

05_conditional_autoencoder_for_asset_pricing_data

September 29, 2021

1 Conditional Autoencoder for Asset Pricing - Part 1: The Data

```
[1]: from pathlib import Path

import numpy as np
import pandas as pd

from statsmodels.regression.rolling import RollingOLS
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: idx = pd.IndexSlice
sns.set_style('whitegrid')
```

```
[3]: results_path = Path('results', 'asset_pricing')
if not results_path.exists():
    results_path.mkdir(parents=True)
```

1.1 Load Data

1.1.1 Prices

```
[4]: prices = pd.read_hdf(results_path / 'data.h5', 'stocks/prices/adjusted')
```

```
[5]: prices.info(show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 17661451 entries, ('A', Timestamp('1999-11-18 00:00:00')) to
('ZYXI', Timestamp('2019-12-31 00:00:00'))
Data columns (total 5 columns):
#   Column    Non-Null Count  Dtype
---  -
0   close     17661451 non-null  float64
1   high      17661451 non-null  float64
2   low       17661451 non-null  float64
3   open      17661451 non-null  float64
4   volume    17661451 non-null  float64
```

```
dtypes: float64(5)
memory usage: 742.0+ MB
```

1.1.2 Metadata

```
[6]: metadata = pd.read_hdf(results_path / 'data.h5', 'stocks/info').
      ↪rename(columns=str.lower)
```

```
[7]: metadata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6262 entries, A to ZYXI
Columns: 109 entries, zip to impliedsharesoutstanding
dtypes: bool(2), float64(75), int64(3), object(29)
memory usage: 5.2+ MB
```

1.1.3 Select tickers with metadata

```
[8]: sectors = (metadata.sector.value_counts() > 50).index
```

```
[9]: tickers_with_errors = ['FTAI', 'AIRT', 'CYBR', 'KTB']
```

```
[10]: tickers_with_metadata = metadata[metadata.sector.isin(sectors) &
                                       metadata.marketcap.notnull() &
                                       metadata.sharesoutstanding.notnull() &
                                       (metadata.sharesoutstanding > 0)].index.
      ↪drop(tickers_with_errors)
```

```
[11]: metadata = metadata.loc[tickers_with_metadata, ['sector', 'sharesoutstanding',
      ↪'marketcap']]
      metadata.index.name = 'ticker'
```

```
[12]: prices = prices.loc[idx[tickers_with_metadata, :], :]
```

```
[13]: prices.info(null_counts=True)
```

```
<ipython-input-13-e4642b41f34d>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
      prices.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 17312229 entries, ('A', Timestamp('1999-11-18 00:00:00')) to
('ZYXI', Timestamp('2019-12-31 00:00:00'))
Data columns (total 5 columns):
#   Column  Non-Null Count  Dtype
---  -
0   close   17312229 non-null  float64
1   high    17312229 non-null  float64
```

```

2   low      17312229 non-null float64
3   open     17312229 non-null float64
4   volume   17312229 non-null float64
dtypes: float64(5)
memory usage: 727.4+ MB

```

```
[14]: metadata.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 5749 entries, A to ZYXI
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   sector                 5749 non-null   object
1   sharesoutstanding      5749 non-null   float64
2   marketcap              5749 non-null   float64
dtypes: float64(2), object(1)
memory usage: 179.7+ KB

```

```
[15]: close = prices.close.unstack('ticker').sort_index()
close.info()
```

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 7559 entries, 1990-01-02 to 2019-12-31
Columns: 4420 entries, A to ZYXI
dtypes: float64(4420)
memory usage: 255.0 MB

```

```
[16]: volume = prices.volume.unstack('ticker').sort_index()
volume.info()
```

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 7559 entries, 1990-01-02 to 2019-12-31
Columns: 4420 entries, A to ZYXI
dtypes: float64(4420)
memory usage: 255.0 MB

```

1.1.4 Create weekly returns

```
[17]: returns = (prices.close
                .unstack('ticker')
                .resample('W-FRI').last()
                .sort_index().pct_change().iloc[1:])
returns.info()
```

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1565 entries, 1990-01-12 to 2020-01-03
Freq: W-FRI

