Machine Learning and Market Analysis

David Lee Max Katz-Christy

Contents

Proposal

We will be spending the semester researching and studying the complex field of machine learning. Machine learning is a way for computers to "learn" with data and be able to predict future outcomes or tendencies. For instance, creating an AI that can play chess, showing curated online ads, facial recognition, and Google Translate are all examples of where machine learning is applied. Our goal is to apply machine learning to various markets; we thought this would be interesting because markets are not very easily predictable by humans but may have predictable trends that a machine could pick up.

Our main project will be to analyze the stock market. We want to create a bot that will be able to predict whether stocks will rise or fall, given the past history of the stock as well as other data (like twitter mentions) that we need to determine. Our reasoning for starting with the stock market is that there are large backlogs of data that we can train and test our models on. We can also invest a ton of money after finishing our bot and get rich! If we are successful in accomplishing this task, we also have plans to apply similar systems to the cryptocurrency market or the job market.

By working together on this project, we will be able to combine the different skill sets we bring from our areas of "expertise". David has worked a lot with machine learning and the theory behind many of the concepts we will deal with through different internships in the past. Max has worked a lot with robotics and has a very well rounded computer science background, as well as currently taking an economics course at Harvard. We are both excited to work on this project for this current semester!

Brief Project Overview

Our goal was to create a machine learning model that could effectively predict stock prices on the stock market. We started with a linear regression model to make sure we could work with the simplest type of model and that the general format of our data was correct. We then quickly ramped up the complexity - we worked on implementing a neural network. After finally figuring out the format of the data that our network wanted, we began the slow and painful process of actually finding the right parameters and data structure that would let our network perform efficiently. At first, our network only predicted accurately on the first few days of data, but eventually over time we got a decent general prediction over the entire log of data and into the future. After this, we found an API for Google Search mentions and incorporated that into our network, and after some fine tuning, our model became very accurate on our data logs and the future. After testing this model on 29 of the largest publicly traded companies in the world, we found that on a good run our network could predict the one month return of 28 out of the 29 companies correctly, and have a 8.4% return compared to the 0.6% average return of the whole market.

Supporting Documents

All of our work is documented and collected in the following github repository: https://github.com/davidinholee/stock-bot/

Annotated Bibliography

Citations	Brief Summary	Uses and limits of this work in relation to our own thinking	New Questions

1. Chollet, Francois. Deep Learning with Python. Manning, 2018.	This book offers an in depth introduction to deep learning in Python. There are many practical, hands-on explorations of machine learning concepts with code snippets you can test for yourself.	The code examples presented in the book use the deep learning framework Keras which is built on top of Google's TensorFlow backend engine. We are going to use these same packages for our exploration, so this book really helps us to understand the basic syntax and starting concepts that we need to understand. However, much of the book is also the many applications of deep learning from computer vision to natural language processing, which is interesting to learn about, but most of it is not useful/relevant for our specific research.	Is Keras, the main Python library we are working with, modularized to be able to have all the functionalities of TensorFlow? Keras basically makes TensorFlow much easier to read and learn (by having easier syntax), so it could be possible there are some functions that are not directly transferable between the two.

2. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly, 2018.	This book also offer an in depth introduction to deep learning in Python. Like Deep Learning with Python, this book also offers many good examples and code snippets, but uses some of the more fundamental problems in machine learning.	Instead of using Keras, this book mainly works with the Python package scikit-learn, which is a more "basic" form of TensorFlow because it cannot construct neural networks. This is very useful to us because many of the normalization and overfitting techniques we have to use are actually built upon scikit-learn code, even though it is implemented in TensorFlow. Also, this book deals with the most famous machine learning problems, like the MNIST dataset. Although this is not directly related to our problem, which is a limitation of the book, it is still really interesting to observe how powerful machine learning can actually be if implemented correctly.	Can we use the concepts that are applied to these famous datasets for the problem that we are dealing with? Although these datasets mostly deal with images or more mundane types of data, many of the big ideas can be applied across the board to almost any machine learning problem?

