Writing Assignment 1

John Caruthers

06Feb24

The *Stock Market Data* problem is supervised as each variable contains an associated label[3]. The label for each variable is the direction the stock market took that day (Up or Down). The problem utilized the previous 5 days of stock market percentage performance to predict the current day's performance. Boxplots were then built with the data to perform data exploration. The book stated that the expectation was low that something so simple could be used to accurately predict market performance[3]. Interestingly, the book states that when trained on 5 years of data, there was better performance in the model, it could predict the direction 60% of the time[3]. Both models mentioned(trained on 5 days and another trained on 5 years or data) are classification problems.

The dataset used for training these models, Smarket, is available from the **ISLP**: *Introduction to Statistical Learning Package*. This dataset is available on the textbooks website[6]. The dataset contains 1250 variable, which are each trading day for the S&P500 from 2001 to 2005. Each variable contains eight features: *Year*, *Lag1*, *Lag2*, *Lag3*, *Lag4*, *Lag5*, *Volume*, *Today* and *Direction*. *Year* is the year in which the variable was documented. *Lag1* is the percentage return for the previous day. *Lag2* is the percentage return for two days previous. *Lag3* is the percentage return for three days previous. *Lag4* is the percentage return for four days previous. *Lag5* is the percentage return for five days previous. *Volume* is the total volume of the day documented in billions of dollars. *Today* is the percentage return of that particular day. Finally, the *Direction* label is either "up" or "down". This label indicates whether the days value increased or decreased respectively. Personally, I believe this data is not the most interesting or best features to use. It however, is the easiest to gather as it could all be pulled from one source such as Yahoo Finance or equivalent - for personal use only[1]! Availability of data can be a limiting factor in feature selection or engineering.

Feature engineering is an important step in machine learning, as it ensures the model is trained on applicable data-preferably data that positively impacts model performance. When feature engineering is performed poorly, bias can be introduced into the model-called feature selection bias[4]. When selecting features, it is important to understand that each feature may contain some bias from generation or transformation. Other features not included may contain important data but not be included for various reasons-such as lack of data, or difficultly in incorporating quantifiable data in the model[4]. Properly selecting all features needed for training the model will help reduce risk of bias. Feature engineering would be implemented by first understanding the data and the problem. What features would impact the target value, what features have no impact on the target value? This can be an incredibly difficult question to answer as the system may be incredibly complex-as is the stock market example. Sometimes experts in that particular system, not the machine learning engineer, would know which features would impact the target value. It is important for the machine learning engineer to work with experts in the field to ensure proper features are selected, and ensure minimal bias is introduced into the model training.

In this example, economists and large financial institutions would best know what features to include. Out of the existing dataset mentioned above, there really seems to be no correlation. If possible, additional data would need to be gathered to built a more accurate model. Even though I am not an expert, I would say corporate earnings calls, news articles, futures, and economic data released by governments/NGOs

would be the features I select to train a similar model. Perhaps more interesting is mining social media for specific words and performed association rule mining, or sentiment analysis. Recently, response to Federal Open Market Committee meetings or articles about the Federal Reserve may be an interesting feature. It is difficult to perform feature engineering with stock market prediction as futures are as much about economics as they are about human psychology. Emotional reactions to events often drive individuals and organizations to buy or sell on the stock market-even at an irrational level. Taking the recent focus on the Federal Reserve's work in reigning inflation. We expect that the Federal Reserve will cut interest rates in the near future as the economy has been relatively strong due to consumer resilience and impressive growth in productivity and GDP, as well as a strong labor market. At the time of writing, inflation is at about 3.5%, above the target rate of 2%. If the Federal Reserve cut the federal funds rate prematurely, it could jump start inflation due to lower interest rates-similar to The Great Inflation from 1965-1985[2]. It seems many individuals would rather see interests rates cut now at the risk of rebounding inflation than cut later, reducing that risk. When the Fed's state they are not cutting interest rates, the stock markets lose value. it is also possible the markets have accounted for a specific number of cuts in their futures, and the markets correcting. This was a long example that could indicate bias amongst the machine learning engineer (me) on stock market. It is important to be aware of internal biases and consult with experts when selecting features.

The ML model mentioned above would be a classification model. It would only try to predict the direction of stock market value, not a quantitative value. I would attempt to utilize nonlinear regression since the behavior of stock market trades would most likely act nonlinearly with selected features. For example, I expect the stock markets to react positively to above expectations of corporate earnings but the stock value would most likely decrease if the earnings were positive, but below expectations. Essentially, positive profits don't always mean stock market value increases, so there is a nonlinear relationship. Including expectations, EPS, futures all increase the complexity of the relationship. The output of the model is a either a 1 or 0, which is the binary representation of the stock going up or down that day.

Predicting stock market performance on a daily basis with any sense of accuracy is a nearly impossible task with current methods and technologies as it is an enormously complex web of interdependencies (companies, governments, trade, technology, human behaviors and psychology). Performing mathematical operations on data containing all these features (1000s or 100,000s of dimensions) would be computationally too expensive for current systems. Secondly, to more accurately make predictions on stock market performance, many decades of data need to be used for model training and testing as economic cycles last decades. Only training the model from data between 2001 and 2005 does not train the model on recessions as seen in 2008/9, or the COVID recession, or economic booms such as the post 1949 economic boom, or even the current economic expansion. it is highly unlikely that this model could accurately predict the direction of the stock market, because if it were that easy, this would have already been made and exploited for profit. Impacts to society and ethical implications would require an educated guess as best and wild speculation at worst. I will attempt to stay away from speculation, no matter how entertaining it would be. If a machine learning model of an accurate ability to predict stock market directions were made, it would certainly be used to make the owner a profit, at least in the short run. To stay relevant with current economic and technological trends, the model would constantly need trained on newer data-called continual learning [5]. The model may eventually become trained on trades that it helped make. If the model is widely accepted and is able to manipulate the market based on it's shear volume of trading, then it may start to become ineffective. The reason being is it was primarily trained on trades made by humans (or bots that humans made), but over time the training data would be "polluted" by trades the AI made in the first place [5]. This is called model collapse and leads to a "forgetting" and failure of the ML model [5].

In conclusion, The *Stock Market Data* problem poses an interesting introduction into machine learning, as the majority of people wish they had more money. The dataset is widely available on the **ISLP** package. I do believe this is interesting and a good example of classification. The book states the data was used in

chapter 4 to build a better performing model using *quadratic discriminant analysis*[3]. Some items I don't like about the example is the data is actually redundant, you can see in Figure 1 below that the *Today* feature and *Direction* class are redundant. All negative values of *Today* are orange (Down) and all positive values of *Today* and blue (up). In my opinion, the *Today* feature could be completely removed as we are only interested in the direction of value, not the percentage change. Secondly, I think this example is just plain boring from a data point of view. It is obvious that the previous day's trades won't really impact the current day. I wish the book would have built a more interesting database such as adjectives used to describe the markets in social media posts, or patent applications by companies on the S&P500. Lastly, the results of the example in the textbook tell me that there is no easy way to predict stock market data and more data is needed. The model trained on 5 years of data had a minor correlation-however it didnt take into account economic cycles. Personally, I will stick with what I know works best, realizing the markets work best over the long term, so emotionally reacting to news is not value added. Also dollar cost averaging is a good method to hit a general average of the good and bad days on the market.

Figure 1 below is a pair-plot of the database from the website. I built it in python using the Seaborn package. This is a valuable way to perform elementary data exploration as the color of the datapoints can be set to equal the class. For example, the direction of the stock market for the day is broken into orange (down) and blue (up). This the pair plot plots every point compared to one another. Seeing the hues, visually indicate a correlation or interesting feature. In this case, it told me that *Today* is a redundant feature that has no value in being used as a training input. The correlation command in Pandas is a good quantifier for the visualization as it will assign a number between -1 and 1 to the pairplot values, 1 being a perfect positive correlation, -1 being a perfect negative correlation and 0 having no correlation. The highest correlation in this example was between *Volume* and *Year* at 0.539006.

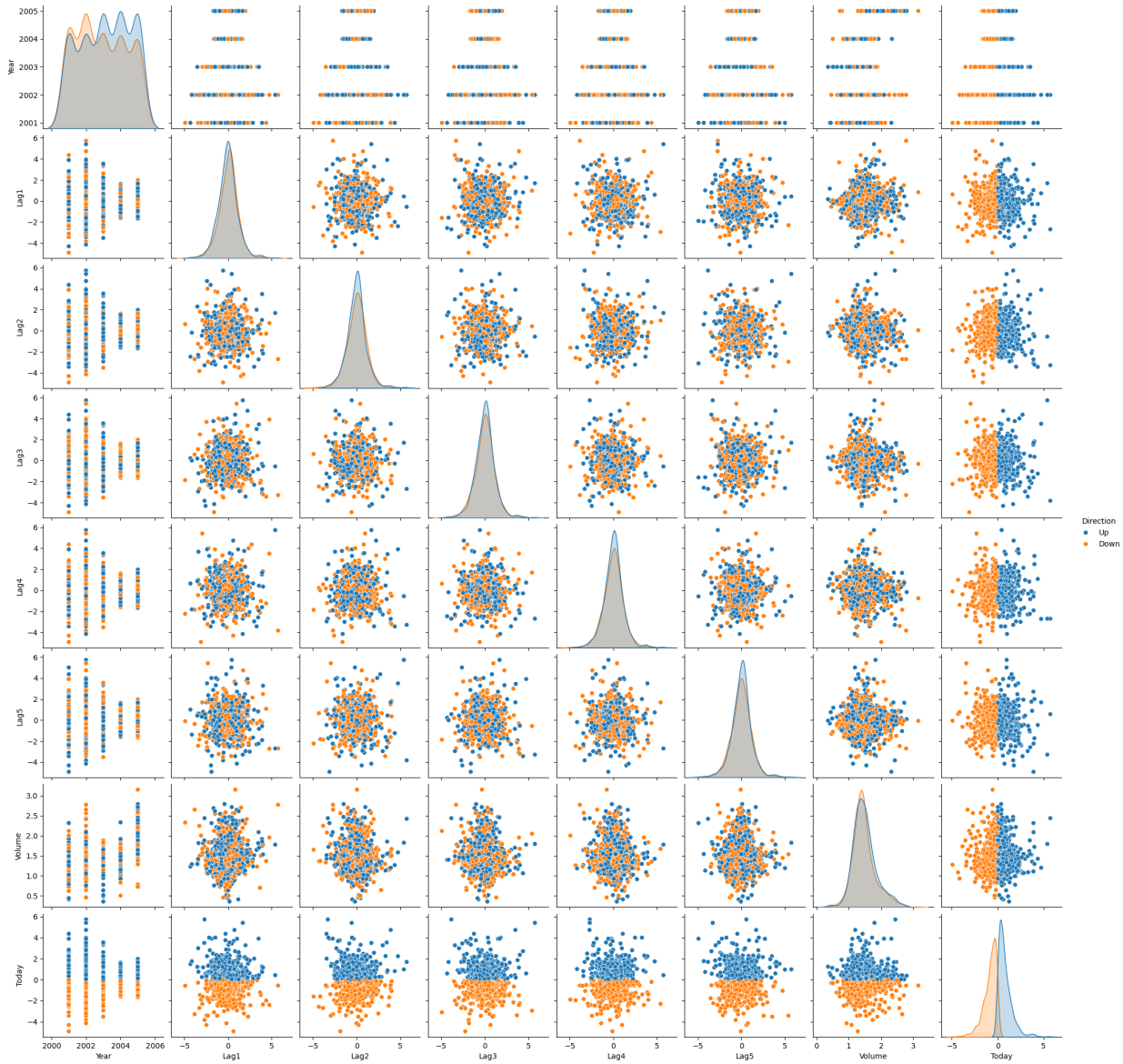

Figure 1: Pairplot of data

References

[1] Ran Aroussi. *Download market data from Yahoo! Finance's API*. URL: https://github.com/ranaroussi/yfinance.

[2] Michael Byran. *The Great Inflation*. URL: https://www.federalreservehistory.org/essays/great-inflation.

[3] Gareth James et al. *An Introduction to Statistical Learning*. Springer Texts in Statistics, 2023.

[4] Drew Roselli, Jeanna Matthews, and Nisha Talagala. "Managing Bias in AI". In: *Companion Proceedings of The 2019 World Wide Web Conference*. WWW '19. San Francisco, USA: Association for Computing Machinery, 2019, pp. 539–544. ISBN: 9781450366755. DOI: 10.1145/3308560.3317590. URL: https://doi.org/10.1145/3308560.3317590.

[5] Ilia Shumailov et al. *The Curse of Recursion: Training on Generated Data Makes Models Forget*. 2023. arXiv: 2305.17493 [cs.LG].

[6] Jonathn Taylor. *S&P Stock Market Data*. URL: https://intro-stat-learning.github.io/ISLP/datasets/Smarket.html.