/' + acc_no_noDash + '/' + acc_no_wDash + url_2  #### ROOM to view


            ### READ THE HTML OF PAGE TO submision_page
            with urllib.request.urlopen(submission_link) as response:
                submission_page = response.read()

            ### PARSE submission_page   TO   soup_submission
            soup_submission = BeautifulSoup(submission_page, 'lxml')
            doc_table = soup_submission.find('table')

            ### LOOP THROUGH  row ->
            for row in doc_table.find_all('tr'):
                for row_entry in row.find_all('td'):
                    link_refs = row_entry.find_all('a')
                    for entry in link_refs:
                        if entry.string[-4:] == '.xml':
                            xml_name = entry.string


            ### RETRIEVE submitted_datetime
            soup_header = soup_submission.find_all('div', {'class':'formGrouping'})
            dt_str = soup_header[0].contents[7].string
            dt = datetime.strptime(dt_str,'%Y-%m-%d %H:%M:%S')

            ### APPEND NEW DOCUMENT INFO TO LIST (NEW acc_no_list TO ACCOUNT FOR BAD URLs AND THE LIKE)
            acc_no_list_returned.append(acc_no)
            form4_names_list.append(xml_name)
            form4_datetime_list.append(dt)
        except:
            pass

    print('done getting doc_names and datetimes for cik #:   ' + cik)
    return(acc_no_list_returned, form4_names_list, form4_datetime_list)
def get_form4_info(cik, acc_no, doc_name, dt):
    transact_date_list_t   = []
    accecpt_date_list_t    = []
    acc_no_list_t          = []
    security_title_list_t  = []
    aqq_disp_list_t        = []
    vol_list_t             = []
    price_list_t           = []
    vol_owned_after_list_t = []
    is_director_list_t     = []
    is_officer_list_t      = []
    percents_10_list_t     = []
    equity_swap_list_t     = []

    ### GETTING URL AND PARSING PAGE
    url_1 = 'https://www.sec.gov/Archives/edgar/data/'
    acc_no_noDash = acc_no.replace('-', '')
    form_url = url_1 + cik + '/' + acc_no_noDash + '/' + doc_name
    # print(form_url)

    with urllib.request.urlopen(form_url) as response:
        xml_doc = response.read()
    soup_xml = BeautifulSoup(xml_doc, 'xml')


    ### VARIED STATE PARSE OBJECTS
    transact_dates   = soup_xml.find_all('transactionDate')
    security_titles  = soup_xml.find_all('securityTitle')
    aqq_disps        = soup_xml.find_all('transactionAcquiredDisposedCode')
    vols             = soup_xml.find_all('transactionShares')
    prices           = soup_xml.find_all('transactionPricePerShare')
    vol_owned_afters = soup_xml.find_all('sharesOwnedFollowingTransaction')
    is_directors     = soup_xml.find_all('isDirector')
    is_officers      = soup_xml.find_all('isOfficer')
    percents_10s     = soup_xml.find_all('isTenPercentOwner')
    equity_swaps     = soup_xml.find_all('equitySwapInvolved')


    ### APPENDING SCRAPE RESULTS TO ALL LISTS
    for index in range(len(transact_dates)):
        transact_date_list_t.append(transact_dates[index].value.string)
        accecpt_date_list_t.append(dt) # STEADY STATE
        acc_no_list_t.append(acc_no) # STEADY STATE
        security_title_list_t.append(security_titles[index].value.string)
        aqq_disp_list_t.append(aqq_disps[index].value.string)
        vol_list_t.append(vols[index].value.string)
        price_list_t.append(prices[index].value.string)
        vol_owned_after_list_t.append(vol_owned_afters[index].value.string)
        if len(is_directors) == 0:
            is_director_list_t.append(0)
        else:
            is_director_list_t.append(is_directors[0].contents[0])
        if len(is_officers) == 0:
            is_officer_list_t.append(0)
        else:
            is_officer_list_t.append(is_officers[0].contents[0])
        if len(percents_10s) == 0:
            percents_10_list_t.append(0)
        else:
            percents_10_list_t.append(percents_10s[0].contents[0])
        equity_swap_list_t.append(equity_swaps[index].contents[0])

    return[transact_date_list_t, accecpt_date_list_t, acc_no_list_t, security_title_list_t,
           aqq_disp_list_t, vol_list_t, price_list_t, vol_owned_after_list_t, is_director_list_t,
           is_officer_list_t, percents_10_list_t, equity_swap_list_t]
### CSV FOR 1 CIK PLAY
def cik_form4s_to_csv(cik):
    start = time.time()

    ### GET each form 5 info: acc_no_list, name_list, dt_list
    print('getting acc_no_list, name_list, dt_list for cik now ...')
    print('...')
    acc_no_list_csv, name_list_csv, dt_list_csv = get_form4names_datetime_lists(cik)
    print('cik #:  ' +cik+  '  contains ' +str(len(acc_no_list_csv))+ '  documents to retrieve info from ')
    print('...')

    ### VARIABLES FOR CSV GENERATION
    row = stocks_v2[stocks_v2.cik_str== cik] # to identify the ticker
    ticker = row.iloc[0]['Ticker']
    transact_date_list   = []
    accecpt_date_list    = []
    acc_no_list          = []
    security_title_list  = []
    aqq_disp_list        = []
    vol_list             = []
    price_list           = []
    vol_owned_after_list = []
    is_director_list     = []
    is_officer_list      = []
    percents_10_list     = []
    equity_swap_list     = []


    ### LOOP: EACH acc_no FOR cik
    for index in range(len(acc_no_list_csv)):
```

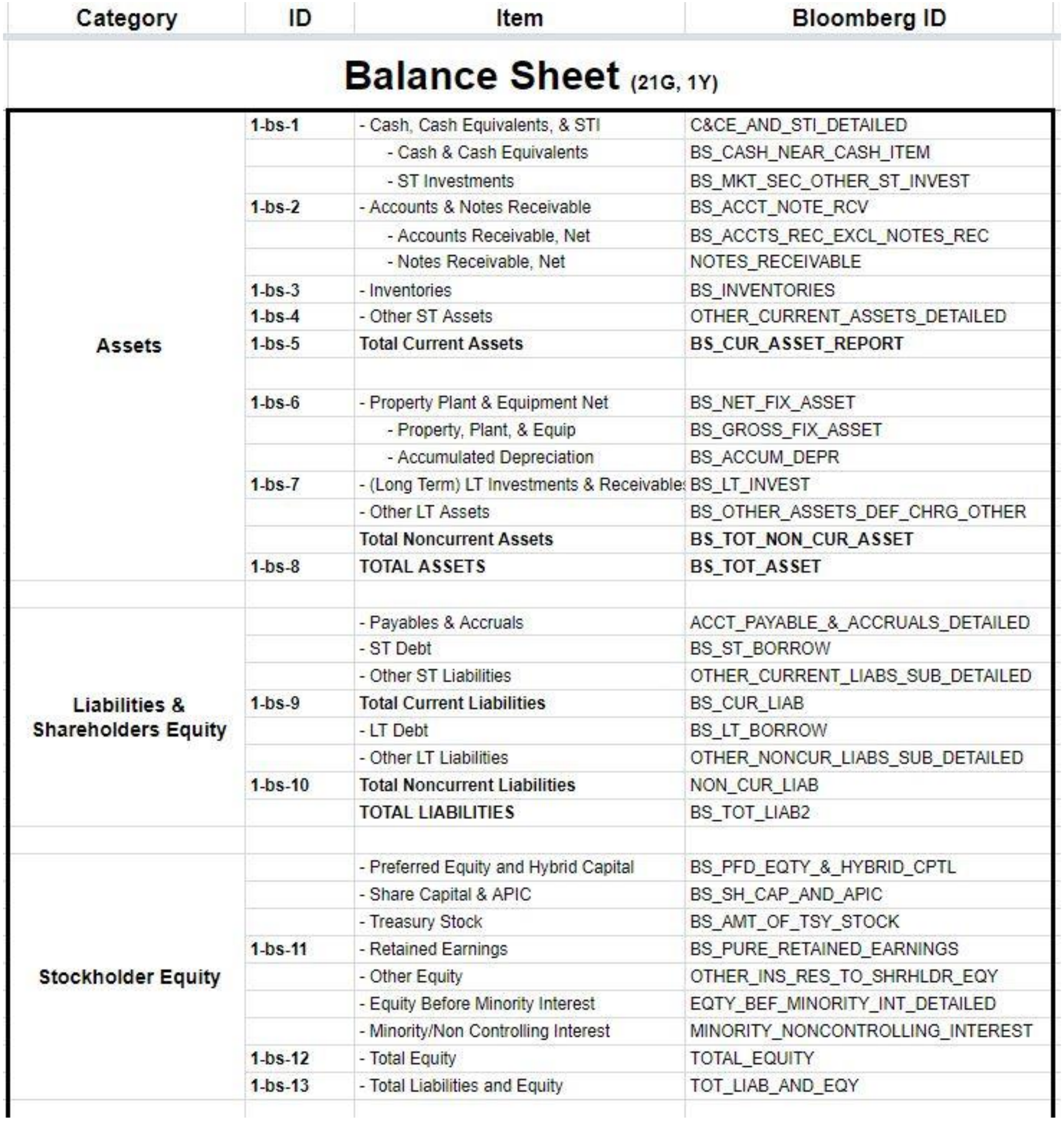
```
    try:
        if index % 100 == 0:
            print('working on csv for cik: ' +cik+ '    doc #: ' + str(index)+ ' out of '+str(len(acc_no_list_csv)))

        acc_no   = str(acc_no_list_csv[index])
        doc_name = str(name_list_csv[index])
        dt       = str(dt_list_csv[index])

        ### GETTING INFO ON 1 FORM ONLY
        [transact_date_list_t,  accecpt_date_list_t,    acc_no_list_t,
        security_title_list_t, aqq_disp_list_t,       vol_list_t,
        price_list_t,          vol_owned_after_list_t, is_director_list_t,
        is_officer_list_t,     percents_10_list_t,     equity_swap_list_t]  = get_form4_info(cik, acc_no, doc_name, dt)

        ### APPENDING
        transact_date_list.extend(transact_date_list_t)
        accecpt_date_list.extend(accecpt_date_list_t)
        acc_no_list.extend(acc_no_list_t)
        security_title_list.extend(security_title_list_t)
        aqq_disp_list.extend(aqq_disp_list_t)
        vol_list.extend(vol_list_t)
        price_list.extend(price_list_t)
        vol_owned_after_list.extend(vol_owned_after_list_t)
        is_director_list.extend(is_director_list_t)
        is_officer_list.extend(is_officer_list_t)
        percents_10_list.extend(percents_10_list_t)
        equity_swap_list.extend(equity_swap_list_t)

    except: #INDICATES TO USER WHERE A FORM HAD INCORRECT INFORMATION
        print('error on index:  ' +str(index)+ '    occured with the following url: ')
        url_1 = 'https://www.sec.gov/Archives/edgar/data/'
        acc_no_noDash = acc_no.replace('-', '')
        form_url = url_1 + cik + '/' + acc_no_noDash + '/' + doc_name
        print(form_url)
        print("cik      = '" + cik+ "'")
        print("acc_no   = '" + acc_no+ "'")
        print("doc_name = '" + doc_name+ "'")
        print("dt       = '" + dt+ "'")
        pass

### WRITE CSV
# path = '/Users/Paul1/Google Drive/Monkey/Algo/v2/1-3  -  Form 4 Info CSVs'
path = 'C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2/1-3  -  Form 4 Info CSVs'
os.chdir(path)

form4_hist_dict =  {'transact_date': transact_date_list, 'accecpt_date':accecpt_date_list, 'acc_no':acc_no_list,
                    'security_title':security_title_list, 'aqq_disp':aqq_disp_list, 'vol':vol_list,
                    'price':price_list, 'vol_owned_after':vol_owned_after_list, 'is_director':is_director_list,
                    'is_officer':is_officer_list, 'percents_10':percents_10_list, 'equity_swap':equity_swap_list}
form4_hist_df = pd.DataFrame(form4_hist_dict)
file = '1-3  -  ' + ticker + '_form4_hist_info.csv'
```

```
    form4_hist_df.to_csv(file)

    print('done with ' + cik)
    print("total time taken this loop: ", time.time() - start)
### GET BATCH OF CSVs
def get_cik_batch_form4_info(start_of_batch, end_of_batch):
    start = time.time()
    cik_batch = stocks_v2.iloc[start_of_batch:end_of_batch]

