Crypto Monero - Exploratory Analysis

August 20, 2018

Exploratory Analysis on Historical Data and Prediction Test using ML for Monero (Crypto-coin)

Author: Leandro Medeiros

Date: 2018-08-18

Dataset from:

Coinmetrics.io

```
In [166]: import warnings
          warnings.filterwarnings('ignore')

          import sklearn.model_selection
          import sklearn.datasets
          import sklearn.metrics
          import autosklearn.regression
          import pandas
          from matplotlib import pyplot
```

```
In [17]: dfxmr = pandas.read_csv('/home/leandro/projects/cryptobot/exploratoryAnalysis/datasets
```

```
In [18]: dfxmr.head()
```

Out[18]:

	date	index	txCount	marketcap(USD)	price(USD)	\
0	2014-05-21	1	347	2079640.0	2.47	
1	2014-05-22	2	477	1371470.0	1.59	
2	2014-05-23	3	661	1816200.0	2.05	
3	2014-05-24	4	1026	2653720.0	2.92	
4	2014-05-25	5	1486	3774890.0	4.04	

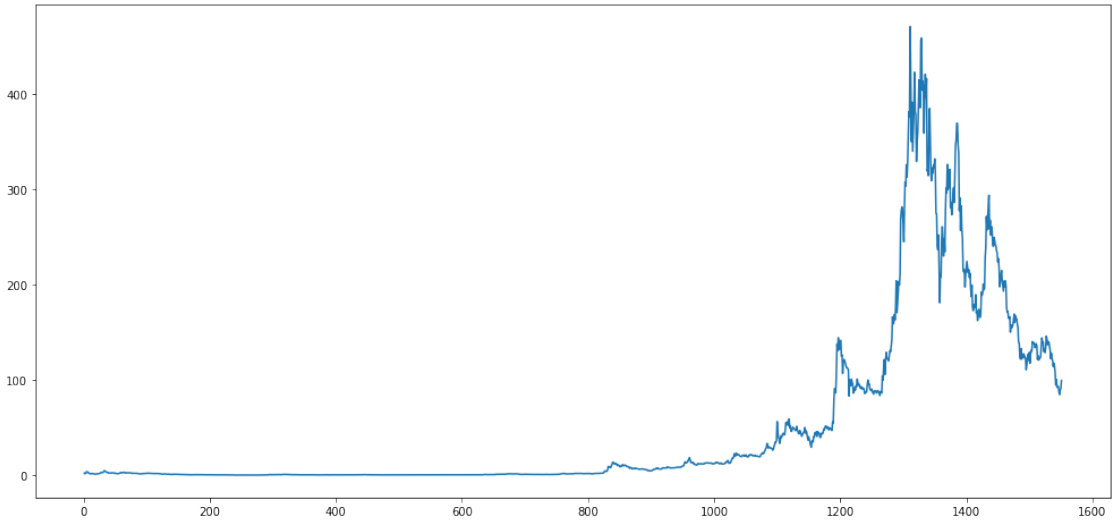
	exchangeVolume(USD)	fees	generatedCoins	activeAddresses	medianFee	\
0	246540.0	0.000469	23625.790320	48255	0.000001	
1	132918.0	0.000657	22666.109152	101585	0.000001	
2	266852.0	0.000911	23167.795605	118412	0.000001	
3	248028.0	0.002842	24975.837341	167516	0.000001	
4	283545.0	0.002369	20133.313716	162895	0.000001	

	averageDifficulty	paymentCount	blockCount	blockSize
0	1.048124e+11	30693	1430	4089846
1	9.775957e+07	77897	1401	6838373

```
2       9.918136e+07          78276         1447     8499038
3       1.304686e+11         117383         1599    12588343
4       1.458712e+11         102566         1362    13957477
```

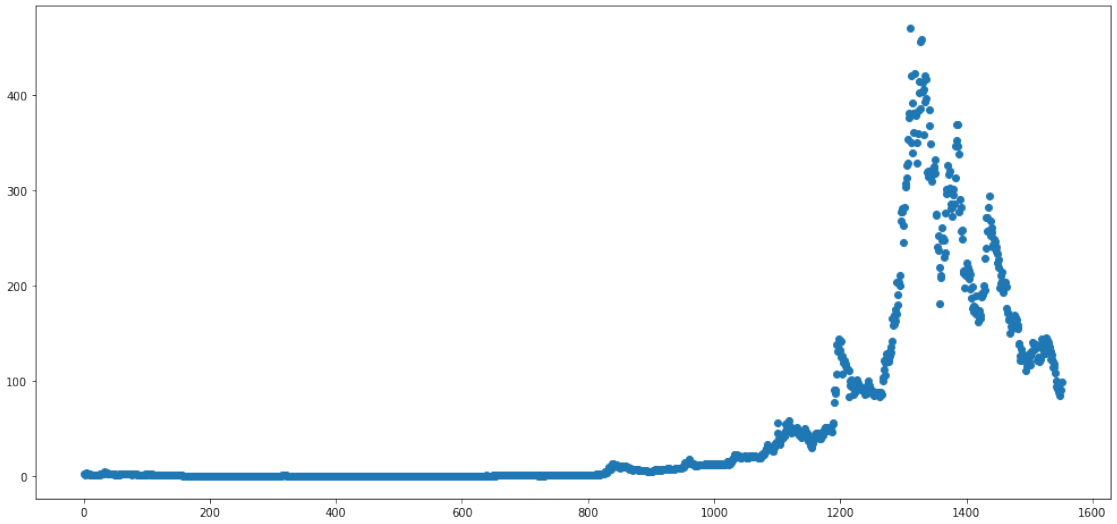
```
In [167]: pyplot.figure(figsize=(17,8))
          pyplot.plot(dfxmr['index'], dfxmr['price(USD)'])
```

```
Out[167]: [<matplotlib.lines.Line2D at 0x7f23f5414128>]
```

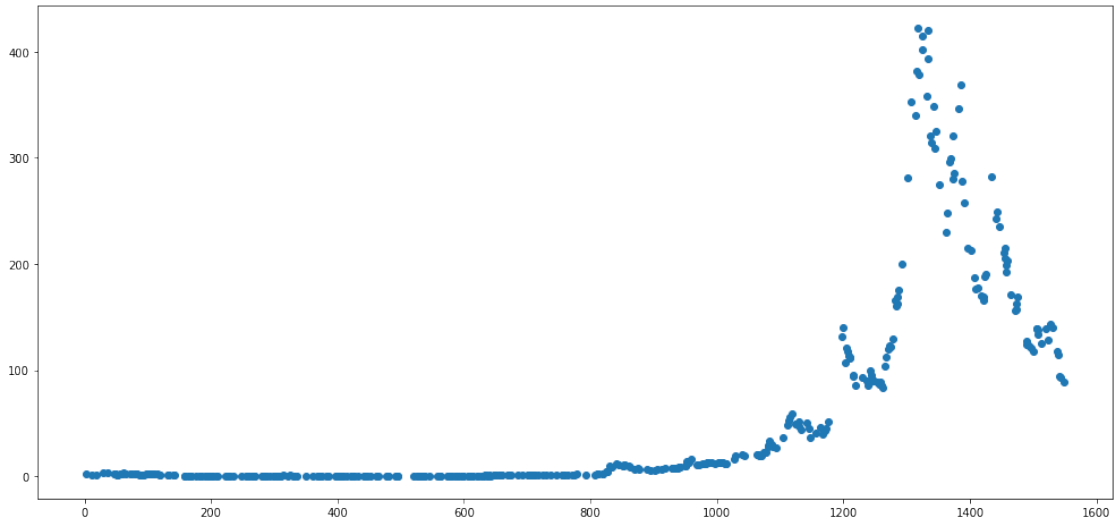



```
In [259]: pyplot.figure(figsize=(17,8))
          pyplot.scatter(x=dfxmr['index'], y=dfxmr['price(USD)'])
```

```
Out[259]: <matplotlib.collections.PathCollection at 0x7f23ed93a4a8>
```

