Cryptocurrency Forecast 2021

Tareq Aghabi

2021-01-08

Introduction

Since the start of 2020, the two main Crypto giants Bitcoin and Ethereum have been gaining steady momentum in the Crypto market, and now at the start of 2021, are just beggining to peak and surpass all-time highs. This surge in price and market cap of both Bitcoin and Ethereum is partly due to Crypto whales and commercial investors aggressively buying up high volumes of Bitcoin and Ethereum. Cryptocurrencies are now being more widely accepted and legitimized by governments and financial institutions.

Using R, this project forecasts the rise of Cryptocurrency price movements using an application of time-series analysis. The main library used will be Prophet. Prophet is a very powerful forecasting tool with time-series applications. It was developed by developers at Facebook and is mainly used for business analytics in forecasting sales volumes and price analytics. Prophet is effective for business related forecasts that exhibit seasonality, trends, dips and high levels of variation, as the case with Cryptocurrency.

Data Installation

This project will install and load four different libraries. Lubridate, for data wrangling. Ggplot2, for data visualization. Tidyverse, for data manipulation. Prophet, for data forecasting.

```
## Loading required package: lubridate

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Loading required package: ggplot2

## Loading required package: tidyverse

## -- Attaching packages -----------------------------------------------------
---------------------------- tidyverse 1.3.0 --

## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------------------------
---------------------- tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()

## Loading required package: prophet

## Loading required package: Rcpp

## Loading required package: rlang

##
## Attaching package: 'rlang'

## The following objects are masked from 'package:purrr':
##
##     %@%, as_function, flatten, flatten_chr, flatten_dbl, flatten_int,
##     flatten_lgl, flatten_raw, invoke, list_along, modify, prepend,
##     splice
```

The Bitcoin and Ethereum price datasets were downloaded directly from the Historical Data tab found on finance.yahoo.com and are loaded as csv files.

```
bitcoin <- read.csv("BTC-USD.csv", header = TRUE)
ethereum <- read.csv("ETH-USD.csv", header = TRUE)
# Both csv files were retrieved on 01/07/2020, directly from Yahoo finance.
# Bitcoin and Ethereum will be compared against the US Dollar.
```

Preview Datasets

Bellow is a preview of both the downloaded Bitcoin and Ethereum datasets.

```
head(bitcoin)

##         Date       Open       High        Low      Close  Adj.Close
Volume
## 1 2014-09-17 465.864014 468.174011 452.421997 457.334015 457.334015
21056800
## 2 2014-09-18 456.859985 456.859985 413.104004 424.440002 424.440002
34483200
## 3 2014-09-19 424.102997 427.834991 384.532013  394.79599  394.79599
37919700
## 4 2014-09-20 394.673004  423.29599 389.882996 408.903992 408.903992
36863600
## 5 2014-09-21 408.084991 412.425995    393.181 398.821014 398.821014
26580100
```

```
## 6 2014-09-22 399.100006 406.915985 397.130005 402.152008 402.152008
24127600
```

```r
head(ethereum)
```

```
##         Date     Open     High      Low    Close Adj.Close  Volume
## 1 2015-08-07  2.83162  3.53661  2.52112  2.77212   2.77212  164329
## 2 2015-08-08  2.79376  2.79881 0.714725 0.753325  0.753325  674188
## 3 2015-08-09 0.706136  0.87981 0.629191 0.701897  0.701897  532170
## 4 2015-08-10 0.713989 0.729854 0.636546 0.708448  0.708448  405283
## 5 2015-08-11 0.708087  1.13141 0.663235  1.06786   1.06786 1463100
## 6 2015-08-12  1.05875  1.28994 0.883608  1.21744   1.21744 2150620
```

Bellow is a preview of the structure of the Bitcoin dataset. There are 2305 observations and 7 variables.

```r
str(bitcoin)
```

```
## 'data.frame':    2305 obs. of  7 variables:
##  $ Date     : chr  "2014-09-17" "2014-09-18" "2014-09-19" "2014-09-20" ...
##  $ Open     : chr  "465.864014" "456.859985" "424.102997" "394.673004" ...
##  $ High     : chr  "468.174011" "456.859985" "427.834991" "423.29599" ...
##  $ Low      : chr  "452.421997" "413.104004" "384.532013" "389.882996" ...
##  $ Close    : chr  "457.334015" "424.440002" "394.79599" "408.903992" ...
##  $ Adj.Close: chr  "457.334015" "424.440002" "394.79599" "408.903992" ...
##  $ Volume   : chr  "21056800" "34483200" "37919700" "36863600" ...
```

Bellow is a preview of the structure of the Ethereum dataset. There are 1981 observations and 7 variables.

```r
str(ethereum)
```

```
## 'data.frame':    1981 obs. of  7 variables:
##  $ Date     : chr  "2015-08-07" "2015-08-08" "2015-08-09" "2015-08-10" ...
##  $ Open     : chr  "2.83162" "2.79376" "0.706136" "0.713989" ...
##  $ High     : chr  "3.53661" "2.79881" "0.87981" "0.729854" ...
##  $ Low      : chr  "2.52112" "0.714725" "0.629191" "0.636546" ...
##  $ Close    : chr  "2.77212" "0.753325" "0.701897" "0.708448" ...
##  $ Adj.Close: chr  "2.77212" "0.753325" "0.701897" "0.708448" ...
##  $ Volume   : chr  "164329" "674188" "532170" "405283" ...
```

Data Wrangling

In this time-series forecasting analysis, the main variables we will be using are Date and Closing Price. Later on we will be using all historical data observations to develop a prediction model for Crypto prices in the next 365 days.

To ensure the Prophet function has proper functionality, the Date and Closing price classes need to be corrected for both datasets.

```r
# Preview class for Bitcoin Date
class(bitcoin$Date)
```

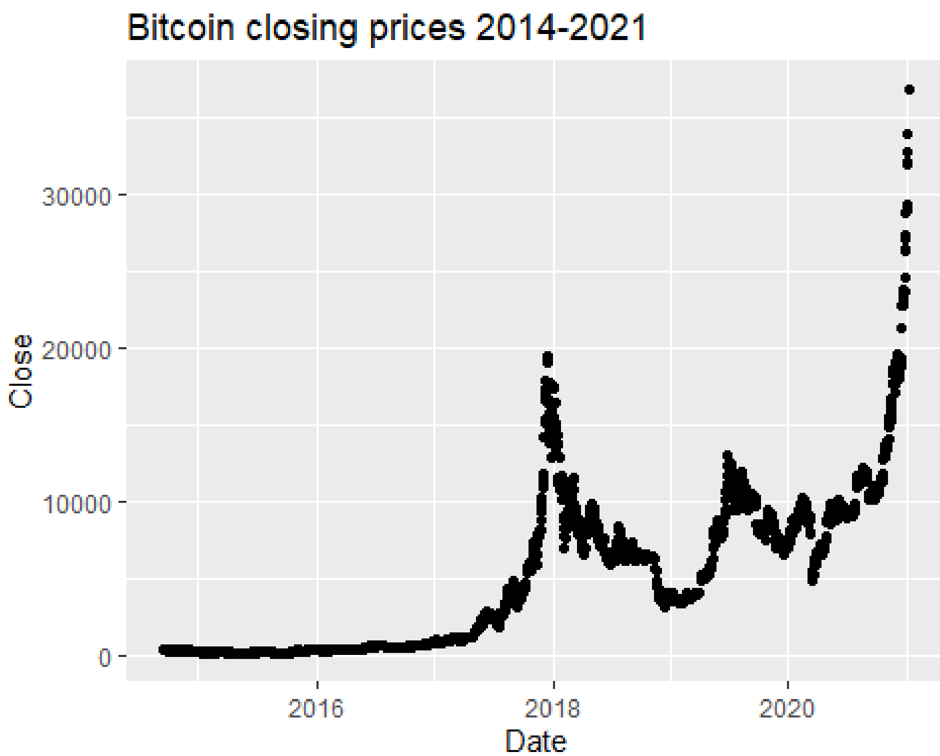
```
## [1] "character"

# Preview class for Bitcoin Close
class(bitcoin$Close)

## [1] "character"

# Preview class for Ethereum Date
class(ethereum$Date)

## [1] "character"

# Preview class for Ethereum Close
class(ethereum$Close)

## [1] "character"

# All classes are in 'character'.
```

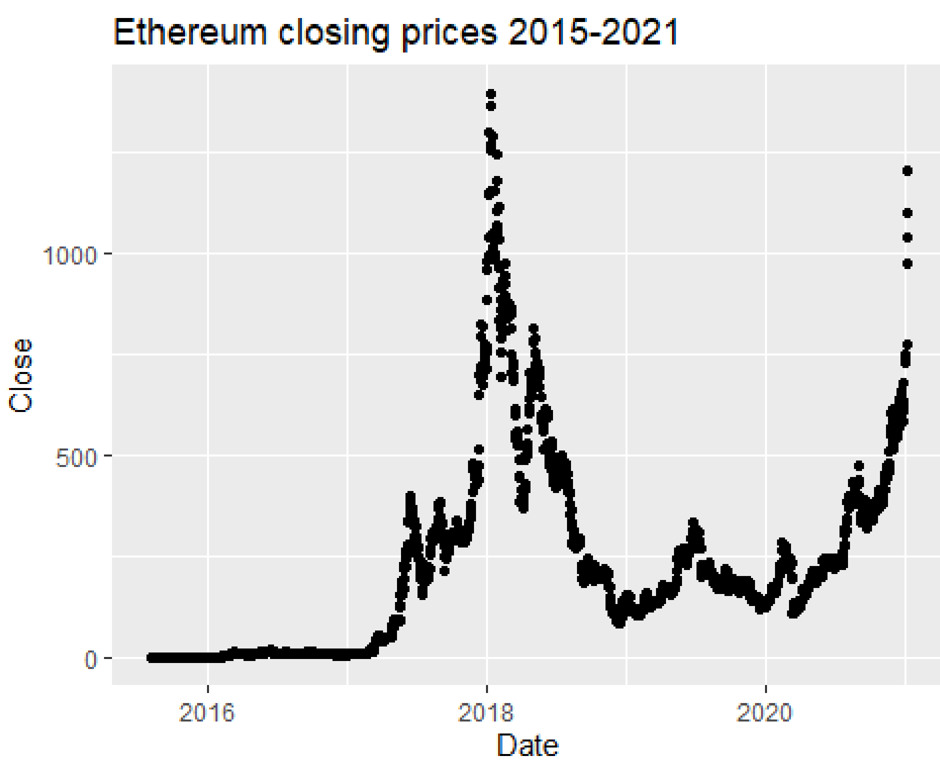
Update class of Date and Close:

```
# Update Date to class Date.
bitcoin$Date <- as.Date(bitcoin$Date)
ethereum$Date <- as.Date(ethereum$Date)

# Update Close to class Numeric.
bitcoin$Close <- suppressWarnings(as.numeric(as.character(bitcoin$Close)))
ethereum$Close <- suppressWarnings(as.numeric(as.character(ethereum$Close)))
```

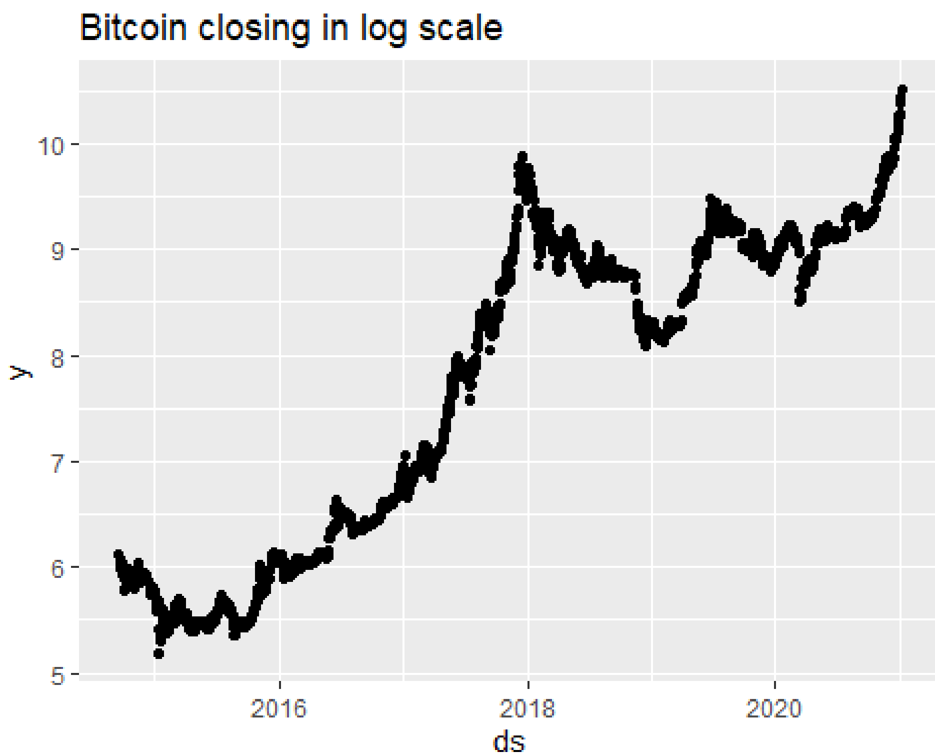
Data Visulization

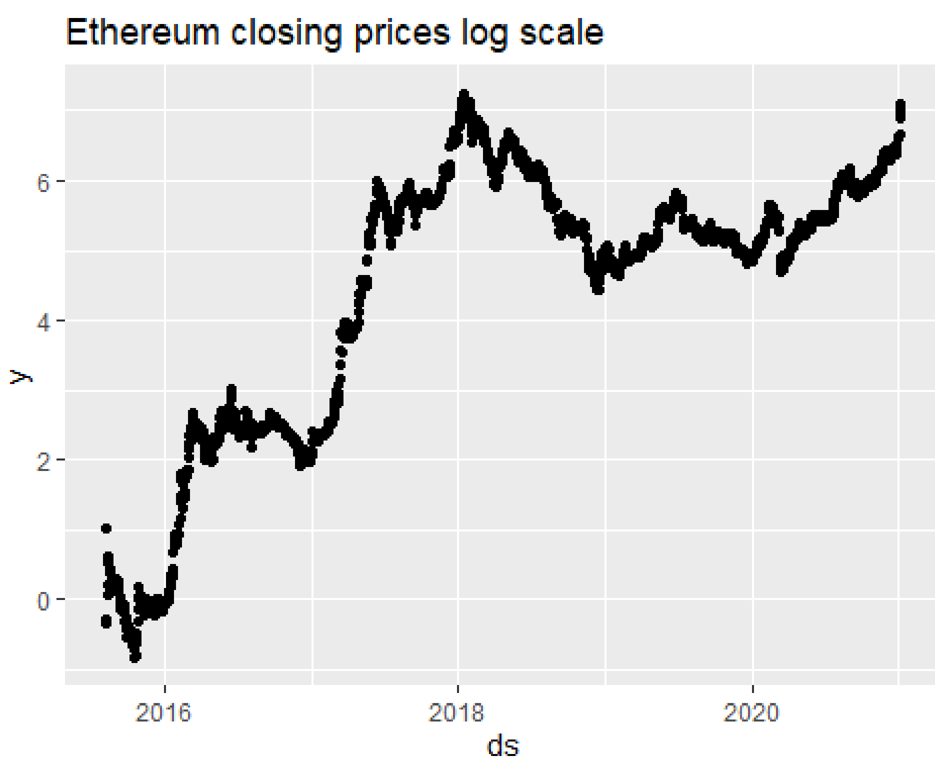
Now that the data has been wrangled and cleaned, we will plot the charts of both datasets.





The closing prices will be transformed using log transformation to enable easier data manipulation and forecasting later on.





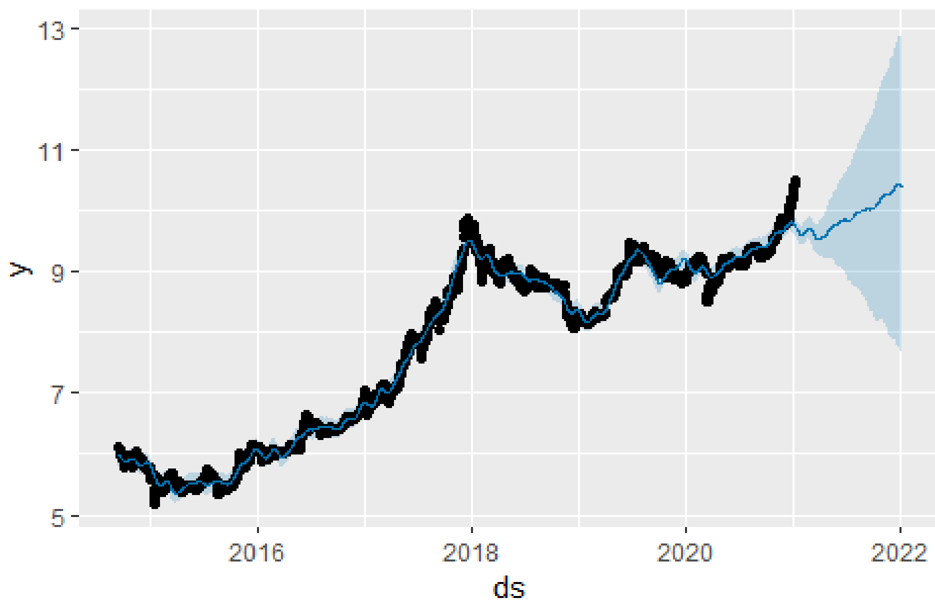
Note that in Prophet, the naming conventions are as follows: ds for date, and y for closing price.

Bitcoin Data Forecasting

```
# Calling the Prophet Function to Fit the Bitcoin Model
ds <- bitcoin$Date
y <- log(bitcoin$Close)
model1 <- prophet(df1)

## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to
override this.

future_price1 <- make_future_dataframe(model1, periods = 365)
# View last few values of the future price 1 data frame
tail(future_price1)

##              ds
## 2665 2022-01-02
## 2666 2022-01-03
## 2667 2022-01-04
## 2668 2022-01-05
## 2669 2022-01-06
## 2670 2022-01-07

# We can validate that the predictions are made until January 2022, 1 year
from now.
forecast1 <- predict(model1, future_price1)
tail(forecast1[c('ds','yhat','yhat_lower','yhat_upper')])

##              ds     yhat yhat_lower yhat_upper
## 2665 2022-01-02 10.41265   7.660972   12.88654
## 2666 2022-01-03 10.41112   7.638563   12.85691
## 2667 2022-01-04 10.40366   7.715476   12.87697
## 2668 2022-01-05 10.39614   7.643338   12.87157
## 2669 2022-01-06 10.38536   7.548252   12.90400
## 2670 2022-01-07 10.37797   7.604926   12.88864
```

yhat represents the predicted value, and the upper and lower bounds represent the confidence interval.