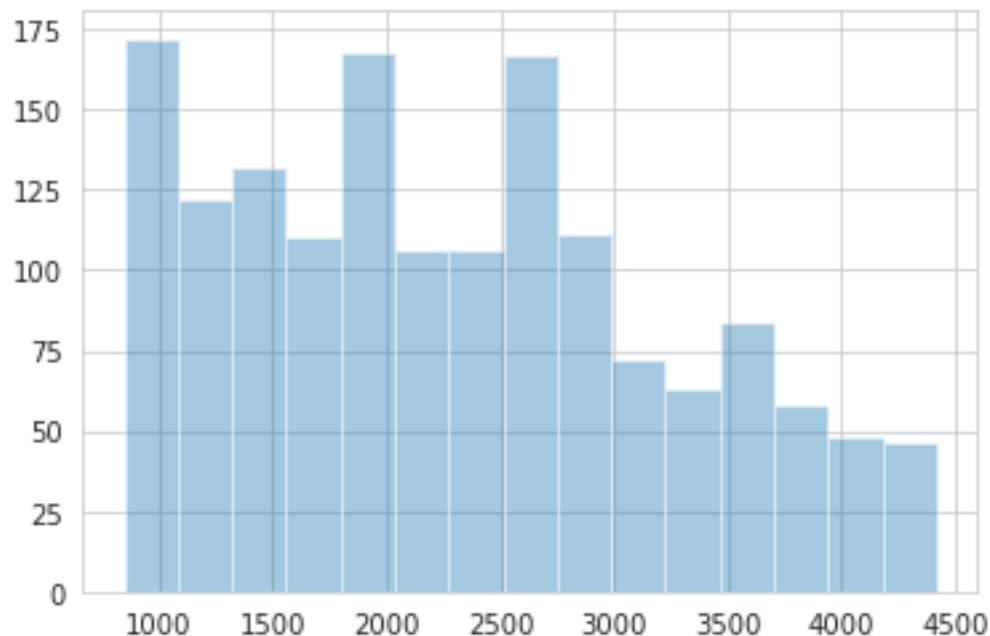
```

```
Columns: 4420 entries, A to ZYXI
dtypes: float64(4420)
memory usage: 52.8 MB
```

```
[18]: dates = returns.index
```

```
[19]: sns.distplot(returns.count(1), kde=False);
```

```
/home/stefan/.pyenv/versions/miniconda3-latest/envs/ml4t-dl/lib/python3.8/site-
packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



```
[20]: with pd.HDFStore(results_path / 'autoencoder.h5') as store:
      store.put('close', close)
      store.put('volume', volume)
      store.put('returns', returns)
      store.put('metadata', metadata)
```

1.2 Factor Engineering

```
[21]: MONTH = 21
```

1.2.1 Price Trend

Short-Term Reversal 1-month cumulative return

```
[22]: dates[:5]
```

```
[22]: DatetimeIndex(['1990-01-12', '1990-01-19', '1990-01-26', '1990-02-02',  
                    '1990-02-09'],  
                  dtype='datetime64[ns]', name='date', freq='W-FRI')
```

```
[23]: mom1m = close.pct_change( periods=MONTH ).resample('W-FRI').last().stack().  
      ↪to_frame('mom1m')  
      mom1m.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
MultiIndex: 3580621 entries, (Timestamp('1990-02-02 00:00:00', freq='W-FRI'),  
                              'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')  
Data columns (total 1 columns):  
#   Column  Dtype  
---  ---  
0   mom1m   float64  
dtypes: float64(1)  
memory usage: 41.2+ MB
```

```
[24]: mom1m.squeeze().to_hdf(results_path / 'autoencoder.h5', 'factor/mom1m')
```

Stock Momentum 11-month cumulative returns ending 1-month before month end

```
[25]: mom12m = (close  
                .pct_change( periods=11 * MONTH )  
                .shift(MONTH)  
                .resample('W-FRI')  
                .last()  
                .stack()  
                .to_frame('mom12m'))
```

```
[26]: mom12m.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>  
MultiIndex: 3375489 entries, (Timestamp('1991-01-04 00:00:00', freq='W-FRI'),  
                              'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')  
Data columns (total 1 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   mom12m  3375489 non-null  float64  
dtypes: float64(1)  
memory usage: 38.8+ MB
```

```
<ipython-input-26-8a049b23d2aa>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
mom12m.info(null_counts=True)
```

```
[27]: mom12m.to_hdf(results_path / 'autoencoder.h5', 'factor/mom12m')
```

Momentum Change Cumulative return from months t-6 to t-1 minus months t-12 to t-7.

```
[28]: chmom = (close
            .pct_change(periods=6 * MONTH)
            .sub(close.pct_change(periods=6 * MONTH).shift(6 * MONTH))
            .resample('W-FRI')
            .last()
            .stack()
            .to_frame('chmom'))
```

```
[29]: chmom.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 3375489 entries, (Timestamp('1991-01-04 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0    chmom   3375489 non-null    float64
dtypes: float64(1)
memory usage: 38.8+ MB

<ipython-input-29-312a8747df17>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
chmom.info(null_counts=True)
```

```
[30]: chmom.to_hdf(results_path / 'autoencoder.h5', 'factor/chmom')
```

Industry Momentum Equal-weighted avg. industry 12-month returns

```
[31]: indmom = (close.pct_change(12*MONTH)
                .resample('W-FRI')
                .last()
                .stack()
                .to_frame('close')
                .join(metadata[['sector']]).groupby(['date', 'sector'])
                .close.mean()
                .to_frame('indmom')
                .reset_index())
```

```
[32]: indmom.info(null_counts=True)
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18495 entries, 0 to 18494
Data columns (total 3 columns):
#   Column   Non-Null Count  Dtype
---  -
0   date     18495 non-null  datetime64[ns]
1   sector   18495 non-null  object
2   indmom    18495 non-null  float64
dtypes: datetime64[ns](1), float64(1), object(1)
memory usage: 433.6+ KB

<ipython-input-32-fcaeea0a7b0b>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
    indmom.info(null_counts=True)

```

```

[33]: indmom = (returns
        .stack()
        .to_frame('ret')
        .join(metadata[['sector']])
        .reset_index()
        .merge(indmom)
        .set_index(['date', 'ticker'])
        .loc[:, ['indmom']])

```

```

[34]: indmom.info(null_counts=True)

```

```

<class 'pandas.core.frame.DataFrame'>
MultiIndex: 3551199 entries, (Timestamp('1991-01-04 00:00:00'), 'AA') to
(Timestamp('2020-01-03 00:00:00'), 'ZTR')
Data columns (total 1 columns):
#   Column   Non-Null Count  Dtype
---  -
0   indmom    3551199 non-null  float64
dtypes: float64(1)
memory usage: 40.8+ MB

<ipython-input-34-fcaeea0a7b0b>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
    indmom.info(null_counts=True)

```

```

[35]: indmom.to_hdf(results_path / 'autoencoder.h5', 'factor/indmom')

```

Recent Max Return Max daily returns from calendar month t-1

```

[36]: maxret = (close
        .pct_change( periods=MONTH)
        .rolling(21)
        .max()

```

```
.resample('W-FRI')
.last()
.stack()
.to_frame('maxret'))
```

```
[37]: maxret.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 3562402 entries, (Timestamp('1990-03-02 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0    maxret  3562402 non-null    float64
dtypes: float64(1)
memory usage: 41.0+ MB

<ipython-input-37-ac905a38795f>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
maxret.info(null_counts=True)
```

```
[38]: maxret.to_hdf(results_path / 'autoencoder.h5', 'factor/maxret')
```

Long-Term Reversal Cumulative returns months t-36 to t-13.

```
[39]: mom36m = (close
                .pct_change(periods=24*MONTH)
                .shift(12*MONTH)
                .resample('W-FRI')
                .last()
                .stack()
                .to_frame('mom36m'))
```

```
[40]: mom36m.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 2967391 entries, (Timestamp('1993-01-01 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0    mom36m  2967391 non-null    float64
dtypes: float64(1)
memory usage: 34.2+ MB

<ipython-input-40-44b3a6a0df39>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
mom36m.info(null_counts=True)
```



```
[41]: mom36m.to_hdf(results_path / 'autoencoder.h5', 'factor/mom36m')
```

1.2.2 Liquidity Metrics

Turnover Avg. monthly trading volume for most recent three months scaled by number of shares; we are using the most recent no of shares from yahoo finance

```
[42]: turn = (volume
            .rolling(3*MONTH)
            .mean()
            .resample('W-FRI')
            .last()
            .div(metadata.sharesoutstanding)
            .stack('ticker')
            .to_frame('turn'))
```

```
[43]: turn.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 3506569 entries, (Timestamp('1990-03-30 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0   turn    3506569 non-null    float64
dtypes: float64(1)
memory usage: 40.3+ MB

<ipython-input-43-1b68d28a79cd>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
    turn.info(null_counts=True)
```

```
[44]: turn.to_hdf(results_path / 'autoencoder.h5', 'factor/turn')
```

Turnover Volatility Monthly std dev of daily share turnover

```
[45]: turn_std = (prices
                .volume
                .unstack('ticker')
                .div(metadata.sharesoutstanding)
                .rolling(MONTH)
                .std()
                .resample('W-FRI')
                .last()
                .stack('ticker')
                .to_frame('turn_std'))
```

```
[46]: turn_std.to_hdf(results_path / 'autoencoder.h5', 'factor/turn_std')
```

Log Market Equity Natural log of market cap at end of month t-1

```
[47]: last_price = close.ffill()
      factor = close.div(last_price.iloc[-1])
      mvel = np.log1p(factor.mul(metadata.marketcap).resample('W-FRI').last()).
      ↪stack().to_frame('mvel')
```

```
[48]: mvel.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 3597636 entries, (Timestamp('1990-01-05 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0    mvel    3597636 non-null    float64
dtypes: float64(1)
memory usage: 41.4+ MB

<ipython-input-48-2c361336080a>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
      mvel.info(null_counts=True)
```

```
[49]: mvel.to_hdf(results_path / 'autoencoder.h5', 'factor/mvel')
```

Dollar Volume Natural log of trading volume time price per share from month t-2

```
[50]: dv = close.mul(volume)
```

```
[51]: dolvol = (np.log1p(dv.rolling(21)
                        .mean()
                        .shift(21)
                        .resample('W-FRI')
                        .last())
            .stack()
            .to_frame('dolvol'))
```

```
[52]: dolvol.to_hdf(results_path / 'autoencoder.h5', 'factor/dolvol')
```

Amihud Illiquidity Average of daily (absolute return / dollar volume)

```
[53]: ill = (close.pct_change().abs()
            .div(dv)
            .rolling(21)
            .mean()
            .resample('W-FRI').last()
            .stack()
            .to_frame('ill'))
```

```
[54]: ill.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 3210773 entries, (Timestamp('1990-02-02 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0   ill      3210773 non-null   float64
dtypes: float64(1)
memory usage: 36.9+ MB

<ipython-input-54-d1823ec8761b>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
    ill.info(null_counts=True)
```

```
[55]: ill.to_hdf(results_path / 'autoencoder.h5', 'factor/ill')
```

1.2.3 Risk Measures

Return Volatility Standard dev of daily returns from month t-1.

```
[56]: retvol = (close.pct_change()
               .rolling(21)
               .std()
               .resample('W-FRI')
               .last()
               .stack()
               .to_frame('retvol'))
```

```
[57]: retvol.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 3580621 entries, (Timestamp('1990-02-02 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0   retvol  3580621 non-null   float64
dtypes: float64(1)
memory usage: 41.2+ MB

<ipython-input-57-b187f925aef0>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
    retvol.info(null_counts=True)
```

```
[58]: retvol.to_hdf(results_path / 'autoencoder.h5', 'factor/retvol')
```

Market Beta Estimated market beta from weekly returns and equal weighted market returns for 3 years ending month t-1 with at least 52 weeks of returns.

```
[59]: index = close.resample('W-FRI').last().pct_change().mean(1).to_frame('x')
```

```
[60]: def get_market_beta(y, x=index):
      df = x.join(y.to_frame('y')).dropna()
      model = RollingOLS(endog=df.y,
                        exog=sm.add_constant(df[['x']]),
                        window=3*52)

      return model.fit(params_only=True).params['x']
```

```
[61]: beta = (returns.dropna(thresh=3*52, axis=1)
             .apply(get_market_beta).stack().to_frame('beta'))
```

```
[62]: beta.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 2969406 entries, (Timestamp('1993-01-01 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0    beta    2969406 non-null    float64
dtypes: float64(1)
memory usage: 34.2+ MB

<ipython-input-62-f1c77092070c>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
      beta.info(null_counts=True)
```

```
[63]: beta.to_hdf(results_path / 'autoencoder.h5', 'factor/beta')
```

Beta Squared Market beta squared

```
[64]: betasq = beta.beta.pow(2).to_frame('betasq')
```

```
[65]: betasq.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 2969406 entries, (Timestamp('1993-01-01 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0    betasq  2969406 non-null    float64
```

```

dtypes: float64(1)
memory usage: 34.2+ MB

<ipython-input-65-5559d38d2dd2>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
    betasq.info(null_counts=True)

```

```
[66]: betasq.to_hdf(results_path / 'autoencoder.h5', 'factor/betasq')
```

Idiosyncratic return volatility Standard dev of a regression of residuals of weekly returns on the returns of an equal weighted market index returns for the prior three years.

This takes a while!

```
[67]: def get_ols_residuals(y, x=index):
      df = x.join(y.to_frame('y')).dropna()
      model = sm.OLS(endog=df.y, exog=sm.add_constant(df[['x']]))
      result = model.fit()
      return result.resid.std()

```

```
[68]: idiovol = (returns.apply(lambda x: x.rolling(3 * 52)
                             .apply(get_ols_residuals)))

```

```
[69]: idiovol = idiovol.stack().to_frame('idiovol')
```

```
[70]: idiovol.info(null_counts=True)
```

```

<class 'pandas.core.frame.DataFrame'>
MultiIndex: 2969406 entries, (Timestamp('1993-01-01 00:00:00', freq='W-FRI'),
'AA') to (Timestamp('2020-01-03 00:00:00', freq='W-FRI'), 'ZYXI')
Data columns (total 1 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   idiovol     2969406 non-null    float64
dtypes: float64(1)
memory usage: 34.2+ MB

<ipython-input-70-eec938dfce7b>:1: FutureWarning: null_counts is deprecated. Use
show_counts instead
    idiovol.info(null_counts=True)

```

```
[71]: idiovol.to_hdf(results_path / 'autoencoder.h5', 'factor/idiovol')
```