Drew Shea, Nikki Duran, Federico Kulyckyj, and Anthony Jonikas
MHacks X

rBitrage Project Spec

Implementing Machine Learning to Analyze Dependent Price Movement between Bitcoin and Ethereum

The purpose of this project was to apply Microsoft's Azure machine learning platform to historical price and volume data on the two largest cryptocurrencies available today. A dataset of historical data was gathered for training and testing to a neural network to determine the accuracy of the results. Then, using a basic iOS app, we display the current market Ethereum price and predict the Ethereum price in several time intervals in the future with a confidence level using our algorithms.

In order to gather historical data, we connected to Coinbase's GDAX API to gather BTC and ETH statistics per minute for the past year. Our training data for the machine learning platform was the previous six months not including September (March, April, May, June, July, August). The total data accumulation was roughly 260,000 time intervals. Each time interval consisted of the currencies high, low, open, and close.

A back-testing program was implemented in C++, where we used the September data set that the machine learning platform had not seen yet. The code for the back-testing is available to look at and is designed to replicate an actual trading algorithm implementation.

The machine learning aspect of the project used Microsoft Azure. We used parameters in a regression based system, boosted decision tree, and neural network.

13 Parameters, 5 Price % variables, 5 % change volume variables, 1 5minute moving average % change, 2 special financial parameters

Price: $\frac{P_n-P_{n-1}}{P_{n-1}}$ $\frac{P_n-P_{n-2}}{P_{n-2}}$ $\frac{P_n-P_{n-3}}{P_{n-3}}$ $\frac{P_n-P_{n-4}}{P_{n-4}}$ $\frac{P_n-P_{n-5}}{P_{n-5}}$

Volume: $\frac{V-V_{n-1}}{V_{n-1}}$ $\frac{V-V_{n-2}}{V_{n-2}}$ $\frac{V-V_{n-3}}{V_{n-3}}$ $\frac{V-V_{n-4}}{V_{n-4}}$ $\frac{V-V_{n-5}}{V_{n-5}}$

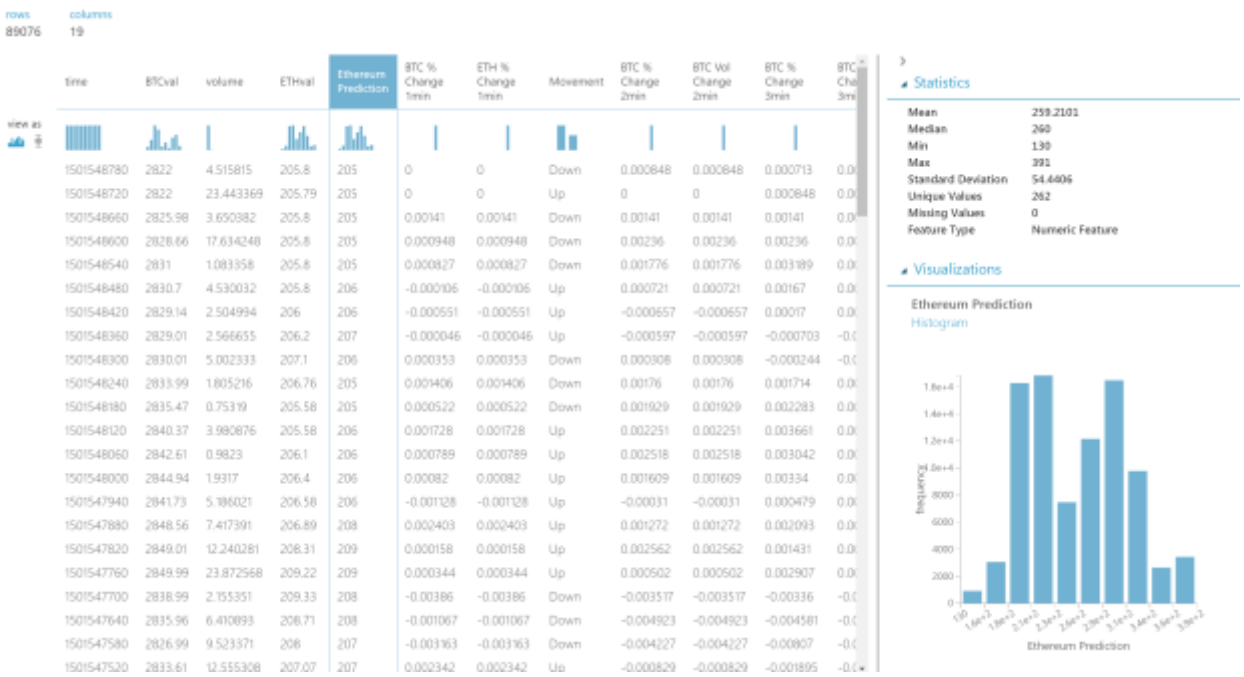
Moving Average: $\frac{P_n-\frac{\sum_{n-5}^{n-1}P}{5}}{\frac{\sum_{n-5}^{n-1}P}{5}}$

