

1.0-initial-data-exploration

April 7, 2022

```
[ ]: import pandas as pd

from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler

import plotly.graph_objects as go
from matplotlib import pyplot as plt
from matplotlib.dates import DateFormatter
import plotly.io as pio
pio.renderers.default = 'pdf'
```

```
[ ]: bitcoin = pd.read_csv('../data/external/bitcoin.csv', parse_dates=['Date'])
cardano = pd.read_csv('../data/external/cardano.csv', parse_dates=['Date'])
ethereum = pd.read_csv('../data/external/ethereum.csv', parse_dates=['Date'])

df = pd.concat([bitcoin, cardano, ethereum], axis=0)
```

```
[ ]: df.shape
```

```
[ ]: (6525, 10)
```

```
[ ]: df.head()
```

```
[ ]: 
```

| | SNo | Name | Symbol | Date | High | Low | \ |
|---|-----|---------|--------|---------------------|------------|------------|---|
| 0 | 1 | Bitcoin | BTC | 2013-04-29 23:59:59 | 147.488007 | 134.000000 | |
| 1 | 2 | Bitcoin | BTC | 2013-04-30 23:59:59 | 146.929993 | 134.050003 | |
| 2 | 3 | Bitcoin | BTC | 2013-05-01 23:59:59 | 139.889999 | 107.720001 | |
| 3 | 4 | Bitcoin | BTC | 2013-05-02 23:59:59 | 125.599998 | 92.281898 | |
| 4 | 5 | Bitcoin | BTC | 2013-05-03 23:59:59 | 108.127998 | 79.099998 | |

| | Open | Close | Volume | Marketcap |
|---|------------|------------|--------|--------------|
| 0 | 134.444000 | 144.539993 | 0.0 | 1.603769e+09 |
| 1 | 144.000000 | 139.000000 | 0.0 | 1.542813e+09 |
| 2 | 139.000000 | 116.989998 | 0.0 | 1.298955e+09 |
| 3 | 116.379997 | 105.209999 | 0.0 | 1.168517e+09 |
| 4 | 106.250000 | 97.750000 | 0.0 | 1.085995e+09 |

```
[ ]: df.value_counts(['Symbol'])
```

```
[ ]: Symbol
      BTC      2991
      ETH      2160
      ADA      1374
      dtype: int64
```

0.1 Data Exploration

0.1.1 Bitcoin

```
[ ]: print(f"Min Date: {df[df.Symbol == 'BTC'].Date.min()}, Max Date: {df[df.Symbol == 'BTC'].Date.max()}")
```

Min Date: 2013-04-29 23:59:59, Max Date: 2021-07-06 23:59:59

```
[ ]: df[df.Symbol == 'BTC'].describe()
```

```
[ ]:
count      SNo      High      Low      Open      Close \
count  2991.000000  2991.000000  2991.000000  2991.000000  2991.000000
mean    1496.000000  6893.326038  6486.009539  6700.146240  6711.290443
std      863.571653  11642.832456  10869.032130  11288.043736  11298.141921
min       1.000000    74.561096    65.526001    68.504997    68.431000
25%      748.500000    436.179001    422.879486    430.445496    430.569489
50%     1496.000000    2387.610107    2178.500000    2269.889893    2286.409912
75%     2243.500000    8733.926948    8289.800459    8569.656494    8576.238715
max     2991.000000   64863.098908   62208.964366   63523.754869   63503.457930

count      Volume      Marketcap
count  2.991000e+03  2.991000e+03
mean    1.090633e+10  1.208761e+11
std      1.888895e+10  2.109438e+11
min      0.000000e+00  7.784112e+08
25%      3.036725e+07  6.305579e+09
50%      9.460360e+08  3.741503e+10
75%      1.592015e+10  1.499957e+11
max      3.509679e+11  1.186364e+12
```

```
[ ]: fig = go.Figure(data=go.Ohlc(x=df[df.Symbol == 'BTC']['Date'],
                                open=df[df.Symbol == 'BTC']['Open'],
                                high=df[df.Symbol == 'BTC']['High'],
                                low=df[df.Symbol == 'BTC']['Low'],
                                close=df[df.Symbol == 'BTC']['Close']))
fig.update_layout(
    title='Bitcoin OHLC',
    yaxis_title='Stock Price (USD)')
```

```
)
fig.show()
```

```
-----
ValueError                                Traceback (most recent call last)
/home/julio/studies/penn-state/crypto-deep-learning/notebooks/1.
↳ 0-initial-data-exploration.ipynb Cell 10' in <cell line: 10>()
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=0'>1</a> fig = go.Figure(data=go.
↳ Ohlc(x=df[df.Symbol == 'BTC']['Date'],
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=1'>2</a>                                open=df[df.Symbol_
↳ == 'BTC']['Open'],
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=2'>3</a>                                high=df[df.Symbol_
↳ == 'BTC']['High'],
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=3'>4</a>                                low=df[df.Symbol =
↳ 'BTC']['Low'],
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=4'>5</a>                                close=df[df.Symbol
↳ == 'BTC']['Close']))
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=5'>6</a> fig.update_layout(
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=6'>7</a>                                title='Bitcoin OHLC',
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=7'>8</a>                                yaxis_title='Stock Price (USD)
    <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-stat /
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=8'>9</a> )
--> <a href='vscode-notebook-cell://wsl%2Bubuntu/home/julio/studies/penn-state
↳ crypto-deep-learning/notebooks/1.0-initial-data-exploration.
↳ ipynb#ch0000009vscode-remote?line=9'>10</a> fig.show()

File ~/.local/share/virtualenvs/crypto-deep-learning-wXDHGxS_/lib/python3.8/
↳ site-packages/plotly/basedatatypes.py:3400, in BaseFigure.show(self, *args,
↳ **kwargs)
    <a href='file:///home/julio/.local/share/virtualenvs/
↳ crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/basedatatype
↳ py?line=3366'>3367</a> """
```

```

    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/basedatatypes.py?line=3367'>3368</a> Show a figure using either the default renderer(s) or
the renderer(s)
    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/basedatatypes.py?line=3368'>3369</a> specified by the renderer argument
    (...)
    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/basedatatypes.py?line=3395'>3396</a> None
    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/basedatatypes.py?line=3396'>3397</a> """
    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/basedatatypes.py?line=3397'>3398</a> import plotly.io as pio
-> <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/basedatatypes.py?line=3399'>3400</a> return pio.show(self, *args, **kwargs)