3. Pearl, Judea, and Dana Mackenzie. The Book of Why: The New Science of Cause and Effect. Penguin Books, 2018.	This book offers a thorough explanation of causal inference and the statistical analysis behind cause and effect. Even more than 20 years ago, statisticians could only prove correlation, not causation, but with new scientific methods the work on causality is finally being tackled.	This book is very relevant to our work because we need to determine if the datasets we will use in our network help determine the predictions because they have a causal effect on the stock or just because they are correlated with the stock. Things like Google search mentions may be a direct result of a large change in a stock's value, it may be the cause, or it may even be neither and confounding variables could have a role. Understanding cause and effect relationships is thus important in gauging the usefulness of the datasets we will use, and if we should continue using them. Obviously this is a book centered around statistics, so much of the actual equations and many of the concepts are irrelevant to our work, which is a limitation.	Can you ever completely prove direct causation between any two variables? It is important to be very careful of declaring things like this because of examples described in the book where people have irrationally concluded things like smoking actually reduced infant mortality.
4. Heinz, Sebastian. "A Simple Deep Learning Model for Stock Price Prediction Using TensorFlow." Medium, ML Review, 9 Nov. 2017, medium.com/mlreview/a-simple-deep-learning-model-for-stock-price-prediction-using-tensorflow-30505541d877 .	This is essentially a tutorial on how to use tensorflow for machine learning with S&P500 data.	This is helpful and something that we could build off of for our algorithm. The strategies used are somewhat generic, but the author uses very helpful illustrations for visualizing the learning process. Understanding the process is key to our ability to optimize our algorithm.	Google invests a lot of time and money into AI and ML... How are they using it for themselves? How much profit are they making from doing all of this research? What makes it worth it for them?

5. “Machine Learning for Trading - Topic Overview.” Sigmoidal, Sigmoidal LLC, 15 Oct. 2018, sigmoidal.io/machine-learning-for-trading/.	This article outlines how machine learning is being used in the stock market and how profitable it has been. It also briefly recognizes google trends as a viable source for trading information.	This article argues that machine learning is very relevant in trading, and is able to boost profits. This confirms our original theories and helps guide us in our overall goal. It provides some insight into our next steps once we get a basic algorithm running. It also highlights the importance of understanding machine learning. It may not help us with the technical details so much as the overall direction.	How helpful will machine learning in 10 years, once it becomes industry standard and every large firm is even again? What will be the next wave of profitable technology in the stock market? Will machine learning be the last large step? What will give companies an edge?
6. Milosevic, Nikola. “Equity Forecast: Predicting Long Term Stock Price Movement Using Machine Learning.” ArXiv, 2 Mar. 2016, arxiv.org/ftp/arxiv/papers/1603/1603.00751.pd f.	This is an academic paper where students created an algorithm for predicting long term stock prices. They were successful in ~75% of cases in predicting if a company would rise 10% in a year or not. They also used some other qualities of an equity’s finances as inputs.	This is a different approach than we have looked at before, and is very interesting. So far, we’ve been training on a year or two of data and predicting <100 days in the future. Our algorithm would be optimized for day trading, where this paper is studying years ahead. This is a different approach that we could try, but it would require a different approach on training algorithms. Regardless, it is good to be thinking of the idea of short term vs long term stock predictions, and will help us find where we want to be on that scale.	What algorithms are optimized for short term and which are for long term? Which of the inputs this study uses would be helpful for short term algorithms as well as long term? Where did they collect their data from? Is trading on long term predictions viable with technology changing the way markets act so rapidly?

7. Koehrsen, William. “Stock Prediction in Python – Towards Data Science.” Towards Data Science, Towards Data Science, 19 Jan. 2018, towardsdatascience.com/stock-prediction-in-python-b66555171a 2.	This article is about someone who uses the stocker platform for data on Amazon. They start with a rudimentary strategy and build it up over time.	This article doesn’t go into the technical details, but does provide a brief story that is helpful for understanding what steps to take to move forward in refining our algorithms. We can take the methods they used into consideration when designing our own algorithms. However, the use a different source for their stock data then we do and the syntax thus differs by a considerable amount, so it is more the big ideas that we are trying to gain an understanding of.	How often does data manipulation occur? In an almost cynical sense, the article talks about how to fudge the data if you obtain undesirable results, which we obviously will not do, but how much does this lying occur in the actual field?
8. Singh, Aishwarya. “Predicting the Stock Market Using Machine Learning and Deep Learning.” Analytics Vidhya, 26 Oct. 2018, www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-p ython/.	This article describes various machine learning algorithms and how effective they are. It works progressively toward more complex algorithms and explain the concepts behind each algorithm.	This article is very helpful in describing which algorithms don’t work and why, and leads to the conclusion that the Long Short Term Memory (LSTM) algorithm might lead to the best results. It doesn’t go in detail into how LSTM works, it does give a brief overview and demonstrates how it can be implemented. It also explains algorithms that aren’t so accurate, but help build an understanding of how LSTM works.	What are the different types of LSTM implementations and how do they work? How can we incorporate other types of data in with the LSTM prediction?

9. Braun, Max. "Trump2Cash." GitHub, 22 Sept. 2018, github.com/maxb braun/trump2cas h.	A program that uses references to stocks in the president's tweets to predict stocks in python	This is a very helpful example of how we can use twitter feeds to provide insight on stock prices. It helps reinforce our original hypotheses on what affects stocks and gives us potential tools to use in python. It isn't in and of itself a tool that we can use, but a reference for how to complete some specific tasks.	How do presidential tweets differ from those of economic advisors and popular stock brokers in their influence? What other popular social media platforms can be analyzed?
10. Hilpisch, Yves. "Algorithmic Trading in Less than 100 Lines of Python Code." O'Reilly Media, 18 Jan. 2017, www.oreilly.com /learning/algor ithmic-trading-in-less-than-10 0-lines-of-pyth on-code.	This is an article about a script that uses time series momentum strategy and the platform Oanda to predict on backlogs of data and compare different variations of the strategies to maximize potential profits.	This is very helpful because it shows a different platform that we can use to gather data. Although it isn't an in depth explanation of every step, this is a script that we could work off of to improve with other inputs. It also mentions Quantopian, which is an online tool for writing and testing algorithms, which could also be useful.	What do the professionals use? What are the best platforms that provide detailed data quickly and in an easy to use format? How effective will the training on one stock be on another stock?

Thank You Letters to Community Members

To Eddie Kohler, Harvard CompSci 61 Professor

Hi Eddie,

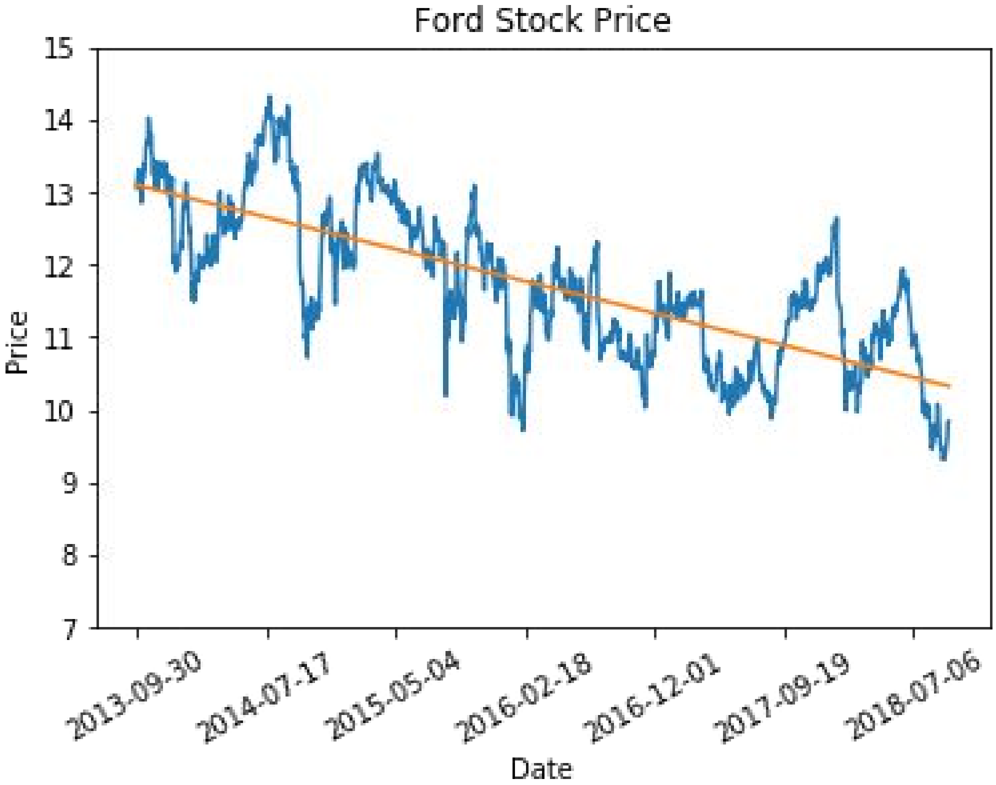
Thanks so much for the class! It was a fantastic experience and I learned so much. I really appreciate your generosity in going out of your way to let high school students to take your class and taking on the extra work that comes with it. On top of that you were a great teacher and made the class very interesting and a lot of fun. A number of CS interested students have approached me inquiring about classes at Harvard, and so if you are still willing next year, there's a good chance that a few of them will be interested.

