Md Saimoom Ferdous

20 July 2020

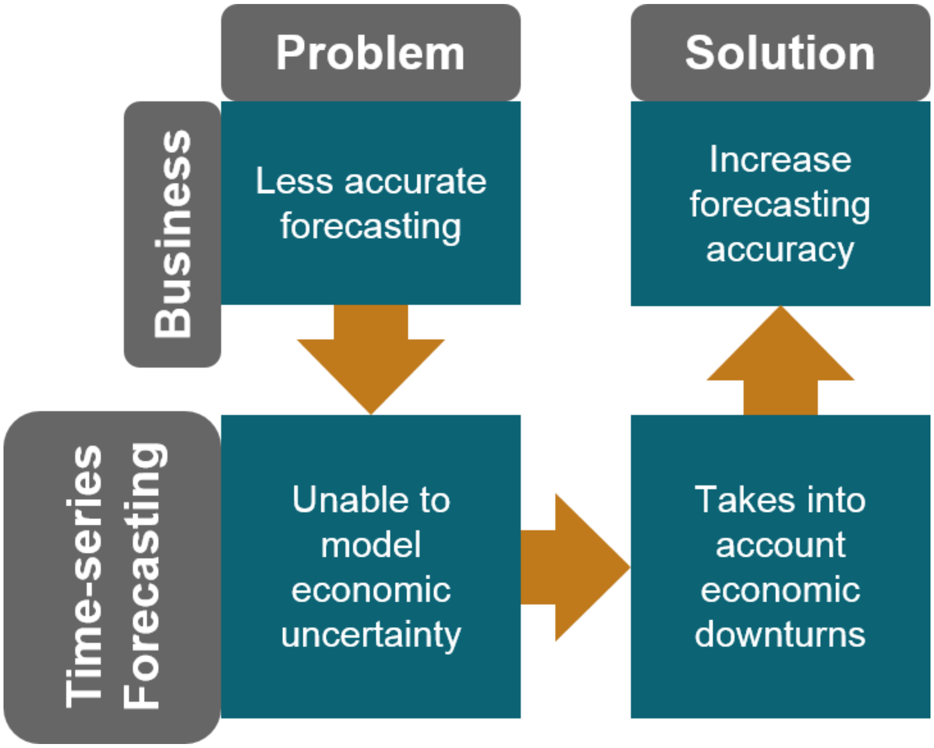
Apple Stock Price Prediction: Final Report

GitHub Link

Table of Contents

Problem Statement

Project Flow



1. **Start with the business problem**
   Apple's stock price forecast is of great interest for millions of individual investors, stakeholders, suppliers, manufacturers. Accurate forecast can yield significant profit gain.
2. **Convert business problem into time-series forecasting problem**
   Conventional models are good for short term forecast (~a year). They cannot consider holiday effects or any trend change (e.g. economic downturn) in history that needed attention in prediction modelling.
3. **Solve time-series forecasting problem**
   FBProphet, an open source library of Facebook, can incorporate holiday effects on top of paying attention to the exceptional circumstances in terms of trend change model.
4. **Convert time-series forecasting solution into business solution**
   Better accuracy can yield significant profit margin. Better long term accuracy can help financial planning of the company itself and the investors, stakeholders

Business Understanding

Apple Inc. is one of American the big tech multinational technology companies, headquartered in California, that is known for the fine design, development and selling of consumer electronics, computer software and online services. It is one of the top 10 Fortune 500 companies and ranks top in the Nasdaq stock exchange. Its stock price is closely watched by millions of people around the world. Short, medium, and long-term stock prediction is of interest for investors, companies to plan their financial strategy well ahead of time. During the economic downturn stock market prediction becomes even more complex. Given the long historic data that captures rare economic downturn events can provide insight for better prediction coupled with relevant features (dividend, earning). This project will aim in that pursuit.

Time-Series Forecasting

The primary objective is to come up with a better forecasting model for Apple's Stock price prediction. The conventional models can consider for past couple of years. It is observed that, during, before or after holidays financial behavior changes and can have effect on the stock price. Also think of the economic recession in 2008 and sharp price fall in 2014. These types of sudden trend changes if incorporated into the forecasting model, can yield better forecasting model.

Data Collection

Data Source

Stock price data was collected from the Kaggle website sourced from NASDAQ, Yahoo finance, Zacks, Alpha vintage. Data consisted of 700+ companies over the years of 1998 – 2020. Along with the stock price information, dividend and earning information came up with the data source.

Getting to Know around the Data Set

The data package consisted of four files: dividend, earning, stock price and summary. The Dividend file consisted of the following columns:

Dividend	
symbol	symbol of a company under which it operates in stock market
date	dividend issue date for a share
dividend	proportion of dividend issued

The 'earning' file looks like:

Earnings	
symbol	symbol of a company under which it operates in stock market
date	earning issue date
qtr	the month and quarter (Q1, Q2, Q3 and Q4) of the year the earning was declared
eps_est	estimated eps (earning per share)

eps	exact eps
release_time	earning issue date before or after the declaration date

The ‘stock price’ file has the following columns:

Stockprice	
symbol	symbol of a company under which it operates in stock market
date	trading day of the year
open	opening price in a day
high	high price for a day
low	low price in a day
close	closing price for a day
close_adjusted	amended price that truly reflects stocks value after any corporate actions
volume	total volume traded in a day
split_coefficient	the ratio by which a firms outstanding share increases following a stock split. Higher is the 'stock split' reduced the price would be

The ‘summary’ file contains the summary of the other dataset, which will be picking up in the exploratory data analysis stage. Therefore, it will not be used for the project.

Data Wrangling

Data wrangling consisted of cleaning up the data by removing NaN in the dataset and looking for duplicates.

Removing NaN

NaN values were only found in ‘dividend’ dataframe. Looked closely, ‘eps’ got 21.5 % NaN which was seen missing during and after economic recession in 2008. Apart from the missing ‘eps’, ‘eps_est’ have 21.4% missing values which were assigned with the ‘eps’ value that would be a close estimate of ‘eps’. The ‘release_time’ had 28% missing value which were forward filled always in the history release time was declared after the eps announce date. The ‘qtr’ had 0.8 % missing value which information were filled from the date column

Duplicates

No duplicated rows were found.

Save Apple Specific Data

The dividend, earning and stockprice information were merged for only Apple and saved for exploratory data analysis (EDA).

Jupyter Notebook for the Data Wrangling part can be found in this link.

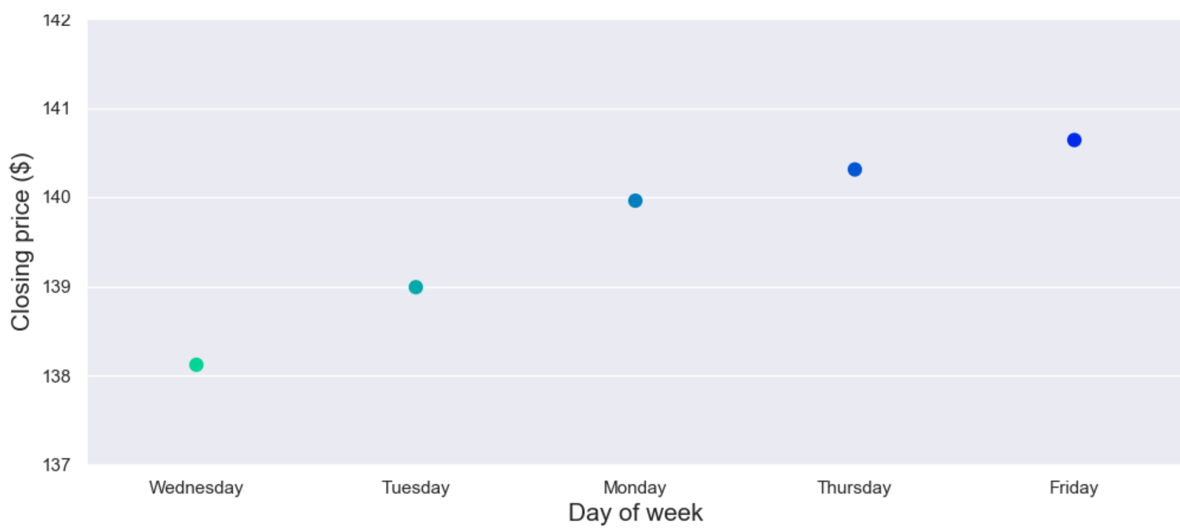
Exploratory Data Analysis

Time-series stock price data along with dividend and earning data can be very insightful in determining, daily, weekly, monthly, quarterly behavior of stock price. Correlation between variables will also be determined. Finally, only the variables relevant for forecast modelling will be saved.

Seasonal Price Variation

Day of the Week

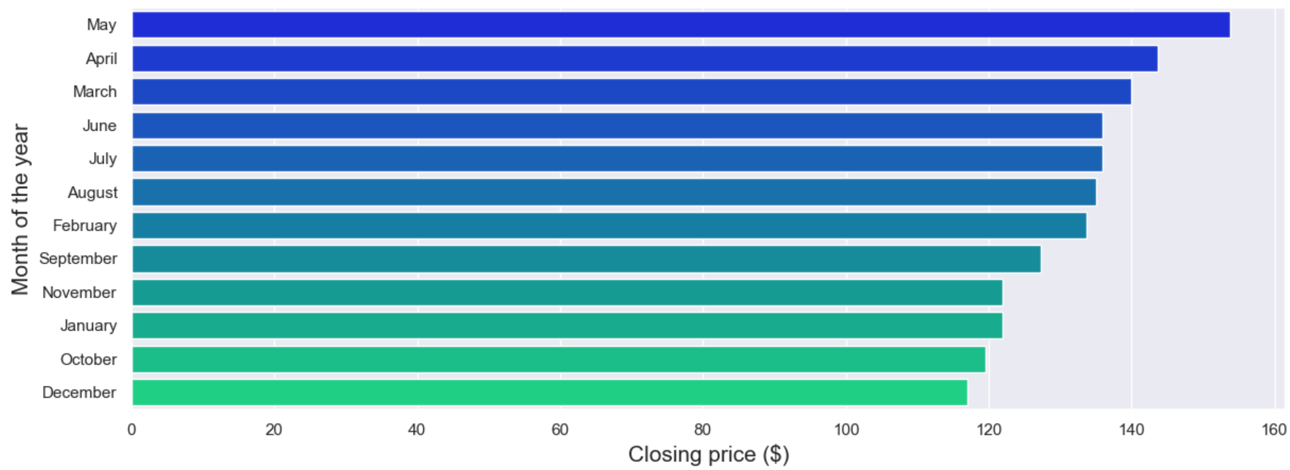
Stock market is open on Weekdays only (Monday to Friday). Closing price was grouped by day of the week and taken 'median' over the years. Median was chosen over mean because, it will put more emphasis on regular prices and less on unusual time such an economic down turn's.



In a week, Friday saw highest average closing price, whereas Wednesday saw the lowest.

Month of the Year

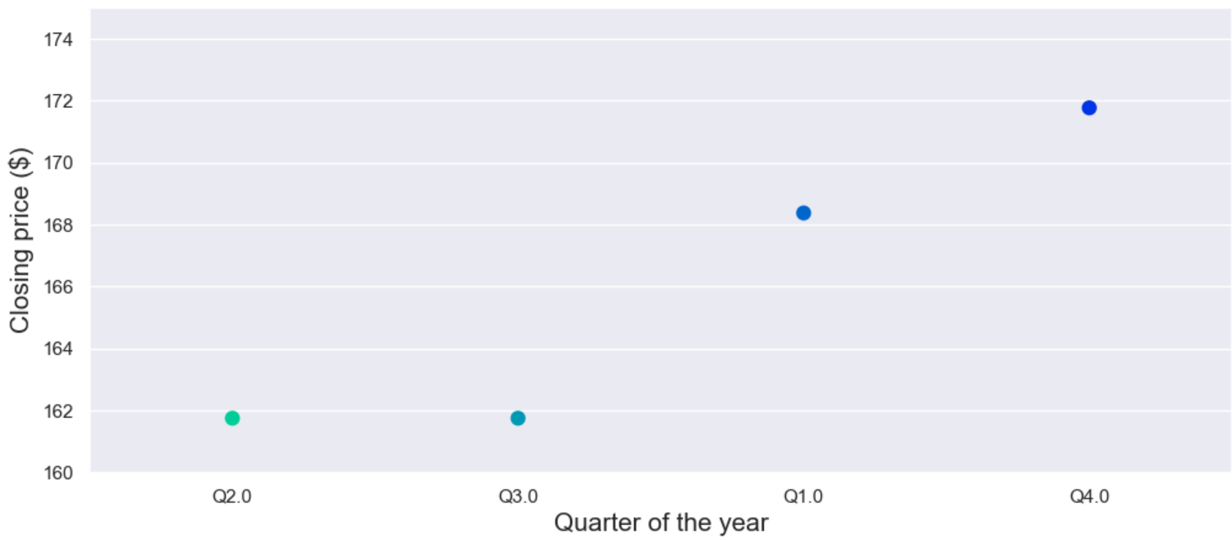
Historically, it was determined which month of the year saw the largest closing price. Closing price was grouped by day of the week and taken 'median' over the years



Month of May saw the highest closing price whereas December saw the lowest.

Quarter of the Year

Determined which quarter of the year saw the largest closing price. Closing price was grouped by quarter of the year and taken 'median' over the years.

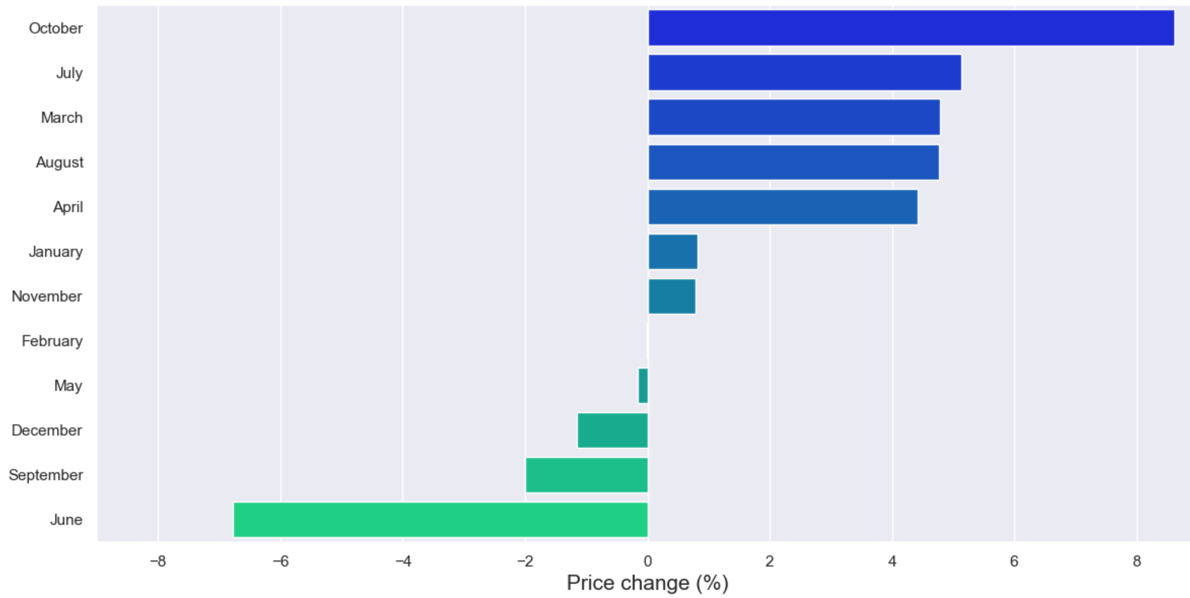


Last quarter of the month saw the highest price followed by the start of the year.