```

File ~/.local/share/virtualenvs/crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py:389, in show(fig, renderer, validate, **kwargs)

```

    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=385'>386</a> fig_dict = validate_coerce_fig_to_dict(fig, validate)
    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=387'>388</a> # Mimetype renderers
--> <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=388'>389</a> bundle = renderers._build_mime_bundle(fig_dict,
renderers_string=renderer, **kwargs)
    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=389'>390</a> if bundle:
    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=390'>391</a>     if not ipython_display:

```

File ~/.local/share/virtualenvs/crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py:297, in RenderersConfig._build_mime_bundle(self, fig_dict, renderers_string, **kwargs)

```

    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=293'>294</a>         if hasattr(renderer, k):
    <a href='file:///home/julio/.local/share/virtualenvs/
crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=294'>295</a>             setattr(renderer, k, v)

```

```
--> <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=296'>297</a>
↳py?line=296'>297</a> bundle.update(renderer.to_mimebundle(fig_dict))
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_renderers.py?line=298'>299</a> return bundle
```

File ~/.local/share/virtualenvs/crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py:128, in ImageRenderer.to_mimebundle(self, fig_dict)

```
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=126'>127</a> def to_mimebundle(self, fig_dict):
--> <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=127'>128</a> image_bytes = to_image(
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=128'>129</a> fig_dict,
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=129'>130</a> format=self.format,
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=130'>131</a> width=self.width,
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=131'>132</a> height=self.height,
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=132'>133</a> scale=self.scale,
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=133'>134</a> validate=False,
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=134'>135</a> engine=self.engine,
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=135'>136</a> )
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=137'>138</a> if self.b64_encode:
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_base_renderers.py?line=138'>139</a> image_str = base64.
↳b64encode(image_bytes).decode("utf8")
```

File ~/.local/share/virtualenvs/crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.py:134, in to_image(fig, format, width, height, scale, validate, engine)

```

    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=131'>132</a>      # Raise informative error message if Kaleido is not
↳installed
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=132'>133</a>      if scope is None:
--> <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=133'>134</a>      raise ValueError(
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=134'>135</a>      """
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=135'>136</a> Image export using the "kaleido" engine requires the
↳kaleido package,
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=136'>137</a> which can be installed using pip:
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=137'>138</a>      $ pip install -U kaleido
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=138'>139</a>      """
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=139'>140</a>      )
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=141'>142</a>      # Validate figure
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=142'>143</a>      # -----
    <a href='file:///home/julio/.local/share/virtualenvs/
↳crypto-deep-learning-wXDHGxS_/lib/python3.8/site-packages/plotly/io/_kaleido.
↳py?line=143'>144</a>      fig_dict = validate_coerce_fig_to_dict(fig, validate

```

ValueError:

Image export using the "kaleido" engine requires the kaleido package,
which can be installed using pip:
\$ pip install -U kaleido

0.2 Cardano

```
[ ]: print(f"Min Date: {df[df.Symbol == 'ADA'].Date.min()}, Max Date: {df[df.Symbol_
↳== 'ADA'].Date.max()}")
```

