Machine Learning Engineer Nanodegree

Capstone Project

Sui Watchorn
September 13, 2019

I. Definition

Project Overview

Harry Markowitz, the 1990 Nobel Laurate for Economics, dedicated his work to the Efficient Market Hypothesis[1] (EMH) maintaining all securities are efficiently priced in the long run and drastic changes are attributed to random fluctuations given the investors behave logically.[2] Under this theory, the stock market is not predictable.

Before embarking in a career in IT, my undergrad studies were focused in Economic & Finance Management and finance or specifically, stocks, have remained a personal interest of mine. Machine Learning (ML) wasn't available to the wider public at the time of my undergraduate studies and if it was, it was more in research and development to define the theories and algorithms that comprise the foundation of ML.

Technological advancement of hardware and software have since enabled the rapid development of ML and it has been widely adopted and used to do predict random stock market fluctuations. These predicts are used to try to beat the market by assigning levels of risk to reward or returns in the stock market. ML strategies from Reinforced Learning to Neural Networks have been used to build models precisely to predict the risk and returns in stock market/prices with the hopes of maximizing rewards/returns.

National Association of Securities Dealers Automated Quotations (NASDAQ) of the United States was founded in 1971 and it makes up part of the stock market. It is an electronic stock exchange with more than 3,300 company listings trading on the

[1] http://www.e-m-h.org/
[2] https://www.guidedchoice.com/video/dr-harry-markowitz-father-of-modern-portfolio-theory/

exchange.[3] With approximately 253 trading days per year, the stock data would be substantial for daily and/or hourly trades. This project will narrow down the scope and explore the use of Machine Learning in stock prediction for daily trading specific to the AT&T stock (T). Facebook's prophet or fbprophet, which incorporates and automates many of the Machine Learning principles such as Baysian sampling for inference into its modeling, will be the method used in this project to predict stock pricing and compare it with actual market performance to gauge model performance or whether T beat the market performance.

Problem Statement

AT&T (T) has been publicly traded since 1983 with total of 9,020 days of trading up to the present, not including holidays and weekends. Additional data is available for hourly trades and transactions, but the scope of this project will use historical data from `1984-07-19 00:00:00 to 2018-03-27 00:00:00.`

Originally, when I set out to research and complete this project, the .csv datasets I explored were sourced from Yahoo! Finance, Quandl, and Google Finance. My intention was to use them to construct a model using a Q-learner. After much time spent obtaining and scrubbing the data, I decided to leverage existing libraries and tools that have Baysian principals incorporated and already available for time-series analysis. The project will focus primarily on AT&T (T) stock and its daily performance with data sourced from Quandl's WIKI prices database to built a model using Facebook Open Source fbprophet.

What is Facebook prophet?

" *Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.*" [4]

The benchmark will be a 'Buy & Hold' strategy to gauge the modeled prediction performance. The database will be used for testing and training of daily stock price datasets for a 1-yr period to construct a prophet model. Then the model will be used to test predictions using varying changepoints. The prophet model will be used to identify

[3] https://new.nasdaq.com/
[4] https://facebook.github.io/prophet/

buy or sell opportunities and compare it with actual market performance to gauge whether the modele beat the market.

Metrics

Ultimately, for any stock investor, the measurement of success is positive return on investment (ROI). The stock returns metric will be used to measure the performance of the prophet model. Direct comparison between the model and actual time-series Open-High-Low-Close-Volume (OHLCV) information for T will be used to evaluate the model. A benchmark with standardized results will show which strategy is more successful—actual market performance vs the predictive model. The analysis will start with 100 shares from the beginning until the selected end date to determine the return versus the Buy & Hold strategy for the same number of 100 shares of T stock.

II. Analysis

Data Exploration

Time series analysis aims to identify the nature of significant occurrences representing the sequence of observations and forecasting future values of those observations. It will use the variables to detect patterns that can be used to fit a model to interpret it in a quantifiable manner that can be used to predict future occurrences.

Fbprophet's time series analysis uses changepoints (CP) to represents those meaningful observations over time to extract characteristics to be used in the prophet model construction. It contains a time series forecasting library that requires no data pre-processing and is simple to implement. The input for Prophet is a dataframe comprised of two columns—date (ds) and target (y). It also tries to capture the seasonality of past data, such as yearly, monthly, and weekly trends.

While the stock market is closed to trading during weekends and holidays, the effects of weekends and holidays on business operations are real, especially companies like AT&T (T) that have retail divisions with sales that are cyclical and influenced by weekend and holiday consumer spending. The stock market rewards the company's good retail performance via increased trading activities that drive up the stock price. The opposite can be said for poor retail performance. This market data will be used to construct a model that can interpret it in a quantifiable way to identify or predict pricing.

AT&T (T) stock information will be obtained from Quandl's WIKI prices database. The database contains historical end of day Open-High-Low-Close-Volume (OHLCV) data

with adjusted OHLCV, Split Ratio and Daily Changes. AT&T (T) has had "three stock splits over the past 30 years: a 3-for-1 split in 1987, a 2-for-1 split in 1993 and another 2-for-1 split in 1998." [5] This information is captured in the database as well. Below is the table of T's historic stock output:

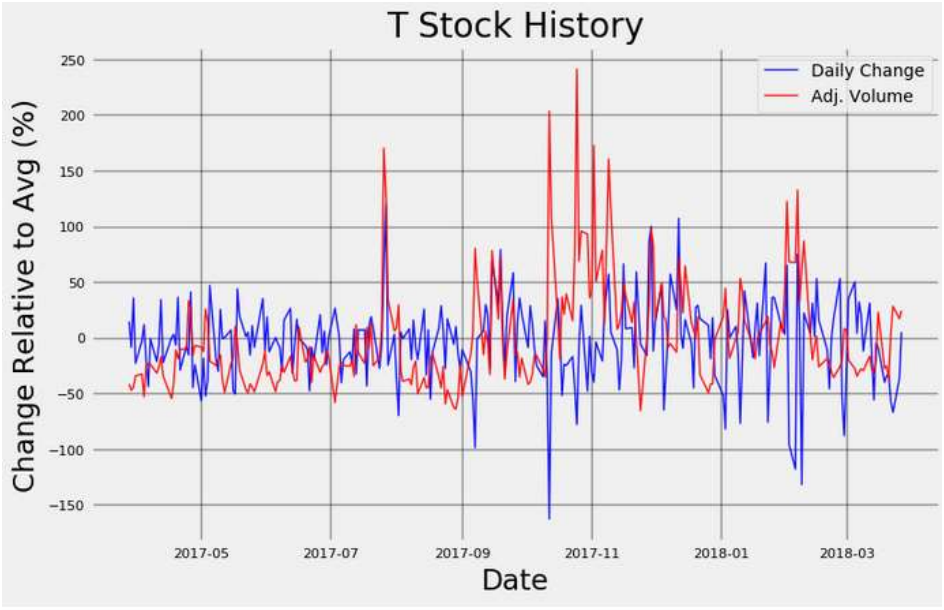
	Date	Open	High	Low	Close	Volume	Ex-Dividend	Split Ratio	Adj. Open	Adj. High	Adj. Low	Adj. Close	Adj. Volume	ds	y	Daily Change
0	1984-07-19	59.75	59.87	59.38	59.38	77900.0	0.0	1.0	1.275248	1.277809	1.267351	1.267351	934800.0	1984-07-19	1.267351	-0.007897
1	1984-07-20	59.62	60.00	59.25	59.50	129800.0	0.0	1.0	1.272474	1.280584	1.264577	1.269913	1557600.0	1984-07-20	1.269913	-0.002561
2	1984-07-23	59.25	59.75	58.75	59.50	276100.0	0.0	1.0	1.264577	1.275248	1.253905	1.269913	3313200.0	1984-07-23	1.269913	0.005336
3	1984-07-24	59.75	60.38	59.62	60.00	129100.0	0.0	1.0	1.275248	1.288694	1.272474	1.280584	1549200.0	1984-07-24	1.280584	0.005336
4	1984-07-25	60.00	61.37	59.87	60.75	152400.0	0.0	1.0	1.280584	1.309824	1.277809	1.296591	1828800.0	1984-07-25	1.296591	0.016007

The stock history Adjusted Close daily price can be obtained and plotted graphically using `plot_stock`.

```
Maximum Adj. Close = 41.03 on 2017-03-17 00:00:00.
Minimum Adj. Close = 1.27 on 1984-07-19 00:00:00.
Current Adj. Close = 34.90 on 2018-03-27 00:00:00.
```

The data can be evaluated further by plotting the Adjusted Volume and Daily Change to gauge the stock's daily activity.

[5] https://www.investopedia.com/articles/markets/020216/if-you-had-invested-right-after-atts-ipo-t.asp



Exploratory Visualization

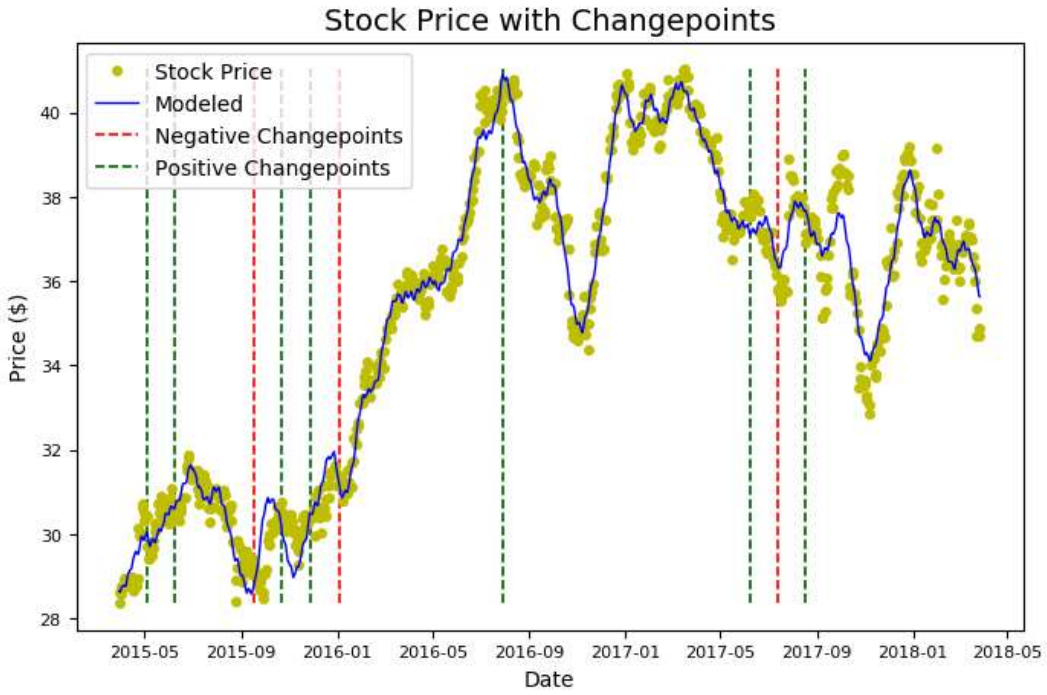
Now that the stock's historical data has been extracted, new ways can be used to incorporate it into a prophet model. Fbprophet accounts for the time series rapid increases or when increases change to decreases and vice versa for both scenarios. These rapid changes are represented by changepoints (CP) and the changepoint prior scale (CPS) are weights applied to the change in trend to fit the prophet model. The higher the CPS, the more weight is placed on changepoints (possible overfitting) and conversely, the lower the CPS, the less weight is placed on the changepoints (possible underfitting)—finding the right balance to determine the best CPS to optimize the model for price prediction is key.

The stock's 10 changepoints are identified using the command:

```
att.changepoint_date_analysis()
```

It will produce the following output and the corresponding changepoints:

```
Changepoints sorted by slope rate of change (2nd derivative):

          Date  Adj. Close     delta
193 2016-01-04   31.109314  3.867750
120 2015-09-18   29.066652  2.853506
144 2015-10-22   30.756108 -2.582493
337 2016-07-29   40.700585 -2.340586
578 2017-07-14   35.844857  2.144585
```



The changepoints from the output graph are aligned with the stock price highs and lows. The graph depicts additional opportunities to improve the model fit as it missed a couple of the drastic stock price increases and decreases.

Algorithms and Techniques

Prophet or additive model's formula consists of the following components at the time period or t:

- y_t – data
- S_t – seasonal component
- $\mathcal{T}_t$ – trend-cycle component
- $\mathcal{R}_t$ – remainder component

The formula can be written as:

$$y_t = S_t + \mathcal{T}_t + \mathcal{R}_t$$[6]

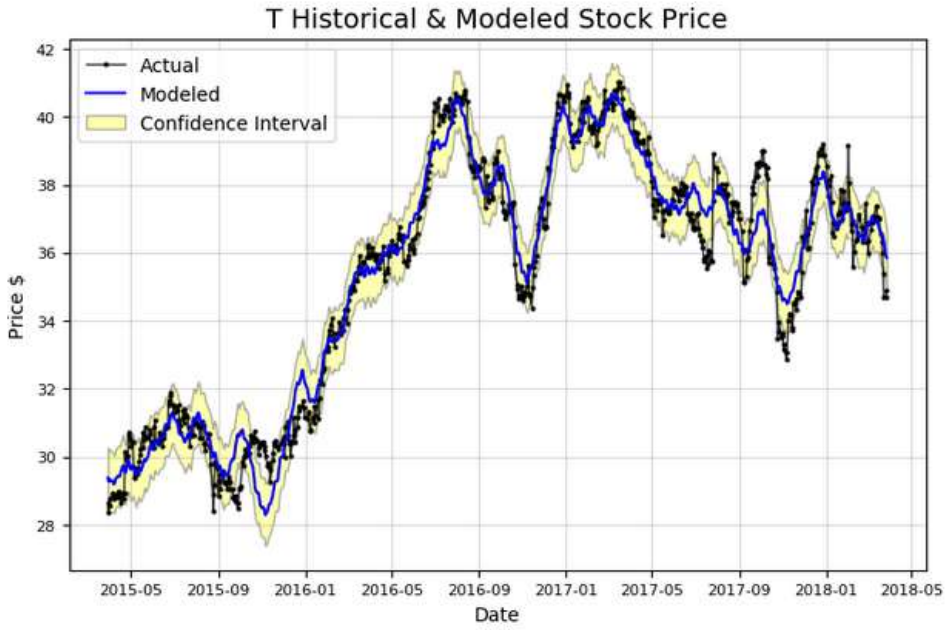
First, the data will be trained and tested with the prophet's built-in seasonal, trend-cycle, and remainder components against the AT&T (T) historical stock performance. The last 3 years of the historical data will be trained and tested to gauge performance. A good

[6] https://otexts.com/fpp2/components.html

thing about fbprophet is that it compiles the data behind the scenes and completes the modeling autonomously. The model can be created using the command:

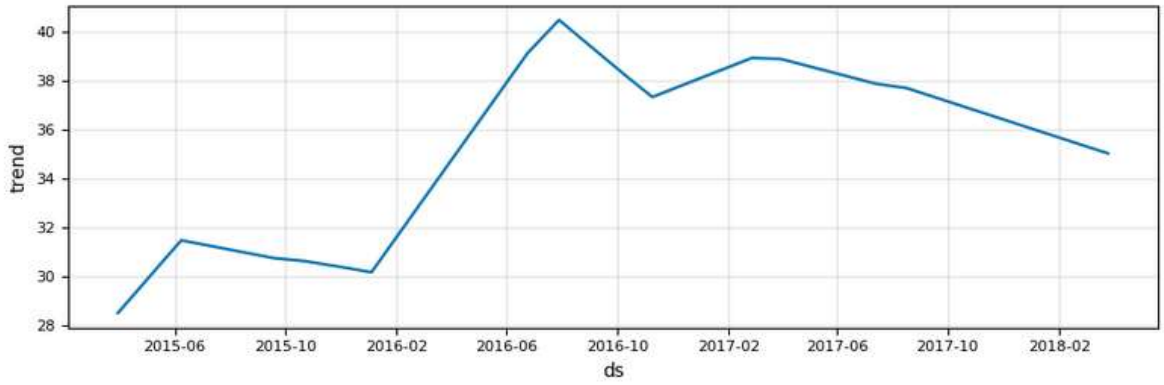
```
model, model_data = att.create_prophet_model()
```

It will also plot the data in a graphical representation between actual and modeled prices within the confidence interval.



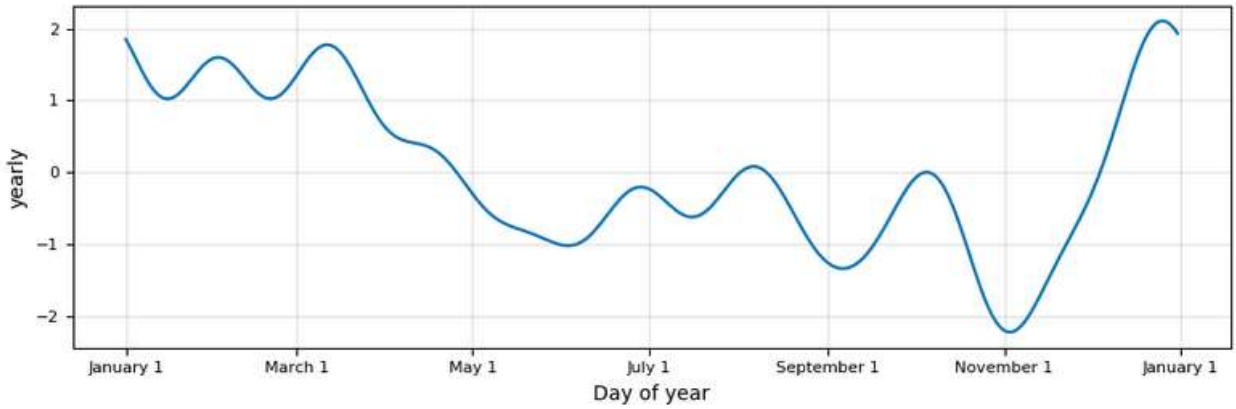
It is apparent from the graph that the blue modeled predictions are comparable to the actual stock performance but there are extreme contrasts in some of the areas the model either overfitted or underfitted—3Q2015, 4Q15, and 3Q2017. The Confidence Interval width of the model is set at 80%, meaning the actual price will fall within that interval 80% of the time. Even with the fluctuations, the modeled prices stayed within the confidence interval.

The prophet model automatically adjusts and smooths out some of the noise in the data and that is why the modeled line doesn't fully line up with the actual market observations. The model takes uncertainty calculations into account for the unexplained random market fluctuations. The calculated data is assigned to variables that can be used to plot the time series components to determine performance trends in monthly and yearly intervals.

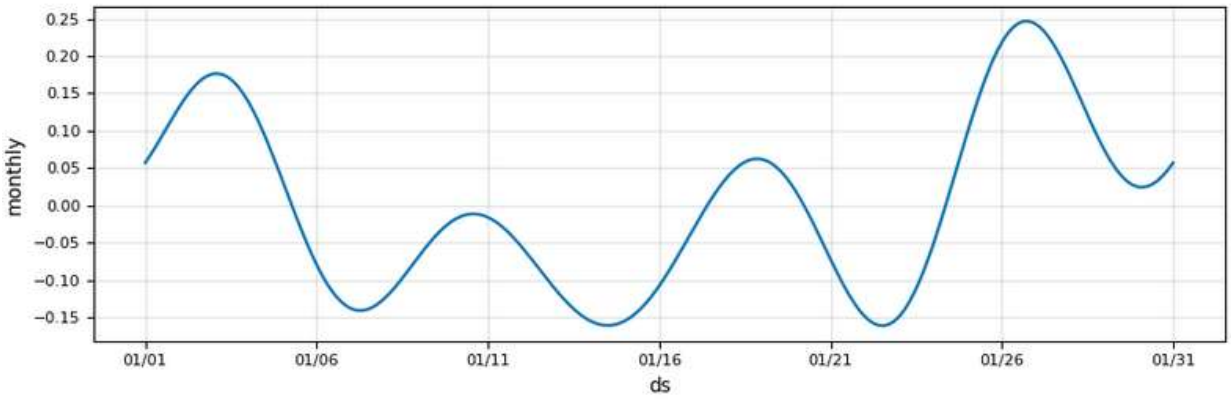


The 3-yr. data shows an increasing trend from the beginning of the period and a big spike around 8/2016 which is then followed by a slowing or decreasing trend. AT&T (T) officially announced the Time Warner deal 10/22/2016[7] and the upward trend may represent speculation of the merger prior to the official announcement. It is likely the post announcement drop can be attributed to the confidence of FFC's approval for the merger. As the merger progresses, the lack of confidence, or the lack of market optimism of the deal closing, is pulling the stock price down.

There is a clear yearly and monthly trend as well, with both tapering down from the start of the period and hit the lowest point during the middle and then a bigger ramp up at the end of the cycle. This trend has a strong correlation to the increase in winter holiday season spending and post-holiday drop, tapering off to bottom from spring into summer. The trends account for the market's response to the company's performance during those times. The same can be observed during the monthly trend where it ramps up towards the last week of the month, which may translate into consumers' confidence, and tend to increase around monthly pay periods.

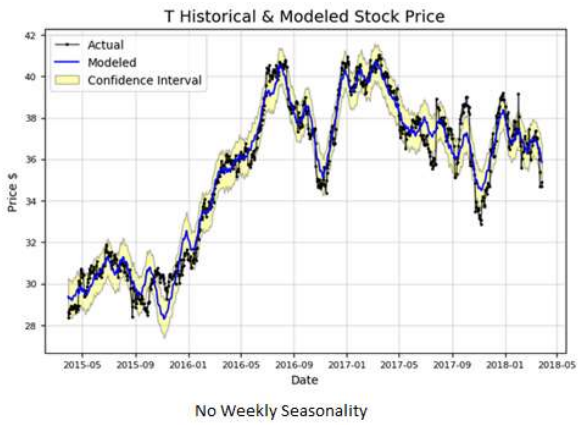


[7] https://about.att.com/story/att_to_acquire_time_warner.html

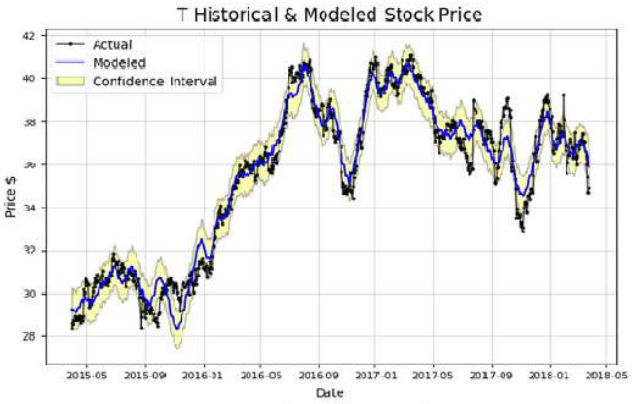


The model can be further trained to include weekly seasonal trends in its calculations.

```
att.weekly_seasonality=True
model, model_data = att.create_prophet_model()
```

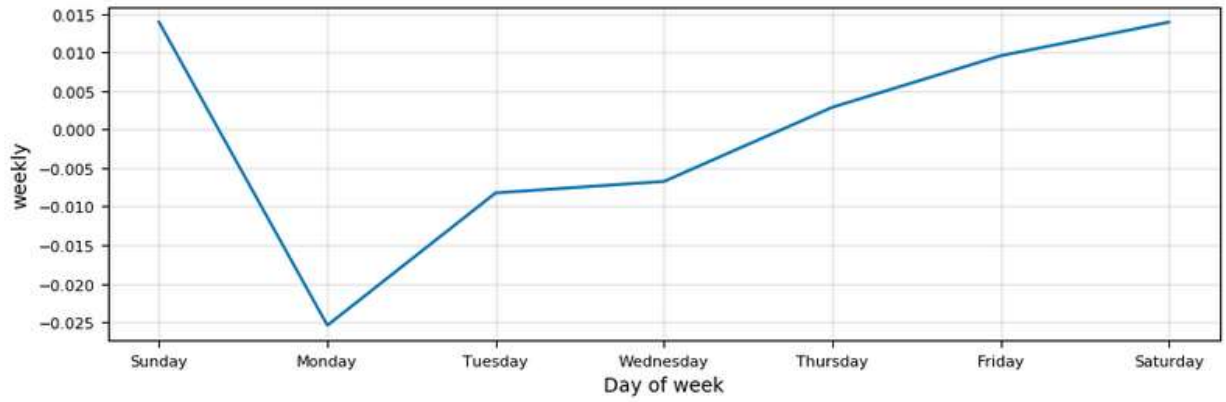


No Weekly Seasonality



With Weekly Seasonality

The modeled performance didn't vary much with the weekly seasonality included. Nonetheless, there is a clear and recognizable trend for weekly seasonality where it drops sharply after the weekend and ramps up through the work week, to hit a peak for the weekend.



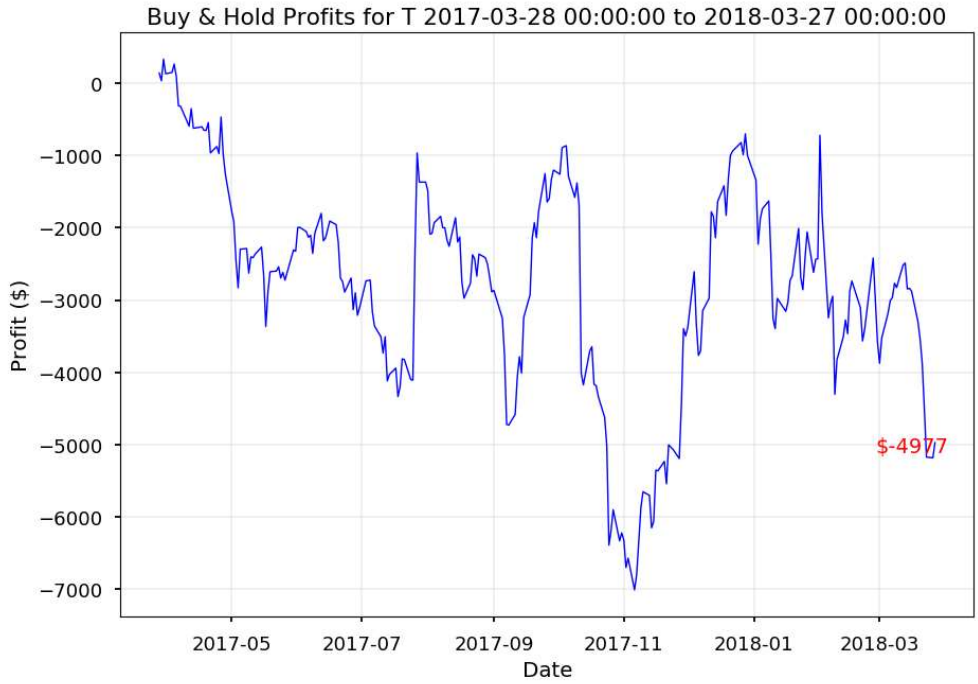
With all the prophet model components identified, testing can be started for accuracy and training, as needed.

Benchmark

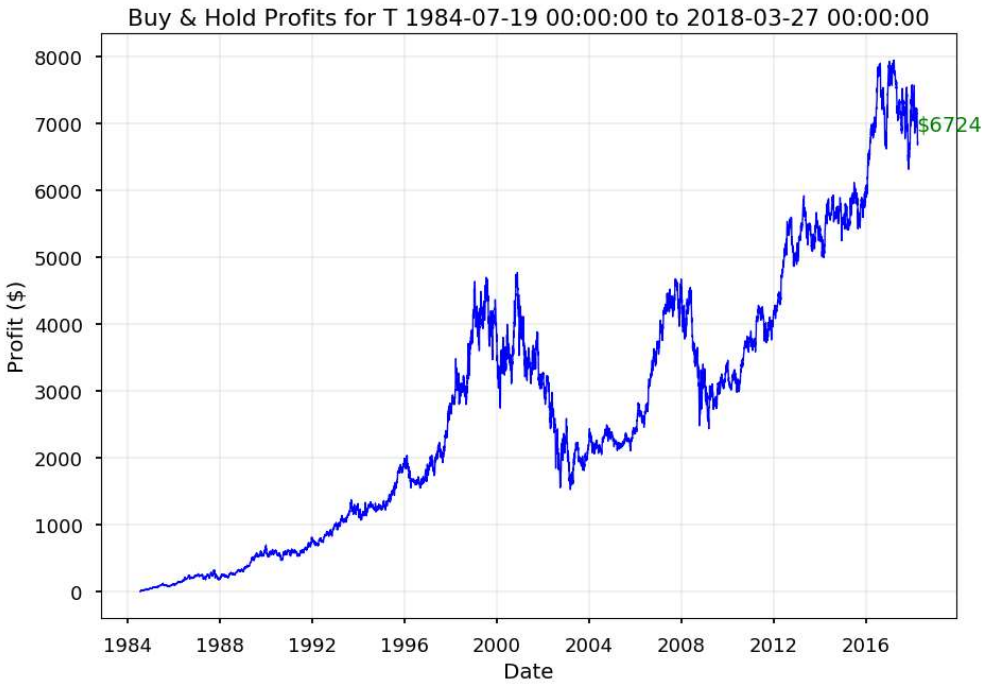
The Buy & Hold Profits strategy will be used as the benchmark to gauge the model performance. A date range can be specified to obtain the benchmark profits for 1000 shares of AT&T (T) stock using the command:

```
att.buy_and_hold(start_date='2017-03-28',
                 end_date='2018-03-27', nshares=1000)
```

This will provide an idea of the stock performance, measured in dollars ($), for the benchmark strategy. The profit for the specified time period from 3/28/2017 to 3/27/2018, is negative (-$4,977), a loss of -$4.98/share. Money was lost with the buy & hold strategy for the selected 1-yr timeframe.



The graphic below provides a look at the entire stock history profitability using the buy & hold strategy for 200 shares of AT&T (T) stock from 7/19/1984 to 3/27/2018. It's known the stock has split a few times since its inception, but that will be ignored for simplicity. The benchmark strategy yields a profit of $6,724 or $33.62/initial share.



III. Methodology

Data Preprocessing

The stock data from Qandl's WIKI database is clean and straightforward with typical Open-High-Low-Close-Volume (OHLCV) data. The `att.plot_stock()` option has additional built-in functions to help classify the data for fbprophet additive modeling. The OHLCV data is adjusted and additional columns added for each adjusted OHLCV value. It also created the required fbprophet columns for classifications—dataframe (ds) and y. ds default format is YYYY-MM-DD HH:MM:SS.

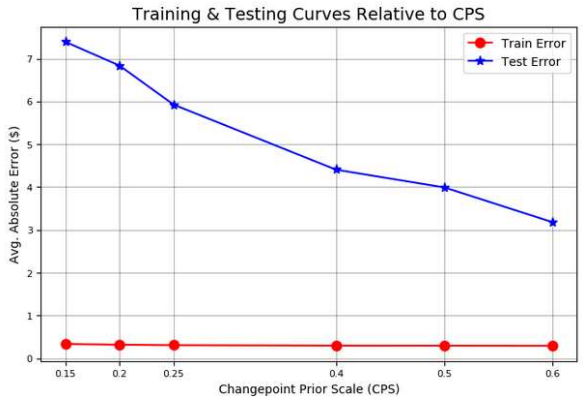
Implementation

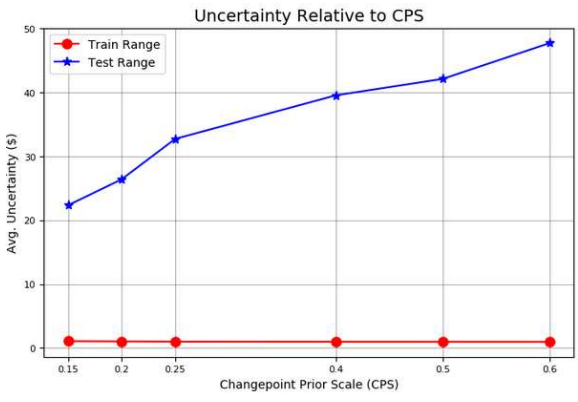
Implementation was challenging in the beginning since fbprophet is new to me, but the challenge made it that much more interesting than my original intent of using Q-learner. It enabled me to explore how fbprophet is currently being used in data analytics across many industries as well as within our company. It strengthened my resolve to continue with it and make it work. This can translate directly into something that is applicable to be used in my day-to-day job responsibilities.

The implementation for this project progressed in the following order:

- Downloaded and installed the needed environments—Quandl, Matplotlib, Numpy, Pandas, and fbprophet (Microsoft C++ is required for fbprophet installation).
- Downloaded stock data from Quandl's WIKI database.
- Extracted, explored, and processed the data to refine and plot it for review.
- Created the Buy & Hold strategy model for benchmark measurements.
- Crafted a prophet model to identify and display the changepoints trends for the selected stock.
- Identified important attributes to be included in the analysis.
- Modeled the prophet trends for yearly, monthly, and weekly seasonality. Reviewed the trends, prophet model, and changepoint prior analysis for changepoint selection to best predict the stock price to measure against the benchmark.
- Reviewed trends to identify which should be used in the stock purchase model.
- Trained and plotted the model using various changepoint priors to account for random fluctuations and trends.
- Tested the model for predictions.
- Compared the predicted model with actual stock historical pricing.
- Validated the prediction model using specific data ranges and changepoints.
- Completed changepoint (CP) prior validations on various changepoints to gauge training and testing error with uncertainty. The testing error decreased as CP increased but the uncertainty increased as CP increased.

```
Validation Range 2016-09-01 00:00:00 to 2017-08-31 00:00:00.

    cps  train_err  train_range  test_err  test_range
0  0.15   0.336794     1.061755  7.391563   22.343930
1  0.20   0.320691     1.022922  6.837557   26.408037
2  0.25   0.309097     0.991598  5.925198   32.732477
3  0.40   0.299476     0.967138  4.406398   39.557035
4  0.50   0.297389     0.960423  3.992560   42.136478
5  0.60   0.294119     0.949886  3.180260   47.750968
```





- Tested the model using specific shares of stock for stated date ranges for change points that have the lowest error, lowest uncertainty, and both error and uncertainty

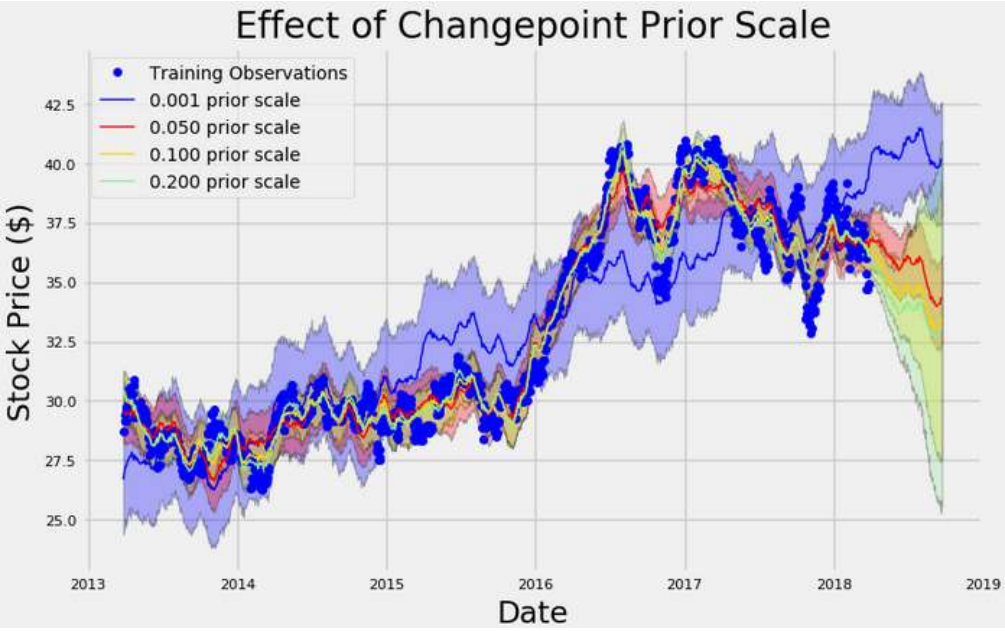
- Identified the best changepoint to be used for stock price prediction comparison with the buy & hold benchmark
- Compared the model changepoint outputs or returns to determine success or failure
- Used the model to predict and plotted future stock price in various ranges from 7 days up to 180 days with dates of increases and decreases
- Validated additional prediction observations by removing seasonality and compared outputs or return with previous predictions

Refinement

Since the changepoint (CP) priors play a crucial role in the fit and performance of the prophet model, the model was further refined by selecting additional changepoint scales (CPS).

```
att.changepoint_prior_analysis(changepoint_priors=[0.001, 0.05, 0.1, 0.2])
```

Visualization with graphical representation of the selected CPS helped provide a visual comparison of each CPS's effects and a means with which to home in on the best CPS to use for our predictions.

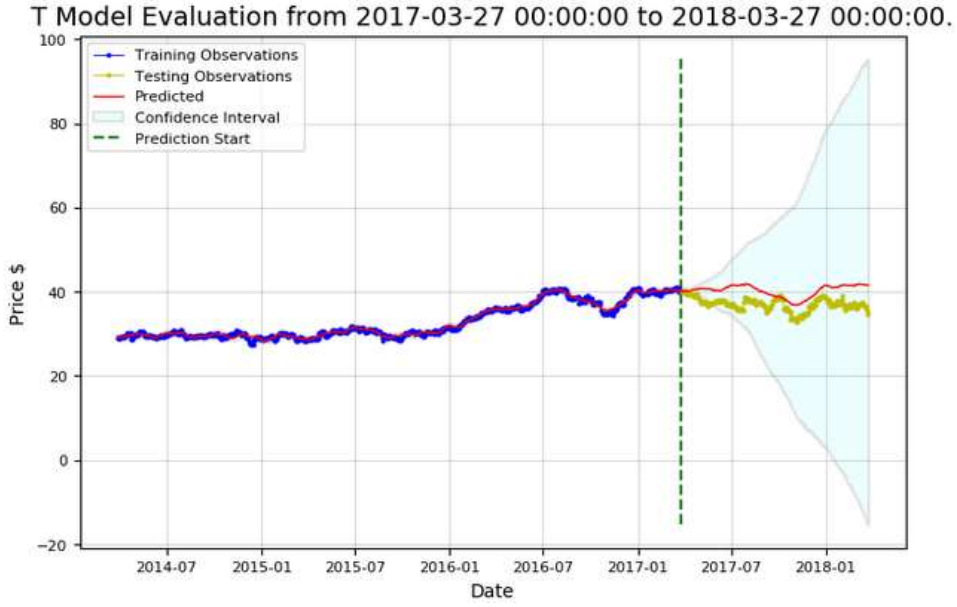


The changepoint validation helped in the CPS selection, insofar that it showed a correlation, where the prior increased the error decreased. It also included the level of

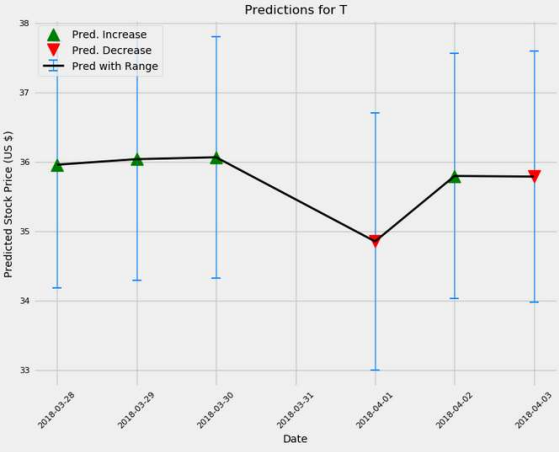
uncertainty for each CPS. The lower the uncertainty, the lower the CPS. Both the error selection and uncertainty represented underfitting and overfitting of the model.

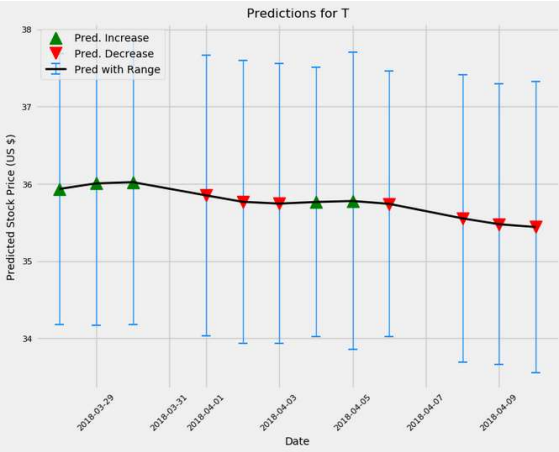
Observations were further validated with additional CPS data to determine the optimal CP. It turned out to be 0.6 with the smallest error. That will be the CPS used to evaluate the model's predictions `att.evaluate_prediction()`.

```
Prediction Range: 2017-03-27 00:00:00 to 2018-03-27 00:00:00.

Predicted price on 2018-03-24 00:00:00 = $41.60.
Actual price on    2018-03-23 00:00:00 = $34.70.

Av. Absolute Error on Training Data = $0.38.
Av. Absolute Error on Testing  Data = $3.12.

The price increased when predicted an increase 49.58% of the time.
The price decreased when predicted a decrease  49.23% of the time.

The actual value was within the 80% confidence interval 88.40% of the time.
```



The initial modeled performance with and without weekly seasonality, didn't produce significant difference or improvement in the modeled stock price predictions. The weekly seasonality also added a significant drop in stock price prediction for the weekends. By adjusting the model, removing the weekly seasonality, the model smoothed out the weekend transitions. The difference can be observed in the graphs below:



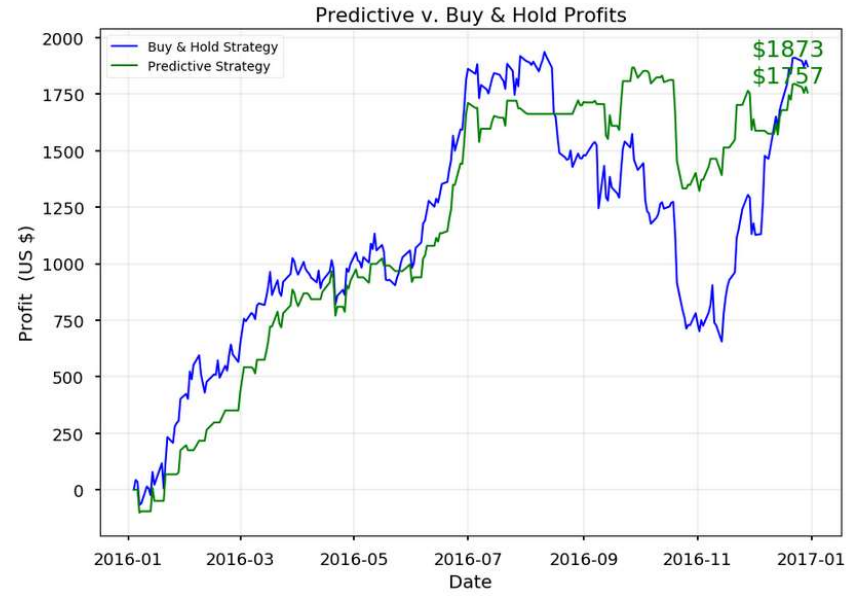


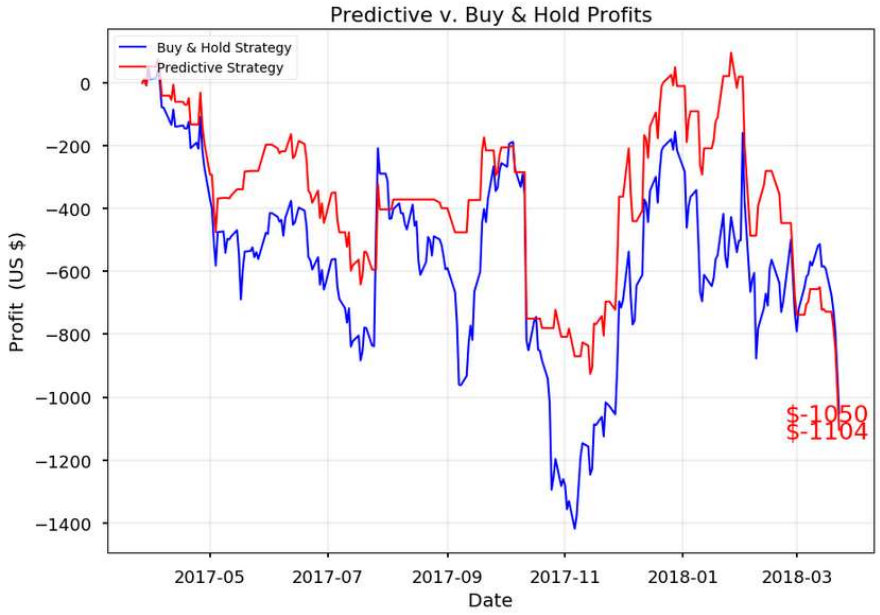
IV. Results

Model Evaluation and Validation

Since historical stock data is being used, we can evaluate the prophet model in stock predictions and compare it with actual stock performance. The model is set to buy stock only on the days that it predicts an increase in price and do nothing on the days it predicts a decrease in price. For the days open with an increase, it triggers a stock purchase, but prices decrease as the day progresses and close, the model register a negative profit.

My theory of seasonality and CPS with error and uncertainty was confirmed with 200 shares of stock from 1/1/2016 to 12/31/2017. Seasonality didn't affect the stock performance significantly and it was removed. A higher CPS produced the closest price predictions. The profit between the predictive strategy and benchmark of buy & hold was the closest in all the scenarios modeled for this project.





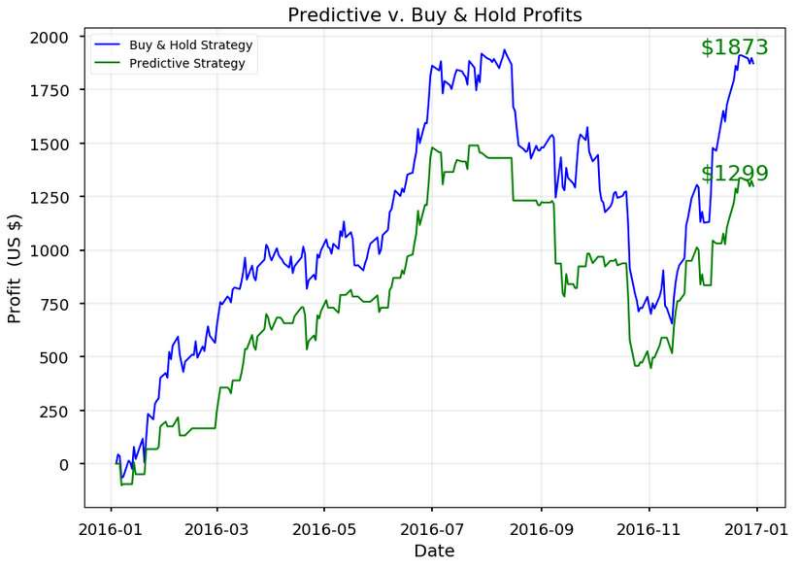
The date range selection affected the overall performance—positive versus negative; depicted by graphs above, with results from different timeframes. The date range mattered more significantly than I initially thought, where if the selected range of dates are for a period with mainly stock price increases, the profit is positive. The same is true for selecting a period where the stock price decreased consistently, the profit is negative. One could 'game' this project for only selecting changepoints priors, confidence interval, and dates that yield the maximum profit.

Hindsight is 20/20! If only we can use historical stock data to complete trades back in time, we'll beat the market every time. This project reaffirmed Harry Markowitz's assertion in Efficient Market Hypothesis (EMH) that the stock market is not predictable.

Justification

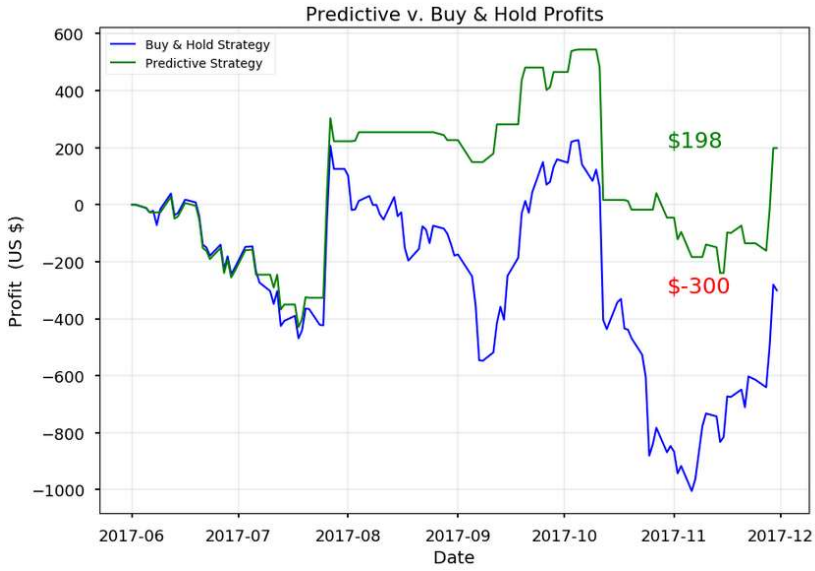
No matter the changepoint chosen, the benchmark model outperformed the prophet model in return on investment or profits.

```
The price increased when predicted an increase 55.94% of the time.
The price decreased when predicted a decrease  50.93% of the time.

Total profit using the Prophet model = $1299.22.
Buy & Hold strategy profit =         $1873.37.
```



With the knowledge of hindsight, we can beat the market if the period range was relatively short, from 6/1/2017 to 11/30/2017.

```
The price increased when predicted an increase 50.00% of the time.
The price decreased when predicted a decrease  55.00% of the time.

Total profit using the Prophet model = $198.37.
Buy & Hold strategy profit =         $-300.67.
```



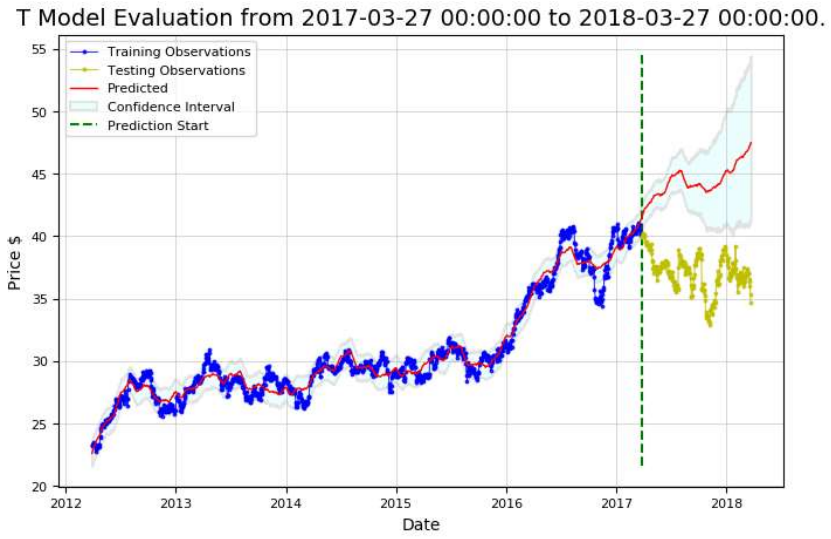
If only it was that easy to work the market in our favor, we would be set. There are other attributes that can be reviewed and tweaked to improve the model's profitability but then we would risk overfitting the model. The model did perform as intended to predict the stock price and overall profits with selected date ranges and to compare it with the benchmark.

V. Conclusion

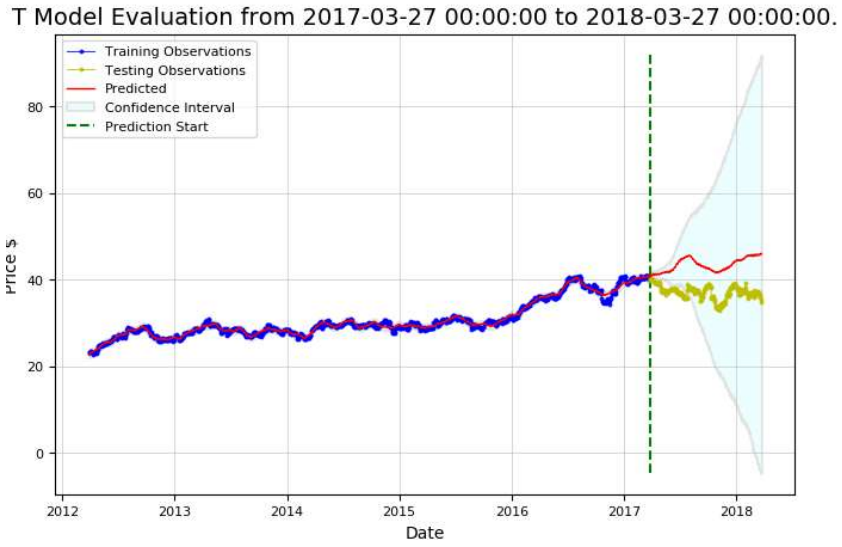
Free-Form Visualization

The most relevant part of this project was the construction of a viable predictive model where the training and testing observations were used to evaluate the modeled prediction, with and without weekly seasonality.

With seasonality:



Without weekly seasonality:



The main goal of this project was to explore the use of Machine Learning, specifically using fbprophet in stock prediction for daily trading. It was used with AT&T (T) historical data to evaluate and compare the predictions against actual market performance. The model's aim is for the modeled T stock price to beat the actual market performance. Despite the market or benchmark outperforming the model, it was still important to learn, construct, and tune the predictive model to understand how the different attributes affected modeling. It provided a good appreciation into the intricacies of what stock analysis entails.

Reflection

The learning curve for fbprophet was steep, since it was new to me. I like that it incorporated a lot of the machine learning principals in the changepoints, error, and uncertainty calculations. It also provided the option for users to manually change and tune the default changepoint range, changepoint prior scale, confidence interval, and trends options.

The model achieved the main goal of this project, which was to predict the stock price and compare the prediction to the benchmark of buy & hold. The model's daily price prediction is based on the previous day's stock price. As the prediction progress further and further into the future stock prices, there was no trend to assist the changepoints, consequently the error and uncertainty rates increased.

Stocks have always held my interest, but this project just reminded me how elusive the reality is to accurately predict the market and beat it. There are so many variables to consider on top of the nature of the specific industry the stock belongs to. I didn't get the chance to test the model with another stock or set of stocks to see if it performed the same, better, or worse than when it was used against AT&T (T) stock. I may continue to fiddle with it beyond this project. I'll continue to explore fbprophet for sure as I can see many real-world on-the-job applications where I can use it.

If the stock market were easy to predict, everyone would be doing it and getting rich.

Improvement

The model was constructed using historical stock data for trend observations and changepoint identification. What if those trends are in the future? There are no practical ways to predict the changepoints of future trends and that is why the uncertainty level increases as the prediction model gets further and further into the future. The model assumes future trends will be somewhat like the observed trends in history and applies a variation or average of those observed changepoints. We can increase the changepoint prior scale (CPS) to add more flexibility to the model for trend prediction, but it also increases the level of uncertainty and may overfit the model. The opposite can be said for decreasing the CPS, which would decrease uncertainty but increase the likelihood of underfitting the model.

Additional areas of improvement, if time wasn't an issue, would be to explore and incorporate additional trends into the prophet modeling analysis. Real-time trends or alerts can be incorporated into the model by adding something like Google Pytrends, where specific search activity of the company could be correlated and affect the prices positively or negatively. This can include specific events, such as the recent deals AT&T struck with IBM and Microsoft for part of the cloud segments, or AT&T's subsidiary DTV

satellite being down or its inability to negotiate the broadcasting rights for a crucial event such as the highly watched NFL Super Bowl.