

Binance: BTC and ETH

MGSC: 310 Ryan Welte, Nathan Cuadros, and Ethan Leimel.

TABLE OF CONTENTS

01

MOTIVATION AND SOURCING

We used a dataset based on trading activities every minute from Binance for recent months

02

THE DATA, DATA CLEANING

Apache Arrow, Dataframe..

03

MODELS, BUSINESS OBJECTIVES

ElasticNet, Linear Regression,
Logistic / Classification

04

RESULTS

Hard to produce viable information from this dataset, but has potential



Motivation

- Understand correlations between cryptocurrencies
- Predict future price action of BTC and ETH
- Potential to be used for hyperactive trading algorithms



Source Image:

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fmarkets.businessinsider.com%2Fnews%2Fcryptocurrencies%2Fbitcoin-vs-ethereum-expert>

Sourcing & The Data

- We searched major cryptocurrency exchanges for their trading data
- Binance is a major, international exchange but also with a US version
- Exchange data is done in “pairs,” wherein different digital assets are exchanged
 - (Versus fiat onboarding)
- We had to clean the data to see the below, as a result of the unique format
- Column meanings, data is every minute for the past 4 year period

BTC_Open <dbl>	BTC_High <dbl>	BTC_Low <dbl>	BTC_Close <dbl>	BTC_Volume <dbl>	BTC_QuoteAssetVolume <dbl>	BTC_NumberOfTrades <int>	BTC_TalkerBuyBaseAssetVolume <dbl>	BTC_TalkerBuyQuoteAssetVolume <dbl>	BTC_ETH_OpenTime <s3::POSIXct>	BTC_PriceDiff <dbl>
33862.11	33862.12	33700.00	33783.78	285.00394	9623330.0	4527	54.372814	1837221.0	2021-07-07 17:00:00	-78.32812
33783.78	33900.00	33769.28	33900.00	129.55704	4382238.0	2303	62.745605	2123122.8	2021-07-07 17:01:00	116.21875
33899.99	33929.64	33821.53	33840.02	263.87085	8940190.0	2579	162.349670	5500115.0	2021-07-07 17:02:00	-59.96875
33840.01	33851.02	33756.94	33789.78	50.25749	1699258.1	1351	16.678175	564000.0	2021-07-07 17:03:00	-50.23047
33789.78	33824.70	33766.66	33816.53	44.73964	1512140.9	972	19.517048	659710.0	2021-07-07 17:04:00	26.75000
33818.56	33822.62	33770.00	33806.76	20.17600	681999.7	925	8.627949	291642.5	2021-07-07 17:05:00	-11.79688

Apache Arrow

Binance trading data came in Parquet files

- Parquet files used often at intersection of trading and data science,
 - “efficient as well as performant flat columnar storage format of data compared to row based files like CSV”
- Read into a dataframe using the Apache Arrow package in R
- Head of one initial dataframe appears below
 - (we used both ETH / USDT trading pair, and BTC / USDT trading pairs)

```
```{r}
BTC_USDT = read_parquet(
 file <- here::here("datasets", "BTC-USDT.parquet")),
 col_select = NULL,
 as_data_frame = TRUE,
 props = ParquetArrowReaderProperties$create()
)

ETH_USDT = read_parquet(
 file <- here::here("datasets", "ETH-USDT.parquet")),
 col_select = NULL,
 as_data_frame = TRUE,
 props = ParquetArrowReaderProperties$create()
)
``````
```

Data Cleaning

Original Dataset was massive (25gb in Apache format)

One observation for every minute for 4 years of data.

For each pair compared to each other. ~ 40 pairs compared to one another

So we decided to scale down...

The screenshot shows a dataset page on Kaggle. The title is "Binance Full History" and it describes "1 minute candlesticks for all 1755 cryptocurrency pairs". It was posted by "Jorijn Jacko Smit" a month ago. The page includes tabs for Data, Tasks, Code (11), Discussion (10), Activity, and Metadata. There is a prominent "Download (25 GB)" button and a "New Topic" button. The background of the page features a dark image of a candlestick chart.

Data Link: <https://www.kaggle.com/jorijnsmit/binance-full-history/discussion/198669>

Data Cleaning - Code

```
#data_cleaning:  
BTC_USDT$BTC_PriceDif <- c(BTC_USDT$close - BTC_USDT$open)  
shifted = lag(BTC_USDT$BTC_PriceDif, n = 1)  
BTC_USDT$BTC_LastMinPriceDiff = shifted  
  
BTC_USDT <- BTC_USDT %>% mutate(BTCBuy = BTC_LastMinPriceDiff > 0)  
  
ETH_USDT$ETH_PriceDif <- c(ETH_USDT$close - ETH_USDT$open)  
shifted = lag(ETH_USDT$ETH_PriceDif, n = 1)  
ETH_USDT$ETH_LastMinPriceDiff = shifted  
  
ETH_USDT <- ETH_USDT %>% mutate(ETHBuy = ETH_LastMinPriceDiff > 0)  
  
names(BTC_USDT)[1:10] <- c("BTC_Open",  
"BTC_High", "BTC_Low", "BTC_Close", "BTC_Volume", "BTC_QuoteAssetVolume", "BTC_NumberofTrades", "BTC_TalkerBuyBase  
Assetvolume", "BTC_TalkerBuyQuoteAssetVolume", "BTC_ETH_OpenTime")  
  
names(ETH_USDT)[1:9] <- c("ETH_Open",  
"ETH_High", "ETH_Low", "ETH_Close", "ETH_Volume", "ETH_QuoteAssetVolume", "ETH_NumberofTrades", "ETH_TalkerBuyBase  
Assetvolume", "ETH_TalkerBuyQuoteAssetVolume")  
  
BTC_USDT1month = slice_tail(BTC_USDT, n = 43200) #one month  
ETH_USDT1month = slice_tail(ETH_USDT, n = 43200) #one month  
  
BTC_USDT1year = slice_tail(BTC_USDT, n = 525600) #one year  
ETH_USDT1year = slice_tail(ETH_USDT, n = 525600) #one year  
  
BTC_USDT11month = slice_head(BTC_USDT1year, n = 482400) #11 months  
ETH_USDT11month = slice_head(ETH_USDT1year, n = 482400) #11 months  
  
ETH_BTC_USDT11month = cbind(BTC_USDT11month, ETH_USDT11month)  
ETH_BTC_USDT11month = subset(ETH_BTC_USDT11month, select = -c(open_time))  
  
ETH_BTC_USDT = cbind(BTC_USDT1month, ETH_USDT1month)  
ETH_BTC_USDT = subset(ETH_BTC_USDT, select = -c(open_time))
```

Data Cleaning

Focus on two pairs:

BTC v USDT

ETH v USDT

Full ~2,081,094 observations

One Year = 525,600 observations

One Month = 43,200 observations

In addition to the variables on the right, we added a few features:

Price Difference (Open - Close)

Last Minute Price Difference

And a boolean variable called Buy:

- marked as a true if the last minute price difference was positive
- marked as a false if the next minute price difference was negative

EACH PAIR HAS FOLLOWING INFO

Open Price

High Price

Low Price

Close Price

Volume Traded

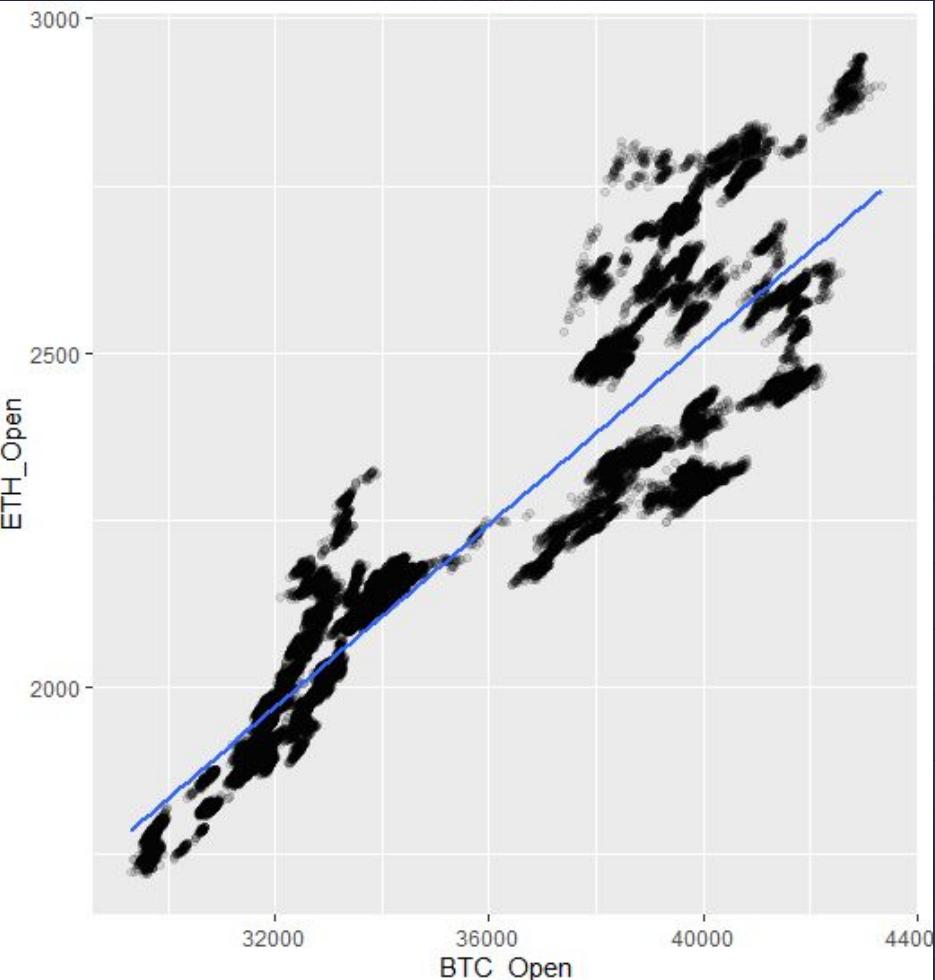
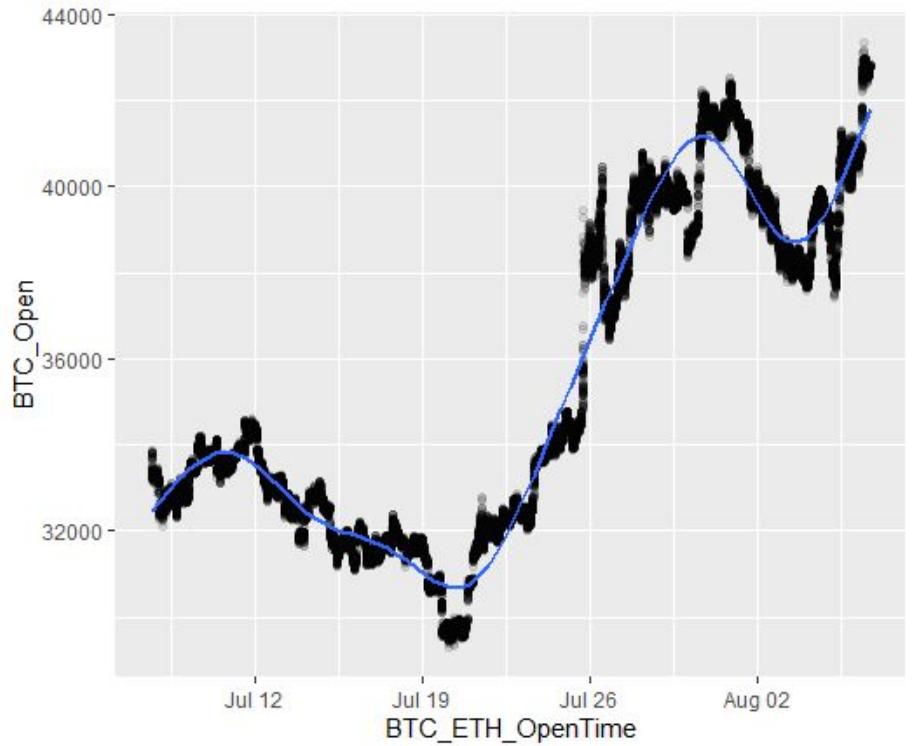
Quote Asset Volume