Min Date: 2017-10-02 23:59:59, Max Date: 2021-07-06 23:59:59

```
[ ]: df[df.Symbol == 'ADA'].describe()
```

```
[ ]:
```

| | SNo | High | Low | Open | Close \ |
|-------|-------------|-------------|-------------|-------------|-------------|
| count | 1374.000000 | 1374.000000 | 1374.000000 | 1374.000000 | 1374.000000 |
| mean | 687.500000 | 0.269807 | 0.239710 | 0.255287 | 0.256313 |
| std | 396.783946 | 0.433523 | 0.380928 | 0.408456 | 0.409691 |
| min | 1.000000 | 0.021050 | 0.017620 | 0.018414 | 0.018539 |
| 25% | 344.250000 | 0.047565 | 0.044601 | 0.045898 | 0.045947 |
| 50% | 687.500000 | 0.090274 | 0.083164 | 0.086867 | 0.087002 |
| 75% | 1030.750000 | 0.194519 | 0.172442 | 0.181374 | 0.183379 |
| max | 1374.000000 | 2.461766 | 2.013285 | 2.300190 | 2.309113 |

| | Volume | Marketcap |
|-------|--------------|--------------|
| count | 1.374000e+03 | 1.374000e+03 |
| mean | 8.934183e+08 | 7.603454e+09 |
| std | 2.107653e+09 | 1.303878e+10 |
| min | 1.739460e+06 | 4.806646e+08 |
| 25% | 5.014830e+07 | 1.191263e+09 |
| 50% | 1.186742e+08 | 2.270889e+09 |
| 75% | 4.875977e+08 | 5.174547e+09 |
| max | 1.914198e+10 | 7.377224e+10 |

```
[ ]: fig = go.Figure(data=go.Ohlc(x=df[df.Symbol == 'ADA']['Date'],
                                open=df[df.Symbol == 'ADA']['Open'],
                                high=df[df.Symbol == 'ADA']['High'],
                                low=df[df.Symbol == 'ADA']['Low'],
                                close=df[df.Symbol == 'ADA']['Close']))
fig.update_layout(
    title='Cardano OHLC',
    yaxis_title='Stock Price (USD)'
)
fig.show()
```

0.3 Ethereum

```
[ ]: print(f"Min Date: {df[df.Symbol == 'ETH'].Date.min()}, Max Date: {df[df.Symbol == 'ETH'].Date.max()}")
```

Min Date: 2015-08-08 23:59:59, Max Date: 2021-07-06 23:59:59

```
[ ]: df[df.Symbol == 'ETH'].describe()
```

```
[ ]:
```

| | SNo | High | Low | Open | Close \ |
|-------|-------------|-------------|-------------|-------------|-------------|
| count | 2160.000000 | 2160.000000 | 2160.000000 | 2160.000000 | 2160.000000 |
| mean | 1080.500000 | 398.258568 | 365.592589 | 382.879899 | 383.910691 |
| std | 623.682612 | 628.082281 | 566.611523 | 599.719862 | 601.078766 |
| min | 1.000000 | 0.482988 | 0.420897 | 0.431589 | 0.434829 |

| | | | | | |
|-----|-------------|-------------|-------------|-------------|-------------|
| 25% | 540.750000 | 14.265225 | 13.190950 | 13.757600 | 13.819200 |
| 50% | 1080.500000 | 205.124631 | 193.302715 | 198.425096 | 198.643691 |
| 75% | 1620.250000 | 396.494561 | 375.146804 | 386.264935 | 386.435272 |
| max | 2160.000000 | 4362.350542 | 3785.848603 | 4174.635873 | 4168.701049 |

| | Volume | Marketcap |
|-------|--------------|--------------|
| count | 2.160000e+03 | 2.160000e+03 |
| mean | 7.057058e+09 | 4.172084e+10 |
| std | 1.064526e+10 | 6.909184e+10 |
| min | 1.021280e+05 | 3.221363e+07 |
| 25% | 3.825102e+07 | 1.135576e+09 |
| 50% | 2.148880e+09 | 2.070063e+10 |
| 75% | 9.629136e+09 | 4.231010e+10 |
| max | 8.448291e+10 | 4.828819e+11 |

```
[ ]: fig = go.Figure(data=go.Ohlc(x=df[df.Symbol == 'ETH']['Date'],
                                open=df[df.Symbol == 'ETH']['Open'],
                                high=df[df.Symbol == 'ETH']['High'],
                                low=df[df.Symbol == 'ETH']['Low'],
                                close=df[df.Symbol == 'ETH']['Close']))
fig.update_layout(
    title='Ethereum OHLC',
    yaxis_title='Stock Price (USD)'
)
fig.show()
```

0.4 Train Test Split

```
[ ]: date_threshold = '2021-04-01'
train = df[df.Date < date_threshold]
test = df[df.Date >= date_threshold]
print(f'Train: {train.shape}, Test: {test.shape}, Proportion: {train.shape[0] /
↳ df.shape[0]:.2%}')
```

