> This is prepared for Udacity Machine Learning Engineer Nanodegree online class
> Author: jtmoogle @github.com
> Date: Feb 6, 2018

Data Collection

Access data- manually pre-download the raw files listed in dataset section. File format could be excel, dat or text files

Data High School Graduation & Census

1. Manually downloaded the zipped file of Data for Diplomas_Merged Data.zip located at https://challenges.s3.amazonaws.com/data_for_diplomas/Data for Diplomas_Merged Data.zip
2. Extract the zipped file
   - Raw data: GRADUATION_WITH_CENSUS.csv contain graudates and census information per school district, state, and county level
   - Definition File: ALL_DATA_SCHEMA_M.pdf

Data Process

In [1]:

```
#0-------- Initialization
import logging
import os.path
import time
import sys
from platform import python_version
import gc
import warnings
warnings.filterwarnings("ignore", category= DeprecationWarning)
warnings.filterwarnings("ignore", category = UserWarning, module = "matplotlib")
import IPython
from IPython.display import display  # Pretty display for notebooks
import numpy as np
import pandas as pd
import sklearn as sk

# global logger to console and file
logger = logging.getLogger('jtMoogle')
logger.setLevel(logging.DEBUG)
console = logging.StreamHandler()
formatter = logging.Formatter('%(asctime)s %(levelname)s %(message)s', datefmt="%H:%M:%S")
console.setFormatter(formatter)
console.setLevel(logging.DEBUG)
logger.addHandler(console)
# logging preference
def info(msg): return (logger.info(msg))
def debug(msg): return (logger.debug(msg))
def clean_mem():
    '''    release unreferenced memory with gc.collect()    '''
    gc.collect()

# file handler
ldir=os.path.realpath(".")
fname = 'exec_log.' + time.strftime('%Y-%m-%d-%H') + ".txt"
filename = os.path.join(ldir, 'log', fname)
notnew = os.path.exists(filename)
fhandler = logging.FileHandler(filename=filename, mode='a')
info(" Logging to "+filename)
fhandler.setFormatter(formatter)
fhandler.setLevel(logging.DEBUG)
logger.addHandler(fhandler)
# Software, API version
info( '--> IPython version: {}'.format(IPython.__version__ ))
info( '--> numpy version: {}'.format(np.__version__))
info( '--> pandas version: {}'.format( pd.__version__))
info( '--> python version: {}'.format(python_version()))
```

```
info( '--> scikit-learn version: {}'.format( sk.__version__))
info( '--> sys version: {} '.format(sys.version ))
```

```
22:35:44 INFO  Logging to I:\_githup\joyce.wrk\capstone-proposal\log\exec_log.2018-02-06-22.txt
22:35:44 INFO --> IPython version: 6.2.1
22:35:44 INFO --> numpy version: 1.14.0
22:35:44 INFO --> pandas version: 0.22.0
22:35:44 INFO --> python version: 3.6.3
22:35:44 INFO --> scikit-learn version: 0.19.1
22:35:44 INFO --> sys version: 3.6.3 |Anaconda, Inc.| (default, Nov  8 2017, 15:10:56) [MSC v.1900 64 b
it (AMD64)]
```

In [2]:

```
#1----- Import Dataset
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
%matplotlib inline

# define function to load train, test, and validation datasets
def load_dataset(path):
    info('Load dataset path={}'.format(path))
    ds = pd.read_csv(path, encoding = "ISO-8859-1",  dtype={"LEAID": str}) #
    return ds
# basic statistics output
def stats(dataset, infotype= 1, detailtype=1 ):
    if (infotype > 0): display( 'Statistics: Dataset row.count col.count -> {}'.format(dataset.shape))
    if (infotype > 1): display(dataset.head(3))
    if (infotype > 2): display( 'corr() -->', dataset.corr())
    if (infotype > 3): display( 'cov() -->', dataset.cov())
    if (detailtype > 0): print(dataset.columns)
    if (detailtype > 1): print(dataset.dtypes)
    if (detailtype > 2): print(dataset.describe())
    if (detailtype > 3):
        str_list = [] # empty list to contain columns with strings (words)
        for colname, colvalue in dataset.iteritems():
            if type(colvalue[1]) == str:
                str_list.append(colname)
        # Get to the numeric columns by inversion
        num_list = dataset.columns.difference(str_list)
        # Create Dataframe containing only numerical features
        dsnum = dataset[num_list]
        f, ax = plt.subplots(figsize=(16, 12))
        plt.title('Pearson Correlation of features')
        # Draw the heatmap using seaborn
        #sns.heatmap(house_num.astype(float).corr(),linewidths=0.25,vmax=1.0, square=True, cmap="PuBuGn
", linecolor='k', annot=True)
        sns.heatmap(dsnum.astype(float).corr(),linewidths=0.25,vmax=1.0, square=True, cmap="cubehelix",
linecolor='k', annot=True)

rawdata = load_dataset('data/Merged Data/GRADUATION_WITH_CENSUS.CSV')  # read data file
stats(rawdata, 2, 3)
```

```
22:35:45 INFO Load dataset path=data/Merged Data/GRADUATION_WITH_CENSUS.CSV
```

```
'Statistics: Dataset row.count col.count -> (9907, 580)'
```

	Unnamed: 0	leaid11	STNAM	FIPST	leanm11	ALL_COHORT_1112	ALL_RATE_1112	MAM_COHORT_1112	MAM_RA
0	1	100005	ALABAMA	1	Albertville City	268	83.0	NaN	NaN
1	2	100006	ALABAMA	1	Marshall County	424	79.0	2.0	PS
2	3	100007	ALABAMA	1	Hoover City	1042	91.0	1.0	PS

