STOCK PREDICTION

Leon Hayden

INTRODUCTION

The lack of a viable simple to use stock predictor is hurting the middle class. It is leaving the opportunity to help build individual wealth left on the sideline. As either being too complex or out of the reach of a normal person by them feeling that they do not have the needed resources to compete. It is my intention to design a stock predictor with Artificial Intelligence that will help the common lay person. Being able to help them to decide how much and what stock has the best possible return using previous stock price data. I hope to enlighten and show one way to use previous data to help increase the chance of having a successful portfolio.

Primary Goals

- **Introduction of Artificial Intelligence to regular lay person**
  - **Artificial Intelligence invokes a sense of overwhelming misunderstanding of what It can do, and how it can help improve the quality of life. I will show how AI can help with building a portfolio of stocks with signs as to when to purchase and sell.**
- **Introduction of methods to build wealth using AI.**
  - **Using Recurrent Neural Networks [RNN]**
  - **Different models have a broad base of indicators to give a better overview of the entire market.**
  - **Different models have different rates of return that can be capitalized.**
  - **Combining models can highlight positives and reduce negatives.**
- **Developing input that will allow inputs.**
  - **Stocks**
  - **Dates of up to thirty years in arrears**
  - **Date of prediction**
  - **Moving average**
- **Prediction of what price will be at N+1.**
  - **Tell what the final closing price will be.**
  - **Tell what intraday trading price will be.**
- **Developing prediction based RNN model.**
  - **Long Short – Term Memory [LSTM]**
  - **Light GBM**
  - **LSTM – XGBoost Hybrid**

Developing the above criteria will give a lay person a good overview of what AI is and how it can be developed to help with making buying and selling decisions.

Methodology

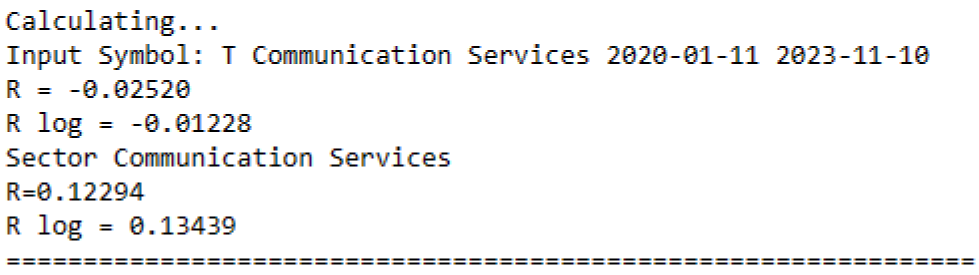
It was decided to use adjusted clothes since it gives a better overall reading of prices. Includes stock splits, dividends, and other adjustments[1].

- **Corporate actions**
    - **Represents stock splits better.**
- **Consistency**
- **Dividend Reinvestment**
    - **Account for a more realistic view of total return on investment**
- **Comparison with Indices**
    - **Allows for uniform comparison for different indices.**
    - **Accurate comparison for stock and indices**

To fill out the missing data from days NaN were removed. Data was normalized based on dates entered.

To determine r and r log for the stock and the sector of the stock for which it lies in. The S&P 1500 was used. This indice contains the S&P 500, S&P Mid-cap, S&P 400 Small –cap. Being a broad base indicator, over 90% of the stocks on the New York Stock Exchange[2]. Nasdaq is covered by this index.

Figure 1



```
Calculating...
Input Symbol: T Communication Services 2020-01-11 2023-11-10
R = -0.02520
R log = -0.01228
Sector Communication Services
R=0.12294
R log = 0.13439
=================================================================
```

This has the stock symbol and dates of the investigation along with the R and R Log of the dates. Sector information and sector R as well as R log is included as a comparison tool to view how the security performed as comparison to its sector.

[1] **Adam Hayes**, "What is closing price? Definition how it's used and example", *Investopedia*

[2] *S&P1500 Composite*

Technical Analysis

Technical Analysis is a method used to evaluate and predict the future price movements of financial assets, such as stocks, currencies, commodities, and indices. Unlike fundamental analysis, which focuses on examining a company's financial health, earnings, and other quantitative factors, technical analysis relies on studying historical price and volume data to identify patterns and trends.

Technical Analysis is being applied to give the user more information to help with the decisions concerning when to buy and when to sell. included are Bollinger Bands (Bollinger, 2021), Moving Average Convergence [MACD] (Dolan, 2023), and Exponential Moving Average (Chen, 2023).

Bollinger Bands

- **Have a centerline and two price channels of bands above and below it.**
- **Centerline is a simple moving average while the price channels are standard deviations of the stock being studied.**
- **The bands expand and contract as price action becomes volatile or bound into a tight trading pattern.**
- **Bands are designated upper and lower as price targets when drawing the bands.**
- **When the price continually touches the upper band, it can show an overbought signal while continually touching the lower band indicates an oversold signal.**

Moving Average Convergence

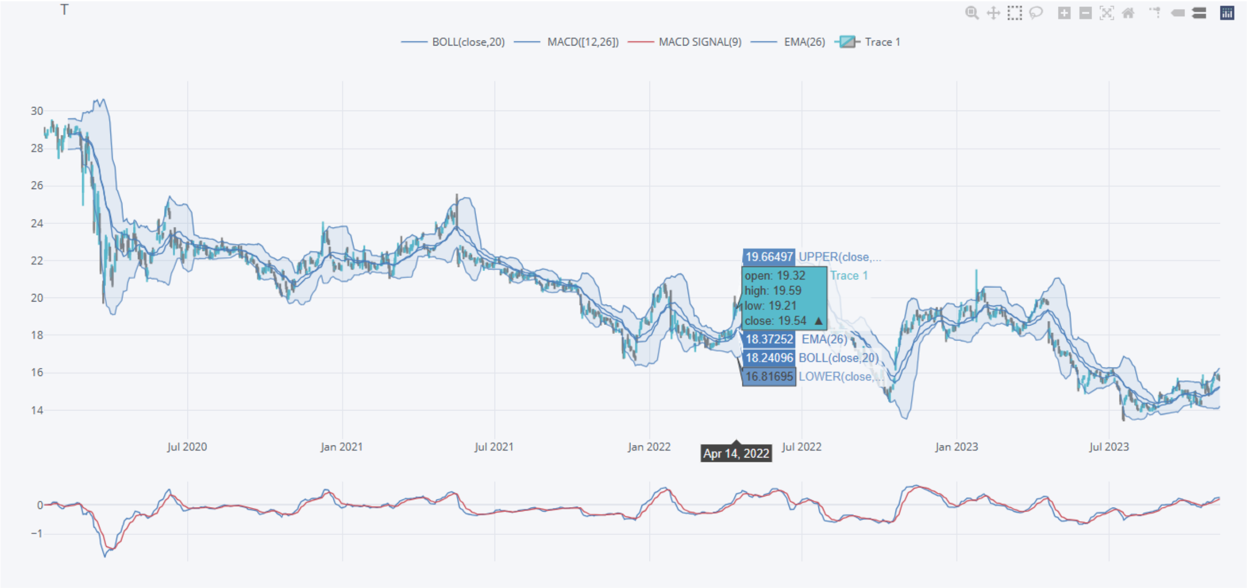
- **A momentum oscillator is primarily used to trade trends.**
- **MACD triggers technical signals when the MACD line crosses above the signal line (to buy) or falls below it (to sell).**
- **MACD can help gauge whether a security is overbought or oversold, alerting traders to the strength of a directional move, and warning of a potential price reversal.**
- **MACD can also alert investors to bullish/bearish divergences (e.g., when a new high in price is not confirmed by a new high in MACD, and vice versa), suggesting a potential failure and reversal.**
- **After a signal line crossover, it is recommended to wait for three or four days to confirm that it is not a false move.**

Exponential Moving Average

- **Exponential Moving Average (EMA) is like Simple Moving Average (SMA), measuring trend direction over a period. However, whereas SMA simply calculates an average of price data, EMA applies more weight to data that is more current. Because of its unique calculation, EMA will follow prices more closely than a corresponding SMA.**

- **Like all moving averages, this technical indicator is used to produce buy and sell signals based on crossovers and divergences from the historical average.**
- **Traders often use several different EMA lengths, such as 10-day, 50-day, and 200-day moving averages.**
- **EMA to determine trend direction, and trade in that direction. When the EMA rises, you may want to consider buying when prices dip near or just below the EMA. When the EMA falls, you may consider selling when prices rally towards or just above the EMA.**
- **Moving averages can also indicate support and resistance areas. A rising EMA tends to support the price action, while a falling EMA tends to provide resistance to price action. This reinforces the strategy of buying when the price is near the rising EMA and selling when the price is near the falling EMA.**

Figure 2

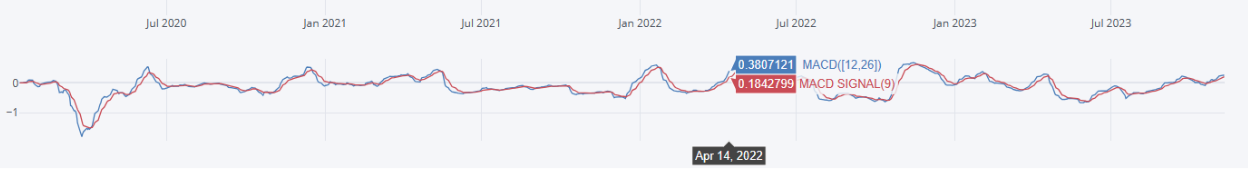


This is the combined graph for Bollinger Bands {Boll}, Moving Average Convergence and Exponential Moving Average

For April 14,2022 for AT&T {T}, the stock opened at 19.32, high for the day 19.59, low for the day 19.21 and closed at 19.54.

Boll indicated before and after the date a narrowing of the bands suggests that volatility is decreasing, showing that the price will be stable. Not a selling or purchasing sign indicator currently.

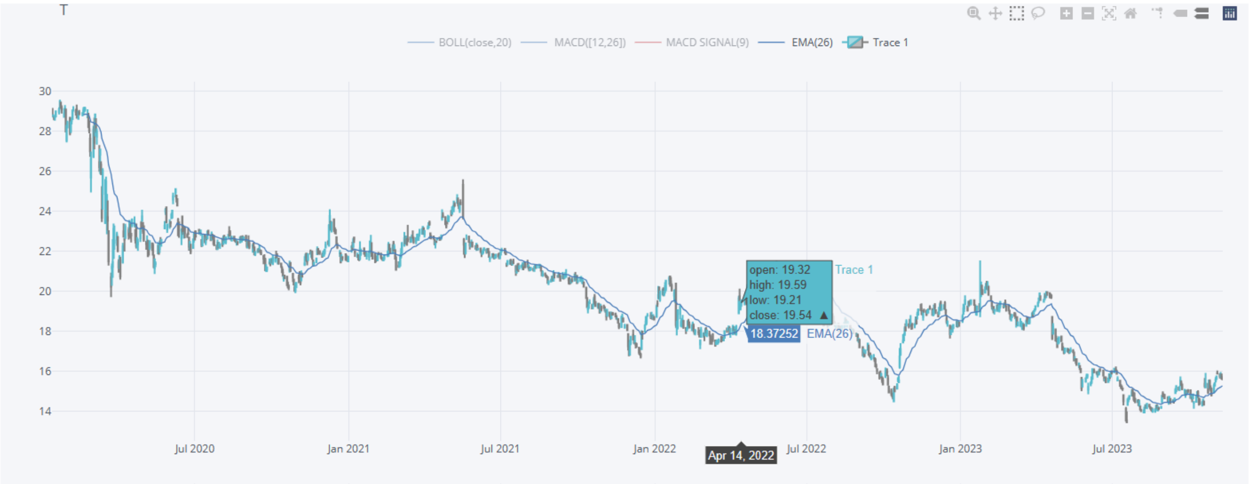
Figure 3



MACD

- In this example we can see where MACD has crossed the zero line, and has turned up from below zero, this is bullish.
- Overall, this indicator shows for this stock during this time frame, this is buy if both MACD and Boll have good indicator. Which is inconclusive at this point. It might be better to wait before taking a position. Since one says buy and Boll says wait for now.

Figure 4



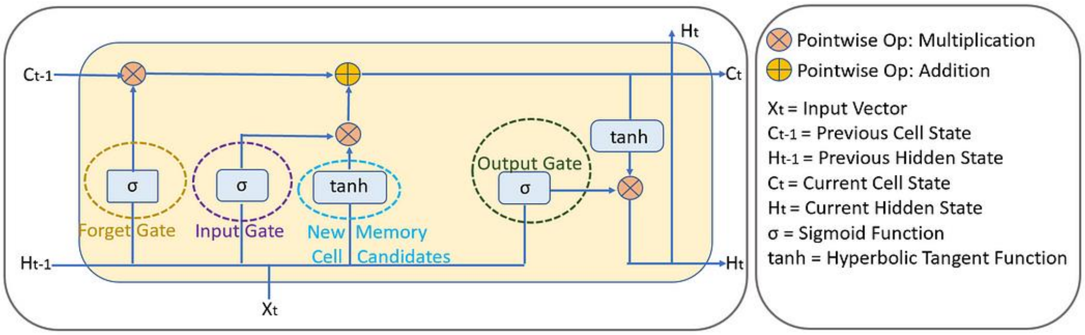
EMA

- Follows prices much better than Simple Moving Average
- Shows a smoother graph with less noise.
- When tied into MACD and Boll gives good signs as to when to sell or buy.

Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to capture and learn patterns in sequential data. They are particularly effective at handling sequences of data, making them well-suited for various tasks, including natural language processing, speech recognition, time series forecasting, and more. LSTM networks work by addressing the limitations of traditional RNNs, which struggle with capturing long-term dependencies in data due to the vanishing gradient problem.

Figure 5


C_{t-1}
H_t
C_t
tanh
Output Gate
σ
σ
σ
tanh
Forget Gate
Input Gate
New Memory
Cell Candidates
H_{t-1}
H_t
X_t
Pointwise Op: Multiplication
Pointwise Op: Addition
X_t = Input Vector
C_{t-1} = Previous Cell State
H_{t-1} = Previous Hidden State
C_t = Current Cell State
H_t = Current Hidden State
σ = Sigmoid Function
tanh = Hyperbolic Tangent Function

(Tourloukis, 2021)

- **Three Gates: The core innovation of LSTMs is the presence of three gates within each LSTM cell: (J., 2020)**
  - **Forget Gate: This gate decides what information from the previous hidden state should be thrown away or kept. It calculates a "forget score" for each element in the previous hidden state and multiplies it by the corresponding elements to forget or retain information.**
  - **Input Gate: This gate determines what new information is added to the hidden state. It calculates an "input score" for each element and updates the hidden state accordingly.**
  - **Output Gate: The output gate decides what information is passed to the output. It calculates an "output score" for the current hidden state and produces the output at the current time step.**
- **Memory Cell: LSTMs maintain a memory cell within each cell, which can store and manipulate information over long sequences. The memory cell helps in capturing long-term dependencies by allowing information to flow through the gates and be retained or discarded over multiple time steps. (J., 2020)**
- **Activation Functions: Inside the gates and the memory cell, LSTMs use activation functions, such as the sigmoid and hyperbolic tangent (tanh) functions. These functions introduce non-linearity and allow the LSTM to model complex patterns in data. (J., 2020)**