Stock Price Change

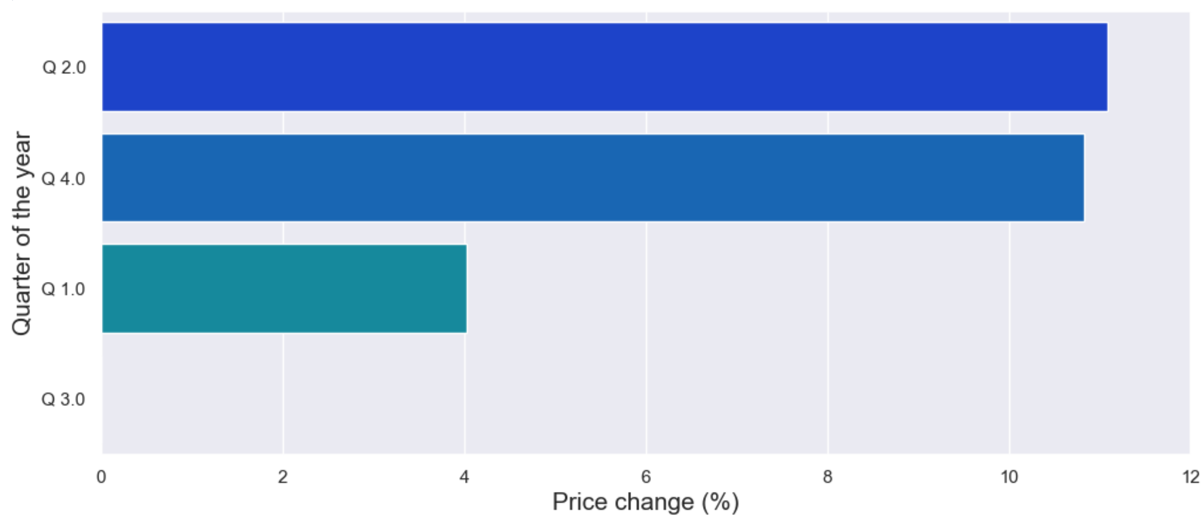
Stock price change (%) over a time duration is a good measure of how investment is growing.

Month of the Year



Apples share holders saw biggest return over the month of October and lowest return in June.

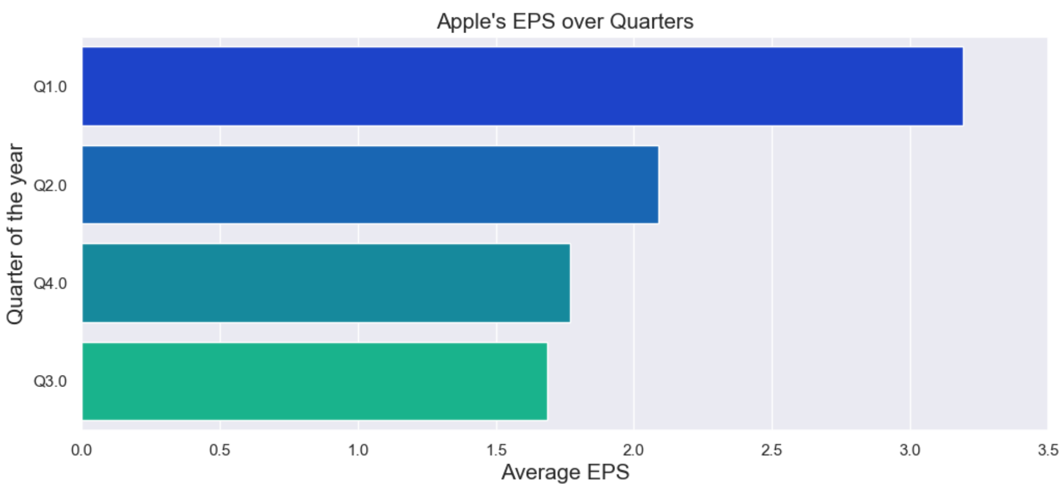
Quarter of the Year



Second quarter of the year saw the biggest return (%) followed by the Q4, whereas Q4 saw the highest quarterly rice shown earlier.

Earning per Share (EPS) over the Quarters

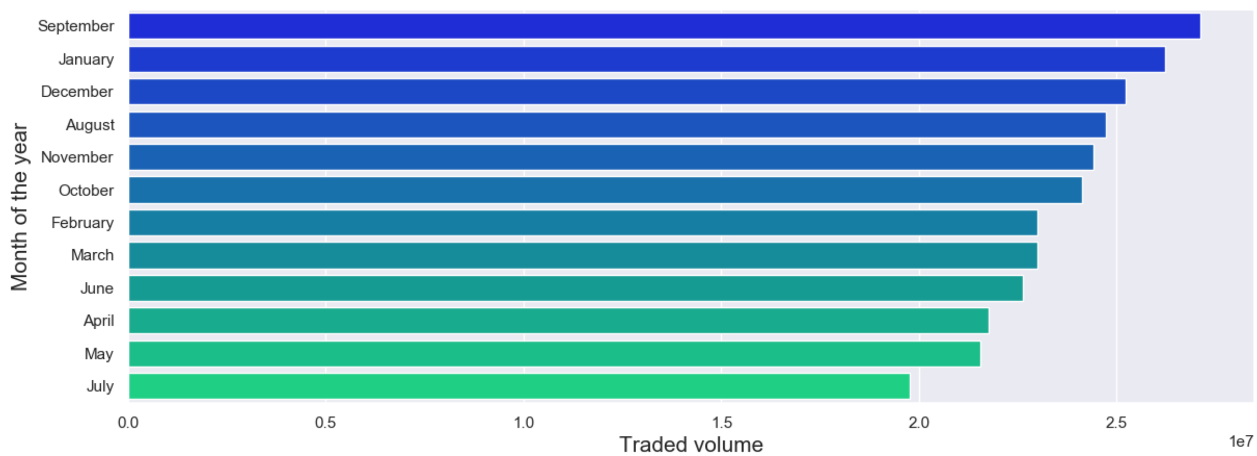
EPS reflects how much revenue the company is generating with the share holder's money.



Apple showed best economic performance in the first quarter (Q1). Just after the Christmas big sale which makes sense.

Traded Volume over Months

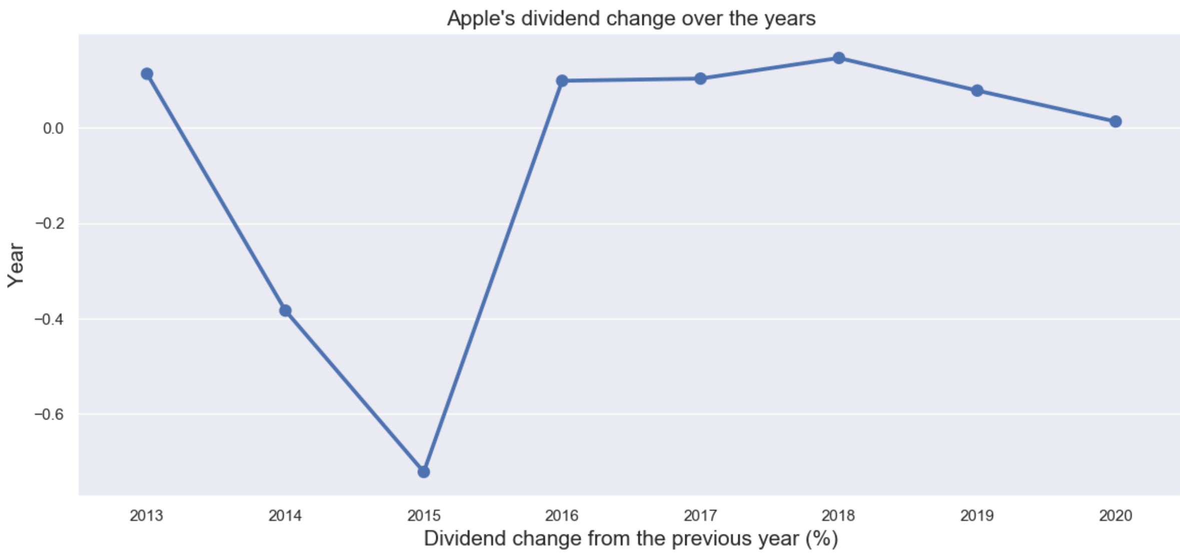
Traded volume data often overlooked while dealing with stock market analysis. Given the availability of time-series volume data, monthly variation of volume over years will be displayed.



Highest traded month is September whereas lowest is July. Looking into the highest closing price month, it is May and lowest is in December. Apparently lowest price is associated with the highest trade volume and vice versa.

Yearly Dividend Paid

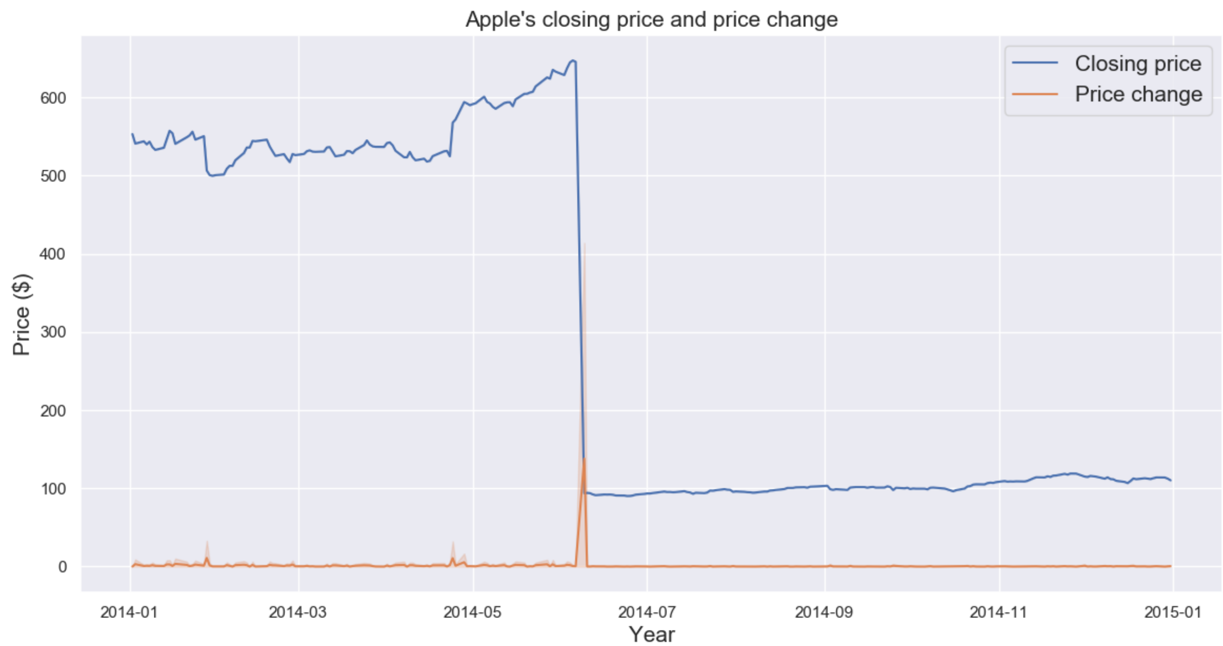
Dividend are share of company's profit paid real to the customers. It is a real income which you can use to buy grocery, go for vacation, or invest back in the market. Higher the dividend better is the chance to increase demand and the share price.



There is a dip noticed in 2014 also visible in sharp price drop during this time. Lately in 2018-2020, although there are increase in dividend but the raise from previous year have dropped.

What happened in 2014?

There is a sharp closing price fall seen in 2014. Let's look closely what triggered the price fall. Apple's stock split happened three times in history, 2000, 2005 and 2014. When stock price crosses 100 $, the company can decide to for stock split to keep the price down to accommodate more investor. In 2014, Apple's [rice soar to 600 $. Just after the stock split by a ratio of 7, price stabilized under 100 $ seen from the following figure. Learn more about the Apple's stock slpit: https://www.cnet.com/news/dont-freak-out-heres-why-apples-stock-is-below-100/



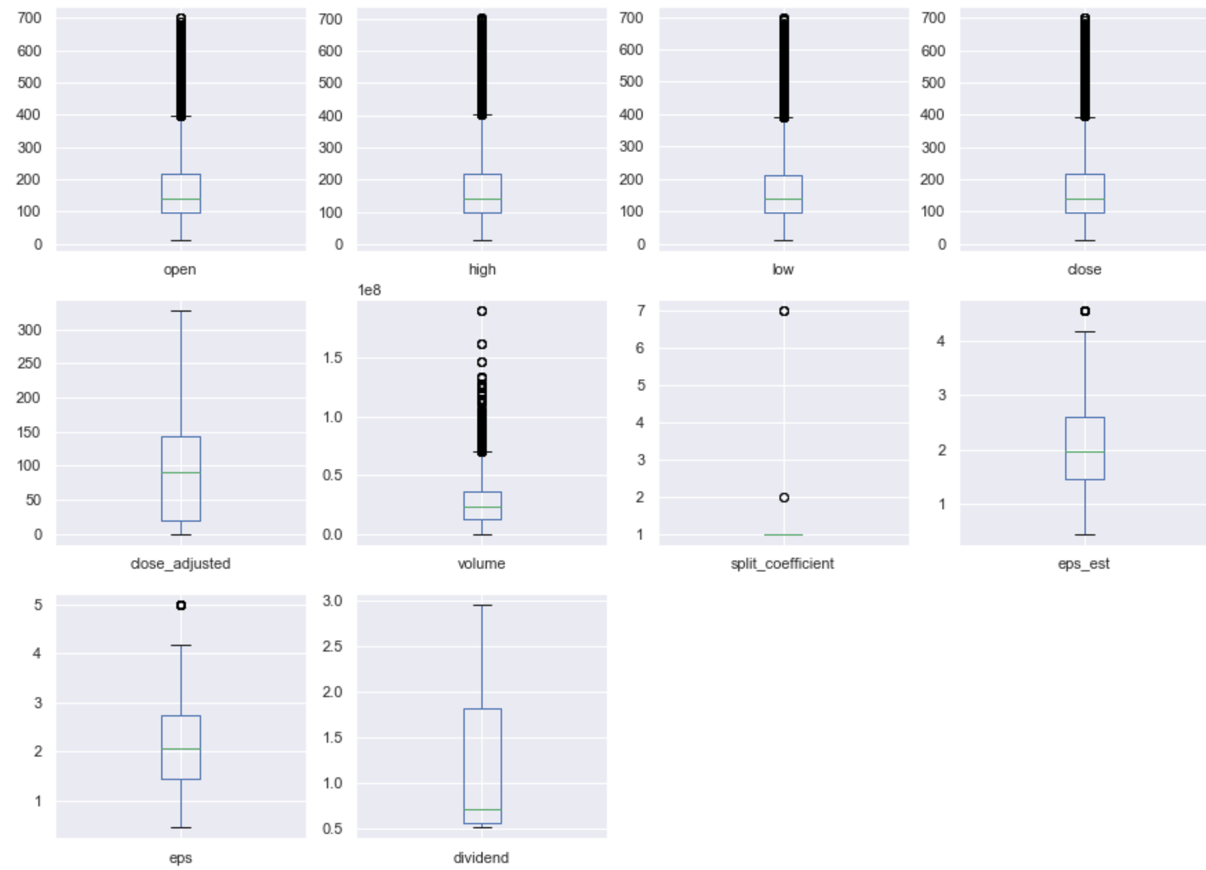
Economic Recession in 2008

Economic recession in 2008 still fresh in the memory for many of us. Let's visualize how the stock's price evolved during that time. In the history, it is known as 'Great recession and officially existed between (December 2007 - June 2009). Link: https://www.investopedia.com/articles/economics/08/past-recessions.asp

Clearly there is a break in the trend change during the recession time (shaded area).

