02_mutual_information

September 29, 2021

1 Using information theory to evaluate features

The mutual information (MI) between a feature and the outcome is a measure of the mutual dependence between the two variables. It extends the notion of correlation to nonlinear relationships. More specifically, it quantifies the information obtained about one random variable through the other random variable.

The concept of MI is closely related to the fundamental notion of entropy of a random variable. Entropy quantifies the amount of information contained in a random variable. Formally, the mutual information—I(X, Y)—of two random variables, X and Y, is defined as the following:

The sklearn function implements feature_selection.mutual_info_regression that computes the mutual information between all features and a continuous outcome to select the features that are most likely to contain predictive information. There is also a classification version (see the documentation for more details).

This notebook contains an application to the financial data we created in Chapter 4, Alpha Factor Research.

[1]:
```
%matplotlib inline

import warnings
from datetime import datetime
import os
from pathlib import Path
import quandl
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pandas_datareader.data as web
from pandas_datareader.famafrench import get_available_datasets
from pyfinance.ols import PandasRollingOLS
from sklearn.feature_selection import mutual_info_classif
```

[2]:
```
warnings.filterwarnings('ignore')
plt.style.use('fivethirtyeight')
idx = pd.IndexSlice
```

1.1 Get Data

We use the data produced in Chapter 4.

[6]:
```
with pd.HDFStore('../data/assets.h5') as store:
    data = store['engineered_features']
```

1.2 Create Dummy variables

[7]:
```
dummy_data = pd.get_dummies(data,
                            columns=['year','month', 'msize', 'age',  'sector'],
                            prefix=['year','month', 'msize', 'age', ''],
                            prefix_sep=['_', '_', '_', '_', ''])
dummy_data = dummy_data.rename(columns={c:c.replace('.0', '') for c in␣
 ↪dummy_data.columns})
dummy_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 445640 entries, (A, 2001-01-31 00:00:00) to (ZUMZ, 2018-02-28
00:00:00)
Data columns (total 89 columns):
return_1m                445640 non-null float64
return_2m                445640 non-null float64
return_3m                445640 non-null float64
return_6m                445640 non-null float64
return_9m                445640 non-null float64
return_12m               445640 non-null float64
CMA                      445640 non-null float64
HML                      445640 non-null float64
Mkt-RF                   445640 non-null float64
RMW                      445640 non-null float64
SMB                      445640 non-null float64
momentum_2               445640 non-null float64
momentum_3               445640 non-null float64
momentum_6               445640 non-null float64
momentum_9               445640 non-null float64
momentum_12              445640 non-null float64
momentum_3_12            445640 non-null float64
return_1m_t-1            443312 non-null float64
return_1m_t-2            440984 non-null float64
return_1m_t-3            438656 non-null float64
return_1m_t-4            436328 non-null float64
return_1m_t-5            434000 non-null float64
return_1m_t-6            431672 non-null float64
target_1m                445189 non-null float64
target_2m                442861 non-null float64
target_3m                440533 non-null float64
target_6m                433549 non-null float64
```

```
target_12m                419581 non-null float64
year_2001                 445640 non-null uint8
year_2002                 445640 non-null uint8
year_2003                 445640 non-null uint8
year_2004                 445640 non-null uint8
year_2005                 445640 non-null uint8
year_2006                 445640 non-null uint8
year_2007                 445640 non-null uint8
year_2008                 445640 non-null uint8
year_2009                 445640 non-null uint8
year_2010                 445640 non-null uint8
year_2011                 445640 non-null uint8
year_2012                 445640 non-null uint8
year_2013                 445640 non-null uint8
year_2014                 445640 non-null uint8
year_2015                 445640 non-null uint8
year_2016                 445640 non-null uint8
year_2017                 445640 non-null uint8
year_2018                 445640 non-null uint8
month_1                   445640 non-null uint8
month_2                   445640 non-null uint8
month_3                   445640 non-null uint8
month_4                   445640 non-null uint8
month_5                   445640 non-null uint8
month_6                   445640 non-null uint8
month_7                   445640 non-null uint8
month_8                   445640 non-null uint8
month_9                   445640 non-null uint8
month_10                  445640 non-null uint8
month_11                  445640 non-null uint8
month_12                  445640 non-null uint8
msize_-1                  445640 non-null uint8
msize_1                   445640 non-null uint8
msize_2                   445640 non-null uint8
msize_3                   445640 non-null uint8
msize_4                   445640 non-null uint8
msize_5                   445640 non-null uint8
msize_6                   445640 non-null uint8
msize_7                   445640 non-null uint8
msize_8                   445640 non-null uint8
msize_9                   445640 non-null uint8
msize_10                  445640 non-null uint8
age_-1                    445640 non-null uint8
age_0                     445640 non-null uint8
age_1                     445640 non-null uint8
age_2                     445640 non-null uint8
age_3                     445640 non-null uint8
age_4                     445640 non-null uint8
```

```
age_5                    445640 non-null uint8
Basic Industries         445640 non-null uint8
Capital Goods            445640 non-null uint8
Consumer Durables        445640 non-null uint8
Consumer Non-Durables    445640 non-null uint8
Consumer Services        445640 non-null uint8
Energy                   445640 non-null uint8
Finance                  445640 non-null uint8
Health Care              445640 non-null uint8
Miscellaneous            445640 non-null uint8
Public Utilities         445640 non-null uint8
Technology               445640 non-null uint8
Transportation           445640 non-null uint8
Unknown                  445640 non-null uint8
dtypes: float64(28), uint8(61)
memory usage: 122.8+ MB
```

