CS230

CS 230 Project Milestone: Multimodal Financial Market-Crash Prediction and Market-Health Analysis

Pratham Soni
Department of Computer Science
Stanford University
prathams@stanford.edu

Harshal Agrawal
Department of Computer Science
Stanford University
aharshal@stanford.edu

Abstract

In this paper, we propose a multi-modal deep-learning methodology to predict the next market crash so that policy-makers and the treasury can be better prepared take preventative measures. Given the inter-connectedness of modern markets and increased susceptibility to volatility, we strive to build upon existing literature in the field and develop a model that incorporates both quantitative financial market data and qualitative sentiments from news headlines to better predict such market-crashes. In our model, we test a variety of encoder architectures and ultimately achieve an AUC of approximately 0.8 across the task using multiple dense layers and dropout.

1 Introduction

Having just come out of the largest market crash in recent history and witnessing the impact that it had on the livelihoods of millions of Americans, we think a deep learning model that could effectively predict such crashes has potential to have great social impact. Further, with the rise of retail investing in recent months courtesy of platforms such as Robinhood and low-to-no commission trading, regular, everyday, non-institutional investors have taken to the markets in hordes with increasing portions of their savings now being susceptible to the volatility of the stock market. Thus, a model that could readily warn these in-experienced investors has greater need now then ever before.

2 Related work

As seen above, the use of predictive algorithms in the finance field has many motivations. This has led to an abundance of literature exploring this concept. Generally, most works note the abundance of information available in making predictions, but tend to focus on some subset of the available data corpus. Furthermore, the ever-increasing abundance of publicly available data has further accelerated progress and developments, such as the use of machine learning to monitor stock-related sentiments through Brexit [5].

Currently, we will focus on some foundational works for this paper. [3] quantifies definitions of stock market crashes based on sustained negative movement; we adopt this definition for this work. [7] shows that sentiment analysis of stock news is in fact a useful data stream for monitoring market health. [6] provides a baseline for the use of BERT encodings in the financial field for stock market prediction. [4] provides background into the use of CNN/LSTM based embeddings constructed from similarly posed multimodal data.

3 Dataset

Our data is of two types: quantitative in the form of market data and qualitative in the form of news data. Our motivating factor behind considering both data types is that given the nature of recent crashes, quantitative market health indicators could be suggestive of market well being but unexpected news events such as the shutting down of the economy due to COVID or Brexit could end-up resulting in market crashes. Due to availability, we only consider trading days between August 8, 2008 and July 1, 2016, for ≈ 2000 days. For market data, we used daily closing prices for 11 Vanguard sector-specific ETFs for industries like Transportation, Energy, Tech/IT, etc. The tickers that we considered were: VOX, VCR, VDC, VDE, VFH, VHT, VIS, VGT, VAW, VNQ, and VPU. In addition, we also considered VIX, the Chicago Board Options Exchange's CBOE Volatility Index, which is the industry-standard measure of market volatility, in particular the SP500 index volatility as measured by options volatility. Our decision to include VIX was influenced by literature in the field, Sotirios et al. in particular, as well as general intuition that certain days with high measured volatility could be the cause of particular market crashes. All market data was retrieved from Yahoo Finance. For input into our model, a 10 by 12 matrix was used, comprising of the closing price for the 12 tickers on the current day as well as the previous 9 days. Each input matrix was labeled 1 or 0 if the closing price for the SP500 for the current day saw a greater than 2% decline when compared to the previous trading day. In other words, if the SP500 saw a 2% decline from the previous trading day, we consider that to be "market crash" for our training purposes.

For news data, we use a curated dataset from Kaggle by user @Aaron7Sun where the top 25 upvoted news headlines for a given trading day are taken from the Reddit World News Channel (r/worldnews) as a 25 by 1 input [1] . Similar to the market data, our input data was a 25 by 10 matrix comprised of the top 25 headlines for the current day and the past 9 days and each matrix was labeled 1 or 0 if there was a significant decline in the SP500 value for the current training day or not, respectively. We used a 60/20/20 split between training/validation/test sets with random assignment of data into each set.

