ShuffleandSplit2C19-Appendix

February 25, 2015

In [56]:
```
import numpy as np
import pandas as pd
```

0.0.1 Load dataframe and count number of cases positive and negative cases

In [57]:
```
isozyme2c19 = pd.read_csv('data/2c19.csv')
```

In [58]:
```
# Renaming the Activity Score column to conform to Python syntax
isozyme2c19.rename(columns={'p450-cyp2c19-ActivityScore': 'ActivityScore'}, inplace=True)
```

In [59]:
```
# Number of substances with an activity scores greater than or equal to 40
n_pos = (isozyme2c19.ActivityScore >= 40).sum()
n_pos
```

Out[59]: 5914

In [60]:
```
# Number of substances with an activity score below 40
n_neg = (isozyme2c19.ActivityScore < 40).sum()
n_neg
```

Out[60]: 11229

0.1 Downsampling the negative cases

This section of code shuffles the order of substances with an Activity Score below 40 (negatives). Then
The seed value is set for the randomizer to ensure reproducibility. Different seeds will result in diff

In [61]:
```
# method adapted from DataRobot post about scikit-learn classification

# Downsample negative cases -- there are many more negatives than positives

indices = np.where(isozyme2c19.ActivityScore < 40)[0]
rng = np.random.RandomState(50)  # sets seed for random number generator
rng.shuffle(indices)             # different seed numbers result in different shuffle
n_pos = (isozyme2c19.ActivityScore >= 40).sum()
balanced = isozyme2c19.drop(isozyme2c19.index[indices[n_pos:]])

balanced.head(10)
```

Out[61]:

	SID	ActivityScore	apol	a_acc	a_acid	a_aro	a_base	a_count	\
0	842238	0	51.111824	1	0	6	1	46	
2	842319	20	52.328274	4	0	6	0	42	
3	842408	90	42.691135	4	0	11	0	31	
4	842584	41	36.787930	3	0	17	0	28	
5	842618	85	70.986168	5	0	17	2	64	

```
6   842697            42  47.032688      6       0     17       0       40
8   842891             0  81.870926      6       0     11       2       75
9   842953            44  62.660240      3       0     12       1       52
10  842968            10  61.196651      3       4     12       0       52
11  843048            43  57.944275      7       0     23       0       47

    a_don  a_heavy    ...       vsa_acid  vsa_base    vsa_don    vsa_hyd  \
0       1       21    ...       0.000000         0   5.682576  286.85770
2       0       24    ...       0.000000         0   0.000000  241.82869
3       0       22    ...       0.000000         0   0.000000  191.26006
4       2       18    ...       0.000000         0   0.000000  180.79523
5       0       31    ...       0.000000         0   0.000000  380.40643
6       1       24    ...       0.000000         0   5.682576  224.58302
8       1       36    ...       0.000000         0   5.682576  385.93207
9       1       29    ...       0.000000         0   0.000000  340.08496
10      0       31    ...      54.267685         0   0.000000  279.94351
11      4       29    ...       0.000000         0  23.425066  232.84666

    vsa_other    vsa_pol     Weight  weinerPath  weinerPol  zagreb
0   12.949531  19.249496  310.84900        1050         30     102
2   36.550735  61.022110  362.47400        1295         41     126
3   42.243458  59.150364  355.41501         983         35     120
4   22.381124  24.524654  255.70799         624         25      94
5   11.190562  43.926376  423.56500        2659         49     164
6   25.239683  59.437412  324.34399        1583         32     120
8   24.140093  59.997055  497.64398        4018         54     190
9   24.140093  33.813168  434.34698        2338         47     154
10  44.575069  72.842117  423.42099        2656         51     158
11  66.910896  67.501205  406.47400        2687         40     150

[10 rows x 188 columns]
```

In [62]:
```
# Demonstrate the dataset is balanced
n_pos = (balanced.ActivityScore >= 40).sum()
n_neg = (balanced.ActivityScore < 40).sum()
n_neg, n_pos
```

Out[62]:
```
(5914, 5914)
```

In [63]:
```
# Repeat code with different seed number to demonstrate shuffling

indices2 = np.where(isozyme2c19.ActivityScore < 40)[0]
rng2 = np.random.RandomState(3489453655)  # sets seed for random number generator
rng2.shuffle(indices2)              # different seed numbers result in different shuffle
n_pos2 = (isozyme2c19.ActivityScore >= 40).sum()
balanced2 = isozyme2c19.drop(isozyme2c19.index[indices2[n_pos2:]])

balanced2.head(10)
```

Out[63]:
```
       SID  ActivityScore       apol  a_acc  a_acid  a_aro  a_base  a_count  \
3   842408             90  42.691135      4       0     11       0       31
4   842584             41  36.787930      3       0     17       0       28
5   842618             85  70.986168      5       0     17       2       64
6   842697             42  47.032688      6       0     17       0       40
7   842789              0  59.769032      3       0     12       0       52
```

```
8   842891              0  81.870926      6       0     11       2       75
9   842953             44  62.660240      3       0     12       1       52
10  842968             10  61.196651      3       4     12       0       52
11  843048             43  57.944275      7       0     23       0       47
12  843170             20  56.986618      4       0      6       0       52

    a_don  a_heavy   ...      vsa_acid  vsa_base    vsa_don    vsa_hyd  \
3       0       22   ...      0.000000         0   0.000000  191.26006
4       2       18   ...      0.000000         0   0.000000  180.79523
5       0       31   ...      0.000000         0   0.000000  380.40643
6       1       24   ...      0.000000         0   5.682576  224.58302
7       3       28   ...      0.000000         0  17.047728  266.20810
8       1       36   ...      0.000000         0   5.682576  385.93207
9       1       29   ...      0.000000         0   0.000000  340.08496
10      0       31   ...     54.267685         0   0.000000  279.94351
11      4       29   ...      0.000000         0  23.425066  232.84666
12      1       26   ...      0.000000         0   5.682576  283.60077

    vsa_other    vsa_pol     Weight  weinerPath  weinerPol  zagreb
3   42.243458  59.150364  355.41501         983         35     120
4   22.381124  24.524654  255.70799         624         25      94
5   11.190562  43.926376  423.56500        2659         49     164
6   25.239683  59.437412  324.34399        1583         32     120
7   43.339603  57.748489  380.44800        2253         40     142
8   24.140093  59.997055  497.64398        4018         54     190
9   24.140093  33.813168  434.34698        2338         47     154
10  44.575069  72.842117  423.42099        2656         51     158
11  66.910896  67.501205  406.47400        2687         40     150
12  43.111317  48.887096  362.42599        1631         44     136

[10 rows x 188 columns]
```

0.2 Write files for analysis

```
In [64]: balanced.to_csv("data/balanced2c19.csv", index=False)
```

0.3 Generate Training and Test Set

```
In [65]: twoC19 = pd.read_csv("data/balanced2c19.csv")

In [66]: twoC19.head()
```

```
Out[66]:       SID  ActivityScore       apol  a_acc  a_acid  a_aro  a_base  a_count  \
0  842238              0  51.111824      1       0      6       1       46
1  842319             20  52.328274      4       0      6       0       42
2  842408             90  42.691135      4       0     11       0       31
3  842584             41  36.787930      3       0     17       0       28
4  842618             85  70.986168      5       0     17       2       64

   a_don  a_heavy   ...      vsa_acid  vsa_base   vsa_don    vsa_hyd  \
0      1       21   ...             0         0  5.682576  286.85770
1      0       24   ...             0         0  0.000000  241.82869
2      0       22   ...             0         0  0.000000  191.26006
3      2       18   ...             0         0  0.000000  180.79523
4      0       31   ...             0         0  0.000000  380.40643
```

```
   vsa_other    vsa_pol     Weight  weinerPath  weinerPol  zagreb
0  12.949531  19.249496  310.84900        1050         30     102
1  36.550735  61.022110  362.47400        1295         41     126
2  42.243458  59.150364  355.41501         983         35     120
3  22.381124  24.524654  255.70799         624         25      94
4  11.190562  43.926376  423.56500        2659         49     164

[5 rows x 188 columns]
```

0.3.1 Shuffle and split dataset while preserving pandas index and metadata.

In [67]:
```
# Method adapted to Python3 from function by boates at https://gist.github.com/boates/5127281
N = len(twoC19)
```

In [68]:
```
l = list(range(N))
```

In [69]:
```
random.seed(76)
random.shuffle(l)
```

In [70]:
```
# get splitting indicies
# Here they are set to 80% training, 0% cross-validation and 20% test sets
trainLen = int(N*.8)
cvLen    = int(N*0.0)
testLen  = int(N*.2)
```

In [71]:
```
# get training, cv, and test sets
training = twoC19.ix[l[:trainLen]]
cv       = twoC19.ix[l[trainLen:trainLen+cvLen]]
test     = twoC19.ix[l[trainLen+cvLen:]]
```

In [72]:
```
# Examine training set
training.head()
```

Out[72]:
```
            SID  ActivityScore       apol  a_acc  a_acid  a_aro  a_base  \
3232    4251837             89  58.610653      5       0     17       0
5553   11112737             43  57.457447      4       0     12       1
2512    4242934             84  50.423859      4       0     11       0
10128  17410368             20  37.776344      3       0     11       0
7432   14741505              0  46.399101      4       0     17       0

       a_count  a_don  a_heavy   ...     vsa_acid  vsa_base    vsa_don  \
3232        49      1       28   ...            0         0   5.682576
5553        49      0       27   ...            0         0   0.000000
2512        44      2       24   ...            0         0  11.365152
10128       28      1       20   ...            0         0   5.682576
7432        37      1       23   ...            0         0   5.682576

         vsa_hyd  vsa_other    vsa_pol     Weight  weinerPath  weinerPol  \
3232   334.55359  26.116156  32.443340  399.47101        2372         41
5553   295.90125  41.257984  21.078190  389.85898        2275         38
2512   208.70059  49.716503  43.506508  328.36798        1434         37
10128  176.45346  48.930611  34.353111  307.13998         821         30
7432   213.52023  43.328411  43.774151  326.38000        1447         29

       zagreb
```

```
3232      140
5553      142
2512      126
10128     106
7432      116

[5 rows x 188 columns]
```

In [73]:
```
test.shape
```

Out[73]:
```
(2366, 188)
```

In [74]:
```
# Check number of actives and inactives in test set
n_pos1 = (test.ActivityScore >= 40).sum()
n_neg1 = (test.ActivityScore < 40).sum()
n_neg1, n_pos1
```

Out[74]:
```
(1173, 1193)
```

In [75]:
```
# Check number of actives and inactives in training set
n_pos2 = (training.ActivityScore >= 40).sum()
n_neg2 = (training.ActivityScore < 40).sum()
n_neg2, n_pos2
```

Out[75]:
```
(4741, 4721)
```

0.4 Write resulting training and test set to files for use in all further analyses.

In [76]:
```
training.to_csv("data/training2c19.csv", index=False)
test.to_csv("data/test2c19.csv", index=False)
```

In [76]: