ML On-Chain Analysis for BTC

Jackie You, Jamal Rizvi, and Clare Collity

Agenda

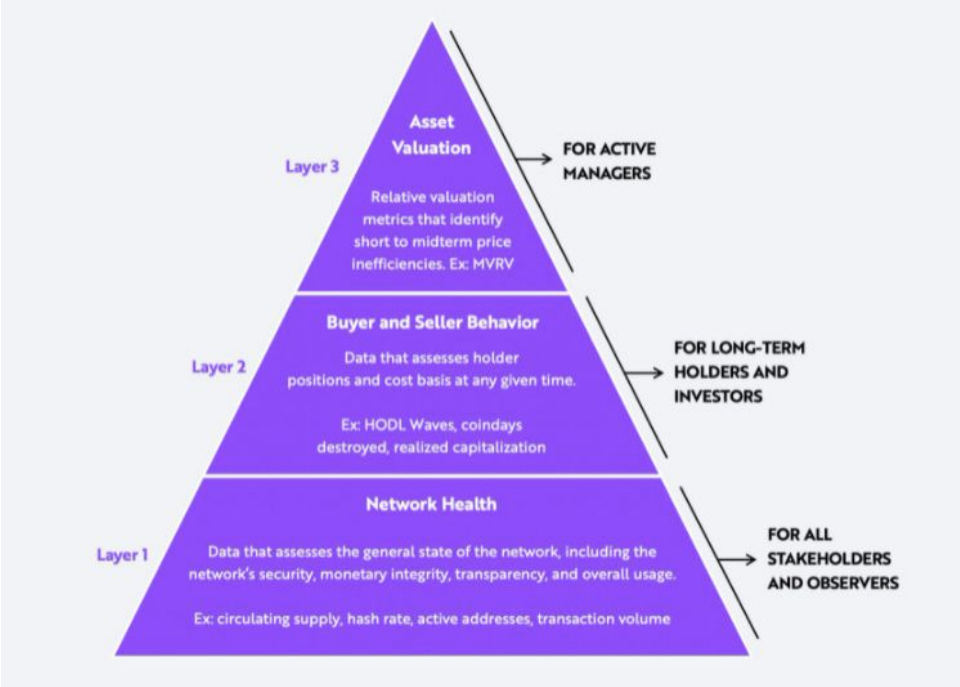
- Overview of project and goals
- Our machine learning models
- Data collection and training process
- Examples and predictions
- Conclusion
- What’s next?

Project Overview + Goals

- Use machine learning on-chain analysis to predict BitCoin prices. By doing so, we attempt to identify the key drivers of the prices.
- Understand more in-depth three different types of ML models.
- Provide insights to drive smarter investment decisions

Framework of on-chain analysis

- Multiple layers used to predict and assess BTC health
- The bottom layer is the straightforward data (usage, monetary integrity)
- Middle layer shows cost bases at any time during the day, dives into short term behavior of buyers and sellers
- Top layer has valuation metrics that managers use to find inefficiencies in bitcoin prices



https://ark-invest.com/articles/analyst-research/bitcoin-buyer-and-seller/

Our Project + the Pyramid of BTC

- Goal was to create application with Machine Learning that acknowledges all aspects of the bitcoin evaluation framework
- Three levels: Network Health, Buyer and Seller Behavior, Valuation
- Goal is to incorporate all three levels of framework into application to predict future prices so that everything is considered
- Using machine learning models to accomplish this

Fear and Greed Index

- Part of valuation, top level of bitcoin on-chain analysis framework
- Analyzes current bitcoin market to determine level of emotional impact
- Pulls data from sources like social media and Google Trends and considers factors like volatility and market momentum

Coding our Own Fear and Greed Index

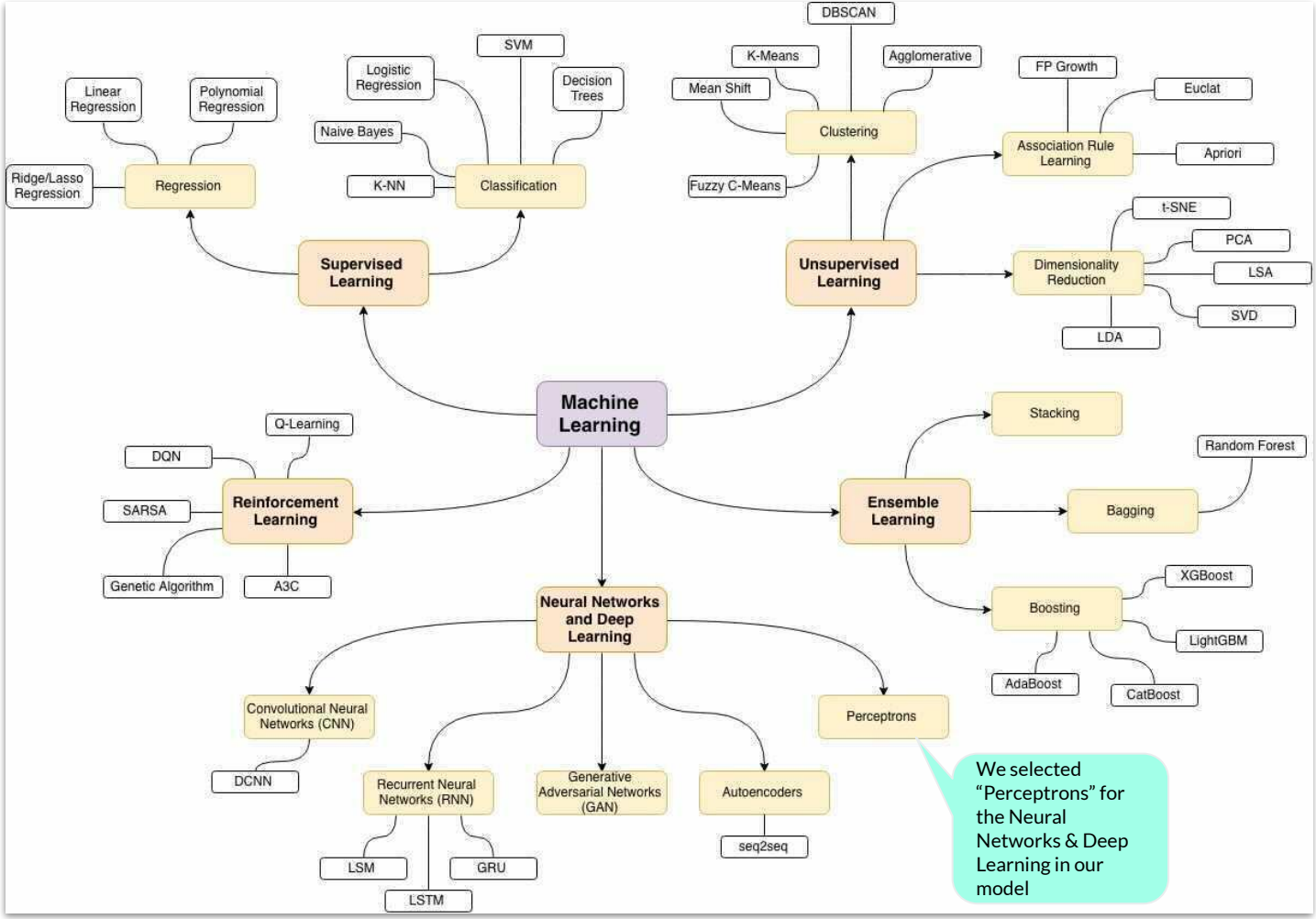
- Retrieved current Fear and Greed Index Crypto API
- Created bitcoin dataframe and gauge to show results
- Extreme fear: investors are very worried, possible buying opportunity
- Extreme greed: market may be due for a correction

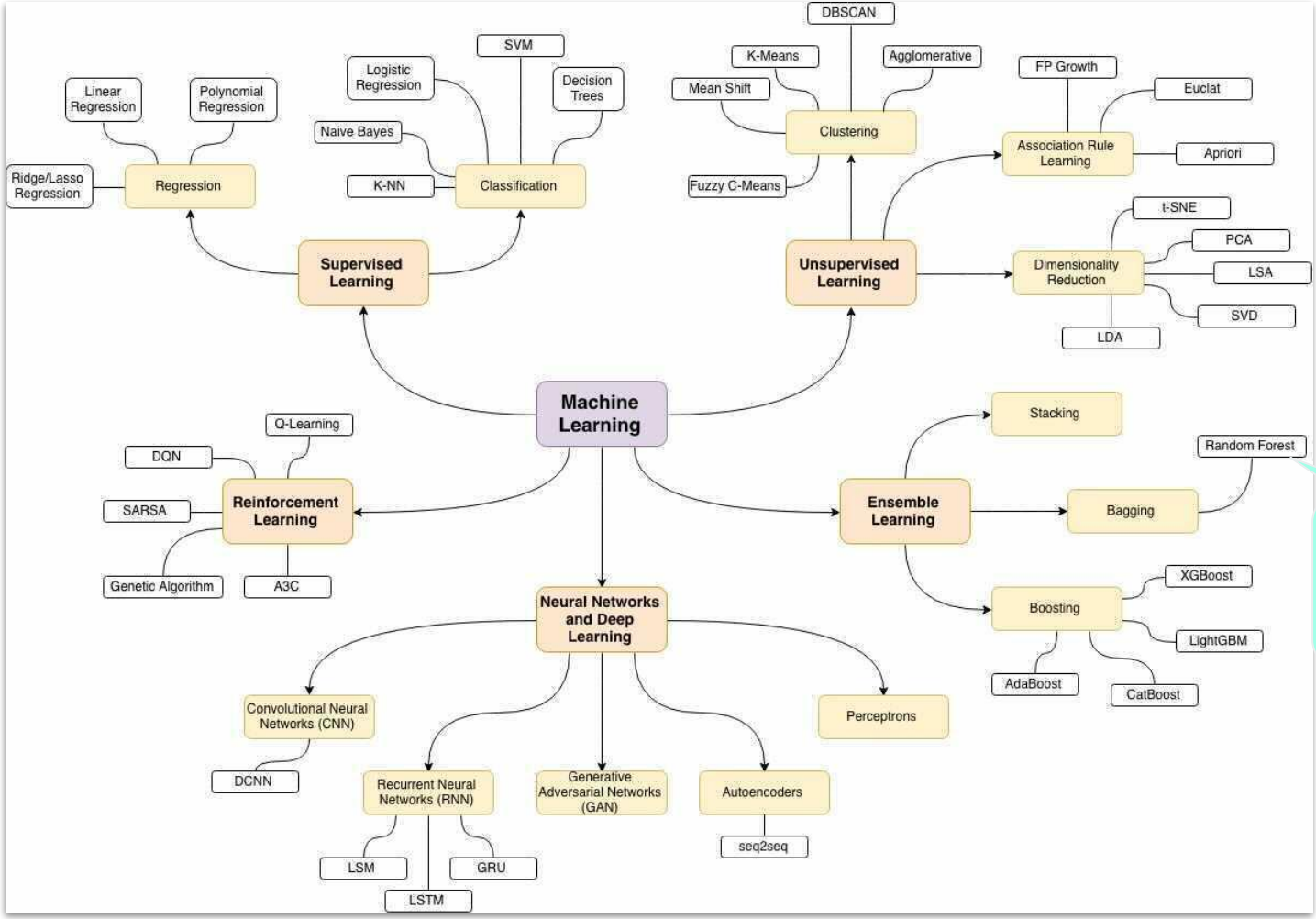
Machine Learning Models

Our Machine Learning Models

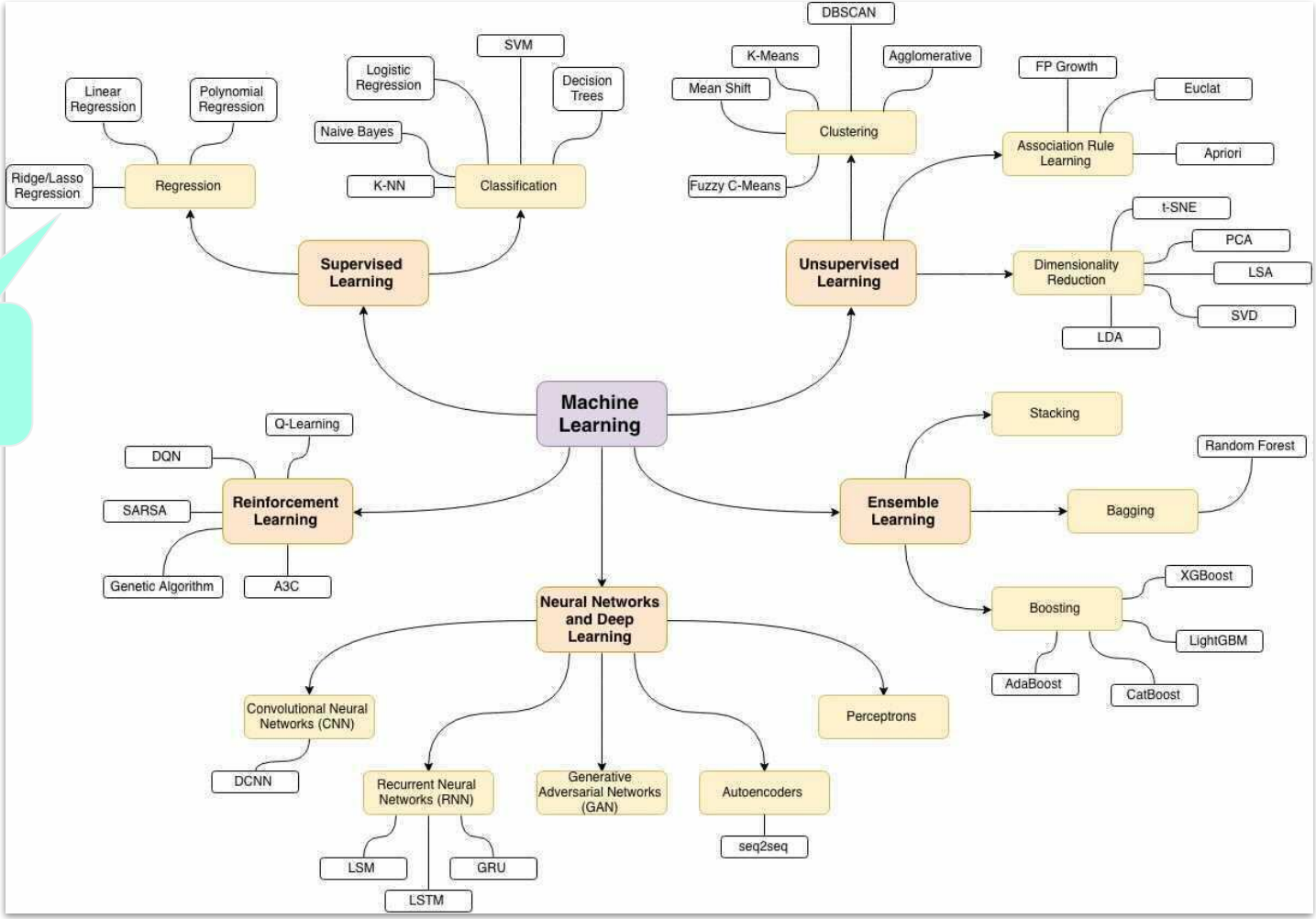
Used three machine learning models:

- Neural Network & Deep Learning: Perceptrons
- Ensemble Learning: Random Forest
- Supervised Learning: Lasso Regression


Machine Learning
Supervised Learning
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Classification
Logistic Regression
SVM
Decision Trees
Naive Bayes
K-NN
Unsupervised Learning
Clustering
K-Means
DBSCAN
Agglomerative
Mean Shift
Fuzzy C-Means
Association Rule Learning
FP Growth
Euclat
Apriori
Dimensionality Reduction
t-SNE
PCA
LSA
SVD
LDA
Reinforcement Learning
Q-Learning
DQN
SARSA
Genetic Algorithm
A3C
Ensemble Learning
Stacking
Bagging
Random Forest
Boosting
XGBoost
LightGBM
AdaBoost
CatBoost
Neural Networks and Deep Learning
Convolutional Neural Networks (CNN)
DCNN
Recurrent Neural Networks (RNN)
LSM
GRU
LSTM
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq
Perceptrons
We selected “Perceptrons” for the Neural Networks & Deep Learning in our model


Machine Learning
Supervised Learning
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Classification
Logistic Regression
SVM
Decision Trees
Naive Bayes
K-NN
Unsupervised Learning
Clustering
K-Means
DBSCAN
Agglomerative
Mean Shift
Fuzzy C-Means
Association Rule Learning
FP Growth
Euclat
Apriori
Dimensionality Reduction
t-SNE
PCA
LSA
SVD
LDA
Reinforcement Learning
Q-Learning
DQN
SARSA
Genetic Algorithm
A3C
Neural Networks and Deep Learning
Convolutional Neural Networks (CNN)
DCNN
Recurrent Neural Networks (RNN)
LSM
GRU
LSTM
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq
Perceptrons
Ensemble Learning
Stacking
Bagging
Random Forest
Boosting
XGBoost
LightGBM
AdaBoost
CatBoost
We selected “Random Forest” for the Ensemble Learning in our model

We selected “Lasso Regression” for the Ensemble Learning in our model



Process

Phases

One: Build X dataframe for features

Two: Build Machine Learning Models

Three: Use FB Prophet to predict future values of each feature in X_scaled.

Four: Apply ML Models

Key Decisions

To predict Prices or Returns?

To let Lasso Regressor or use correlations to select features?

To use Classifier Train_Test_Split or TimeSeriesSplit?

Phase One -

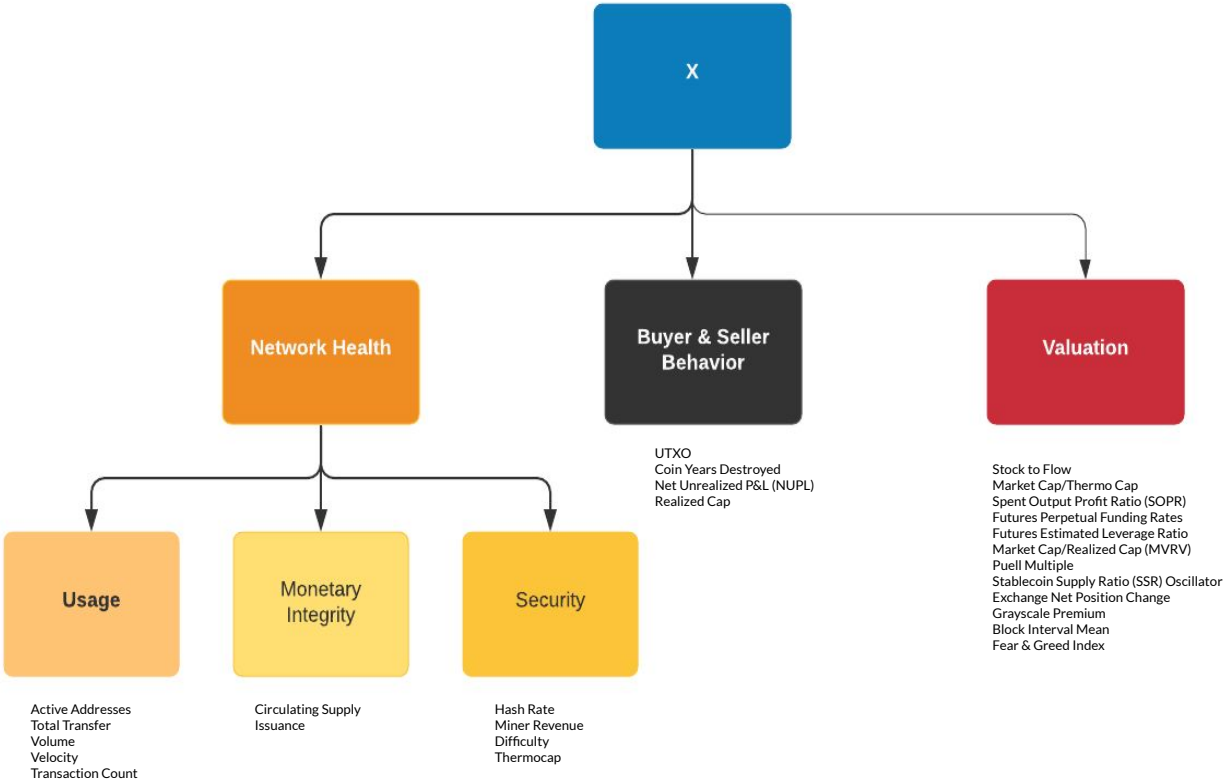
Build X dataframe

Establish Data Frames for 27 Features

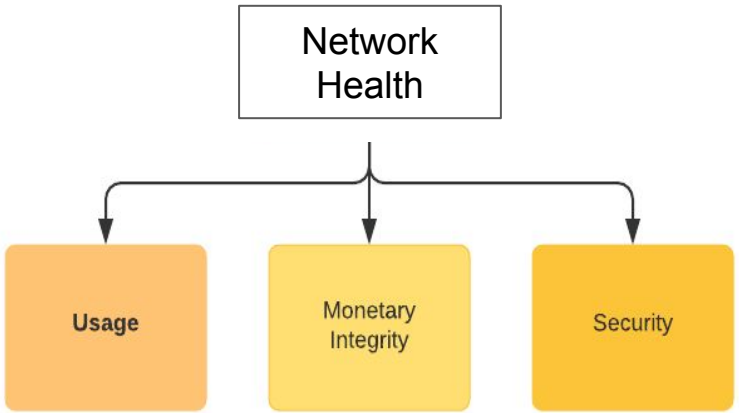
From :

Glassnode.com

alternative.me



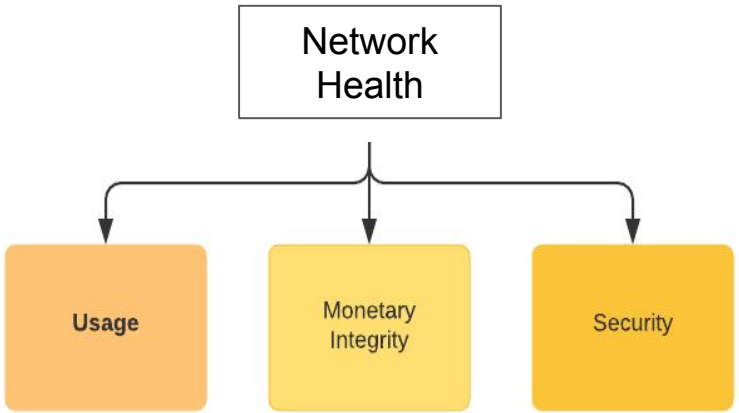
Breaking Down the Diagram



Usage:

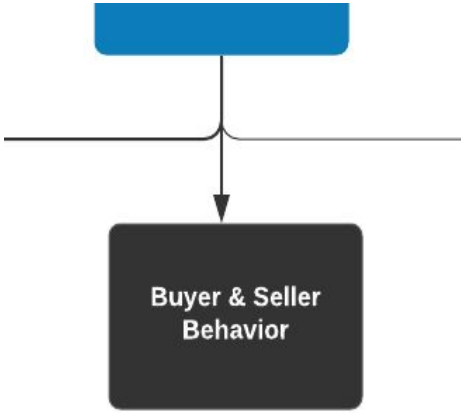
- Active address
  - Total active address: gn/addresses/active addresses
  - Address supply distribution: gn/addresses/address supply distribution
  - Total value stacked/locked
- Transactions
  - Gas costs
  - Transaction volume: gn/network stats/total transfer volume
  - Velocity: gn/market indicators/velocity
  - Transaction count: gn/network stats/number of transactions

Breaking Down the Diagram



- Monetary Integrity:
  - Circulating supply: gn/network stats/~
  - Inflation target
  - Issuance: gn/miners/~
- Security:
  - Hash rate: gn/miners/mean hash rate (miners' investment)
  - Miner revenue: gn/miners/Total miner revenue

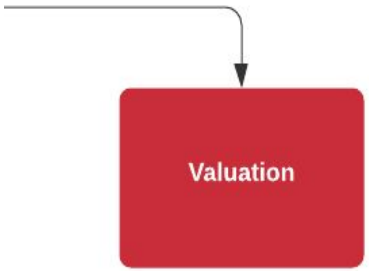
Breaking Down the Diagram (cont.)



Buyer and Seller Behavior:

- UTXO
  - Spent Output Age bands: watch old coins behavior with price actions
  - Spent output lifespan: gn/lifespan/~
- Coin years destroyed:
  - glassnode/lifespan/CDD or CYD (volume x holding period, not $)
  - glassnode/entities/entity-adj coin years destroyed
- On Chain P&L
  - Unrealized gain/liquidity: GN/market indicators/unrealized profit, GN/”profit/loss”/”supply in profit/loss”

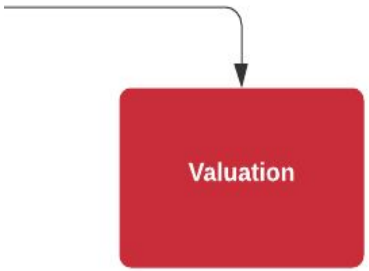
Breaking Down the Diagram (cont.)



Valuation:

- Short Term:
  - Sentiment-Fear & greed
  - Miners & mining pools: selling pressure tracks newly issued coins.
  - Short term holder SOPR
  - Leverages
    - Leverage level
    - Cost of leverage:
      - Into the block: borrow rates/lending rates
      - The block: BTC/ETH funding rates
  - MVRV: gn/market indicators/mvrv (short term)
  - Purell multiple: gn/market indicators/~ (daily issuance/365 day average)

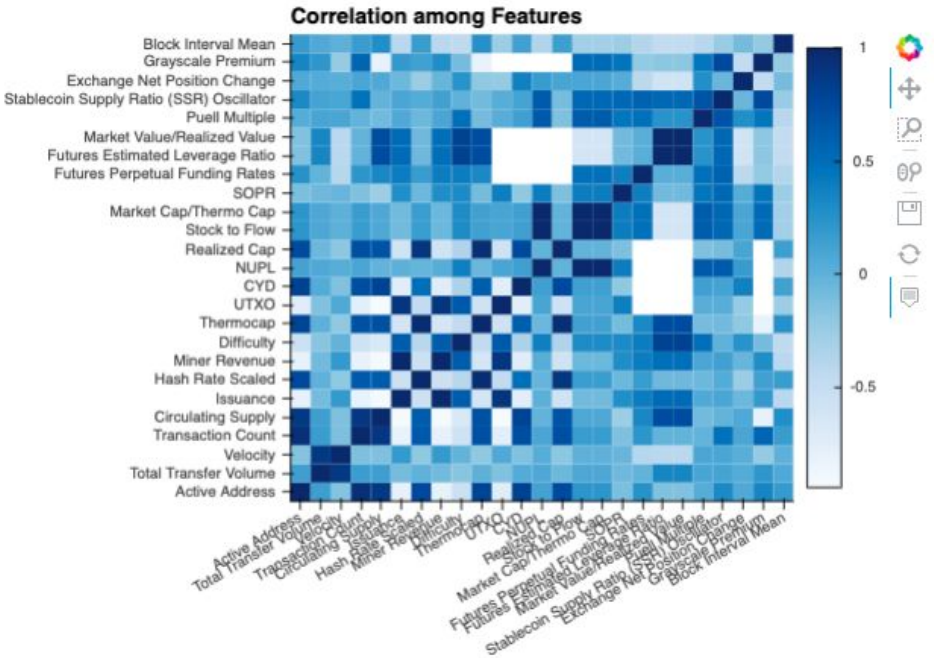
Breaking Down the Diagram (cont.)



- Short Term cont.
  - Stable coin supply ratio: gn/ Stablecoin supply ratio oscillator: ratio of bitcoin supply and the stablecoin supply denoted in BTC. When ratio is low, buying power for bitcoin by USD is high. When the ratio increases, capital flows into BTC.
  - Exchange inflow volume – outflow volume
  - GBTC premium/discount
  - Mean block interval
- Long term
  - Stock to flow: GM/market indicators/~
  - Inflation target

Correlation and Cleaning

- Measure correlation among features in each data frame block
- Reduce dimensions by deleting highly correlated features
- Further reduce dimensions by deleting features that do not have sufficient history

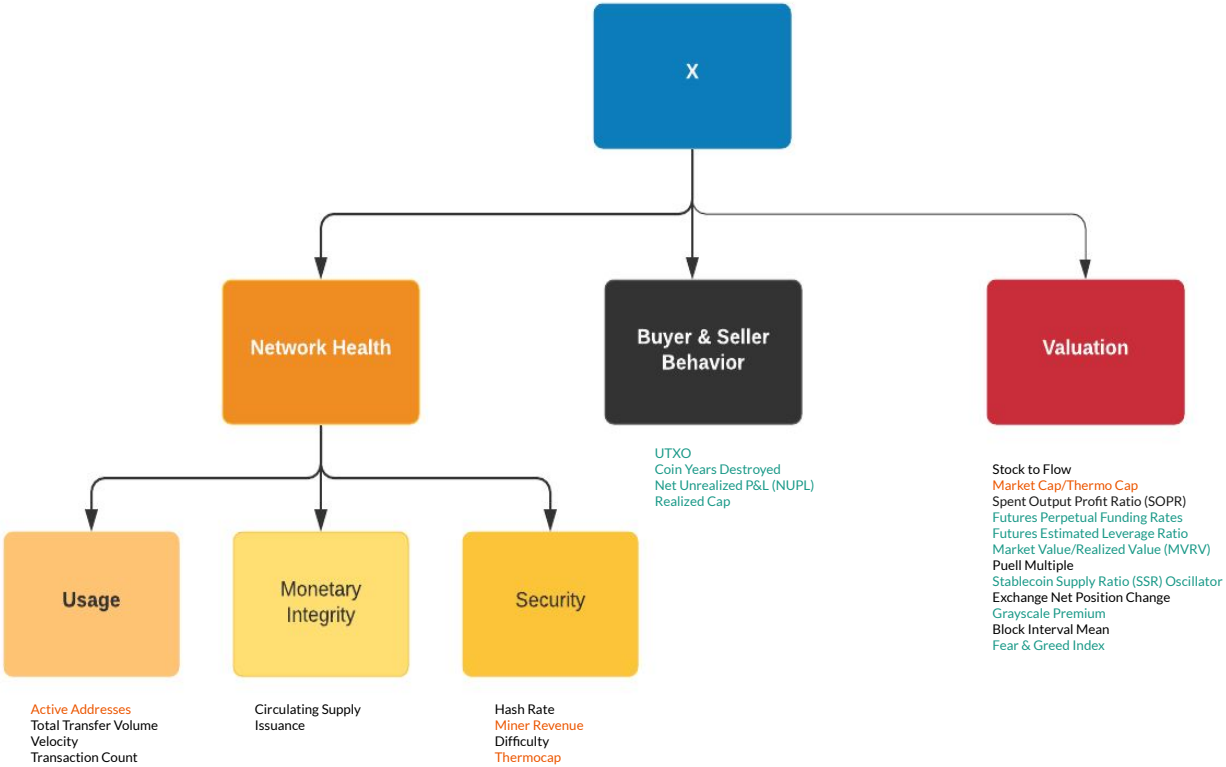


12 Features Remained With data since 2011

Deletion due to :

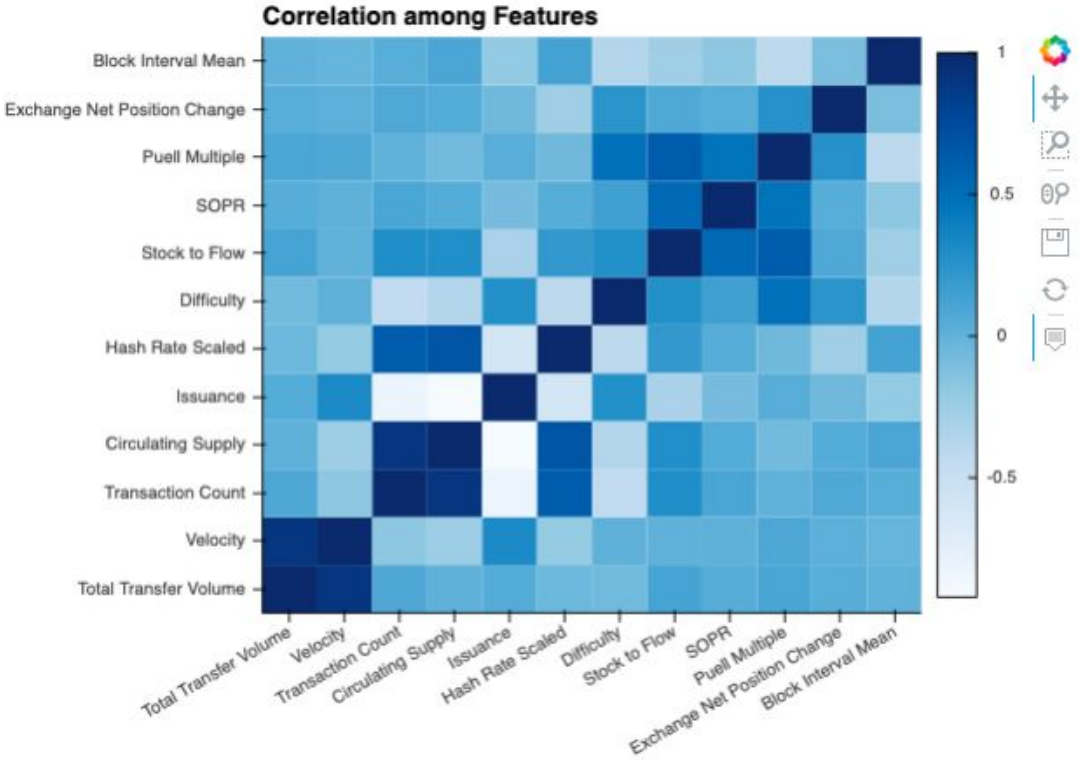
Correlation

Insufficient history



Key Decision:

