Fun with Cryptocurrencies

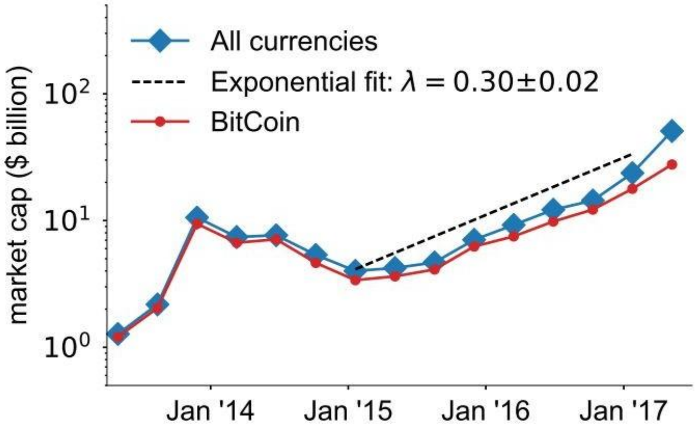
The 21st Century Gold-Rushers

Jordan Mincey, Lance Simmons,
Sinan Ulkuatam, & Bill Young

Department of Computer Science
School of Engineering and Applied Science

Problem and Motivation

- Cryptocurrencies are booming! [1] Unfortunately, cryptocurrencies are less stable and more poorly understood than traditional fiat currencies.
- Can machine learning techniques isolate the **predominant forces** driving prices?
- Can **short-term price** movement be predicted with >50% accuracy?
- Can we successfully predict the **next day's price?**



Background: Challenges and Goals

- Cryptocurrencies are volatile and heavily dependent on side-channel, qualitative information
    - e.g., a positive press release could double or triple the price of a coin
- We understood *a priori* we would not get high accuracy based on the nature of cryptocurrencies
- **Goal**: utilize machine learning techniques to predict short-term cryptocurrency price movement

Background: State-of-the-art

- While autotraders for the stock market are commonplace, the same techniques do not work on cryptocurrencies…

- Due to the large amount of trading bots in existence, individuals usually fail to make any significant profit trading technical indicators
    - Few existing academic works, but plenty of personal blogs

Data

- Historical market data from a cryptocurrency dataset maintained by Sudalai Rajkumar on Kaggle [3]
    - Thousands of days' worth of cryptocurrency prices ranging from mid-2009 through September 30th, 2017.
    - High, Low, Open, Close, & Volume attributes
    - Bitcoin contains additional attributes

Historical Data

- Bitcoin
- Ethereum
- Ripple
- Bitcoin cash
- Bitconnect
- Dash
- Ethereum Classic
- Iota
- Litecoin
- Monero
- Nem
- Neo
- Numeraire
- Stratis
- Waves

Available at: https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory

Data Preprocessing

- Elimination of NaN values
- Addition of training attributes
  - price change (%)
  - simple moving averages
  - S&P 500 and Bitcoin values for some cryptos
- Addition of target attributes
  - does the price increase tomorrow? (classification)
  - tomorrow’s price (regression)

Feature Selection

- We performed Recursive Feature Elimination with sk-learn’s Gradient Boost model.
    - Gradient Boost was used since it was not among our existing models; this was done to prevent bias towards a specific model.

1. RFE trains a model on the training set using all features.
2. Feature importances are ranked.
3. The least important *n* features are removed, then RFE trains a model with the reduced feature set.
4. Repeat steps *2* and *3* until we are left with a set of features of some predetermined size *m*.

Experimental Design - Classification Task

- Acquire, parse, and clean the dataset
- Perform feature selection according to the Recursive Feature Elimination technique
- Train and test each algorithm in accordance wth double-resampling, using nested 5-fold cross validation to select an optimal model and 5-fold cross validation to test sequestered data using the optimal model
- Tabulate metrics associate with training and testing results, including confusion matrices, the Matthews Correlation Coefficient, and validation/testing accuracy

Experimental Design - Regression Task

- Acquire, parse, and clean the dataset
    - Preprocessing: target prices
- Train and test each algorithm in accordance wth double-resampling, using nested 5-fold cross validation to select an optimal model and 5-fold cross validation to test sequestered data using the optimal model
- Tabulate metrics associate with training and testing results, including R-squared values and mean square error

Experimental Design: Algorithms

- Random Forest Classifier & Regressor
- Adaboost (Decision Tree) Classifier & Regressor
- Decision Tree Classifier & Regressor
- Multi-Layer Perceptron (Neural Network) Classifier
- k-Nearest Neighbors Classifier & Regressor

Experimental Design: Model Selection

- **Double resampling**
  - five differently-tuned versions of each algorithm were trained and validated, and the **best** model was tested on the sequestered outer-fold data
  - k-Fold Cross Validation: k = 5 → *double nested*

- **Optimality Metrics** (for selecting optimal model)
  - Accuracy (classification)
  - R-squared (regression)