It is important to note that in modern times, a 2% decline is relatively frequent due to all-time high market volatility. However, due to the lack of major, frequent market crash events (15% decline or more), such as the 2008 recession, within our time span, there would be virtually no positive samples in our data set if our criteria for positive samples was increased from 2% to 15%. With a benchmark of 2%, we found 106 positive samples, or days where there was 2% market decline from the previous trading day, in our data span over the course of 8 years.

4 Models and Methodology

4.1 Architecture

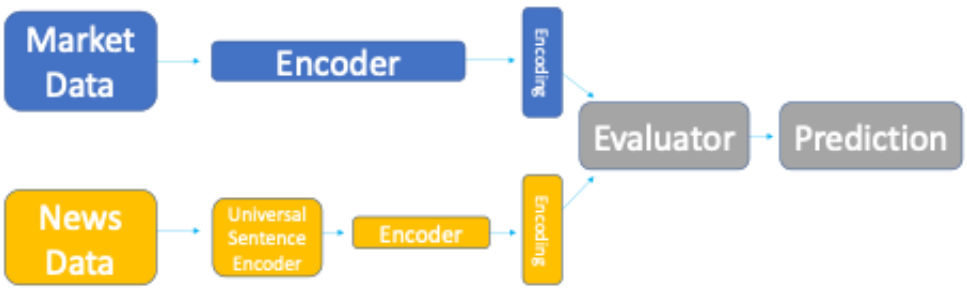

Figure 1: Model Architecture

Our model can broken down into three components, one for each of the two data types we are incorporating, and the final encoder. For the first component, comprised of market data, we take input matrices of size 10 by 12 consisting of closing prices for 12 different tickers and run them through our encoder. For our choice of encoders, we tested 16-dense, 32-dense, 64-dense, 8-LSTM, 16-LSTM, 32-LSTM, and a 2-CNN (3x3x8 convolutional layer followed by 1x1x4).

For our second component, comprised of news data, we take the input matrices of news headlines, of size 25 by 10, and run them first through a universal sentence encoder [2]. The universal sentence

encoder is a pre-trained model that is publicly available in TensorFlow Hub. To feed the matrix into this pre-trained model, we first merge the top 25 headlines for each day into a singular paragraph, thus resulting in a 10 by 1 vector comprised of 10 paragraphs. We then feed this vector into the universal sentence encoder which provides us with a 10 by 512 size encoding for this vector of ten paragraphs. This encoding is numerical and can be further fed into our second encoder. We tested 64-dense, 32-dense, 16-dense, 2-dense, 8-LSTM, 4-LSTM, 2-LSTM, and a 2-CNN as options for this second encoder.

For both components, we also used a "none" encoder where no further processing was done on the input to get a benchmark against which we can compare AUC. For both components, an optimal choice of encoder was made based on testing AUC and the resulting encodings were concatenated into one vector component which was fed into an evaluator. For the evaluator, we have our third component, the final encoder. For this encoder, we tested 1-dense, 2-dense, dropuout only, and batch-norm only. The encoding produced by the optimal encoder was fed into a sigmoid-activated dense output layer to make a prediction of 1 or 0. The latent space dimensionality for each encoder-family is shown in Table 1.

Encoder	Output Dimension
n-Dense	n
n-LSTM	n
2-CNN	2048

Table 1: Encoder Latent Spaces

4.2 Methodology

For training we utilize a Binary Cross Entropy evaluation loss function as shown in Equation 1. Defining a crash as class 1 and not crash as class 0, we wish to have p, the model's prediction as a probability, match the class label value.

$$-(y \cdot \log(p) + (1 - y) \cdot \log(1 - p)) \tag{1}$$

However, in light of the dataset's enormous class imbalance, with a $20 : 1$ ration between positive and negative samples, we evaluate the model with the metric of AUC-ROC. Thus, we are able to assess the overall quality of the model across all possible probability cutoffs.

For testing the encoders, we append a output dense layer, to measure the performance of each model arm. For the overall model, we append the output layer only to the evaluator.

For a given architecture, we train and test 10 independent initializations and average their respective AUC values.

5 Results

In Figure 2 above, wee see that the best performing model is the 64-dense encoder whereas the worst performing model is the 2-CNN.

In Figure 3, we see that the best performing model is the "none" encoder, or a lack of an encoder at all. However, for computational stability, we cannot proceed with a "none" encoder as this would pass on a 5120 dimension encoding in addition to the market data encoding to the evaluator. Thus, we choose the second-best performing model which is a 2-dense encoder. Similar to the market data, we see that the worst performing model is the 2-CNN.

In Figure 4, we see that the best performing model is the dropout only model.

There are several notable trends among this data. First, we remark that the lightest models proved the most successful overall. In the case of the news data, we can primarily attribute this to the quality of the Universal Sentence Encoder embeddings. For the market data, we remark that the overall lack of variability and small time-scope of the data prohibits the use of large models at the risk of overfitting. Furthermore, we see that a CNN is not able to take advantage of the "spacial" information presented

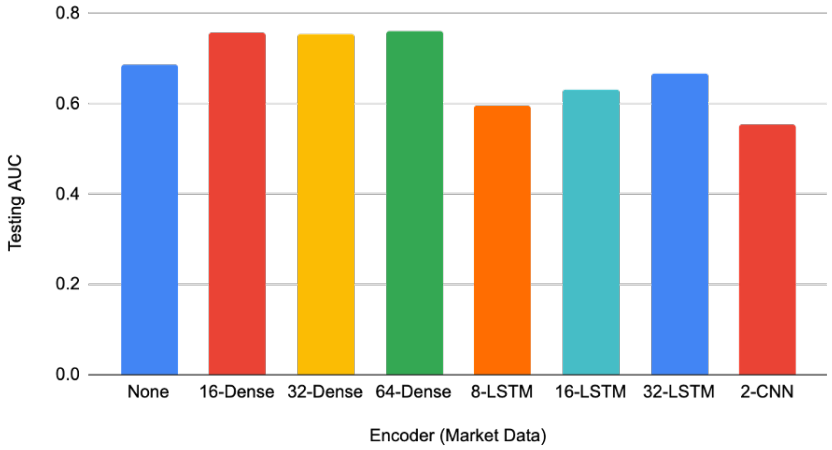

Figure 2: Performance for choice of market data encoder

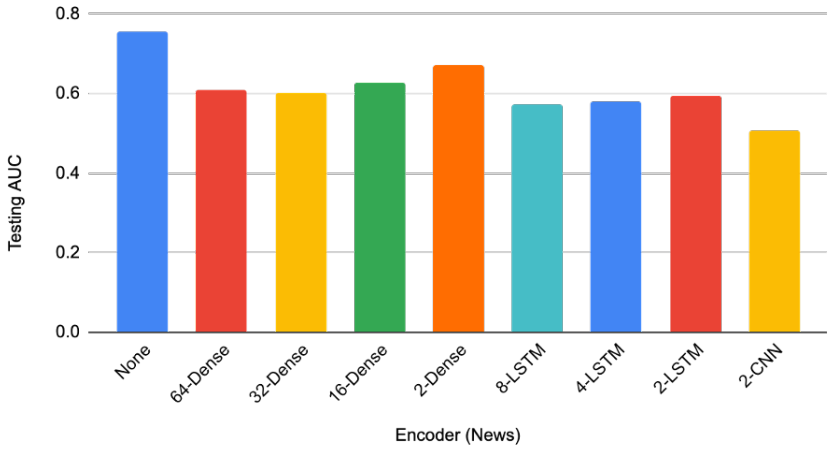

Figure 3: Performance for choice of news data encoder

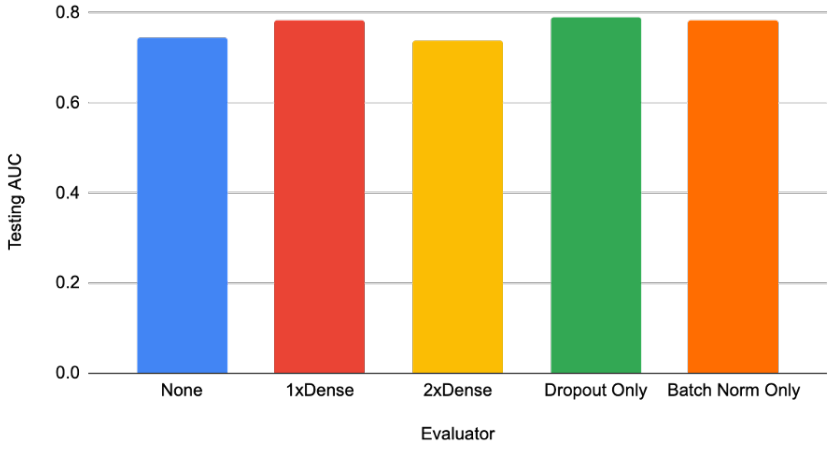

Figure 4: Performance for choice of evaluator

in the case of the financial data in the form of cross index correlations. Further, the LSTM-based encoders also prove to be too heavy but promising due to their natural association with temporal data.

6 Conclusion

In conclusion, we found the best model to be one that uses a 64-dense encoding for the market data, a 2-dense encoding for the news data, and a drop-out only evaluator for making predictions. Our precision metric for determining the best model was AUC (Area under Receiver Operator Curve) and not accuracy. Through the AUC, we will be able to examine the trade off between our model's precision verse recall. Our driving motivation behind not using accuracy as the metric was that given our limited amount of positive (>10%) samples, it is very easy for any model to learn this data and predict with high-accuracy. In fact, a majority of the models we trained had an accuracy greater than 95%. Our findings lead us to believe that there is potential for there to be such a deep learning model that can effectively predict market crashes.

However, being cognizant of the limitations of our dataset, we do want to acknowledge that our model falls short of this goal. One major short coming was the restriction of positive samples, or market crashes, in recent history. Using a 2% decline as the definition of a market crash provided us with utility and enabled for there to be enough positive samples. However, in today's market environment, a 2% decline is "just another Tuesday" and for such a model to have real-world implications, it would need to accurately predict declines on the magnitude of 5% or greater. However, to have enough positive samples to achieve such a task, one would have to analyze 20-30 years of market history. Even in that case, by increasing your sample size, you are further diluting the positive sample rate to be less than 10%.

Further, given the complexity of the modern stock market, to effectively predict a decline in the U.S. for example, one would have to look at the health of markets in India, China, and UK as well. Global economic connectivity has led to increased market volatility as government actions in one part of the world could have massive repercussions for markets in another. Additionally, causes for such market declines transcend just equity markets and include other asset classes such as bonds, options, currency, heavy metals, oil, etc. Future models with drastically greater computing capacities would need to use data for these assets across various global markets for extended periods of time.

Our key contribution to the field is proposing that any such model would benefit from consider qualitative news data in addition to economic, financial data.

References

[1] Aaron7sun. *Daily News for Stock Market Prediction*. Nov. 2019. URL: https://www.kaggle.com/aaron7sun/stocknews.

[2] Daniel Cer et al. *Universal Sentence Encoder*. 2018. arXiv: 1803.11175 [cs.CL].

[3] Sotirios P. Chatzis et al. "Forecasting stock market crisis events using deep and statistical machine learning techniques". In: *Expert Systems with Applications* 112 (2018), pp. 353–371. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.06.032. URL: http://www.sciencedirect.com/science/article/pii/S0957417418303798.

[4] P. Oncharoen and P. Vateekul. "Deep Learning for Stock Market Prediction Using Event Embedding and Technical Indicators". In: *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*. 2018, pp. 19–24. DOI: 10.1109/ICAICTA.2018.8541310.

[5] Stathis Polyzos, Aristeidis Samitas, and Marina-Selini Katsaiti. "Who is unhappy for Brexit? A machine-learning, agent-based study on financial instability". In: *International Review of Financial Analysis* 72 (2020), p. 101590. ISSN: 1057-5219. DOI: https://doi.org/10.1016/j.irfa.2020.101590. URL: http://www.sciencedirect.com/science/article/pii/S1057521920302349.

[6] M. G. Sousa et al. "BERT for Stock Market Sentiment Analysis". In: *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. 2019, pp. 1597–1601. DOI: 10.1109/ICTAI.2019.00231.

[7] M. R. Vargas, B. S. L. P. de Lima, and A. G. Evsukoff. "Deep learning for stock market prediction from financial news articles". In: *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*. 2017, pp. 60–65. DOI: 10.1109/CIVEMSA.2017.7995302.