1.3 Mutual Information

1.3.1 Original Data

[5]:
```
target_labels = [f'target_{i}m' for i in [1,2,3,6,12]]
targets = data.dropna().loc[:, target_labels]

features = data.dropna().drop(target_labels, axis=1)
features.sector = pd.factorize(features.sector)[0]

cat_cols = ['year', 'month', 'msize', 'age', 'sector']
discrete_features = [features.columns.get_loc(c) for c in cat_cols]
```

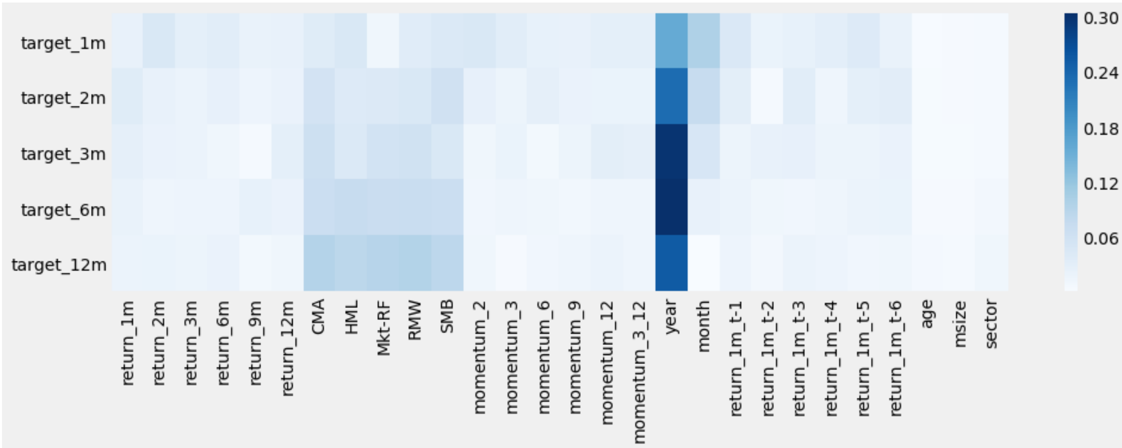
[6]:
```
mutual_info = pd.DataFrame()
for label in target_labels:
    mi = mutual_info_classif(X=features,
                             y=(targets[label]> 0).astype(int),
                             discrete_features=discrete_features,
                             random_state=42
                            )
    mutual_info[label] = pd.Series(mi, index=features.columns)
```

[7]:
```
mutual_info.sum()
```

[7]:
```
target_1m     0.053056
target_2m     0.079370
target_3m     0.102913
target_6m     0.156282
target_12m    0.221293
dtype: float64
```

1.3.2 Normalized MI Heatmap

[8]:
```python
fig, ax= plt.subplots(figsize=(15, 4))
sns.heatmap(mutual_info.div(mutual_info.sum()).T, ax=ax, cmap='Blues');
```



1.3.3 Dummy Data

[9]:
```python
target_labels = [f'target_{i}m' for i in [1, 2, 3, 6, 12]]
dummy_targets = dummy_data.dropna().loc[:, target_labels]

dummy_features = dummy_data.dropna().drop(target_labels, axis=1)
cat_cols = [c for c in dummy_features.columns if c not in features.columns]
discrete_features = [dummy_features.columns.get_loc(c) for c in cat_cols]
```

