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]:
```
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
%matplotlib inline

import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.feature_selection import mutual_info_classif
```

[3]:
```
sns.set_style('whitegrid')
idx = pd.IndexSlice
```

1.1 Get Data

We use the data produced in Chapter 4.

[4]:
```
with pd.HDFStore('../data/assets.h5') as store:
    data = store['engineered_features']
```

1.2 Create Dummy variables

[5]:
```
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: 358914 entries, ('A', Timestamp('2001-01-31 00:00:00')) to ('ZUMZ',
Timestamp('2018-02-28 00:00:00'))
Data columns (total 88 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   return_1m              358914 non-null  float64
 1   return_2m              358914 non-null  float64
 2   return_3m              358914 non-null  float64
 3   return_6m              358914 non-null  float64
 4   return_9m              358914 non-null  float64
 5   return_12m             358914 non-null  float64
 6   Mkt-RF                 358914 non-null  float64
 7   SMB                    358914 non-null  float64
 8   HML                    358914 non-null  float64
 9   RMW                    358914 non-null  float64
 10  CMA                    358914 non-null  float64
 11  momentum_2             358914 non-null  float64
 12  momentum_3             358914 non-null  float64
 13  momentum_6             358914 non-null  float64
 14  momentum_9             358914 non-null  float64
 15  momentum_12            358914 non-null  float64
 16  momentum_3_12          358914 non-null  float64
 17  return_1m_t-1          357076 non-null  float64
 18  return_1m_t-2          355238 non-null  float64
 19  return_1m_t-3          353400 non-null  float64
 20  return_1m_t-4          351562 non-null  float64
 21  return_1m_t-5          349724 non-null  float64
 22  return_1m_t-6          347886 non-null  float64
 23  target_1m              358914 non-null  float64
 24  target_2m              357076 non-null  float64
 25  target_3m              355238 non-null  float64
 26  target_6m              349724 non-null  float64
 27  target_12m             338696 non-null  float64
 28  year_2001              358914 non-null  uint8
 29  year_2002              358914 non-null  uint8
 30  year_2003              358914 non-null  uint8
 31  year_2004              358914 non-null  uint8
```

```
32  year_2005              358914 non-null  uint8
33  year_2006              358914 non-null  uint8
34  year_2007              358914 non-null  uint8
35  year_2008              358914 non-null  uint8
36  year_2009              358914 non-null  uint8
37  year_2010              358914 non-null  uint8
38  year_2011              358914 non-null  uint8
39  year_2012              358914 non-null  uint8
40  year_2013              358914 non-null  uint8
41  year_2014              358914 non-null  uint8
42  year_2015              358914 non-null  uint8
43  year_2016              358914 non-null  uint8
44  year_2017              358914 non-null  uint8
45  year_2018              358914 non-null  uint8
46  month_1                358914 non-null  uint8
47  month_2                358914 non-null  uint8
48  month_3                358914 non-null  uint8
49  month_4                358914 non-null  uint8
50  month_5                358914 non-null  uint8
51  month_6                358914 non-null  uint8
52  month_7                358914 non-null  uint8
53  month_8                358914 non-null  uint8
54  month_9                358914 non-null  uint8
55  month_10               358914 non-null  uint8
56  month_11               358914 non-null  uint8
57  month_12               358914 non-null  uint8
58  msize_-1               358914 non-null  uint8
59  msize_1                358914 non-null  uint8
60  msize_2                358914 non-null  uint8
61  msize_3                358914 non-null  uint8
62  msize_4                358914 non-null  uint8
63  msize_5                358914 non-null  uint8
64  msize_6                358914 non-null  uint8
65  msize_7                358914 non-null  uint8
66  msize_8                358914 non-null  uint8
67  msize_9                358914 non-null  uint8
68  msize_10               358914 non-null  uint8
69  age_0                  358914 non-null  uint8
70  age_1                  358914 non-null  uint8
71  age_2                  358914 non-null  uint8
72  age_3                  358914 non-null  uint8
73  age_4                  358914 non-null  uint8
74  age_5                  358914 non-null  uint8
75  Basic Industries       358914 non-null  uint8
76  Capital Goods          358914 non-null  uint8
77  Consumer Durables      358914 non-null  uint8
78  Consumer Non-Durables  358914 non-null  uint8
79  Consumer Services      358914 non-null  uint8
```

```
 80  Energy                 358914 non-null  uint8
 81  Finance                358914 non-null  uint8
 82  Health Care            358914 non-null  uint8
 83  Miscellaneous          358914 non-null  uint8
 84  Public Utilities       358914 non-null  uint8
 85  Technology             358914 non-null  uint8
 86  Transportation         358914 non-null  uint8
 87  Unknown                358914 non-null  uint8
dtypes: float64(28), uint8(60)
memory usage: 98.7+ MB
```