3 rows × 580 columns

```
Index(['Unnamed: 0', 'leaid11', 'STNAM', 'FIPST', 'leanm11', 'ALL_COHORT_1112',
       'ALL_RATE_1112', 'MAM_COHORT_1112', 'MAM_RATE_1112', 'MAS_COHORT_1112',
       ...
       'pct_TEA_MailOutMailBack_CEN_2010', 'pct_TEA_Update_Leave_CEN_2010',
```

```
       'pct_Census_Mail_Returns_CEN_2010', 'pct_Vacant_CEN_2010',
       'pct_Deletes_CEN_2010', 'pct_Census_UAA_CEN_2010',
       'pct_Mailback_Count_CEN_2010', 'pct_FRST_FRMS_CEN_2010',
       'pct_RPLCMNT_FRMS_CEN_2010', 'pct_BILQ_Mailout_count_CEN_2010'],
      dtype='object', length=580)
Unnamed: 0                          int64
leaid11                             int64
STNAM                              object
FIPST                               int64
leanm11                            object
ALL_COHORT_1112                     int64
ALL_RATE_1112                     float64
MAM_COHORT_1112                   float64
MAM_RATE_1112                      object
MAS_COHORT_1112                   float64
MAS_RATE_1112                      object
MBL_COHORT_1112                   float64
MBL_RATE_1112                      object
MHI_COHORT_1112                   float64
MHI_RATE_1112                      object
MTR_COHORT_1112                   float64
MTR_RATE_1112                      object
MWH_COHORT_1112                   float64
MWH_RATE_1112                      object
CWD_COHORT_1112                   float64
CWD_RATE_1112                      object
ECD_COHORT_1112                   float64
ECD_RATE_1112                      object
LEP_COHORT_1112                   float64
Percentage                        float64
State                               int64
County                              int64
Tract.Code                          int64
School.District                    object
District.ID                         int64
                                    ...
pct_Renter_Occp_HU_ACSMOE_08_12     float64
pct_Owner_Occp_HU_CEN_2010          float64
pct_Owner_Occp_HU_ACS_08_12         float64
pct_Owner_Occp_HU_ACSMOE_08_12      float64
pct_Single_Unit_ACS_08_12           float64
pct_Single_Unit_ACSMOE_08_12        float64
pct_MLT_U2_9_STRC_ACS_08_12         float64
pct_MLT_U2_9_STRC_ACSMOE_08_12      float64
pct_MLT_U10p_ACS_08_12              float64
pct_MLT_U10p_ACSMOE_08_12           float64
pct_Mobile_Homes_ACS_08_12          float64
pct_Mobile_Homes_ACSMOE_08_12       float64
pct_Crowd_Occp_U_ACS_08_12          float64
pct_Crowd_Occp_U_ACSMOE_08_12       float64
pct_NO_PH_SRVC_ACS_08_12            float64
pct_NO_PH_SRVC_ACSMOE_08_12         float64
pct_No_Plumb_ACS_08_12              float64
pct_No_Plumb_ACSMOE_08_12           float64
pct_Recent_Built_HU_ACS_08_12       float64
pct_Recent_Built_HU_ACSMOE_08_12    float64
pct_TEA_MailOutMailBack_CEN_2010    float64
pct_TEA_Update_Leave_CEN_2010       float64
pct_Census_Mail_Returns_CEN_2010    float64
pct_Vacant_CEN_2010                 float64
pct_Deletes_CEN_2010                float64
pct_Census_UAA_CEN_2010             float64
pct_Mailback_Count_CEN_2010         float64
pct_FRST_FRMS_CEN_2010              float64
pct_RPLCMNT_FRMS_CEN_2010           float64
pct_BILQ_Mailout_count_CEN_2010     float64
Length: 580, dtype: object
        Unnamed: 0       leaid11        FIPST  ALL_COHORT_1112  ALL_RATE_1112  \
count  9907.000000  9.907000e+03  9907.000000      9907.000000     9785.00000
mean   4954.000000  3.092285e+06    30.786515       333.867266       83.03909
std    2860.048892  1.472512e+06    14.712891       995.643288       11.87376
min       1.000000  1.000050e+05     1.000000         1.000000       18.00000
25%    2477.500000  1.919965e+06    19.000000        50.000000       80.00000
50%    4954.000000  3.100122e+06    31.000000       121.000000       87.00000
75%    7430.500000  4.218700e+06    42.000000       284.000000       92.00000
max    9907.000000  5.606240e+06    56.000000     43098.000000       99.00000
```

```
       MAM_COHORT_1112  MAS_COHORT_1112  MBL_COHORT_1112  MHI_COHORT_1112  \
count      3793.000000      5136.000000      6284.000000      7233.000000
mean          8.163723        32.824961        88.001750        91.042030
std          29.858498       196.339923       401.610909       544.021167
min           1.000000         1.000000         1.000000         1.000000
25%           1.000000         1.000000         2.000000         3.000000
50%           2.000000         4.000000         8.000000         8.000000
75%           6.000000        15.000000        48.000000        36.000000
max         988.000000     10882.000000     14134.000000     32047.000000

       MTR_COHORT_1112              ...                \
count      4157.000000              ...
mean         11.228290              ...
std          33.001452              ...
min           1.000000              ...
25%           1.000000              ...
50%           3.000000              ...
75%           8.000000              ...
max        1229.000000              ...

       pct_TEA_MailOutMailBack_CEN_2010  pct_TEA_Update_Leave_CEN_2010  \
count                       9677.000000                    9677.000000
mean                          72.313594                      27.686407
std                           40.196231                      40.196232
min                            0.000000                       0.000000
25%                           38.240000                       0.000000
50%                          100.000000                       0.000000
75%                          100.000000                      61.760000
max                          100.000000                     100.000000

       pct_Census_Mail_Returns_CEN_2010  pct_Vacant_CEN_2010  \
count                       9677.000000          9677.000000
mean                          65.107947             8.385380
std                           12.060588             9.999988
min                            0.000000             0.000000
25%                           58.470000             2.330000
50%                           67.100000             3.960000
75%                           73.560000            10.640000
max                          100.000000            82.680000

       pct_Deletes_CEN_2010  pct_Census_UAA_CEN_2010  \
count           9677.000000              9677.000000
mean               0.952053                10.492945
std                2.210318                10.538552
min                0.000000                 0.000000
25%                0.000000                 3.140000
50%                0.000000                 8.250000
75%                1.170000                14.610000
max              100.000000               100.000000

       pct_Mailback_Count_CEN_2010  pct_FRST_FRMS_CEN_2010  \
count                  9677.000000             9677.000000
mean                     80.169658               63.479170
std                      12.425322               12.797607
min                       0.000000                0.000000
25%                      74.570000               55.390000
50%                      83.100000               65.180000
75%                      88.970000               73.110000
max                     100.000000              100.000000

       pct_RPLCMNT_FRMS_CEN_2010  pct_BILQ_Mailout_count_CEN_2010
count                9677.000000                      9677.000000
mean                    1.628788                         4.564316
std                     2.773349                        20.438712
min                     0.000000                         0.000000
25%                     0.000000                         0.000000
50%                     0.000000                         0.000000
75%                     2.900000                         0.000000
max                    19.570000                       100.000000

[8 rows x 559 columns]
```

In [3]:

```
#2 ------ Target variables
cls_target_col = 'Success_Pass_90'  # classification
rgs_target_col = 'ALL_RATE_1112'    # Regression
```

```
cmpl_ds = rawdata
cmpl_ds[cls_target_col] = (rawdata['ALL_RATE_1112'] >= 90.0) * 1
print(cmpl_ds[[cls_target_col, rgs_target_col]].head())
stats(cmpl_ds[[cls_target_col]], 1, 3)
stats(cmpl_ds[[rgs_target_col]], 1, 3)
```

```
   Success_Pass_90  ALL_RATE_1112
0                0           83.0
1                0           79.0
2                1           91.0
3                1           91.0
4                0           72.0
```

```
'Statistics: Dataset row.count col.count -> (9907, 1)'
```

```
Index(['Success_Pass_90'], dtype='object')
Success_Pass_90    int32
dtype: object
       Success_Pass_90
count      9907.000000
mean          0.389321
std           0.487621
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max           1.000000
```

```
'Statistics: Dataset row.count col.count -> (9907, 1)'
```

```
Index(['ALL_RATE_1112'], dtype='object')
ALL_RATE_1112    float64
dtype: object
       ALL_RATE_1112
count     9785.00000
mean        83.03909
std         11.87376
min         18.00000
25%         80.00000
50%         87.00000
75%         92.00000
max         99.00000
```

In [4]:

```
#2.1--- Classification target variable
info('Classification target variable')
stats(cmpl_ds[[cls_target_col]], 1, 3)
```

```
22:35:47 INFO Classification target variable
```

```
'Statistics: Dataset row.count col.count -> (9907, 1)'
```

```
Index(['Success_Pass_90'], dtype='object')
Success_Pass_90    int32
dtype: object
       Success_Pass_90
count      9907.000000
mean          0.389321
std           0.487621
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max           1.000000
```