    ### LOOP GETTING ALL FORM 4's FOR 1 COMPANY
    for tick_counter, cik in enumerate(cik_batch['cik_str']):
        try:
            print(str(tick_counter) + str(cik))
            #print("for cik batch starting at #:   " + str(start) + '   & stop at:   ' + str(stop) + '    working on cik #: '+ str(tick_counter))
            cik_form4s_to_csv(cik)

        except:
            print('error on: ' +str(tick_counter) + str(cik))
            continue
```

Appendix Item C

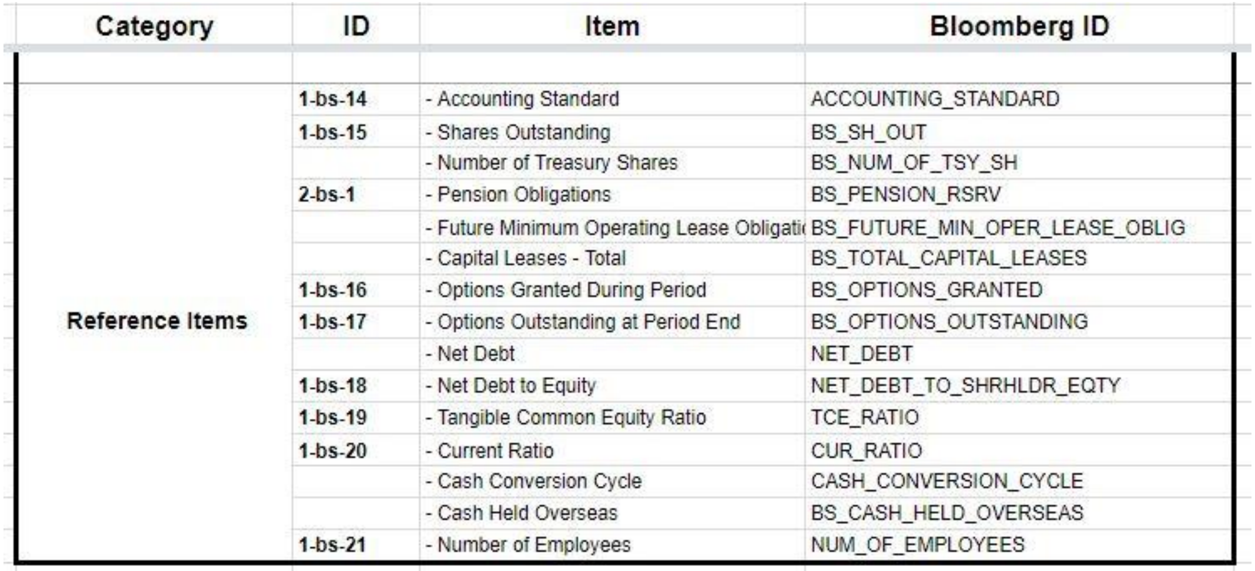
All Information on Balance Sheet from Bloomberg Terminals

Balance Sheet Part 1 (of 2)



Category	ID	Item	Bloomberg ID
Balance Sheet (21G, 1Y)			
Assets	**1-bs-1**	- Cash, Cash Equivalents, & STI	C&CE_AND_STI_DETAILED
		- Cash & Cash Equivalents	BS_CASH_NEAR_CASH_ITEM
		- ST Investments	BS_MKT_SEC_OTHER_ST_INVEST
	1-bs-2	- Accounts & Notes Receivable	BS_ACCT_NOTE_RCV
		- Accounts Receivable, Net	BS_ACCTS_REC_EXCL_NOTES_REC
		- Notes Receivable, Net	NOTES_RECEIVABLE
	1-bs-3	- Inventories	BS_INVENTORIES
	1-bs-4	- Other ST Assets	OTHER_CURRENT_ASSETS_DETAILED
	1-bs-5	**Total Current Assets**	**BS_CUR_ASSET_REPORT**
	1-bs-6	- Property Plant & Equipment Net	BS_NET_FIX_ASSET
		- Property, Plant, & Equip	BS_GROSS_FIX_ASSET
		- Accumulated Depreciation	BS_ACCUM_DEPR
	1-bs-7	- (Long Term) LT Investments & Receivable	BS_LT_INVEST
		- Other LT Assets	BS_OTHER_ASSETS_DEF_CHRG_OTHER
		Total Noncurrent Assets	**BS_TOT_NON_CUR_ASSET**
	1-bs-8	**TOTAL ASSETS**	**BS_TOT_ASSET**
Liabilities & Shareholders Equity		- Payables & Accruals	ACCT_PAYABLE_&_ACCRUALS_DETAILED
		- ST Debt	BS_ST_BORROW
		- Other ST Liabilities	OTHER_CURRENT_LIABS_SUB_DETAILED
	1-bs-9	**Total Current Liabilities**	BS_CUR_LIAB
		- LT Debt	BS_LT_BORROW
		- Other LT Liabilities	OTHER_NONCUR_LIABS_SUB_DETAILED
	1-bs-10	**Total Noncurrent Liabilities**	NON_CUR_LIAB
		TOTAL LIABILITIES	BS_TOT_LIAB2
Stockholder Equity		- Preferred Equity and Hybrid Capital	BS_PFD_EQTY_&_HYBRID_CPTL
		- Share Capital & APIC	BS_SH_CAP_AND_APIC
		- Treasury Stock	BS_AMT_OF_TSY_STOCK
	1-bs-11	- Retained Earnings	BS_PURE_RETAINED_EARNINGS
		- Other Equity	OTHER_INS_RES_TO_SHRHLDR_EQY
		- Equity Before Minority Interest	EQTY_BEF_MINORITY_INT_DETAILED
		- Minority/Non Controlling Interest	MINORITY_NONCONTROLLING_INTEREST
	1-bs-12	- Total Equity	TOTAL_EQUITY
	1-bs-13	- Total Liabilities and Equity	TOT_LIAB_AND_EQY

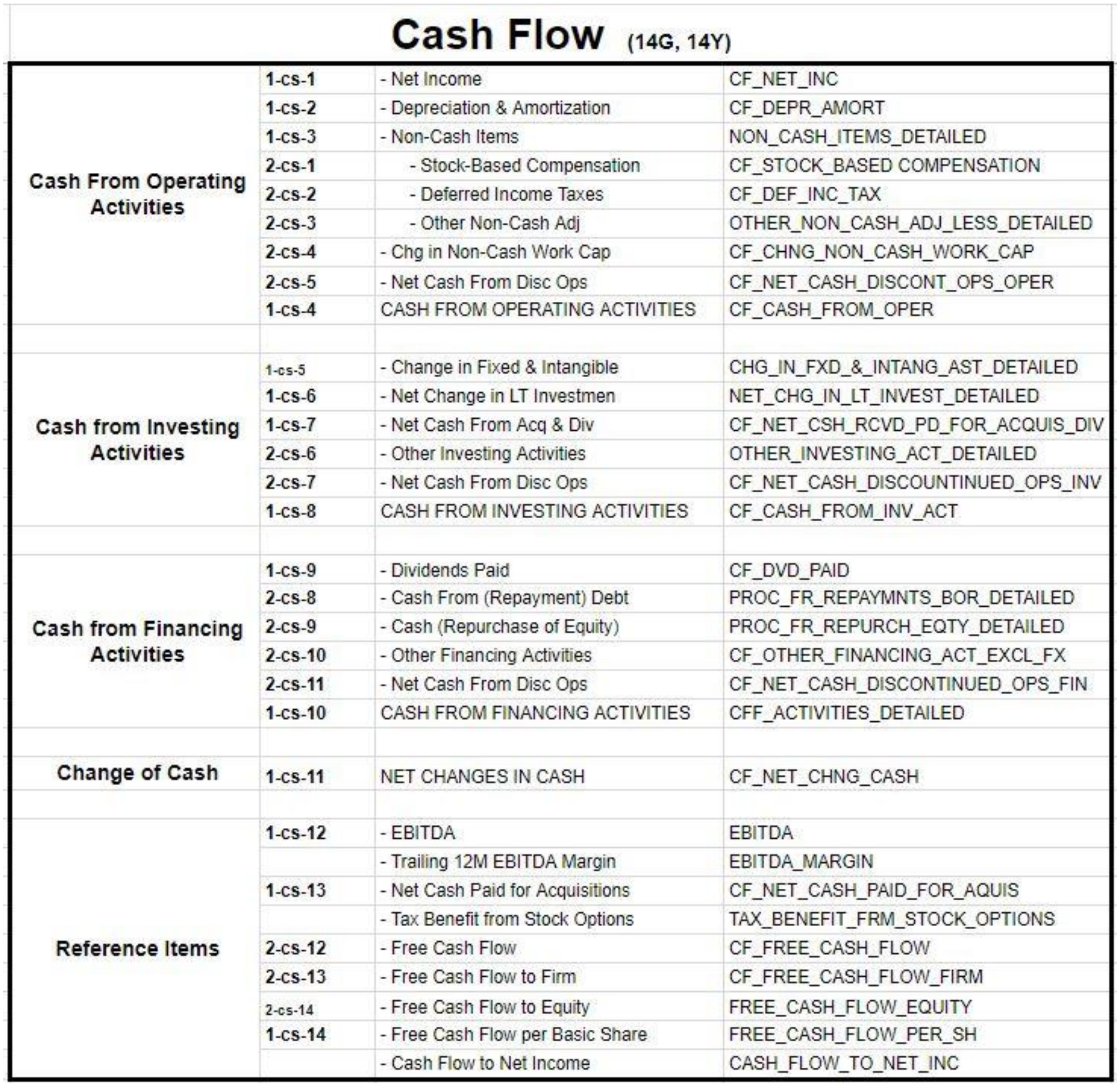
Balance Sheet Part 2 (of 2)



Category	ID	Item	Bloomberg ID
Reference Items	1-bs-14	- Accounting Standard	ACCOUNTING_STANDARD
	1-bs-15	- Shares Outstanding	BS_SH_OUT
		- Number of Treasury Shares	BS_NUM_OF_TSY_SH
	2-bs-1	- Pension Obligations	BS_PENSION_RSRV
		- Future Minimum Operating Lease Obligati	BS_FUTURE_MIN_OPER_LEASE_OBLIG
		- Capital Leases - Total	BS_TOTAL_CAPITAL_LEASES
	1-bs-16	- Options Granted During Period	BS_OPTIONS_GRANTED
	1-bs-17	- Options Outstanding at Period End	BS_OPTIONS_OUTSTANDING
		- Net Debt	NET_DEBT
	1-bs-18	- Net Debt to Equity	NET_DEBT_TO_SHRHLDR_EQTY
	1-bs-19	- Tangible Common Equity Ratio	TCE_RATIO
	1-bs-20	- Current Ratio	CUR_RATIO
		- Cash Conversion Cycle	CASH_CONVERSION_CYCLE
		- Cash Held Overseas	BS_CASH_HELD_OVERSEAS
	1-bs-21	- Number of Employees	NUM_OF_EMPLOYEES

Appendix Item D

All Information on Cash Flow Statement from Bloomberg Terminals

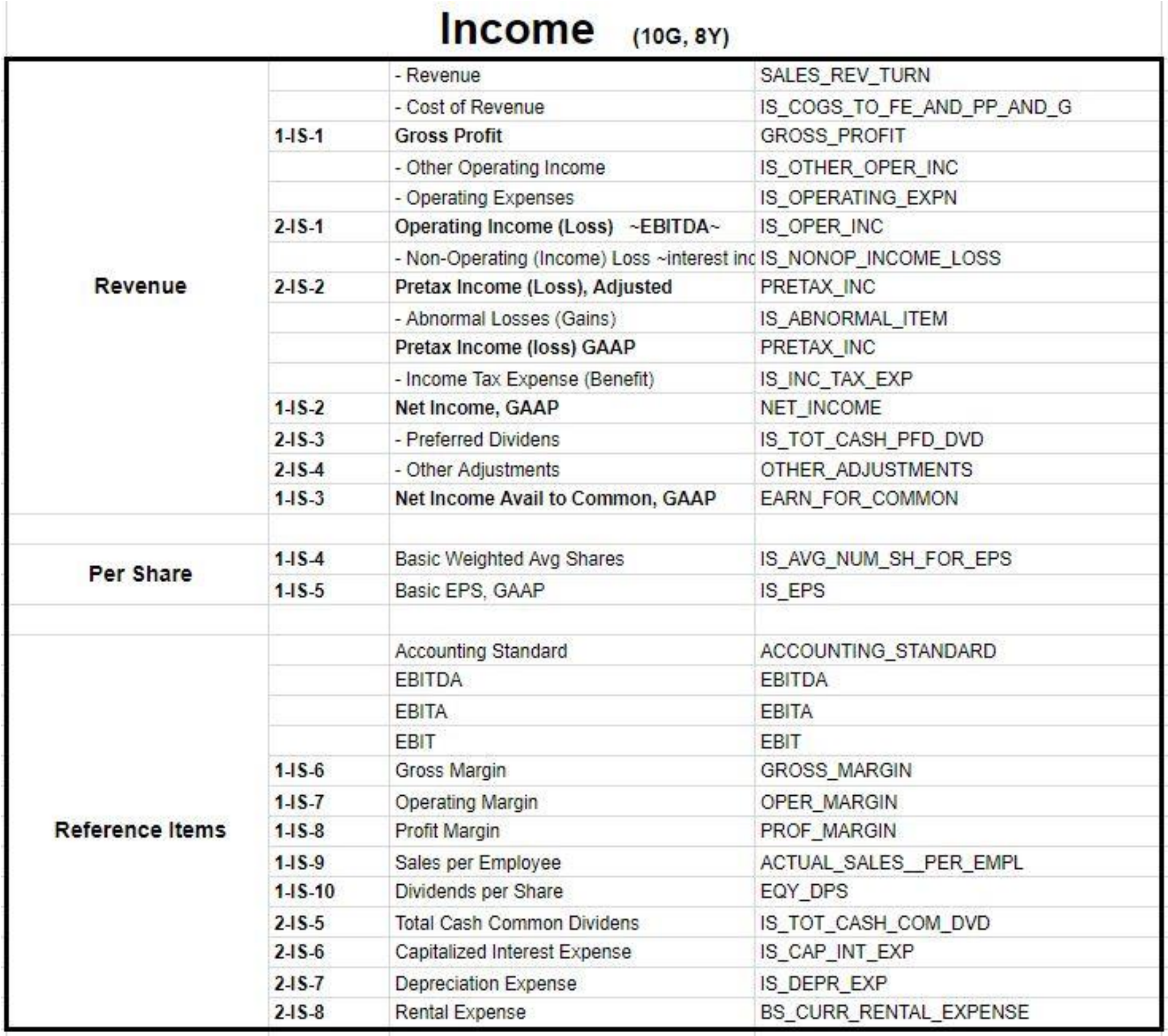


Cash Flow (14G, 14Y)

Cash From Operating Activities	1-cs-1	- Net Income	CF_NET_INC
	1-cs-2	- Depreciation & Amortization	CF_DEPR_AMORT
	1-cs-3	- Non-Cash Items	NON_CASH_ITEMS_DETAILED
	2-cs-1	- Stock-Based Compensation	CF_STOCK_BASED COMPENSATION
	2-cs-2	- Deferred Income Taxes	CF_DEF_INC_TAX
	2-cs-3	- Other Non-Cash Adj	OTHER_NON_CASH_ADJ_LESS_DETAILED
	2-cs-4	- Chg in Non-Cash Work Cap	CF_CHNG_NON_CASH_WORK_CAP
	2-cs-5	- Net Cash From Disc Ops	CF_NET_CASH_DISCONT_OPS_OPER
	1-cs-4	CASH FROM OPERATING ACTIVITIES	CF_CASH_FROM_OPER
Cash from Investing Activities	1-cs-5	- Change in Fixed & Intangible	CHG_IN_FXD_&_INTANG_AST_DETAILED
	1-cs-6	- Net Change in LT Investmen	NET_CHG_IN_LT_INVEST_DETAILED
	1-cs-7	- Net Cash From Acq & Div	CF_NET_CSH_RCVD_PD_FOR_ACQUIS_DIV
	2-cs-6	- Other Investing Activities	OTHER_INVESTING_ACT_DETAILED
	2-cs-7	- Net Cash From Disc Ops	CF_NET_CASH_DISCOUNTINUED_OPS_INV
	1-cs-8	CASH FROM INVESTING ACTIVITIES	CF_CASH_FROM_INV_ACT
Cash from Financing Activities	1-cs-9	- Dividends Paid	CF_DVD_PAID
	2-cs-8	- Cash From (Repayment) Debt	PROC_FR_REPAYMNTS_BOR_DETAILED
	2-cs-9	- Cash (Repurchase of Equity)	PROC_FR_REPURCH_EQTY_DETAILED
	2-cs-10	- Other Financing Activities	CF_OTHER_FINANCING_ACT_EXCL_FX
	2-cs-11	- Net Cash From Disc Ops	CF_NET_CASH_DISCONTINUED_OPS_FIN
	1-cs-10	CASH FROM FINANCING ACTIVITIES	CFF_ACTIVITIES_DETAILED
Change of Cash	1-cs-11	NET CHANGES IN CASH	CF_NET_CHNG_CASH
Reference Items	1-cs-12	- EBITDA	EBITDA
		- Trailing 12M EBITDA Margin	EBITDA_MARGIN
	1-cs-13	- Net Cash Paid for Acquisitions	CF_NET_CASH_PAID_FOR_AQUIS
		- Tax Benefit from Stock Options	TAX_BENEFIT_FRM_STOCK_OPTIONS
	2-cs-12	- Free Cash Flow	CF_FREE_CASH_FLOW
	2-cs-13	- Free Cash Flow to Firm	CF_FREE_CASH_FLOW_FIRM
	2-cs-14	- Free Cash Flow to Equity	FREE_CASH_FLOW_EQUITY
	1-cs-14	- Free Cash Flow per Basic Share	FREE_CASH_FLOW_PER_SH
		- Cash Flow to Net Income	CASH_FLOW_TO_NET_INC

Appendix Item E

All Information on Income Statement from Bloomberg Terminals



Income (10G, 8Y)

Revenue		- Revenue	SALES_REV_TURN
		- Cost of Revenue	IS_COGS_TO_FE_AND_PP_AND_G
	1-IS-1	**Gross Profit**	GROSS_PROFIT
		- Other Operating Income	IS_OTHER_OPER_INC
		- Operating Expenses	IS_OPERATING_EXPN
	2-IS-1	**Operating Income (Loss) ~EBITDA~**	IS_OPER_INC
		- Non-Operating (Income) Loss ~interest inc	IS_NONOP_INCOME_LOSS
	2-IS-2	**Pretax Income (Loss), Adjusted**	PRETAX_INC
		- Abnormal Losses (Gains)	IS_ABNORMAL_ITEM
		Pretax Income (loss) GAAP	PRETAX_INC
		- Income Tax Expense (Benefit)	IS_INC_TAX_EXP
	1-IS-2	**Net Income, GAAP**	NET_INCOME
	2-IS-3	- Preferred Dividens	IS_TOT_CASH_PFD_DVD
	2-IS-4	- Other Adjustments	OTHER_ADJUSTMENTS
	1-IS-3	**Net Income Avail to Common, GAAP**	EARN_FOR_COMMON
Per Share	**1-IS-4**	Basic Weighted Avg Shares	IS_AVG_NUM_SH_FOR_EPS
	1-IS-5	Basic EPS, GAAP	IS_EPS
Reference Items		Accounting Standard	ACCOUNTING_STANDARD
		EBITDA	EBITDA
		EBITA	EBITA
		EBIT	EBIT
	1-IS-6	Gross Margin	GROSS_MARGIN
	1-IS-7	Operating Margin	OPER_MARGIN
	1-IS-8	Profit Margin	PROF_MARGIN
	1-IS-9	Sales per Employee	ACTUAL_SALES__PER_EMPL
	1-IS-10	Dividends per Share	EQY_DPS
	2-IS-5	Total Cash Common Dividens	IS_TOT_CASH_COM_DVD
	2-IS-6	Capitalized Interest Expense	IS_CAP_INT_EXP
	2-IS-7	Depreciation Expense	IS_DEPR_EXP
	2-IS-8	Rental Expense	BS_CURR_RENTAL_EXPENSE

Appendix Item F

Quarterly Financial State Scraping Code

```
### IMPORTS
require(data.table)
require(Rblpapi)

library(Rblpapi)
con <- blpConnect()

### This uses a csv from a stock screener found on zacks.com to generate a more comprehensive list of information on
### all companies in a watch list of companies

### CLEARS THE CONSOLE
cat("\014")

### Set wd for Paul (Mac)
#setwd("/Users/Paul1/Google Drive/Monkey/Algo/v2")
###   Set wd for Paul (Windows)
#setwd("C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2")
###   Set wd for Paul (Windows)
setwd("D:/")

require(data.table)
require(Rblpapi)
library(Rblpapi)
con <- blpConnect()

### READING CSV FILE AND GENERATING DATAFRAME
stocks = read.csv("stocks_v4.csv")

### GETTING INFORMATION IN CORRECT FORMAT FOR ONE COMPANY
setwd("D:/1-4  -  Full State History CSVs")


### LOOPING THROUGH ALL COMPANIES IN stocks_v3 DATAFRAME
for(index in 2040:nrow(stocks)){
  # if (index > 1) {
  #   break
  # }
  state_data = data.frame()

  tryCatch({ ##############################

    ### SETTING TICKER FOR QUERY
    ticker <- stocks[index, 'Ticker']
    start_date <- as.Date(stocks[index, 'bloom_date']);
```

```
bdh_ticker <- paste(ticker,  "US Equity")
print(paste("ticker:", bdh_ticker,  "Start Date:", start_date))

state_data_1 <- bdh(bdh_ticker, c(
  'C&CE_AND_STI_DETAILED', 'BS_CASH_NEAR_CASH_ITEM', 'BS_MKT_SEC_OTHER_ST_INVEST',
  'BS_ACCT_NOTE_RCV', 'BS_ACCTS_REC_EXCL_NOTES_REC', 'NOTES_RECEIVABLE',
  'BS_INVENTORIES', 'OTHER_CURRENT_ASSETS_DETAILED', 'BS_CUR_ASSET_REPORT',
  'BS_NET_FIX_ASSET', 'BS_GROSS_FIX_ASSET', 'BS_ACCUM_DEPR',
  'BS_LT_INVEST', 'BS_OTHER_ASSETS_DEF_CHRG_OTHER', 'BS_TOT_NON_CUR_ASSET'
), start.date=start_date )

state_data_2 <- bdh(bdh_ticker, c(
  'BS_TOT_ASSET','ACCT_PAYABLE_&_ACCRUALS_DETAILED', 'BS_ST_BORROW',
  'OTHER_CURRENT_LIABS_SUB_DETAILED', 'BS_CUR_LIAB', 'BS_LT_BORROW',
  'OTHER_NONCUR_LIABS_SUB_DETAILED', 'NON_CUR_LIAB', 'BS_TOT_LIAB2',
  'BS_PFD_EQTY_&_HYBRID_CPTL', 'BS_SH_CAP_AND_APIC', 'BS_AMT_OF_TSY_STOCK',
  'BS_PURE_RETAINED_EARNINGS', 'OTHER_INS_RES_TO_SHRHLDR_EQY', 'EQTY_BEF_MINORITY_INT_DETAILED'
), start.date=start_date )

state_data_3 <- bdh(bdh_ticker, c(
  'MINORITY_NONCONTROLLING_INTEREST', 'TOTAL_EQUITY', 'TOT_LIAB_AND_EQY',
  'ACCOUNTING_STANDARD', 'BS_SH_OUT', 'BS_NUM_OF_TSY_SH',
  'BS_PENSION_RSRV', 'BS_FUTURE_MIN_OPER_LEASE_OBLIG', 'BS_TOTAL_CAPITAL_LEASES',
  'BS_OPTIONS_GRANTED', 'BS_OPTIONS_OUTSTANDING','NET_DEBT',
  'NET_DEBT_TO_SHRHLDR_EQTY', 'TCE_RATIO', 'CUR_RATIO'
), start.date=start_date )

state_data_4 <- bdh(bdh_ticker, c(
  'CASH_CONVERSION_CYCLE', 'BS_CASH_HELD_OVERSEAS', 'NUM_OF_EMPLOYEES',
  'CF_NET_INC', 'CF_DEPR_AMORT', 'NON_CASH_ITEMS_DETAILED',
  'CF_STOCK_BASED_COMPENSATION','CF_DEF_INC_TAX', 'OTHER_NON_CASH_ADJ_LESS_DETAILED', # bad line
  'CF_CHNG_NON_CASH_WORK_CAP', 'CF_NET_CASH_DISCONT_OPS_OPER', 'CF_CASH_FROM_OPER',
  'CHG_IN_FXD_&_INTANG_AST_DETAILED', 'NET_CHG_IN_LT_INVEST_DETAILED',
  'CF_NT_CSH_RCVD_PD_FOR_ACQUIS_DIV'
), start.date=start_date )

state_data_5 <- bdh(bdh_ticker, c(
  'OTHER_INVESTING_ACT_DETAILED','CF_NET_CASH_DISCONTINUED_OPS_INV',
  'CF_CASH_FROM_INV_ACT', 'CF_DVD_PAID', 'PROC_FR_REPAYMNTS_BOR_DETAILED',
  'PROC_FR_REPURCH_EQTY_DETAILED', 'CF_OTHER_FINANCING_ACT_EXCL_FX',
  'CF_NET_CASH_DISCONTINUED_OPS_FIN','CFF_ACTIVITIES_DETAILED',
  'CF_NET_CHNG_CASH', 'EBITDA', 'EBITDA_MARGIN', 'CF_NET_CASH_PAID_FOR_AQUIS',
  'CF_TAX_BENEFIT_FRM_STOCK_OPTIONS', 'CF_FREE_CASH_FLOW'
), start.date=start_date )

state_data_6 <- bdh(bdh_ticker, c(
  'CF_FREE_CASH_FLOW_FIRM', 'FREE_CASH_FLOW_EQUITY', 'FREE_CASH_FLOW_PER_SH',
  'CASH_FLOW_TO_NET_INC', 'SALES_REV_TURN', # 'PX_TO_FREE_CASH_FLOW', <<--  bad variable
  'IS_COGS_TO_FE_AND_PP_AND_G', 'GROSS_PROFIT', 'IS_OTHER_OPER_INC',
  'IS_OPERATING_EXPN', 'IS_OPER_INC', 'IS_NONOP_INCOME_LOSS',
  'IS_ABNORMAL_ITEM', 'PRETAX_INC'
```

```
    ), start.date=start_date )

    state_data_7 <- bdh(bdh_ticker, c(
      'IS_INC_TAX_EXP', 'NET_INCOME', 'IS_TOT_CASH_PFD_DVD',
      'OTHER_ADJUSTMENTS', 'EARN_FOR_COMMON', 'IS_AVG_NUM_SH_FOR_EPS',
      'IS_EPS', 'ACCOUNTING_STANDARD', 'EBITDA','EBITA', 'EBIT', 'GROSS_MARGIN',
      'OPER_MARGIN', 'PROF_MARGIN', 'ACTUAL_SALES_PER_EMPL'
    ), start.date=start_date )

    state_data_8 <- bdh(bdh_ticker, c(
      'EQY_DPS', 'IS_TOT_CASH_COM_DVD', 'IS_CAP_INT_EXP',
      'IS_DEPR_EXP', 'BS_CURR_RENTAL_EXPENSE'
    ), start.date=start_date )


    state_data <- merge(state_data_1, state_data_2, by='date', all=T)
    state_data <- merge(state_data,   state_data_3, by='date', all=T)
    state_data <- merge(state_data,   state_data_4, by='date', all=T)
    state_data <- merge(state_data,   state_data_5, by='date', all=T)
    state_data <- merge(state_data,   state_data_6, by='date', all=T)
    state_data <- merge(state_data,   state_data_7, by='date', all=T)
    state_data <- merge(state_data,   state_data_8, by='date', all=T)

  }, error = function(e) {print(paste('the loop errored on ticker = ',ticker))}) #############################

  ### SAVE DATAFRAME TO CSV FILE THEN IN WORKSPACE
  write.csv(file=paste(toString(ticker),' state hist.csv', sep=''), x=state_data)
  print(paste('done with: ',ticker, ' index #  ', index))
}
```

Appendix Item G

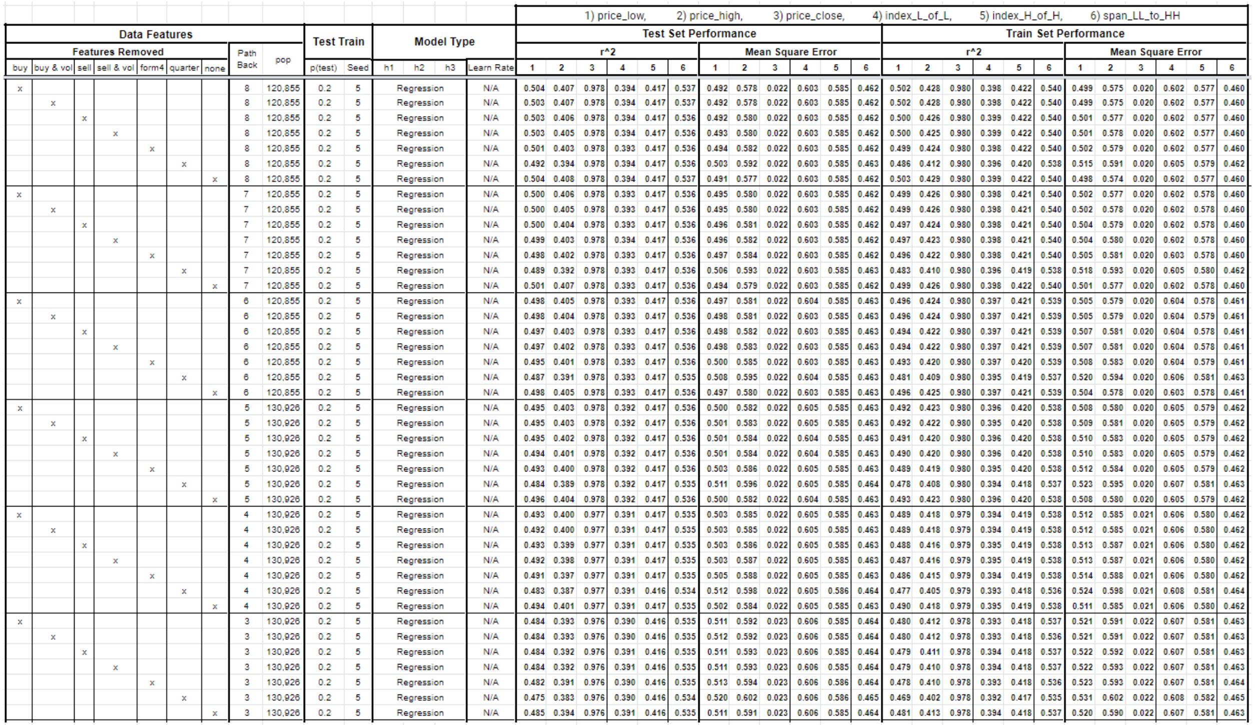
Confidence Interval Creation Script from Chapter 4.3

```
### IMPORTS
import builtins
import csv
import datetime
from datetime import timedelta
import math
import numpy as np
import os
import pandas as pd
import pickle
import scipy.stats
import statistics
import sys
import time

path = '/Users/Paul1/Google Drive/Monkey/Algo/v2'
path = 'C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2'
os.chdir(path)

### READ stocks_v4 FROM PICKLE
with open('stocks_v4.pickle', 'rb') as handle:
    stocks_v4 = pickle.load(handle)

### READ IN SP500
SP500 = pd.read_csv('sp500.csv')
SP500['Date'] = pd.to_datetime(SP500['Date'], format="%Y-%m-%d")

### DIRECTORY
path = "/Users/Paul1/Google Drive/Monkey/Algo/v2"
path = "C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2"
os.chdir(path)

with open('stocks_v4.pickle', 'rb') as handle:
    stocks_v4 = pickle.load(handle)

def get_form4(ticker):
    path = '/Users/Paul1/Google Drive/Monkey/Algo/v2/1-3  -  Form 4 Info CSVs'
    path = 'C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2/1-3  -  Form 4 Info CSVs'
    os.chdir(path)

    filename = '1-5  -  ' + ticker + '_form4_hist_info.csv'
    form4_hist = pd.read_csv(filename)
    form4_hist['transact_date'] = pd.to_datetime(form4_hist['transact_date'], format="%Y-%m-%d")
    form4_hist = form4_hist.sort_values(by = 'transact_date')
    form4_hist = form4_hist.reset_index()
    del form4_hist['index']
    return form4_hist


def get_prices(ticker):
    try:
        path_price_info = '/Users/Paul1/Google Drive/Monkey/Algo/v2/1-2  -  Price CSVs'
        path_price_info = 'C:/Users/Paul Boehringer/Google Drive/Monkey/Algo/v2/1-2  -  Price CSVs'
        os.chdir(path_price_info)

        filename = ticker + '.csv'
        price_df = pd.read_csv(filename)
        price_df['dates'] = pd.to_datetime(price_df['dates'], format="%Y-%m-%d")
        del price_df['Unnamed: 0']
        price_df = price_df.sort_values(by = 'dates')
        price_df = price_df.reset_index()
        del price_df['index']
        return price_df
    except (KeyError, FileNotFoundError):
        pass

def mean_confidence_interval(data, confidence=0.95):
    a = 1.0 * np.array(data)
    n = len(a)
    m, se = np.mean(a), scipy.stats.sem(a)
    h = se * scipy.stats.t.ppf((1 + confidence) / 2., n-1)
    return m-h, m, m+h

def get_buy_sell_deltas(ticker, interval):
    ### GET TICKER DATA
    # print(ticker)
    form4  = get_form4(ticker)
    prices = get_prices(ticker)

    ### MAX DATE ALLOWED ON FORM 4 ... and dealing with transactions reported on weekends (very rare cases)
    max_form4_date = max(prices['dates'])
    max_form4_date = max_form4_date -  timedelta(days = interval+3) ### + 3 TO BE SURE NO DATES REQUESTED
THAT PRICE CANT PULLED FOR
    form4 = form4[form4['transact_date'] < max_form4_date]
    form4 = form4[form4['transact_date'].isin(list(prices['dates']))]  ### weekend exception
    form4_buy  = form4[form4['aqq_disp'] == 'A']
    form4_sell = form4[form4['aqq_disp'] == 'D']

    ### the most recent form 4 date that can be used is the most recent price date minus the interval length
    dates_buy  = form4_buy['transact_date']
    dates_sell = form4_sell['transact_date']
    date_counts_buy = dates_buy.value_counts().sort_index().values
    date_counts_sell = dates_sell.value_counts().sort_index().values


    ### GET PRICE INDICIES...
    indicies_buy  = prices[prices['dates'].isin(dates_buy)].index
    indicies_sell = prices[prices['dates'].isin(dates_sell)].index

    ### builds indexes to retreive with proper repitition to account for multiple transactions on a date
    transaction_indicies_buy  = []
    transaction_indicies_sell = []

    ### GET INDICIE LIST FOR TRANSACTIONS
    for index1 in range(len(date_counts_buy)):
        indicie_count_buy  = [indicies_buy[index1]]  * date_counts_buy[index1]
        transaction_indicies_buy.extend(indicie_count_buy)

    for index2 in range(len(date_counts_sell)):
        indicie_count_sell = [indicies_sell[index2]] * date_counts_sell[index2]
        transaction_indicies_sell.extend(indicie_count_sell)

    ### GET INDICIE LIST FOR TIME PERIOD OUT
    delta_indicies_buy  = [x+math.ceil(interval*0.68767) for x in transaction_indicies_buy]
    delta_indicies_sell = [x+math.ceil(interval*0.68767) for x in transaction_indicies_sell]

    ### GETTING PRICE CHANGES
    buy_prices_start = prices['closes'][transaction_indicies_buy].values
    buy_prices_end   = prices['closes'][delta_indicies_buy].values
    buy_deltas = (buy_prices_end - buy_prices_start)/buy_prices_start

    sell_prices_start = prices['closes'][transaction_indicies_sell].values
    sell_prices_end   = prices['closes'][delta_indicies_sell].values
    sell_deltas = (sell_prices_end - sell_prices_start)/sell_prices_start

    return buy_deltas, sell_deltas

%%time
all_buy_deltas = []
all_sell_deltas = []

index = 0

for ticker in stocks_v4['Ticker']:
    if index%100 == 0:
        print(ticker)
    index +=1

    try:
        buy_deltas, sell_deltas = get_buy_sell_deltas(ticker, 45)
    except KeyError:
        print('KeyError on' + ticker)

    all_buy_deltas.extend(buy_deltas)
    all_sell_deltas.extend(sell_deltas)
print("DONE")
```

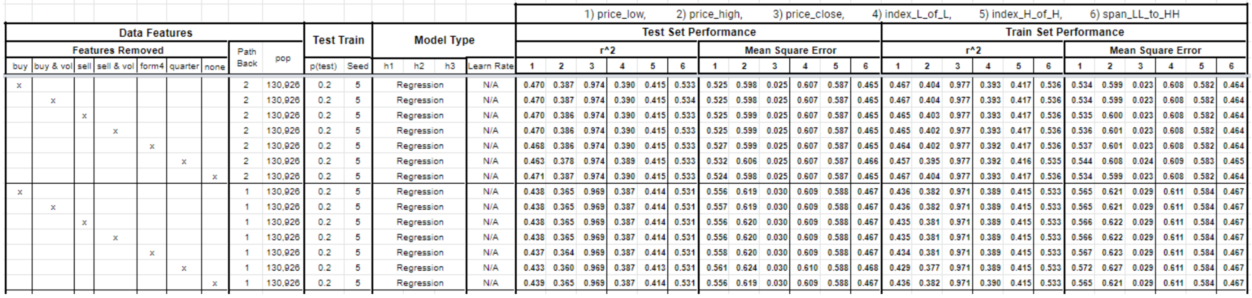
```
### TURNING DELTAS TO NUMPY
all_buy_deltas  = np.array(all_buy_deltas)
all_sell_deltas = np.array(all_sell_deltas)

### MASK TO GET RID OF NaNs
buy_keepers  = ~np.isnan(all_buy_deltas)
sell_keepers = ~np.isnan(all_sell_deltas)

### REMOVE NaNs
all_buy_deltas = all_buy_deltas[buy_keepers]
all_sell_deltas = all_sell_deltas[sell_keepers]

mean_confidence_interval(all_buy_deltas, confidence=0.95)
```

Appendix Item H

Results of Linear Regression Models



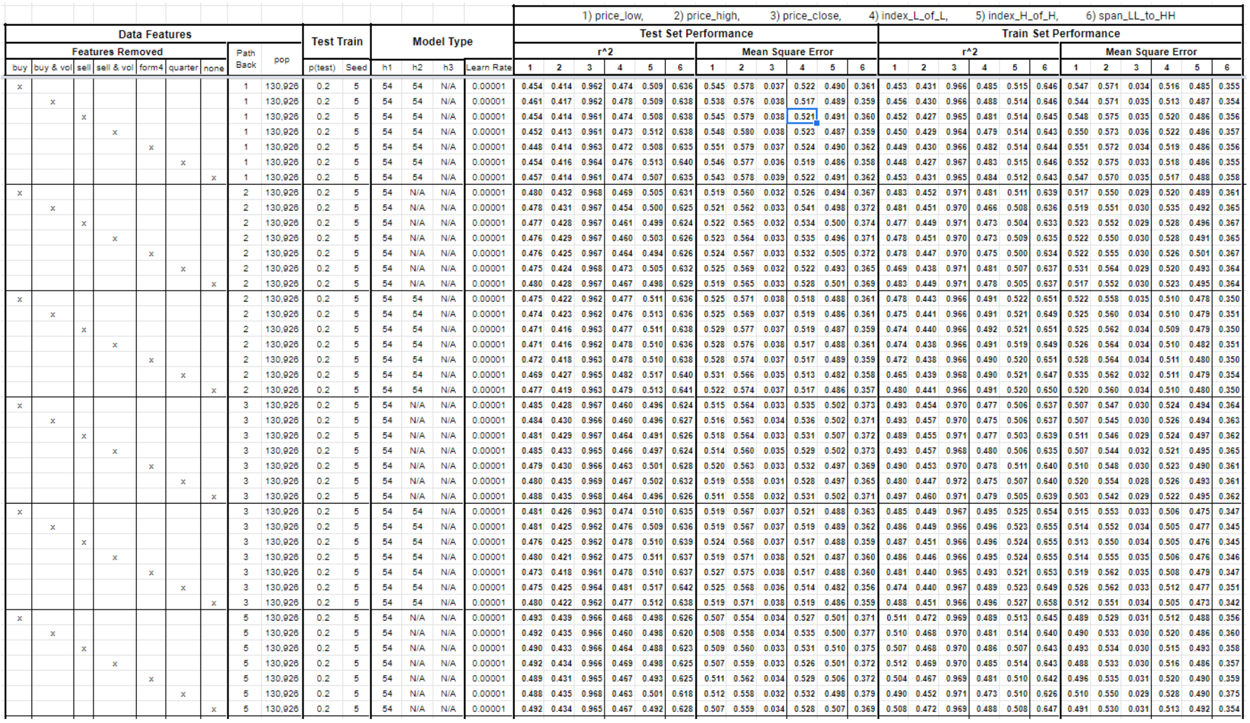
															1) price_low, 2) price_high, 3) price_close, 4) index_L_of_L, 5) index_H_of_H, 6) span_LL_to_HH																							
Data Features									Test Train		Model Type				Test Set Performance												Train Set Performance											
Features Removed							Path Back	pop							r^2						Mean Square Error						r^2						Mean Square Error					
buy	buy & vol	sell	sell & vol	form4	quarter	none			p(test)	Seed	h1	h2	h3	Learn Rate	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
x							8	120,855	0.2	5	Regression			N/A	0.504	0.407	0.978	0.394	0.417	0.537	0.492	0.578	0.022	0.603	0.585	0.462	0.502	0.428	0.980	0.398	0.422	0.540	0.499	0.575	0.020	0.602	0.577	0.460
	x						8	120,855	0.2	5	Regression			N/A	0.503	0.407	0.978	0.394	0.417	0.537	0.492	0.578	0.022	0.603	0.585	0.462	0.502	0.428	0.980	0.398	0.422	0.540	0.499	0.575	0.020	0.602	0.577	0.460
		x					8	120,855	0.2	5	Regression			N/A	0.503	0.406	0.978	0.394	0.417	0.536	0.492	0.580	0.022	0.603	0.585	0.462	0.500	0.426	0.980	0.399	0.422	0.540	0.501	0.577	0.020	0.602	0.577	0.460
			x				8	120,855	0.2	5	Regression			N/A	0.503	0.405	0.978	0.394	0.417	0.536	0.493	0.580	0.022	0.603	0.585	0.462	0.500	0.425	0.980	0.399	0.422	0.540	0.501	0.578	0.020	0.602	0.577	0.460
				x			8	120,855	0.2	5	Regression			N/A	0.501	0.403	0.978	0.393	0.417	0.536	0.494	0.582	0.022	0.603	0.585	0.462	0.499	0.424	0.980	0.398	0.422	0.540	0.502	0.579	0.020	0.602	0.577	0.460
					x		8	120,855	0.2	5	Regression			N/A	0.492	0.394	0.978	0.394	0.417	0.536	0.503	0.592	0.022	0.603	0.585	0.463	0.486	0.412	0.980	0.396	0.420	0.538	0.515	0.591	0.020	0.605	0.579	0.462
						x	8	120,855	0.2	5	Regression			N/A	0.504	0.408	0.978	0.394	0.417	0.537	0.491	0.577	0.022	0.603	0.585	0.462	0.503	0.429	0.980	0.399	0.422	0.540	0.498	0.574	0.020	0.602	0.577	0.460
x							7	120,855	0.2	5	Regression			N/A	0.500	0.406	0.978	0.393	0.417	0.536	0.495	0.580	0.022	0.603	0.585	0.462	0.499	0.426	0.980	0.398	0.421	0.540	0.502	0.577	0.020	0.602	0.578	0.460
	x						7	120,855	0.2	5	Regression			N/A	0.500	0.405	0.978	0.393	0.417	0.536	0.495	0.580	0.022	0.603	0.585	0.462	0.499	0.426	0.980	0.398	0.421	0.540	0.502	0.578	0.020	0.602	0.578	0.460
		x					7	120,855	0.2	5	Regression			N/A	0.500	0.404	0.978	0.393	0.417	0.536	0.496	0.581	0.022	0.603	0.585	0.462	0.497	0.424	0.980	0.398	0.421	0.540	0.504	0.579	0.020	0.602	0.578	0.460
			x				7	120,855	0.2	5	Regression			N/A	0.499	0.403	0.978	0.394	0.417	0.536	0.496	0.582	0.022	0.603	0.585	0.462	0.497	0.423	0.980	0.398	0.421	0.540	0.504	0.580	0.020	0.602	0.578	0.460
				x			7	120,855	0.2	5	Regression			N/A	0.498	0.402	0.978	0.393	0.417	0.536	0.497	0.584	0.022	0.603	0.585	0.462	0.496	0.422	0.980	0.398	0.421	0.540	0.505	0.581	0.020	0.603	0.578	0.460
					x		7	120,855	0.2	5	Regression			N/A	0.489	0.392	0.978	0.393	0.417	0.536	0.506	0.593	0.022	0.603	0.585	0.463	0.483	0.410	0.980	0.396	0.419	0.538	0.518	0.593	0.020	0.605	0.580	0.462
						x	7	120,855	0.2	5	Regression			N/A	0.501	0.407	0.978	0.393	0.417	0.536	0.494	0.579	0.022	0.603	0.585	0.462	0.499	0.426	0.980	0.398	0.422	0.540	0.501	0.577	0.020	0.602	0.578	0.460
x							6	120,855	0.2	5	Regression			N/A	0.498	0.405	0.978	0.393	0.417	0.536	0.497	0.581	0.022	0.604	0.585	0.463	0.496	0.424	0.980	0.397	0.421	0.539	0.505	0.579	0.020	0.604	0.578	0.461
	x						6	120,855	0.2	5	Regression			N/A	0.498	0.404	0.978	0.393	0.417	0.536	0.498	0.581	0.022	0.603	0.585	0.463	0.496	0.424	0.980	0.397	0.421	0.539	0.505	0.579	0.020	0.604	0.579	0.461
		x					6	120,855	0.2	5	Regression			N/A	0.497	0.403	0.978	0.393	0.417	0.536	0.498	0.582	0.022	0.603	0.585	0.463	0.494	0.422	0.980	0.397	0.421	0.539	0.507	0.581	0.020	0.604	0.578	0.461
			x				6	120,855	0.2	5	Regression			N/A	0.497	0.402	0.978	0.393	0.417	0.536	0.498	0.583	0.022	0.603	0.585	0.463	0.494	0.422	0.980	0.397	0.421	0.539	0.507	0.581	0.020	0.604	0.578	0.461
				x			6	120,855	0.2	5	Regression			N/A	0.495	0.401	0.978	0.393	0.417	0.536	0.500	0.585	0.022	0.603	0.585	0.463	0.493	0.420	0.980	0.397	0.420	0.539	0.508	0.583	0.020	0.604	0.579	0.461
					x		6	120,855	0.2	5	Regression			N/A	0.487	0.391	0.978	0.393	0.417	0.535	0.508	0.595	0.022	0.604	0.585	0.463	0.481	0.409	0.980	0.395	0.419	0.537	0.520	0.594	0.020	0.606	0.581	0.463
						x	6	120,855	0.2	5	Regression			N/A	0.498	0.405	0.978	0.393	0.417	0.536	0.497	0.580	0.022	0.603	0.585	0.463	0.496	0.425	0.980	0.397	0.421	0.539	0.504	0.578	0.020	0.603	0.578	0.461
x							5	130,926	0.2	5	Regression			N/A	0.495	0.403	0.978	0.392	0.417	0.536	0.500	0.582	0.022	0.605	0.585	0.463	0.492	0.423	0.980	0.396	0.420	0.538	0.508	0.580	0.020	0.605	0.579	0.462
	x						5	130,926	0.2	5	Regression			N/A	0.495	0.403	0.978	0.392	0.417	0.536	0.501	0.583	0.022	0.605	0.585	0.463	0.492	0.422	0.980	0.395	0.420	0.538	0.509	0.581	0.020	0.605	0.579	0.462
		x					5	130,926	0.2	5	Regression			N/A	0.495	0.402	0.978	0.392	0.417	0.536	0.501	0.584	0.022	0.604	0.585	0.463	0.491	0.420	0.980	0.396	0.420	0.538	0.510	0.583	0.020	0.605	0.579	0.462
			x				5	130,926	0.2	5	Regression			N/A	0.494	0.401	0.978	0.392	0.417	0.536	0.501	0.584	0.022	0.604	0.585	0.463	0.490	0.420	0.980	0.396	0.420	0.538	0.510	0.583	0.020	0.605	0.579	0.462
				x			5	130,926	0.2	5	Regression			N/A	0.493	0.400	0.978	0.392	0.417	0.536	0.503	0.586	0.022	0.605	0.585	0.463	0.489	0.419	0.980	0.395	0.420	0.538	0.512	0.584	0.020	0.605	0.579	0.462
					x		5	130,926	0.2	5	Regression			N/A	0.484	0.389	0.978	0.392	0.417	0.535	0.511	0.596	0.022	0.605	0.585	0.464	0.478	0.408	0.980	0.394	0.418	0.537	0.523	0.595	0.020	0.607	0.581	0.463
						x	5	130,926	0.2	5	Regression			N/A	0.496	0.404	0.978	0.392	0.417	0.536	0.500	0.582	0.022	0.604	0.585	0.463	0.493	0.423	0.980	0.396	0.420	0.538	0.508	0.580	0.020	0.605	0.579	0.462
x							4	130,926	0.2	5	Regression			N/A	0.493	0.400	0.977	0.391	0.417	0.535	0.503	0.585	0.022	0.605	0.585	0.463	0.489	0.418	0.979	0.394	0.419	0.538	0.512	0.585	0.021	0.606	0.580	0.462
	x						4	130,926	0.2	5	Regression			N/A	0.492	0.400	0.977	0.391	0.417	0.535	0.503	0.585	0.022	0.605	0.585	0.463	0.489	0.418	0.979	0.394	0.419	0.538	0.512	0.585	0.021	0.606	0.580	0.462
		x					4	130,926	0.2	5	Regression			N/A	0.493	0.399	0.977	0.391	0.417	0.535	0.503	0.586	0.022	0.605	0.585	0.463	0.488	0.416	0.979	0.395	0.419	0.538	0.513	0.587	0.021	0.606	0.580	0.462
			x				4	130,926	0.2	5	Regression			N/A	0.492	0.398	0.977	0.391	0.417	0.535	0.503	0.587	0.022	0.605	0.585	0.463	0.487	0.416	0.979	0.395	0.419	0.538	0.513	0.587	0.021	0.606	0.580	0.462
				x			4	130,926	0.2	5	Regression			N/A	0.491	0.397	0.977	0.391	0.417	0.535	0.505	0.588	0.022	0.605	0.585	0.463	0.486	0.415	0.979	0.394	0.419	0.538	0.514	0.588	0.021	0.606	0.580	0.462
					x		4	130,926	0.2	5	Regression			N/A	0.483	0.387	0.977	0.391	0.416	0.534	0.512	0.598	0.022	0.605	0.586	0.464	0.477	0.405	0.979	0.393	0.418	0.536	0.524	0.598	0.021	0.608	0.581	0.464
						x	4	130,926	0.2	5	Regression			N/A	0.494	0.401	0.977	0.391	0.417	0.535	0.502	0.584	0.022	0.605	0.585	0.463	0.490	0.418	0.979	0.395	0.419	0.538	0.511	0.585	0.021	0.606	0.580	0.462
x							3	130,926	0.2	5	Regression			N/A	0.484	0.393	0.976	0.390	0.416	0.535	0.511	0.592	0.023	0.606	0.585	0.464	0.480	0.412	0.978	0.393	0.418	0.537	0.521	0.591	0.022	0.607	0.581	0.463
	x						3	130,926	0.2	5	Regression			N/A	0.484	0.393	0.976	0.390	0.416	0.535	0.512	0.592	0.023	0.606	0.585	0.464	0.480	0.412	0.978	0.393	0.418	0.536	0.521	0.591	0.022	0.607	0.581	0.463
		x					3	130,926	0.2	5	Regression			N/A	0.484	0.392	0.976	0.391	0.416	0.535	0.511	0.593	0.023	0.606	0.585	0.464	0.479	0.411	0.978	0.394	0.418	0.537	0.522	0.592	0.022	0.607	0.581	0.463
			x				3	130,926	0.2	5	Regression			N/A	0.484	0.392	0.976	0.391	0.416	0.535	0.511	0.593	0.023	0.606	0.585	0.464	0.479	0.410	0.978	0.394	0.418	0.537	0.522	0.593	0.022	0.607	0.581	0.463
				x			3	130,926	0.2	5	Regression			N/A	0.482	0.391	0.976	0.390	0.416	0.535	0.513	0.594	0.023	0.606	0.586	0.464	0.478	0.410	0.978	0.393	0.418	0.536	0.523	0.593	0.022	0.607	0.581	0.464
					x		3	130,926	0.2	5	Regression			N/A	0.475	0.383	0.976	0.390	0.416	0.534	0.520	0.602	0.023	0.606	0.586	0.465	0.469	0.402	0.978	0.392	0.417	0.535	0.531	0.602	0.022	0.608	0.582	0.465