1.3 Mutual Information

1.3.1 Original Data

[6]:
```
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]
```

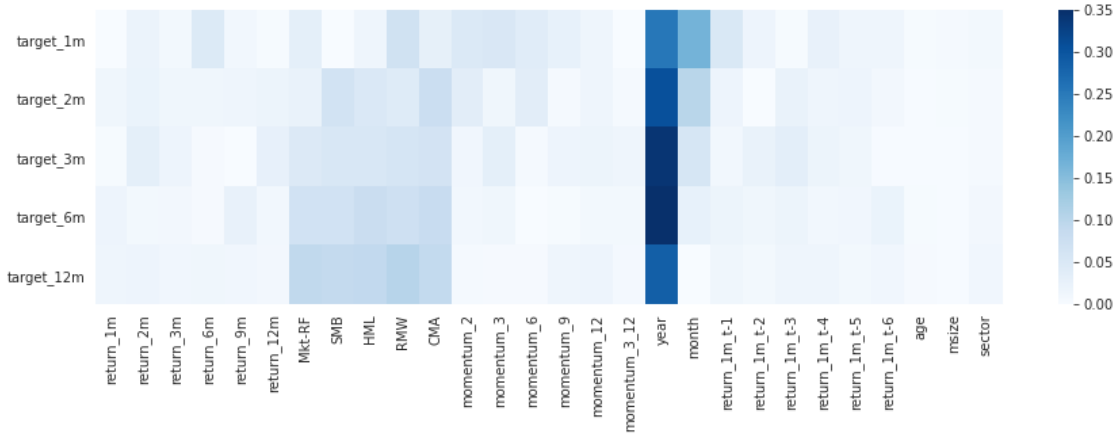
[7]:
```
mutual_info = pd.DataFrame()
for label in target_labels:
    mi = mutual_info_classif(X=features,
                             y=(targets[label]> 0).astype(int),
                             discrete_features=discrete_features,
                             random_state=42
                            )
    mutual_info[label] = pd.Series(mi, index=features.columns)
```

[8]:
```
mutual_info.sum()
```

[8]:
```
target_1m     0.032688
target_2m     0.060667
target_3m     0.090360
target_6m     0.136456
target_12m    0.198843
dtype: float64
```