Anomalies and Outliers

Stock price is very unpredictable, and the data could have lots of variabilities. In determining 'Anomalies' boxplot for individual variables were plotted and number of outliers was determined by inter quantile range (IQR).



Anomalous data calculated is 19.58 (%). Which is quite significant and high possibility they are coming from the unpredictable stock price swing.

Correlation between Variables

Based on Pearson's correlation coefficient, the variables will be grouped into high, moderate and low similarities (https://www.statisticssolutions.com/pearsons-correlation%20coefficient/). Here only the highly correlated variables will be displayed.

Highly Correlated Variables

From high to low correlation found between:

- prices (close, high, low, open) ----- dividend (0.80). This tells there is a high chance when price goes up, dividend goes up too and vice versa
- close adjusted ----- eps (0.72)
- close adjusted ----- dividend (0.61)
- eps ----- dividend (0.52)

Feature Removal and Preparing for Modelling Stage

As the goal is to predict stock price based on historical data, only closing stock price data was saved. The low and moderate correlated features are needed when doing regression analysis

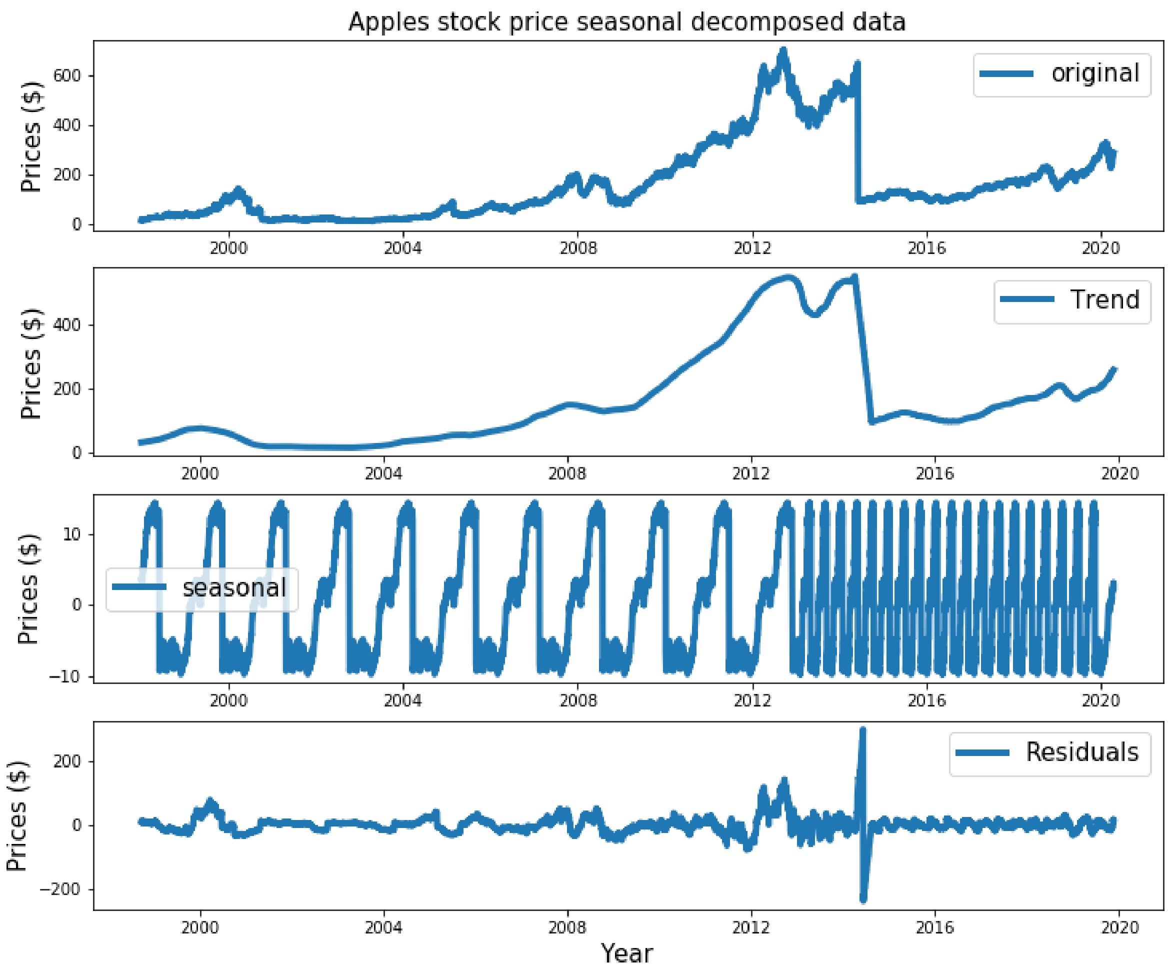
Jupyter Notebook link for the EDA part can be accessed through this link.

Data Pre-processing

Trend-Seasonality Decomposition

Investors always hope the prices would increase day by day. But price fluctuates over days and eventually see some gain over months, quarters, or years. Like any time-series data, stock price can be decomposed into trend, seasonality, and residual parts. This was done with 'statsmodel' library in Python.

- **Trend:** price tendency over a time. e.g. if the price is increasing/decreasing over a year
- **Seasonality:** periodic variation in the price that we see every year. It tells which part of the year price increases/decreases and that happens in cyclic manner over the years
- **Residual:** non-systematic component of the price which is not structured and termed as noise



The trend for Apples stock price is increasing for most of the time, there are few decreasing trends which come from stock splits and market adjustment. Seasonality is constant over time but follows strictly cyclic manner. That is every year there are specific times when stock rises and falls, and the pattern follows every year.

Stationarity Check

A time-series is called stationary if the mean and standard deviation remains constant over time.

Looking into the above figure, it is clear the mean and standard deviation is time varying and non-stationary. Before feeding the data for time series forecasting, the data needs to be tested for stationarity and transformed if required.

There is many testing for stationarity. Of the many, two mostly used stationarity checks will be done based on statistical hypothetical testing.

ADF (Augmented Dickey-Fuller) test:

This method looks for unit root in the series. The hypothesis for this test is:

Null Hypothesis: The series has a unit root (value of a =1) (non-stationary)

Alternate Hypothesis: The series has no unit root (stationary)

If the p-value is less than 0.05 then we can reject the null hypothesis i.e. the data is stationary and the data has constant mean and variance over time.

KPSS (Kwiatkowski–Phillips–Schmidt–Shin) test:

Slightly less popular than ADF, but needed as a double check along with ADF test. The hypothesis for KPSS test is opposite of ADF test:

Null Hypothesis: The process in trend stationary (non-stationary)

Alternate Hypothesis: The series has a unit root (stationary)

If the p-value is greater than 0.05 then we can reject the null hypothesis i.e. the data is stationary, and the data has constant mean and variance over time.

STATIONARITY TEST ON THE ORIGINAL TIME-SERIES DATA		
METHOD	p-value	Decision
ADF	0.21 > 0.05	Non-stationary
KPSS	0.01 < 0.05	Non-stationary

Both tests confirmed non-stationarity of the original time-series stock data.

Transform Data for Stationarity

Mainly two types of transformation can be applied:

1. Differencing: defined as, y(t) = y(t) - y(t-n). when n=1 then we are taking differences between two days (given daily data). like wise, if n=30, then taking changes for a month. differencing method is especially useful when the mean of the data is time varying

2. Transformation: generally applying logarithm. root or power transform to make the time varying variance stationary. Log is used to dampen out highly varying data. whereas power is used when the variance decays down over time.

For our case, we will start with differencing method as from the seasonality we observed varying mean over time.

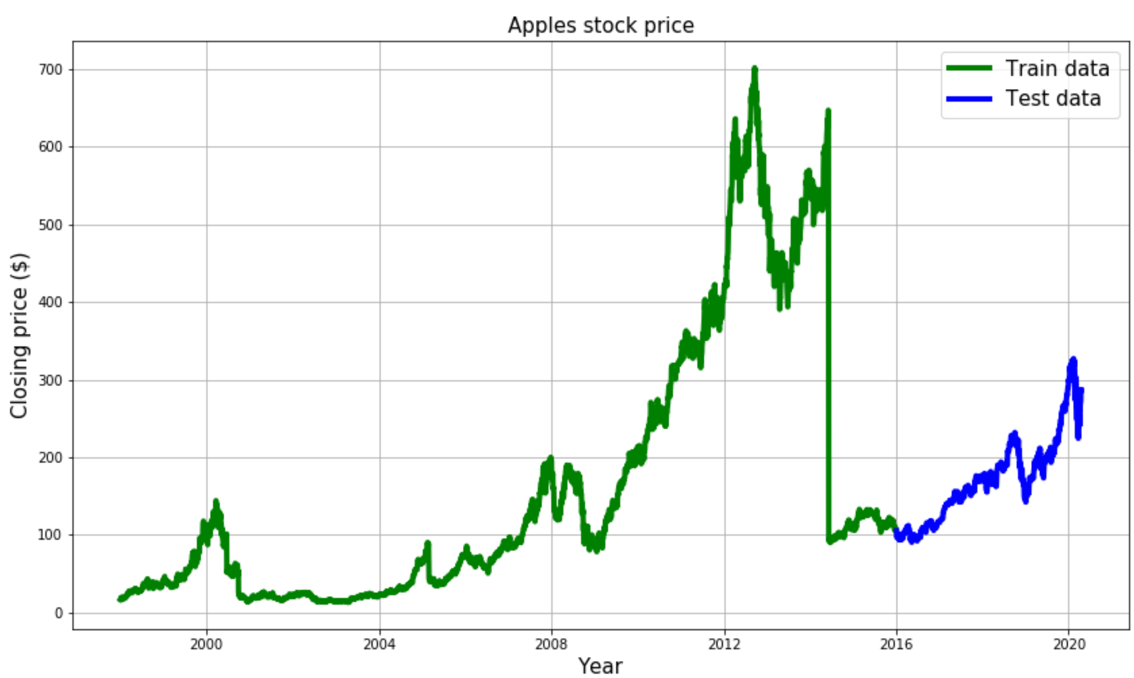
Time variation on the rolling mean and standard deviation have greatly reduced.

STATIONARITY TEST ON THE DIFFERENCED TIME-SERIES DATA		
METHOD	p-value	Decision
ADF	0.00 < 0.05	Stationary
KPSS	0.1 > 0.05	Stationary

Suggests the Apple daily price differenced data is stationary.

Train-Test Split

Like any other analysis time series forecasting also requires, test data which will be used to test the integrity of the prediction. We have almost 20 years of stock price data. The training data will be taken from the 1998 till end of 2015. The rest of the data (2016-2020) will be used at test data.



[Jupyter notebook for the pre-processing step can be found in this link.]

Machine Learning Modelling

Auto Regressive Integrated Moving Average (ARIMA)

ARIMA model will be used which is one of the most used tools for forecasting time-series data and stands for Auto Regressive Integrated Moving Average. Generally, ARIMA model can be expressed as:

predicted Y(t) = constant + linear combination lags of Y(t) (p lags) + linear combination lags of error in prediction terms (q lags)

For example, Y(t) = 5 + 3 Y(t-1) + 2 ϕ(t-1).

Here, p=q=1. d stands for differencing needed for Y(t) to make the stationary.

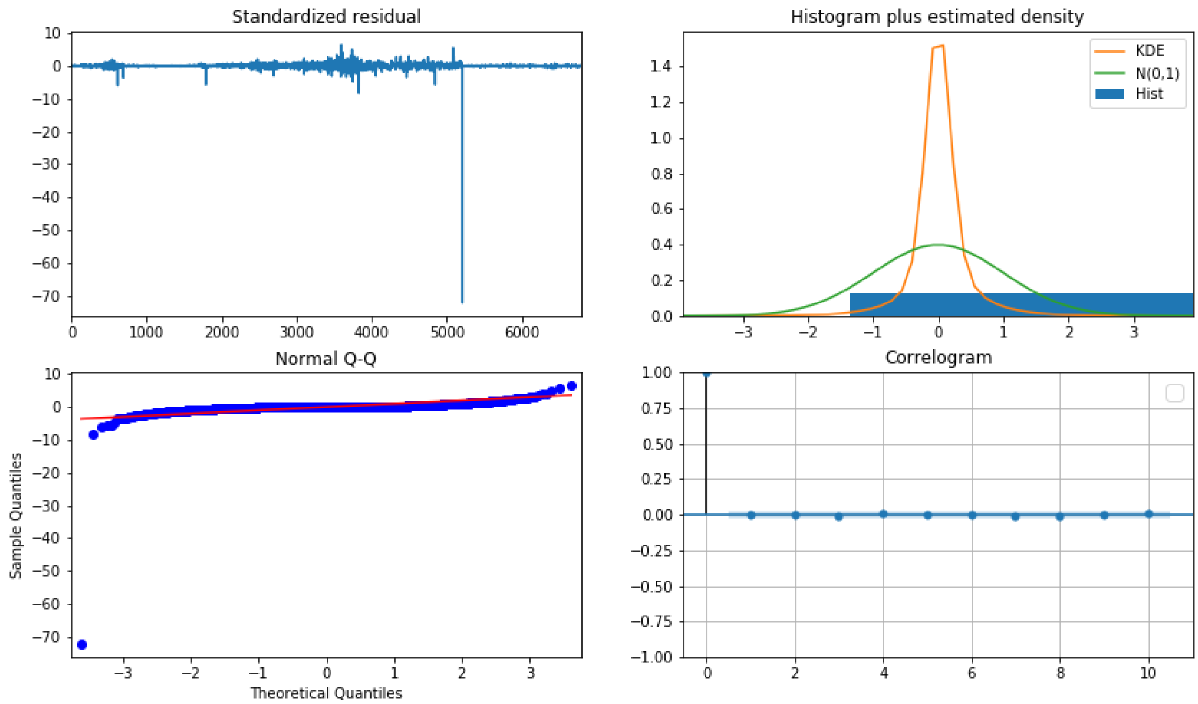
There are many ways to determine best possible combinations of p, d and q values for a ARIMA model given a time series data. One way would be manually plotting auto correlation function (ACF) and partial autocorrelation function (PACF) for combinations of p, d and q parameters and determine the case for near zero ACF and PACF values. Here we will use auto_arima function that will output best optimized model parameters for ranges of p, q values based on Akaike information criterion (AIC). AIC is an estimator which assess the statistical quality of a model. Model with a lower AIC value results in the best fit with the training data with least features.

The auto_arima function can do the grid search over p, d, q (related to ARIMA model) and P, D and Q (related to seasonal components) parameters and report back the model with best AIC value.

Finding Best Model Parameters

The auto_arima function was applied on the time-series stock data and best p, d, q parameters were search by grid search method. Best ARIMA model order was found to be (p, d, q) = (0,1,0), with AIC = 46910.

Residual statistics from auto ARIMA model needs to be reviewed for integrity of the model. Residuals should be closer to noise like statistics, which confirms not containing any information from the data



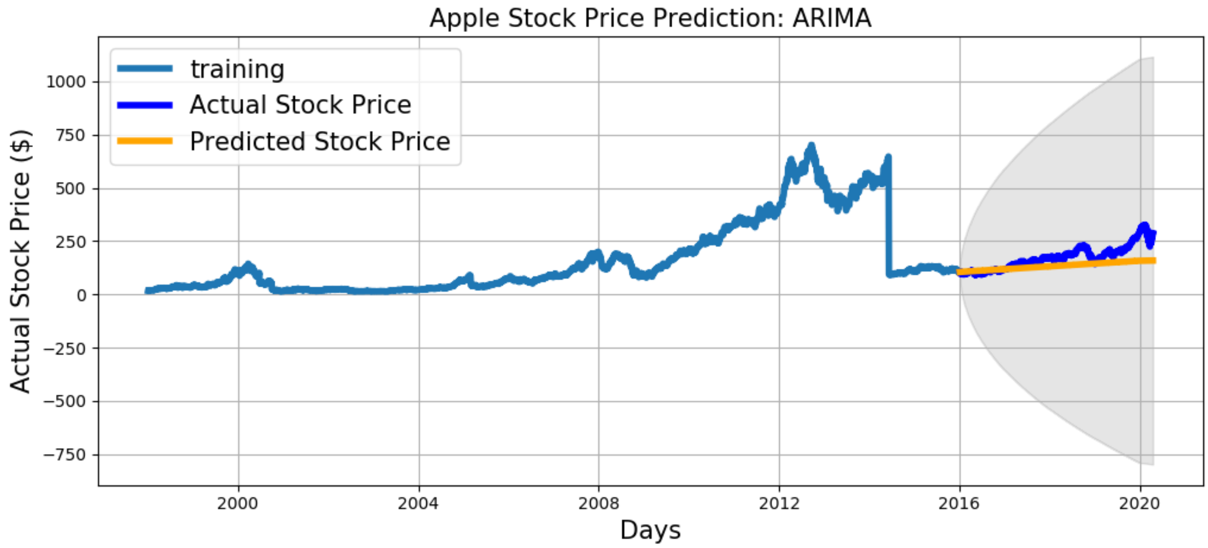
Couple of observations from the above residual plot:

- **Top left**: residual values seem to fluctuate around zero values and uniform variance except at the tail end. This is coming from the steep downing of the price after stock split in 2014
- **Top right**: The density plot suggests resemblances of residual distribution (orange color) with normal distribution (green one), with a mean zero
- **Bottom left**: The blue dots should perfectly align with the red line for a ideal scenario. This is the best possible alignment with the data
- **Bottom right**: The ACF of residuals are not correlated from the plot. Any correlation would suggest there are residual pattern in the data which is not explained in the model and needs parameter adjustment

Overall, it appears to be a good model and ready to use for forecasting.

Forecast with ARIMA Model

Now take the best ARIMA model to predict the stock price and compare with the test data set.

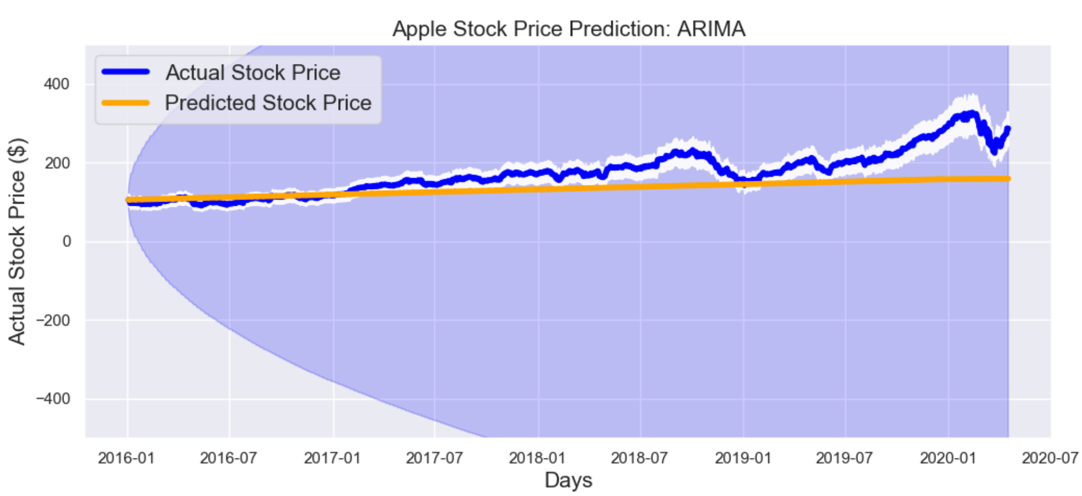


Observations:

- The predicted stock price (in orange) seemed to follow the test data (blue) well initially. After that it deviated from the observed test data
- The shaded grey region is the 95% confidence interval. Over time the confidence interval widens up, meaning the model loses its capability to confidently predict price over time

Look Closely into the Predicted Price

Prediction will be plotted only for 2016-2020 for better observations



Observations:

- The predicted price (orange) falls within the 15% price variation (white shaded region) over the 2016-2017
- Beyond 2017, predictions are in proximity with observed price and follow the trend but outside of 15% margin
- This suggests the built model can predict price well for a year
- Prediction line seems like a linear line, not following the randomness patterns of the stock price data

Performance Measure

Commonly used accuracy measures for time-series forecasting models are mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE). We will be using these metrics on the forecast and test data to measure model performances.

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
ARIMA	2288.39	36.92	47.83	0.19	593.33

- Lower the metrics better the model would be
- We will compare this performance result with FBProphet model in the next part of the project

FBProphet

Classic forecasting models such as ARIMA needs lots of parameter tuning and expert knowledge in statistics and analytics. Facebook developed an open source library called FBProphet, which requires truly little domain knowledge, easy to integrate in automated production environment. BBProphet decomposes any time series data into trend, seasonality, event, or holidays components and can be written as:

Y(t) = T(t) + S(t) + H(t) + ϵ

T(t): piecewise linear or logistic growth curve for modelling trend components **S(t):** cyclic changes in the time-series (dily/weekly/monthly/quarterly) **H(t):** effect of holidays or unscheduled events **ϵ**: noisy term that can not be modelled with equation.

As opposed to time-based dependence, FBProphet considers forecasting as curve fitting problem.

In this section, FBProphet will be used to model Apple stock price prediction. Modelling will start with a base model, then adding main three aspects of FBProphet, saturating growth, trend change and holiday effect.

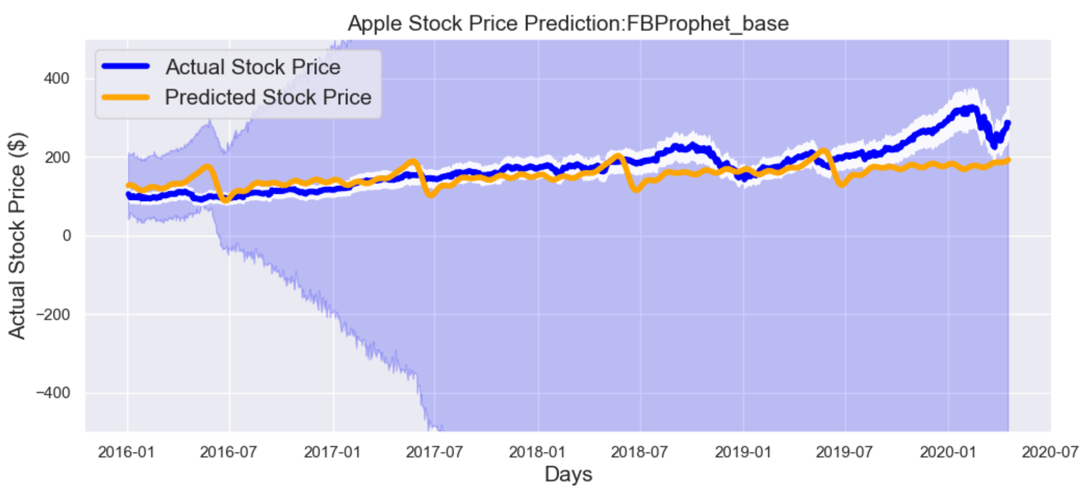
Base Model

Let’s build first FBProphet model with Apple stock price data with 95% confidence level. The base model included the weekly, yearly seasonality.

Plot the Forecast

The 'dark blue' line is the observed data. The 'light blue' line is the forecast data. The shaded region is the 95% confidence area.

Look Closely on the Test Data



The predicted price (orange) added randomness and following well with the randomness of the test data.

Model Performance Evaluation

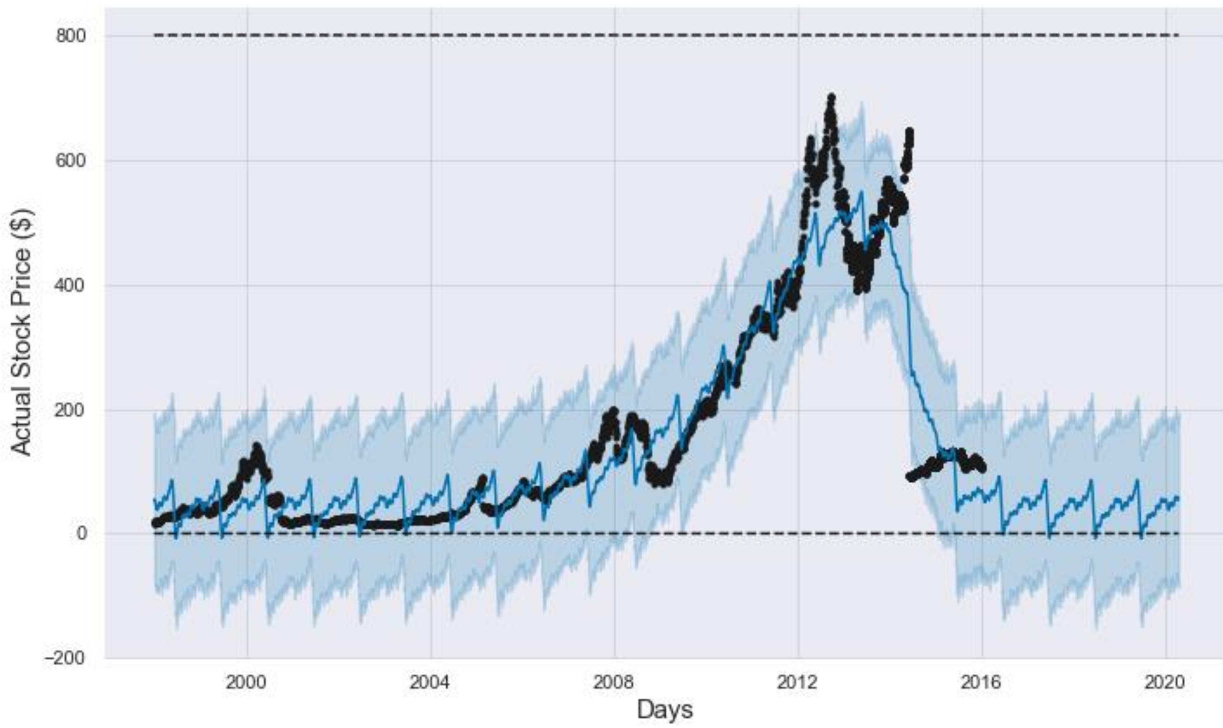
MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_BASE	2277.77	35.64	47.72	0.19	590.33

The average_score for FBProphet base model and ARIMA model are neck to neck.

Saturation Forecast

Sometimes prior knowledge of the maximum and minimum possible of forecast values can be useful, in keeping the curve fitting on track. Here, we will add maximum stock price of 800 USD, which is a safe assumption, given stock split occurs usually at 100 USD closing price. Minimum price will be set at 0 USD.

Plot the Forecasting



Looking into the predicted line (deep blue), frequent cycles appeared which does not follow much with the real data.

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_SATURATING	20332.25	129.26	142.59	0.71	5151.20

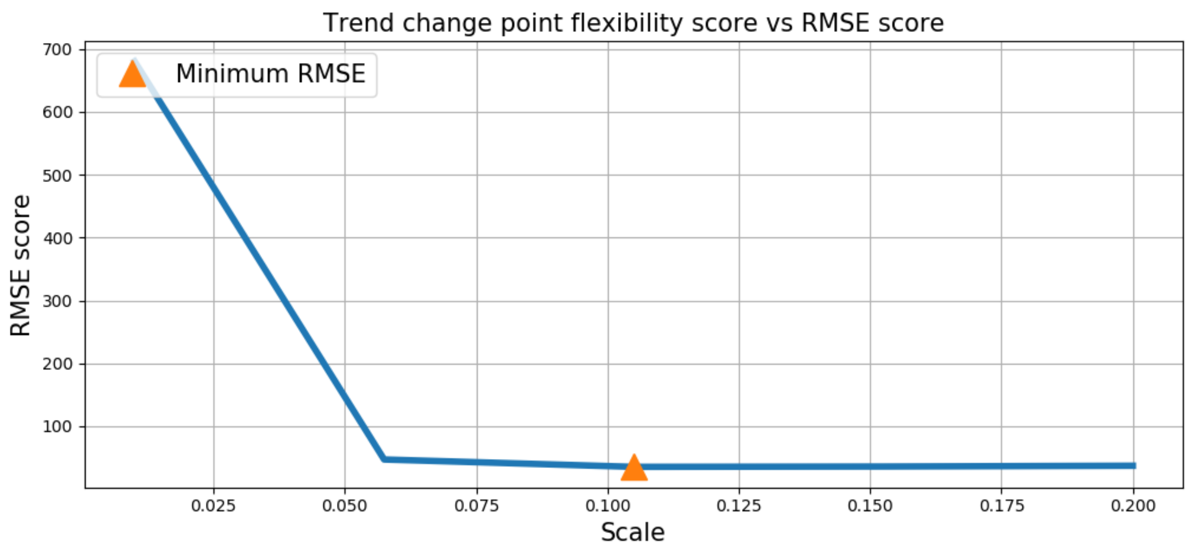
Considering saturating point, model performed worse than the base model which is reflected on the average_score.

Trend Changepoints

FBProphet automatically detects trend change points in the time-series. However, there are options for finer control if the expected trend change points are not captured with automatic control. It will be interesting to see, if FBProphet can capture major economic downturns happened in the past decades which might have impact on the forecasting.