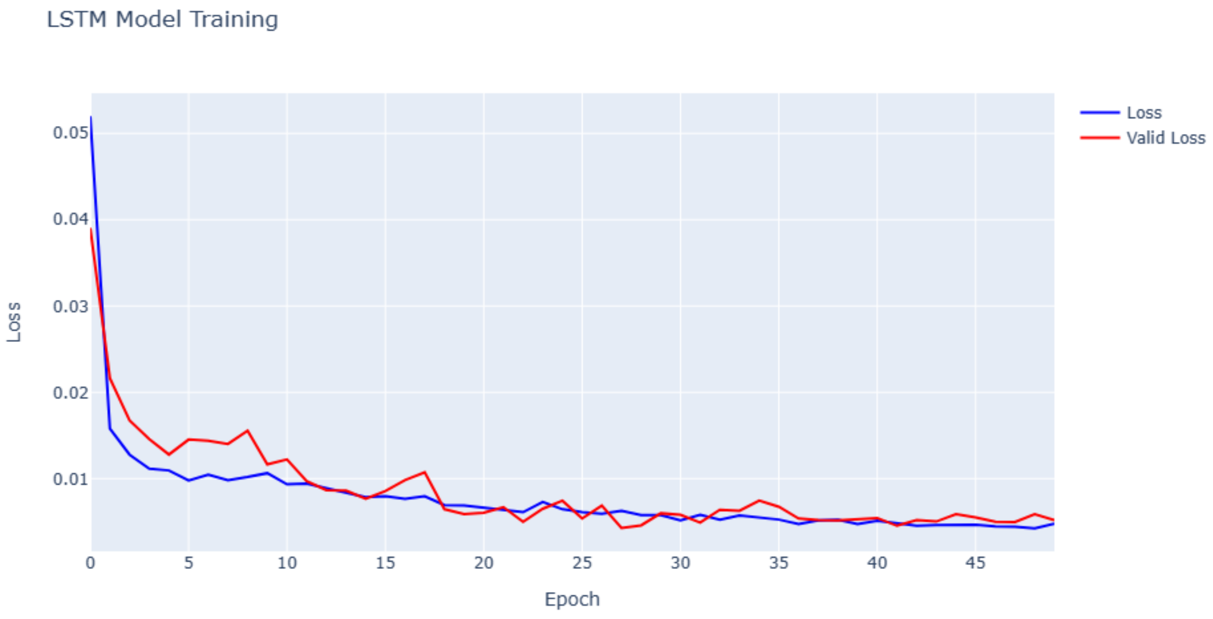
- **Backpropagation Through Time:** LSTMs are trained using the backpropagation through time (BPTT) algorithm, which calculates gradients to update the model's parameters (weights and biases) based on the error between predicted and actual outputs. The use of the gradient is what enables the network to learn from data. (J., 2020)
- **Long-Term Dependencies:** The presence of the forget gate, input gate, and memory cell allows LSTMs to effectively manage and propagate information over long sequences. The network can learn to remember valuable information over many time steps and forget irrelevant information. (J., 2020)
- **Variable Sequence Length:** LSTMs can handle sequences of varying lengths for both input and output, making them versatile for different applications. (J., 2020)

Testing was using 80% of available data, training used the remainder. Features were reshaped for LSTM input.

Epochs were set at 50 to have reasonable run times.

Data was normalized.

Figure 6

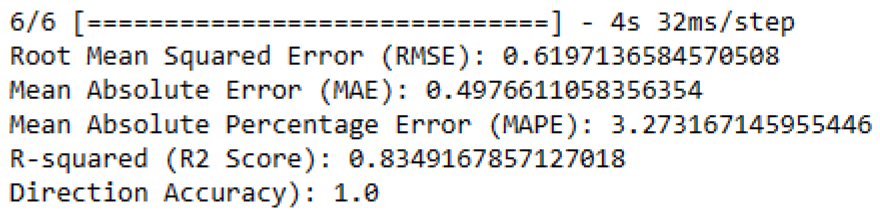


A concern I had was the scaling on the left-hand side of the graph. At first, I was concerned that it was overfitting. Looking closer I was able to find out that the scaling was smaller than what was expected. After closer inspection, the graph is good.

Measurement Metrics

Metrics show a way to test how the predictive model works over actual data. The same metrics were applied to all models for a comparison.

Figure 7 [Data Frame 01-11-2020 to 11-10-2023; Stock AT&T]

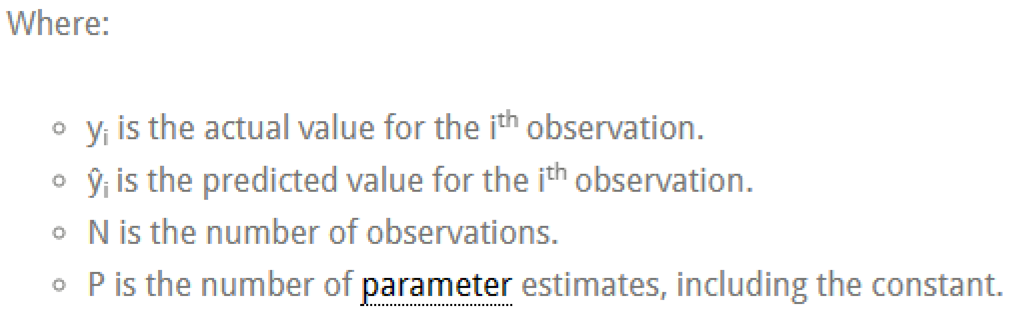


```
6/6 [==============================] - 4s 32ms/step
Root Mean Squared Error (RMSE): 0.6197136584570508
Mean Absolute Error (MAE): 0.4976611058356354
Mean Absolute Percentage Error (MAPE): 3.273167145955446
R-squared (R2 Score): 0.8349167857127018
Direction Accuracy): 1.0
```

<u>RMSE:</u> Measure of average difference between predicted values and the actual values. Can also be defined as standard deviation of the residuals. With residuals being distance between regression line and data points.

Formula used[3]

$$RSME = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{N - P}}$$



Where:

- y_i is the actual value for the ith observation.
- $\hat{y}_i$ is the predicted value for the ith observation.
- N is the number of observations.
- P is the number of parameter estimates, including the constant.

Since the value is extremely low this is a good indicator, based on the size of the data frame. The model can predict house prices accurately. A higher number tells predictions are more widely spread away from observations.

[3] Jim Frost, “Root Mean Square Error (RMSE),” in Statistics by Jim.

MAE: is a measure of the average magnitude of the errors between predicted values and observed values. It is a loss function commonly used in machine learning and statistics to evaluate the performance of regression models.

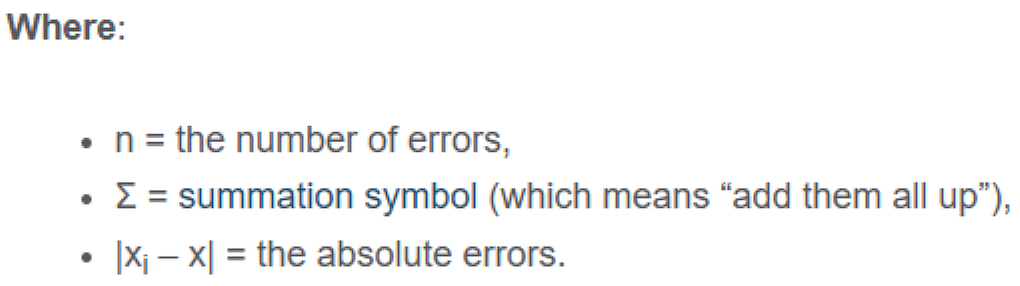
The lower the MAE, the better the model fits the data frame.

Higher the MAE, data frame does not fit the model well.

MAE is a more robust measure of error than mean squared error (MSE) because it is less sensitive to outliers. This is because MSE squares the differences between predicted and observed values, which can amplify the impact of outliers. MAE, on the other hand, simply takes the absolute difference, which is not affected by outliers.

Formula used[4]

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} | x_i - x |$$



Where:

- n = the number of errors,
- Σ = summation symbol (which means “add them all up”),
- $|x_i – x|$ = the absolute errors.

The value being less than one indicates that this model fits the data frame quite well. If it was greater than one it says that either the data or the model needs to be fine-tuned for a better fit.

[4] “Absolute Error & Mean Absolute Error,” in Statistics How To

<u>MAPE:</u> Between the predicted and actual values, is a measure of the accuracy of forecasting methods. It is a relative error metric that expresses the average magnitude of the absolute percentage errors between predicted and actual values. MAPE is often used in situations where the scale of the data varies, as it allows for meaningful comparisons between different models or time periods.

A lower MAPE indicates a more accurate prediction. A higher MAPE indicates a less accurate prediction. A MAPE of 0% indicates that the predictions are perfectly accurate.