						x	3	130,926	0.2	5	Regression			N/A	0.485	0.394	0.976	0.391	0.416	0.535	0.511	0.591	0.023	0.606	0.585	0.464	0.481	0.413	0.978	0.394	0.418	0.537	0.520	0.590	0.022	0.607	0.581	0.463



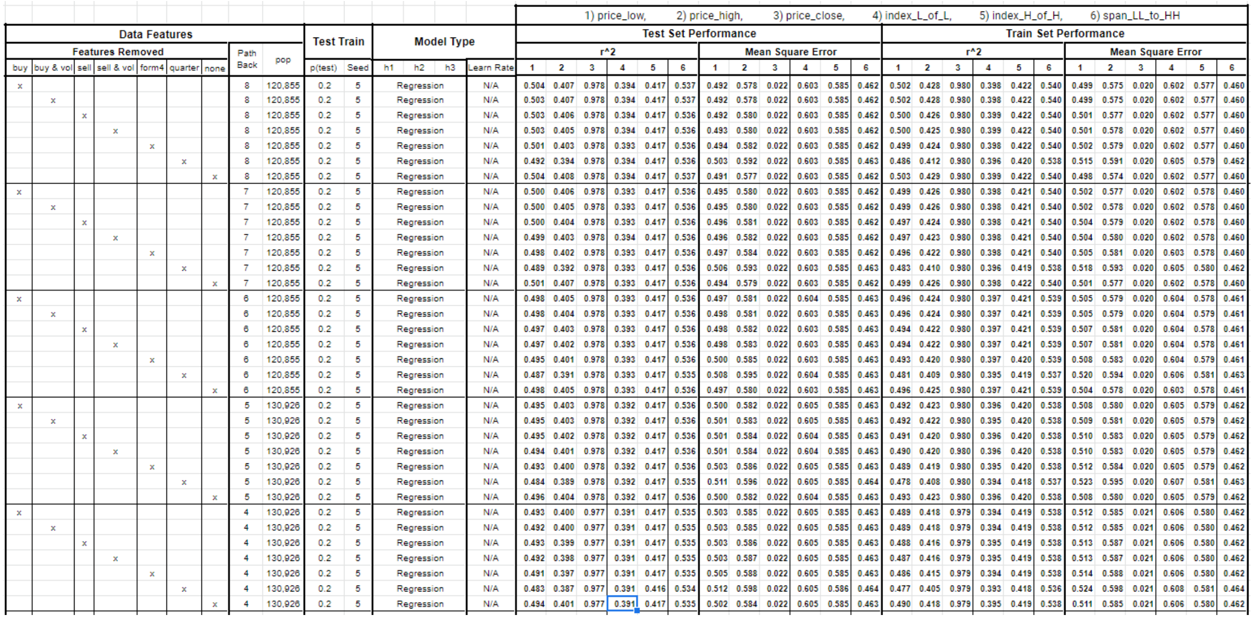
															1) price_low, 2) price_high, 3) price_close, 4) index_L_of_L, 5) index_H_of_H, 6) span_LL_to_HH																							
Data Features									Test Train		Model Type				Test Set Performance												Train Set Performance											
Features Removed							Path Back	pop							r^2						Mean Square Error						r^2						Mean Square Error					
buy	buy & vol	sell	sell & vol	form4	quarter	none			p(test)	Seed	h1	h2	h3	Learn Rate	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
x							2	130,926	0.2	5	Regression			N/A	0.470	0.387	0.974	0.390	0.415	0.533	0.525	0.598	0.025	0.607	0.587	0.465	0.467	0.404	0.977	0.393	0.417	0.536	0.534	0.599	0.023	0.608	0.582	0.464
	x						2	130,926	0.2	5	Regression			N/A	0.470	0.387	0.974	0.390	0.415	0.534	0.525	0.598	0.025	0.607	0.587	0.465	0.467	0.404	0.977	0.393	0.417	0.536	0.534	0.599	0.023	0.608	0.582	0.464
		x					2	130,926	0.2	5	Regression			N/A	0.470	0.386	0.974	0.390	0.415	0.533	0.525	0.599	0.025	0.607	0.587	0.465	0.465	0.403	0.977	0.393	0.417	0.536	0.535	0.600	0.023	0.608	0.582	0.464
			x				2	130,926	0.2	5	Regression			N/A	0.470	0.386	0.974	0.390	0.415	0.533	0.525	0.599	0.025	0.607	0.587	0.465	0.465	0.402	0.977	0.393	0.417	0.536	0.536	0.601	0.023	0.608	0.582	0.464
				x			2	130,926	0.2	5	Regression			N/A	0.468	0.386	0.974	0.390	0.415	0.533	0.527	0.599	0.025	0.607	0.587	0.465	0.464	0.402	0.977	0.392	0.417	0.536	0.537	0.601	0.023	0.608	0.582	0.464
					x		2	130,926	0.2	5	Regression			N/A	0.463	0.378	0.974	0.389	0.415	0.533	0.532	0.606	0.025	0.607	0.587	0.466	0.457	0.395	0.977	0.392	0.416	0.535	0.544	0.608	0.024	0.609	0.583	0.465
						x	2	130,926	0.2	5	Regression			N/A	0.471	0.387	0.974	0.390	0.415	0.533	0.524	0.598	0.025	0.607	0.587	0.465	0.467	0.404	0.977	0.393	0.417	0.536	0.534	0.599	0.023	0.608	0.582	0.464
x							1	130,926	0.2	5	Regression			N/A	0.438	0.365	0.969	0.387	0.414	0.531	0.556	0.619	0.030	0.609	0.588	0.467	0.436	0.382	0.971	0.389	0.415	0.533	0.565	0.621	0.029	0.611	0.584	0.467
	x						1	130,926	0.2	5	Regression			N/A	0.438	0.365	0.969	0.387	0.414	0.531	0.557	0.619	0.030	0.609	0.588	0.467	0.436	0.382	0.971	0.389	0.415	0.533	0.565	0.621	0.029	0.611	0.584	0.467
		x					1	130,926	0.2	5	Regression			N/A	0.438	0.365	0.969	0.387	0.414	0.531	0.556	0.620	0.030	0.609	0.588	0.467	0.435	0.381	0.971	0.389	0.415	0.533	0.566	0.622	0.029	0.611	0.584	0.467
			x				1	130,926	0.2	5	Regression			N/A	0.438	0.365	0.969	0.387	0.414	0.531	0.556	0.620	0.030	0.609	0.588	0.467	0.435	0.381	0.971	0.389	0.415	0.533	0.566	0.622	0.029	0.611	0.584	0.467
				x			1	130,926	0.2	5	Regression			N/A	0.437	0.364	0.969	0.387	0.414	0.531	0.558	0.620	0.030	0.609	0.588	0.467	0.434	0.381	0.971	0.389	0.415	0.533	0.567	0.623	0.029	0.611	0.584	0.467
					x		1	130,926	0.2	5	Regression			N/A	0.433	0.360	0.969	0.387	0.413	0.531	0.561	0.624	0.030	0.610	0.588	0.468	0.429	0.377	0.971	0.389	0.415	0.533	0.572	0.627	0.029	0.611	0.584	0.467
						x	1	130,926	0.2	5	Regression			N/A	0.439	0.365	0.969	0.387	0.414	0.531	0.556	0.619	0.030	0.609	0.588	0.467	0.436	0.382	0.971	0.390	0.415	0.533	0.565	0.621	0.029	0.611	0.584	0.467

Appendix Item I

Results of MLP Network Models



															1) price_low, 2) price_high, 3) price_close, 4) index_L_of_L, 5) index_H_of_H, 6) span_LL_to_HH																							
Data Features									Test Train		Model Type				Test Set Performance												Train Set Performance											
Features Removed							Path Back	pop							r^2						Mean Square Error						r^2						Mean Square Error					
buy	buy & vol	sell	sell & vol	form4	quarter	none			p(test)	Seed	h1	h2	h3	Learn Rate	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
x							1	130,926	0.2	5	54	54	N/A	0.00001	0.454	0.414	0.962	0.474	0.509	0.636	0.545	0.578	0.037	0.522	0.490	0.361	0.453	0.431	0.966	0.485	0.515	0.646	0.547	0.571	0.034	0.516	0.485	0.355
	x						1	130,926	0.2	5	54	54	N/A	0.00001	0.461	0.417	0.962	0.478	0.509	0.638	0.538	0.576	0.038	0.517	0.489	0.359	0.456	0.430	0.966	0.488	0.514	0.646	0.544	0.571	0.035	0.513	0.487	0.354
		x					1	130,926	0.2	5	54	54	N/A	0.00001	0.454	0.414	0.961	0.474	0.508	0.638	0.545	0.579	0.038	0.521	0.491	0.360	0.452	0.427	0.965	0.481	0.514	0.645	0.548	0.575	0.035	0.520	0.486	0.356
			x				1	130,926	0.2	5	54	54	N/A	0.00001	0.452	0.413	0.961	0.473	0.512	0.638	0.548	0.580	0.038	0.523	0.487	0.359	0.450	0.429	0.964	0.479	0.514	0.643	0.550	0.573	0.036	0.522	0.486	0.357
				x			1	130,926	0.2	5	54	54	N/A	0.00001	0.448	0.414	0.963	0.472	0.508	0.635	0.551	0.579	0.037	0.524	0.490	0.362	0.449	0.430	0.966	0.482	0.514	0.644	0.551	0.572	0.034	0.519	0.486	0.356
					x		1	130,926	0.2	5	54	54	N/A	0.00001	0.454	0.416	0.964	0.476	0.513	0.640	0.546	0.577	0.036	0.519	0.486	0.358	0.448	0.427	0.967	0.483	0.515	0.646	0.552	0.575	0.033	0.518	0.486	0.355
						x	1	130,926	0.2	5	54	54	N/A	0.00001	0.457	0.414	0.961	0.474	0.507	0.635	0.543	0.578	0.039	0.522	0.491	0.362	0.453	0.431	0.965	0.484	0.512	0.643	0.547	0.570	0.035	0.517	0.488	0.358
x							2	130,926	0.2	5	54	N/A	N/A	0.00001	0.480	0.432	0.968	0.469	0.505	0.631	0.519	0.560	0.032	0.526	0.494	0.367	0.483	0.452	0.971	0.481	0.511	0.639	0.517	0.550	0.029	0.520	0.489	0.361
	x						2	130,926	0.2	5	54	N/A	N/A	0.00001	0.478	0.431	0.967	0.454	0.500	0.625	0.521	0.562	0.033	0.541	0.498	0.372	0.481	0.451	0.970	0.466	0.508	0.636	0.519	0.551	0.030	0.535	0.492	0.365
		x					2	130,926	0.2	5	54	N/A	N/A	0.00001	0.477	0.428	0.967	0.461	0.499	0.624	0.522	0.565	0.032	0.534	0.500	0.374	0.477	0.449	0.971	0.473	0.504	0.633	0.523	0.552	0.029	0.528	0.496	0.367
			x				2	130,926	0.2	5	54	N/A	N/A	0.00001	0.476	0.429	0.967	0.460	0.503	0.626	0.523	0.564	0.033	0.535	0.496	0.371	0.478	0.451	0.970	0.473	0.509	0.635	0.522	0.550	0.030	0.528	0.491	0.365
				x			2	130,926	0.2	5	54	N/A	N/A	0.00001	0.476	0.425	0.967	0.464	0.494	0.626	0.524	0.567	0.033	0.532	0.505	0.372	0.478	0.447	0.970	0.475	0.500	0.634	0.522	0.555	0.030	0.526	0.501	0.367
					x		2	130,926	0.2	5	54	N/A	N/A	0.00001	0.475	0.424	0.968	0.473	0.505	0.632	0.525	0.569	0.032	0.522	0.493	0.365	0.469	0.438	0.971	0.481	0.507	0.637	0.531	0.564	0.029	0.520	0.493	0.364
						x	2	130,926	0.2	5	54	N/A	N/A	0.00001	0.480	0.428	0.967	0.467	0.498	0.629	0.519	0.565	0.033	0.528	0.501	0.369	0.483	0.449	0.971	0.478	0.505	0.637	0.517	0.552	0.030	0.523	0.495	0.364
x							2	130,926	0.2	5	54	54	N/A	0.00001	0.475	0.422	0.962	0.477	0.511	0.636	0.525	0.571	0.038	0.518	0.488	0.361	0.478	0.443	0.966	0.491	0.522	0.651	0.522	0.558	0.035	0.510	0.478	0.350
	x						2	130,926	0.2	5	54	54	N/A	0.00001	0.474	0.423	0.962	0.476	0.513	0.636	0.525	0.569	0.037	0.519	0.486	0.361	0.475	0.441	0.966	0.491	0.521	0.649	0.525	0.560	0.034	0.510	0.479	0.351
		x					2	130,926	0.2	5	54	54	N/A	0.00001	0.471	0.416	0.963	0.477	0.511	0.638	0.529	0.577	0.037	0.519	0.487	0.359	0.474	0.440	0.966	0.492	0.521	0.651	0.525	0.562	0.034	0.509	0.479	0.350
			x				2	130,926	0.2	5	54	54	N/A	0.00001	0.471	0.416	0.962	0.478	0.510	0.636	0.528	0.576	0.038	0.517	0.488	0.361	0.474	0.438	0.966	0.491	0.519	0.649	0.526	0.564	0.034	0.510	0.482	0.351
				x			2	130,926	0.2	5	54	54	N/A	0.00001	0.472	0.418	0.963	0.478	0.510	0.638	0.528	0.574	0.037	0.517	0.489	0.359	0.472	0.438	0.966	0.490	0.520	0.651	0.528	0.564	0.034	0.511	0.480	0.350
					x		2	130,926	0.2	5	54	54	N/A	0.00001	0.469	0.427	0.965	0.482	0.517	0.640	0.531	0.566	0.035	0.513	0.482	0.358	0.465	0.439	0.968	0.490	0.521	0.647	0.535	0.562	0.032	0.511	0.479	0.354
						x	2	130,926	0.2	5	54	54	N/A	0.00001	0.477	0.419	0.963	0.479	0.513	0.641	0.522	0.574	0.037	0.517	0.486	0.357	0.480	0.441	0.966	0.491	0.520	0.650	0.520	0.560	0.034	0.510	0.480	0.350
x							3	130,926	0.2	5	54	N/A	N/A	0.00001	0.485	0.428	0.967	0.460	0.496	0.624	0.515	0.564	0.033	0.535	0.502	0.373	0.493	0.454	0.970	0.477	0.506	0.637	0.507	0.547	0.030	0.524	0.494	0.364
	x						3	130,926	0.2	5	54	N/A	N/A	0.00001	0.484	0.430	0.966	0.460	0.496	0.627	0.516	0.563	0.034	0.536	0.502	0.371	0.493	0.457	0.970	0.475	0.506	0.637	0.507	0.545	0.030	0.526	0.494	0.363
		x					3	130,926	0.2	5	54	N/A	N/A	0.00001	0.481	0.429	0.967	0.464	0.491	0.626	0.518	0.564	0.033	0.531	0.507	0.372	0.489	0.455	0.971	0.477	0.503	0.639	0.511	0.546	0.029	0.524	0.497	0.362
			x				3	130,926	0.2	5	54	N/A	N/A	0.00001	0.485	0.433	0.965	0.466	0.497	0.624	0.514	0.560	0.035	0.529	0.502	0.373	0.493	0.457	0.968	0.480	0.506	0.635	0.507	0.544	0.032	0.521	0.495	0.365
				x			3	130,926	0.2	5	54	N/A	N/A	0.00001	0.479	0.430	0.966	0.463	0.501	0.628	0.520	0.563	0.033	0.532	0.497	0.369	0.490	0.453	0.970	0.478	0.511	0.640	0.510	0.548	0.030	0.523	0.490	0.361
					x		3	130,926	0.2	5	54	N/A	N/A	0.00001	0.480	0.435	0.969	0.467	0.502	0.632	0.519	0.558	0.031	0.528	0.497	0.365	0.480	0.447	0.972	0.475	0.507	0.640	0.520	0.554	0.028	0.526	0.493	0.361
						x	3	130,926	0.2	5	54	N/A	N/A	0.00001	0.488	0.435	0.968	0.464	0.496	0.626	0.511	0.558	0.032	0.531	0.502	0.371	0.497	0.460	0.971	0.479	0.505	0.639	0.503	0.542	0.029	0.522	0.495	0.362
x							3	130,926	0.2	5	54	54	N/A	0.00001	0.481	0.426	0.963	0.474	0.510	0.635	0.519	0.567	0.037	0.521	0.488	0.363	0.485	0.449	0.967	0.495	0.525	0.654	0.515	0.553	0.033	0.506	0.475	0.347
	x						3	130,926	0.2	5	54	54	N/A	0.00001	0.481	0.425	0.962	0.476	0.509	0.636	0.519	0.567	0.037	0.519	0.489	0.362	0.486	0.449	0.966	0.496	0.523	0.655	0.514	0.552	0.034	0.505	0.477	0.345
		x					3	130,926	0.2	5	54	54	N/A	0.00001	0.476	0.425	0.962	0.478	0.510	0.639	0.524	0.568	0.037	0.517	0.488	0.359	0.487	0.451	0.966	0.496	0.524	0.655	0.513	0.550	0.034	0.505	0.476	0.345
			x				3	130,926	0.2	5	54	54	N/A	0.00001	0.480	0.421	0.962	0.475	0.511	0.637	0.519	0.571	0.038	0.521	0.487	0.360	0.486	0.446	0.966	0.495	0.524	0.655	0.514	0.555	0.035	0.506	0.476	0.346
				x			3	130,926	0.2	5	54	54	N/A	0.00001	0.473	0.418	0.961	0.478	0.510	0.637	0.527	0.575	0.038	0.517	0.488	0.360	0.481	0.440	0.965	0.493	0.521	0.653	0.519	0.562	0.035	0.508	0.479	0.347
					x		3	130,926	0.2	5	54	54	N/A	0.00001	0.475	0.425	0.964	0.481	0.517	0.642	0.525	0.568	0.036	0.514	0.482	0.356	0.474	0.440	0.967	0.489	0.523	0.649	0.526	0.562	0.033	0.512	0.477	0.351
						x	3	130,926	0.2	5	54	54	N/A	0.00001	0.480	0.422	0.962	0.477	0.512	0.638	0.519	0.571	0.038	0.519	0.486	0.359	0.488	0.451	0.966	0.496	0.527	0.658	0.512	0.551	0.034	0.505	0.473	0.342
x							5	130,926	0.2	5	54	N/A	N/A	0.00001	0.493	0.439	0.966	0.468	0.498	0.626	0.507	0.554	0.034	0.527	0.501	0.371	0.511	0.472	0.969	0.489	0.513	0.645	0.489	0.529	0.031	0.512	0.488	0.356
	x						5	130,926	0.2	5	54	N/A	N/A	0.00001	0.492	0.435	0.966	0.460	0.498	0.620	0.508	0.558	0.034	0.535	0.500	0.377	0.510	0.468	0.970	0.481	0.514	0.640	0.490	0.533	0.030	0.520	0.486	0.360
		x					5	130,926	0.2	5	54	N/A	N/A	0.00001	0.490	0.433	0.966	0.464	0.488	0.623	0.509	0.560	0.033	0.531	0.510	0.375	0.507	0.468	0.970	0.486	0.507	0.643	0.493	0.534	0.030	0.515	0.493	0.358
			x				5	130,926	0.2	5	54	N/A	N/A	0.00001	0.492	0.434	0.966	0.469	0.498	0.625	0.507	0.559	0.033	0.526	0.501	0.372	0.512	0.469	0.970	0.485	0.514	0.643	0.488	0.533	0.030	0.516	0.486	0.357
				x			5	130,926	0.2	5	54	N/A	N/A	0.00001	0.489	0.431	0.965	0.467	0.493	0.625	0.511	0.562	0.034	0.529	0.506	0.372	0.504	0.467	0.969	0.481	0.510	0.642	0.496	0.535	0.031	0.520	0.490	0.359
					x		5	130,926	0.2	5	54	N/A	N/A	0.00001	0.488	0.435	0.968	0.463	0.501	0.618	0.512	0.558	0.032	0.532	0.498	0.379	0.490	0.452	0.971	0.473	0.510	0.626	0.510	0.550	0.029	0.528	0.490	0.375
						x	5	130,926	0.2	5	54	N/A	N/A	0.00001	0.492	0.434	0.965	0.467	0.492	0.628	0.507	0.559	0.034	0.528	0.507	0.369	0.508	0.472	0.969	0.488	0.508	0.647	0.491	0.530	0.031	0.513	0.492	0.354



															1) price_low, 2) price_high, 3) price_close, 4) index_L_of_L, 5) index_H_of_H, 6) span_LL_to_HH																							
Data Features									Test Train		Model Type				Test Set Performance												Train Set Performance											
Features Removed							Path Back	pop							r^2						Mean Square Error						r^2						Mean Square Error					
buy	buy & vol	sell	sell & vol	form4	quarter	none			p(test)	Seed	h1	h2	h3	Learn Rate	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
x							8	120,855	0.2	5	Regression			N/A	0.504	0.407	0.978	0.394	0.417	0.537	0.492	0.578	0.022	0.603	0.585	0.462	0.502	0.428	0.980	0.398	0.422	0.540	0.499	0.575	0.020	0.602	0.577	0.460
	x						8	120,855	0.2	5	Regression			N/A	0.503	0.407	0.978	0.394	0.417	0.537	0.492	0.578	0.022	0.603	0.585	0.462	0.502	0.428	0.980	0.398	0.422	0.540	0.499	0.575	0.020	0.602	0.577	0.460
		x					8	120,855	0.2	5	Regression			N/A	0.503	0.406	0.978	0.394	0.417	0.536	0.492	0.580	0.022	0.603	0.585	0.462	0.500	0.426	0.980	0.399	0.422	0.540	0.501	0.577	0.020	0.602	0.577	0.460
			x				8	120,855	0.2	5	Regression			N/A	0.503	0.405	0.978	0.394	0.417	0.536	0.493	0.580	0.022	0.603	0.585	0.462	0.500	0.425	0.980	0.399	0.422	0.540	0.501	0.578	0.020	0.602	0.577	0.460
				x			8	120,855	0.2	5	Regression			N/A	0.501	0.403	0.978	0.393	0.417	0.536	0.494	0.582	0.022	0.603	0.585	0.462	0.499	0.424	0.980	0.398	0.422	0.540	0.502	0.579	0.020	0.602	0.577	0.460
					x		8	120,855	0.2	5	Regression			N/A	0.492	0.394	0.978	0.394	0.417	0.536	0.503	0.592	0.022	0.603	0.585	0.463	0.486	0.412	0.980	0.396	0.420	0.538	0.515	0.591	0.020	0.605	0.579	0.462
						x	8	120,855	0.2	5	Regression			N/A	0.504	0.408	0.978	0.394	0.417	0.537	0.491	0.577	0.022	0.603	0.585	0.462	0.503	0.429	0.980	0.399	0.422	0.540	0.498	0.574	0.020	0.602	0.577	0.460
x							7	120,855	0.2	5	Regression			N/A	0.500	0.406	0.978	0.393	0.417	0.536	0.495	0.580	0.022	0.603	0.585	0.462	0.499	0.426	0.980	0.398	0.421	0.540	0.502	0.577	0.020	0.602	0.578	0.460
	x						7	120,855	0.2	5	Regression			N/A	0.500	0.405	0.978	0.393	0.417	0.536	0.495	0.580	0.022	0.603	0.585	0.462	0.499	0.426	0.980	0.398	0.421	0.540	0.502	0.578	0.020	0.602	0.578	0.460
		x					7	120,855	0.2	5	Regression			N/A	0.500	0.404	0.978	0.393	0.417	0.536	0.496	0.581	0.022	0.603	0.585	0.462	0.497	0.424	0.980	0.398	0.421	0.540	0.504	0.579	0.020	0.602	0.578	0.460
			x				7	120,855	0.2	5	Regression			N/A	0.499	0.403	0.978	0.394	0.417	0.536	0.496	0.582	0.022	0.603	0.585	0.462	0.497	0.423	0.980	0.398	0.421	0.540	0.504	0.580	0.020	0.602	0.578	0.460
				x			7	120,855	0.2	5	Regression			N/A	0.498	0.402	0.978	0.393	0.417	0.536	0.497	0.584	0.022	0.603	0.585	0.462	0.496	0.422	0.980	0.398	0.421	0.540	0.505	0.581	0.020	0.603	0.578	0.460
					x		7	120,855	0.2	5	Regression			N/A	0.489	0.392	0.978	0.393	0.417	0.536	0.506	0.593	0.022	0.603	0.585	0.463	0.483	0.410	0.980	0.396	0.419	0.538	0.518	0.593	0.020	0.605	0.580	0.462
						x	7	120,855	0.2	5	Regression			N/A	0.501	0.407	0.978	0.393	0.417	0.536	0.494	0.579	0.022	0.603	0.585	0.462	0.499	0.426	0.980	0.398	0.422	0.540	0.501	0.577	0.020	0.602	0.578	0.460
x							6	120,855	0.2	5	Regression			N/A	0.498	0.405	0.978	0.393	0.417	0.536	0.497	0.581	0.022	0.604	0.585	0.463	0.496	0.424	0.980	0.397	0.421	0.539	0.505	0.579	0.020	0.604	0.578	0.461
	x						6	120,855	0.2	5	Regression			N/A	0.498	0.404	0.978	0.393	0.417	0.536	0.498	0.581	0.022	0.603	0.585	0.463	0.496	0.424	0.980	0.397	0.421	0.539	0.505	0.579	0.020	0.604	0.579	0.461
		x					6	120,855	0.2	5	Regression			N/A	0.497	0.403	0.978	0.393	0.417	0.536	0.498	0.582	0.022	0.603	0.585	0.463	0.494	0.422	0.980	0.397	0.421	0.539	0.507	0.581	0.020	0.604	0.578	0.461
			x				6	120,855	0.2	5	Regression			N/A	0.497	0.402	0.978	0.393	0.417	0.536	0.498	0.583	0.022	0.603	0.585	0.463	0.494	0.422	0.980	0.397	0.421	0.539	0.507	0.581	0.020	0.604	0.578	0.461
				x			6	120,855	0.2	5	Regression			N/A	0.495	0.401	0.978	0.393	0.417	0.536	0.500	0.585	0.022	0.603	0.585	0.463	0.493	0.420	0.980	0.397	0.420	0.539	0.508	0.583	0.020	0.604	0.579	0.461
					x		6	120,855	0.2	5	Regression			N/A	0.487	0.391	0.978	0.393	0.417	0.535	0.508	0.595	0.022	0.604	0.585	0.463	0.481	0.409	0.980	0.395	0.419	0.537	0.520	0.594	0.020	0.606	0.581	0.463
						x	6	120,855	0.2	5	Regression			N/A	0.498	0.405	0.978	0.393	0.417	0.536	0.497	0.580	0.022	0.603	0.585	0.463	0.496	0.425	0.980	0.397	0.421	0.539	0.504	0.578	0.020	0.603	0.578	0.461
x							5	130,926	0.2	5	Regression			N/A	0.495	0.403	0.978	0.392	0.417	0.536	0.500	0.582	0.022	0.605	0.585	0.463	0.492	0.423	0.980	0.396	0.420	0.538	0.508	0.580	0.020	0.605	0.579	0.462
	x						5	130,926	0.2	5	Regression			N/A	0.495	0.403	0.978	0.392	0.417	0.536	0.501	0.583	0.022	0.605	0.585	0.463	0.492	0.422	0.980	0.395	0.420	0.538	0.509	0.581	0.020	0.605	0.579	0.462
		x					5	130,926	0.2	5	Regression			N/A	0.495	0.402	0.978	0.392	0.417	0.536	0.501	0.584	0.022	0.604	0.585	0.463	0.491	0.420	0.980	0.396	0.420	0.538	0.510	0.583	0.020	0.605	0.579	0.462
			x				5	130,926	0.2	5	Regression			N/A	0.494	0.401	0.978	0.392	0.417	0.536	0.501	0.584	0.022	0.604	0.585	0.463	0.490	0.420	0.980	0.396	0.420	0.538	0.510	0.583	0.020	0.605	0.579	0.462