Number of Trades

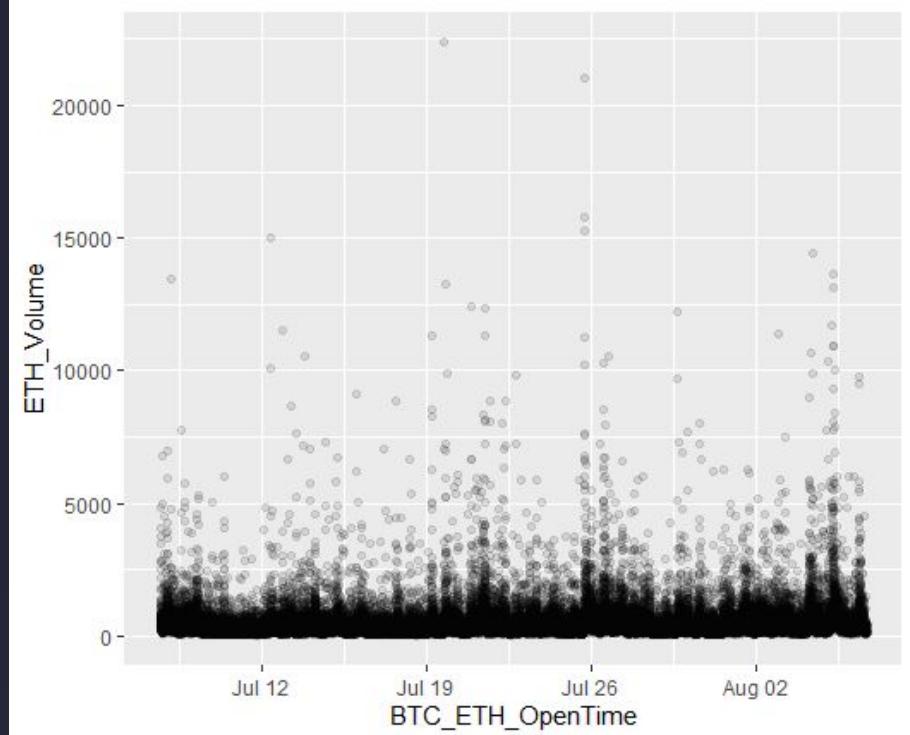
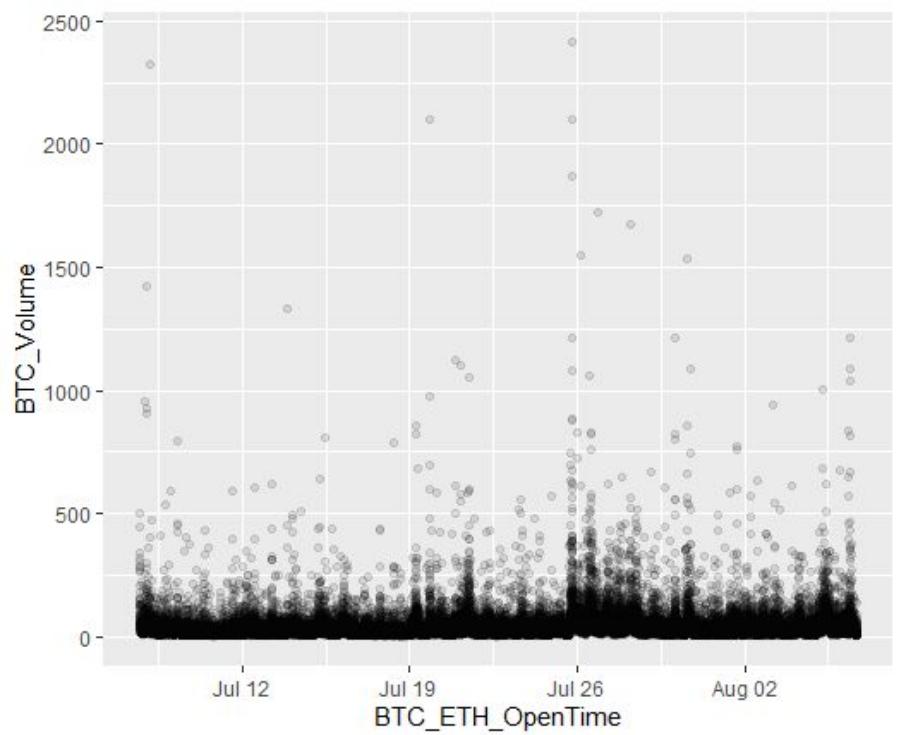
Taker Buy Base Asset Volume

Taker Buy Quote Asset Volume

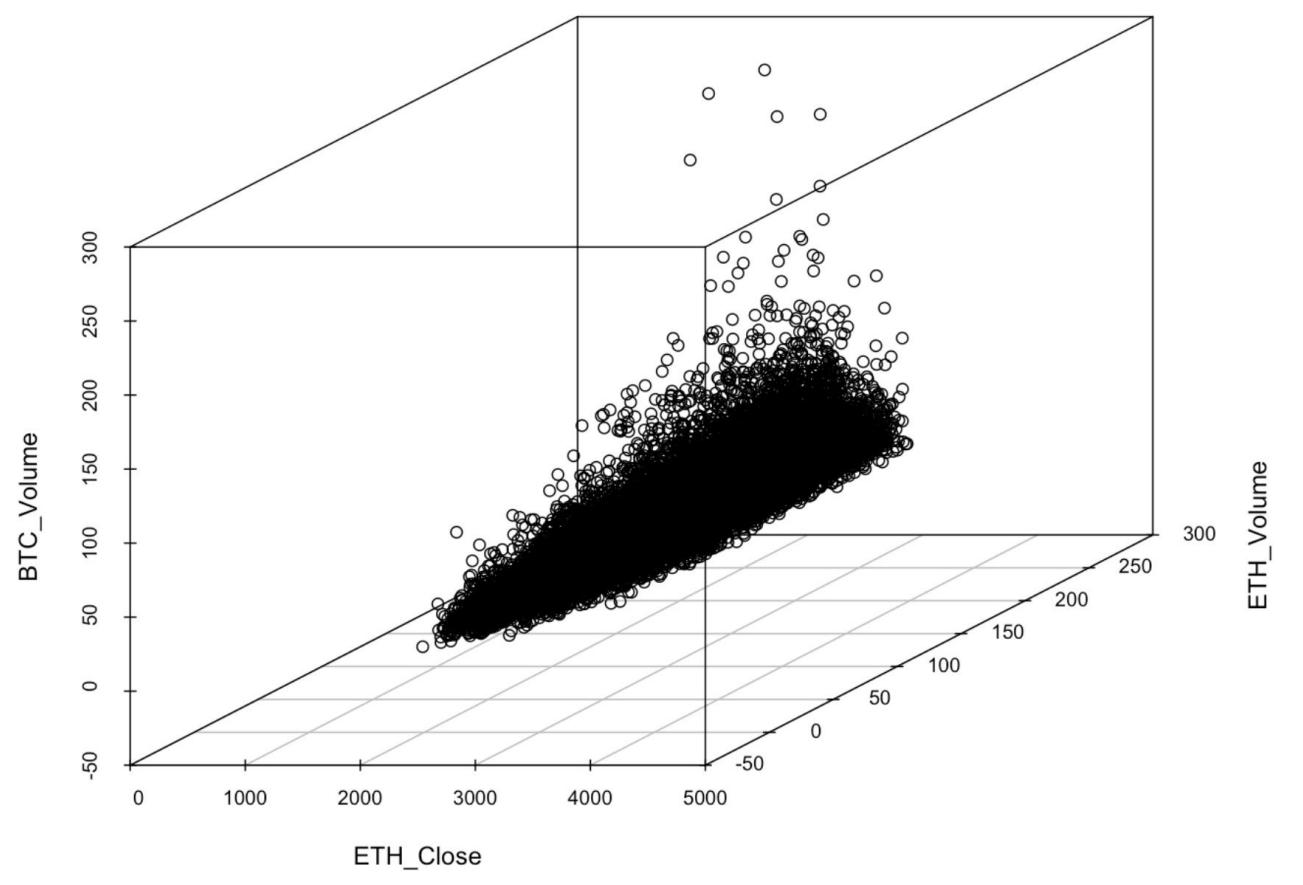
Open Time



DATA VISUALIZED



DATA VISUALIZED



DATA VISUALIZED

Reasons for higher volume

Final Data

All of the previous Variables but for BTC and ETH:

11 Month Data = 482400 obs. 25 Variables wide

| BTC_Open | BTC_High | BTC_Low | BTC_Close | BTC_Volume | BTC_QuoteAssetVolume | BTC_NumberOfTrades | BTC_TalkerBuyBaseAssetVolume | BTC_TalkerBuyQuoteAssetVolume | BTC_ETH_OpenTime | BTC_PriceDif | BTC_LastMinPriceDiff | BTCBuy |
|----------|----------|----------|-----------|------------|----------------------|--------------------|------------------------------|-------------------------------|---------------------|--------------|----------------------|--------|
| 11632.08 | 11640.21 | 11631.89 | 11640.20 | 35.59820 | 414194.0 | 455 | 21.666656 | 252091.52 | 2020-08-06 01:04:00 | 8.12011719 | 3.13964844 | TRUE |
| 11640.20 | 11662.71 | 11640.19 | 11661.15 | 63.01425 | 734114.2 | 911 | 35.758518 | 416570.06 | 2020-08-06 01:05:00 | 20.95019531 | 8.12011719 | TRUE |
| 11661.15 | 11665.29 | 11658.41 | 11665.29 | 40.74549 | 475116.8 | 467 | 13.504595 | 157481.38 | 2020-08-06 01:06:00 | 4.13964844 | 20.95019531 | TRUE |
| 11665.29 | 11666.37 | 11660.34 | 11664.99 | 56.22023 | 655721.6 | 413 | 16.494774 | 192381.55 | 2020-08-06 01:07:00 | -0.29980469 | 4.13964844 | TRUE |
| 11664.68 | 11670.00 | 11664.65 | 11667.44 | 57.79376 | 674305.8 | 531 | 34.298206 | 400172.16 | 2020-08-06 01:08:00 | 2.76074219 | -0.29980469 | FALSE |
| 11667.45 | 11670.00 | 11663.30 | 11664.90 | 74.34583 | 867526.2 | 511 | 37.331669 | 435616.59 | 2020-08-06 01:09:00 | -2.54980469 | 2.76074219 | TRUE |
| 11664.89 | 11670.00 | 11664.89 | 11669.40 | 41.16197 | 480262.9 | 426 | 19.480251 | 227298.58 | 2020-08-06 01:10:00 | 4.51074219 | -2.54980469 | FALSE |
| 11669.41 | 11674.37 | 11669.40 | 11672.76 | 114.24286 | 1333247.5 | 516 | 96.457245 | 1125666.25 | 2020-08-06 01:11:00 | 3.34960938 | 4.51074219 | TRUE |