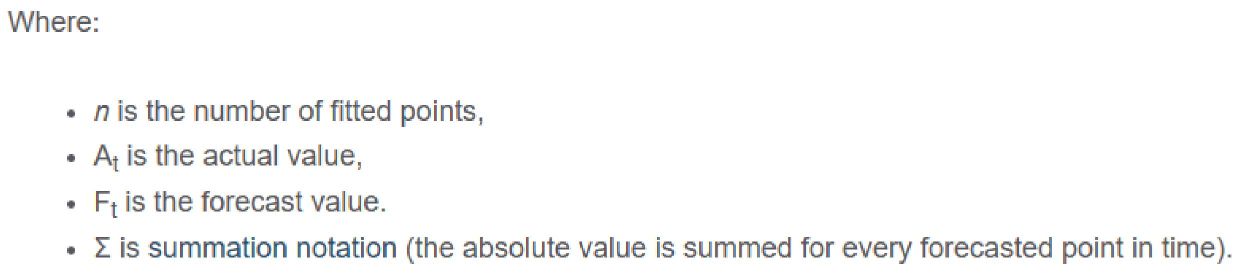
MAPE is a popular choice for evaluating forecasting models because it is easy to understand and interpret. However, it is important to note that MAPE can be sensitive to outliers, and it can be difficult to compare MAPE values across different datasets with different scales.

MAPE is used to evaluate the accuracy of stock price forecasts. A lower MAPE indicates that the forecasts are more accurate.

Good for comparing accuracy within data frames.

Formula used[5]

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$



Where:

- *n* is the number of fitted points,
- A_t is the actual value,
- F_t is the forecast value.
- Σ is summation notation (the absolute value is summed for every forecasted point in time).

As a rule of thumb, a number less than 10% shows that this model is a good fit for its data.

As a conclusion, metrics can show that the best model is the one with the lowest numbers. Based on criteria that were given in the last three measurable metric categories.

[5] **<u>“Mean Absolute Percentage Error,” in Statistics How To</u>**

R-squared: The proportion of the variation in the dependent variable that is predictable from the independent variable. R-squared is often interpreted as a measure of the goodness of fit of a regression model. A higher R-squared indicates a better fit of the model to the data. However, it is important to note that R-squared does not necessarily indicate that the model is a good predictor of unseen data. For example, an R-squared of one can be obtained by simply fitting a line that passes through all the data points, but this model would not be a good predictor of unseen data.

For our use, a better understanding of this would be the actual price over price predictions.

Formula used[6]

$$R^2 = 1 - \frac{Actual\ Prices}{Predictions}$$

The closer the value is to one. The better the fit is to predictions. In this case since the value is at .85 predictions are running remarkably close to actual prices.

[6] Jason Fernando, "R-Squared: Definition, Calculation Formula, Uses, and Limitations ," Investopedia.

Directional Accuracy: A measure of how often a forecasting method correctly predicts the direction of change in a time series. It is a popular metric for evaluating the performance of forecasting models, particularly in finance.

Refers to the ability to predict the future direction of stock prices, trends, or asset values. Shows degree of precision or correctness in finding a specific direction for a time series. Simply put tells if trend is going up or down.[7]

Formula used[8]

$$\frac{1}{N}\sum_{t}\mathbf{1}_{\operatorname{sgn}(A_t - A_{t-1}) = \operatorname{sgn}(F_t - A_{t-1})}$$

Where

A_t* is the actual value at time *t

***F_t* is the forecast value at time *t*.**

Variable *N* represents the number of forecasting points.

The function sgn( ) is sign function and 1 is the indicator function.

Since the value is going to one. Shows that this stock is on the upswing.

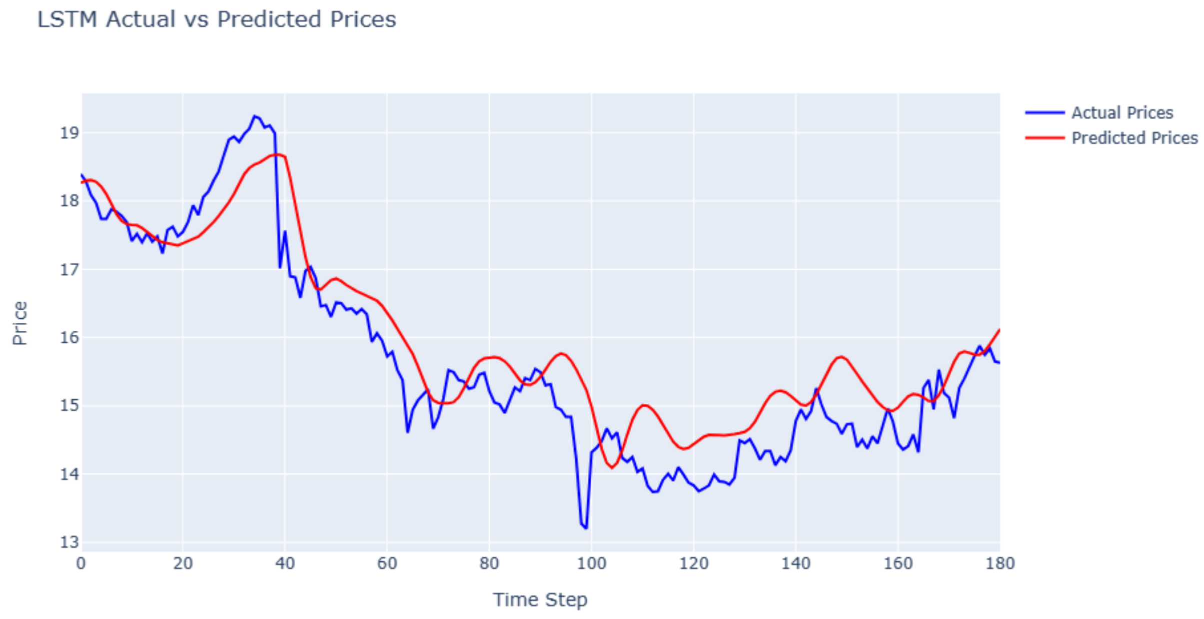
Summary: This model has great individualities that make it a great tool for giving out suggestions as to when to buy or sell and individual security. The metrics for this model put it right inline as when compared to established norms for the security market.

[7] **"Mean directional Accuracy" Wikipedia**

[8] **"Mean directional Accuracy" Wikipedia**

Predictions

Figure 7



At time step 20 the actual price is 17.54778

At time step 20 the predicted price is 17.3812

Happy with the results! Based on adjusted close

Rates of Return

A way to measure the effectiveness of AI based on the time frame entered.

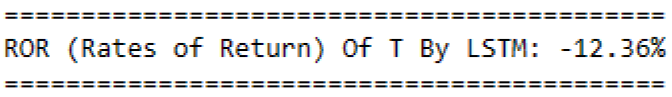
Formula Used (Kenton, 2023)

$$Rate\ of\ return = \left(\frac{Current\ Value\ of\ Investment - Initial\ Cost\ of\ Investment}{Initial\ Cost\ of\ Investment}\right) x\ 100$$

Where

Current Value of Investment: The current worth of the investment, including any capital gains or losses.

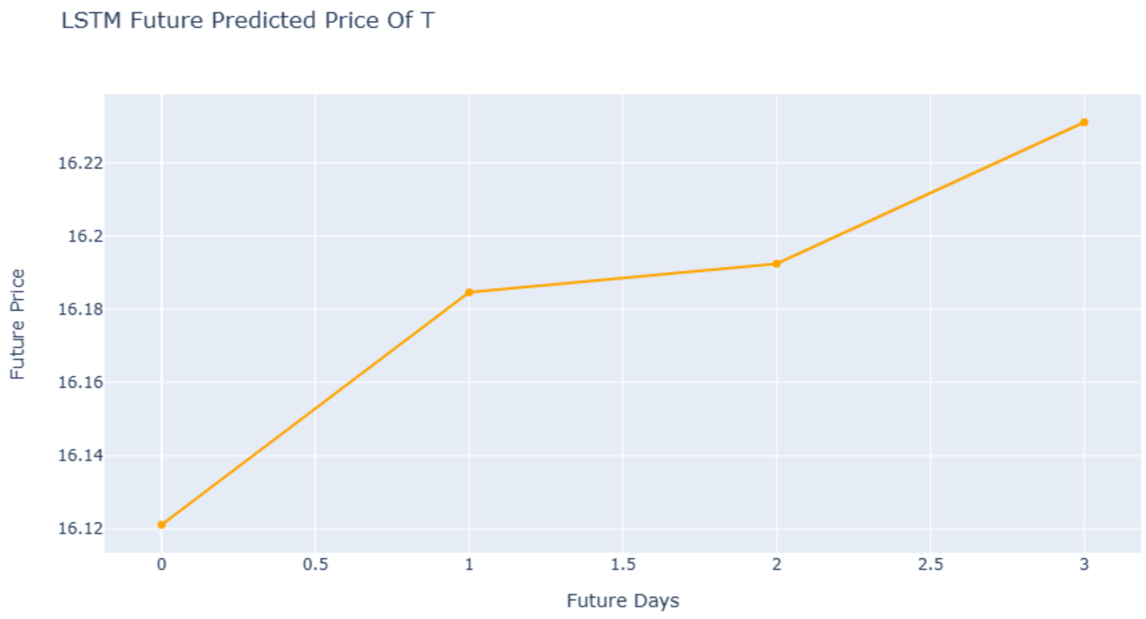
Initial Cost of Investment: The original amount of the stock at start



```
============================================
ROR (Rates of Return) Of T By LSTM: -12.36%
============================================
```

Based on data frame period.

Figure 8



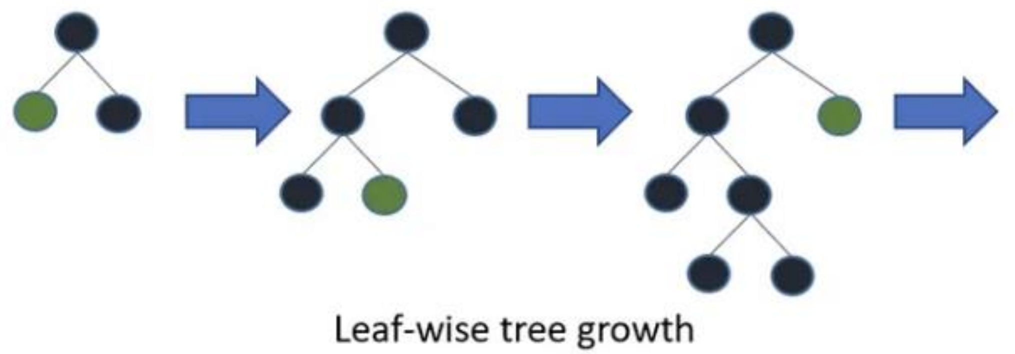
This shows what the price will be at N+1 days. This graph was incredibly good for predition of intraday pricing. "Future Days" tells at .5 into day one after the last date for the data frame. Says at or around noon the price will be just under 16.16 USD.

Having this level of information, can have the lay person enter a buy or sell limit order with a time frame of, "good til close of market". Maximizing the clients purchasing or selling power.

Light Gradient Boosting Machine (LGBM)

LGBM (Gradient Boosting Machine) is a high-performance, distributed machine learning framework that falls under the category of ensemble learning. It is designed for speed and efficiency, making it particularly useful for large datasets and complex tasks. Uses a gradient boosting framework that uses a tree-based learning algorithm.

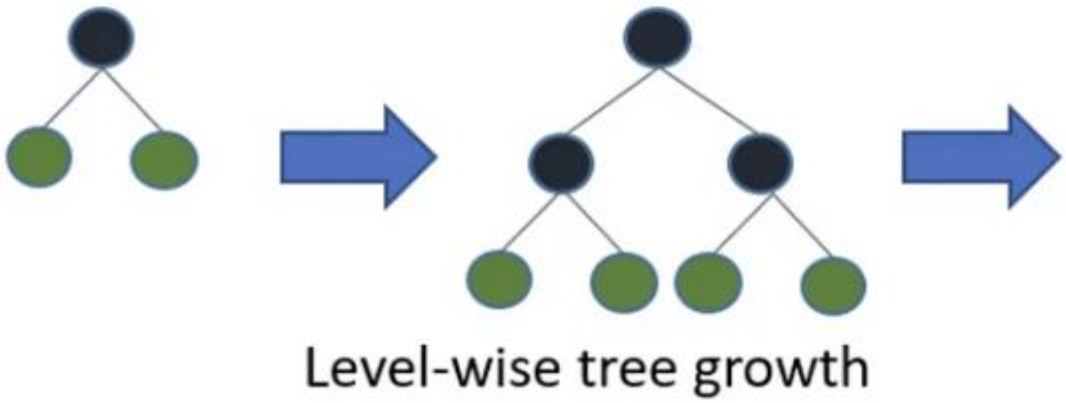
Figure 9 (Mandot, 2017)



Explains how LightGBM works

The figure above shows that LGBM grows tree vertically while other algorithms grow trees horizontally stating that LGBM grows tree leaf-wise while another algorithm grows level-wise as shown below. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm. This strategy can lead to a more accurate model with fewer levels in the trees, reducing the complexity. (Mandot, 2017)

Figure 10 (Mandot, 2017)



Advantages

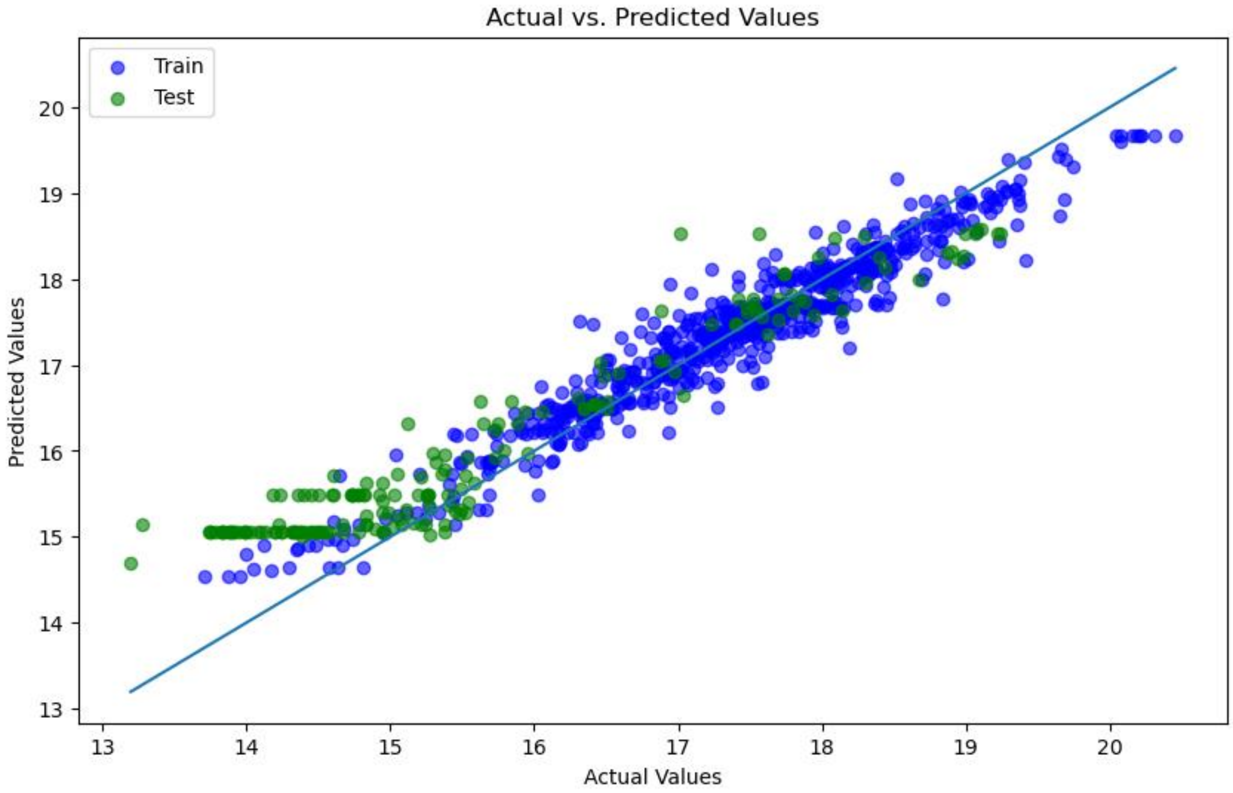
- Faster training speed with higher accuracy
- Lower memory usage
- Better accuracy than any other boosting algorithm especially handles the overfitting very well when working with a small dataset.
- Compatibility with large datasets
- Parallel learning support

Disadvantages

- While LGBM provides numerous parameters for fine-tuning, the complexity of tuning these hyperparameters can be a disadvantage. Finding the right combination of hyperparameters can be a time-consuming process.
- **Overfitting Risk:** As with other gradient boosting techniques, LGBM models are susceptible to overfitting if not properly regularized.
- **Limited Parallelism for Small Datasets:** LGBM's strengths lie in large datasets, but for small datasets, the overhead of distributed training can make it slower than some other algorithms.
- **Large Memory Usage:** LGBM constructs histograms for feature discretization during training, which can lead to high memory usage for large datasets or with a large number of features. This can be a limitation when working with limited memory resources.
- **Limited Interpretability:** Gradient boosting models, including LGBM, are seen often as "black box" models. They can be difficult to interpret, and understanding the feature importance can be challenging.
- **Complexity of GPU Usage:** While LGBM supports GPU acceleration, configuring and using GPUs for training can be more complex and might require additional setup.
- **Limited Parallelism for Small Datasets:** LGBM's strengths lie in large datasets, but for small datasets, the overhead of distributed training can make it slower than some other algorithms.

Training and Testing LGBM

Figure 11



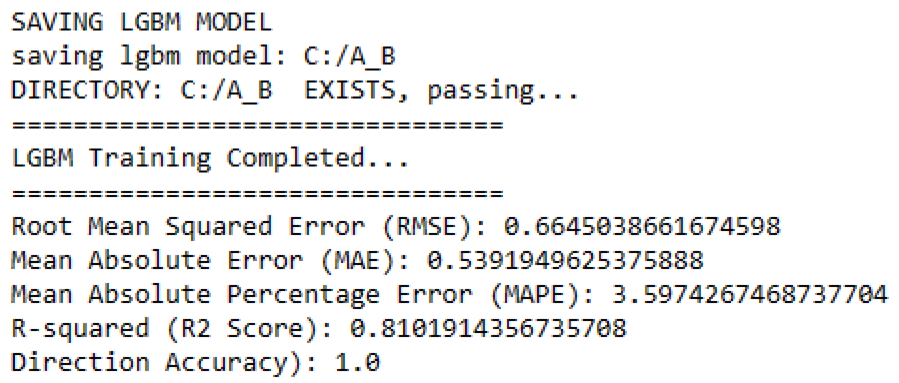
This is a model of the train and test data points. Judging by what is shown and the tightness of the data around the symmetry line. Both are fitting as they should as it was shown in LSTM modeling.

This figure shows:

- **Possible overfitting since predicated values are higher than actual values.**
- **Linear relationship with data since predicated values and test are clustered around line of symmetry. Shows that model is good for finding general trends of data.**
- **Clustering around predicated value of 15 could indicate an outlier effect.**
- **That cluster could also indicate that the data distribution drove results towards that value.**

One fix would be to fine tune the model to drive that cluster away. I do not see that as being a safe way to correct that issue. Since the data frame is not large enough to be able correct this. A larger frame would give the model a better symmetry towards the top and bottom of the line of symmetry.

Metrics LGBM



```
SAVING LGBM MODEL
saving lgbm model: C:/A_B
DIRECTORY: C:/A_B  EXISTS, passing...
================================
LGBM Training Completed...
================================
Root Mean Squared Error (RMSE): 0.6645038661674598
Mean Absolute Error (MAE): 0.5391949625375888
Mean Absolute Percentage Error (MAPE): 3.5974267468737704
R-squared (R2 Score): 0.8101914356735708
Direction Accuracy): 1.0
```

RMSE: Since the value is extremely low this is a good indicator, based on the size of the data frame. The model can predict house prices accurately. A higher number tells predictions are more widely spread away from observations.

MAE: In the context of LightGBM, MAE can be one of the loss functions to be minimized during training. However, because MAE is less sensitive to outliers and does not penalize large errors as heavily as RMSE, it might be chosen over RMSE when the model needs to be robust against outliers. This is shown in the training graph; however, it is felt that the small number of data points has driven this response. This model is good for large data frames, due to the limits of this project. Needed size of greater than five years of previous data was not feasible at this time.

MAPE: The error rate being as low as it is indicating predictions are close to actual values with the rate being as low as it is. Shows consistency with the level of accuracy. One limitation is that with the prediction of errors either by overestimation or underestimation if values have small numbers.

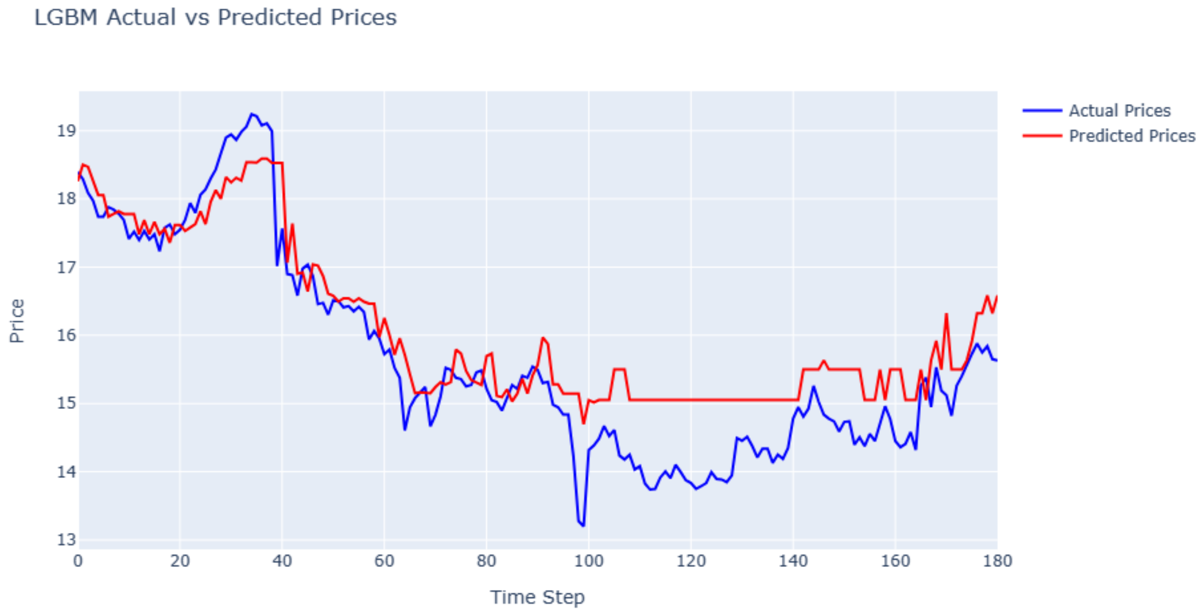
R-Squared: Close to one tells this model captures a good portion of variance in the dependent variable. However, a high number does not always tell that the model is good.

Direction Accuracy: Moving towards one, tells that the time series being analyzed is going towards up.

Summary: To make sure that predictions vs actual is good and valid all metrics for the model need to be considered. When deciding whether your responses are valid vs the data frame being used. I have shown that this model has good characteristics that would make it particularly useful in predicting prices.

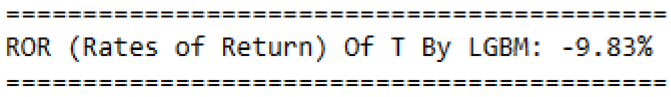
Predicated Price

Figure 12



The <u>actual price</u> on time step 25 was 18.13714.
The <u>predicted price</u> on time step 25 was 17.62915.

Rates of Return

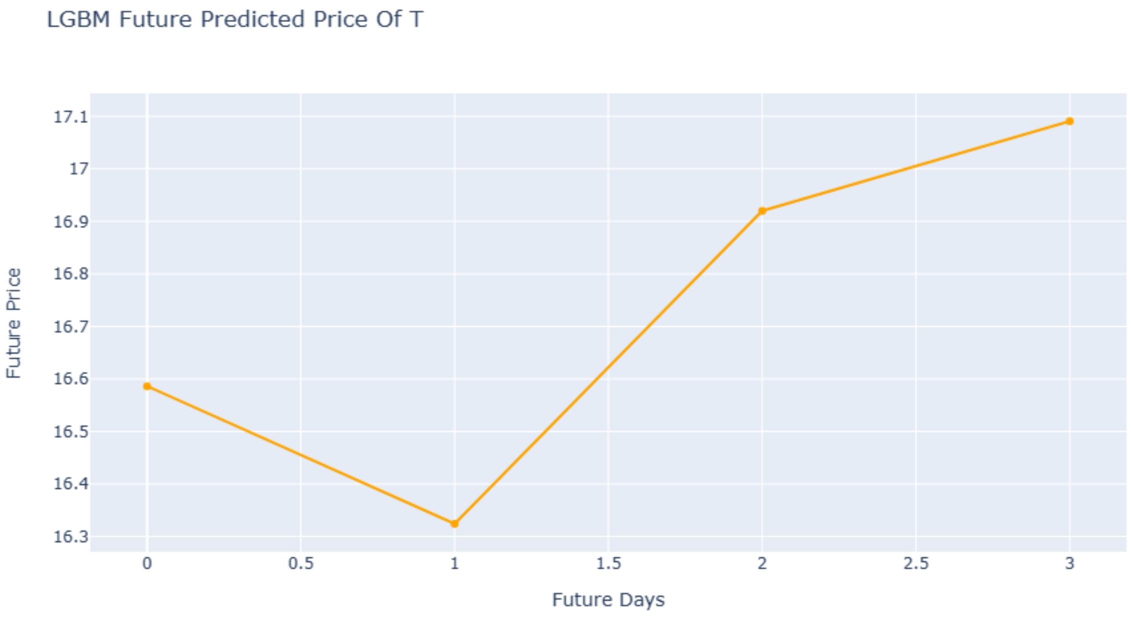


```
===========================================
ROR (Rates of Return) Of T By LGBM: -9.83%
===========================================
```

Based on 01-11-2020 to 11-10-2023 for AT&T stock

LGBM Future Predicted Price

Figure 13



.5 future days is equivalent to 4 hours of the market being open.

At the close of market on 11-13-2023 the predicted price for AT&T stock is 17.090

At the close of market on 11-13-2023 the actual price for AT&T stock is 15.58

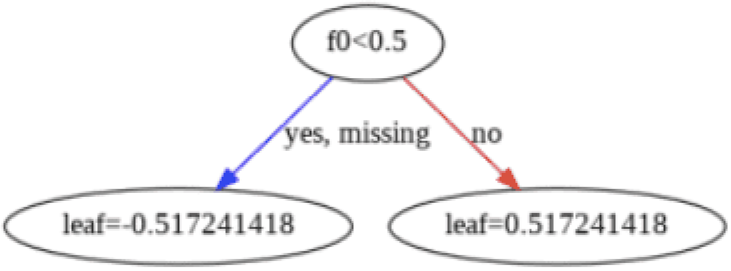
LSTM-XGBOOST

Combination of LSTM and XGBoost an ensemble learning a of machine learning that enlists both models to make predictions together. Boosting algorithms are distinguished from other ensemble learning techniques by building a sequence of initially weak models into increasingly more powerful models. Gradient boosting algorithms choose how to build a more powerful model using the gradient of a loss function that captures the performance of a model.

What is XGBoost, a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. It is powerful for structured data prediction and is often used for classification and regression tasks. XGBoost is known for its performance and speed. (Gupta, 2021)

XGBoost operates on decision trees, models that construct a graph that examines the input under various "if" statements (vertices in the graph). Whether the "if" condition is satisfied influences the next "if" condition and eventual prediction. XGBoost the Algorithm progressively adds more "if" conditions to the decision tree to build a stronger model. (Gupta, 2021)

Figure 14 (Gupta, 2021)



XGBoost considers the leaves of the current decision tree and questions whether turning that leaf into a new "if" statement with separate predictions would benefit the model. The benefit to the model depends on the "if" statement chosen and which leaf it is placed on—this can be determined using the gradient of the loss. The loss includes a scoring function that measures algorithm performance. (Gupta, 2021)

The hybrid consists of LSTM to help understand the characteristics of input and will be able to learn features automatically, with XGBoost to help with classification.

A disadvantage, to this model, is that complexity with combining both models as well as hyperparameter tuning which models need for optimal performance.

Methods

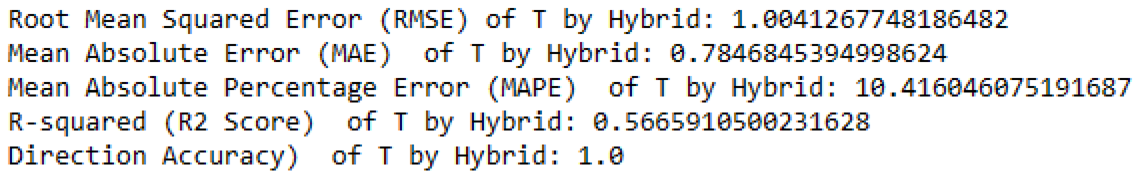
XGBoost used a process called windowing breaking the data, where data is broken down into a smaller sequence that can be processed by an algorithm. Two dimensional tables that have data take a slice and put into three dimensions.

LSTM training and testing were competed.

XGBoost training and testing were competed.

Run LSTM first then combine XGBoost as a method for fine tuning.

Metrics



```
 Root Mean Squared Error (RMSE) of T by Hybrid: 1.0041267748186482
 Mean Absolute Error (MAE)  of T by Hybrid: 0.7846845394998624
 Mean Absolute Percentage Error (MAPE)  of T by Hybrid: 10.416046075191687
 R-squared (R2 Score)  of T by Hybrid: 0.5665910500231628
 Direction Accuracy)  of T by Hybrid: 1.0
```

RMSE: Since this is above one there is a possibility that the prediction. Could indicate that the data frame is not fitting prediction well. A hyperparameter tune-up could help to improve the outcome of this metric. A bigger sample size can help to bring this below one.

MAE: Lower than one, shows good measure between predicted and actual values.

MAPE: High, tells that performance might be skewed since the consistency is a part of the level of accuracy. One limitation is that with the prediction of errors either by overestimation or underestimation if values have large numbers. Metrics would have looked at overall and possibility of a tuning could help lower this metric.

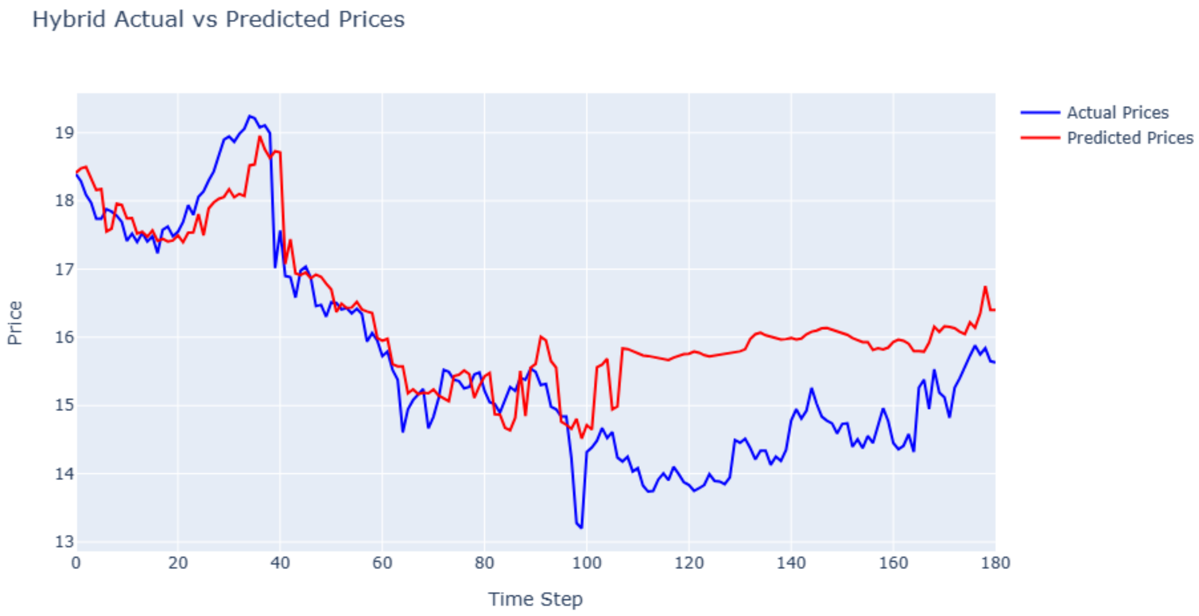
R-Squared: Since the lower indicates that the dependent variable is as goodness of fit as the other two models had. Hyperparameter tuning would help with this metric.

Direction Accuracy: No change when comparing with the other two models. Everything is moving in the correct direction.

Summary: Metrics for this model were not as robust as the other two models. I found that even with the hybrid metric values were down across the board. To me this was interesting in that one of the models was LSTM which fit the data quite well. I thought that would carry over to this model. I expected XGBoost to help fine tune the results without having to use overbearing hyperparameter tuning. I believe that a larger data frame would help with getting better metrics without having to go the hyperparameter route.

Predicted Price

Figure 15

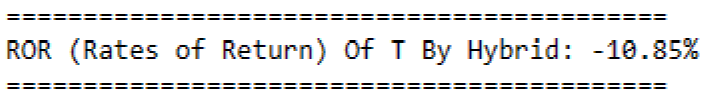


The actual price on time step 25 was 18.13714.
The predicted price on time step 25 was 17.49549.

I noticed that for the same time step that the actual and predicted prices were similar to LGBM. Even though the methods to find predictions were dissimilar. I believe that since driven by XGBoost it allowed for the forecasting to be tuned at a small level. Prediction does show that there is a difference between predicted and actual of approximately 4%.

Rate of Return

Figure 16

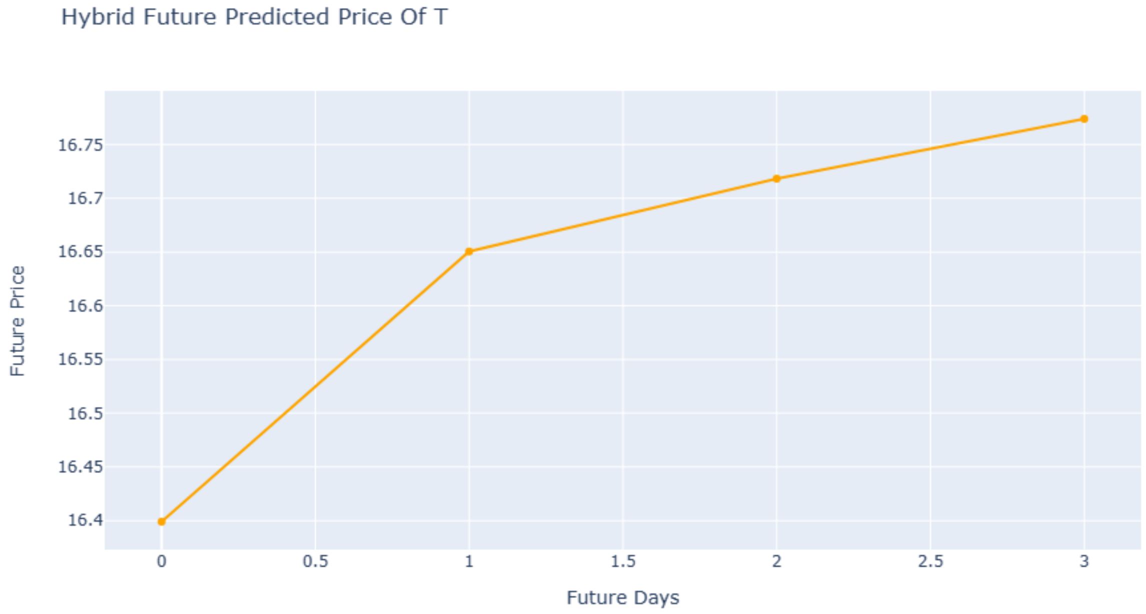


```
============================================
ROR (Rates of Return) Of T By Hybrid: -10.85%
============================================
```

Based on 01-11-2020 to 11-10-2023 for AT&T stock

LSTM – XGBoost Hybrid

Figure 17



.5 future days is equivalent to 4 hours of the market being open.

At the close of market on 11-13-2023 the predicted price for AT&T stock is 16.77389

At the close of market on 11-13-2023 the actual price for AT&T stock is 15.58

What I found

The thing that impressed me the most was the depth of information available for this endeavor. If I wanted to, I could have spent several months of research and development to create new models for use. Of course, it would have taken more than many months to develop a new model. The level of granularity that one could use to find new areas to research and become an expert is considerable.

One area that is worthy of further investment is developing a system where a lay person will enter an amount they have to invest, their level of risk, and have the algorithm return four to five securities that are investment grade. I believe that one way to solve this would be to have a rating system based on certain metrics that are graded. The algorithm would pull those securities with the desired risk level and run them though the any of the above AI to see if the picker has a good point for the client to purchase.

Developing, the R and R log for each individual security along with each sector was difficult to say the least. A challenge for me was being able to figure out where to get and calculate what R and R log were for the sector in question and have it return for all securities entered.

Having the models work as well as they did was quite a surprise. I knew that they were going to work. But work at this level was very impressive to me. I know that if I let my friends and family use this algorithm, feel greatly confident that they will make money.

Stock prediction is a busy field, many algorithms are available to predict what the possible price could be in the future. I have presented three models that can predict future prices. In researching this field, I was able to determine that there are enough primary and secondary fields in this area to research for the next two years. My report only scratched the surface of what is possible.

The future is big in this area, since there are not many opportunities for regular people to be introduced to AI for personal finance at an affordable level. I hope that I have shown a light to an extremely exciting and deep field.

This was an enjoyable project my undergraduate was in Finance with emphasis on investments. A return to an area that I still like and enjoy!

Review

LSTM

- **Good**
  - **A well-known predictor of time series problems**
  - **Works well with small sized data frame and larger**
  - **Good for finding dependencies.**
  - **Finding Complex Relationships**
  - **Good for working with sequential data.**
  - **Adaptability to diverse data**
- **Bad**
  - **Can use resources for computation that are invasive.**
  - **Hyperparameter Tuning can be complex.**
  - **Prone to overfitting**
  - **Large amounts of high-quality data is needed**

LGBM

- **Good**
  - **Fast Training and Efficiency.**
  - **Low Memory Usage**
  - **Improved Accuracy**
  - **Flexibility and Customization**
  - **Robustness to Outliers**
- **Bad**
  - **Higher Learning Rate Requirements**
  - **Sensitivity to Hyperparameters**
  - **Difficult to understand how results were obtained.**
  - **Prone to overfitting**
  - **Limited Feature Selection**

LSTM-XGBoost

- **Good**
  - **LSTM does many of the calculations by capturing temporal dependencies.**
  - **XGBoost fine tunes by removing non-linear features (i.e., market sentiment, and economic indicators)**
  - **Lower memory usage**
  - **Improved Accuracy**
  - **Can do parallel and distributed learning.**
  - **Reduced Overfitting**
- **Bad**
  - **Complexity increases due to combining two models.**
  - **Tuning challenges**
  - **Being new creates many challenges that are followed industry wide**

References

Bollinger, J. (2021). *What are Bollinger Bands*. Retrieved from Bollinger Bands: https://www.bollingerbands.com/bollinger-bands

Chen, J. (2023, March 31). *What is EMA? How to Use Exponential Moving Average With Formula*. Retrieved from Investopedia: https://www.investopedia.com/terms/e/ema.asp

Dolan, B. (2023, July 21). *MACD Indicator Explained, with Formula, Examples, and Limitations*. Retrieved from investopedia: https://www.investopedia.com/terms/m/macd.asp

Financial Modeling Prep. (n.d.). Retrieved from financialmodelingprep.com: https://site.financialmodelingprep.com/

Ganti, A. (2020, December 28). *Adjusted Closting Price:How it works, types, Pros & Cons*. Retrieved from Investopedia: https://www.investopedia.com/terms/a/adjusted_closing_price.asp

Gupta, S. (2021, March 8). *XGBoost Simply Explained (With an Example in Python)*. Retrieved from Springboard: https://www.springboard.com/blog/data-science/xgboost-explainer/

J., R. T. (2020, September 2). *LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras*. Retrieved from Medium: https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2

Kenton, W. (2023, September 29). *Rate of Return (RoR) Meaning, Formula, and Examples*. Retrieved from Investopedia: https://www.investopedia.com/terms/r/rateofreturn.asp

Mandot, P. (2017, August 17). *What is LightGBM, How to implement it? How to fine tune the parameters?* Retrieved from Medium: https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc

Tourloukis, G. (2021, August 18). *Sequential Time Series Forecasting | LSTM | Stock Market Dataset*. Retrieved from Medium: https://medium.com/@geotourloukis/sequential-time-series-forecasting-lstm-stock-price-prediction-8794c9ecac89