Special Algorithms: $\frac{Pbtc_n-Peth_n}{P_{btc(n-5)}-P_{eth(n-5)}}$ BTC – ETH $\frac{\frac{V-v_{n-5}}{v_{n-5}}}{\frac{v-v_{n-5}}{v_{n-5}}}$ BTC/ETH

The algorithms above were used to test and train the neural network, and the special algorithms were just designed by us to test some theories regarding what specifically influences the relationship between BTC and ETH.

```
1504227540	392.44	392.45	392.45	392.45	93.22040999	1504227540	4765.49	4765.49	4765.49	4765.49	3.55418055
1504227480	392.44	392.45	392.45	392.45	60.00382487	1504227480	4765.48	4765.49	4765.49	4765.49	3.76806493
1504227420	391.68	392.49	391.68	392.45	133.968599	1504227420	4765.48	4765.49	4765.49	4765.49	6.00310294
1504227360	391.65	391.68	391.66	391.68	57.61896336	1504227360	4765.48	4765.49	4765.49	4765.48	3.64841937
1504227300	391.59	391.66	391.59	391.66	25.4256483	1504227300	4765.48	4765.49	4765.49	4765.49	4.41516708
1504227240	391.88	391.89	391.89	391.89	69.50386886	1504227240	4765.48	4765.49	4765.49	4765.48	3.48072247
1504227180	391.58	391.93	391.92	391.58	54.35158896	1504227180	4765.49	4765.49	4765.49	4765.49	3.24487559
1504227120	391.93	392.8	392.8	391.93	605.6311575	1504227120	4765.49	4765.49	4765.49	4765.49	13.60573237
1504227060	392.79	392.8	392.8	392.8	100.8220636	1504227060	4765.49	4765.49	4765.49	4765.49	6.99230817
1504227000	392.79	392.8	392.8	392.79	71.60567564	1504227000	4765.49	4765.49	4765.49	4765.49	10.26167775
1504226940	392.79	392.8	392.8	392.79	151.0370872	1504226940	4765.49	4765.49	4765.49	4765.49	6.38116901
1504226880	391.69	392.8	391.7	392.8	724.0621705	1504226880	4765.49	4765.49	4765.49	4765.49	3.9284663
1504226820	391.49	391.7	391.6	391.7	317.3932361	1504226820	4765.48	4765.49	4765.49	4765.49	3.51096867
1504226760	391.49	391.5	391.5	391.5	156.0560619	1504226760	4765.48	4765.49	4765.49	4765.48	14.91533292
1504226700	391.49	391.6	391.6	391.6	10.11651095	1504226700	4765.48	4765.49	4765.49	4765.49	1.67302863
1504226640	391.49	391.5	391.6	391.5	65.05424283	1504226640	4765.49	4765.49	4765.49	4765.49	5.00906727
1504226580	391.49	391.5	391.6	391.5	40.64330114	1504226580	4764.98	4765.48	4764.99	4765.48	17.16258145
1504226520	391.47	391.5	391.48	391.5	178.6093371	1504226520	4764.99	4764.99	4764.99	4764.99	9.13473206
1504226460	391.44	391.48	391.44	391.47	81.05891818	1504226460	4764.99	4764.99	4764.99	4764.99	9.38297951
1504226400	391.06	391.44	391.07	391.43	130.4917079	1504226400	4759.36	4764.99	4759.36	4764.99	7.35456252
1504226340	390.99	391.26	390.99	391.06	808.0576618	1504226340	4759.3	4759.36	4759.3	4759.36	11.88493282
1504226280	390.44	391	390.45	390.99	334.7294388	1504226280	4759.3	4759.3	4759.3	4759.3	9.16335857
1504226220	390.44	390.45	390.44	390.45	29.80420195	1504226220	4751.96	4759.3	4751.96	4759.3	14.01219674
1504226160	390.44	390.45	390.44	390.44	106.1364818	1504226160	4750	4750.88	4750	4750.88	15.99927804
1504226100	390.44	390.45	390.45	390.45	61.63899844	1504226100	4750	4750	4750	4750	7.44526394
1504226040	390.44	390.45	390.45	390.45	166.3042803	1504226040	4750	4750	4750	4750	10.73760439
1504225980	390.17	390.98	390.49	390.46	316.016031	1504225980	4750	4750	4750	4750	16.09088144
1504225920	390.33	390.71	390.7	390.33	166.0437467	1504225920	4749.98	4750	4749.99	4750	18.04294991
1504225860	390.7	390.91	390.91	390.7	275.3540652	1504225860	4749.68	4750	4749.69	4749.99	16.55165061
1504225800	390.6	391	390.6	390.86	764.0153131	1504225800	4748	4749.69	4748	4749.69	13.74292040
1504225740	389.9	390.5	389.98	390.5	1513.447216	1504225740	4748	4748	4748	4748	22.5087613
1504225680	389.92	390	389.94	389.98	126.3041045	1504225680	4744.99	4745	4745	4745	6.86641157
1504225620	389.92	389.95	389.92	389.95	95.61360674	1504225620	4744.98	4745	4744.98	4745	22.37709672
1504225560	389.92	389.94	389.94	389.92	16.90928196	1504225560	4744.98	4744.99	4744.98	4744.99	6.59406208
1504225500	389.93	389.94	389.93	389.93	32.83605896	1504225500	4744.5	4744.98	4744.5	4744.98	4.43870196
1504225440	389.93	389.96	389.93	389.96	95.83911983	1504225440	4744.01	4744.5	4744.01	4744.5	15.99986871
1504225380	389.93	389.94	389.93	389.94	17.99737827	1504225380	4744	4744.01	4744.01	4744.01	4.20893057
1504225320	389.92	389.93	389.93	389.93	46.21632095	1504225320	4744.01	4744.01	4744.01	4744.01	2.30523709
1504225260	389.93	389.93	389.93	389.92	139.1117573	1504225260	4744	4744.01	4744.01	4744.01	10.59145002
1504225200	389.74	390	389.76	389.92	260.7100167	1504225200	4744	4744.01	4744.01	4744.01	2.60023156
1504225140	389.73	389.75	389.75	389.75	67.22000183	1504225140	4744	4744.01	4744.01	4744.01	11.33092475
1504225080	389.73	389.75	389.75	389.74	142.7135363	1504225080	4744	4744.01	4744.01	4744.01	4.77302906
1504225020	389.74	389.75	389.74	389.75	141.034465	1504225020	4744.01	4744.01	4744.01	4744.01	3.7662126
1504224960	389.73	389.74	389.74	389.73	85.79586981	1504224960	4744.01	4744.01	4744.01	4744.01	0.35874605
1504224900	389.71	389.75	389.73	389.72	41.41150132	1504224900	4744	4744.01	4744	4744.01	2.62701792
1504224840	389.6	389.87	389.66	389.71	69.96828655	1504224840	4744	4744.01	4744.01	4744.01	6.00161883
```

```cpp
int main() {
    //Initalize BUY/SELL
    sellETH = false;
    buyETH = false;
    //Initalize quantities
    quantityCash = 10000;
    quantityETH = 0;
    //Read in September Data
    ifstream input("SeptemberDatatext.txt");
    input >> timeETHOriginal >> lowETHOriginal >> highETHOriginal >> openETHOriginal >> closeETHOriginal >> volumeETHOriginal >> timeBTCOriginal >> lowBTCOriginal
        >> highBTCOriginal >> openBTCOriginal >> closeBTCOriginal >> volumeBTCOriginal;
    if (input.is_open()){
        while (input >> timeETHNext >> lowETHNext >> highETHNext >> openETHNext >> closeETHNext >> volumeETHNext >> timeBTCNext >> lowBTCNext >> highBTCNext >>
            openBTCNext >> closeBTCNext >> volumeBTCNext){
            blackBox(openBTCOriginal);
            //Determine Investment Amount
            //Backtest
            //Update Holdings
            //Reset Next to Original for next iteration
            timeETHOriginal = timeETHNext;
            timeBTCOriginal = timeBTCNext;
            lowETHOriginal = lowETHNext;
            lowBTCOriginal = lowBTCNext;
            highETHOriginal = highETHNext;
            highBTCOriginal = highBTCNext;
            openETHOriginal = openETHNext;
            openBTCOriginal = openBTCNext;
            closeETHOriginal = closeETHNext;
            closeBTCOriginal = closeBTCNext;
            volumeETHOriginal = volumeETHNext;
            volumeBTCOriginal = volumeBTCNext;
        }
    }
    else {
        cout <<"There was an error reading the September Data file" << endl;
    }
    cout << "Closing Cash $" << quantityCash << endl;
    cout << "Closing Quantity of ETH " << quantityETH << endl;
    cout << "Closing Price of ETH $" <<  closeETHNext << endl;
    cout << "Closing Amount of Holdings $" << (quantityCash + quantityETH * closeETHNext) << endl;
    double percentReturns = ((quantityCash + quantityETH * closeETHNext)-10000)/ 10000 * 100;
    cout << "Backtesting algorithm provides " << percentReturns << "% for month of September";
```



rows 89076 columns 19

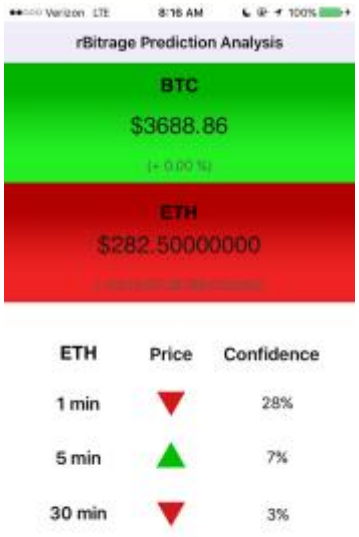
time	BTCval	volume	ETHval	Ethereum Prediction	BTC % Change 1min	ETH % Change 1min	Movement	BTC % Change 2min	BTC Vol Change 2min	BTC % Change 3min	BTC Ch 3mi
1501548780	2822	4.515815	205.8	205	0	0	Down	0.000848	0.000848	0.000713	0.0
1501548720	2822	23.443369	205.79	205	0	0	Up	0	0	0.000848	0.0
1501548660	2825.98	3.650382	205.8	205	0.00141	0.00141	Down	0.00141	0.00141	0.00141	0.0
1501548600	2828.66	17.634248	205.8	205	0.000948	0.000948	Down	0.00236	0.00236	0.00236	0.0
1501548540	2831	1.083358	205.8	205	0.000827	0.000827	Down	0.001776	0.001776	0.003189	0.0
1501548480	2830.7	4.530032	205.8	206	-0.000106	-0.000106	Up	0.000721	0.000721	0.00167	0.0
1501548420	2829.14	2.504994	206	206	-0.000551	-0.000551	Up	-0.000657	-0.000657	0.00017	0.0
1501548360	2829.01	2.566655	206.2	207	-0.000046	-0.000046	Up	-0.000597	-0.000597	-0.000703	-0.0
1501548300	2830.01	5.002333	207.1	206	0.000353	0.000353	Down	0.000308	0.000308	-0.000244	-0.0
1501548240	2833.99	1.805216	206.76	205	0.001406	0.001406	Down	0.00176	0.00176	0.001714	0.0
1501548180	2835.47	0.75319	205.58	205	0.000522	0.000522	Down	0.001929	0.001929	0.002283	0.0
1501548120	2840.37	3.980876	205.58	206	0.001728	0.001728	Up	0.002251	0.002251	0.003661	0.0
1501548060	2842.61	0.9823	206.1	206	0.000789	0.000789	Up	0.002518	0.002518	0.003042	0.0
1501548000	2844.94	1.9317	206.4	206	0.00082	0.00082	Up	0.001609	0.001609	0.00334	0.0
1501547940	2841.73	5.186021	206.58	206	-0.001128	-0.001128	Up	-0.00031	-0.00031	0.000479	0.0
1501547880	2848.56	7.417391	206.89	208	0.002403	0.002403	Up	0.001272	0.001272	0.002093	0.0
1501547820	2849.01	12.240281	208.31	209	0.000158	0.000158	Up	0.002562	0.002562	0.001431	0.0
1501547760	2849.99	23.872568	209.22	209	0.000344	0.000344	Up	0.000502	0.000502	0.002907	0.0
1501547700	2838.99	2.155351	209.33	208	-0.00386	-0.00386	Down	-0.003517	-0.003517	-0.00336	-0.0
1501547640	2835.96	6.410893	208.71	208	-0.001067	-0.001067	Down	-0.004923	-0.004923	-0.004581	-0.0
1501547580	2826.99	9.523371	208	207	-0.003163	-0.003163	Down	-0.004227	-0.004227	-0.00807	-0.0
1501547520	2833.61	12.555308	207.07	207	0.002342	0.002342	Up	-0.000829	-0.000829	-0.001895	-0.0

Statistics

Mean	259.2101
Median	260
Min	130
Max	391
Standard Deviation	54.4406
Unique Values	262
Missing Values	0
Feature Type	Numeric Feature

Visualizations

Ethereum Prediction
Histogram

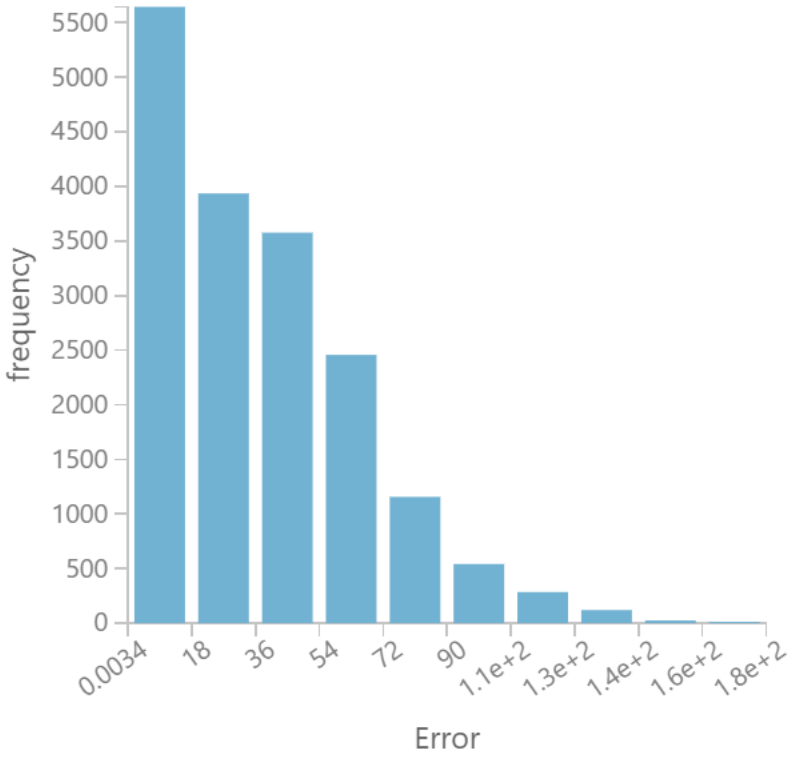


ETH	Price	Confidence
1 min	▼	28%
5 min	▲	7%
30 min	▼	3%

Metrics

Mean Absolute Error	37.49719
Root Mean Squared Error	47.412749
Relative Absolute Error	0.799697
Relative Squared Error	0.757922
Coefficient of Determination	0.242078

Error Histogram



```
rows, cols = combined_df.shape
rowCount_int = 0
predictAhead =  1 #1


while rowCount_int < rows - predictAhead:
    if (rowCount_int > 1):
        #Calculate BTC val and BTC val at (position -1)
        BTC_valCurrent_float = (combined_df.iloc[rowCount_int,BTC_val_ind])
        BTC_valPrevious_float = (combined_df.iloc[rowCount_int - 1,BTC_val_ind])
        BTC_percentChange_float = (BTC_valCurrent_float - BTC_valPrevious_float) / (BTC_valPrevious_float)
        combined_df.at[rowCount_int, 'BTC % Change 1min'] = BTC_percentChange_float

        #ETH_valCurrent_float = (combined_df.iloc[rowCount_int,ETH_val_ind])
        #ETH_valPrevious_float = (combined_df.iloc[rowCount_int - 1,ETH_val_ind])
        #ETH_percentChange_float = (ETH_valCurrent_float - ETH_valPrevious_float) / (ETH_valPrevious_float)
        BTC_valCurrent_float = (combined_df.iloc[rowCount_int,BTC_val_ind])
        BTC_valPrevious_float = (combined_df.iloc[rowCount_int - 1,BTC_val_ind])
        BTC_percentChange_float = (BTC_valCurrent_float - BTC_valPrevious_float) / (BTC_valPrevious_float)
        combined_df.at[rowCount_int, 'ETH % Change 1min'] = BTC_percentChange_float

        #ETH_valCurrent_float = (combined_df.iloc[rowCount_int,ETH_val_ind])
        #ETH_valPrevious_float = (combined_df.iloc[rowCount_int - 1,ETH_val_ind])
        #ETH_percentChange_float = (ETH_valCurrent_float - ETH_valPrevious_float) / (ETH_valPrevious_float)

        #combined_df.at[rowCount_int, 'ETH % Change'] = ETH_percentChange_float

        #combined_df.at[rowCount_int, 'Difference'] = BTC_percentChange_float - ETH_percentChange_float

        combined_df.at[rowCount_int, 'Ethereum Prediction'] = (combined_df.iloc[rowCount_int +  predictAhead, ETH_val_ind])

        if ((combined_df.iloc[rowCount_int +  predictAhead, ETH_val_ind]) > (combined_df.iloc[rowCount_int, ETH_val_ind])):
            combined_df.at[rowCount_int, 'Movement'] = 'Up'
        else:
            combined_df.at[rowCount_int, 'Movement'] = 'Down'
```