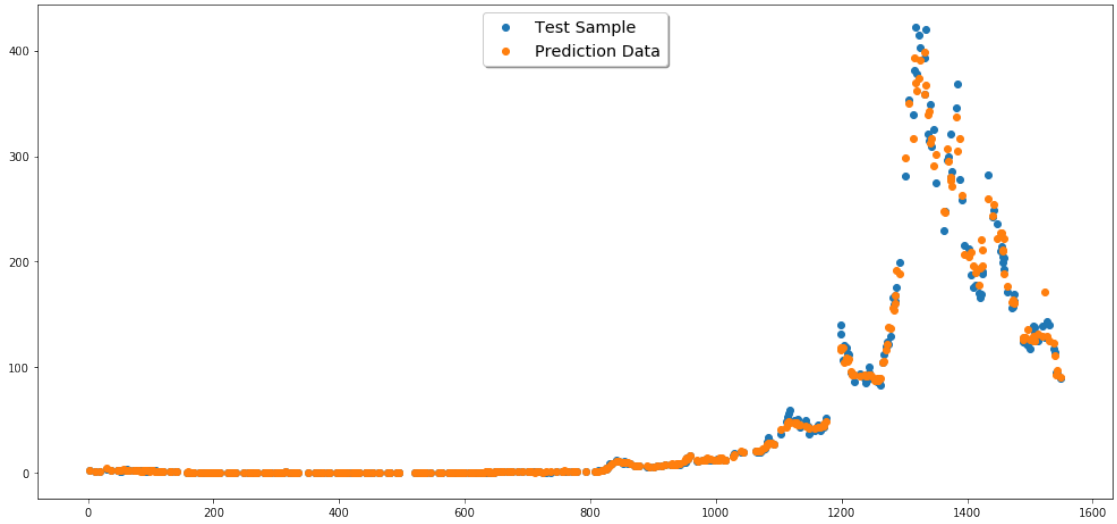


```
In [168]: dfxmr.shape

Out[168]: (1551, 14)

In [245]: x = dfxmr.loc[:,['index','txCount','exchangeVolume(USD)']]

In [246]: x.shape

Out[246]: (1551, 3)

In [247]: y = dfxmr['price(USD)']

In [232]: y.shape

Out[232]: (1551,)

In [248]: X_train, X_test, y_train, y_test = \
                  sklearn.model_selection.train_test_split(x, y, random_state=1)

In [260]: pyplot.figure(figsize=(17,8))
          pyplot.scatter(x=y_test.to_frame().index,y=y_test.to_frame()['price(USD)'])

Out[260]: <matplotlib.collections.PathCollection at 0x7f23ed911780>
```




```
In [250]: automl = autosklearn.regression.AutoSklearnRegressor(
                  time_left_for_this_task=120,
                  per_run_time_limit=30
              )
```

```
In [251]: feature_types = (['numerical'] + ['numerical'] + ['Numerical'])

In [252]: automl.fit(X=X_train,y=y_train, dataset_name='test',feat_type=feature_types)

[WARNING] [2018-08-20 00:40:25,170:AutoMLSMBO(1)::test] Could not find meta-data directory /hor
[WARNING] [2018-08-20 00:40:25,182:EnsembleBuilder(1):test] No models better than random - usir
[WARNING] [2018-08-20 00:40:25,191:EnsembleBuilder(1):test] No models better than random - usir


Out[252]: AutoSklearnRegressor(delete_output_folder_after_terminate=True,
                               delete_tmp_folder_after_terminate=True,
                               disable_evaluator_output=False, ensemble_nbest=50,
                               ensemble_size=50, exclude_estimators=None,
                               exclude_preprocessors=None, get_smac_object_callback=None,
                               include_estimators=None, include_preprocessors=None,
                               initial_configurations_via_metalearning=25,
                               ml_memory_limit=3072, output_folder=None, per_run_time_limit=30,
                               resampling_strategy='holdout',
                               resampling_strategy_arguments=None, seed=1, shared_mode=False,
                               smac_scenario_args=None, time_left_for_this_task=120,
                               tmp_folder=None)

In [253]: print(automl.show_models())

[(1.000000, SimpleRegressionPipeline({'categorical_encoding:__choice__': 'one_hot_encoding', '
dataset_properties={
  'task': 4,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'regression',
  'signed': False})),
]


In [254]: predictions = automl.predict(X_test)

In [255]: print("MAE score:", sklearn.metrics.mean_absolute_error(y_test, predictions))
          print('R2 score',  sklearn.metrics.r2_score(y_test, predictions))

MAE score: 3.87514754788179
R2 score 0.9903448170996811


In [264]: pyplot.figure(figsize=(17,8))
          pyplot.scatter(x=y_test.to_frame().index,y=y_test.to_frame()['price(USD)'], label='T
          pyplot.scatter(x=y_test.to_frame().index,y=predictions, label="Prediction Data")
          pyplot.legend(loc='upper center', shadow=True, fontsize='x-large')

Out[264]: <matplotlib.legend.Legend at 0x7f23ef1fb320>
```


Test Sample
Prediction Data
400
300
200
100
0
0
200
400
600
800
1000
1200
1400
1600