# DA 516 Final Project
# Forecasting Bitcoin price using Machine Learning

**Ramesh Mainali**[1]

[1]Catholic University of America

***Abstract.*** *Having an ability to forecast future price of stock has tremendous financial interests to the investors. However, traditionally it has been extremely difficult to develop tools to accurately predict stock price since its value depends on several unseen factors. With the advent of several machine learning and deep learning algorithms in the recent years, the interest to use historical data to make data driven forecast of stock has revamped. In this project, I worked on historical data of Bitcoin to predict its future value using deep learning LSTM model and Facebook Prophet. I performed a detailed data mining and trained the two models to predict future Bitcoin price. Finally, I discuss the implications of machine learning application to time series stock data.*

## 1. Introduction:

The popularity of machine learning data driven approach has grown tremendously in financial market. In recent years, we have seen financial sectors adopting varieties of machine learning approaches for predictive analysis. One such example is the use of machine learning analytic in the stock market. For investors, it is incredibly valuable to make correct prediction of stock price. A successful prediction could yield significant profits to both investing firms as well as retail investors. However, traditionally it has been very difficult to make such prediction on stock market since varieties of unseen factors come into play in determining the stock price. Recent development of machine learning and deep learning algorithms may present opportunity to make first stride toward a successful forecasting models. The primary goal of this project is to leverage different machine learning method to predict future stock price.

For this study, I selected a crypto-currency Bitcoin which has been very popular in recent times. Bitcoin is a digital currency which uses peer-to-peer technology to facilitate instant payments. It was first created in 2009, but its popularity has grown exponentially over the past five years. This is demonstrated by the fact that the Bitcoin has grown over 100 times in the past five years. In this project, I study machine learning applications to forecast future price of Bitcoin.

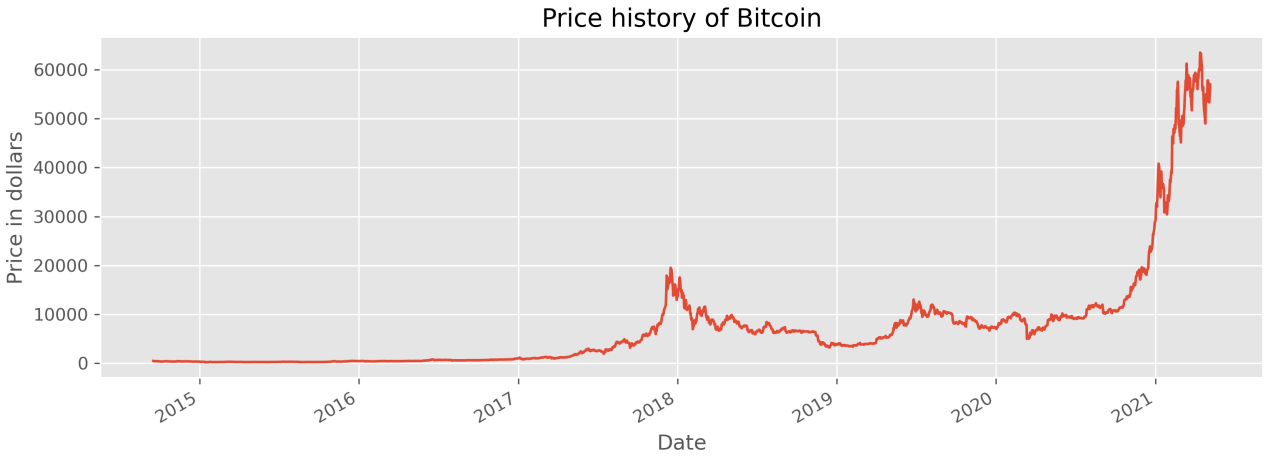


**Figure 1. Plot showing the historical stock price data of Bitcoin from 2014 to April, 2021. The x-axis shows the date and y-axis shows Bitcoin price in dollars.**

## 2. Data:

The historical stock price of Bitcoin is collected from Yahoo Finance. A publicly available API called "yfinance" allows effortless download all historical stock data. The downloaded Bitcoin dataset has 2419 rows and 7 columns. The 2730 rows indicates daily stock price over the past 2419 trading day. The seven columns indicates the following features:

1. **Open:** It is the stock price of the very first transaction in a given business day.
2. **Close:** This is the price of stock for the last transaction of the business day.
3. **High:** This is the highest price of the stock that traded in a particular business day.
4. **Low:** Similar to High price, Low is the lowest price of the stock traded in the business day.
5. **Volume:** In stocks, the volume represents the number of shares traded in a given business day.
6. **Dividends:** It is an amount paid by company regularly to its shareholders.
7. **Stock Splits:** Stock splits increases total shares of company without actually impacting total market capitalization of the company.