Plot Bitcoin

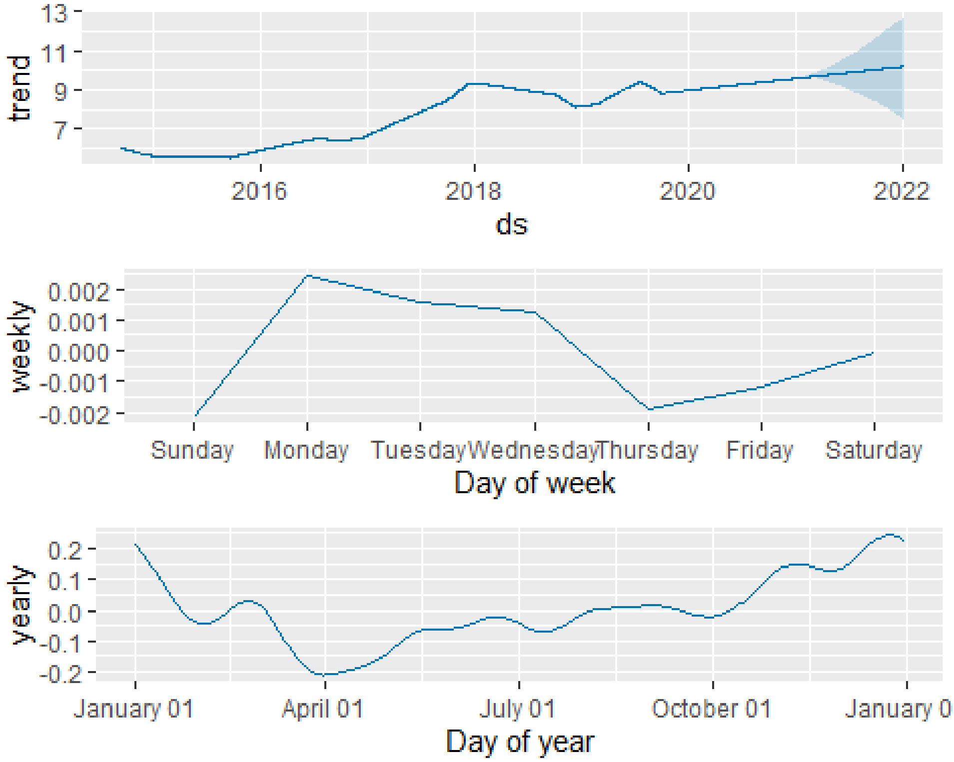


The black dots represent the actual data points. The dark blue line represents the predicted close estimate. The light blue area represents the confidence interval, and with more time, the confidence expands wider due to uncertainty.

Interactive plot that provides date, actual and predicted values

As seen in both plots, the price of Bitcoin begins to surge away from the predicted line at the start of 2021. The new Bitcoin peak reached at the start of 2021 can be calculated using the exponent function to transform back from log scale. On January 5th, the Actual price was33860.35, while the predicted value was 1767.65. A difference of 16183.7.

Interactive plot that provides date, actual and predicted values



As seen in the first graph, Bitcoin will continue its increasing trend from 2020 into 2022. As seen in the second graph, Bitcoin prices are at usually their highest during the start of the working week, and usually at their lowest towards the end of the working week. Looking at the yearly pattern, closing prices are at their highest at January.

Ethereum Data Forecasting

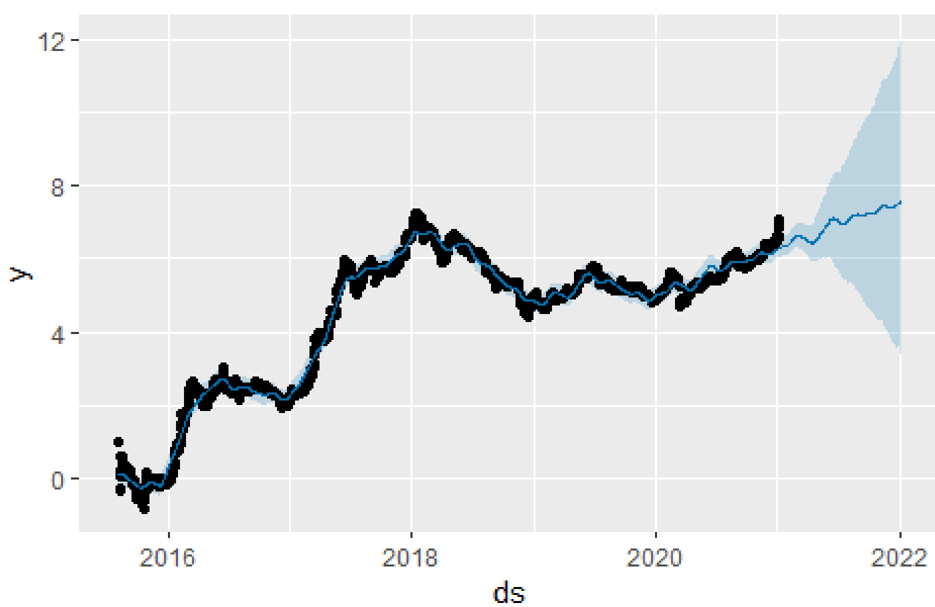
```
# Calling the Prophet Function to Fit the Ethereum Model
ds <- ethereum$Date
y <- log(ethereum$Close)
model2 <- prophet(df2)

## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to
override this.

future_price2 <- make_future_dataframe(model2, periods = 365)
# View last few values of the future price 1 data frame
tail(future_price2)

##                ds
## 2341 2022-01-02
## 2342 2022-01-03
## 2343 2022-01-04
## 2344 2022-01-05
## 2345 2022-01-06
## 2346 2022-01-07
```

```
# We can validate that the predictions are made until January 2022, 1 year from now.
forecast2 <- predict(model2, future_price2)
tail(forecast2[c('ds','yhat','yhat_lower','yhat_upper')])
```

```
##                ds     yhat yhat_lower yhat_upper
## 2341 2022-01-02 7.561743   3.485771   11.96681
## 2342 2022-01-03 7.566729   3.503804   11.96310
## 2343 2022-01-04 7.574096   3.457594   11.84573
## 2344 2022-01-05 7.580456   3.379558   11.97043
## 2345 2022-01-06 7.583273   3.505973   11.95183
## 2346 2022-01-07 7.590966   3.464816   12.01527
```