| ETH_Open | ETH_High | ETH_Low | ETH_Close | ETH_Volume | ETH_QuoteAssetVolume | ETH_NumberOfTrades | ETH_TalkerBuyBaseAssetVolume | ETH_TalkerBuyQuoteAssetVolume | ETH_PriceDif | ETH_LastMinPriceDiff | ETHBuy |
|----------|----------|---------|-----------|------------|----------------------|--------------------|------------------------------|-------------------------------|--------------|----------------------|--------|
| 394.54 | 394.73 | 394.40 | 394.67 | 1210.24048 | 477507.50 | 500 | 579.77637 | 228747.281 | 0.130004883 | -0.059997559 | FALSE |
| 394.66 | 394.95 | 394.65 | 394.95 | 863.37103 | 340832.56 | 241 | 478.49503 | 188912.984 | 0.290008545 | 0.130004883 | TRUE |
| 394.94 | 395.49 | 394.92 | 395.41 | 542.61682 | 214412.58 | 230 | 392.28162 | 155002.266 | 0.470001221 | 0.290008545 | TRUE |
| 395.43 | 395.97 | 395.36 | 395.96 | 574.83020 | 227416.62 | 305 | 385.03357 | 152306.641 | 0.529998779 | 0.470001221 | TRUE |
| 395.96 | 396.00 | 395.56 | 395.90 | 326.35202 | 129171.44 | 201 | 168.64676 | 66746.117 | -0.059997559 | 0.529998779 | TRUE |
| 395.90 | 396.38 | 395.90 | 396.26 | 1108.85950 | 439311.59 | 448 | 465.04163 | 184217.984 | 0.360015869 | -0.059997559 | FALSE |
| 396.27 | 396.53 | 396.20 | 396.22 | 702.19690 | 278337.62 | 284 | 220.39751 | 87354.914 | -0.049987793 | 0.360015869 | TRUE |
| 396.22 | 396.54 | 396.15 | 396.47 | 603.87799 | 239389.62 | 246 | 173.07681 | 68611.438 | 0.250000000 | -0.049987793 | FALSE |
| 396.49 | 396.73 | 396.31 | 396.56 | 690.17590 | 273682.41 | 295 | 362.46872 | 143720.109 | 0.070007324 | 0.250000000 | TRUE |

We have the Data formatted... Now what?

We decided that our training set would be 11 months and our test set would be 1 month of data

This allows for plenty of training space for the model yet a large amount of test points due to the size of the dataset.

Dates of the sets:

Train: 08-06-2020 01:04:00-> 07-07-2021 16:59:00

Test: 07-07-21 17:00:00 -> 08-06-2021 16:59:00

Data Questions

Question 1:

Since we have split the dataset into an 11-month period followed by a 1-month, can we gain any valuable correlation or predictive insight from a simple regression analysis based on variables we think are relevant to price?

Question 2:

Is it possible to predict the minute change of the other crypto pair given information solely of the other? EX: Predict BTC based on only ETH data?

Question 3:

Can we predict whether or not to buy , hold, or sell Bitcoin or Ethereum based off the past minute of price action?

Model I: Linear Regression

Can we determine whether there is a correlation between closing price compared to the other columns we are given, or can they be used to make general predictions regarding price?

Linear Regression

- Used variables as predictors which seemed, intuitively, would have an impact on closing price, such as the closing price of the inverse currency
- BTC and ETH known to be highly correlated, but with ETH more volatile and often lagging behind BTC, reactionarily

```
total_split <- initial_split(ETH_BTC_USDT, 0.9)
total_train <- training(total_split)
total_test <- testing(total_split)

lm_mod1 = lm(ETH_Close ~ BTC_Volume+ BTC_Close + ETH_Volume + BTC_PriceDif, data = total_train)
summary(lm_mod1)
lm_mod2 = lm(BTC_Close ~ BTC_Volume + ETH_Close + ETH_Volume + ETH_PriceDif, data = total_train)
summary(lm_mod2)

preds_train <- predict(lm_mod1, newdata = total_train)
preds_test <- predict(lm_mod1, newdata = total_test)

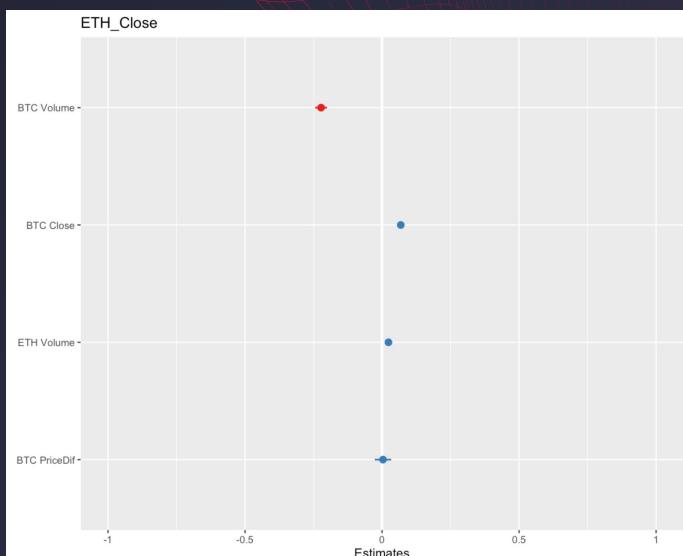
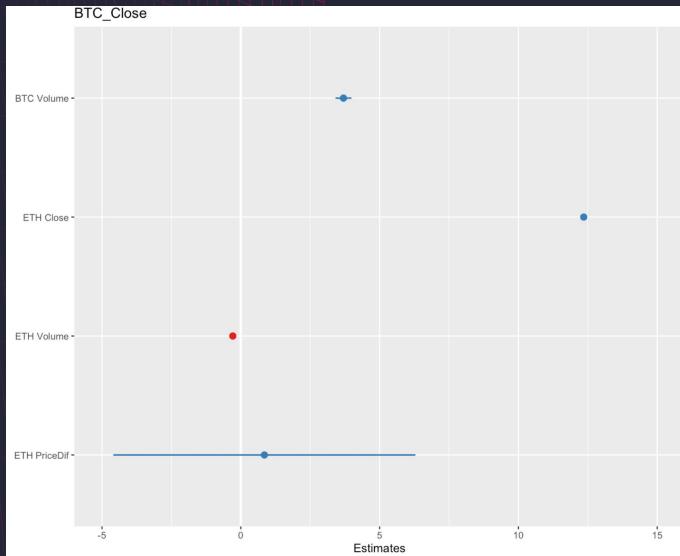
plot_model(lm_mod1)
plot_model(lm_mod2)
```

Linear Regression

Coefficient tables..

Show ETH Price Difference has a large impact on BTC_Close, but with a wide range

For ETH_Close, BTC_Volume seems to be strongest predictor, and it's a negative correlation



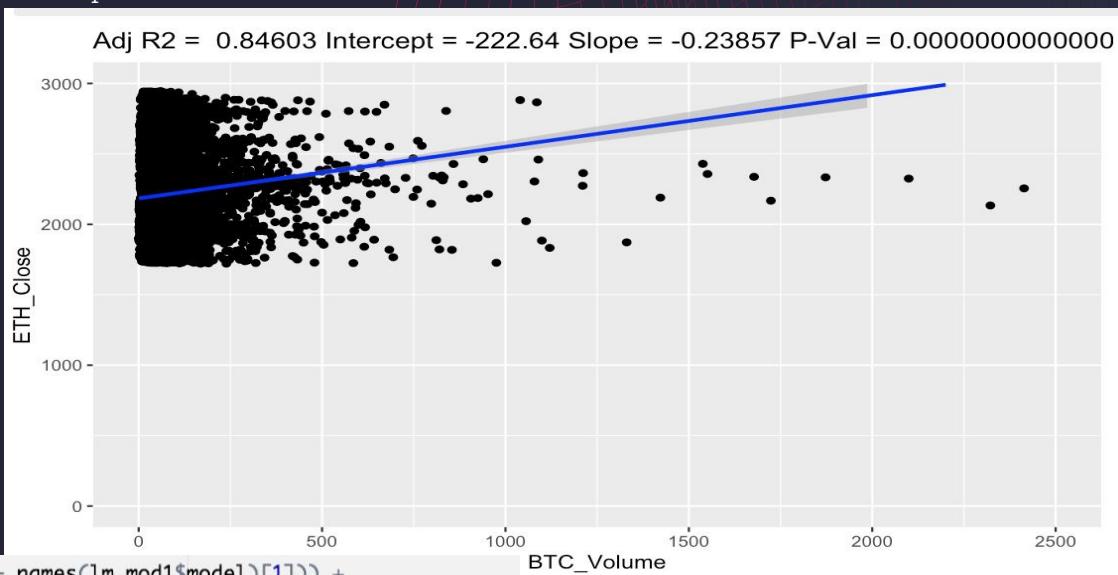
Linear Regression

- sjPlot Tab Model

| ETH_Close | | | | BTC_Close | | | |
|--|---------------|-------------------|--------|--|---------------|-------------------|--------|
| Predictors | Estimates | CI | p | Predictors | Estimates | CI | p |
| (Intercept) | -221.36 | -231.68 – -211.04 | <0.001 | (Intercept) | 8187.10 | 8070.93 – 8303.27 | <0.001 |
| BTC Volume | -0.22 | -0.24 – -0.20 | <0.001 | BTC Volume | 3.70 | 3.41 – 3.99 | <0.001 |
| BTC Close | 0.07 | 0.07 – 0.07 | <0.001 | ETH Close | 12.35 | 12.30 – 12.41 | <0.001 |
| ETH Volume | 0.02 | 0.02 – 0.03 | <0.001 | ETH Volume | -0.29 | -0.31 – -0.26 | <0.001 |
| BTC PriceDif | 0.00 | -0.03 – 0.03 | 0.820 | ETH PriceDif | 0.85 | -4.59 – 6.29 | 0.760 |
| Observations | 38880 | | | Observations | 38880 | | |
| R ² / R ² adjusted | 0.846 / 0.846 | | | R ² / R ² adjusted | 0.846 / 0.846 | | |

Linear Regression

- Shows that BTC volume does not adequately explain the whole ETH model, because even though it has a coef of -0.24, there is still a positive linear correlation when incorporating all the variables



```
ggplot(lm_mod1$model, aes_string(x = names(lm_mod1$model)[2], y = names(lm_mod1$model)[1])) +  
  geom_point() +  
  stat_smooth(method = "lm", col = "blue") +  
  labs(title = paste("Adj R2 = ", signif(summary(lm_mod1)$adj.r.squared, 5),  
                 "Intercept = ", signif(lm_mod1$coef[[1]], 5),  
                 "Slope = ", signif(lm_mod1$coef[[2]], 5),  
                 "P-Val = ", signif(summary(lm_mod1)$coef[2,4], 5))) + xlim(20,15000) + ylim(1500, 3000)
```

Linear Regression

```
Call:  
lm(formula = ETH_Close ~ BTC_Volume + BTC_Close + ETH_Volume +  
    BTC_PriceDif, data = total_train)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-536.11  -61.01   -10.64    58.27  565.99  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -221.3614343  5.2656771 -42.039 <0.000000000000002 ***  
BTC_Volume   -0.2224421  0.0109470 -20.320 <0.000000000000002 ***  
BTC_Close     0.0683782  0.0001487 459.780 <0.000000000000002 ***  
ETH_Volume    0.0233634  0.0009613  24.305 <0.000000000000002 ***  
BTC_PriceDif  0.0034745  0.0152495   0.228          0.82  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 109.1 on 38875 degrees of freedom  
Multiple R-squared:  0.8461, Adjusted R-squared:  0.8461  
F-statistic: 5.343e+04 on 4 and 38875 DF, p-value: < 0.000000000000022
```