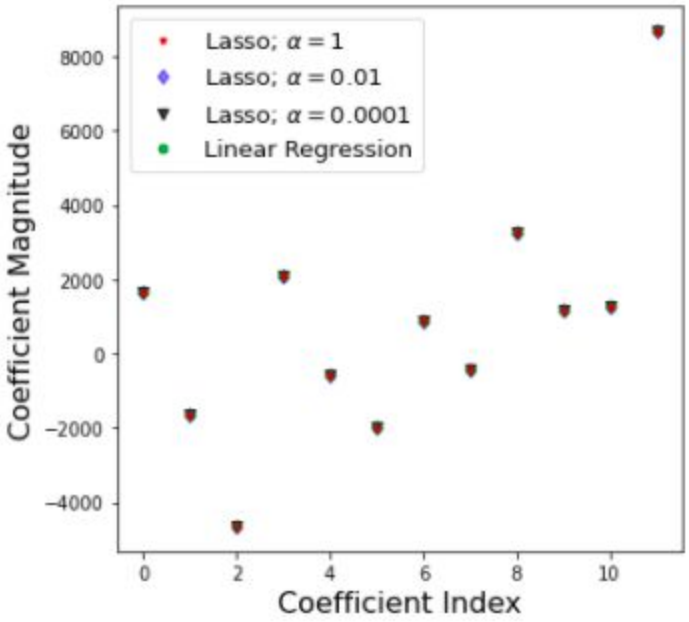
- Select features based on correlations



Phase Two -

Build Machine Learning Models

Lasso Regressor




```
training score: 0.7361097735922011
test score:  0.7125637021074427
number of features used:  12
training score for alpha=0.01: 0.7361104050016812
test score for alpha =0.01:  0.7125485871800203
number of features used: for alpha =0.01: 12
training score for alpha=0.0001: 0.7361104050665593
test score for alpha =0.0001:  0.7125484272712526
number of features used: for alpha =0.0001: 12
LR training score: 0.7361104050665659
LR test score:  0.7125484256964185
```

	lr_prediction	lasso001_predition	lasso0001_predition	Actual Price
2013-07-11	-1023.950418	-1023.911921	-1023.950036	87.634667
2021-02-06	32348.898616	32348.822287	32348.897846	39271.996768
2011-12-18	-4436.481014	-4436.405988	-4436.480276	3.192505
2020-10-30	21100.256446	21100.178095	21100.255659	13567.751090
2012-09-01	96.693020	96.796410	96.694069	9.976960

Random Forrest Regressor

```
print(mse_rf)
print(rmse_rf)
```

```
345387.42652002675
587.6967130417072
```

```
r2_test = model_rf.score(X_test_scaled, y_test)
r2_train = model_rf.score(X_train_scaled, y_train)
print(r2_test, r2_train)
```

```
0.9968800242256421 0.9980215997867252
```

	Random Forrest	Actual Price
2013-07-11	86.943019	87.634667
2021-02-06	40794.201051	39271.996768
2011-12-18	3.211293	3.192505
2020-10-30	13530.827289	13567.751090
2012-09-01	10.245629	9.976960
...	...	...
2011-10-14	3.957921	3.987940
2016-10-09	613.093147	615.650775
2011-09-28	4.744417	4.709347
2016-11-28	728.775733	729.913159
2015-05-22	238.757770	239.875307

359 rows × 2 columns

Perceptrons

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 6)                 78
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 21
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 8
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 3
=================================================================
Total params: 110
Trainable params: 110
Non-trainable params: 0
_________________________________________________________________
```

```
# Evaluate the model using testing data
model_loss, model_mse = nn.evaluate(X_test_scaled, y_test, verbose=2)
```

```
12/12 - 0s - loss: 20829890.0000 - mse: 20829890.0000
```

```
rmse_perceptrons = np.sqrt(model_mse)
print(rmse_perceptrons)
```

```
4563.977432021328
```

	Perceptrons	Actual Price
2013-07-11	-0.662277	87.634667
2021-02-06	36573.039062	39271.996768
2011-12-18	-0.662277	3.192505
2020-10-30	23347.810547	13567.751090
2012-09-01	-0.662277	9.976960
...	...	...
2011-10-14	-0.662277	3.987940
2016-10-09	-0.662277	615.650775
2011-09-28	-0.662277	4.709347
2016-11-28	-0.662277	729.913159
2015-05-22	-0.662277	239.875307

359 rows × 2 columns

Key Decisions:

Based on model performance results

- Predict prices instead of returns;
- Use train_test_split instead of TimeSeries Split

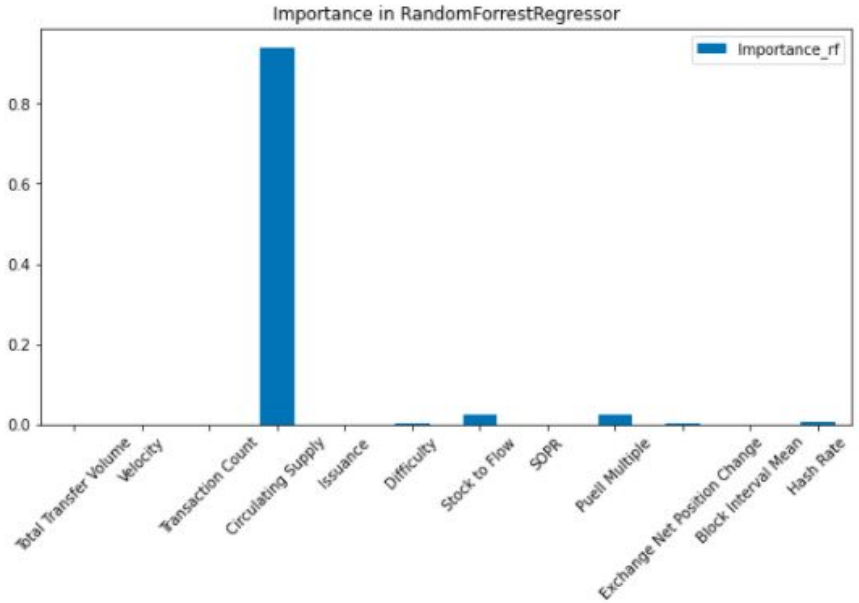
t	Predicted Price by Random Forrest	Actual Price
2012-09-17	11.431827	11.897082
2012-09-18	11.413294	12.294733
2012-09-19	11.275946	12.521990
2012-09-20	11.369711	12.337124
2012-09-21	11.554095	12.344473
...	...	...
2013-09-05	12.089775	122.619851
2013-09-06	12.089775	117.215590
2013-09-07	12.086780	119.922000
2013-09-08	12.083639	118.340000
2013-09-09	12.118092	121.243896

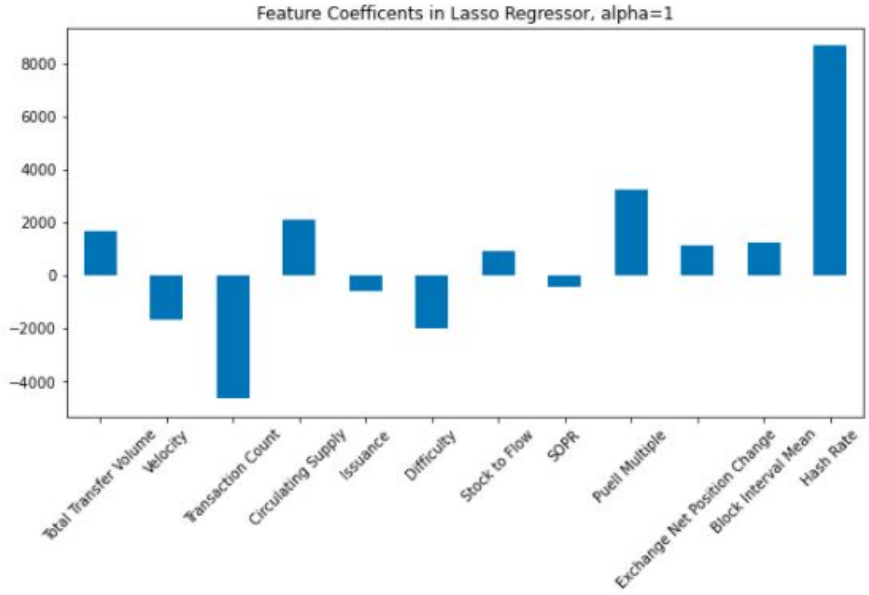
358 rows × 2 columns

```
BTC Price by Lasso     7.040456
dtype: float64
BTC Price by Lasso     21.239683
dtype: float64
```

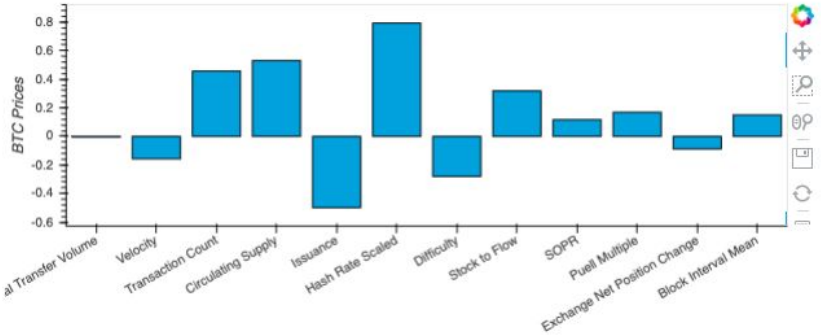
RandomForrest Regressor

Feature Importance/Coefficient

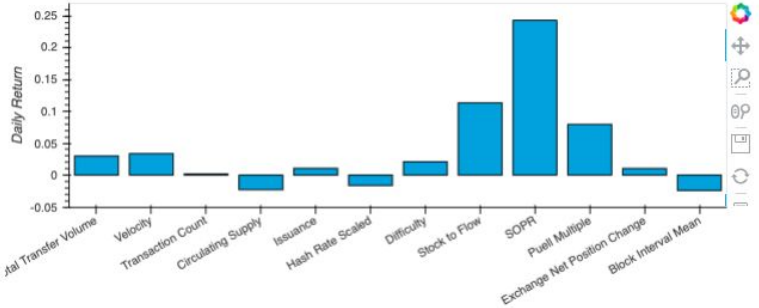




Correlation between X & y



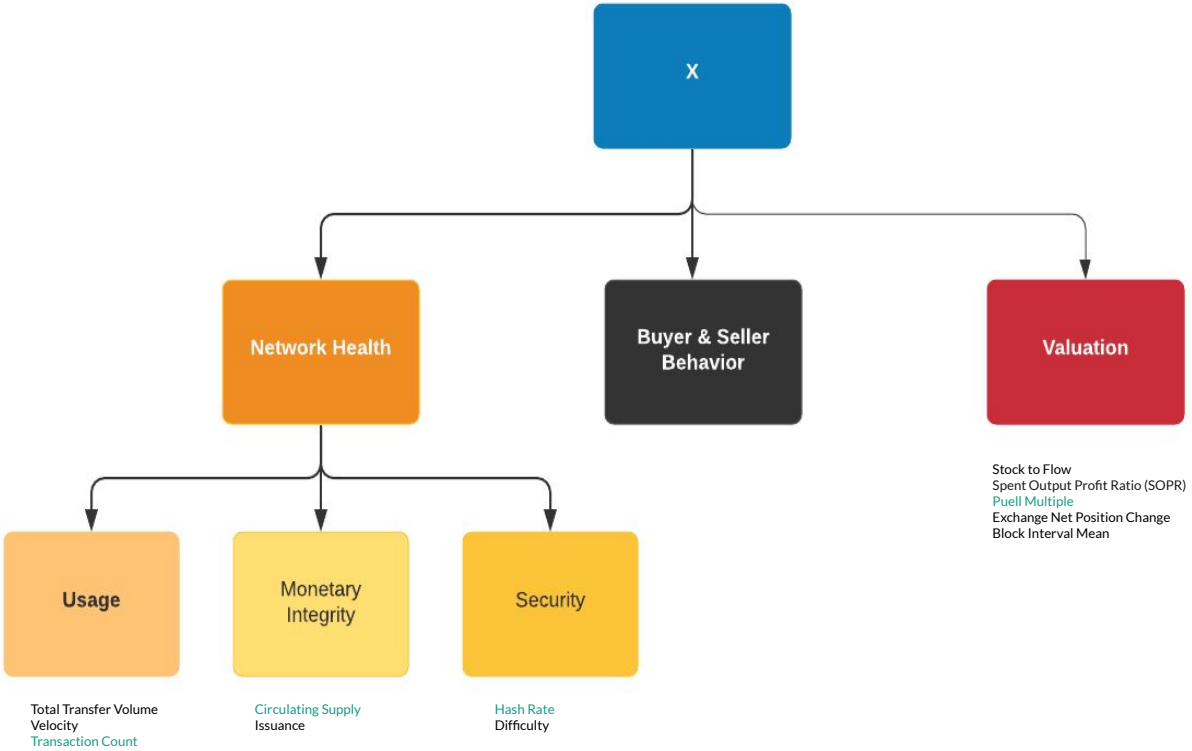
with Price



with Daily Return

Key Features

- Circulating Supply
- Hash Rate
- Transaction Count



Correlations among Key Features

	Circulating Supply	Transaction Count	Hash Rate Scaled
Circulating Supply	1.000000	0.992541	0.923430
Transaction Count	0.992541	1.000000	0.912242
Hash Rate Scaled	0.923430	0.912242	1.000000

High correlation among the top three levels were not this high in the vetting process.

Definitions

Circulating Supply	The total amount of all coins ever created/issued, i.e. the circulating supply.
Hash Rate	The average estimated number of hashes per second produced by the miners in the network.
Transaction Count	The total amount of transactions. Only successful transactions are counted.

Phase Three -

FB Prophet

Predictions on Features

ds	index	Total Transfer Volume	Velocity	Transaction Count	Circulating Supply	Issuance	Difficulty	Stock to Flow	SOPR	Puell Multiple	Exchange Net Position Change	Block Interval Mean	Hash Rate
Lower	2021-07-21 00:00:00	-1.10644	-1.61654	0.409763	1.24878	-1.71161	-1.26388	0.603051	-0.751395	-0.783732	-2.54638	-0.568605	2.44
Most Likely	2021-07-21 00:00:00	0.100085	-0.36461	0.743303	1.26067	-1.3113	-0.886085	1.00865	0.406343	0.275469	-1.19866	0.668783	2.668
Upper	2021-07-21 00:00:00	1.19472	0.644437	1.04145	1.2667	-1.00019	-0.540826	1.74353	1.58997	0.896134	-0.370223	1.76833	2.87395

532441 Combinations of Feature possibilities

	Total Transfer Volume	Velocity	Transaction Count	Circulating Supply	Issuance	Difficulty	Stock to Flow	SOPR	Puell Multiple	Exchange Net Position Change	Block Interval Mean	Hash Rate
0	-1.106445	-1.616540	0.409763	1.248785	-1.711606	-1.263884	0.603051	-0.751395	-0.783732	-2.546379	-0.568605	2.439997
1	-1.106445	-1.616540	0.409763	1.248785	-1.711606	-1.263884	0.603051	-0.751395	-0.783732	-2.546379	-0.568605	2.668002
2	-1.106445	-1.616540	0.409763	1.248785	-1.711606	-1.263884	0.603051	-0.751395	-0.783732	-2.546379	-0.568605	2.873947
3	-1.106445	-1.616540	0.409763	1.248785	-1.711606	-1.263884	0.603051	-0.751395	-0.783732	-2.546379	0.668783	2.439997
4	-1.106445	-1.616540	0.409763	1.248785	-1.711606	-1.263884	0.603051	-0.751395	-0.783732	-2.546379	0.668783	2.668002
...	...	...	...	...	...	...	...	...	...	...	...	...
531436	1.194717	0.644437	1.041447	1.266705	-1.000190	-0.540826	1.743525	1.589968	0.896134	-0.370223	0.668783	2.668002
531437	1.194717	0.644437	1.041447	1.266705	-1.000190	-0.540826	1.743525	1.589968	0.896134	-0.370223	0.668783	2.873947
531438	1.194717	0.644437	1.041447	1.266705	-1.000190	-0.540826	1.743525	1.589968	0.896134	-0.370223	1.768331	2.439997
531439	1.194717	0.644437	1.041447	1.266705	-1.000190	-0.540826	1.743525	1.589968	0.896134	-0.370223	1.768331	2.668002
531440	1.194717	0.644437	1.041447	1.266705	-1.000190	-0.540826	1.743525	1.589968	0.896134	-0.370223	1.768331	2.873947

531441 rows × 12 columns

Phase Four -

Apply ML Models

Prediction Result Examples

	Date	Lower	Most Likely	Upper
Lasso	2021-07-21	16981	32366	46132
Random Forrest	2021-07-21	37168	48889	55870
Perceptrons	2021-07-21	11505	35458	59483

	Date	Lower	Most Likely	Upper
Lasso	2022-01-10	22795	37701	53938
Random Forrest	2022-01-10	44328	51452	60158
Perceptrons	2022-01-10	17570	40267	65868

Learnings

Unexpected Insights

Dropped feature candidates due to high correlation:

Dropped	Correlation	With
Active Addresses	94.3%	Transaction Count
Miner Revenue	99.9%	Issuance
Thermocap	96.9%	Hash Rate
	95.1%	Realized Cap
NUPL	100%	Stock to Flow
	100%	MarketCap/ThermoCap
MarketCap/ThermoCap	100%	Stock to Flow

Correlations among two features are relative!

Correlations change based on how many features are considered.

Lasso Regressor only kept Hash Rate among 27 features!

Challenges

Unanticipated problems that arose and how they were resolved:

- Lasso model singled out Hash Rate as the only feature upon first try but model error spiked.
- TimeSeriesSplit cause model errors to spike and predicted values were extremely low.
- When applying Daily Return as y, predictive abilities worsened

Choices were made to facilitate findings at this stage. Need further exploration.

Challenges

Unanticipated problems that arose and how they were resolved:

- Hash Rate:
  - Large number, 32 digits. Use json.loads()
  - Interferes with read_json. Patch it.
  - Hash_rate_df wouldn’t work in code. Scale it.
  - In scaling, convert list of dict to list of lists.
  - In scaling X, not to scale scaled data.
- Itertools
- Data manipulation (convert Numpy array to a Tensor, DateTime conversion, dataframes)

Findings

- On BTC:
  - Key drivers of BTC prices are activity levels, indicated by the top three features.
  - Better understand the correlations among features.
- On ML models:
  - Random Forrest had the best test results, and future projection range appears to be the tightest.
  - Lasso Regression can be a good tool to select features, but need to learn more.
  - Result accuracy hinges on
    - Availability of historical data
    - Predictive capabilities of FB Prophet on future values of Features
    - ML model selection and parameter setting
  - One needs to understand the models in-depth to use them well.

Steps after MVP

More questions than answers:

- Further explore the ML models to improve results
  - Model selection / combination
  - Learn to use Lasso for feature selection
  - Better understand how parameters impact model results (X, y split; y as price or return)
  - Incorporate features with shorter history
- Adapt the model for Etherem and DeFi protocols


QUESTIONS


Thank You