yhat represents the predicted value, and the upper and lower bounds represent the confidence interval.

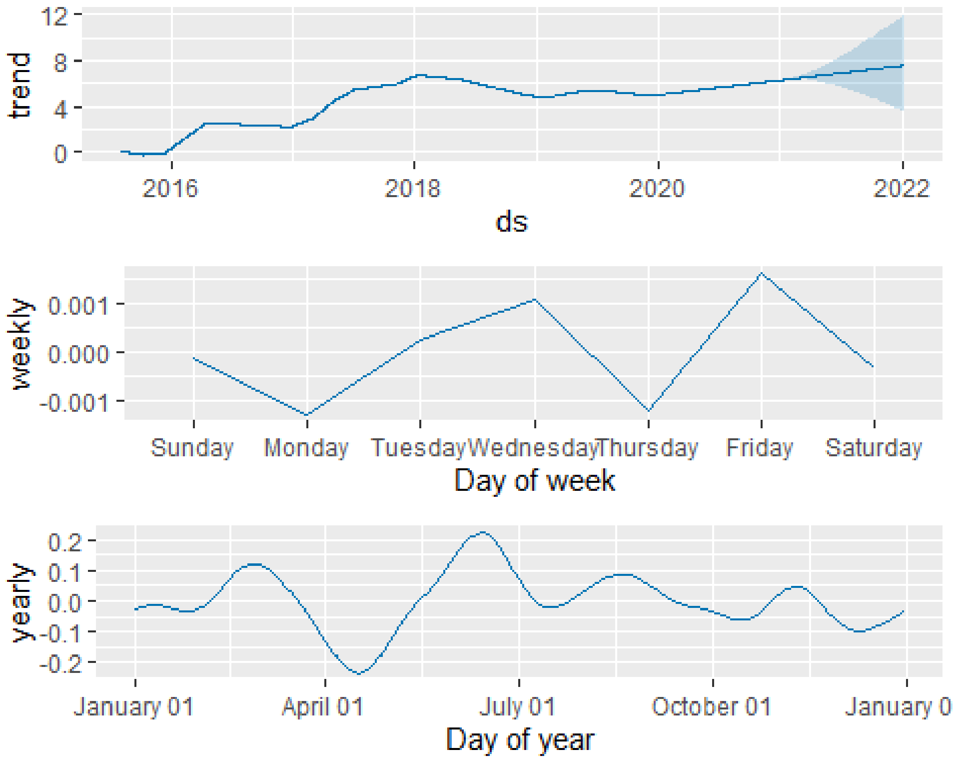
Plot Ethereum



The black dots are the actual data points and the dark blue line is the predicted close estimate. In the case of Ethereum, the light blue confidence interval limits are wider to that of Bitcoin due to higher levels of uncertainty, but the trendline still points to an increasing trend in closing price.

Interactive plot that provides date, actual and predicted values

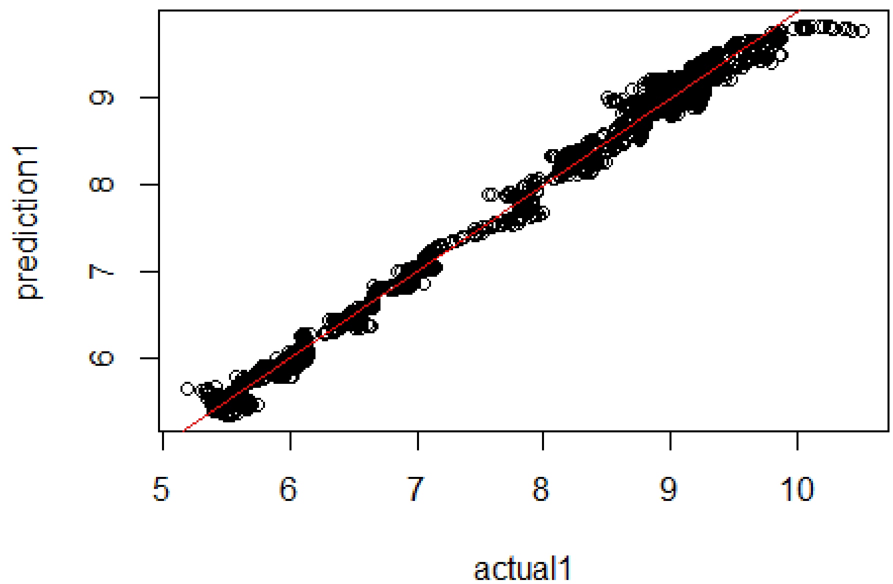
Again with Ethereum, the price begins to sure at the start of 2021. on January 5th, the Actual price was the Actual price was 1096.63, while the predicted value was 544.57. A difference of 522.



Similar to Bitcoin, the trendline continues its increase from 2020 into the rest of 2022. Historically, Ethereum closing prices peak on Fridays and Wednesdays, and are at their lowest on Mondays. Throughout a Historical year, Ethereum prices Peak at June and are at their lowest around May.

Bitcoin Results

```
prediction1 <- forecast1$yhat[1:2304]
actual1 <- model1$history$y
plot(actual1, prediction1)
abline(lm(prediction1 ~ actual1), col = 'red')
```



The predicted points fall close to the predicted line with some variability. To study this variability, a linear regression model will be used.

```
summary(lm(prediction1 ~ actual1))

##
## Call:
## lm(formula = prediction1 ~ actual1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72160 -0.07472  0.00432  0.07820  0.49280
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.062154   0.013470   4.614 4.16e-06 ***
## actual1     0.991945   0.001714 578.624  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1217 on 2302 degrees of freedom
## Multiple R-squared:  0.9932, Adjusted R-squared:  0.9932
## F-statistic: 3.348e+05 on 1 and 2302 DF,  p-value: < 2.2e-16
```

R-Squared is 0.9932, so about 99% of this model is explained by variability.

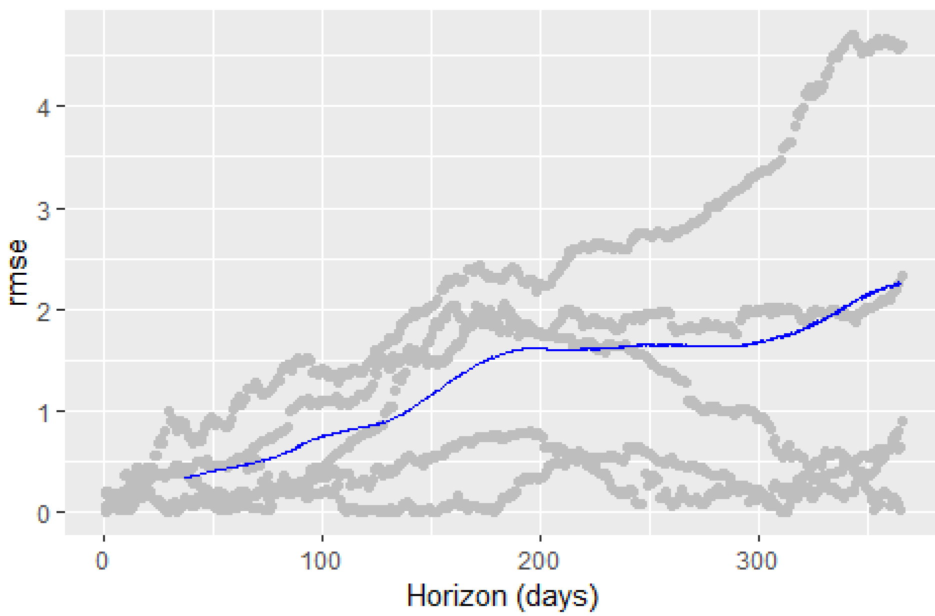
```
x1 <- cross_validation(model1, 365, units = "days")
```

```
## Making 5 forecasts with cutoffs between 2018-01-07 and 2020-01-07
```

```r
performance_metrics(x1, rolling_window = 0.1)
```

With more data points, RMSE increases along with uncertainty. RMSE starts at 0.322 and ends at 2.27

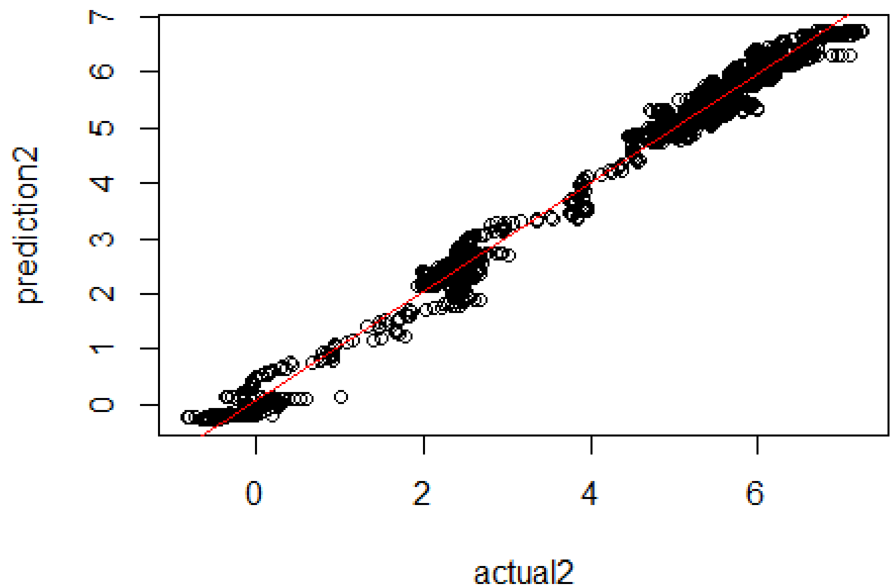
```r
plot_cross_validation_metric(x1, metric = 'rmse', rolling_window = 0.1)
```



Root mean square error is represented in the grey plots. As shown in the graph, the rolling window line moves along with the RMSE.

Ethereum Results

```r
prediction2 <- forecast2$yhat[1:1980]
actual2 <- model2$history$y
plot(actual2, prediction2)
abline(lm(prediction2 ~ actual2), col = 'red')
```



The predicted points fall close to the predicted line with some variability. To study this variability, a linear regression model will be used.

```r
summary(lm(prediction2 ~ actual2))

##
## Call:
## lm(formula = prediction2 ~ actual2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94243 -0.13208  0.00539  0.13010  0.60702
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.068422   0.011222   6.097  1.3e-09 ***
## actual2     0.984402   0.002333 421.884  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2048 on 1978 degrees of freedom
## Multiple R-squared:  0.989,  Adjusted R-squared:  0.989
## F-statistic: 1.78e+05 on 1 and 1978 DF,  p-value: < 2.2e-16
```

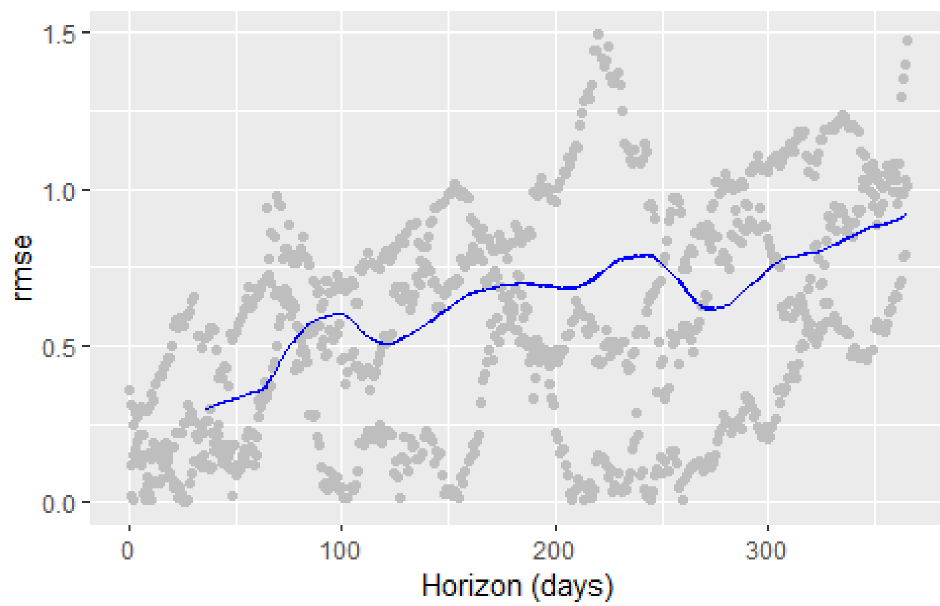
R-Squared is 0.989, so about 98% of this model is explained by variability

```r
x2 <- cross_validation(model2, 365, units = "days")
```

```
## Making 3 forecasts with cutoffs between 2019-01-07 and 2020-01-07
performance_metrics(x2, rolling_window = 0.1)
```

With more data points, RMSE increases along with uncertainty. RMSE starts at 0.29 and ends at 0.92

```
plot_cross_validation_metric(x2, metric = 'rmse', rolling_window = 0.1)
```



Root mean square error is represented in the grey plots. As shown in the graph, the rolling window line moves along with the RMSE.

Conclusion

At the start of 2021, the price of Bitcoin and Ethereum began to skyrocket due to heavy market activity from commercial investors. Governments and private financial institutions have injected billions of dollars into the market and have greatly influenced the surge we see today.

In this project, we used R to forecast Cryptocurrency prices for 365 days using the Prophet function, which which was developed by Facebook developers for business analytics and price forecasting. The models were validated using a linear regression model and root mean square error. The resulting R-Square was 99% and 98% for Bitcoin and Ethereum, respectively. The RMSE starts off close to 0.3 and increases to 2 and 0.9 for Bitcoin and Ethereum, respectively.

A future addition to this forecast would be including a prediction variable representing market activity, the volume of Crypto bought and sold everyday. This would help better capture the relationship between future closing prices and market activity, and would add another layer of complexity and accuracy to the prediction model.