Happy New Years!

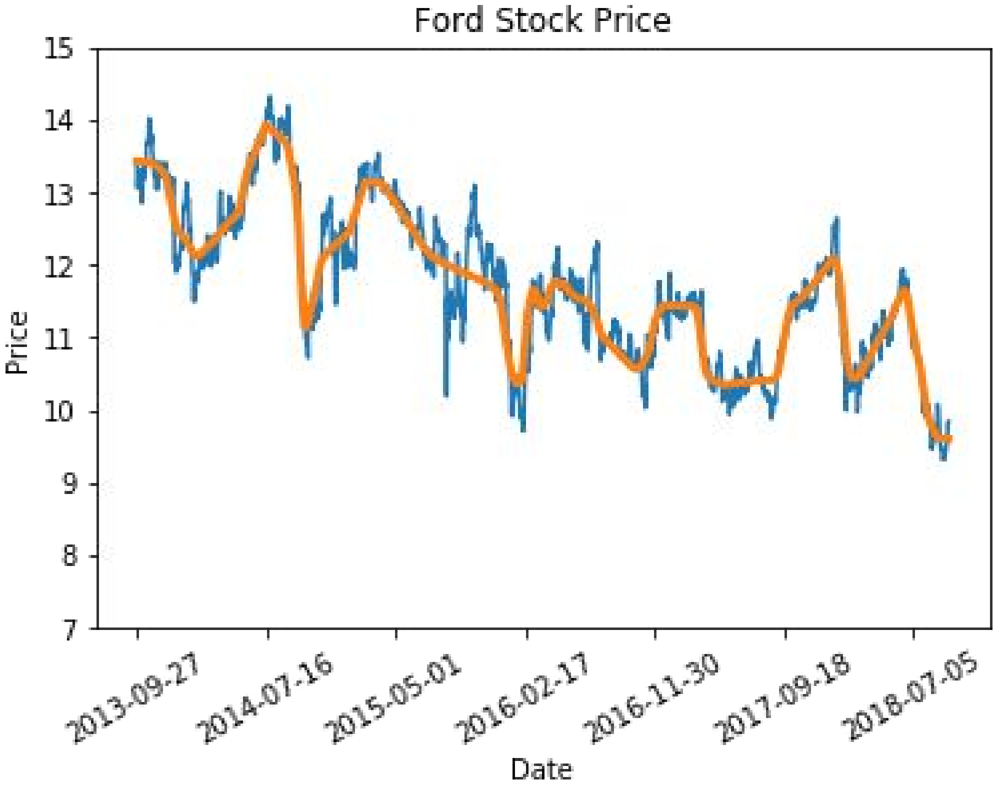
Best, Max Katz-Christy and David Inho Lee

Formative Reflections

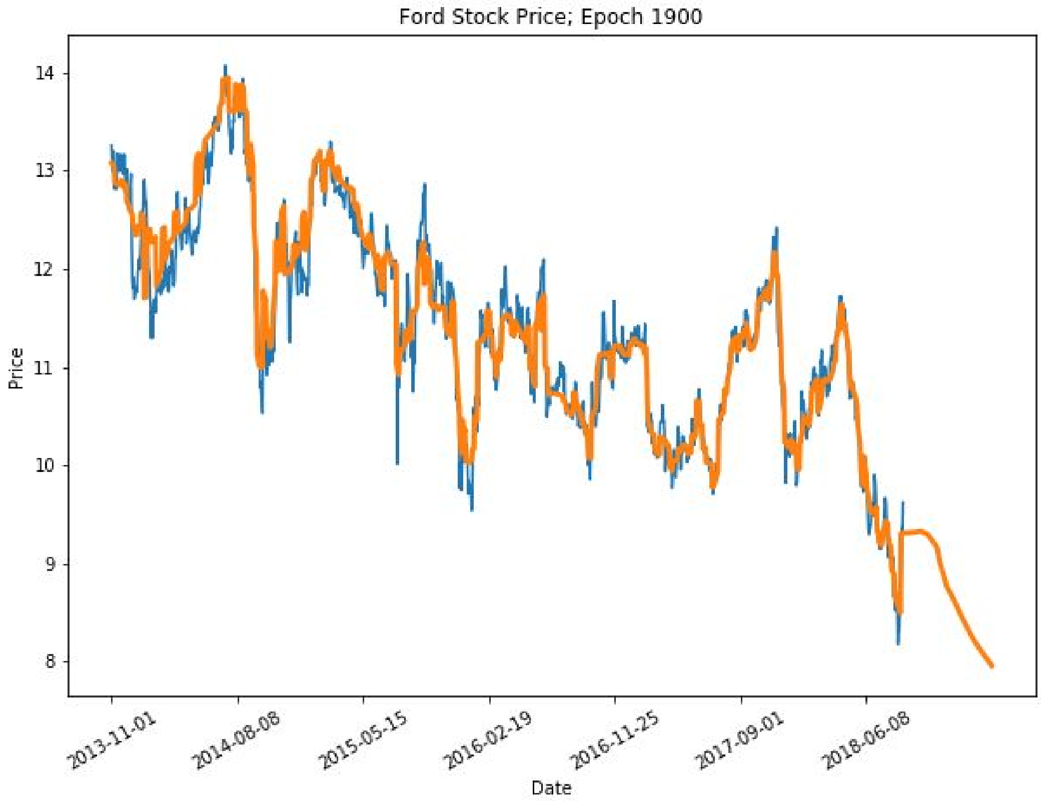
- Logistic Regression Model: We were able to build an effective logistic regression machine learnign model fairy easily. The hardest task was figuring out the best way to obtain the data that we needed. We spent a lot of time looking for the best API for getting consistant, reliable stock data. After deciding on the API, pandas-datareader, we spent time playing around with the library to see what data we could access. We ended up finding out we could reliably get daily stock data for the past five years, giving us approximately 1250 points of data. We then theorized about early preprocessing methods, and organized the stock data into the format we wanted. Creating the actual machine learning model was pretty simple after that, especially because we had a lot of experience with this type of model from previous internships.



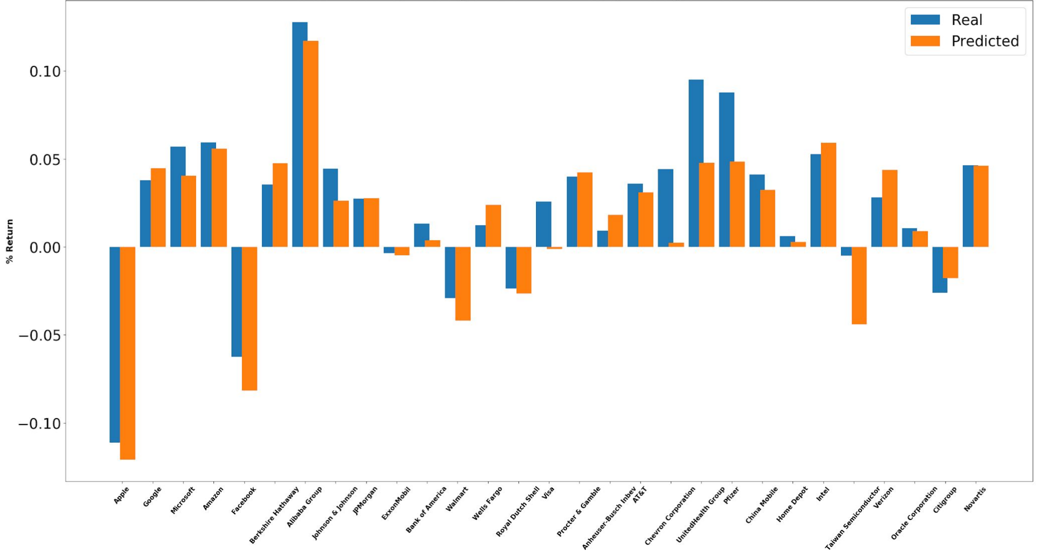
- Early Neural Network Development: We had many challenges when first implementing a neural network for our project. It was challenging to figure out how to shape the data to the very specific requirements of the network. Even after finishing this, we had to figure out what to do with all the hyperparameters of the actual network. After completing the very first generation of our network, the performance we were getting was terrible. The model performed fairly well for the first few days of the stock data, but then quickly predicted linearly after that. (insert image from slide 17 here) The most amount of time spent during this whole process was trying to figure out how exactly we needed to normalize our data so that the neural network would put the same amount of emphasis to each day of data instead of the first few days like we were thinking. We also had to keep tinkering with the number of epochs, the size of each layer, the number of layers, whether or not to include dropout or other types of layers, loss function, etc. to find the best possible performance for our network. But after working at this problem for a long time, we eventually landed a model that we were happy with and was performing decently well.



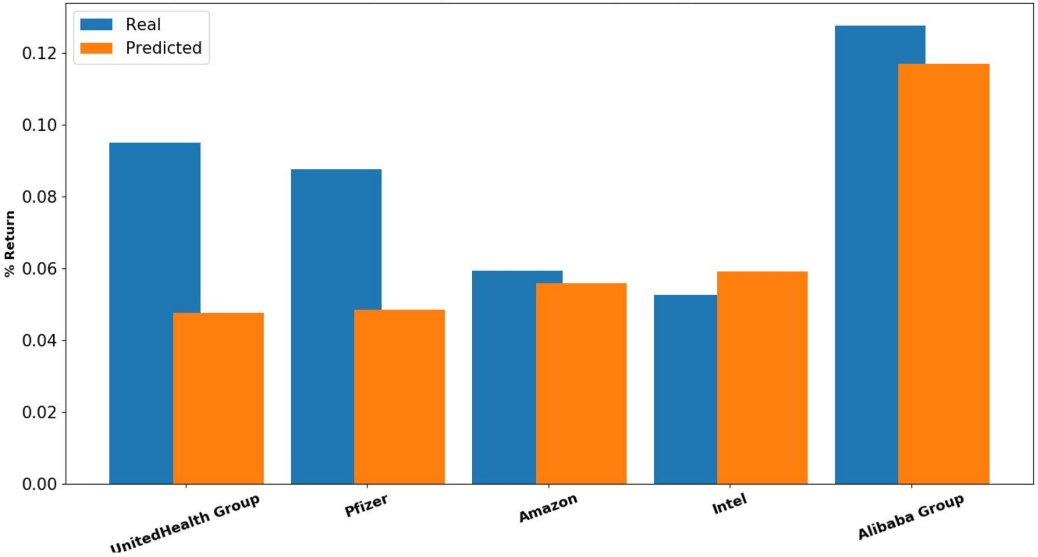
- Later Neural Network Development: We decided as a next step to incorporate more data into the network, as the ~1250 samples we were using for our input was not very large at all compared to the hundreds of thousands to millions of data samples that is usually recommended for machine learning models. We thus ended up looking for an API that could access twitter data or some other data relevant to internet mentions. We landed on using the pytrends package which accesses Google Trends data. This worked well for us because it would give the network information on how much the stock of interest was being mentioned in the news, which definitely has a direct effect on the actual stock price. We had to restructure our entire data input to accommodate for this new source of data, but after finally implementing this we saw incredible results. Our network's prediction now precisely followed the stock data without showing many signs of overfitting. The loss of the network was at its lowest point yet and future stock predictions appeared legitimate. At this point, we could start to finalize the structure of our model.



- Final Steps and Testing: As a next step, we needed to make our network actually usable to outside users. We decided to create a script using the stockbot class we had created to predict what the best and worst performing stocks would be in the future. The actual code implementation of this idea was not hard at all, we just needed to decide on the specifics of the numbers we would use. We ended up deciding that a one month testing period was a good enough heuristic - our model would take the top 29 stocks in the world, train and predict on the data for each of them, and predict how each of them would perform in the next month, saving multiple graphs to display the predicted performance of these stocks. We also created a checking script to evaluate the performance of our model. We found that on average our network would predict whether a stock would rise or fall in the next month correctly for 28 out of the 29 stocks. The only difficult part about this final process was that the running time for our code at this point neared two hours, so if there was any bug in our code it took a long time to actually be able to fix.



Real vs. Predicted Returns; Average Return: 0.02333; Average Predicted Return: 0.01489



Top 5 Predicted Returns; Average Return: 0.02333; Average Predicted Return: 0.06565

Summative Reflection Letter

Dear Presentation Panel,

Working on our graduation project this semester was a very unique and fulfilling experience for the both of us. We both had done work in the past either directly or indirectly relating to the field of machine learning, but this was the first time we were given such independence to do whatever we wanted. We landed on the idea of creating a stock bot because it was an idea we had vaguely heard about but didn't know too much about and it was something very different from what either of us had done in the past. The methodology of our work process was a little different as well. We were used to being told where to start and what resources to look at to begin, but in this project we had to come up with everything from scratch. For us this was a huge plus, the process of brainstorming, which both of us had been doing since the previous year, made our project in the end really feel like it was ours and made the whole experience a lot more fulfilling.

Looking back on our project, we are really happy about what we were able to accomplish and where we ended up finishing. We both were honestly astounded by how well our model was able to perform in the end: we had an average yearly return rate 14 times greater than that of the overall stock market, which we thought was more than impressive in any context. We successfully were able to incorporate more than just stock data into our model, which was another one of our goals, and the overall user friendliness and object orientedness of our code was very satisfying. There are however a couple of things we would have liked to add to our project that we did not get to. The final two scripts we ended up writing to evaluate model performance and give future stock predictions are honestly pretty limited in scope, so being able to make these into full fledged libraries in the future is a direction we could head in. We also could always add more data sources, whether that be twitter mentions or news reports or something we haven't even thought about yet, and we never got to the point where we could use smaller neural networks to predict the data to go into the larger neural networks.

Comparing our work to the literature that is out there, we have some similarities and differences. The overall structure of our data and our network is largely similar to those who have done similar work to make stockbots like ours: the major difference is the methodology in which we came to the final product. We made it a point to ourselves to write our entire project completely ourselves: obviously we could consult stack overflow and other sources for the more technical aspects of our code, but we did not want to just reuse code that someone had already written and just get the same results as them. Because of this choice, our process probably took a lot longer than it should have. We are both amateurs in this field, so there were a lot of things we did not think about in the early stages of our project that came around to bite us in the back later on, costing us a lot of time that we could have used to add more functionality. But we think it is because of our painstaking methodology that we have been able to gain so much more knowledge about what we did as well as machine learning as a broader field.

One thing we would like to do in the future is to create a fully fledged website, application, or tool using the network and related scripts we created. Making our work user friendly aesthetically pleasing to look at should be a high priority especially given the complex nature of our project. Having an easily accessible place to go would be a really nice way to be able to show off all our work to somebody who might not be that tech savy or just in a quick and easy way. We also never got to actually test our model on the real world by investing actual money into the stock market. We think it would be really cool and interesting to invest a small amount of money into the stock market based on what our model says we should do, and evaluate how well it did by the end of second semester, kind of as a followup to this project. It would really show that we have faith in what we created and would be a great ending to this journey that has had all of its ups and downs. But overall, we are really happy with what we were able to accomplish and we will both definitely continue to pursue this type of work in the future.

Sincerely, David and Max