Modelling

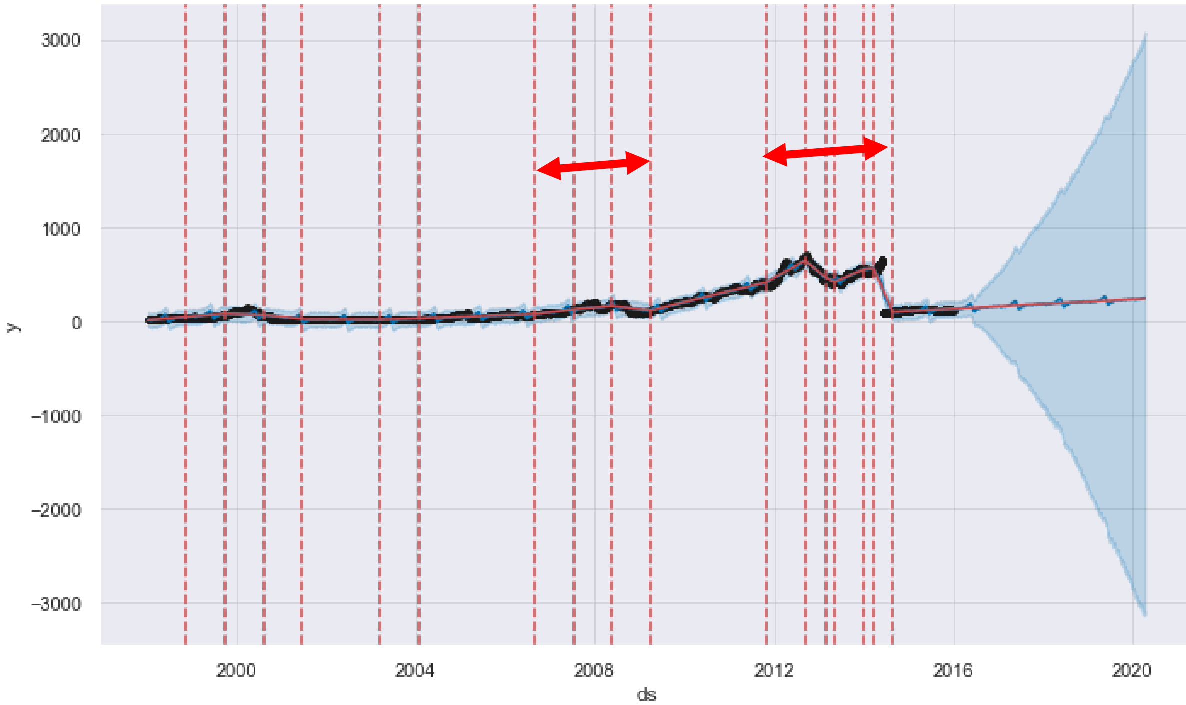
We figured out an important parameter for the prophet model is changepoint_prior_scale, which determines the trend flexibility in the model (default value is 0.05). Higher value leads to overfit (more flexible) and lower value leads to less flexible model. This best scale parameter was found measuring the lowest RMSE point over 0.01 and .02 ranges.



Best scale for minimum RMSE score is 0.105. With this, trend change model will be trained and performance measured.

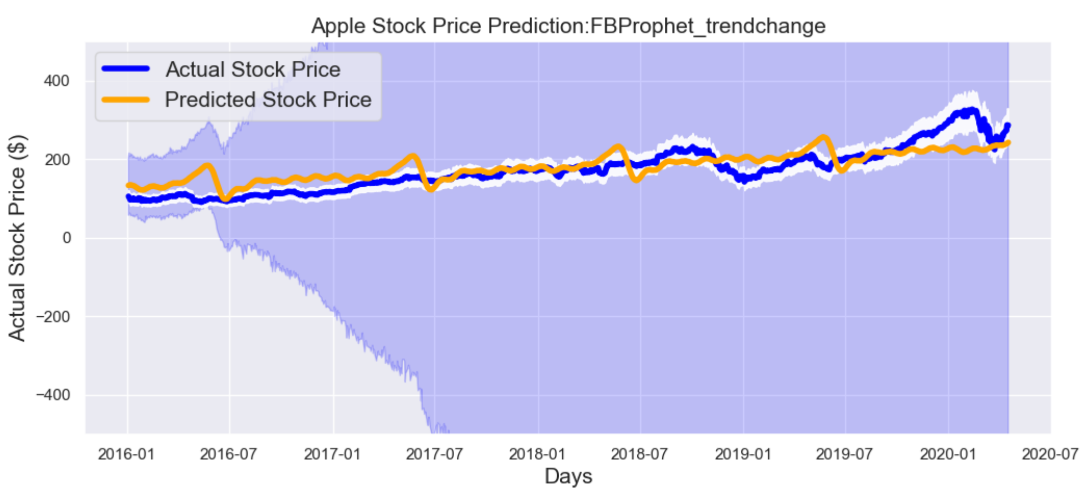
Plot the Forecasting

In FBProphet model, the time stamp column is denoted as ‘ds’ and the value column name is denoted by ‘y’.



- Seeing the changepoints, the model is considering the trend changes during 2008 recession (middle stripped red lines)
- It is also including high number of trend changes visibly seen after 2012
- FBProphet automatically tracks trend changes and it is doing a good job for this scenario considering 2008 recession and 2014 sharp stock split. Trend change points can be manually set as well.

Look Closely on the Test Data



Tuning up with trend change options, alignment between the observed (dark blue) and predicted (orange) data visibly improved a lot and persisted for most of the duration.

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_TRENDCHANGE	1241.24	28.52	35.23	0.18	326.29

Considering trend change points yielded much better result than any other models and reflected with lowest average_score so far.

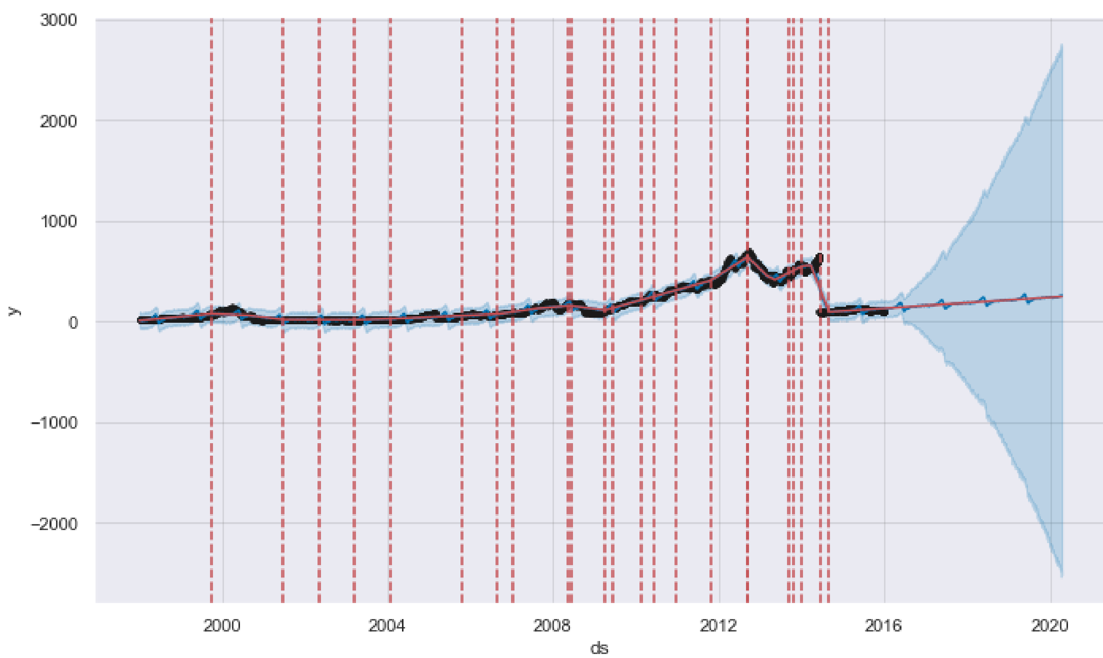
Trend Change Model with iPhone Release Day

FBProphets trend change model automatically finds the best trend change points from the data. In this section, in addition to the best trend change points found in the previous section, Apple's iPhone release days will be added as trend change points to see the effect on the modelling results.

Modelling

Trend change model parameters were taken from the previous section. In addition, iPhone release day for the duration of 1998-2015 were added as trend change point. The iPhone release days were taken from (https://www.whistleout.com/CellPhones/Guides/iphone-release-dates).2016 data will be excluded to add flexibility to the algorithm not to overfit.

Plot the Forecasting



Increased trend points observed (red dotted lines) in the time-series data

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_TRENDCHANGE_RELEASEDAY	1314.17	30.19	36.25	0.19	345.20

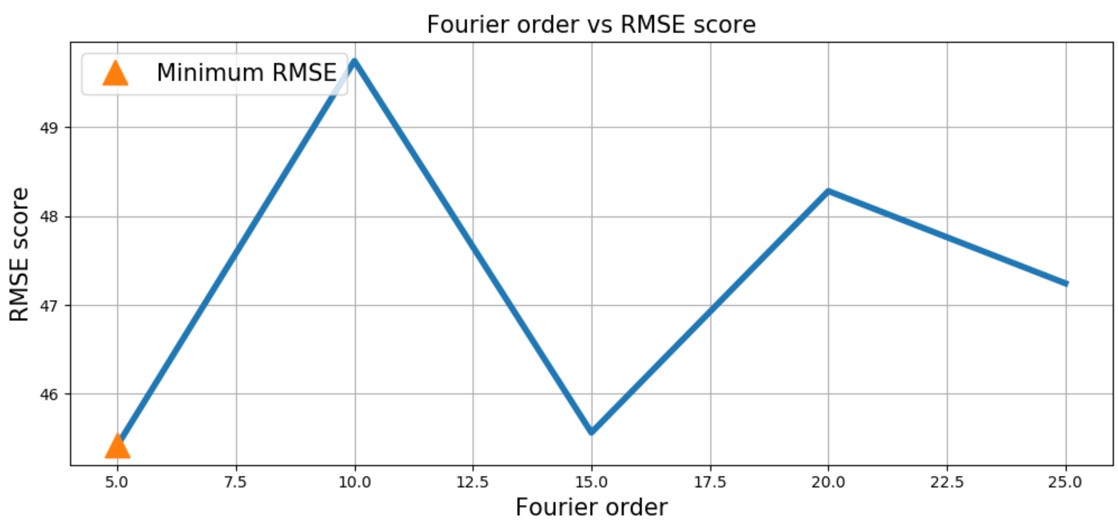
Added trend change points over fit the model hence reduced performance than the automatic trend detection model showed earlier.

Seasonality

By default, FBProphet will fit weekly and yearly seasonality and daily if the data is at least two cylces long. Other seasonality's (monthly, quarterly) can be added manually. From the EDA of Apples stock price it was seen that, there are monthly and yearly stock price trends were seen. This will be interesting to note in the Seasonality analysis.

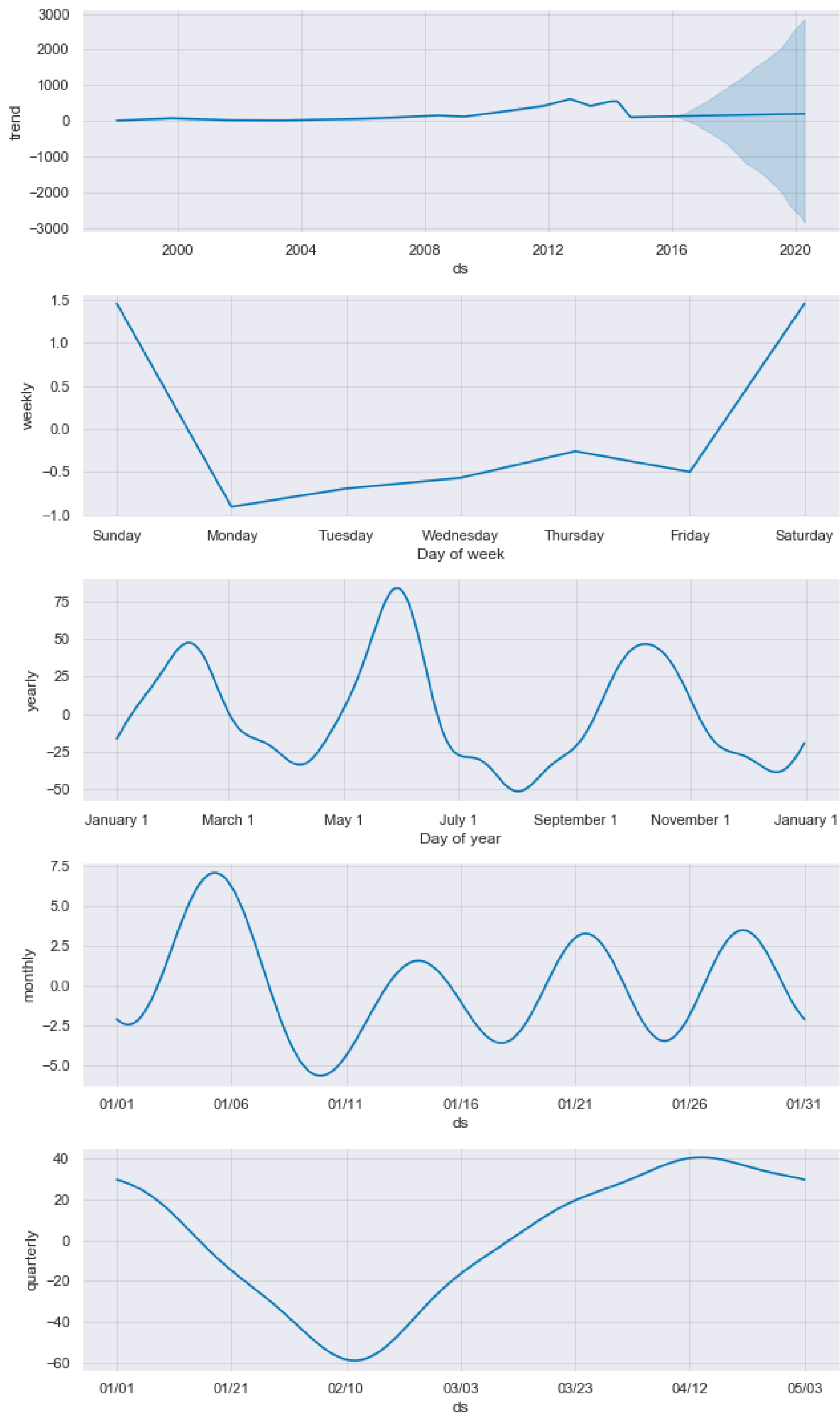
Modelling

Every seasonality's can be represented by collection of frequency components (known as fourier_order), which is important to determine for time-series modelling with seasonality. Here, we will run ranges of fourier_order to find best possible RMSE values. For simplicity we will consider the fourier_order for the month and quarter are the same.



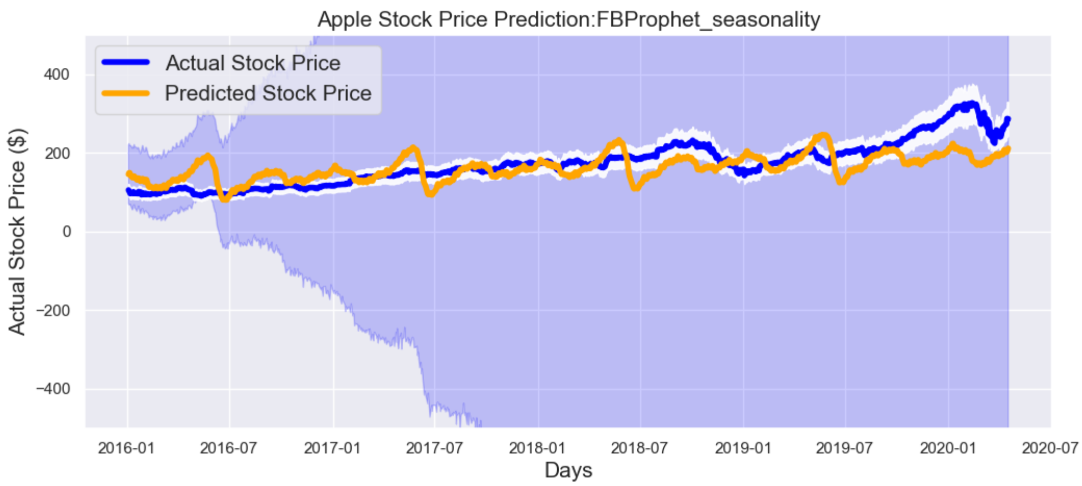
Best order for minimum RMSE score was found to be 5. With this the seasonality model will be trained and tested.