Evaluation Metrics

- **Accuracy** (Percent of items correctly identified)
  - *classification only*
- **Matthews Correlation Coefficient** (range in [0 .. ± 1])
  - *classification only*
- **Mean Square Error**
  - *regression only*
- **R-squared**
  - *regression only*

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$|\text{MCC}| = \sqrt{\frac{\chi^2}{n}}$$

Results - Classification

	Bitcoin		*Ethereum*		*Ripple*		*NEO*	
Model	**Accuracy**	**MCC**	**Accuracy**	**MCC**	**Accuracy**	**MCC**	**Accuracy**	**MCC**
Random Forest	59.04%	**0.233**	**56.8%**	**0.14**	**57.9%**	**0.139**	**61.3%**	**0.22**
AdaBoost	59.95%	0.193	54.1%	0.12	56.4%	0.090	51.4%	0.03
Decision Tree	**61.10%**	0.229	49.66%	0.07	53.7%	0.025	55.4%	0.15
MLP	56.58%	0.104	48.64%	0.03	53.5%	0.007	53.2%	0.07
k-NN	58.04%	0.153	45.9%	0.08	53.6%	0.033	58.1%	0.15

Model comparison across all four cryptocurrencies during the classification task

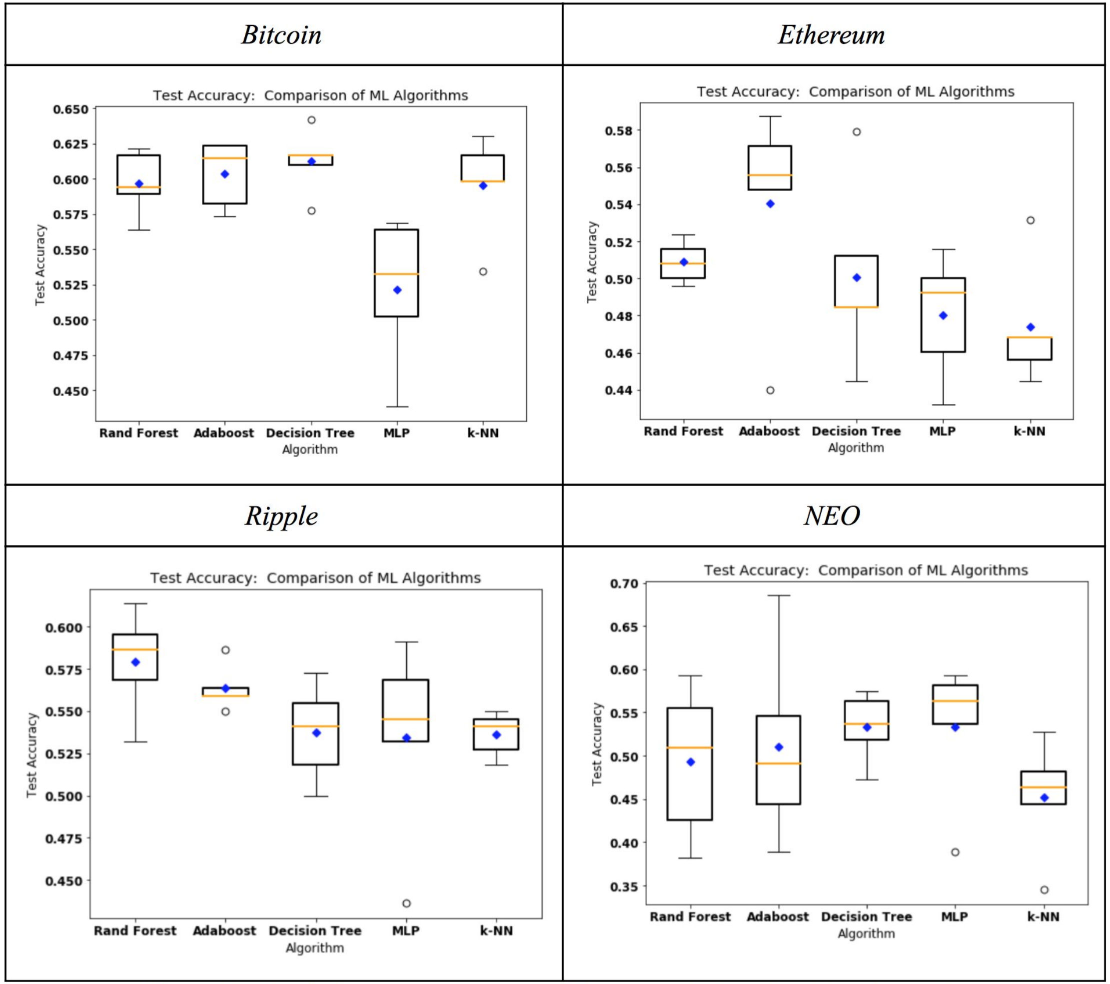


Figure 1. Statistical summary of model testing during classification task.