1.3.2 Normalized MI Heatmap

[9]:
```
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

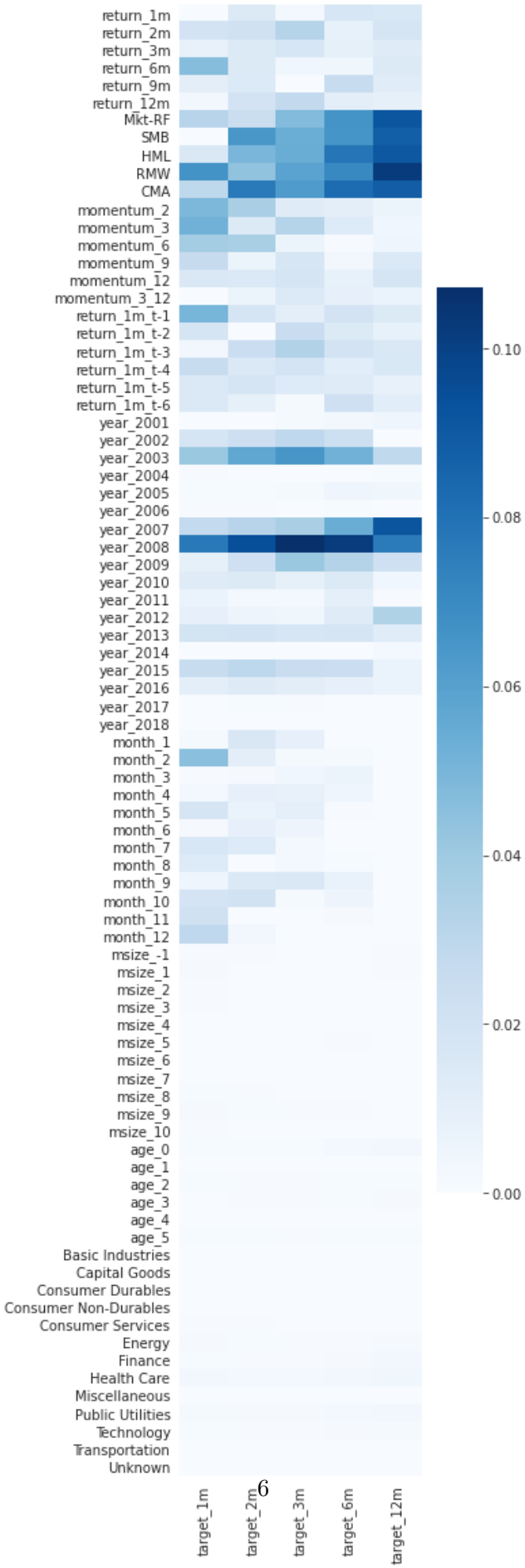
[10]:
```
target_labels = [f'target_{i}m' for i in [1, 2, 3, 6, 12]]
dummy_targets = dummy_data.dropna().loc[:, target_labels]

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]
```

[11]:
```
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)
```

[12]:
```
mutual_info_dummies.sum()
```

[12]:
```
target_1m     0.033829
target_2m     0.062598
target_3m     0.093065
target_6m     0.140429
target_12m    0.203577
dtype: float64
```

[13]:
```
fig, ax= plt.subplots(figsize=(4, 20))
sns.heatmap(mutual_info_dummies.div(mutual_info_dummies.sum()), ax=ax,␣
 ↪cmap='Blues');
```


return_1m
return_2m
return_3m
return_6m
return_9m
return_12m
Mkt-RF
SMB
HML
RMW
CMA
momentum_2
momentum_3
momentum_6
momentum_9
momentum_12
momentum_3_12
return_1m_t-1
return_1m_t-2
return_1m_t-3
return_1m_t-4
return_1m_t-5
return_1m_t-6
year_2001
year_2002
year_2003
year_2004
year_2005
year_2006
year_2007
year_2008
year_2009
year_2010
year_2011
year_2012
year_2013
year_2014
year_2015
year_2016
year_2017
year_2018
month_1
month_2
month_3
month_4
month_5
month_6
month_7
month_8
month_9
month_10
month_11
month_12
msize_-1
msize_1
msize_2
msize_3
msize_4
msize_5
msize_6
msize_7
msize_8
msize_9
msize_10
age_0
age_1
age_2
age_3
age_4
age_5
Basic Industries
Capital Goods
Consumer Durables
Consumer Non-Durables
Consumer Services
Energy
Finance
Health Care
Miscellaneous
Public Utilities
Technology
Transportation
Unknown
target_1m
target_2m
target_3m
target_6m
target_12m
0.10
0.08
0.06
0.04
0.02
0.00