I first downloaded the data set and examined for any null values in the data. The data lacked any null value. In Figure 1, I plotted the historical Bitcoin price ("Close price" in this case) of the Bitcoin to examine how the its price evolved over the time period. The bitcoin price first peaked at the end of 2018, slowed down a bit and remained mostly steady until mid 2020. Toward the end of 2020, bitcoin pick up once again at a rapid pace, growing by a factor of over 6 in a matter of few months.

### 2.1. Feature Selection:

First of all, I removed the "Dividends" feature from the dataset. I found all the entries were zero (’0’) for dividends, which makes sense for any crypto currency. I then calculated

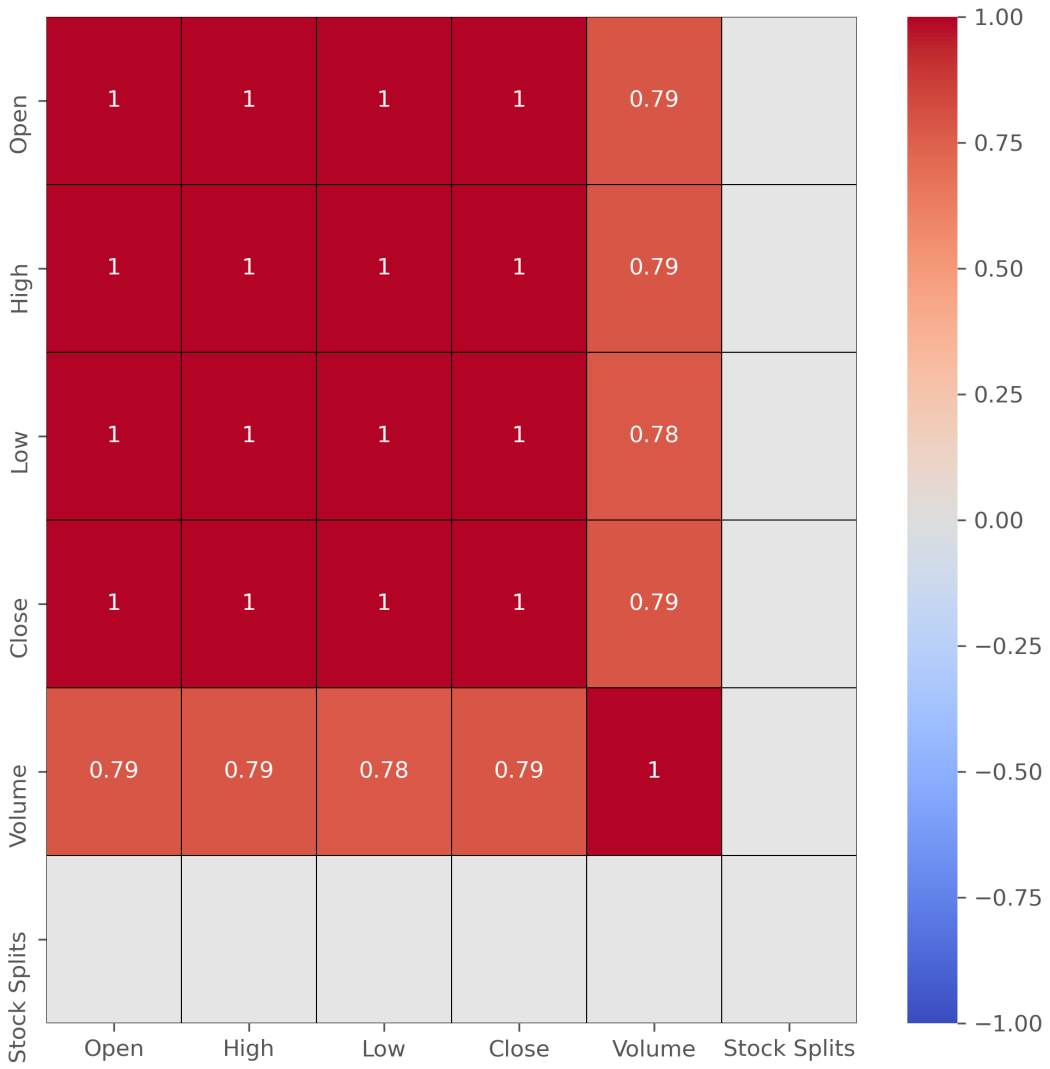


**Figure 2. Plot showing correlation map of various features in the Bitcoin data set. We can see that different stock prices ('Open', 'Close', 'High' and 'Low') are highly correlated. Stock Splits feature doesn't show any correlation with any prices.