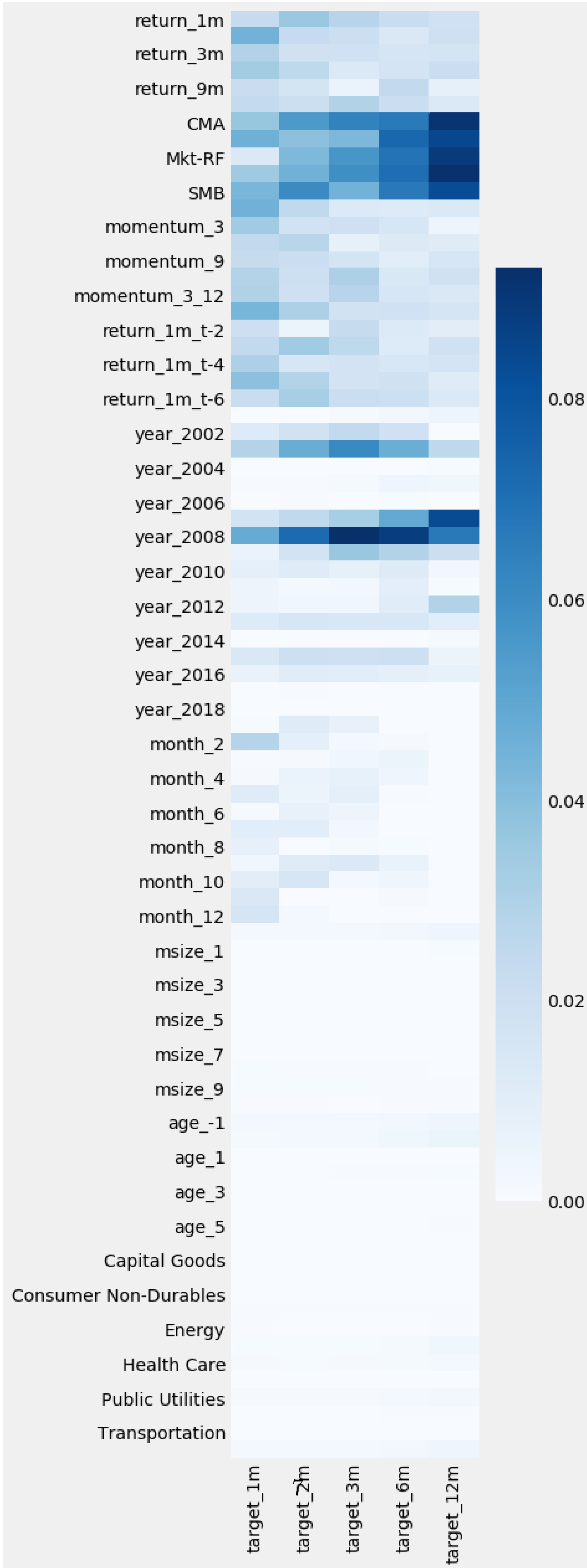
[10]:
```python
mutual_info_dummies = pd.DataFrame()
for label in target_labels:
    mi = mutual_info_classif(X=dummy_features,
                             y=(dummy_targets[label]> 0).astype(int),
                             discrete_features=discrete_features,
                             random_state=42
                            )
    mutual_info_dummies[label] = pd.Series(mi, index=dummy_features.columns)
```

[11]:
```python
mutual_info_dummies.sum()
```

[11]:
```
target_1m     0.054261
target_2m     0.081427
target_3m     0.105798
target_6m     0.160603
target_12m    0.226720
dtype: float64
```

[12]:

```
fig, ax= plt.subplots(figsize=(4, 20))
sns.heatmap(mutual_info_dummies.div(mutual_info_dummies.sum()), ax=ax,␣
↪cmap='Blues');
```


return_1m
return_3m
return_9m
CMA
Mkt-RF
SMB
momentum_3
momentum_9
momentum_3_12
return_1m_t-2
return_1m_t-4
return_1m_t-6
year_2002
year_2004
year_2006
year_2008
year_2010
year_2012
year_2014
year_2016
year_2018
month_2
month_4
month_6
month_8
month_10
month_12
msize_1
msize_3
msize_5
msize_7
msize_9
age_-1
age_1
age_3
age_5
Capital Goods
Consumer Non-Durables
Energy
Health Care
Public Utilities
Transportation
target_1m
target_2m
target_3m
target_6m
target_12m
0.08
0.06
0.04
0.02
0.00

```
[ ]:
```