Building a LSTM Deep Learning Model to predict Yahoo S&P-500 Stock Prices

Sonny Darji*
Marquette University
Milwaukee, Wisconsin
sonny.darji@gmail.com

ABSTRACT

This paper provides a context in determining the best machine learning technique used to predict prices using Yahoo S/P-500 data. The methods in this examples I primarily wanted to focus on were a linear regression method know as Ridge Regression and a predicative model used for both classification and regression models is Gradient Boosting. I wanted to compare these methods as Ridge Regression adds a fixed amount of predication error (ridge) in exchange for a higher accuracy in the model. While Ridge Regression method relies on more inference based approach, Gradient Boosting focuses more on the predicative accuracy and validation is check by the performance of the model on the testing set. This methods were then compared to the performance of using a 7 layer deep learning network model with 2 Convoluted layers, 2 LSTM (Long-Short-Term-Memory) layers and 3 Dense layers with output layer is one as we are looking for the price. The overall results performed showed that Ridge Regression performed the best compared to the LSTM Neural Network Architecture and Gradient Boosting approaches.

ACM Reference Format:
Sonny Darji. 2019. Building a LSTM Deep Learning Model to predict Yahoo S&P-500 Stock Prices. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Artificial Intelligence started with the emerging theory of Turing's proposal to change the questions to ask whether a machine was intelligent, to "whether or not it is possible for machinery to show intelligent behaviour." We live now in a time and world whether the fundamentals of AI are now can be done with a laptop using the help of cloud computing power to create any kind of application. AI can provide impact on almost every industry we look at. Over the past several years, the financial industry has greatly benefited through the use of automated trading bots and using AI to leverage these tools.

Some Background according to Forbes, is that by 2035 AI could boost economic growth and labor productivity by over 40 percent. Large Banks with vasts amount of data to create financial reports with the associated compliance and regulation requirements. These rules provide a great way for software engineering to provide insight in this space with the use of robotic process automation (RPA). RPA can be coded to deal with the rules and cases needed to follow as well through the help of machine learning can be testing and train with validation to gain better knowledge of the data provided.[1]

The input of my Deep Learning model is using from Yahoo stocks database specifically looking over time the stocks for S&P-500 data.[2] The data set is formatted as CSV (comma separated values). Which to make the data more use able. We will not be using the date to predict price rather the Adjusted closing price, which is the price adjusted for Split Stocks, Dividends, and rights offerings according to Investopeida as it determines closer to the true price of the stock which can mean any news of after the trading day is closed still be affected by anything that can effected the demand. [3] The output will be after leveraging adjusted closing price for batches of 32 days of prices to predict the 33rd day being the current day. After building our model we can use the predicated model use the 32 days and predict the 33rd day and next days after from 10 days to 2 years.

2 LITERATURE REVIEW

Shruti Shakhla1, Bhavya Shah, Niket Shah, Vyom Unadkat, and Pratik Kanani wrote a paper for Stock price predication using Multiple Linear Regression using the opening and High price for APPL stock to predict insights on the data. To gauge their performance using Root Mean Squared Error (RSME). This Paper was quite less detailed and a less robust analysis and this analysis could have been done easier and better with R to be able to plot residuals and obtain more graph analysis to the model and analyze it further. A clever and smart use is Least Squares Regression methods which fits a best line with residuals sum of squares minimized. [1]

A paper from Research Center for Social Computing and Information Retrieval in the Harbin Institute of Technology, China This paper also similar to mine used a deep learning mode to predict stock prices but what was different was the use of Event-Driven predication using 10 million events from news data sets like Reuters and Bloomburg using a Event Embedding training process. Using a Deep Convolutional Neural Network (CNN) to perform semantic composition over the input sequence. This approach was clever than my model in regards to incorporate for the mismatch of financial news as a predictor tool for the model.[2]

*

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1]https://www.forbes.com/sites/steveculp/2017/02/15/artificial-intelligence-is-becoming-a-major-disruptive-force-in-banks-finance-departments/2cdc360b4f62-
[2]https://sg.finance.yahoo.com/quote/
[3]https://www.investopedia.com/terms/a/adjusted$_{c}losing_{p}rice.asp$

A paper from Yuan Ze University in Taiwan using Sentiment analysis (SA) of stock news and technical analysis (TA) of trading information. The SA are based on Point-wise mutual information (PMI) which a term expansion method from an multidimensional seed word. The technical analysis and sentiment analysis is combined together in an Support Vector Regression model (SVR) which is an machine learning technique quite clever and different approach. This approach is used over SVM for numerical attributes. One very interesting method of improving performance of the model was the use of Radial basis function (RBF) which is a real-valued function which the input is dependent on the distance from input and some fixed point. [3]

3 DATA SET AND FEATURES

Describe your data set: how many training/validation/test examples do you have? Is there any preprocessing you did? What about normalization or data augmentation? You should also talk about the features you used.

The data set will be used right from the Yahoo finance website; with a specific focus on S&P 500 data from 2014-2019.The features given for the Yahoo data set: [Date, Open, High, Low, Close, Adj. close, Volume], the close price was adjusted for splits and the adjusted close was adjusted for both dividends and splits. We will be using the [Adj. Close] column for our training of our model. We can first look at Table 1 is the descriptive statistics of the Adjusted Closing Price which can be interpreted understand the number of samples and ranges of values our data set is describing.

Perform train/test split such that the first 80% of the prices is the training set and the last 20% of the prices is the test data set. We will train on the training set and test the performance on the test set to see how well we can forecast the price for S&P 500 stock. We create a 2D matrix such that we have 33 columns per example. The training samples will be the first 32 columns with the target variable being the last column. To make this more robust to noise and to leverage past values, we will not use the date to predict the stock prices, but we will use batches of 32 prices where each training example, we leverage 32 prices from the past to predict the price on the 33rd day. We will thus have N32 training examples with N being the number of days we've downloaded for the stock data. We will thus create a 2D matrix of training samples with each row being the prices 32 consecutive days in the past with the price to predict being the current day. Our X variable will represent 32 days in the past ; and the Y variable will represent the 33rd day as our output value. We format this using python libraries and our variables are X_train , y_train, X_test , y_test. Also, Neural Networks learn better when the data is normalized so we will normalize the data to the [1,1] range by using Scikit-learn's MinMaxScaler.[7] Therefore, we need to find apply this scaling on all of the price data, then decompose it into the training and test sets again.

4 METHODS

My goal of this project is to compare the use of different learning algorithms, specifically focusing on the using Python's sklearn package between Ridge Regression for the training and test set. The results were that the training and testing set matched very well and accurately using ridge regression. Figure one depicts the

Table 1: Descriptive Statistics for S&P-500 data from 2014-19

Describe	Adj. Closing Price
count	1258
mean	2383.771573
std	,333.912890
min	1829.079956
25%	2081.884888
50%	2350.175049
75%	2711.667480
max	3025.860107

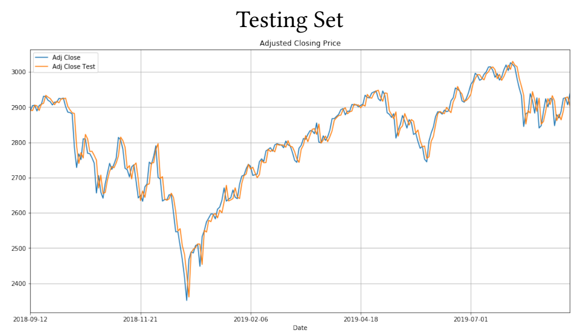

Figure 1: A line graph depicting Testing Set after when using Ridge Regression.

Training set when performing it on the model. Figure two depicts the testing set after performing ridge regression. Ridge Regression is a approximation algorithm for finding "best" solution for an equation with no unique solution. [5]

Figure 2: A line graph depicting Training Set after when using Ridge Regression.

Another method used as a comparing learning algorithm was Gradient Boosting Trees. Gradient Boosting Trees involves three things 1. Loss Function depends on the type of problem solved. 2. Weak Learner , are described as decision trees as the weak learner in boosting. 3. Additive Model states that trees are added one at a time and existing trees cannot be changed. [6] Figure one depicts

Testing Set-(gradient boosting trees)

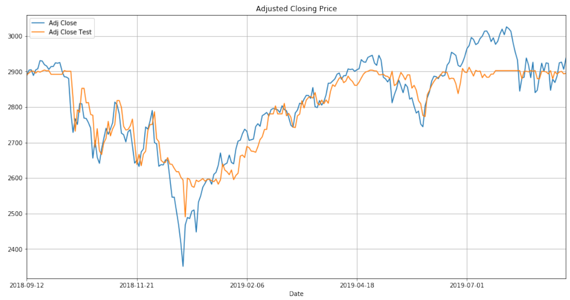

Figure 3: A line graph depicting Testing Set after when using Gradient Boosting Trees.

Testing Set-(gradient boosting trees)

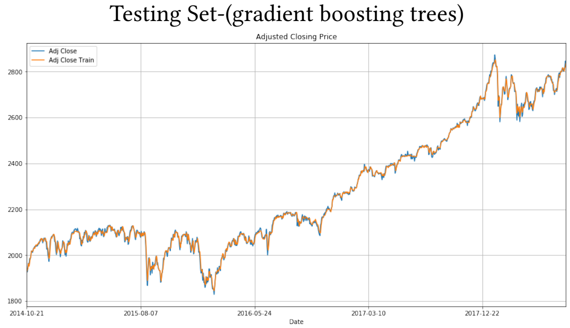

Figure 4: A line graph depicting Training Set after when using Gradient Boosting Trees.

Table 2: Architecture of CNN Model

Layer	Output Shape	Param #
Conv1D	(32, 128)	768
Conv1D	32, 128	82048
LSTM	(None, 128)	131584
LSTM	(None, 128)	131584
Dense	(None, 64)	8256
Dense	(None, 64)	4160
Dense	(None, 1)	65

the training set when using the python sklearn GradientBoostingRegressor algorithm. [7] Figure two also depicts the testing set when used by the learning algorithm.

The final method of comparison is using Long short-term memory (LSTM). My goal is to build a neural network based one using Keras / Tensor flow.

We use a stack of two convolutional layers with output shape of 31 by 128. LSTM or long short term memory is an artificial recurrent neural network architecture is different from a typical neuron cell. This neural unit uses an input, output gates but what is different is the use of a forget gate.

then followed by one LSTM layer with output shape of 31 by 138 and next LSTM layer used a 0 by 128 as the output shape. Lastly with three Dense layer of 0 by 64 for the first and second layers' output shape and the last dense layers output shape is 0 by 1 tensor as we are looking for predicated price. Table 1 describes the type of layer used for our seven layer model as well as the input shape, and number of parameters.[4] The activation function chosen for the model is a tanh function. A tanh function which goes from the range -1 to 1. the optimizer used for training this model that was chosen was the SGD optimizer.

5 RESULTS

5.1 Validation Loss

Since we know from the training of the model. The data shows that the validation loss was little bigger than the learning rate which shows some chance of over fitting may occur. The loss function I decided to use for this model was Huber Loss. Huber loss is used in robust regression, that is less sensitive to outliers in data than squared error loss. Figure 5 shows the validation loss for the trained model over time (epochs).

Validation Loss

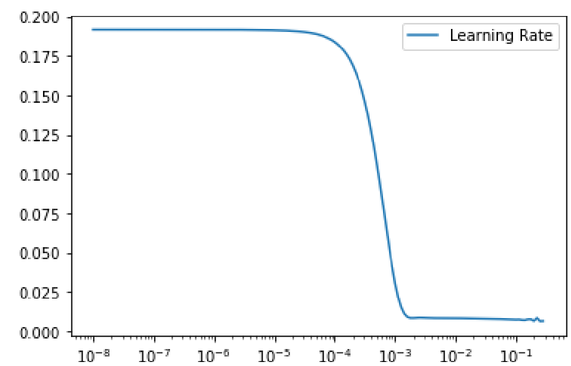

Figure 5: A line graph depicting validation loss for the model.

Learning Rate for model

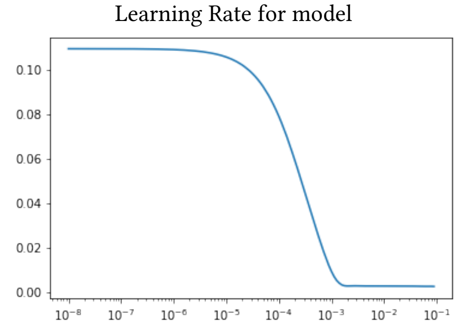

Figure 6: A bar graph depicting Learning over time in the model.

5.2 Learning Rate

I used keras LearningRateScheduler to help find the optimal learning rate after training.[11] This is a technique to find the optimal learning rate for the neural network. Specifically we gradually increase the learning rate at each epoch, recording the loss for each then choosing the learning rate that has the smallest loss overall. The most optimal learning rate found was 0.001. Figure 6 describes the Learning rate for the model over the 150 epochs.

5.3 Training/Testing results for model compared to original

Training Set

Figure 7: A line graph depicting training set for the model.

My Training set results after training the model compared to the original data set. The blue line indicates the original adjusted closing price of the data set and the orange line shows the trained model data results. In comparison to the original the training model performed quite well to get the overall trend of the data set.

Testing Set

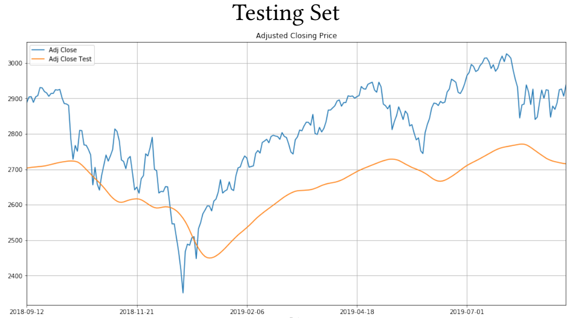

Figure 8: A line graph depicting Testing Set for the model.

My Testing set after training the model comparing to the original data set. Figure 8 depicts the results of the testing set after the model has performed. The results show that the model performed well but decided to undershoot the original given values which is a interesting observation.

5.4 All Learning methods visualized compared to original

My Fourth Visualization was to compare all the learning methods to the S&P-500 original data set I choose. Figure 9 depicts the Adjusted closing price in a blue line, Ridge Regression as orange line, Gradient Boosting as green line, and our deep learning model shown in red.

Adjusted Closing Price for all Learning Methods

Figure 9: This plots a line plot comparing the original adjusted closing price (blue), Ridge Regression (orange), Gradient Boosting (green), and LSTM (red) for the testing set.

6 DISCUSSION

Figures 1&2 showcase the training/testing results when ridge regression algorithm is applied. The ridge regression performed the best due to introducing a fixed amount of predication error gave the opportunity for the accuracy of the model to be quite high. As the graph shows the orange are neck in neck in par with the original data set results. the alpha for our ridge regression model was 0.001.

Figures 3&4 represent the performance of the training/testing set of the model by method of gradient boosting trees. which focuses on predicative accuracy didn't perform as well as the other learning methods. It is managed to perform best for training set; but the testing set beginning of the data was overshoot and under predicated the end region of the data.

Figure 5 depicts the performance of model learning by 7-layer deep learning network shows the validation loss which describes the model.

Figure 6 depicts the learning rate for the model. The optimal learning rate chosen for this model was 0.001 we used this to train for another 100 epochs with same optimizer and metrics.

Table 2 describes the architecture parameters used for the building of the model. Our model consists of two Conv1D layers with output shape of (32,128), and two LSTM layers, and followed by three dense layers with the last layer of output shape of 1. SGD stands for Stochastic Gradient Descent. The optimizer chosen for the model was SGD as an iterative method which is used to find the values of the parameters of a function that minimizes the cost function as much as possible.[14] Our performance measurement for our model used was mean absolute error which measures the difference between two continuous variables.

Figure 7&8 depict the result for the training set and testing set which are represented as orange lines. The training set results tell us that our model performance quite accurately. The testing set results show that the variance is a more than training and looks like the overall trend was captured but the predictions made were lower than the original.

Figure 9 describes the comparison of all the learning methods explored in this example using the S&P-500 data set with our variable as adjusted closing price. The performing learning methods were ridge regression and the LSTM deep learning model.

7 CONCLUSIONS

Overall, there can be a lot extracted from the S&P-500 data set . This data set accounts for 80% percent of the US market capitalization.[16] This stock is a conglomerate of over 50 large corporations. It in a invaluable learning experience to research whether we can accurately predict stock price.

The main ideas to gain from this study are the analysis of three different machine learning techniques: Ridge Regression, Gradient Boosting, and the use of a CNN-LSTM deep learning model. These methods were each unique, ridge regression performed the best and coming in a close second was the deep learning model. I thinks the results of my model make sense as ridge regression parameters cannot be zero which is good for financial data as the value wouldn't reach zero unless the corporation was no more but in the case of the S&P-500 only the top largest corporations are factored into the market value of the stock.

All in all, given more time to research I would love to implement a Recurrent Neural Network (RNN) with the memory coming from a sentimental analysis of financial news. The RNN would use time series stock data to train. To make this more feasible possible would need more higher computational power to implement a robust model taken from thousands if not a million news articles to learn from. I would also like to explore the concept of transfer learning. Transfer learning provides the concept of training the model and instead of creating an output layer we feed this predicated model as the learning models' knowledge to be use and stored and applied to a different but related problem.[17]

REFERENCES

[1] Shruti Shakhla "Stock Price Trend Prediction Using Multiple Linear Regression "International Journal of Engineering Science Invention (IJESI), vol. 07, no. 10, 2018, pp 29-33

[2] Ding, Xiao, Zhang, Yue, Liu, Ting, AND Duan, Junwen. "Deep Learning for Event-Driven Stock Prediction" International Joint Conference on Artificial Intelligence (2015): n. pag. Web. 28 Nov. 2019

[3] Wu, Jheng-Long, et al. "Stock Price Predication Using Combinational Features from Sentimental Analysis of Stock News and Technical Analysis of Trading Information ." Stock Price Predication Using Combinational Features from Sentimental Analysis of Stock News and Technical Analysis of Trading Information , Department of Information Management, Yuan Ze University, 2011, www.ipedr.com/vol55/008-ICEBM2012-K00016.pdf.

[4] ando, koki. "koki25ando/NBA-Franchise-Record." GitHub,2016.

[5] Ridge Regression. Brilliant.org. Retrieved 11:33, November 30, 2019, from https://brilliant.org/wiki/ridge-regression/

[6] Brownlee, Jason. "A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning." Machine Learning Mastery, 21 Aug. 2019, machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning.

[7] "Sklearn." PyPI, 2015, pypi.org/project/sklearn/.

[8] "Sklearn.linear_model.Ridge." Scikit, scikit learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html.

[9] "Matplotlib." Pythonspot, 31 May 2016, pythonspot.com/matplotlib/.

[10] Collet. "Keras: The Python Deep Learning Library." Convolutional Layers - Keras, keras.io/layers/convolutional/.

[11] chollet2015keras, title=Keras, author=Chollet, François and others, year=2015, howpublished=https://keras.io,

[12] "TensorFlow." TensorFlow, Google, 2016, www.tensorflow.org/.

[13] Collet. "Keras: The Python Deep Learning Library." Home - Keras Documentation, keras.io/losses/.

[14] Rahul_RoyAn ordinary kid with a passion for coding, et al. "ML: Stochastic Gradient Descent (SGD)." GeeksforGeeks, 17 Feb. 2019, www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/.

[15] "Long Short-Term Memory." Wikipedia, Wikimedia Foundation, 1 Dec. 2019, en.wikipedia.org/wiki/Long_short-term_memory.

[16] Staff, Motley Fool. "What Is the Samp;P 500?" The Motley Fool, The Motley Fool, 22 June 2016, www.fool.com/knowledge-center/what-is-the-sp-500.aspx.

[17] Sebastian Ruder. "Transfer Learning - Machine Learning's Next Frontier.", 16 Apr. 2019, ruder.io/transfer-learning/.

A INTRODUCTION

B LITERATURE REVIEW

C METHODS

C.1 Ridge Regression

C.2 Gradient Boosting Trees

D RESULTS

D.1 Validation Loss

D.2 Learning Rate

D.3 Training/Testing Set

D.4 All Learning Algorithms results

E DISCUSSION

F CONCLUSIONS

G REFERENCES