- Model Performance
 - Explains nearly 84% of the price change for both BTC and ETH with the predictors selected: closing price of other, volume of other, volume, and volatility
 - This means that we can easily predict price based on historical data,
 - ETH MAE:
Testing: 85.50178
Training: 85.16782
 - BTC MAE:
Testing: 1107.509
Training: 1099.556

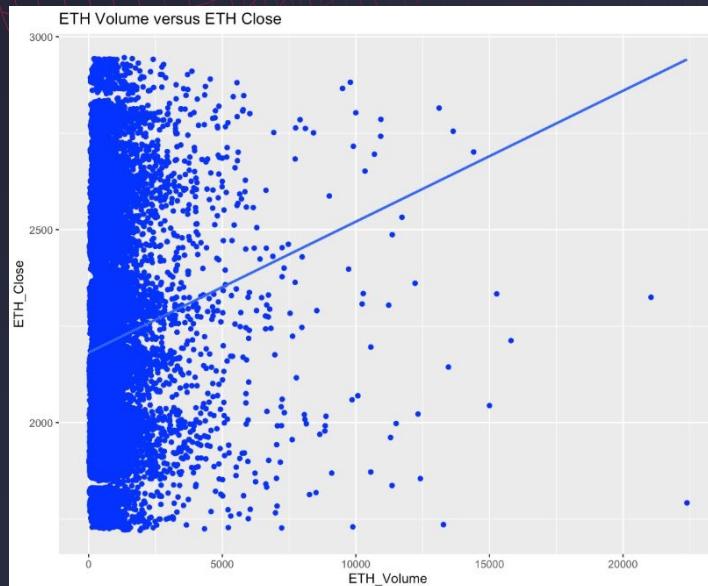
```
Call:  
lm(formula = BTC_Close ~ BTC_Volume + ETH_Close + ETH_Volume +  
    ETH_PriceDif, data = total_train)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-9651.3  -839.8  -330.1   328.8  5664.0  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  8187.10083  59.27049 138.131 <0.000000000000002 ***  
BTC_Volume    3.69794   0.14660  25.226 <0.000000000000002 ***  
ETH_Close     12.35282   0.02687 459.747 <0.000000000000002 ***  
ETH_Volume   -0.28877   0.01293 -22.327 <0.000000000000002 ***  
ETH_PriceDif  0.84882   2.77584   0.306          0.76  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 1467 on 38875 degrees of freedom  
Multiple R-squared:  0.8464, Adjusted R-squared:  0.8464  
F-statistic: 5.355e+04 on 4 and 38875 DF, p-value: < 0.000000000000022
```

Linear Regression



Linear Regression – Conclusion

- Linear Regression is most helpful for determining general trends related to how the changes in factors related to the variable on which we are predicting, but more importantly, attributes of the opposite currency, have on price.
- Helpful insight to build another model off of, but obviously you could not start with some of the information we had
- ETH is positively correlated with its own volume, and BTC Price, with no correlation for BTC Price change. It is negatively correlated to BTC volume which may show a flow from Ether to Bitcoin
- BTC is positively correlated with its own volume, and ETH Price, positively with price changes in ETH, and negatively with the opposite volume, similarly to ETH
- Strongest correlation to volume of other one (negative), and price of other one (positive)



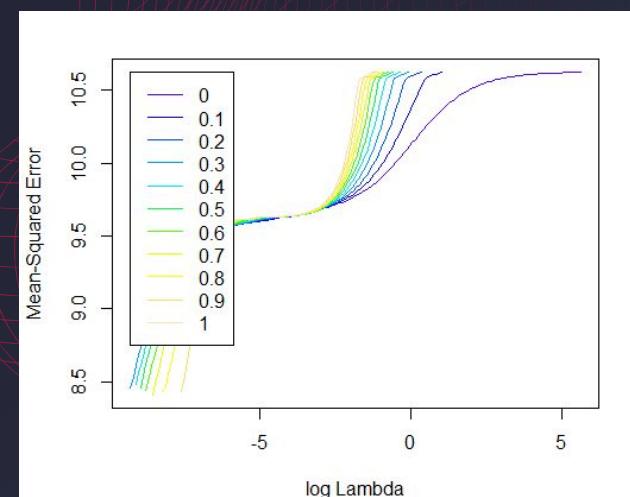
Model 2: Elastic Net

Is it possible to predict the minute change of the other crypto pair given information solely of the other? EX: Predict BTC based on only ETH data?

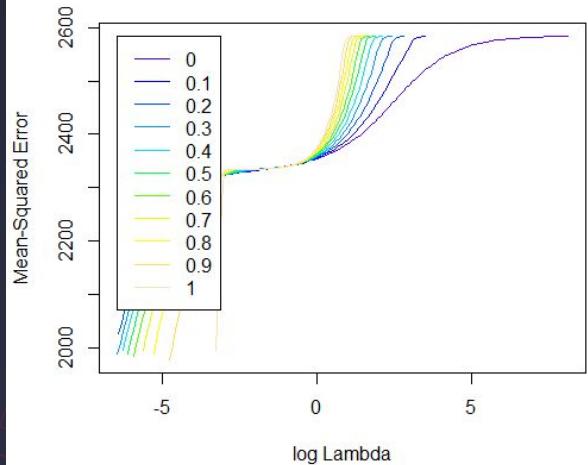
Elastic Net

```
enet_mod_ETH <- cva.glmnet(ETH_PriceDiff ~ BTC_Open + BTC_High +  
BTC_Low + BTC_Close + BTC_Volume + BTC_QuoteAssetVolume +  
BTC_NumberOfTrades + BTC_TalkerBuyBaseAssetVolume +  
BTC_TalkerBuyQuoteAssetVolume,  
                           data = ETH_BTC_USDT11month,  
                           alpha = seq(0,1, by = 0.1))  
  
enet_mod_BTC <- cva.glmnet(BTC_PriceDiff ~ ETH_Open + ETH_High +  
ETH_Low + ETH_Close + ETH_Volume + ETH_QuoteAssetVolume +  
ETH_NumberOfTrades + ETH_TalkerBuyBaseAssetVolume +  
ETH_TalkerBuyQuoteAssetVolume,  
                           data = ETH_BTC_USDT11month,  
                           alpha = seq(0,1, by = 0.1))
```

Predicting
ETH



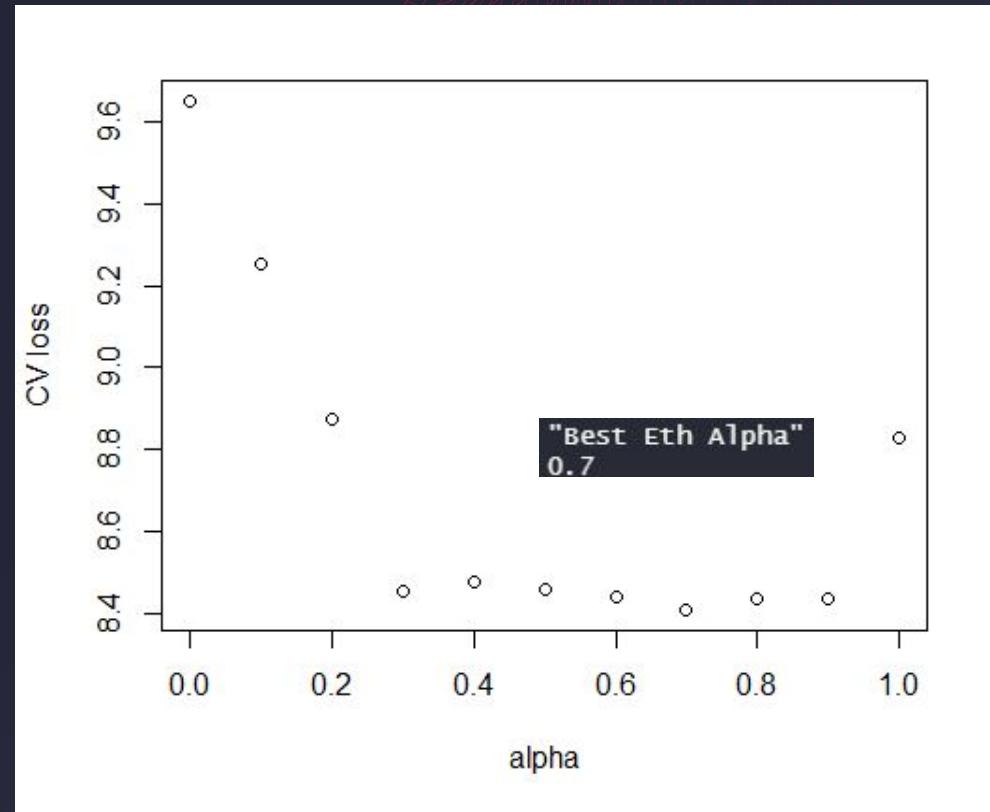
Predicting
BTC



Elastic Net

Predicting
ETH

```
[1] "Coefs For ETH"
10 x 1 sparse Matrix of class "dgCMatrix"
           s1
(Intercept) 0.001080469336
BTC_Open     -0.006286752650
BTC_High    0.001041556294
BTC_Low     0.000058624609
BTC_Close   0.005189141334
BTC_Volume  0.013257193272
BTC_QuoteAssetVolume -0.000001258945
BTC_NumberofTrades -0.000114053769
BTC_TalkerBuyBaseAssetvolume -0.024411716492
BTC_talkerbuyquoteassetvolume 0.000002516258
```



alpha
<dbl>

0.7

lambdaMin
<dbl>

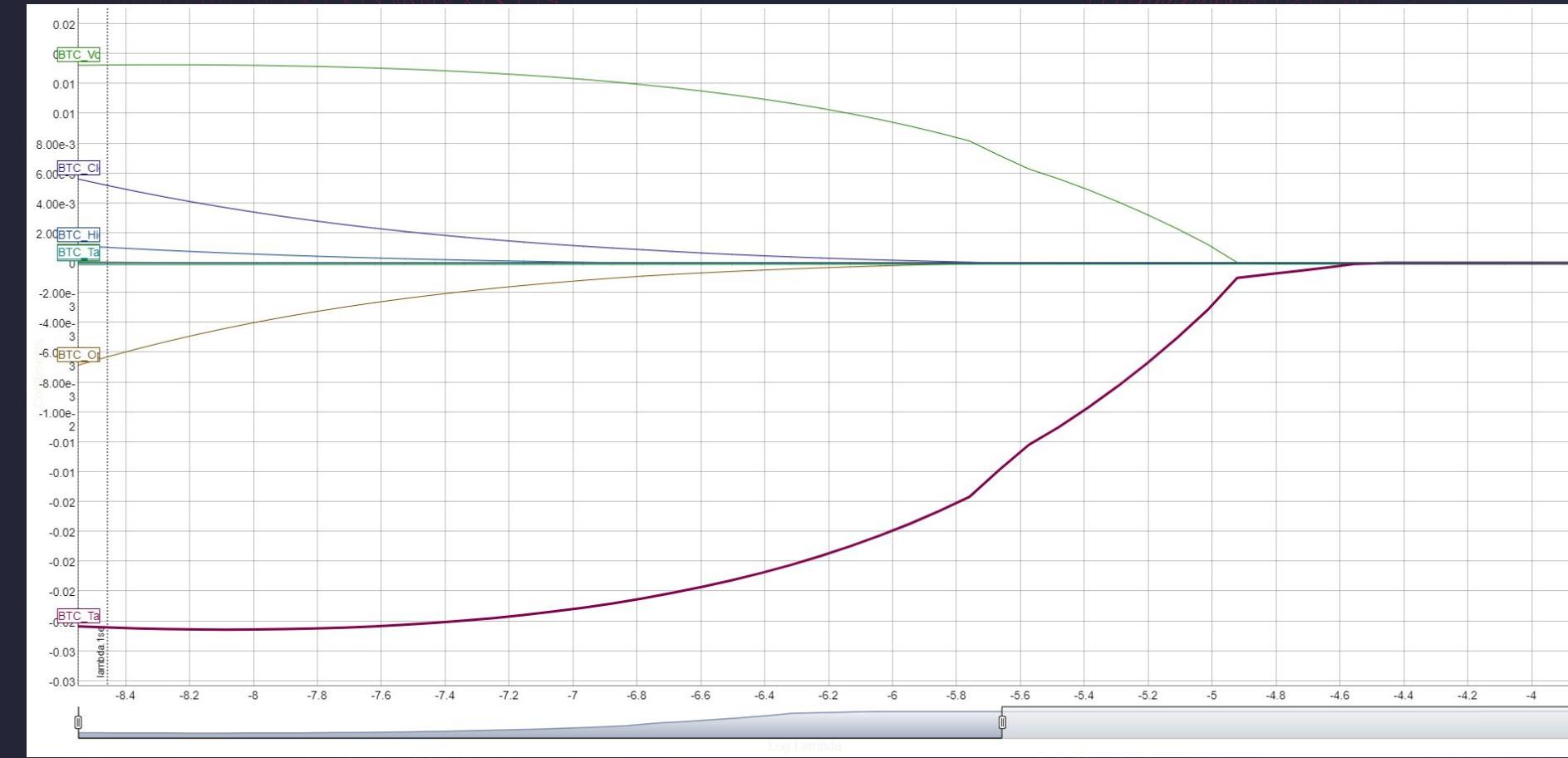
0.0001937685

lambdaSE
<dbl>

0.0002126607

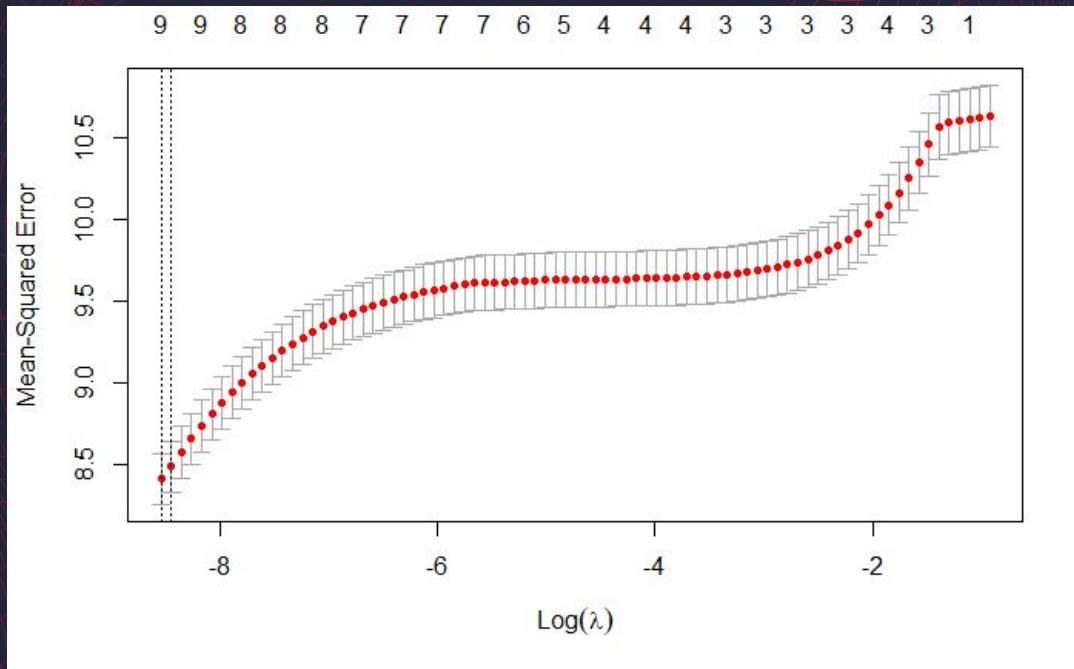
eror
<dbl>

8.40991



Elastic Net

Predicting
ETH



alpha
<dbl>

0.7

lambdaMin
<dbl>

0.0001937685

lambdaSE
<dbl>

0.0002126607

eror
<dbl>

8.40991

Elastic Net

Predicting
BTC

```
[1] "Coefs For BTC"
10 x 1 sparse Matrix of class "dgCMatrix"

           s1
(Intercept) 0.07052978029
ETH_Open     -1.91581542700
ETH_High     0.12789285855
ETH_Low      -
ETH_Close    1.78820788396
ETH_Volume   0.000090799059
ETH_QuoteAssetVolume -0.00002558075
ETH_NumberOfTrades -0.00089071199
ETH_TalkerBuyBaseAssetVolume -0.00107270106
ETH_TalkerBuyQuoteAssetVolume 0.00005091806
```

alpha
<dbl>

0.9

lambdaMin
<dbl>

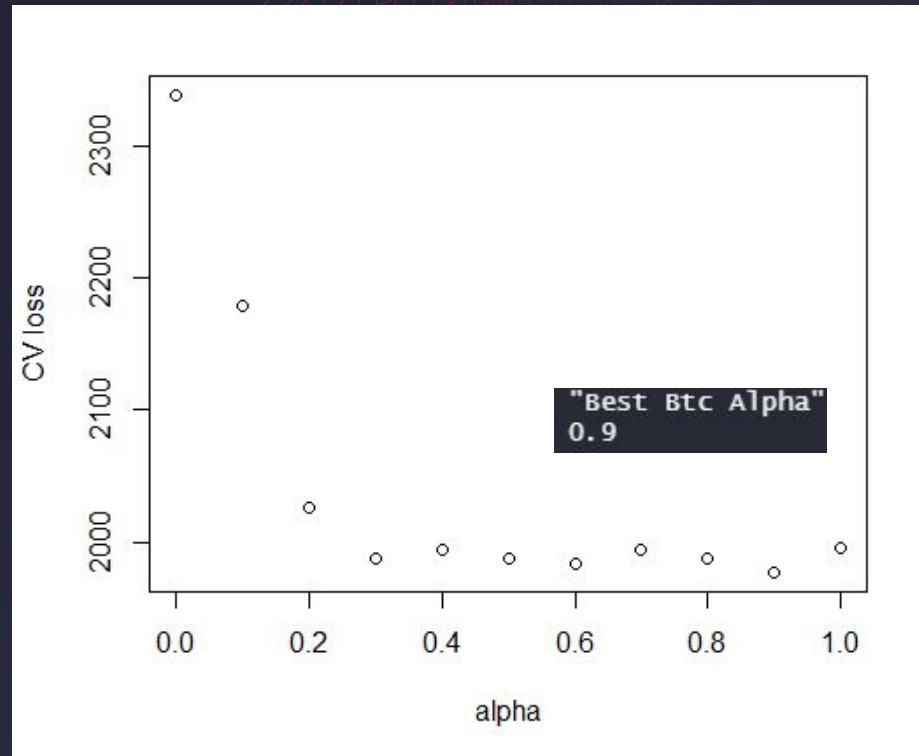
0.008634273

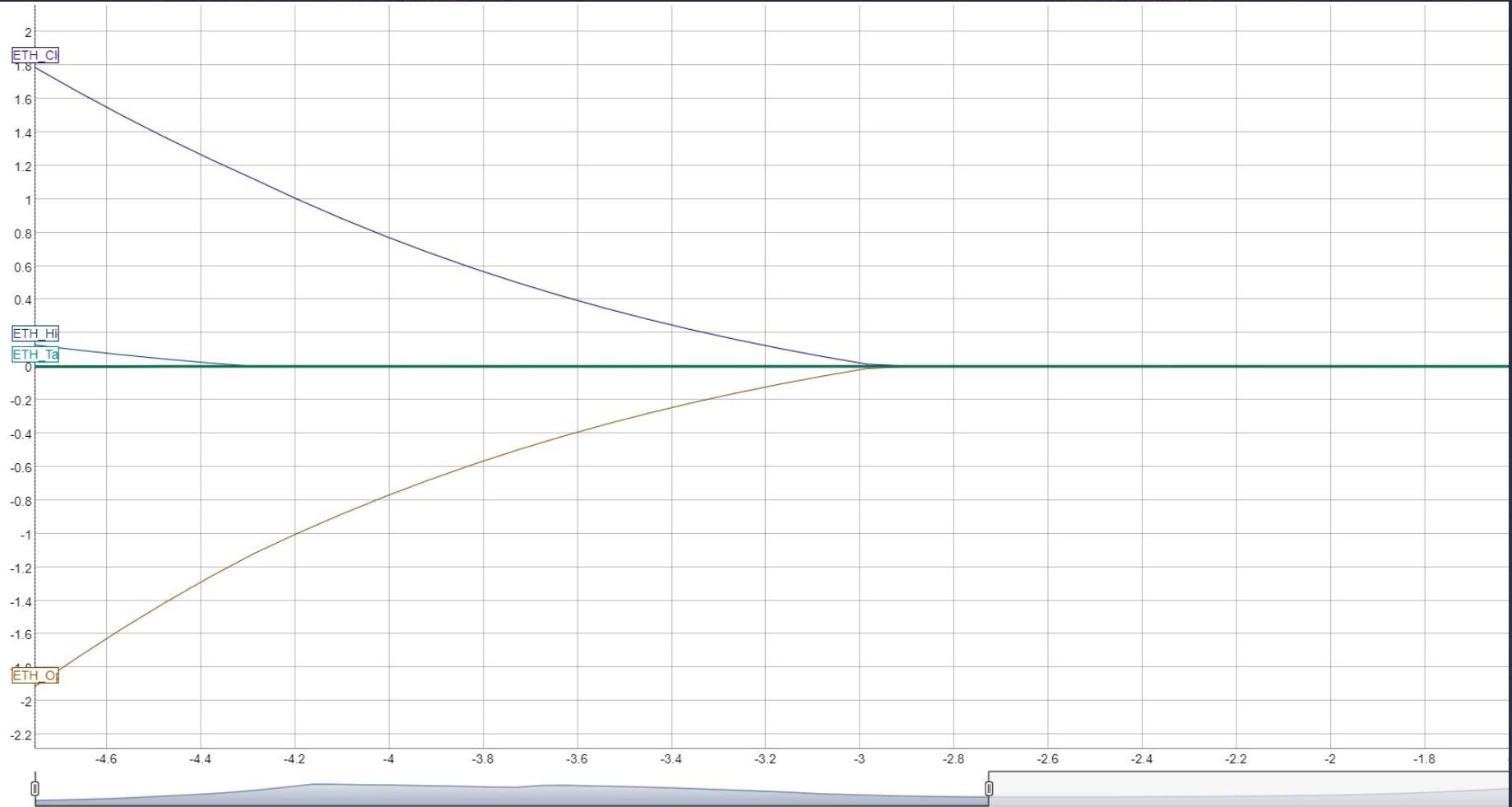
lambdaSE
<dbl>

0.008634273

eror
<dbl>

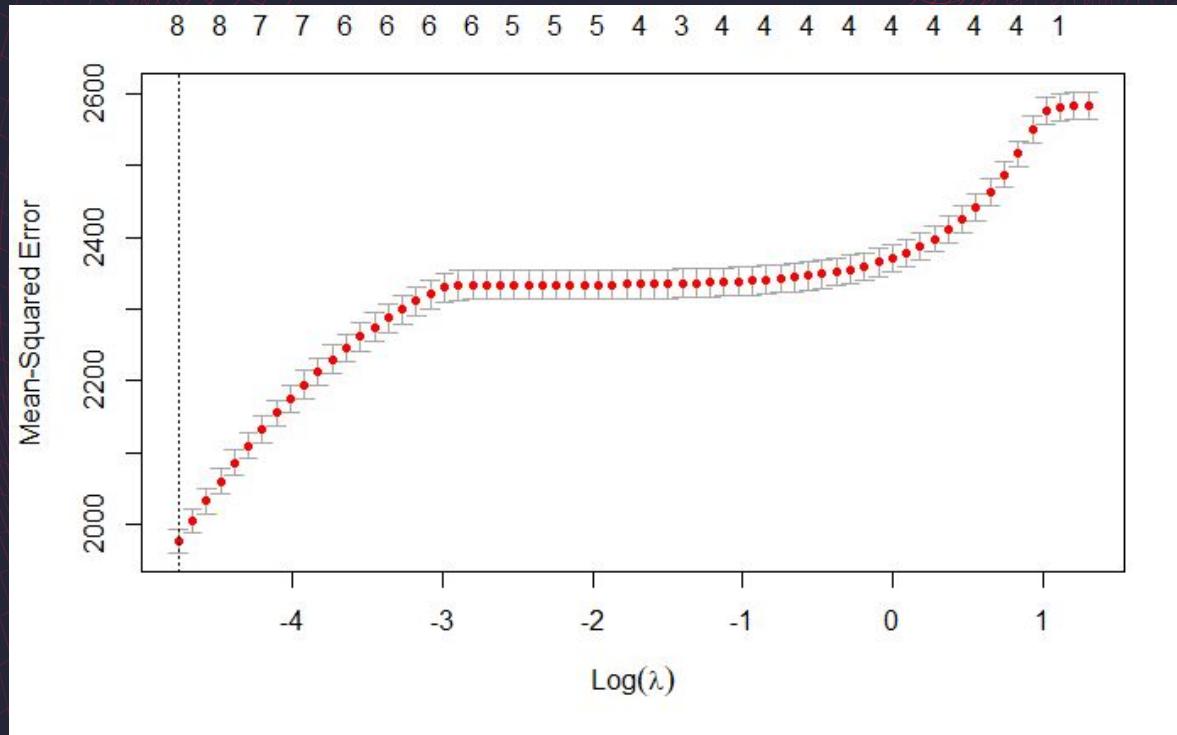
1977.347





Elastic Net

Predicting
BTC



alpha
<dbl>

0.9

lambdaMin
<dbl>

0.008634273

lambdaSE
<dbl>

0.008634273

eror
<dbl>

1977.347

Elastic Net: So, how did it do?

```
varsBTC <- c('BTC_Open', 'BTC_High', 'BTC_Low', 'BTC_Close', 'BTC_Volume', 'BTC_QuoteAssetVolume', 'BTC_NumberOfTrades', 'BTC_TalkerBuyBaseAssetVolume',  
'BTC_TalkerBuyQuoteAssetVolume')  
BTC_USDT1monthX = BTC_USDT1month[varsBTC]  
BTC_USDT11monthX = BTC_USDT11month[varsBTC]  
BTC_USDT1monthY = BTC_USDT1month['BTC_PriceDif']  
BTC_USDT11monthY = BTC_USDT11month['BTC_PriceDif']  
  
varsETH <- c('ETH_Open', 'ETH_High', 'ETH_Low', 'ETH_Close', 'ETH_Volume', 'ETH_QuoteAssetVolume', 'ETH_NumberOfTrades', 'ETH_TalkerBuyBaseAssetVolume',  
'ETH_TalkerBuyQuoteAssetVolume')  
ETH_USDT1monthX = ETH_USDT1month[varsETH]  
ETH_USDT11monthX = ETH_USDT11month[varsETH]  
ETH_USDT1monthY = ETH_USDT1month['ETH_PriceDif']  
ETH_USDT11monthY = ETH_USDT11month['ETH_PriceDif']  
  
#preds test  
EthPreds1month = predict(enet_mod_ETH, BTC_USDT1monthX, alpha = best_alpha_ETH)  
BtcPreds1month = predict(enet_mod_BTC, ETH_USDT1monthX, alpha = best_alpha_BTC)  
#preds train  
EthPreds11month = predict(enet_mod_ETH, BTC_USDT11monthX, alpha = best_alpha_ETH)  
BtcPreds11month = predict(enet_mod_BTC, ETH_USDT11monthX, alpha = best_alpha_BTC)  
  
BTC_USDT1monthY$pred <- BtcPreds1month  
ETH_USDT1monthY$pred <- EthPreds1month  
  
BTC_USDT11monthY$pred <- BtcPreds11month  
ETH_USDT11monthY$pred <- EthPreds11month
```

Elastic Net: So, how did it do?

```
"Test MSE"  
"BTC"  
19.54091  
"ETH"  
1.656208  
"Test RSQ"  
"BTC"  
0.2716237  
"ETH"  
0.2225512
```

```
[1] "1 month"  
[1] "BTC"  
  BTC_PriceDif      pred.lambda.1se  
Min. :-759.4102  Min. :-610.2004  
1st Qu.: -16.1509 1st Qu.: -5.2223  
Median :  0.1406  Median :  0.4253  
Mean   :  0.2143  Mean   :  0.2096  
3rd Qu.: 16.4102  3rd Qu.:  5.8609  
Max.  :1783.9023  Max.  : 724.5594  
[1] "ETH"  
  ETH_PriceDif      pred.lambda.1se  
Min. :-33.00000  Min. :-35.34904  
1st Qu.: -1.40002 1st Qu.: -0.20731  
Median :  0.01001  Median :  0.03939  
Mean   :  0.01290  Mean   :  0.03694  
3rd Qu.:  1.41016  3rd Qu.:  0.27604  
Max.  : 61.12012  Max.  : 37.23607
```

```
"Train MSE"  
"BTC"  
25.83561  
"ETH"  
1.457341  
"Train RSQ"  
"BTC"  
0.2728549  
"ETH"  
0.2255522
```

```
[1] "11 months"  
[1] "BTC"  
  BTC_PriceDif      pred.lambda.1se  
Min. :-2157.9102  Min. :-696.8154  
1st Qu.: -14.1211  1st Qu.: -2.7760  
Median : -0.0195  Median :  0.1893  
Mean   :  0.0443  Mean   :  0.0443  
3rd Qu.: 13.9902  3rd Qu.:  3.3004  
Max.  : 1510.6797  Max.  : 529.6729  
[1] "ETH"  
  ETH_PriceDif      pred.lambda.1se  
Min. :-215.46997  Min. :-39.59875  
1st Qu.: -0.64999  1st Qu.: -0.16782  
Median :  0.00000  Median :  0.00866  
Mean   :  0.00339  Mean   :  0.00339  
3rd Qu.:  0.65997  3rd Qu.:  0.19438  
Max.  : 168.91003  Max.  : 33.66883
```

Model Conclusions: Elastic Net

This model works well. It's not great at predicting extremes of the data.

Neither model was extremely overfit or underfit

MSE's were small in comparison to domain of dataset

Still not very applicable to a business money making method. (not accurate enough) and I did not apply any time series methodology so the model is not built for forecasting.

Still useful for analyzing past data.

Verdict: semi-usuable

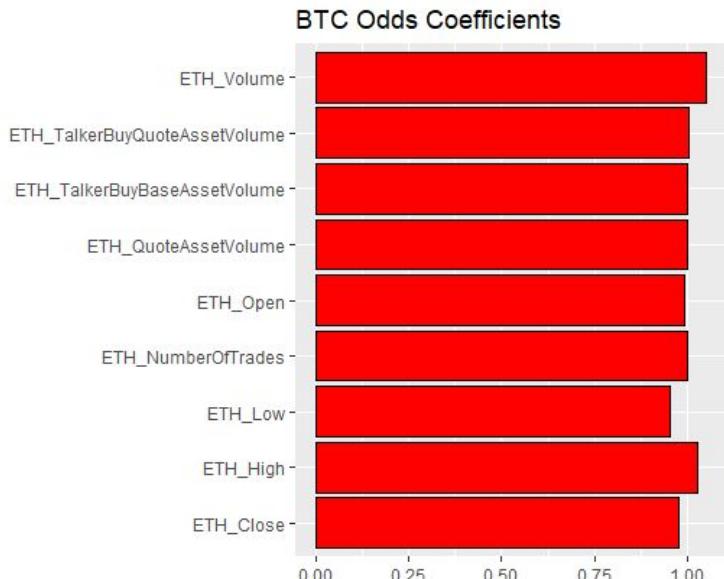
Model 3: Logistic



Can we predict whether or not to buy or sell
Bitcoin or Ethereum based off the past minute's
price action?

Logistic: BTC Model

Exponentiated



```
plot <- ggplot(coefs, aes(x = Coefficient.values, y = conditional))  
+ geom_bar(stat="identity", color = "black", fill = "red")  
+ ggtitle("BTC Odds Coefficients") + labs(x = "", y = "")  
  
print(plot)
```

```
logit_fit_train <- glm(BTCBuy ~ ETH_Open + ETH_High + ETH_Low + ETH_Close  
+ ETH_Volume + ETH_QuoteAssetVolume + ETH_NumberOfTrades  
+ ETH_TalkerBuyBaseAssetVolume  
+ ETH_TalkerBuyQuoteAssetVolume,  
family = binomial,  
data = ETH_BTC_USDT1month)
```

Unexponentiated

```
Deviance Residuals:  
    Min      1Q   Median      3Q     Max  
-6.2014 -1.1699 -0.5004  1.1671  4.6493  
  
Coefficients:  
                Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.00749080551 0.00681254398 -1.100 0.27152  
ETH_Open      0.02475682392 0.00200516220 12.347 < 2e-16 ***  
ETH_High     -0.04753044083 0.00228497439 -20.801 < 2e-16 ***  
ETH_Low       -0.02410570019 0.00234362654 -10.286 < 2e-16 ***  
ETH_Close      0.04690118463 0.00217199293 21.594 < 2e-16 ***  
ETH_Volume    -0.00181868409 0.00002160060 -84.196 < 2e-16 ***  
ETH_QuoteAssetVolume 0.00000057117 0.00000001305 43.769 < 2e-16 ***  
ETH_NumberOfTrades 0.00003459287 0.00001057229  3.272 0.00107 **  
ETH_TalkerBuyBaseAssetVolume 0.00360274403 0.00004121384 87.416 < 2e-16 ***  
ETH_TalkerBuyQuoteAssetVolume -0.00000115869 0.00000002369 -48.905 < 2e-16 ***  
---  
Signif. codes: 0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 668728 on 482399 degrees of freedom  
Residual deviance: 655496 on 482390 degrees of freedom  
AIC: 655516
```

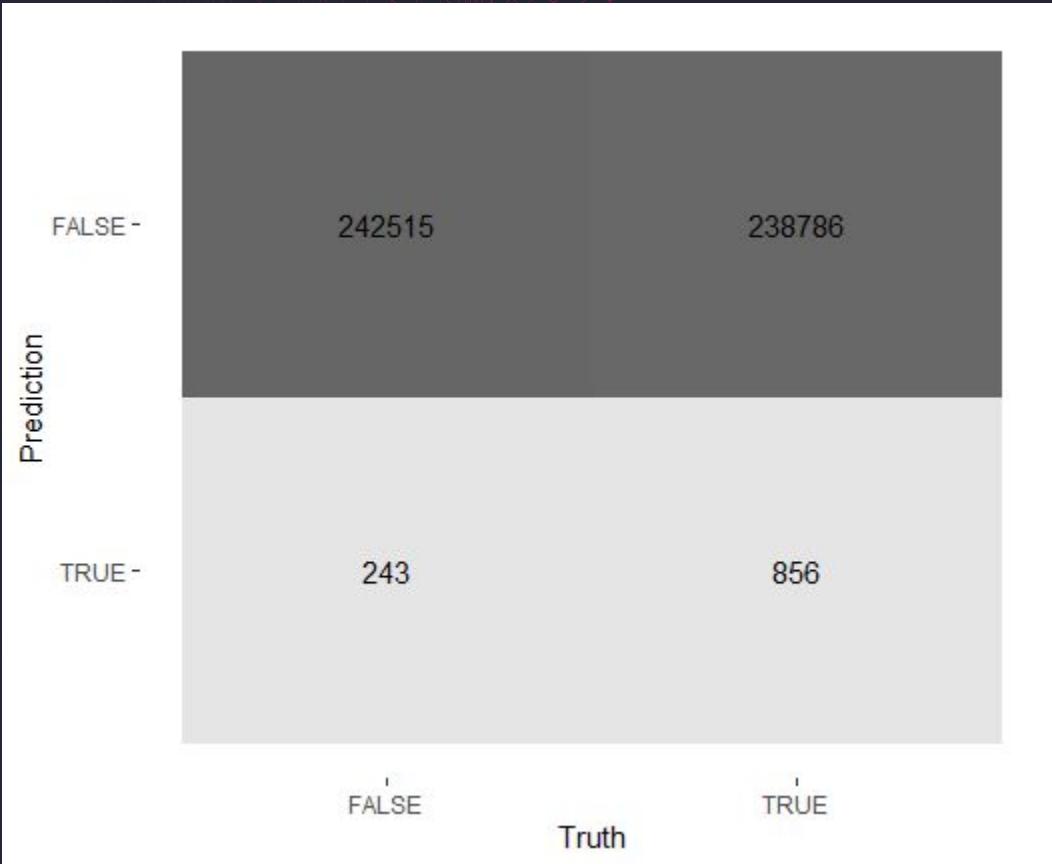
Number of Fisher Scoring iterations: 4

Logistic: BTC Confusion Matrix



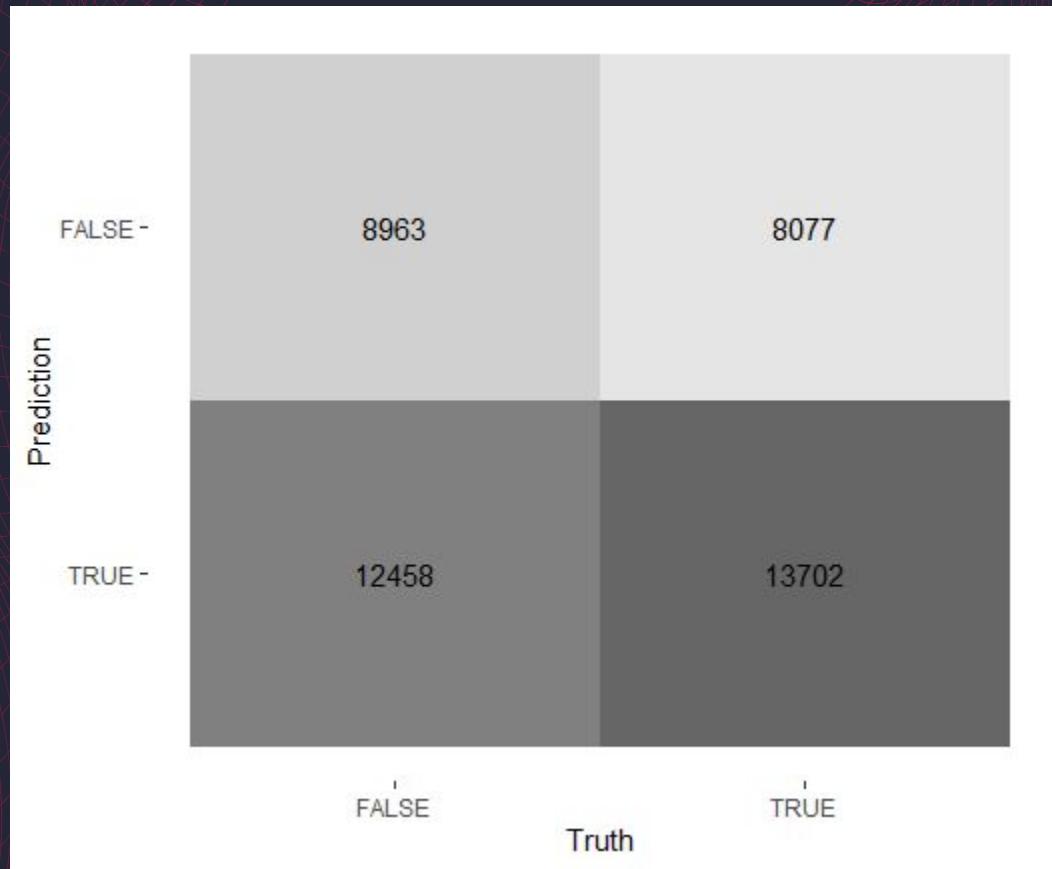
```
results_train <- data.frame(  
  `truth` = ETH_BTC_USDT11month %>% select(BTCBuy),  
  `Class1` = BTCBuy_train_pred,  
  `type` = rep("train", length(BTCBuy_train_pred))  
)  
  
results_test <- data.frame(  
  `truth` = ETH_BTC_USDT %>% select(BTCBuy),  
  `Class1` = BTCBuy_test_pred,  
  `type` = rep("test", length(BTCBuy_test_pred))  
)  
  
results <- bind_rows(results_train, results_test)  
  
results_logit <- data.frame(  
  `truth` = ETH_BTC_USDT11month$BTCBuy,  
  `Class1` = BTCBuy_train_pred,  
  `Class2` = 1 - BTCBuy_train_pred,  
  `predicted` = as.factor(ifelse(BTCBuy_train_pred > 0.5,  
                                 "TRUE", "FALSE"))  
)  
  
cm <- conf_mat(results_logit,  
                 truth = truth,  
                 estimate = predicted)  
  
print(cm)  
autoplot(cm, "heatmap")
```

Logistic: BTC Confusion Matrix w/ threshold of 0.9



- More conservative threshold
- Better at predicting when not to Buy
- AUC score is still poor

Logistic: BTC Test Confusion Matrix



Logistic: BTC ROC Plot



```
p <- ggplot(results_logit,
  aes(m = Class1, d = truth)) +
  geom_roc(labelsize = 3.5,
  cutoffs.at =
    c(0.9,0.7,0.5,0.3,0)) +
  theme_minimal(base_size = 14)

print(p)
calc_auc(p)

roc_auc(results_logit,
  truth = truth,
  estimate = predicted)
```

| PANEL
<fctr> | group
<int> | AUC
<dbl> |
|-----------------|----------------|--------------|
| 1 | -1 | 0.5984277 |

Logistic: ETH Model

```
logit_fit_train2 <- glm(ETHBuy ~ BTC_Open + BTC_High + BTC_Low  
+ BTC_Close + BTC_Volume + BTC_QuoteAssetVolume  
+ BTC_NumberOfTrades  
+ BTC_TalkerBuyBaseAssetVolume  
+ BTC_TalkerBuyQuoteAssetVolume,  
family = binomial,  
data = ETH_BTC_USDT11month)
```

Unexponentiated

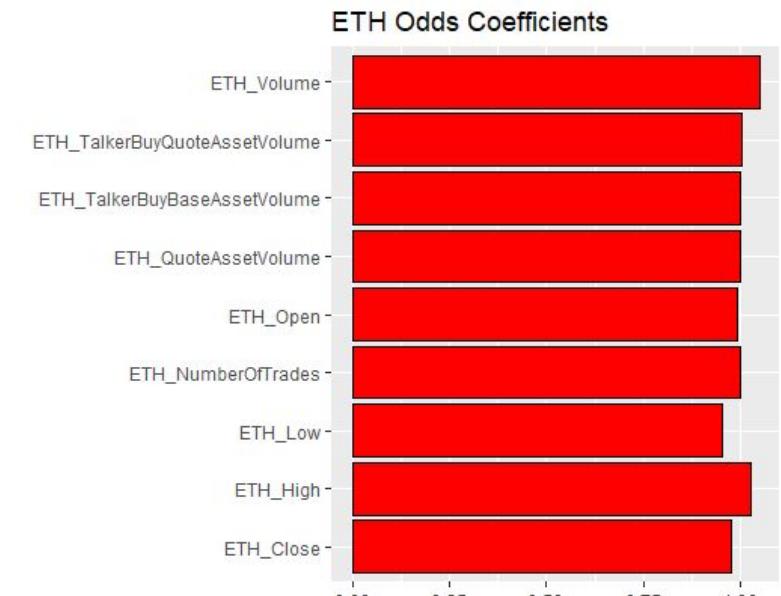
```
Deviance Residuals:  
Min 1Q Median 3Q Max  
-3.3557 -1.1644 -0.9681 1.1871 3.4945  
  
Coefficients:  
Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.058072398386 0.008069928921 -7.196 0.000000000000619 ***  
BTC_Open 0.003076653563 0.000124858477 24.641 < 2e-16 ***  
BTC_High -0.002653871939 0.000140607902 -18.874 < 2e-16 ***  
BTC_Low -0.001373104357 0.000143627872 -9.560 < 2e-16 ***  
BTC_Close 0.000952782301 0.000133319363 7.147 0.000000000000889 ***  
BTC_Volume -0.000198500216 0.000298735319 -0.664 0.506  
BTC_QuoteAssetVolume -0.000000164108 0.000000009388 -17.480 < 2e-16 ***  
BTC_NumberOfTrades -0.000011552409 0.000007976316 -1.448 0.148  
BTC_TalkerBuyBaseAssetVolume 0.000465575471 0.000545438609 0.854 0.393  
BTC_TalkerBuyQuoteAssetVolume 0.000000342080 0.000000017196 19.893 < 2e-16 ***  
---  
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 668703 on 482399 degrees of freedom  
Residual deviance: 666240 on 482390 degrees of freedom  
AIC: 666260
```