**

the correlation coefficients for remaining features. In Figure 2, I show correlation map showing correlations among six remaining features in the dataset. As we can see in the figure, the correlation coefficient is equal to 1 between all four different prices (i.e. Open, Close, High and Low). As such, for the historical datasets we can only keep one price and leave other prices. For the further analysis, I choose to keep "Close" price as the stock price, and remove other prices from the dataset. Also, "Stock Split" doesn't show any strong correlation with price. So, I dropped that feature from my analysis.

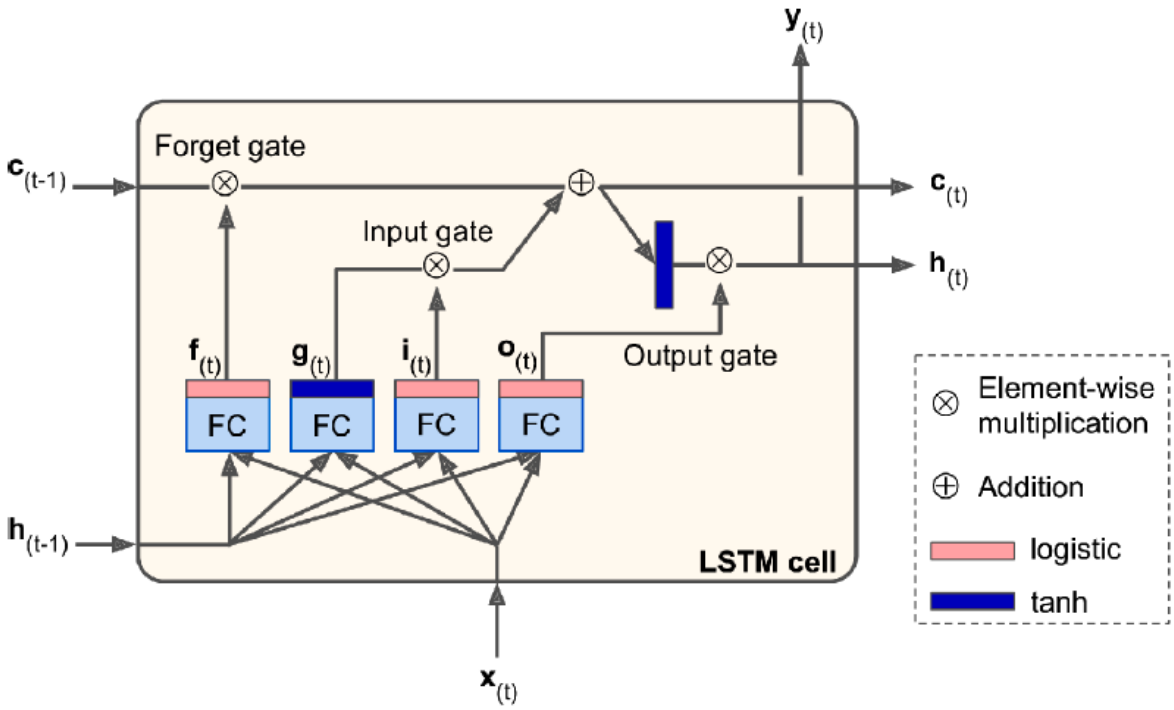


**Figure 3. LSTM cell taken from the course book.**

### 2.2. Data Preparation:

Since the dataset is time series in nature, and the goal is to predict future price of the stock, I keep last 200 days stock price as testing dataset for modeling purposes. For training dataset, I select the first 2219 data sets. The model trained to analyze the behaviour from training dataset will be used to make prediction and eventually compared with testing dataset.

The resulting training dataset is scaled to lie in the range of 0 to 1. For this, I used MinMax scaler from scikit-learn package. The same transformation from the training dataset is then used in the testing dataset in order to prepare for the modeling and evaluation.

## 3. Time Series Forecasting:

I considered two methods to model the time series Bitcoin price. The first method is a special type of Recurrent Neural Network, known as Long Short Term Memory (LSTM) and the second one is open-source library known as Facebook Prophet. Below I briefly discuss the two types of the methods:

### 3.1. Long Short Term Memory (LSTM):

For time series analysis, a special class of deep learning network, known as Recurrent neural networks (RNNs) is found to perform well. In contrast to a traditional neural

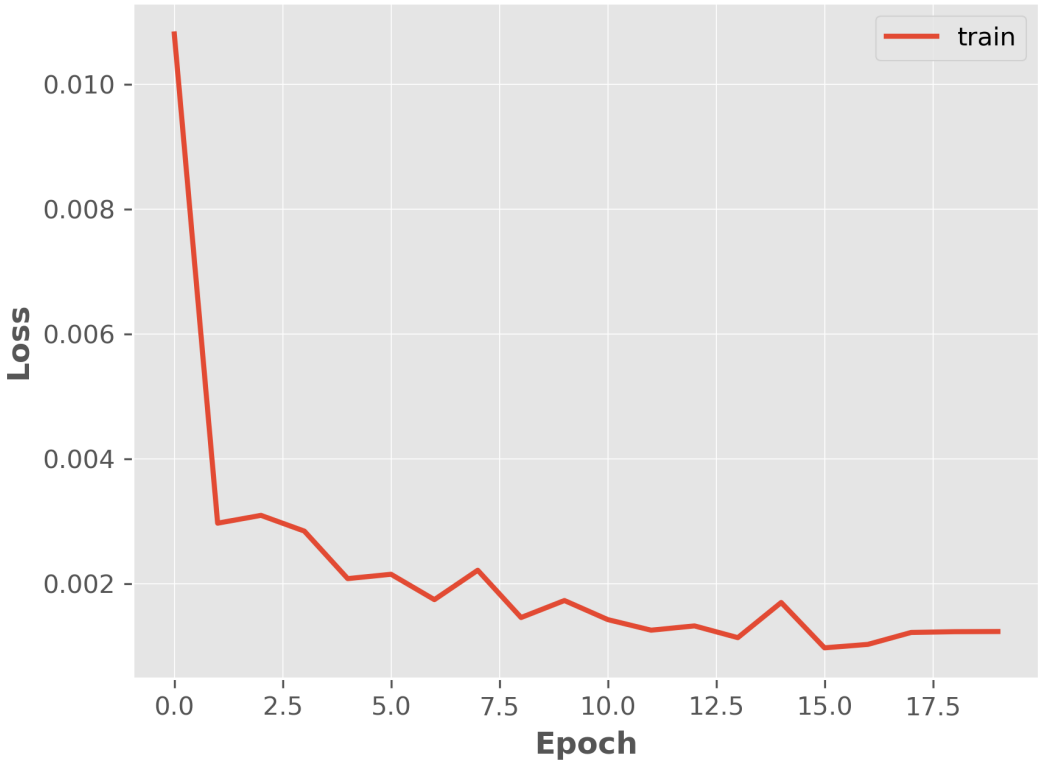


**Figure 4. Plot showing loss curve for LSTM model on training data sets.**

network, Recurrent neural network takes output of a particular layer and feeds this back as input to the layer. This method has found tremendous application in sequential data like speech recognition and natural language processing.

While RNNs perform well on short term memory, they can hardly be used for long term dependencies. As the time steps increases, RNNs find it difficult to keep track with it. In order to deal with long term memory capture, a special type of recurrent neural network known as long short term memory (LSTM) has grown in popularity. A LSTM architecture comprises of a structure called memory cell and various gates (see Figure 3). The gates include input gate, output gate and forget gate. The cell allows transfer of sequential information for the whole sequence. On the other hand, gates serve to modulate these information. These gates are different neural network which learn to keep or forget relevant information during the data training.

The goal of this project is to use the historical data and train them using neural networks in order to make reliable forecast. As such, I use LSTM model to train the historical dataset which may allow the model to learn the forecasting ability based on the past data. LSTM model in particular allows the model to learn from the past serveral data

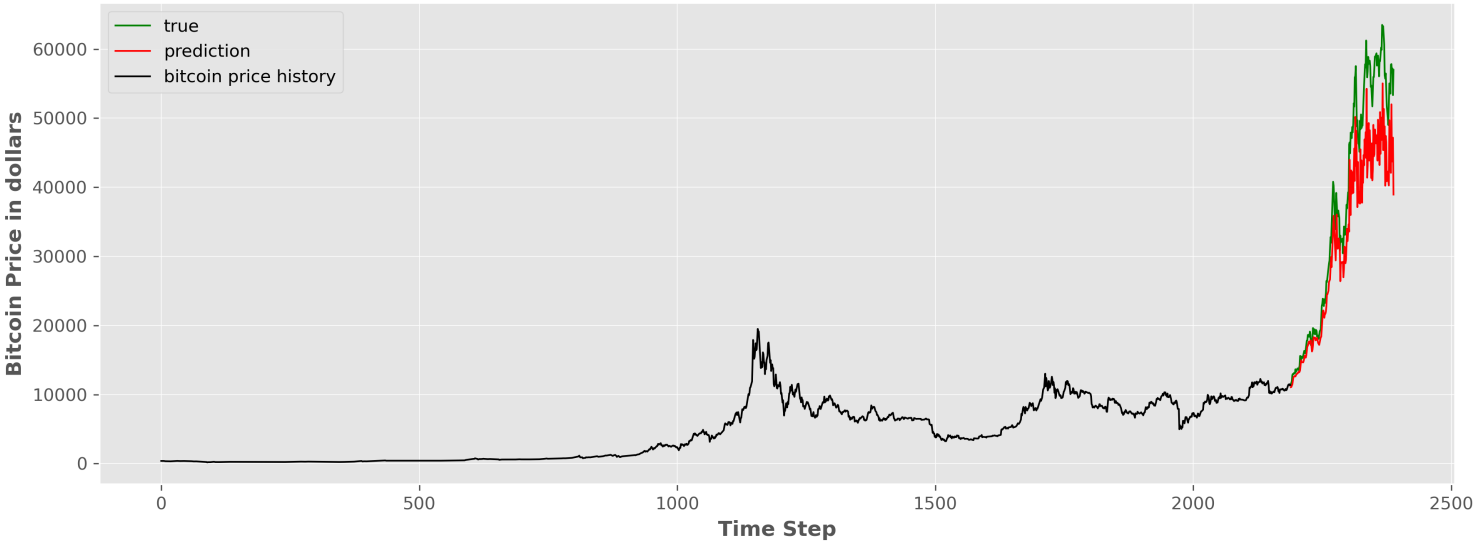


**Figure 5. Plot showing stock prediction from LSTM model. The black curve shows historical data sets. Green curve shows actual Bitcoin price over the last 200 trading days. Red curve shows predicted value using the LSTM model over the last 200 trading days.**

set in order to find the pattern for future prediction.

I then build a LSTM using "keras" sequential model. The first layer of the LSTM model is selected to have 256 units. I then select dropout of 0.2 which means the output of 20% of hidden cell is set to zero in order to prevent over fitting. In total four hidden layer is created, which is finally connected to Dense layer. This layer ensures the neurons from previous layer is fully connected. The number of unit in the last Dense layer is set to be 1, since we are interested in predicting a single stock price value. For activation function, I used the default "tahn" activation function. The model is trained with the training dataset prepared in the previous step.