In [5]:

```
#2.2--- Regression target variable
info('Regression target variable')
stats(cmpl_ds[[rgs_target_col]], 1, 3)
```

```
22:35:47 INFO Regression target variable
```

```
'Statistics: Dataset row.count col.count -> (9907, 1)'
```

```
Index(['ALL_RATE_1112'], dtype='object')
ALL_RATE_1112    float64
dtype: object
       ALL_RATE_1112
count     9785.00000
mean        83.03909
```

```
mean         83.03909
std          11.87376
min          18.00000
25%          80.00000
50%          87.00000
75%          92.00000
max          99.00000
```

In [6]:

```
#3-------- Pre-process data: clean dataset
def preprocdata( dataset, target_col, sel_regex, drop_regex ):      # pre process data

    debug('1. Drop colums regex={}'.format(drop_regex))
    fulldata = dataset.drop(dataset.filter(regex=drop_regex), axis = 1)
    #print( 'colums={}'.format(fulldata.columns))

    debug('2. Select colums regex={}'.format(sel_regex))
    fulldata = fulldata.filter(regex=sel_regex, axis=1)
    #print( 'colums={}'.format(fulldata.columns))

    debug('3. Filter only datatype float64, int32/int64')
    fulldata = fulldata.select_dtypes(include=['float64', 'int32', 'int64'])

    debug('4. Drop rows if col has NaN value')
    fulldata = fulldata.dropna(subset=[target_col])
    fulldata = fulldata.dropna( thresh=10 ) # if count(Nan)>= 10

    debug('5. Get target data for targe column ')
    targetdata = fulldata[[target_col]]
    try:
        featuredata = fulldata.drop(columns=[target_col])
        featuredata = featuredata.drop(columns=[rgs_target_col])
        featuredata = featuredata.drop(columns=[cls_target_col])
    except:
        None

    debug('5. Fill in missing data with zero - impute NaN with zero')
    featuredata.fillna(0,  inplace=True) # impute with zero.  NOT delete featuredata = featuredata.drop
na(axis=1, how='any')

    featurecols = featuredata.columns

    return featurecols, featuredata, targetdata
```

In [7]:

```
#3.1 ----- Data for classification features
#keycols=['leadid11', 'State', 'County', 'District.ID']
selcol_regex = 'leaid|State.1|County.1|Inc|INC|_COHORT_|pct_|avg_|_House_|_AREA_|ALL_|Success'
dropcol_regex = 'MOE_|_FRMS_|_Mail'

cls_feature_cols, cls_feature_data, cls_target_data = preprocdata(
    cmpl_ds, cls_target_col, selcol_regex, dropcol_regex )
info( 'feature columns')
print(cls_feature_cols)
info('cls_feature_data')
stats(cls_feature_data, 3, 1)
info('cls_target_data')
stats(cls_target_data, 2)
```

```
22:35:48 DEBUG 1. Drop colums regex=MOE_|_FRMS_|_Mail
22:35:48 DEBUG 2. Select colums regex=leaid|State.1|County.1|Inc|INC|_COHORT_|pct_|avg_|_House_|_AREA_|
ALL_|Success
22:35:48 DEBUG 3. Filter only datatype float64, int32/int64
22:35:48 DEBUG 4. Drop rows if col has NaN value
22:35:48 DEBUG 5. Get target data for targe column
22:35:48 DEBUG 5. Fill in missing data with zero - impute NaN with zero
22:35:48 INFO feature columns
22:35:48 INFO cls_feature_data
```

```
Index(['leaid11', 'ALL_COHORT_1112', 'MAM_COHORT_1112', 'MAS_COHORT_1112',
       'MBL_COHORT_1112', 'MHI_COHORT_1112', 'MTR_COHORT_1112',
       'MWH_COHORT_1112', 'CWD_COHORT_1112', 'ECD_COHORT_1112',
       ...
       'pct_MLT_U10p_ACS_08_12', 'pct_Mobile_Homes_ACS_08_12',
       'pct_Crowd_Occp_U_ACS_08_12', 'pct_NO_PH_SRVC_ACS_08_12',
       'pct_No_Plumb_ACS_08_12', 'pct_Recent_Built_HU_ACS_08_12',
```

```
       'pct_TEA_Update_Leave_CEN_2010', 'pct_Vacant_CEN_2010',
       'pct_Deletes_CEN_2010', 'pct_Census_UAA_CEN_2010'],
      dtype='object', length=159)
```

```
'Statistics: Dataset row.count col.count -> (9907, 159)'
```

	leaid11	ALL_COHORT_1112	MAM_COHORT_1112	MAS_COHORT_1112	MBL_COHORT_1112	MHI_COHORT_1112	M
0	100005	268	0.0	0.0	6.0	49.0	0.
1	100006	424	2.0	1.0	4.0	26.0	0.
2	100007	1042	1.0	71.0	224.0	52.0	5.

3 rows × 159 columns

```
'corr() -->'
```

	leaid11	ALL_COHORT_1112	MAM_COHORT_1112	MAS_COHORT_1112	MBL
leaid11	1.000000	-0.079944	-0.064203	-0.071248	-0.04
ALL_COHORT_1112	-0.079944	1.000000	0.256169	0.545393	0.752
MAM_COHORT_1112	-0.064203	0.256169	1.000000	0.143561	0.13
MAS_COHORT_1112	-0.071248	0.545393	0.143561	1.000000	0.26
MBL_COHORT_1112	-0.044662	0.752941	0.136351	0.261471	1.00
MHI_COHORT_1112	-0.069349	0.824600	0.204371	0.375788	0.46
MTR_COHORT_1112	-0.062245	0.617613	0.199647	0.361685	0.38
MWH_COHORT_1112	-0.055610	0.751343	0.201455	0.362612	0.44
CWD_COHORT_1112	-0.075946	0.963346	0.244822	0.514667	0.76
ECD_COHORT_1112	-0.080248	0.921003	0.236222	0.470519	0.71
LEP_COHORT_1112	-0.098532	0.776887	0.222771	0.451276	0.41
State.1	0.999954	-0.079593	-0.063938	-0.071354	-0.04
County.1	0.197465	-0.017038	-0.038056	-0.031654	0.02
URBANIZED_AREA_POP_CEN_2010	-0.045561	0.281868	0.043789	0.153978	0.15
PUB_ASST_INC_ACS_08_12	-0.024972	0.033052	0.093572	0.027922	0.02
pct_URBANIZED_AREA_POP_CEN_2010	-0.038766	0.321516	0.052387	0.174466	0.20
pct_URBAN_CLUSTER_POP_CEN_2010	-0.005428	-0.078836	0.029616	-0.055794	-0.06
pct_RURAL_POP_CEN_2010	0.043707	-0.254139	-0.074163	-0.126420	-0.15
pct_Males_CEN_2010	0.015943	-0.053461	-0.001261	-0.021460	-0.05
pct_Males_ACS_08_12	0.012788	-0.047224	0.005908	-0.014331	-0.05
pct_Females_CEN_2010	0.000447	0.027588	-0.001773	0.019844	0.02
pct_Females_ACS_08_12	0.004142	0.023898	-0.007925	0.013392	0.02
pct_Pop_Under_5_CEN_2010	-0.029217	0.065278	0.121526	0.022277	0.05

	leaid11	ALL_COHORT_1112	MAM_COHORT_1112	MAS_COHORT_1112	MBL
pct_Pop_Under_5_ACS_08_12	0.029448	0.058695	0.085718	0.024289	0.045
pct_Pop_5_17_CEN_2010	-0.014856	0.005601	0.064163	0.007974	-0.03
pct_Pop_5_17_ACS_08_12	-0.005170	-0.000115	0.059071	0.005095	-0.03
pct_Pop_18_24_CEN_2010	-0.003776	0.064716	0.038120	0.022343	0.060
pct_Pop_18_24_ACS_08_12	-0.006947	0.065254	0.036199	0.023642	0.069
pct_Pop_25_44_CEN_2010	-0.029645	0.167660	0.055165	0.084367	0.13
pct_Pop_25_44_ACS_08_12	-0.017786	0.151757	0.053140	0.074546	0.110
...	...	...	...	...	...
pct_Sngl_Prns_HHD_CEN_2010	0.010145	-0.053651	-0.022749	-0.049602	0.027
pct_Sngl_Prns_HHD_ACS_08_12	-0.000489	-0.031646	-0.012141	-0.039356	0.045
pct_HHD_PPL_Und_18_CEN_2010	-0.045613	0.104345	0.105173	0.068613	0.029
pct_HHD_PPL_Und_18_ACS_08_12	-0.039007	0.100220	0.082639	0.068961	0.022
avg_Tot_Prns_in_HHD_CEN_2010	-0.047852	0.105654	0.098433	0.087846	0.022
avg_Tot_Prns_in_HHD_ACS_08_12	-0.048946	0.103667	0.118840	0.085637	0.029
pct_Rel_Under_6_CEN_2010	-0.057253	0.133094	0.153516	0.059739	0.11
pct_Rel_Under_6_ACS_08_12	-0.039127	0.107456	0.103466	0.059078	0.082
pct_HHD_Moved_in_ACS_08_12	-0.045699	0.134700	0.052525	0.047756	0.12
pct_PUB_ASST_INC_ACS_08_12	-0.048863	0.020353	0.122412	0.032449	0.029
pct_Tot_Occp_Units_CEN_2010	-0.019575	0.097620	0.008335	0.064066	0.030
pct_Tot_Occp_Units_ACS_08_12	-0.015027	0.091368	-0.004413	0.065614	0.019
pct_Vacant_Units_CEN_2010	0.030095	-0.122537	-0.010923	-0.073549	-0.04
pct_Vacant_Units_ACS_08_12	0.024458	-0.118577	0.004807	-0.076494	-0.03
pct_Renter_Occp_HU_CEN_2010	-0.046683	0.128066	0.086663	0.077271	0.142
pct_Renter_Occp_HU_ACS_08_12	-0.046195	0.125324	0.084458	0.079496	0.139
pct_Owner_Occp_HU_CEN_2010	0.048964	-0.129020	-0.082888	-0.073198	-0.14
pct_Owner_Occp_HU_ACS_08_12	0.048188	-0.125497	-0.079190	-0.074172	-0.13
pct_Single_Unit_ACS_08_12	-0.001169	-0.113917	-0.062603	-0.082169	-0.11
pct_MLT_U2_9_STRC_ACS_08_12	-0.002955	0.115914	0.036679	0.069383	0.124
pct_MLT_U10p_ACS_08_12	-0.012091	0.195918	0.045770	0.162603	0.142

pct_Mobile_Homes_ACS_08_12	0.[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
pct_Crowd_Occp_U_ACS_08_12	-0.097758	0.109817	0.192535	0.147677	0.04(
pct_NO_PH_SRVC_ACS_08_12	-0.055397	-0.007958	0.153271	-0.021400	0.02
pct_No_Plumb_ACS_08_12	-0.024658	-0.110221	0.093096	-0.060414	-0.03
pct_Recent_Built_HU_ACS_08_12	0.027372	0.037978	0.016996	0.017706	0.00
pct_TEA_Update_Leave_CEN_2010	-0.045087	-0.167861	-0.053418	-0.073896	-0.09
pct_Vacant_CEN_2010	-0.022459	-0.131950	-0.042118	-0.060589	-0.06
pct_Deletes_CEN_2010	-0.017856	-0.108845	-0.035818	-0.048617	-0.06
pct_Census_UAA_CEN_2010	0.103219	-0.020392	-0.019612	-0.041535	0.01

159 rows × 159 columns

22:35:48 INFO cls_target_data

```
Index(['leaid11', 'ALL_COHORT_1112', 'MAM_COHORT_1112', 'MAS_COHORT_1112',
       'MBL_COHORT_1112', 'MHI_COHORT_1112', 'MTR_COHORT_1112',
       'MWH_COHORT_1112', 'CWD_COHORT_1112', 'ECD_COHORT_1112',
       ...
       'pct_MLT_U10p_ACS_08_12', 'pct_Mobile_Homes_ACS_08_12',
       'pct_Crowd_Occp_U_ACS_08_12', 'pct_NO_PH_SRVC_ACS_08_12',
       'pct_No_Plumb_ACS_08_12', 'pct_Recent_Built_HU_ACS_08_12',
       'pct_TEA_Update_Leave_CEN_2010', 'pct_Vacant_CEN_2010',
       'pct_Deletes_CEN_2010', 'pct_Census_UAA_CEN_2010'],
      dtype='object', length=159)
```

```
'Statistics: Dataset row.count col.count -> (9907, 1)'
```

	Success_Pass_90
0	0
1	0
2	1

```
Index(['Success_Pass_90'], dtype='object')
```

In [8]:

```
#3.2 ----- Data for regression features
#keycols=['leadid11', 'State', 'County', 'District.ID']
selcol_regex = 'leaid|State.1|County.1|Inc|INC|_COHORT_|pct_|avg_|_House_|_AREA_|ALL_'
dropcol_regex = 'MOE_|_FRMS_|_Mail'

rgs_feature_cols, rgs_feature_data, rgs_target_data = preprocdata(
    cmpl_ds, rgs_target_col, selcol_regex, dropcol_regex)
info( 'rgs_feature columns')
print(rgs_feature_cols)
info('rgs_feature_data')
stats(rgs_feature_data, 3, 1)
info('rgs_target_data')
stats(rgs_target_data)
```

```
22:35:49 DEBUG 1. Drop colums regex=MOE_|_FRMS_|_Mail
22:35:49 DEBUG 2. Select colums regex=leaid|State.1|County.1|Inc|INC|_COHORT_|pct_|avg_|_House_|_AREA_|
ALL_
22:35:49 DEBUG 3. Filter only datatype float64, int32/int64
22:35:49 DEBUG 4. Drop rows if col has NaN value
22:35:49 DEBUG 5. Get target data for targe column
22:35:49 DEBUG 5. Fill in missing data with zero - impute NaN with zero
22:35:49 INFO rgs_feature columns
22:35:49 INFO rgs_feature_data
```

```
Index(['leaid11', 'ALL_COHORT_1112', 'MAM_COHORT_1112', 'MAS_COHORT_1112',
       'MBL_COHORT_1112', 'MHI_COHORT_1112', 'MTR_COHORT_1112',
```

```
       'MWH_COHORT_1112', 'CWD_COHORT_1112', 'ECD_COHORT_1112',
       ...
       'pct_MLT_U10p_ACS_08_12', 'pct_Mobile_Homes_ACS_08_12',
       'pct_Crowd_Occp_U_ACS_08_12', 'pct_NO_PH_SRVC_ACS_08_12',
       'pct_No_Plumb_ACS_08_12', 'pct_Recent_Built_HU_ACS_08_12',
       'pct_TEA_Update_Leave_CEN_2010', 'pct_Vacant_CEN_2010',
       'pct_Deletes_CEN_2010', 'pct_Census_UAA_CEN_2010'],
      dtype='object', length=159)
```

```
'Statistics: Dataset row.count col.count -> (9785, 159)'
```

	leaid11	ALL_COHORT_1112	MAM_COHORT_1112	MAS_COHORT_1112	MBL_COHORT_1112	MHI_COHORT_1112	M
0	100005	268	0.0	0.0	6.0	49.0	0.
1	100006	424	2.0	1.0	4.0	26.0	0.
2	100007	1042	1.0	71.0	224.0	52.0	5.