Number of Fisher Scoring iterations: 3

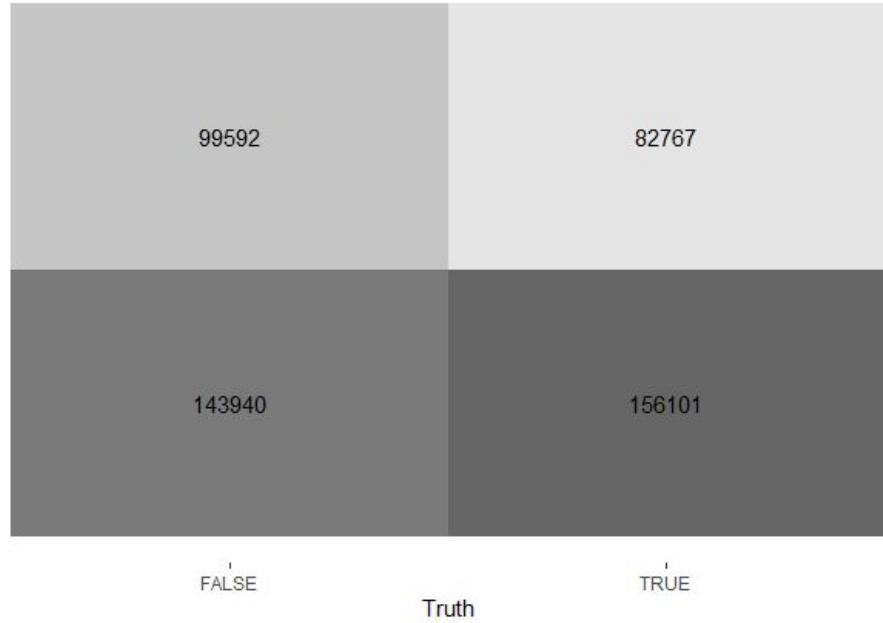
Exponentiated



```
plot2 <- ggplot(coeffs2, aes(x = Coefficient.values, y = conditional))  
+ geom_bar(stat = "identity", color = "black", fill = "red")  
+ ggtitle("ETH Odds Coefficients")  
+ labs(x = "", y = "")  
  
print(plot2)
```

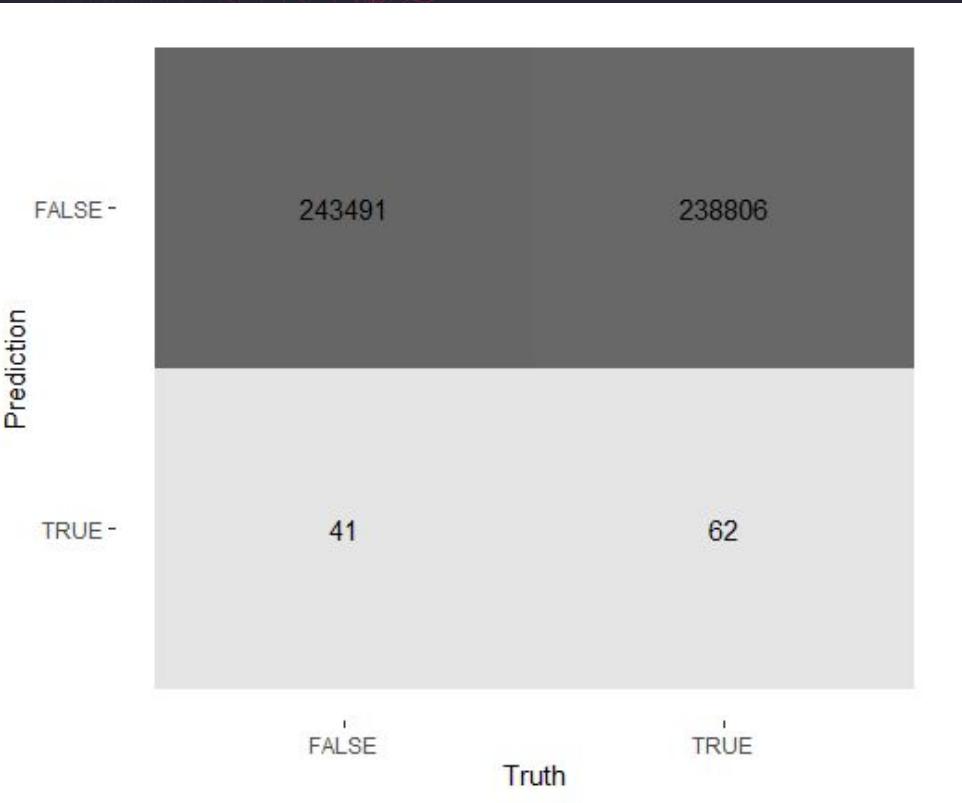
Logistic: ETH Confusion Matrix

Prediction



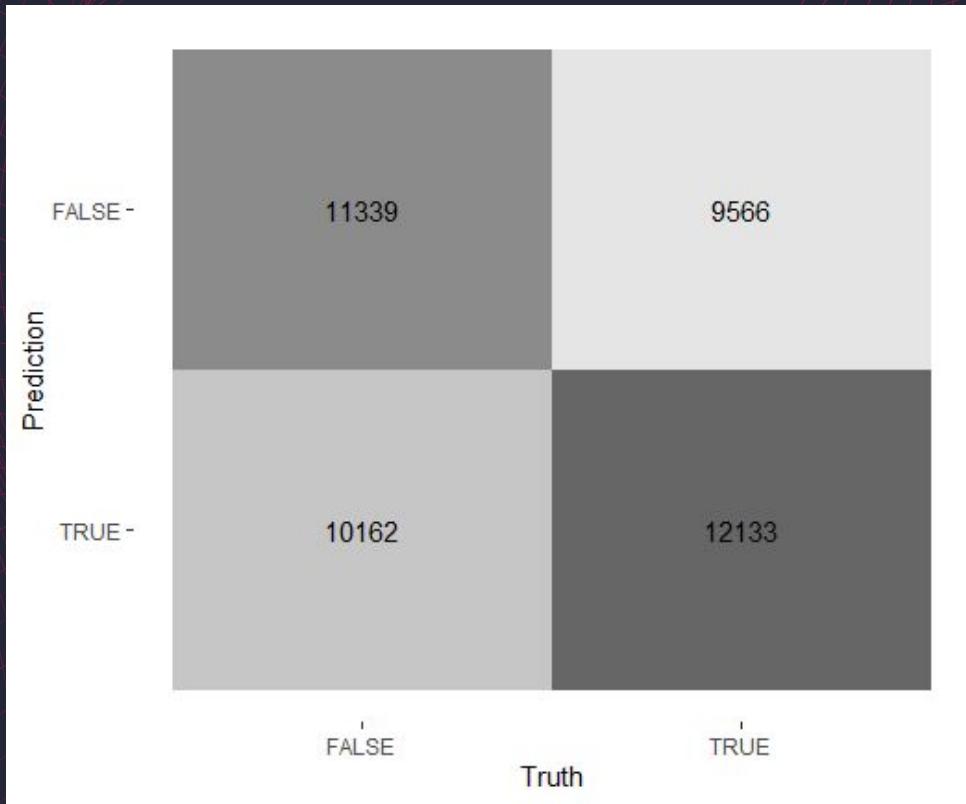
```
results_train2 <- data.frame(  
  `truth` = ETH_BTC_USDT1month %>% select(ETHBuy),  
  `Class1` = ETHBuy_train_pred,  
  `type` = rep("train",length(ETHBuy_train_pred))  
)  
  
results_test2 <- data.frame(  
  `truth` = ETH_BTC_USDT %>% select(ETHBuy),  
  `Class1` = ETHBuy_test_pred,  
  `type` = rep("test",length(ETHBuy_test_pred))  
)  
  
results2 <- bind_rows(results_train2, results_test2)  
  
results_logit2 <- data.frame(  
  `truth` = ETH_BTC_USDT1month$ETHBuy,  
  `Class1` = ETHBuy_train_pred,  
  `Class2` = 1 - ETHBuy_train_pred,  
  `predicted` = as.factor(ifelse(ETHBuy_train_pred > 0.49,  
    "TRUE", "FALSE"))  
)  
  
cm2 <- conf_mat(results_logit2,  
  truth = truth,  
  estimate = predicted)  
  
print(cm2)|  
autoplot(cm2, "heatmap")
```

Logistic: ETH Confusion Matrix threshold of 0.9

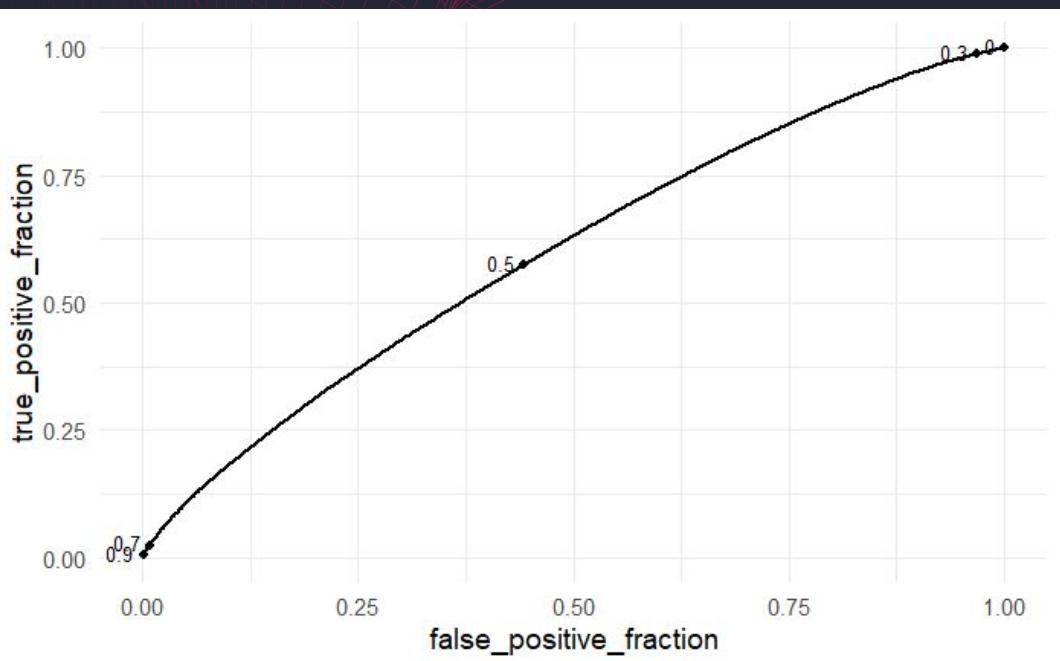


- Not a powerful threshold
- Small opportunities for true positives
- AUC Score very poor

Logistic: ETH Test Confusion Matrix



Logistic: ETH ROC Plot



```
p2 <- ggplot(results_logit2,  
               aes(m = Class1, d = truth)) +  
  geom_roc(labelsize = 3.5,  
            cutoffs.at =  
              c(0.99,0.9,0.7,0.5,0.3,0.1,0)) +  
  theme_minimal(base_size = 14)  
print(p2)  
calc_auc(p2)  
  
roc_auc(results_logit2,  
        truth = truth,  
        estimate = predicted)
```

| PANEL
<fctr> | group
<int> | AUC
<dbl> |
|-----------------|----------------|--------------|
| 1 | -1 | 0.548675 |

Logistic: Conclusion

Model is not helpful

Too many false positives and true negatives

Trading Algorithm would do poor job

Low AUC scores

Model Comparisons

1. Linear Regression <- Useful
2. Elastic Net <- Useful
3. Logistic Regression <- Not Useful

But none are anything close to ready to apply to a commercial scale for current Crypto numbers.

Data Link: <https://www.kaggle.com/jorijnsmit/binance-full-history/discussion/198669>

THANKS!

Test Our Code!!!



RatFuryJunior/MGSC_310_Binance_Analysis

https://github.com/RatFuryJunior/MGSC_310_Binance_Analysis

Questions??

MGSC: 310 Ryan Welte, Nathan Cuadros, and Ethan Leimel.