Results - Regression

	Bitcoin	*Ethereum*	*Ripple*	*NEO*
Model	**R-squared**	**R-squared**	**R-squared**	**R-squared**
Random Forest	**0.9958**	**0.9914**	**0.9881**	**0.9967**
AdaBoost	0.9841	0.9837	0.9783	0.9897
Decision Tree	0.9925	0.9828	0.9717	0.9899
k-NN	0.9948	0.9365	0.9763	0.9953

Table 2. Model comparison across all four cryptocurrencies during the regression task

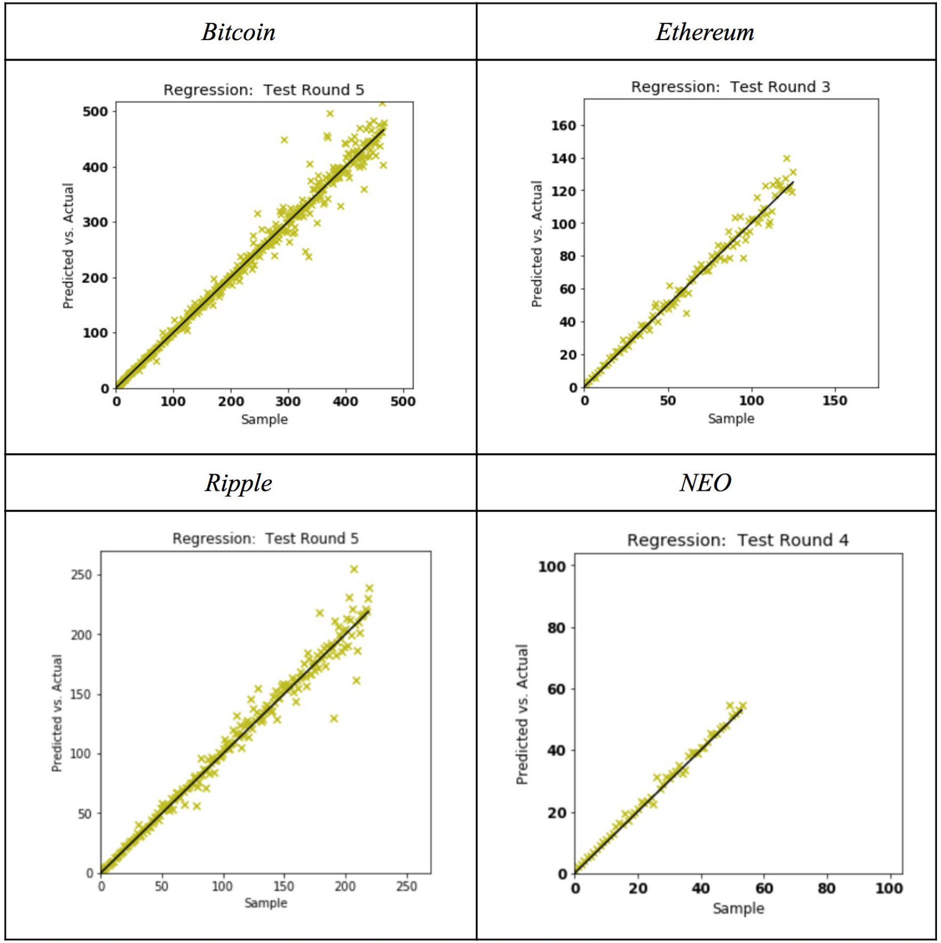


Figure 2. Visualization of regression quality for best Random Forest regressor models. The black line of slope 1.0 represents perfect prediction accuracy, i.e. the prediction matched the ground-truth.

Software & Hardware

- Python 2.7 inside a Juypter Notebook (single-pass script)
- Used regular commodity laptops running either macOS or Windows 7/8.
  - The entire test script completed on all laptops within 4 minutes, so no specialty hardware was required
- The Bitcoin dataset was the largest at 717 KB
- The other datasets were all less than 350 KB

Discussion & Conclusion

- Our classification algorithms performed about how we expected with around 60% accuracy at best
- The regressors performed well with the best R-squared values being around 0.99
  - These values are deceptive!
  - Next day predictions can be predicted with high accuracy by predicting no change.
- Regressors for predicting prices farther than two weeks out would decrease significantly in accuracy due to nature of market.

Bitcoin increased $1,000 dollars yesterday without any technical indicators!

References

[1] https://www.technologyreview.com/s/607947/the-cryptocurrency-market-is-growing-exponentially

[2] https://bitcoin.org/bitcoin.pdf

[3] https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory

[4] https://arxiv.org/abs/1410.1231

[5] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2051138

[6] https://en.wikipedia.org/wiki/Technical_analysis

[7] https://storeofvalue.github.io/posts/technical-analysis-and-cryptocurrencies/

[8] https://arxiv.org/abs/1603.00751

[9] https://bittrex.com/

Questions?

Shovels for sale here -->