3 rows × 159 columns

```
'corr() -->'
```

	leaid11	ALL_COHORT_1112	MAM_COHORT_1112	MAS_COHORT_1112	MBL
leaid11	1.000000	-0.080839	-0.065010	-0.071834	-0.04
ALL_COHORT_1112	-0.080839	1.000000	0.256299	0.545321	0.75
MAM_COHORT_1112	-0.065010	0.256299	1.000000	0.143640	0.13
MAS_COHORT_1112	-0.071834	0.545321	0.143640	1.000000	0.26
MBL_COHORT_1112	-0.045119	0.752881	0.136379	0.261285	1.00
MHI_COHORT_1112	-0.069944	0.824680	0.204520	0.375658	0.46
MTR_COHORT_1112	-0.062886	0.617336	0.199753	0.361503	0.38
MWH_COHORT_1112	-0.056557	0.750977	0.201539	0.362477	0.44
CWD_COHORT_1112	-0.076801	0.963297	0.244970	0.514573	0.76
ECD_COHORT_1112	-0.081026	0.921002	0.236308	0.470376	0.71
LEP_COHORT_1112	-0.099297	0.776994	0.222941	0.451179	0.41
State.1	0.999954	-0.080487	-0.064728	-0.071941	-0.04
County.1	0.200958	-0.017583	-0.038783	-0.031950	0.02
URBANIZED_AREA_POP_CEN_2010	-0.046588	0.280499	0.043446	0.153560	0.15
PUB_ASST_INC_ACS_08_12	-0.025916	0.033519	0.093016	0.028305	0.02
pct_URBANIZED_AREA_POP_CEN_2010	-0.039762	0.320123	0.052028	0.174069	0.20
pct_URBAN_CLUSTER_POP_CEN_2010	-0.006149	-0.080565	0.029380	-0.056471	-0.06
pct_RURAL_POP_CEN_2010	0.045420	-0.252244	-0.073888	-0.125902	-0.15
pct_Males_CEN_2010	0.017603	-0.052133	-0.000897	-0.020973	-0.05
pct_Males_ACS_08_12	0.013982	-0.046164	0.006190	-0.013930	-0.05

	leaid11	ALL_COHORT_1112	MAM_COHORT_1112	MAS_COHORT_1112	MB
pct_Females_CEN_2010	-0.000976	0.026588	-0.002045	0.012494	0.02
pct_Females_ACS_08_12	0.003197	0.023167	-0.008111	0.013134	0.028
pct_Pop_Under_5_CEN_2010	-0.029510	0.065639	0.120960	0.022391	0.056
pct_Pop_Under_5_ACS_08_12	-0.028146	0.058945	0.085501	0.024394	0.045
pct_Pop_5_17_CEN_2010	-0.016518	0.005274	0.062937	0.007903	-0.03
pct_Pop_5_17_ACS_08_12	-0.007332	-0.000539	0.058272	0.004989	-0.03
pct_Pop_18_24_CEN_2010	-0.003500	0.063648	0.037252	0.021945	0.065
pct_Pop_18_24_ACS_08_12	-0.006177	0.064334	0.035034	0.023300	0.068
pct_Pop_25_44_CEN_2010	-0.028360	0.166406	0.054524	0.083978	0.130
pct_Pop_25_44_ACS_08_12	-0.016392	0.150375	0.052999	0.074102	0.110
...	...	...	...	...	...
pct_Sngl_Prns_HHD_CEN_2010	0.012842	-0.052982	-0.021853	-0.049462	0.028
pct_Sngl_Prns_HHD_ACS_08_12	0.001470	-0.031091	-0.011521	-0.039262	0.040
pct_HHD_PPL_Und_18_CEN_2010	-0.046500	0.103779	0.103311	0.068651	0.028
pct_HHD_PPL_Und_18_ACS_08_12	-0.039735	0.099490	0.081445	0.068945	0.021
avg_Tot_Prns_in_HHD_CEN_2010	-0.049841	0.105879	0.096641	0.088334	0.022
avg_Tot_Prns_in_HHD_ACS_08_12	-0.049931	0.104089	0.117323	0.086171	0.025
pct_Rel_Under_6_CEN_2010	-0.056881	0.133343	0.152524	0.059922	0.111
pct_Rel_Under_6_ACS_08_12	-0.038041	0.107377	0.102581	0.059205	0.083
pct_HHD_Moved_in_ACS_08_12	-0.046021	0.134356	0.052084	0.047603	0.121
pct_PUB_ASST_INC_ACS_08_12	-0.047634	0.022076	0.121744	0.033481	0.030
pct_Tot_Occp_Units_CEN_2010	-0.022304	0.095546	0.007454	0.063809	0.028
pct_Tot_Occp_Units_ACS_08_12	-0.017762	0.089247	-0.005105	0.065352	0.011
pct_Vacant_Units_CEN_2010	0.033659	-0.120914	-0.009954	-0.073644	-0.04
pct_Vacant_Units_ACS_08_12	0.028058	-0.116830	0.005712	-0.076611	-0.03
pct_Renter_Occp_HU_CEN_2010	-0.045672	0.128142	0.086360	0.077376	0.142
pct_Renter_Occp_HU_ACS_08_12	-0.045081	0.125421	0.084101	0.079637	0.135
pct_Owner_Occp_HU_CEN_2010	0.048071	-0.128938	-0.082545	-0.073223	-0.14
pct_Owner_Occp_HU_ACS_08_12	0.047217	-0.125389	-0.078780	-0.074208	-0.13
pct_Single_Unit_ACS_08_12	0.000579	-0.113005	-0.062071	-0.081995	-0.11
pct_MLT_U2_9_STRC_ACS_08_12	-	0.115210	0.036474	0.069251	0.124

	leaid11	ALL_COHORT_1112	MAM_COHORT_1112	MAS_COHORT_1112	MBL
pct_MLT_U10p_ACS_08_12	-0.012394	0.194818	0.045657	0.162323	0.14
pct_Mobile_Homes_ACS_08_12	0.018878	-0.104995	0.023739	-0.069061	-0.07
pct_Crowd_Occp_U_ACS_08_12	-0.097256	0.113320	0.193781	0.151176	0.04
pct_NO_PH_SRVC_ACS_08_12	-0.056218	-0.007596	0.152697	-0.021459	0.02
pct_No_Plumb_ACS_08_12	-0.016114	-0.110483	0.094084	-0.061415	-0.03
pct_Recent_Built_HU_ACS_08_12	0.026767	0.038496	0.017253	0.017906	0.00
pct_TEA_Update_Leave_CEN_2010	-0.043987	-0.166231	-0.052024	-0.073431	-0.09
pct_Vacant_CEN_2010	-0.022344	-0.130885	-0.041092	-0.060456	-0.06
pct_Deletes_CEN_2010	-0.017627	-0.107903	-0.034864	-0.048354	-0.05
pct_Census_UAA_CEN_2010	0.099550	-0.022239	-0.019794	-0.042445	0.01

159 rows × 159 columns

```
22:35:49 INFO rgs_target_data
```

```
Index(['leaid11', 'ALL_COHORT_1112', 'MAM_COHORT_1112', 'MAS_COHORT_1112',
       'MBL_COHORT_1112', 'MHI_COHORT_1112', 'MTR_COHORT_1112',
       'MWH_COHORT_1112', 'CWD_COHORT_1112', 'ECD_COHORT_1112',
       ...
       'pct_MLT_U10p_ACS_08_12', 'pct_Mobile_Homes_ACS_08_12',
       'pct_Crowd_Occp_U_ACS_08_12', 'pct_NO_PH_SRVC_ACS_08_12',
       'pct_No_Plumb_ACS_08_12', 'pct_Recent_Built_HU_ACS_08_12',
       'pct_TEA_Update_Leave_CEN_2010', 'pct_Vacant_CEN_2010',
       'pct_Deletes_CEN_2010', 'pct_Census_UAA_CEN_2010'],
      dtype='object', length=159)
```

```
'Statistics: Dataset row.count col.count -> (9785, 1)'
```

```
Index(['ALL_RATE_1112'], dtype='object')
```

In [9]:

```
#4 --- Stepwise selection
# cited: Does scikit-learn have forward selection/stepwise regression algorithm?
# https://datascience.stackexchange.com/questions/937/does-scikit-learn-have-forward-selection-stepwise
-regression-algorithm
#
import pandas as pd
import numpy as np
import statsmodels.api as sm
import json, codecs

def stepwise_selection(X, y, initial_list=[], threshold_in=0.01, threshold_out = 0.05):
    """ Perform a forward-backward feature selection based on p-value from statsmodels.api.OLS
    Arguments:
        X - pandas.DataFrame with candidate features
        y - list-like with the target
        initial_list - list of features to start with (column names of X)
        threshold_in - include a feature if its p-value < threshold_in
        threshold_out - exclude a feature if its p-value > threshold_out
    Returns: list of selected features
    Always set threshold_in < threshold_out to avoid infinite looping.
    See https://en.wikipedia.org/wiki/Stepwise_regression for the details
    """
    included = list(initial_list)
    while True:
        changed=False
        # forward step
        excluded = list(set(X.columns)-set(included))
        new_pval = pd.Series(index=excluded)
```

```
        for new_column in excluded:
            model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included+[new_column]]))).fit()
            #debug(model.summary())
            new_pval[new_column] = model.pvalues[new_column]
        best_pval = new_pval.min()
        if best_pval < threshold_in:
            best_feature = new_pval.argmin()
            included.append(best_feature)
            changed=True
            print( 'Add  {:30} with p-value {:.6}'.format(best_feature, best_pval))

        # backward step
        model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included]))).fit()
        #debug(model.summary())
        # use all coefs except intercept
        pvalues = model.pvalues.iloc[1:]
        worst_pval = pvalues.max() # null if pvalues is empty
        if worst_pval > threshold_out:
            changed=True
            worst_feature = pvalues.argmax()
            included.remove(worst_feature)
            print('Drop {:30} with p-value {:.6}'.format(worst_feature, worst_pval))
        if not changed:
            break
    return included
```

```
C:\Users\joyce\AppData\Local\conda\conda\envs\capstone\lib\site-packages\statsmodels\compat\pandas.py
:56: FutureWarning: The pandas.core.datetools module is deprecated and will be removed in a future vers
ion. Please use the pandas.tseries module instead.
  from pandas.core import datetools
```

In [10]:

```
#4.1 --- regression
savefname='saved/rgs_stepwise_result.txt'
redofit=True
inc_cols=[]

rgs_X = pd.DataFrame( rgs_feature_data, columns= rgs_feature_cols)
rgs_y = rgs_target_data

if (not redofit) and (os.path.exists(savefname)):
    info( '{} exist.  stepwise_selection loaded from a file'.format(savefname))
    with open(savefname) as data_file:
        rgs_selresult = json.load(data_file)
else:
    rgs_selresult = stepwise_selection(rgs_X, rgs_y)
    print('Stepwise selection features:')
    display(rgs_selresult)
    with open(savefname, 'wb') as f:
        info( 'Save stepwise_selection to a file {}'.format(savefname))
        json.dump(rgs_selresult, codecs.getwriter('utf-8')(f), ensure_ascii=False)
```

```
C:\Users\joyce\AppData\Local\conda\conda\envs\capstone\lib\site-packages\ipykernel_launcher.py:35: Fut
ureWarning: 'argmin' is deprecated. Use 'idxmin' instead. The behavior of 'argmin' will be corrected to
return the positional minimum in the future. Use 'series.values.argmin' to get the position of the mini
mum now.
```

```
Add  pct_No_Plumb_ACS_08_12          with p-value 3.98472e-196
Add  pct_Prs_Blw_Pov_Lev_ACS_08_12   with p-value 3.67852e-85
Add  pct_Vacant_CEN_2010             with p-value 7.32569e-76
Add  pct_NH_AIAN_alone_CEN_2010      with p-value 1.24769e-46
Add  pct_NH_Blk_alone_CEN_2010       with p-value 1.17739e-56
Add  pct_College_ACS_08_12           with p-value 4.27032e-36
Add  MBL_COHORT_1112                 with p-value 1.02546e-27
Add  pct_HHD_PPL_Und_18_CEN_2010     with p-value 2.05965e-19
Add  pct_Pop_Under_5_CEN_2010        with p-value 5.85114e-18
Add  pct_URBAN_CLUSTER_POP_CEN_2010 with p-value 9.92684e-19
Add  State.1                         with p-value 2.49778e-14
Add  pct_Hispanic_CEN_2010           with p-value 1.34424e-09
Add  pct_URBANIZED_AREA_POP_CEN_2010 with p-value 3.64746e-09
Add  pct_PUB_ASST_INC_ACS_08_12      with p-value 2.03309e-07
Add  pct_NO_PH_SRVC_ACS_08_12        with p-value 1.97523e-07
Add  pct_Female_No_HB_ACS_08_12      with p-value 1.6557e-07
Add  pct_MLT_U10p_ACS_08_12          with p-value 1.13414e-05
Add  County.1                        with p-value 9.3131e-06
Add  pct_Civ_unemp_16p_ACS_08_12     with p-value 1.89565e-05
Add  pct_Rel_Under_6_CEN_2010        with p-value 2.03328e-05
```

```
Add  pct_Pop_45_64_ACS_08_12        with p-value 2.30077e-05
Add  pct_Pop_25yrs_Over_ACS_08_12   with p-value 4.13466e-06
Add  pct_Census_UAA_CEN_2010        with p-value 0.000263252
Add  pct_TEA_Update_Leave_CEN_2010  with p-value 0.00050568
Drop pct_URBANIZED_AREA_POP_CEN_2010 with p-value 0.356547
```

```
C:\Users\joyce\AppData\Local\conda\conda\envs\capstone\lib\site-packages\ipykernel_launcher.py:48: Fut
ureWarning: 'argmax' is deprecated. Use 'idxmax' instead. The behavior of 'argmax' will be corrected to
return the positional maximum in the future. Use 'series.values.argmax' to get the position of the maxi
mum now.
```

```
Add  pct_NH_NHOPI_alone_CEN_2010    with p-value 0.000917519
Add  pct_Age5p_Scandinav_ACS_08_12  with p-value 0.00315333
Add  pct_Age5p_Navajo_ACS_08_12     with p-value 0.00481829
Add  PUB_ASST_INC_ACS_08_12         with p-value 0.00643402
Add  pct_Age5p_WGerman_ACS_08_12    with p-value 0.00679172
Add  pct_Age5p_German_ACS_08_12     with p-value 0.000457511
Add  pct_Females_CEN_2010           with p-value 0.00705298
Add  pct_Inst_GQ_CEN_2010           with p-value 6.34452e-05
Drop pct_Pop_25yrs_Over_ACS_08_12   with p-value 0.192977
Add  pct_Males_CEN_2010             with p-value 0.000331079
Add  pct_Pop_25_44_CEN_2010         with p-value 0.000165853
Add  pct_NH_AIAN_alone_ACS_08_12    with p-value 0.00741088
Stepwise selection features:
```

```
['pct_No_Plumb_ACS_08_12',
 'pct_Prs_Blw_Pov_Lev_ACS_08_12',
 'pct_Vacant_CEN_2010',
 'pct_NH_AIAN_alone_CEN_2010',
 'pct_NH_Blk_alone_CEN_2010',
 'pct_College_ACS_08_12',
 'MBL_COHORT_1112',
 'pct_HHD_PPL_Und_18_CEN_2010',
 'pct_Pop_Under_5_CEN_2010',
 'pct_URBAN_CLUSTER_POP_CEN_2010',
 'State.1',
 'pct_Hispanic_CEN_2010',
 'pct_PUB_ASST_INC_ACS_08_12',
 'pct_NO_PH_SRVC_ACS_08_12',
 'pct_Female_No_HB_ACS_08_12',
 'pct_MLT_U10p_ACS_08_12',
 'County.1',
 'pct_Civ_unemp_16p_ACS_08_12',
 'pct_Rel_Under_6_CEN_2010',
 'pct_Pop_45_64_ACS_08_12',
 'pct_Census_UAA_CEN_2010',
 'pct_TEA_Update_Leave_CEN_2010',
 'pct_NH_NHOPI_alone_CEN_2010',
 'pct_Age5p_Scandinav_ACS_08_12',
 'pct_Age5p_Navajo_ACS_08_12',
 'PUB_ASST_INC_ACS_08_12',
 'pct_Age5p_WGerman_ACS_08_12',
 'pct_Age5p_German_ACS_08_12',
 'pct_Females_CEN_2010',
 'pct_Inst_GQ_CEN_2010',
 'pct_Males_CEN_2010',
 'pct_Pop_25_44_CEN_2010',
 'pct_NH_AIAN_alone_ACS_08_12']
```

```
22:37:43 INFO Save stepwise_selection to a file saved/rgs_stepwise_result.txt
```

In [11]:

```
#4.2 --- regression
rgs_feature_data = rgs_feature_data[rgs_selresult]
stats(rgs_feature_data)
```

```
'Statistics: Dataset row.count col.count -> (9785, 33)'
```

```
Index(['pct_No_Plumb_ACS_08_12', 'pct_Prs_Blw_Pov_Lev_ACS_08_12',
       'pct_Vacant_CEN_2010', 'pct_NH_AIAN_alone_CEN_2010',
       'pct_NH_Blk_alone_CEN_2010', 'pct_College_ACS_08_12', 'MBL_COHORT_1112',
       'pct_HHD_PPL_Und_18_CEN_2010', 'pct_Pop_Under_5_CEN_2010',
       'pct_URBAN_CLUSTER_POP_CEN_2010', 'State.1', 'pct_Hispanic_CEN_2010',
       'pct_PUB_ASST_INC_ACS_08_12', 'pct_NO_PH_SRVC_ACS_08_12',
       'pct_Female_No_HB_ACS_08_12', 'pct_MLT_U10p_ACS_08_12', 'County.1',
       'pct_Civ_unemp_16p_ACS_08_12', 'pct_Rel_Under_6_CEN_2010',
       'pct_Pop_45_64_ACS_08_12', 'pct_Census_UAA_CEN_2010',
       'pct_TEA_Update_Leave_CEN_2010', 'pct_NH_NHOPI_alone_CEN_2010',
```

```
       'pct_Age5p_Scandinav_ACS_08_12', 'pct_Age5p_Navajo_ACS_08_12',
       'PUB_ASST_INC_ACS_08_12', 'pct_Age5p_WGerman_ACS_08_12',
       'pct_Age5p_German_ACS_08_12', 'pct_Females_CEN_2010',
       'pct_Inst_GQ_CEN_2010', 'pct_Males_CEN_2010', 'pct_Pop_25_44_CEN_2010',
       'pct_NH_AIAN_alone_ACS_08_12'],
      dtype='object')
```

In [12]:

```
#4.3 --- classification
savefname='saved/cls_stepwise_result.txt'
redofit=True

cls_X = pd.DataFrame( cls_feature_data, columns= cls_feature_cols)
cls_y = cls_target_data
inc_cols=[]

if (not redofit) and (os.path.exists(savefname)):
    info( '{} exist.  stepwise_selection loaded from a file'.format(savefname))
    with open(savefname) as data_file:
        cls_selresult = json.load(data_file)
else:
    cls_selresult = stepwise_selection(cls_X, cls_y)
    print('Stepwise selection features:')
    display(cls_selresult)
    with open(savefname, 'wb') as f:
        info( 'Save stepwise_selection to a file {}'.format(savefname))
        json.dump(cls_selresult, codecs.getwriter('utf-8')(f), ensure_ascii=False)
```

```
C:\Users\joyce\AppData\Local\conda\conda\envs\capstone\lib\site-packages\ipykernel_launcher.py:35: Fut
ureWarning: 'argmin' is deprecated. Use 'idxmin' instead. The behavior of 'argmin' will be corrected to
return the positional minimum in the future. Use 'series.values.argmin' to get the position of the mini
mum now.
```

```
Add  pct_Prs_Blw_Pov_Lev_ACS_08_12  with p-value 3.61376e-135
Add  pct_Vacant_Units_ACS_08_12     with p-value 2.26467e-58
Add  pct_College_ACS_08_12          with p-value 1.29885e-46
Add  leaid11                        with p-value 1.34225e-37
Add  pct_MrdCple_HHD_CEN_2010       with p-value 4.29997e-32
Add  CWD_COHORT_1112                with p-value 2.26316e-20
Add  County.1                       with p-value 2.08507e-16
Add  pct_Mobile_Homes_ACS_08_12     with p-value 1.73795e-12
Add  pct_NH_Blk_alone_CEN_2010      with p-value 2.05705e-09
Add  pct_TEA_Update_Leave_CEN_2010  with p-value 2.09595e-09
Add  pct_NH_AIAN_alone_ACS_08_12    with p-value 9.5369e-07
Add  pct_Census_UAA_CEN_2010        with p-value 1.0184e-05
Add  pct_Pop_5_17_CEN_2010          with p-value 0.000193281
Add  pct_Pop_Under_5_CEN_2010       with p-value 1.63552e-11
Add  pct_Pop_45_64_CEN_2010         with p-value 1.53972e-08
Add  pct_Hispanic_CEN_2010          with p-value 5.76716e-07
Add  pct_Inst_GQ_CEN_2010           with p-value 8.10125e-07
Add  pct_Female_No_HB_ACS_08_12     with p-value 0.0016426
Add  pct_MLT_U10p_ACS_08_12         with p-value 0.0013046
Add  pct_Males_CEN_2010             with p-value 0.000129439
Add  MHI_COHORT_1112                with p-value 0.00357753
Add  pct_HHD_PPL_Und_18_CEN_2010    with p-value 0.00644943
Add  pct_Sngl_Prns_HHD_CEN_2010     with p-value 0.000478885
Add  pct_NH_NHOPI_alone_ACS_08_12   with p-value 0.00544154
Add  MAS_COHORT_1112                with p-value 0.00212339
Add  pct_Age5p_OthPacIsl_ACS_08_12  with p-value 0.00763461
Stepwise selection features:
```

```
['pct_Prs_Blw_Pov_Lev_ACS_08_12',
 'pct_Vacant_Units_ACS_08_12',
 'pct_College_ACS_08_12',
 'leaid11',
 'pct_MrdCple_HHD_CEN_2010',
 'CWD_COHORT_1112',
 'County.1',
 'pct_Mobile_Homes_ACS_08_12',
 'pct_NH_Blk_alone_CEN_2010',
 'pct_TEA_Update_Leave_CEN_2010',
 'pct_NH_AIAN_alone_ACS_08_12',
 'pct_Census_UAA_CEN_2010',
 'pct_Pop_5_17_CEN_2010',
 'pct_Pop_Under_5_CEN_2010',
 'pct_Pop_45_64_CEN_2010',
 'pct_Hispanic_CEN_2010',
 'pct_Inst_GQ_CEN_2010'
```

```
 'pct_Inst_GQ_CEN_2010',
 'pct_Female_No_HB_ACS_08_12',
 'pct_MLT_U10p_ACS_08_12',
 'pct_Males_CEN_2010',
 'MHI_COHORT_1112',
 'pct_HHD_PPL_Und_18_CEN_2010',
 'pct_Sngl_Prns_HHD_CEN_2010',
 'pct_NH_NHOPI_alone_ACS_08_12',
 'MAS_COHORT_1112',
 'pct_Age5p_OthPacIsl_ACS_08_12']
```

```
22:38:53 INFO Save stepwise_selection to a file saved/cls_stepwise_result.txt
```

In [13]:

```
#4.4 --- classification
cls_feature_data = cls_feature_data[cls_selresult]
stats(cls_feature_data)
```

```
'Statistics: Dataset row.count col.count -> (9907, 26)'
```

```
Index(['pct_Prs_Blw_Pov_Lev_ACS_08_12', 'pct_Vacant_Units_ACS_08_12',
       'pct_College_ACS_08_12', 'leaid11', 'pct_MrdCple_HHD_CEN_2010',
       'CWD_COHORT_1112', 'County.1', 'pct_Mobile_Homes_ACS_08_12',
       'pct_NH_Blk_alone_CEN_2010', 'pct_TEA_Update_Leave_CEN_2010',
       'pct_NH_AIAN_alone_ACS_08_12', 'pct_Census_UAA_CEN_2010',
       'pct_Pop_5_17_CEN_2010', 'pct_Pop_Under_5_CEN_2010',
       'pct_Pop_45_64_CEN_2010', 'pct_Hispanic_CEN_2010',
       'pct_Inst_GQ_CEN_2010', 'pct_Female_No_HB_ACS_08_12',
       'pct_MLT_U10p_ACS_08_12', 'pct_Males_CEN_2010', 'MHI_COHORT_1112',
       'pct_HHD_PPL_Und_18_CEN_2010', 'pct_Sngl_Prns_HHD_CEN_2010',
       'pct_NH_NHOPI_alone_ACS_08_12', 'MAS_COHORT_1112',
       'pct_Age5p_OthPacIsl_ACS_08_12'],
      dtype='object')
```

In [14]:

```
stats(rgs_target_data, 2)
```

```
'Statistics: Dataset row.count col.count -> (9785, 1)'
```

	ALL_RATE_1112
0	83.0
1	79.0
2	91.0

```
Index(['ALL_RATE_1112'], dtype='object')
```

In [15]:

```
#5-------- shuffle and split data: training 80%, testing 20%
from sklearn.model_selection import train_test_split
np.random.seed(99)

# Split the feature and target data into 70% for training and 30% for testing sets
cls_X_train, cls_X_test, cls_y_train, cls_y_test = \
train_test_split( cls_feature_data, cls_target_data, test_size = 0.3, random_state = 99)

# Success
info("cls Training and testing split was successful.  \nCount of training set is {} ({:.2f}%)  testing
set is {}  ({:.2f}%)  in total {}.".format(
        cls_X_train.shape[0], 100 * cls_X_train.shape[0]/cls_feature_data.shape[0],
        cls_X_test.shape[0], 100 * cls_X_test.shape[0]/cls_feature_data.shape[0],
        cls_feature_data.shape[0] ))

stats(cls_X_train, 2)
stats(cls_y_train, 2)
```

```
22:38:53 INFO cls Training and testing split was successful.
Count of training set is 6934 (69.99%)  testing set is 2973  (30.01%)  in total 9907.
```

```
'Statistics: Dataset row.count col.count -> (6934, 26)'
```

	pct_Prs_Blw_Pov_Lev_ACS_08_12	pct_Vacant_Units_ACS_08_12	pct_College_ACS_08_12	leaid11	pct_MrdCple
8218	8.575353	16.496283	8.419244	4816200	52.23

2130	pct_Prs_Blw_Pov_Lev_ACS_08_12	pct_Vacant_Units_ACS_08_12	pct_College_ACS_08_12	leaid11	pct_MrdCple
6448	8.396947	20.132450	20.900901	3813200	64.59

3 rows × 26 columns

```
Index(['pct_Prs_Blw_Pov_Lev_ACS_08_12', 'pct_Vacant_Units_ACS_08_12',
       'pct_College_ACS_08_12', 'leaid11', 'pct_MrdCple_HHD_CEN_2010',
       'CWD_COHORT_1112', 'County.1', 'pct_Mobile_Homes_ACS_08_12',
       'pct_NH_Blk_alone_CEN_2010', 'pct_TEA_Update_Leave_CEN_2010',
       'pct_NH_AIAN_alone_ACS_08_12', 'pct_Census_UAA_CEN_2010',
       'pct_Pop_5_17_CEN_2010', 'pct_Pop_Under_5_CEN_2010',
       'pct_Pop_45_64_CEN_2010', 'pct_Hispanic_CEN_2010',
       'pct_Inst_GQ_CEN_2010', 'pct_Female_No_HB_ACS_08_12',
       'pct_MLT_U10p_ACS_08_12', 'pct_Males_CEN_2010', 'MHI_COHORT_1112',
       'pct_HHD_PPL_Und_18_CEN_2010', 'pct_Sngl_Prns_HHD_CEN_2010',
       'pct_NH_NHOPI_alone_ACS_08_12', 'MAS_COHORT_1112',
       'pct_Age5p_OthPacIsl_ACS_08_12'],
      dtype='object')
```

```
'Statistics: Dataset row.count col.count -> (6934, 1)'
```

	Success_Pass_90
8218	0
2130	0
6448	0

```
Index(['Success_Pass_90'], dtype='object')
```

Reference:

- Stepwise regression
- Feature selection
- forward-selection-stepwise-regression-algorithm