				x			5	130,926	0.2	5	Regression			N/A	0.493	0.400	0.978	0.392	0.417	0.536	0.503	0.586	0.022	0.605	0.585	0.463	0.489	0.419	0.980	0.395	0.420	0.538	0.512	0.584	0.020	0.605	0.579	0.462
					x		5	130,926	0.2	5	Regression			N/A	0.484	0.389	0.978	0.392	0.417	0.535	0.511	0.596	0.022	0.605	0.585	0.464	0.478	0.408	0.980	0.394	0.418	0.537	0.523	0.595	0.020	0.607	0.581	0.463
						x	5	130,926	0.2	5	Regression			N/A	0.496	0.404	0.978	0.392	0.417	0.536	0.500	0.582	0.022	0.604	0.585	0.463	0.493	0.423	0.980	0.396	0.420	0.538	0.508	0.580	0.020	0.605	0.579	0.462
x							4	130,926	0.2	5	Regression			N/A	0.493	0.400	0.977	0.391	0.417	0.535	0.503	0.585	0.022	0.605	0.585	0.463	0.489	0.418	0.979	0.394	0.419	0.538	0.512	0.585	0.021	0.606	0.580	0.462
	x						4	130,926	0.2	5	Regression			N/A	0.492	0.400	0.977	0.391	0.417	0.535	0.503	0.585	0.022	0.605	0.585	0.463	0.489	0.418	0.979	0.394	0.419	0.538	0.512	0.585	0.021	0.606	0.580	0.462
		x					4	130,926	0.2	5	Regression			N/A	0.493	0.399	0.977	0.391	0.417	0.535	0.503	0.586	0.022	0.605	0.585	0.463	0.488	0.416	0.979	0.395	0.419	0.538	0.513	0.587	0.021	0.606	0.580	0.462
			x				4	130,926	0.2	5	Regression			N/A	0.492	0.398	0.977	0.391	0.417	0.535	0.503	0.587	0.022	0.605	0.585	0.463	0.487	0.416	0.979	0.395	0.419	0.538	0.513	0.587	0.021	0.606	0.580	0.462
				x			4	130,926	0.2	5	Regression			N/A	0.491	0.397	0.977	0.391	0.417	0.535	0.505	0.588	0.022	0.605	0.585	0.463	0.486	0.415	0.979	0.394	0.419	0.538	0.514	0.588	0.021	0.606	0.580	0.462
					x		4	130,926	0.2	5	Regression			N/A	0.483	0.387	0.977	0.391	0.416	0.534	0.512	0.598	0.022	0.605	0.586	0.464	0.477	0.405	0.979	0.393	0.418	0.536	0.524	0.598	0.021	0.608	0.581	0.464
						x	4	130,926	0.2	5	Regression			N/A	0.494	0.401	0.977	0.391	0.417	0.535	0.502	0.584	0.022	0.605	0.585	0.463	0.490	0.418	0.979	0.395	0.419	0.538	0.511	0.585	0.021	0.606	0.580	0.462

BIBLIOGRAPHY

1. Armstrong, W., Eddelbuettel D., Lain J. (2018). *Rblpapi: An R Interface to 'Bloomberg' is provided via the 'Blp API'*. R Package version 0.3.8. Retrieved from: https://cran.r-project.org/web/packages/Rblpapi/index.html

2. Bainbridge, S. (1998, October). *Insider Trading: An Overview*. University of California, Los Angeles (UCLA) – School of Law. Retrieved from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=132529

3. Brochet, F. (2010). *Information content of insider trades before and after the Sarbanes-Oxley Act*. The Accounting Review 85: 419-446.

4. Cont, R., Sirignano J. (2018, March). *Universal Features of Price Formation in Financial Markets: Perspectives from Deep Learning*. University of Illinois at Urbana-Champaign

5. Core, J. E., Guay, W. R. (2004, April). *Stock Market Anomalies: What Can We Learn from Repurchases and insider Trading?.* Massachusetts Institute of Technology & University of Pennsylvania.   Retrieved from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=533323

6. Hutchinson, W. (2016). *Presentation on theme: Deep Networks for Natural Language Processing*. Retrieved from: http://slideplayer.com/slide/4455849/

7. Gajanan, M. (2017, September). *Mark Zukerberg Will Sell Up to $12.8 Billion Worth of Facebook Shares Over the Next 18 Months*. Fortune News Corporation. Retrieved from: https://www.bing.com/search?q=Mark+Zukerberg+Will+Sell+Up+to+%2412.8+Billion+Worth+of+Facebook+Shares+Over+the+Next+18+Months&PC=U316&FORM=CHROMN

8. Garfinkel, J., (1997). *New evidence on the effects of federal regulations on insider trading: The Insider Trading and Securities Fraud Enforcement Act (ITSFEA)*. Journal of Corporate Finance 3, 89-111.

9. Guido, S., Muller, A. C. (2016, October). *Introduction to Machine Learning with Python*. pg. 132. O'Reilly Media Inc.

10. Ke, B., Huddart, J. S., Petroni, R. K., (2002, March). *What Insiders Know About Future Earnings and How They Use it: Evidence from Insider Trades*. National University of Singapore and Pennsylvania State University. Retrieved from: https://ssrn.com/abstract=278055

11. Matthews, D. (2013, July). *Insider Trading Enriches and Informs Us, and Could Prevent Scandals. Legalize It*. The Washington Post. Retrieved form: https://www.washingtonpost.com/news/wonk/wp/2013/07/26/insider-trading-makes-us-

richer-better-informed-and-could-prevent-corporate-scandals-legalize-it/?noredirect=on&utm_term=.f6f8b629649d

12. Marcelo, P. (2018). *BatchGetSymbols: Downloads and Organizes Financial Data for Multiple Tickers*. R package version 2.1. Retrieved from: https://CRAN.R-project.org/package=BatchGetSymbols

13. Pedregosa, F. (2011). *Scikit-learn: Machine Learning in Python*. Retrieved from: https://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron

14. Richardson, L. (2018). *BeautifulSoup4: A Library That Makes It Easy to Scrape Web Pages*. Python Package Version 4.7.1. Retrieved from: https://pypi.org/project/beautifulsoup4/

15. Roberts, E. (2013, September). *When Is Insider Selling a Bad Sign?*. Investor Place. Retrieved from: https://investorplace.com/2013/09/when-is-insider-selling-a-bad-sign/

16. Rodgers, J. L., Skinner, D. J., Zechman S. L. (2017, November). *Run EDGAR run SEC Dissemination in a high-frequency world*. Fama-Miller Center for Research in Finance. Retrieved from: http://online.wsj.com/public/resources/documents/1029sechft.pdf

17. Ryle, J. J. (2017, May). *Building an Insider Trading Database and Predicting Future Equity Returns*. Northwestern University Predictive Analytics. Retrieved from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3000704

18. SEC Forms List, (2018 December). *Retrieved From:* https://www.sec.gov/forms

19. Seward, L. (2016). *finreportr: Financial Data from U.S. Securities and Exchange Commission*. R package version 1.0.1. Retrieved from: https://CRAN.R-project.org/package=finreportr

20. Sun, L. (2015, August). *Twitter Inc. Insiders Sold Over $100 Million in Shares in 3 Months*. The Motley Fool. Retrieved from: https://www.fool.com/investing/general/2015/08/04/twitter-inc-insiders-sell-over-100-million-in-shar.aspx

21. Yu, R. (2017, March). *Caterpillar Shares Fall After Tax and Accounting Fraud Repot*. USA Today. Retrieved from: https://www.usatoday.com/story/money/2017/03/08/caterpillar-shares-fall-after-tax-accounting-fraud-report/98892874/

22. Zacks.com Stock Screener. (2018, August). Retrieved From: https://www.zacks.com/screening/stock-screener

ACADEMIC VITA

Paul Boehringer **PaulBoehringer0989@gmail.com**

EDUCATION

The Pennsylvania State University – Schreyer Honors College **University Park, PA**
Bachelor of Science in Industrial Engineering & Math Minor *May 2019*

Stanford University **Stanford, CA**
Masters of Data Science June 2021

PROFESSIONAL & ACADEMIC EXPERIENCES

Deloitte Summer Scholar - Technology Consultant **Pittsburgh, PA**
Strategy and analytics division for a healthcare data processing project *Summer 2018*

- Contributed recommended operating approaches to Deloitte's 'playbook' for big data solutions

Defending Fake News with Booze Allen Hamilton **State College, PA**
Built a language processing (NLP) algorithm to predict a reliability of a news piece *Spring 2018*

- 3rd place at Penn State capstone showcase out of 200+ projects
- Full stack Java development for web crawling and chrome plug-in
- Wrote Python scripts for language processing of articles analysis on webpages

Financial Network Failure Propagation Research **State College, PA**
Investigated financial network structure's role in a firm's bankruptcy risk *Spring 2018*

- Researched and implemented stochastic processes on various types of financial networks
- Model financial crises in R to find liability structures which could have prevented cascades

Wayne Automation Corporation **Norristown, PA**
Automate manufacturing and supply chain processes *Summer 2015 & 2016*

- Implemented new automated iOS based inventory system to speed up part picking process
- Developed new machine accountability system for customer maintenance and warranty tracking
- Redesigned assemblies welded to machines to replace them with bolted connections

FIRST FRC Team 1218 Vulcan Robotics **Philadelphia, PA**
Mentor for the mechanical and computer aided design (CAD) team *Summer 2014 - Present*

- Sponsor liaison for all supporters such as Boeing, Vulcan Springs, TE Connectivity, & NASA
- 7th in 2014 & 5th in 2015 at FRC World Championship

PERSONAL INFORMATION & ACTIVITIES

Skills:

Python	R	SQL
Web Scraping	Machine Learning	Computer Aided Design (CAD)