The model accuracy is then estimated by calculating R2 score of the predicted stock price against the actual stock price. The result from the LSTM modeling is shown in Figure 5. In the figure, the black curve shows the historical Bitcoin price (also training data), the green shows the bitcoin price for the latest 200 days, and red curve shows the LSTM prediction for the last 200 days. The prediction and actual data for the past 200 days is further shown in a zoom in figure 6. The Figure 6 suggests that the model perform decently with R2 score of 0.66. This means the model can explain 66% variance in the data.

However, upon looking more closely into the model prediction, I noticed that the result shown in Figure 6 might actually be misleading. Any model predicting with such

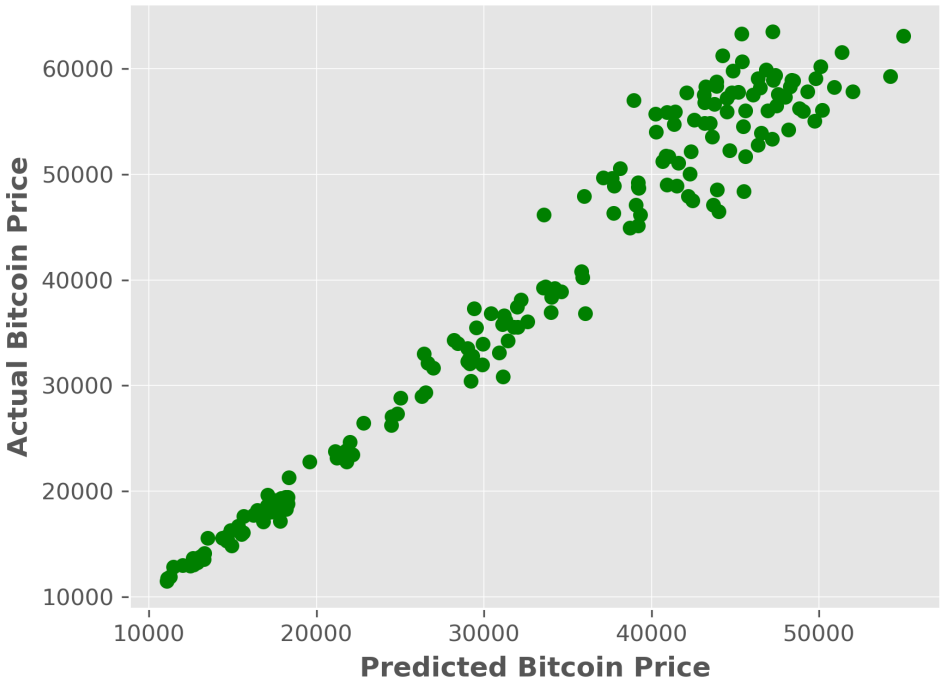


**Figure 6. Plot showing predicted vs actual price of the Bitcoin from the past 200 business days. The two values have r2 score of 0.95, suggesting that the LSTM model can explain 95% variance in the data.**

accuracy would fetch millions, if not billions of dollars in stock market. I think this issue arises because LSTM actually only make prediction for the next day, and relies on past actual data (look back time) to make prediction. If the model really succeeded in predicting the Bitcoin price, then it should make good prediction for daily change in price. This is because when the model predicts for next one day, the daily price change should be the most robust prediction of the model. So, in the Figure 7, I plotted the model predicted daily price change and actual daily price change for the last 200 days. As we can see in the plot, the two values do not match with each other, and the data points are all over the plot. This suggest that the LSTM model cannot reliably predict the daily price change of Bitcoin. The model prediction shown in figure 5 simply comes from the persistence of the past data in the model.

### 3.2. Facebook Prophet:

Facebook Prophet is an open-source library released by Facebook in 2017 which is mainly used for forecasting time series data. Facebook Prophet has found great success in forecasting time series data using intuitive parameters and introducing seasonality and holiday effects in the modeling. It is mainly based on three decomposable models - general trend in the data, seasonality and holiday effects. This can be simplified in the below equation:

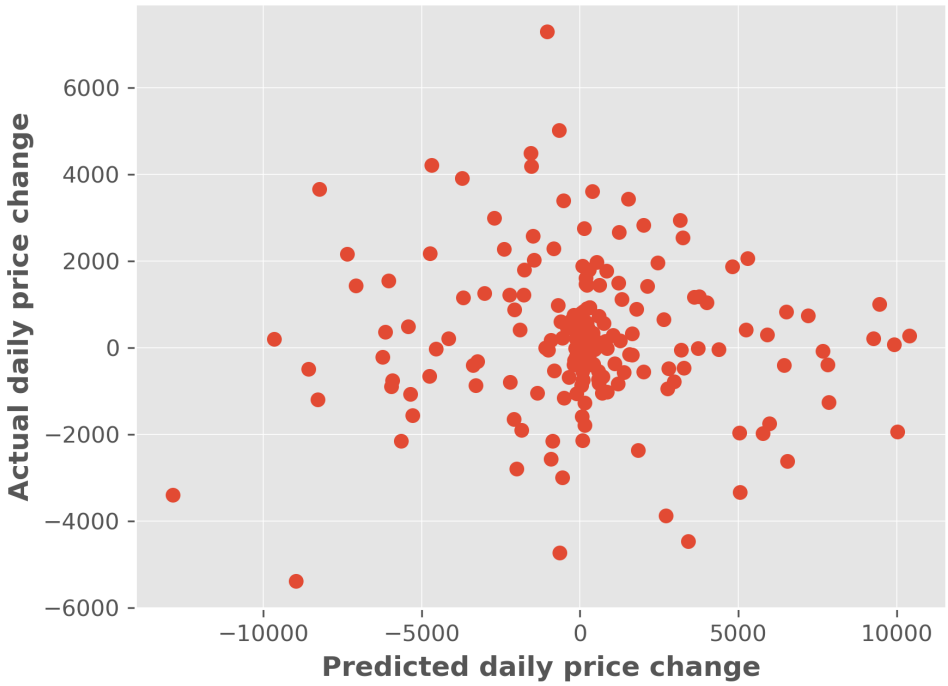


**Figure 7. Plot showing model predicted daily price change vs actual daily price change of Bitcoin in the most recent 100 business days. The data point are all over the plot, suggesting the model can't reliably predict per day price change of Bitcoin.**

$$y(t) = g(t) + s(t) + h(t) + e(t) \tag{1}$$

where,
y(t) : denotes the prophet model,
g(t) : denotes the general trend in the data or growth over the time,
s(t) : denotes the seasonality effects in the data,
e(t) : denotes the noise in the data,

In order to train the model, I first split Bitcoin dataset into training and testing sets. The training set consisted of all historical data except the most recent 100 Bitcoin prices. The Prophet model is trained with the training dataset using the default model parameters. Then the output for the next 100 days is forecasted using the trained model. In Figure 9, we can see that the Facebook Prophet trained model is actually divided into three model components: trend, weekly seasonality and yearly seasonality. The prediction from the model is then plotted in Figure 8. As we can see, the prophet model succeeded in predicting overall trend in the data. However the model prediction is off from the actual Bitcoin price.

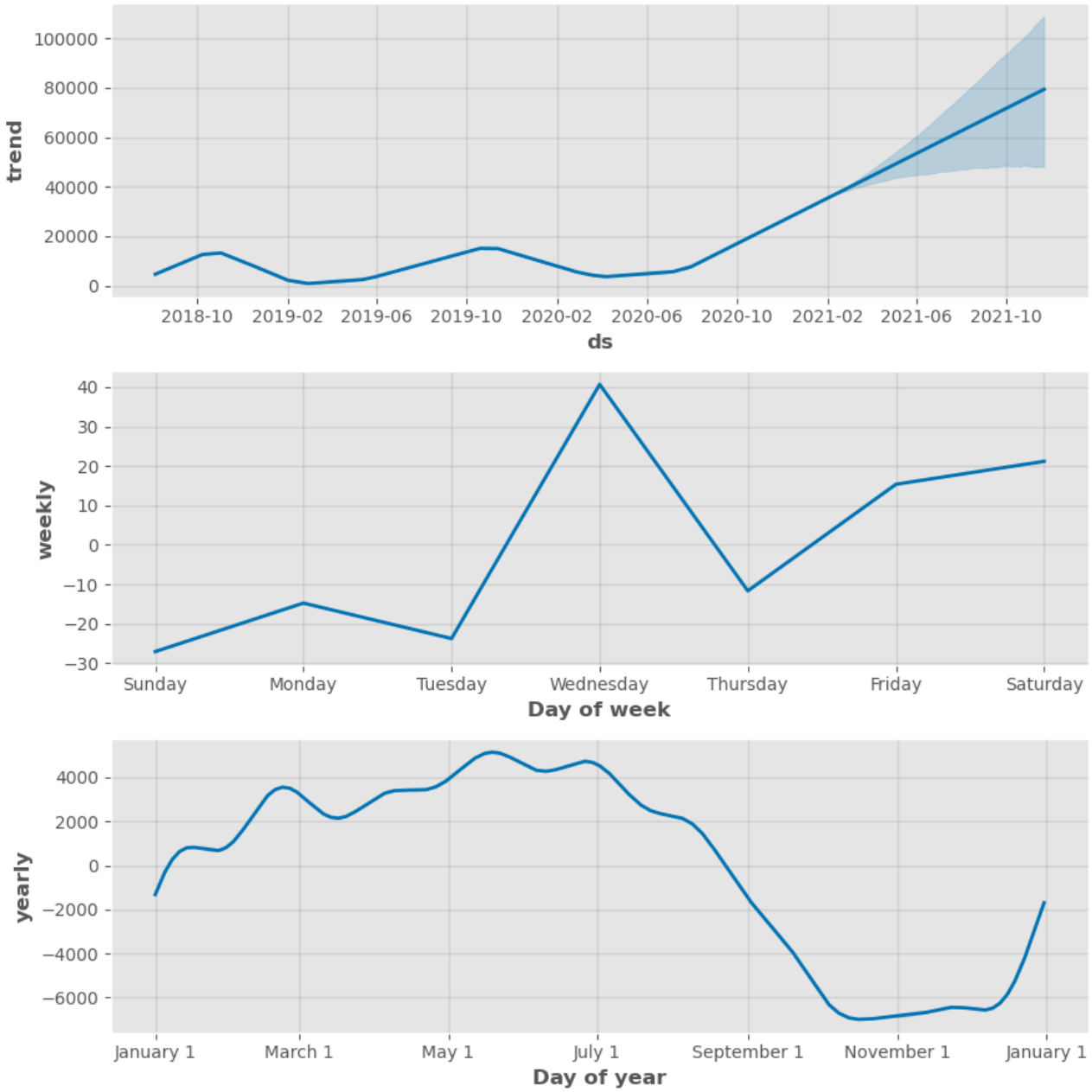


**Figure 8. Plot showing different model components of Facebook prophet. On the top, we see the general trend in the Bitcoin price, in the middle weekly seasonality and in the bottom yearly seasonality of Bitcoin price as modeled by Prophet.**

## 4. Discussion and Summary:

In this project, I considered two time series forecasting Machine learning methods, LSTM and Facebook prophet, to predict future price of Bitcoin. At first glance, LSTM appeared to well predict Bitcoin price. However, such interpretation was misleading as the model was only using the recent data to infer next day price, which falsely appeared to make right prediction. Upon checking for daily price change, it is clear that the model lacks


**Figure 9. Plot showing stock prediction from Facebook Prophet. The black point shows historical data sets used for training the model. The blue curve shows the model predicted price. The red points represents the actual data point for the last 100 days.**

the predictive ability. On the other hand, Facebook Prophet is found to predict the overall recent trend in data. However, the model still lacked ability to accurately predict Bitcoin price.

In general these models only takes into consideration historical dataset, but stock market appears to be driven by many other external factors. For example, even a random tweets from an influential person can drive the price of a particular stock. In that sense, Bitcoin price, or stock price in general is difficult to reliably model.

## 5. References:

1. https://www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/
2. https://medium.com/future-vision/the-math-of-prophet-46864fa9c55a
3. https://towardsdatascience.com/predicting-stock-prices-using-a-keras-lstm-model-4225457f0233