Train: (6234, 10), Test: (291, 10), Proportion: 95.54%

0.5 Feature Engineering

TODO: Test min max scaler over windows

0.5.1 Columns Selection

```
[ ]: class ColumnDropTransformer():
    def __init__(self, columns):
        self.columns=columns

    def transform(self, X, y=None):
```



```

        return X.drop(self.columns,axis=1)

    def fit(self, X, y=None):
        return self

```

```
[ ]: df.columns
```

```
[ ]: Index(['SNo', 'Name', 'Symbol', 'Date', 'High', 'Low', 'Open', 'Close',
          'Volume', 'Marketcap'],
          dtype='object')
```

```
[ ]: minmax_transformer = Pipeline(steps=[
        ('minmax', MinMaxScaler())])

pipeline = Pipeline([
    ("column_dropper", ColumnDropTransformer(['SNo', 'Name'])),
    ('normalization', ColumnTransformer(
        remainder='passthrough',
        transformers=[
            ('mm', minmax_transformer , ['High', 'Low', 'Open', 'Close', 'Volume', 'Marketcap'])
        ])
    ])

```

```
[ ]: train = pipeline.fit_transform(train)
```

```
[ ]: train[0]
```

```
[ ]: array([0.0023906901581641727, 0.002259311720669535, 0.002195738911411989,
          0.0023598010859048656, 0.0, 0.001375737032009516, 'Bitcoin', 'BTC',
          Timestamp('2013-04-29 23:59:59')], dtype=object)
```

0.5.2 Coins Correlation

```
[ ]: bitcoin_train = train[train[:,-2] == 'BTC']
    cardano_train = train[train[:,-2] == 'ADA']
    ethereum_train = train[train[:,-2] == 'ETH']

```

```
[ ]: fig, ax = plt.subplots(figsize=(12, 6))

ax.plot(bitcoin_train[:,-1],bitcoin_train[:,3], label='bitcoin')
ax.plot(cardano_train[:,-1],cardano_train[:,3], label='cardano')
ax.plot(ethereum_train[:,-1],ethereum_train[:,3], label='ethereum')
ax.set(xlabel="Date",
      ylabel="Normalized Close Price",
      title="Coins Close Price",

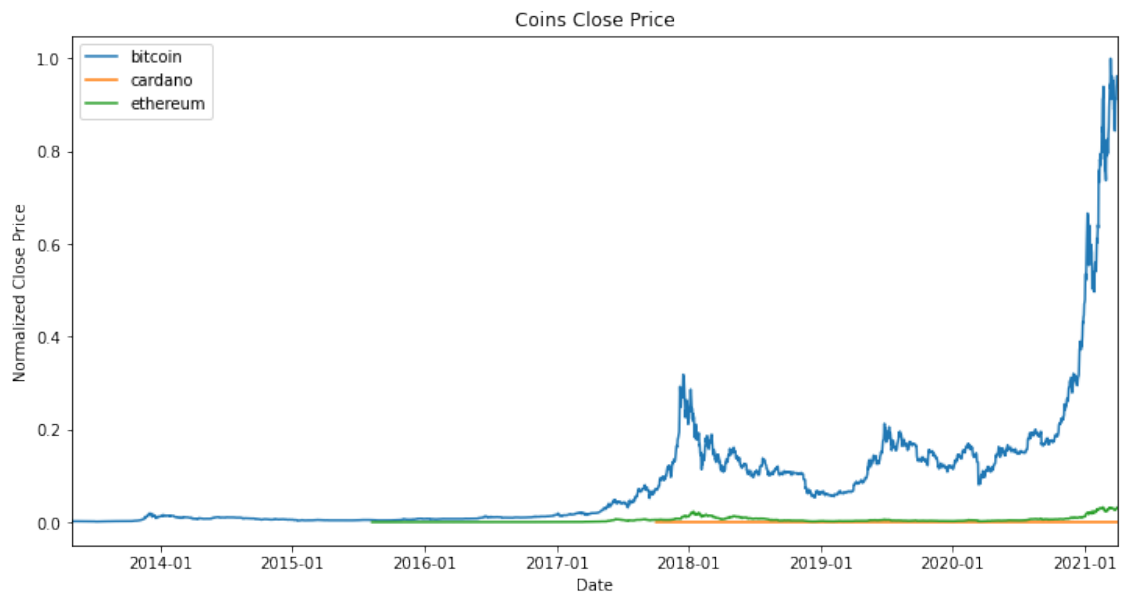
```

```

xlim=[min(train[:,-1]), max(train[:,-1])]
ax.legend(loc='best')

date_form = DateFormatter("%Y-%m")
ax.xaxis.set_major_formatter(date_form)
plt.show()

```



[]: