feature_selection

May 15, 2020

[1]:
```
# store start time to get execution time of entire script
import time
start_time = time.time()
```

[2]:
```
import pandas as pd
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
import csv

df = pd.read_csv('data/data_superset.csv')
df = df.sample(frac=1).reset_index(drop=True)
df.head()
```

[2]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	ID	State	City	agyaddr	xobsyr_0	Illicit_Days5	Illicit_Cens5	female_cd	nonwhite_cd	unemplmt_cd	prsatx_cd	gvsg_cd	CWSg_0_cd	srprobg_cd	dssg_0_cd	epsg_0_cd	adhdg_0_cd	cdsg_0_cd	cjsig_0_cd
0	2427	7157	7161	16820	TX	Houston	237 Social Work Building	2011	196	0	0	1	0	0	0	0	0	0	0	0	0	1
1	9953	21076	21090	17753	AZ	Tucson	3130 E. Broadway Blvd, Suite 180	2011	176	0	1	1	0	0	2	0	1	2	2	1	1	0
2	5602	12216	12224	19420	TX	Laredo	2387 E. Saunders	2011	123	1	0	1	1	1	0	0						
3	3165	8711	8717	11503	TX	Laredo	2386 E. Saunders, Suite 2	2008	30	1	0	1	0	0	2	0						
4	6540	14056	14066	18639	CA	Downey	11500 Paramount Blvd	2010	365	0	0	1	0	0	1	0						

```
2          2          2          1          2          1          0
3          1          0          1          0          2          2
4          0          2          1          1          0          0

   lrig_0_cd  srig_0_cd  SESg_0_cd  r4ag_0_cd  primsev_cd_1  primsev_cd_2  \
0          2          2          2          0             0             0
1          1          2          0          1             0             0
2          2          2          2          2             0             0
3          1          1          0          0             0             1
4          2          2          0          2             0             0

   primsev_cd_3  primsev_cd_4  primsev_cd_5  primsev_cd_6  B2a_0g  SUDSy_0  \
0             1             0             0             0       1        0
1             1             0             0             0       1        8
2             0             0             1             0       2        3
3             0             0             0             0       0        6
4             1             0             0             0       2        1

                                      Address        lat         lng  \
0        237 Social Work Building, Houston, TX  29.948496  -95.470979
1  3130 E. Broadway Blvd, Suite 180, Tucson, AZ  32.221465 -110.926070
2                 2387 E. Saunders, Laredo, TX  27.530458  -99.472336
3        2386 E. Saunders, Suite 2, Laredo, TX  27.530608  -99.472335
4             11500 Paramount Blvd, Downey, CA  33.930804 -118.146842

   state_name  county_FIPS    block_FIPS  murder_numg  %_dropoutg  %_povertyg  \
0       Texas      48201.0  4.820155e+14            1         0.0         0.0
1     Arizona       4019.0  4.019002e+13            0         0.0         0.0
2       Texas      48479.0  4.847900e+14            0         0.0         1.0
3       Texas      48479.0  4.847900e+14            0         0.0         1.0
4  California       6037.0  6.037552e+13            0         0.0         0.0

   %_public_assistanceg  %_unemployedg                     closest  gran  \
0                   0.0            0.0                         NaN   0.0
1                   0.0            0.0                         NaN   1.0
2                   0.0            0.0                         NaN   0.0
3                   0.0            0.0  ('27.530608', '-99.472335')   2.0
4                   0.0            0.0                         NaN   1.0

                                 point  pop_deng
0   ('29.948496000000002', '-95.470979')       0.0
1  ('32.2214651', '-110.92607029999999')       0.0
2   ('27.530458000000003', '-99.472336')       0.0
3            ('27.530608', '-99.472335')       0.0
4  ('33.9308039', '-118.14684199999999')       0.0
```

[3]:
```
# drop unnecessary columns
cols_to_drop = ['Address','lat','lng','xobsyr_0','Unnamed: 0','Unnamed: 0.
↪1','Unnamed: 0.1.1',
                ␣
↪'ID','State','City','agyaddr','state_name','gran','srprobg_cd','county_FIPS','block_FIPS',
                 'point','closest','SUDSy_0']

df.drop(columns=cols_to_drop, inplace=True)
df.head()
```

[3]:

	Illicit_Days5	Illicit_Cens5	female_cd	nonwhite_cd	unemplmt_cd	\
0	196	0	0	1	0	
1	176	0	1	1	0	
2	123	1	0	1	1	
3	30	1	0	1	0	
4	365	0	0	1	0	

	prsatx_cd	gvsg_cd	CWSg_0_cd	dssg_0_cd	epsg_0_cd	adhdg_0_cd	cdsg_0_cd	\
0	0	0	0	0	0	0	0	
1	0	2	0	2	2	1	1	
2	1	0	0	2	1	2	1	
3	0	2	0	0	1	0	2	
4	0	1	0	2	1	1	0	

	cjsig_0_cd	lrig_0_cd	srig_0_cd	SESg_0_cd	r4ag_0_cd	primsev_cd_1	\
0	1	2	2	2	0	0	
1	0	1	2	0	1	0	
2	0	2	2	2	2	0	
3	2	1	1	0	0	0	
4	0	2	2	0	2	0	

	primsev_cd_2	primsev_cd_3	primsev_cd_4	primsev_cd_5	primsev_cd_6	\
0	0	1	0	0	0	
1	0	1	0	0	0	
2	0	0	0	1	0	
3	1	0	0	0	0	
4	0	1	0	0	0	

	B2a_0g	murder_numg	%_dropoutg	%_povertyg	%_public_assistanceg	\
0	1	1	0.0	0.0	0.0	
1	1	0	0.0	0.0	0.0	
2	2	0	0.0	1.0	0.0	
3	0	0	0.0	1.0	0.0	
4	2	0	0.0	0.0	0.0	

	%_unemployedg	pop_deng
0	0.0	0.0

```
1            0.0       0.0
2            0.0       0.0
3            0.0       0.0
4            0.0       0.0
```

[4]:
```
from sksurv.util import Surv

predictor_var = 'Illicit_Days5'
censoring_var = 'Illicit_Cens5'

X = df.copy()
Y = X[[censoring_var, predictor_var]]
X.drop(columns=[censoring_var, predictor_var], inplace=True)
y = Surv.from_arrays(Y[censoring_var], Y[predictor_var]) # structured array to␣
 ↪ensure correct censoring

print(X.shape, y.shape)
```

```
(10068, 28) (10068,)
```

[5]:
```
import numpy as np
from sklearn.model_selection import cross_validate

def forward_feature_selection(model, X, y):
    selected_features = []
    score = -1

    while (True):
        temp_score = -1
        next_feature = None
        for f in X.columns:
            if f not in selected_features:
                selected_features.append(f)
                temp = X[selected_features]

                scores = cross_validate(model, temp, y, cv=5)
                s = scores['test_score'].mean()
                #print(selected_features, '->', s)
                if s > temp_score:
                    temp_score = s
                    next_feature = f

                selected_features.pop()

        if temp_score <= score: # stop if no more change in improvement
            print('Optimal Features:', selected_features)
            print('Optimal Score:', score)
```

```
            break
        else:
            selected_features.append(next_feature)
            score = temp_score
        print('Added Feature:', next_feature)
        print('Score:', score)

    return selected_features
```

[6]:
```
def backward_feature_selection(model, X, y):
    selected_features = list(X.columns)
    score = -1

    while (True):
        temp_score = -1
        worst_feature = None
        for i in range(len(selected_features)):
            f = selected_features.pop()
            temp = X[selected_features]

            scores = cross_validate(model, temp, y, cv=5)
            s = scores['test_score'].mean()
            #print(selected_features, '->', s)
            if s > temp_score:
                temp_score = s
                worst_feature = f

            selected_features.insert(0, f)
        if temp_score <= score: # stop if no more change in improvement
            print('Optimal Features:', selected_features)
            print('Optimal Score:', score)
            break
        else:
            selected_features.remove(worst_feature)
            score = temp_score
        print('Dropped Feature:', worst_feature)
        print('Score:', score)

    return selected_features
```

[7]:
```
def feature_selection_statistics(FFS, BFS, features):
    stats = {}

    FFS = np.array(FFS)
    BFS = np.array(BFS)
    features = np.array(features)
```

```
    stats['Both_Used'] = list(np.intersect1d(FFS, BFS))
    stats['FFS_Used'] = list(np.setdiff1d(FFS, stats['Both_Used']))
    stats['BFS_Used'] = list(np.setdiff1d(BFS, stats['Both_Used']))
    stats['Both_Dropped'] = list(np.setdiff1d(features, np.union1d(FFS, BFS)))
    stats['FFS_Dropped'] = list(np.setdiff1d(features, np.union1d(FFS,␣
↪stats['Both_Dropped'])))
    stats['BFS_Dropped'] = list(np.setdiff1d(features, np.union1d(BFS,␣
↪stats['Both_Dropped'])))

    return stats
```

[8]:
```
%%time
from sksurv.linear_model import CoxnetSurvivalAnalysis

# l1_ratio = 1 adjusts model to implement LASSO method for penalties
rcr = CoxnetSurvivalAnalysis(l1_ratio=1)
rcr_BFS = backward_feature_selection(rcr, X, y)
```

```
Dropped Feature: pop_deng
Score: 0.6760885893770648
Dropped Feature: B2a_0g
Score: 0.676574105529251
Dropped Feature: female_cd
Score: 0.6768420522968134
Dropped Feature: murder_numg
Score: 0.677134228100414
Dropped Feature: lrig_0_cd
Score: 0.6773121646009029
Dropped Feature: primsev_cd_1
Score: 0.6774121386230144
Dropped Feature: primsev_cd_2
Score: 0.6775436141724663
Dropped Feature: unemplmt_cd
Score: 0.6775989077777093
Dropped Feature: gvsg_cd
Score: 0.6776179053764813
Dropped Feature: primsev_cd_3
Score: 0.6776434242487729
Dropped Feature: %_dropoutg
Score: 0.6776588129011195
Optimal Features: ['nonwhite_cd', 'prsatx_cd', 'CWSg_0_cd', 'dssg_0_cd',
'epsg_0_cd', 'adhdg_0_cd', 'cdsg_0_cd', 'cjsig_0_cd', 'srig_0_cd', 'SESg_0_cd',
'r4ag_0_cd', 'primsev_cd_4', 'primsev_cd_5', 'primsev_cd_6', '%_povertyg',
'%_public_assistanceg', '%_unemployedg']
Optimal Score: 0.6776588129011195
CPU times: user 4min 18s, sys: 18.5 s, total: 4min 37s
Wall time: 2min 48s
```

[9]:
```
%%time

rcr_FFS = forward_feature_selection(rcr, X, y)
```

```
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
```

```
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
//anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_validation.py:514: UserWarning: all
coefficients are zero, consider decreasing alpha.
  estimator.fit(X_train, y_train, **fit_params)
```

```
Added Feature: dssg_0_cd
Score: 0.600963905872708
Added Feature: r4ag_0_cd
Score: 0.6345411519078523
Added Feature: prsatx_cd
Score: 0.6437760250287912
Added Feature: SESg_0_cd
Score: 0.6529195463131539
Added Feature: cdsg_0_cd
Score: 0.6597671702588404
Added Feature: primsev_cd_5
Score: 0.6647272241326584
Added Feature: %_povertyg
Score: 0.669344880133808
Added Feature: srig_0_cd
Score: 0.6717639311559671
Added Feature: nonwhite_cd
Score: 0.6737998469686873
Added Feature: cjsig_0_cd
Score: 0.6751984384041073
Added Feature: primsev_cd_6
Score: 0.6764908552122334
Added Feature: adhdg_0_cd
Score: 0.676991134853712
Added Feature: primsev_cd_4
Score: 0.6773298106150195
Added Feature: CWSg_0_cd
Score: 0.6775957246817649
Added Feature: epsg_0_cd
Score: 0.6776605413963954
```

```
Added Feature: primsev_cd_2
Score: 0.6777095112305578
Optimal Features: ['dssg_0_cd', 'r4ag_0_cd', 'prsatx_cd', 'SESg_0_cd',
'cdsg_0_cd', 'primsev_cd_5', '%_povertyg', 'srig_0_cd', 'nonwhite_cd',
'cjsig_0_cd', 'primsev_cd_6', 'adhdg_0_cd', 'primsev_cd_4', 'CWSg_0_cd',
'epsg_0_cd', 'primsev_cd_2']
Optimal Score: 0.6777095112305578
CPU times: user 3min 6s, sys: 12.3 s, total: 3min 19s
Wall time: 2min 4s
```

[10]:
```
import json

stats = feature_selection_statistics(rcr_FFS, rcr_BFS, X.columns)
print(json.dumps(stats, indent=1))
```


```
{
 "Both_Used": [
  "%_povertyg",
  "CWSg_0_cd",
  "SESg_0_cd",
  "adhdg_0_cd",
  "cdsg_0_cd",
  "cjsig_0_cd",
  "dssg_0_cd",
  "epsg_0_cd",
  "nonwhite_cd",
  "primsev_cd_4",
  "primsev_cd_5",
  "primsev_cd_6",
  "prsatx_cd",
  "r4ag_0_cd",
  "srig_0_cd"
 ],
 "FFS_Used": [
  "primsev_cd_2"
 ],
 "BFS_Used": [
  "%_public_assistanceg",
  "%_unemployedg"
 ],
 "Both_Dropped": [
  "%_dropoutg",
  "B2a_0g",
  "female_cd",
  "gvsg_cd",
  "lrig_0_cd",
  "murder_numg",
  "pop_deng",
```

```
  "primsev_cd_1",
  "primsev_cd_3",
  "unemplmt_cd"
 ],
 "FFS_Dropped": [
  "%_public_assistanceg",
  "%_unemployedg"
 ],
 "BFS_Dropped": [
  "primsev_cd_2"
 ]
}
```

[11]:
```
%%time
from sksurv.ensemble import RandomSurvivalForest

rsf = RandomSurvivalForest()
rsf_BFS = backward_feature_selection(rsf, X, y)
```

```
//anaconda3/lib/python3.7/importlib/_bootstrap.py:219: RuntimeWarning:
sklearn.tree._splitter.Splitter size changed, may indicate binary
incompatibility. Expected 360 from C header, got 368 from PyObject
  return f(*args, **kwds)
```

```
Dropped Feature: %_dropoutg
Score: 0.6708214947596508
Optimal Features: ['female_cd', 'nonwhite_cd', 'unemplmt_cd', 'prsatx_cd',
'gvsg_cd', 'CWSg_0_cd', 'dssg_0_cd', 'epsg_0_cd', 'adhdg_0_cd', 'cdsg_0_cd',
'cjsig_0_cd', 'lrig_0_cd', 'srig_0_cd', 'SESg_0_cd', 'r4ag_0_cd',
'primsev_cd_1', 'primsev_cd_2', 'primsev_cd_3', 'primsev_cd_4', 'primsev_cd_5',
'primsev_cd_6', 'B2a_0g', 'murder_numg', '%_povertyg', '%_public_assistanceg',
'%_unemployedg', 'pop_deng']
Optimal Score: 0.6708214947596508
CPU times: user 16min 38s, sys: 1min 56s, total: 18min 35s
Wall time: 19min 39s
```

[12]:
```
%%time

rsf_FFS = forward_feature_selection(rsf, X, y)
```

```
Added Feature: dssg_0_cd
Score: 0.600963905872708
Added Feature: r4ag_0_cd
Score: 0.6346451627823293
Added Feature: SESg_0_cd
Score: 0.6457088110003675
Added Feature: prsatx_cd
Score: 0.6508620358535185
```

```
Added Feature: primsev_cd_5
Score: 0.6531618457422678
Added Feature: primsev_cd_6
Score: 0.6559288379752741
Added Feature: murder_numg
Score: 0.6593064367668062
Added Feature: %_unemployedg
Score: 0.6593630386617033
Added Feature: %_public_assistanceg
Score: 0.6597501463910019
Optimal Features: ['dssg_0_cd', 'r4ag_0_cd', 'SESg_0_cd', 'prsatx_cd',
'primsev_cd_5', 'primsev_cd_6', 'murder_numg', '%_unemployedg',
'%_public_assistanceg']
Optimal Score: 0.6597501463910019
CPU times: user 21min 27s, sys: 1min 1s, total: 22min 29s
Wall time: 23min 30s
```

[13]:
```
stats = feature_selection_statistics(rsf_FFS, rsf_BFS, X.columns)
print(json.dumps(stats, indent=1))
```


```
{
 "Both_Used": [
  "%_public_assistanceg",
  "%_unemployedg",
  "SESg_0_cd",
  "dssg_0_cd",
  "murder_numg",
  "primsev_cd_5",
  "primsev_cd_6",
  "prsatx_cd",
  "r4ag_0_cd"
 ],
 "FFS_Used": [],
 "BFS_Used": [
  "%_povertyg",
  "B2a_0g",
  "CWSg_0_cd",
  "adhdg_0_cd",
  "cdsg_0_cd",
  "cjsig_0_cd",
  "epsg_0_cd",
  "female_cd",
  "gvsg_cd",
  "lrig_0_cd",
  "nonwhite_cd",
  "pop_deng",
  "primsev_cd_1",
  "primsev_cd_2",
```

```
  "primsev_cd_3",
  "primsev_cd_4",
  "srig_0_cd",
  "unemplmt_cd"
 ],
 "Both_Dropped": [
  "%_dropoutg"
 ],
 "FFS_Dropped": [
  "%_povertyg",
  "B2a_0g",
  "CWSg_0_cd",
  "adhdg_0_cd",
  "cdsg_0_cd",
  "cjsig_0_cd",
  "epsg_0_cd",
  "female_cd",
  "gvsg_cd",
  "lrig_0_cd",
  "nonwhite_cd",
  "pop_deng",
  "primsev_cd_1",
  "primsev_cd_2",
  "primsev_cd_3",
  "primsev_cd_4",
  "srig_0_cd",
  "unemplmt_cd"
 ],
 "BFS_Dropped": []
}
```

[14]:
```
# print out total notebook execution time
total_seconds = int(time.time() - start_time)
minutes = total_seconds // 60
seconds = total_seconds % 60
print("--- " + str(minutes) + " minutes " + str(seconds) + " seconds ---")
```

```
--- 48 minutes 5 seconds ---
```