Plot the Forecasting



Monthly, weekly, daywise and quarterly break down of the stock price from the modelling provide insights about the trend in the price.

Look Closely on the Test Data



Adding seasonality included more randomness in the prediction and alignment with the test data.

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_SEASONALITY	2063.07	35.23	45.42	0.20	535.98

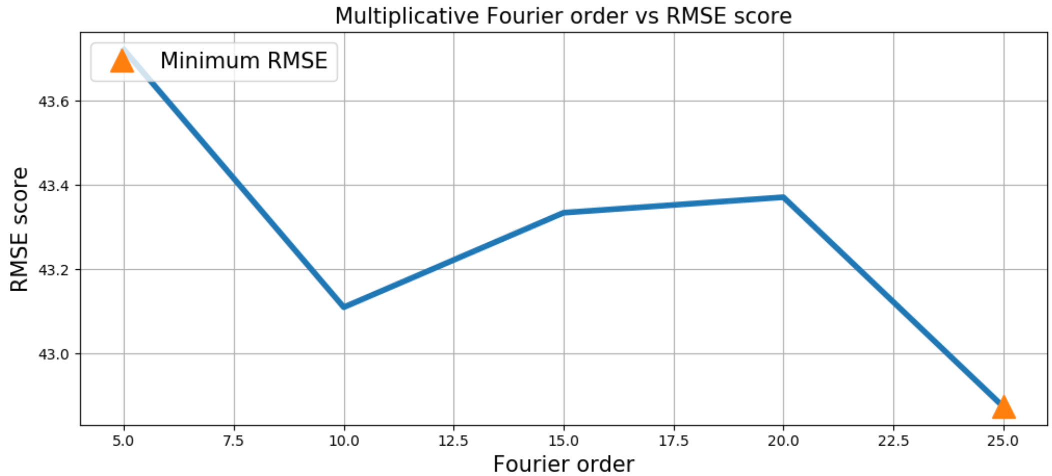
Adding seasonality in the model improved the prediction score by more than 55 margins than the base model.

Multiplicative Seasonality

By default, Phophet models forecast with seasonality and trend by additive method. It may not be always the case. Visually it is hard to tell from the time-series plot whether an additive or multiplicative model would be good fit. In this section we will explore how multiplicative seasonality perform over additive model.

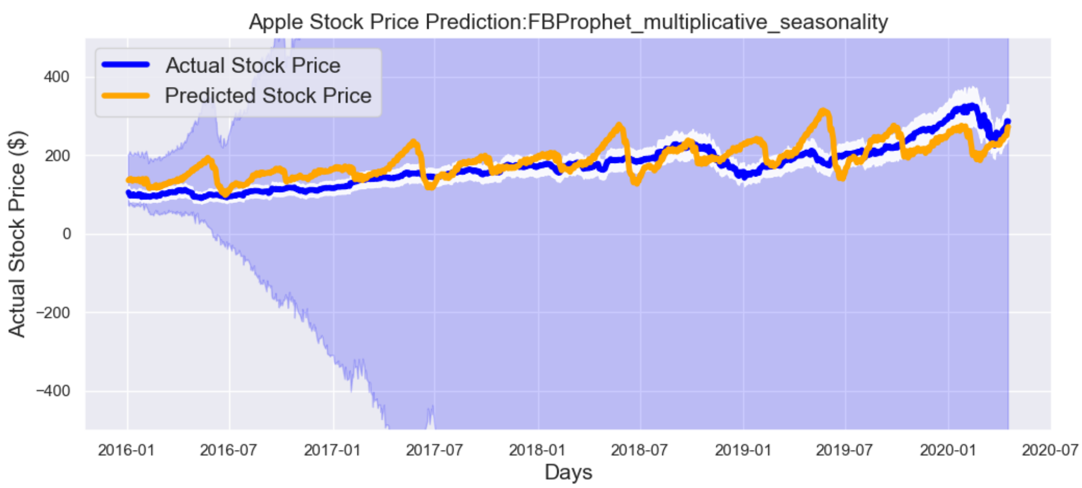
Modelling

Like previous section, we will figure out the optimum Fourier components for multiplicative model.



Best order for minimum RMSE score was found to be 25.

Look Closely on the Test Data



Although there are series of spikes seen over the forecast line, the trend matches with the observation till the tail end.

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_MULTIPLE_SEASONALITIES	1838.21	34.55	42.87	0.22	478.96

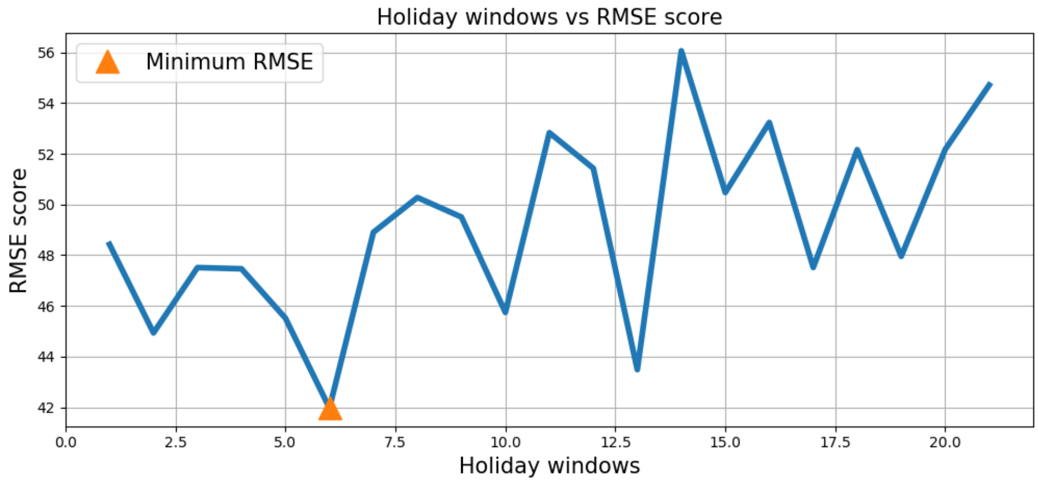
Multiplicative model improved over additive model by a margin of 57 points in terms of average_score.

Holiday Effects

Holidays can have drastic effect on the stock price, trend. For example, stock seen historically slack time during Christmas, because people are busy buying in the shopping market other than the stock. FBProphet can include list of holidays over the train and test duration. The model pays special attention to these points and learns from the changing trend accordingly.

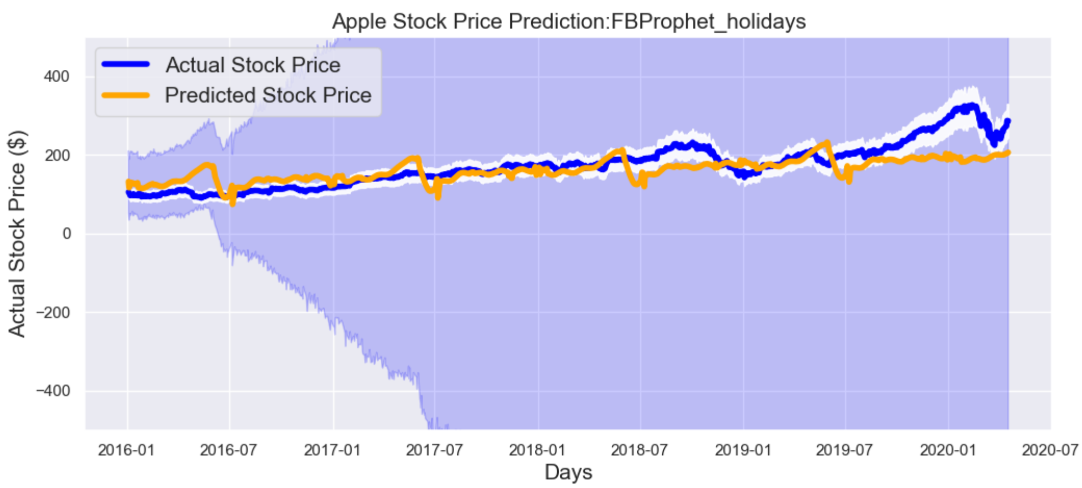
Modelling

in this section, we ran a function to determine the optimal effective days for which the RMSE error will be minimum. It would be best to get individual effective days for every types of holidays. For simplicity of analysis we will find a single effective day for all holidays.



This shows, on average after 6 days of holidays the stock prediction gets its best

Look Closely on the Test Data



Prediction follows the test data over most of the duration.

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_HOLIDAYS	1761.25	31.33	41.96	0.17	458.68

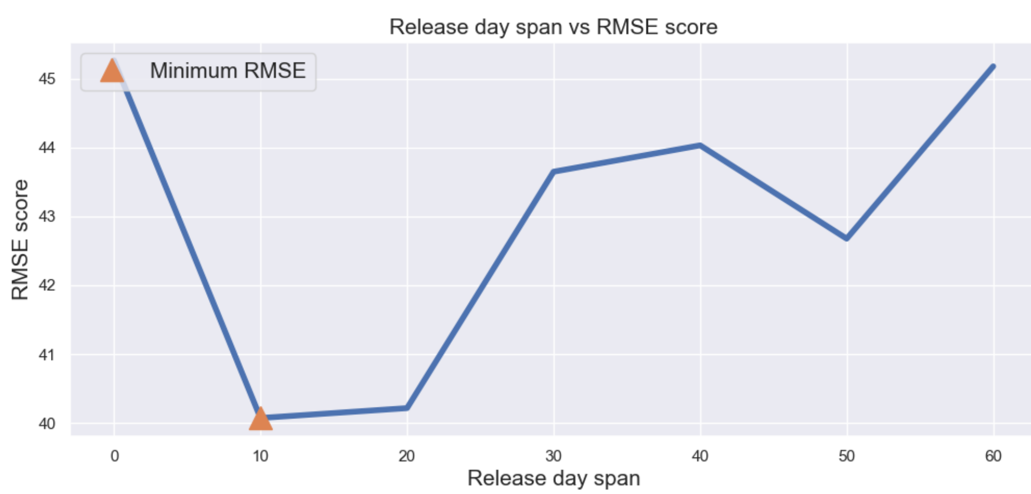
Adding holidays improved the average_score by a margin of 132 than the base model.

Product Release Event

Product relesae date can have significant impact on the stock price. This article (https://www.marketwatch.com/story/heres-how-apples-stock-performs-around-iphone-launch-events-2018-09-12) mentioned on an average 60 days before and after the iPhone release date when, noticeable price gain was marked. Apple also releases other products in the market which can have impact on the stock price. In this section, a prediction model will be developed based on the historical product release dates. First, optimum reaction days will be determined based on the ranges of values that yields lowest RMSE error. iPhone is one of the Apples most popular products. iPhone release dates will be added separately.

Modelling

Best reaction days will be searched over the range between 0 to 60 days (according to the above weblink). Pre and post release days will be equally spanned. Product release dates will be modelled as special events. Product release days will be collected from this site (https://en.wikipedia.org/wiki/Timeline_of_Apple_Inc._products). iPhone release dates will be added from this site (https://www.whistleout.com/CellPhones/Guides/iphone-release-dates). There are certain windows of days around the product release days where stock price sees the biggest effect. Over a span of 0 to 60 days we will look for day span where the RMSE score is the lowest.



On average 10 days span was felt on the stock price for any holiday, product, or iPhone release days.

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_HOLIDAYS_RELEASE_EVENT	1605.75	30.88	40.07	0.19	419.22

Considering product/iPhone release dates along with hoidays improved average score by 39 unit.

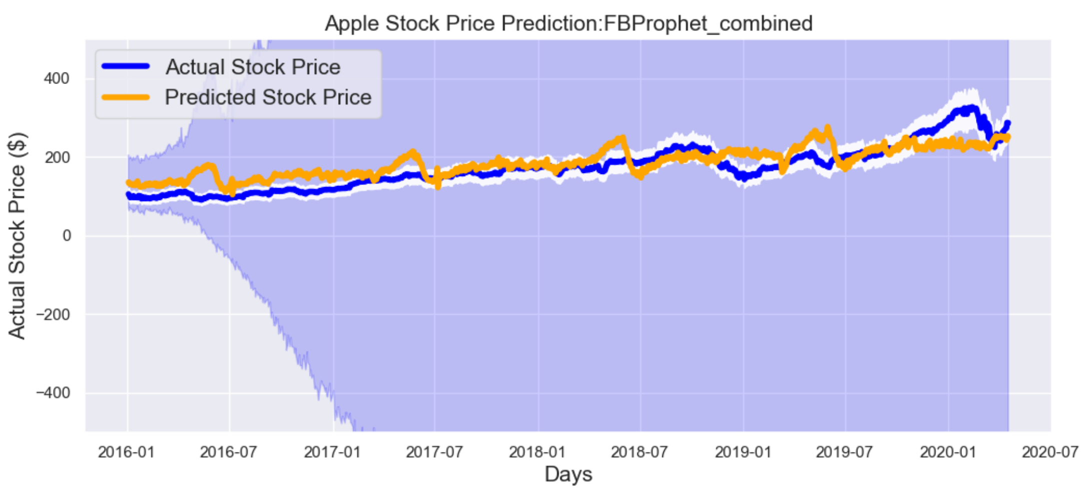
Combined Model

Improvements were seen in terms of score and trend alignment when trend change option was tweaked, seasonality and holidays added in the models. Here we will come up with a single model with the combinations all these effects.

Modelling

Best parameters determined earlier for trend change scale, multiplicative seasonality Fourier order and holidays effect will be combined for this model. It was found that removing quarterly seasonality yield best possible model performances.

Look Closely on the Test Data



Over the test data duration, predicted price showed better alignments with the observed price data.

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_COMBINED	1450.45	31.41	38.08	0.20	380.04

Although the average_score is not the best from the FBProphet models, this model includes important factors such as trend chage points, seasonalities (monthly, yearly) and holidays effect.

Residual Modelling

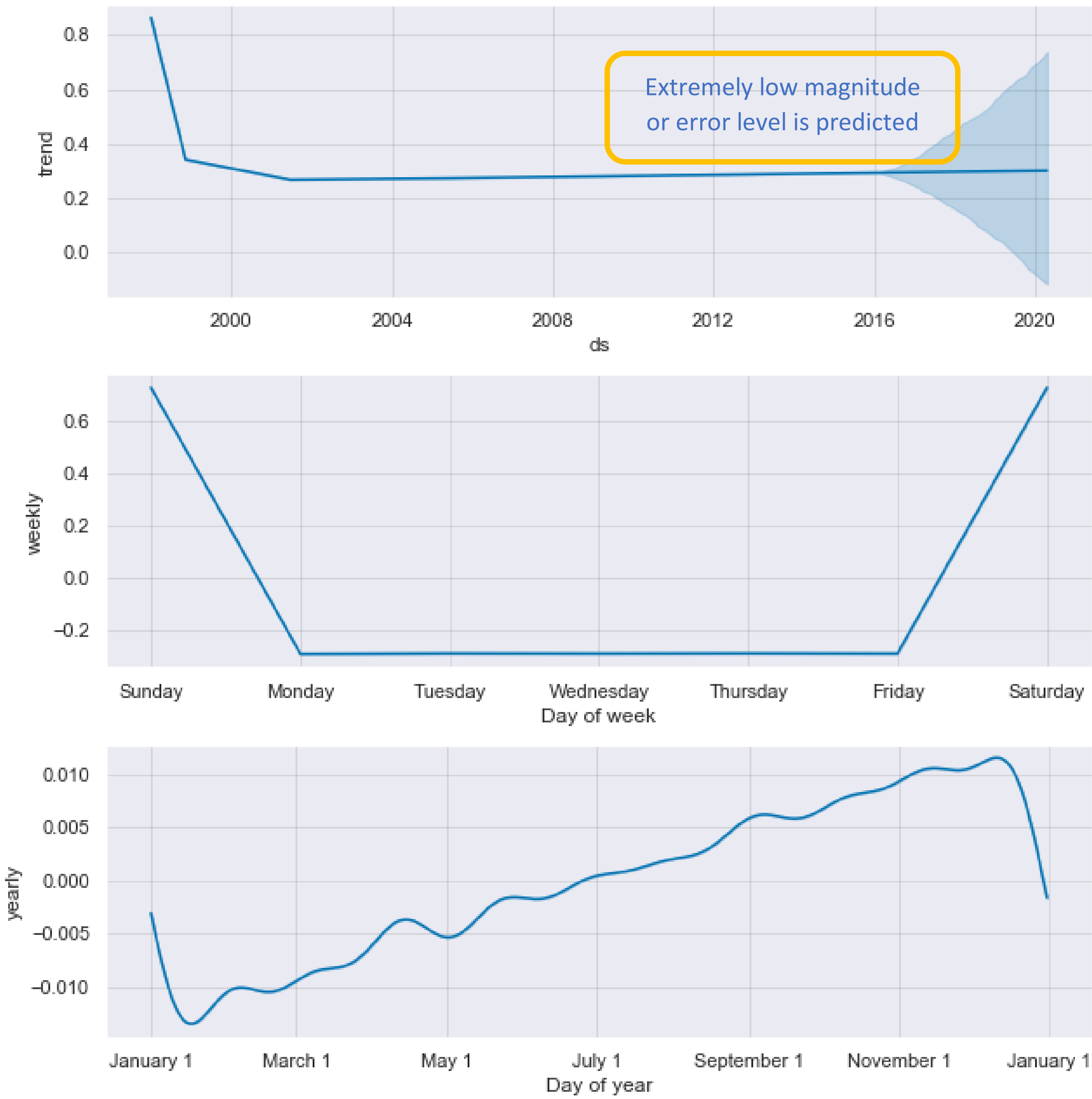
The difference between observed and predicted model is called residue of a model. Generally, the residuals are considered white noise (if modelled properly) and can not be modelled by any mathematical trend or seasonality components. There are many ways to model residual error with the main model. Here, we will utilise the 'add regressor' option of the FBProphet. The 'add regressor' puts extra weights while building the model. The weights are proportional to the regressor values added. As a regressor value we will add the residuals in our model, that is the difference between the observed values and the predicted values.

Predict Residual based on the Train Residual

Residual calculated from the train_data will be used as a train data to predict error for the test_data duration. This error prediction will be used as a test regressor along with the train regressor for modelling the final model.

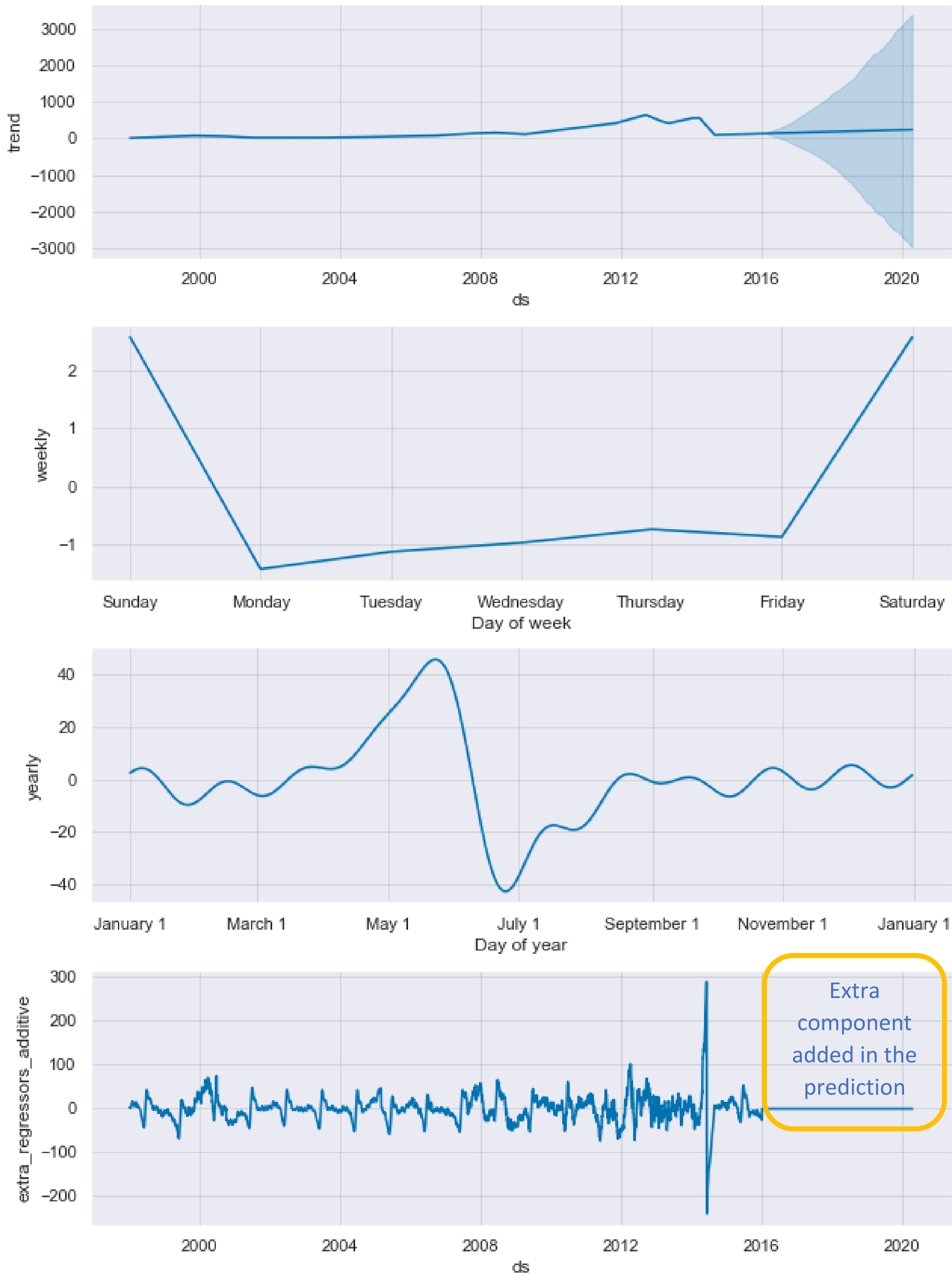
So far, the 'trend change' model performed best in terms of average_score. We will take the residuals from the observed data and the forecast data from that model.

Plot the Forecast Residual



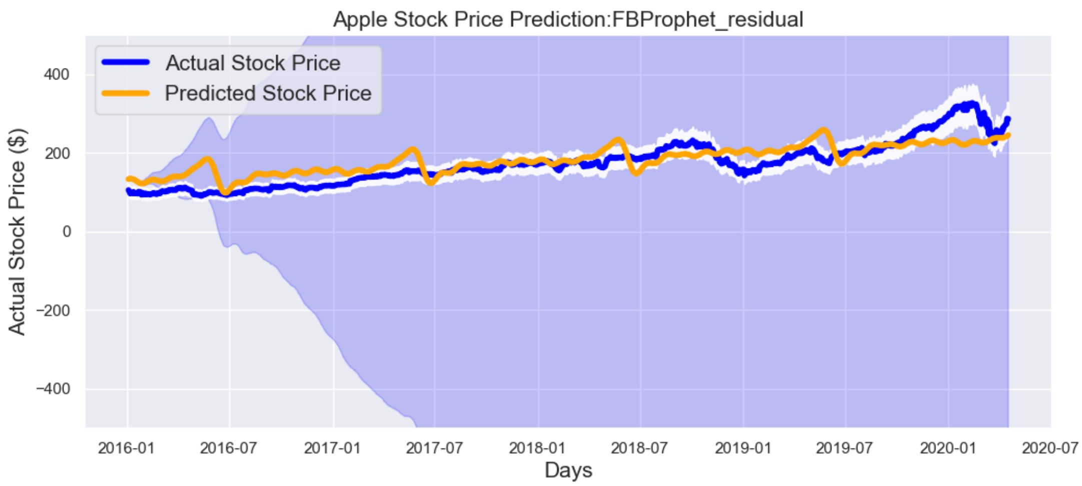
The model predicted exceptionally low magnitude of errors which will be appended with the train error and included in the modelling as ‘add regressor’ which adds effect on the linear part of the prediction model.

Plot the Forecasting



We can see from the last plot of above figure, the linear component added in the prediction from error. Next section will evaluate the model performance.

Look Closely on the Test Data



Visually the predicted values align well with the observed data.

Model Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
FBPROPHET_RESIDUAL	1236.84	28.47	35.16	0.18	325.16

The average_score is the lowest of all the models discussed so far.

Modelling with XGboost

XGBoost is a type of decision tree which is popularly used for regression problem. In time series prediction, only the value to predict is required along with time stamp. To frame a time-series probem into a regression problem, new features needs to be added (feature variables) and the predictor variable acts as target variable. In this sction, at first, feature engineering will be done to convert time-series to regression problem. Next thing will be to train the model and evaluate the result with XGboost algorithm. Feature importance will be extracted to know which variable is contributing dominantly for the predictor variable. Finally, prediction will be generated and evaluated across observed data.

Feature Creation

Two types of features will be generated. One, effect of the time of the year in the form of from week, month, quarter, day of year etc. Two, effect of holidays, product release and iPhone release dates as events feature.

Features for Train and Test Data

In the train data, along with the price information the holidays, product and iPhone release dates information were added as new columns marking corresponding day by the value of 1. In addition, time features were also added. Below is a glimpse of the train data features. The predictor variable is the closing price.

```
X_train
```

	dayofweek	quarter	month	dayofyear	dayofmonth	weekofyear	holiday	product_release_day	iPhone_release_day
0	4	1	1	2	2	1	0.0	0.0	0.0
1	0	1	1	5	5	2	0.0	0.0	0.0
2	1	1	1	6	6	2	0.0	0.0	0.0
3	2	1	1	7	7	2	0.0	0.0	0.0
4	3	1	1	8	8	2	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
4524	3	4	12	358	24	52	0.0	0.0	0.0
4525	0	4	12	362	28	53	0.0	0.0	0.0
4526	1	4	12	363	29	53	0.0	0.0	0.0
4527	2	4	12	364	30	53	0.0	0.0	0.0
4528	3	4	12	365	31	53	0.0	0.0	0.0

Hyper-Parameter Tuning with RandomizedSearchCV

Hyper parameters for the XGBoost model will be determined by RandomSearchCV method. The ranges of parameters were set from this site (https://medium.com/genesis-media/time-series-forecasting-number-of-sessions-on-web-site-c36c85ebdbc). For the evaluation metric, RMSE score was used.

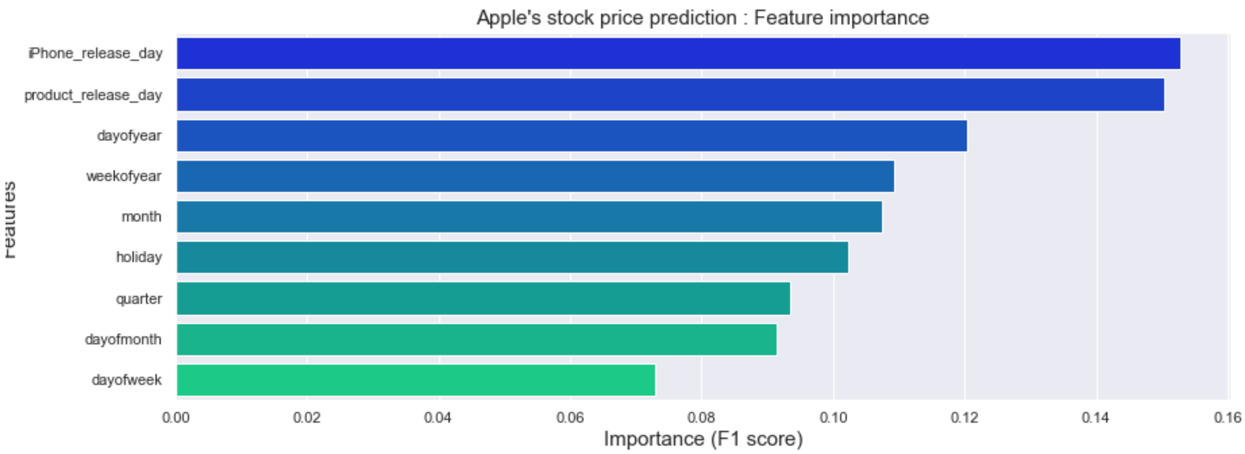
XGBOOST BEST MODEL PARAMETERS

PARAMETERS	Value
SUBSAMPLE	0.6
NUM_LEAVES	50
MIN_CHILD_WEIGHT	2
MAX_DEPTH	5
LEARNING_RATE	0.01
GAMMA	1.5
COLSAMPLE_BUTREE	1
N_ESTIMATORS	50

With these parameters the XGBoost model was trained with the price data and feature importance and prediction were done in the next sections.

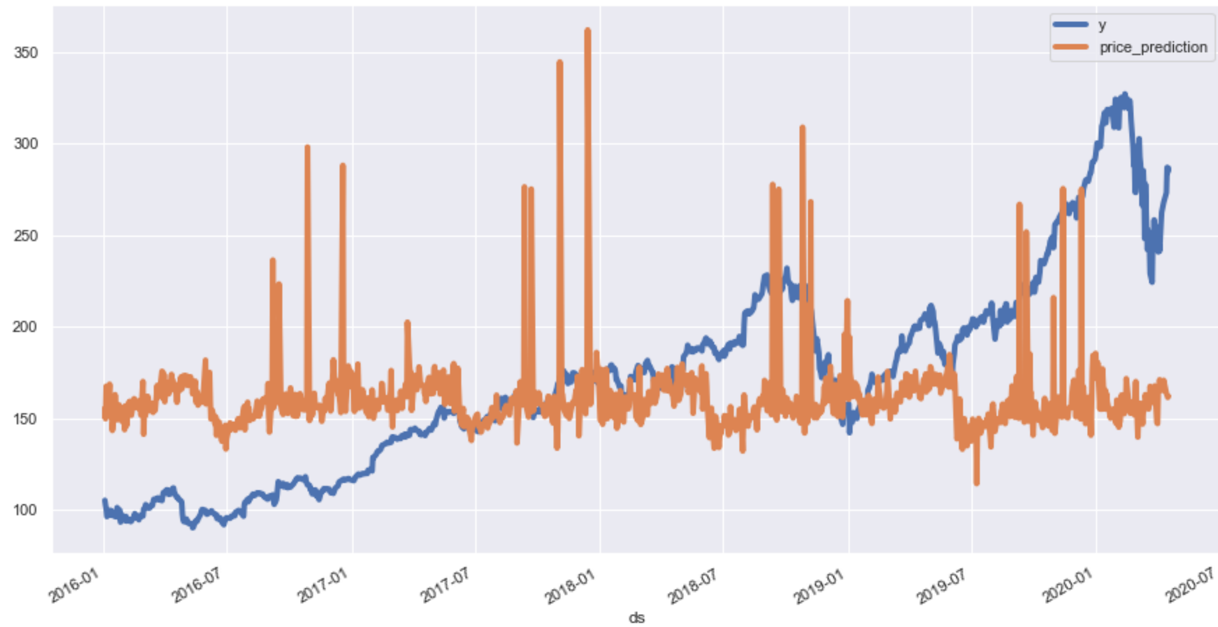
Feature Importance

Feature importance was extracted and sorted by descending values. The importance score was measured by F1 score (https://en.wikipedia.org/wiki/F1_score).



- iPhone release dates impact the stock price prediction most followed by other product release dates. Holidays have medium strength on the prediction.
- From the time of year features, day of year is significant for stock price prediction.

Model Prediction



Clearly, the predicted values do not fit well with the observed results.

Performance Evaluation

MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
XGBOOST	3511.63	45.55	59.25	0.29	904.18

Average score is highest among the models described so far.

Model Evaluations

Now score data from all the models are available in terms of MSE, MAE, RMSE, MAPE and average_score. We will display and compare results from the various modelling.

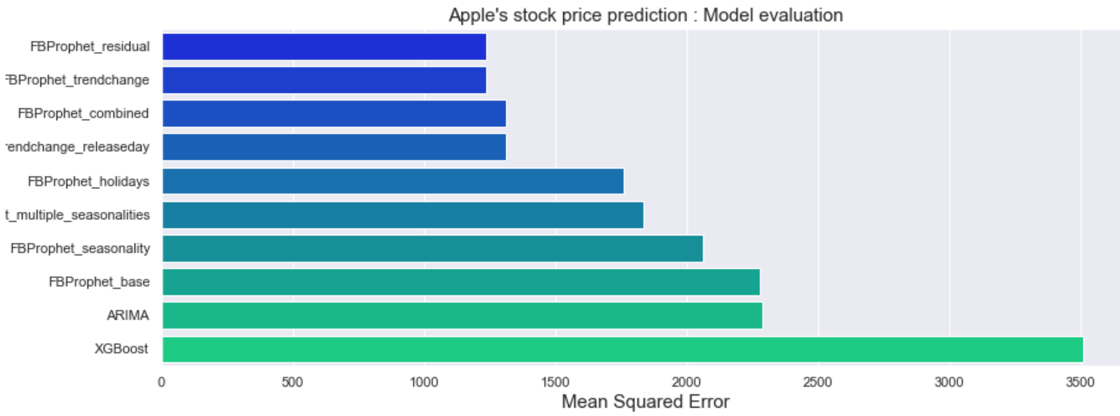
Tabular Chart

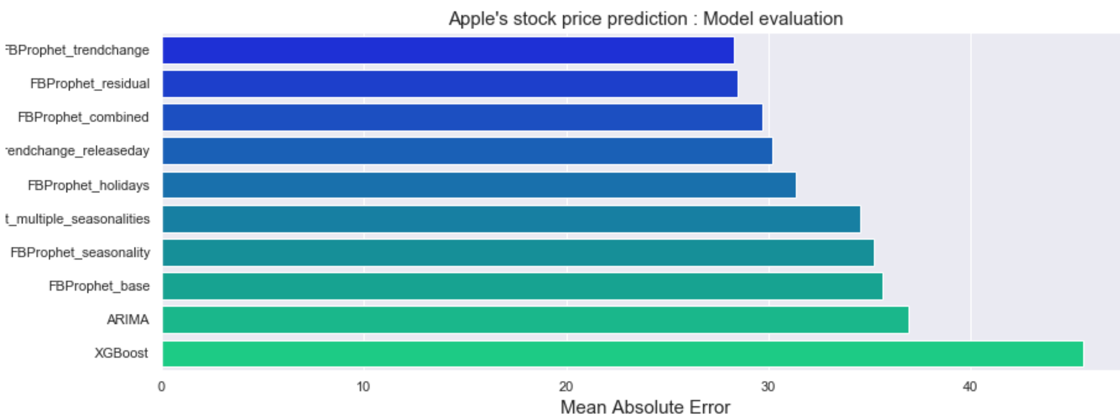
Let us tabulate the performance score for the models we have discussed so far.

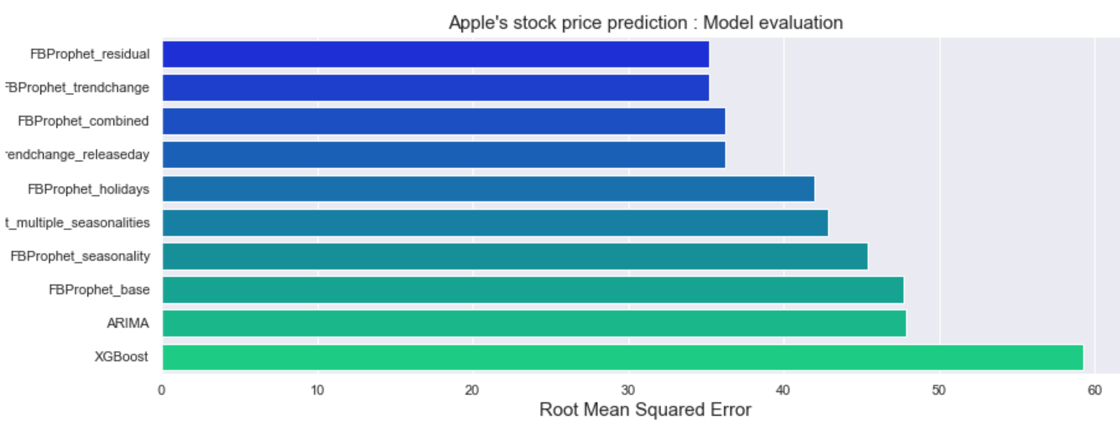
MODEL	MSE	MAE	RMSE	MAPE	AVERAGE_SCORE
ARIMA	2288.39	36.92	47.83	0.24	593.350712
FBPROPHET_BASE	2277.77	35.64	47.72	0.22	590.34
FBPROPHET_TRENDCHANGE	1237.16	28.29	35.17	0.18	325.20
FBPROPHET_TRENDCHANGE_RELEASEDAY	1314.17	30.19	36.25	0.19	345.20
FBPROPHET_SEASONALITY	2063.07	35.23	45.42	0.22	535.98
FBPROPHET_MULTIPLE_SEASONALITIES	1838.21	34.55	42.87	0.22	478.96
FBPROPHET_HOLIDAYS	1761.25	31.33	41.96	0.20	458.68
FBPROPHET_COMBINED	1312.28	29.72	36.22	0.19	344.60
FBPROPHET_RESIDUAL	1236.84	28.47	35.16	0.18	325.16
XGBOOST	3511.63	45.55	59.25	0.29	904.18

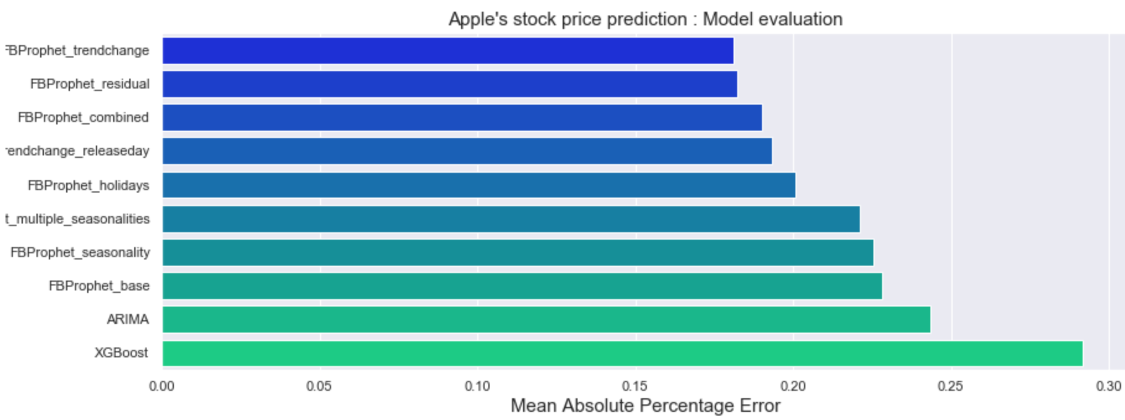
We can see that, the FBProphet model will residual modelling yield best possible scores.

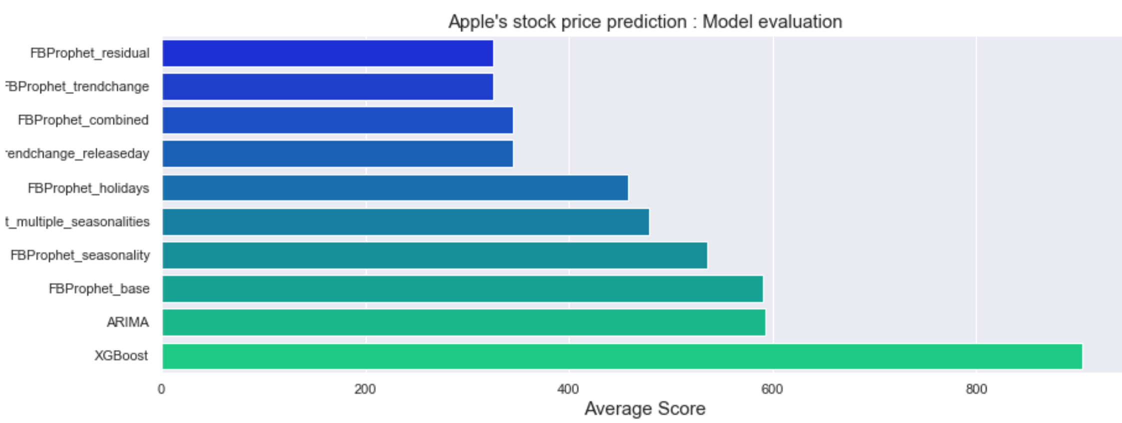
Bar Plot


Apple's stock price prediction : Model evaluation
FBProphet_residual
FBProphet_trendchange
FBProphet_combined
endchange_releaseday
FBProphet_holidays
t_multiple_seasonalities
FBProphet_seasonality
FBProphet_base
ARIMA
XGBoost
0
500
1000
1500
2000
2500
3000
3500
Mean Squared Error


Apple's stock price prediction : Model evaluation
FBProphet_trendchange
FBProphet_residual
FBProphet_combined
endchange_releaseday
FBProphet_holidays
t_multiple_seasonalities
FBProphet_seasonality
FBProphet_base
ARIMA
XGBoost
0
10
20
30
40
Mean Absolute Error


Apple's stock price prediction : Model evaluation
FBProphet_residual
FBProphet_trendchange
FBProphet_combined
endchange_releaseday
FBProphet_holidays
t_multiple_seasonalities
FBProphet_seasonality
FBProphet_base
ARIMA
XGBoost
0
10
20
30
40
50
60
Root Mean Squared Error

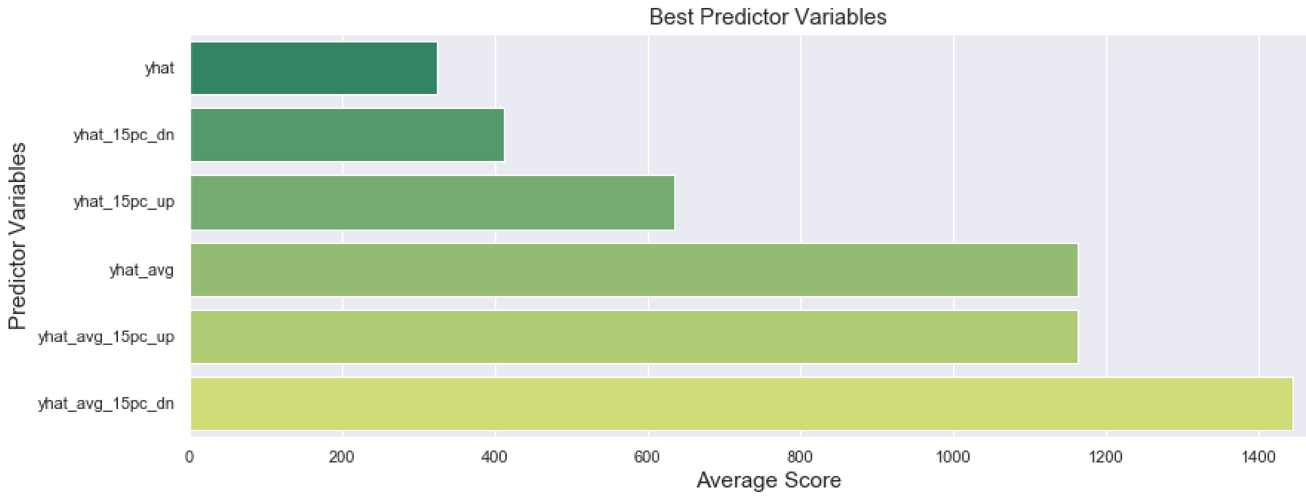




In every measure FBprophet with residual effect tops the chart.

Best Forecast Variable

Sometimes, the predicted variable (yhat) may not always be the best version of the prediction. It would be good to see if any combinations of the yhat variables can yield best prediction. Here, in addition to the yhat cople of other variables will be tried. 15% upper and lower of yhat, yhat average taking the mean of yhat and its upper and lower valus, and the 15% upper and lower of the yhat average.



We can conclude that or this work, yhat would be the best predictor variable.

Conclusion

Apple's stock price forecasting was modelled with historical stock price data from 1998-2020

Three models were tried:

- ARIMA
- FBPhophet
- XGBoost

Training data was split up to 2016 and the rest of the data was used for testing

Four scoring measures were used: mean square error (MSE), mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAEP). 'Average_score' measure was introduced which takes average of all the metrics and provides a single number to represent the goodness of a model

ARIMA

Hyperparameters of ARIMA model were optimized using grid search method. The best parameters were found to be (p, d, q = 0 , 1, 0)

Average_score for the best ARIMA model is 593

FBProphet

Following model parameters were tuned:

Base model: With default parameters yielded average score of 590

Saturating: Bound by the minimum and maximum predictable values which did not show promising result

Trend Change: Automatically detects trend changes and considers in modelling. This one produced a score of 325.2

Trend change with iPhone release dates: Includes iPhone release dates in the trend but seemed to overfit the model

Seasonality vs multiplicative seasonality: Adds custom seasonalities (weekly, quarterly, monthly, yearly) in the model. Multiplicative seasonality modelling produced better result than with only seasonality

Holidays and special events: Adding US holidays and special events such as iPhone release dates and other product release dates produced average score of 458

Combining all effects: Combining trend change, seasonality, holidays, and special events yielded score of 344.6

Trend change with Error Modelling: Trend change hit the best score and to further improve, error was modelled and added with the forecast which produced the lowest score of 325.16

Couple of observations from the seasonality trend:

day wise: Stock price seen gradual increase from Monday to Friday with Thursday hit the maximum price

yearly: Gradual increase of stock price forecast from January to June. July hit the lowest price of the year and then gradual increase till December

XGBoost

Feature Importance: iPhone release days along with the product release days are important in predicting the price. Which is also seen in real too. When there is a new iPhone release market goes up in the specific time window.

Forecast score found to be 904

Which model is the best here?

FBProhet model considering trend change and residual error produced the best RMSE score of 35.16 which is an improvement from the ARIMA by 26.5% and XGBoost by 40.6%.

Future Directions

- Grid search technique can be deployed to know the best possible hyperparameters for FBProphet which is computationally expensive. This can be accomplished over cloud computation.
- App deployment can be done to predict for next couple of days or months stock data with confidence level.