> This is prepared for Udacity Machine Learning Engineer Nanodegree online class
> Author: jtmoogle @github.com
> Date: May 11, 2018

The *jtmoogle APIs* were developed for Predicting US High School Graduation Success capstone report.

Contact info: jtmoogle@gmail.com for question or suggestion

The sources files for APIs were available in GitHub at [jtmoogle]

- helper.py contains classes as static methods
  - MyLogger - logging to a file or console output for purpose of debug, info, error
  - MyHelper - common functions to load dataset, print out statistic summary, save to result to a file
- hsgraduation.py contains functions and methods
  1. load_gradcensus - load jtmoogle/data/GRADUATION_WITH_CENSUS.csv
  2. Classification
     (1) plot_cls_gradcensus - illustrate classification graduation census data
     (2) preproc_cls_data - prepare classification data cleaning
     (3) cls_feature_sel - classification feature selection
     (4) compare_cls_featranking - compared classification feature ranking (5) cls_stats - classification statistic summary
     (6) cls_acc_featimportance - mean accuracy score and feature importance result for classification feature (7) create_cls_sample - create sample for Training and Testing datasets
     (8) cls_acc_featimportance - calculated F score, accuracy, feature importance
     (9) cls_visual_benchmark - benchmark result and visualization
     (10) handy functions: cls_pca - PCA result for classification
  3. Regression
     (1) plot_rgs_gradcensus - illustrate regression graduation census data
     (2) preproc_rgs_data - prepare Regression data cleaning
     (3) rgs_feature_sel - regression feature selection
     (4) compare_rgs_featranking - compared regression feature ranking (5) rgs_stats - regression statistic summary
     (6) create_rgs_sample - create sample for Training and Testing datasets
     (7) rgs_r2_featimportanct - R2 score and feature importance for regression features
     (8) rgs_visual_benchmark - benchmark result and visualization
- runme.py which was capable to reproducible statistics summary results, and visualization seen in this report
- data/GRADUATION_WITH_CENSUS.csv (raw data) GRADUATION_WITH_CENSUS.csv.ds.csv (cleaner data, output data file via load_gradcensus method) ALL_DATA_SCHEMA_M.pdf (field definition)

How to execute python source from local environment Python IDE (i.e Spyder 3.2.6)

1. Manually load helper.py command: runfile('../capstone/jtmoogle/helper.py')
2. Manually load hsgraduation.py command: runfile('../capstone/jtmoogle/hsgraduation.py')
3. Manually load runme.py command: runfile('../capstone/jtmoogle/runme.py')
4. Execute runme program command: runme(3,3)

Note: Output of analytic results were saved to .txt, .csv files. The plotting images were saved to .png files.
The capstone report pulls content directly from output files and images located at ../saved folder.

How to execute runme.ipynb

1. Manually launch Anaconda 3 Prompt
2. type commands

```
set mypath=c:\github\capstone
setfile=runme.ipynb
activate capstone

pip install --ignore-installed --upgrade tensorflow-gpu
pip install ipykernel
```

```
cd %mypath%
python -c "from keras import backend"
python -m ipykernel install --user --name capstone --display-name "Python (capstone)"
python -c "import pandas"
python -c "import jtmoogle.helper"
python -c "import jtmoogle.hsgraduation"

jupyter notebook  %myfile%
```

3. Expect browser to launch 'runme.ipynb' URL= http://localhost:8888/notebooks/runme.ipynb
4. Manually click top menu "Cell" -> "Run All Below"

Expect each block to be executed sequentially. Time taken would be over 20 minutes depending on your GPU power and memory.

In [1]:

```
import os.path
from time import time
from jtmoogle.helper import MyHelper
from jtmoogle.hsgraduation import HSGraduation

t0 = time()
t1 = time()

MyHelper.printversion()

# create high school graduation class
hs = HSGraduation()

# load jtmoogle/data/GRADUATION_WITH_CENSUS.csv
rawdata = hs.load_gradcensus()
MyHelper.stats(rawdata, 1, 0)

load_time = time() - t1
print("--- load_time:  {0:8.3f}s".format(load_time))
t1 = time()
```

```
------------------------------
--> IPython version: 6.2.1
--> numpy version: 1.14.3
--> pandas version: 0.22.0
--> python version: 3.5.5
--> scikit-learn version: 0.19.1
--> sys version: 3.5.5 |Anaconda, Inc.| (default, Mar 12 2018, 17:44:09) [MSC v.1900 64 bit (AMD64)]
--> tensorflow version: 1.8.0
------------------------------
Currnet wdir=I:\_githup\capstone-report   Loading jtmoogle/data/GRADUATION_WITH_CENSUS.csv
Load dataset path=jtmoogle/data/GRADUATION_WITH_CENSUS.csv

'Statistics: dataset has 9907 (rows) samples with  576 (columns) features each'

Unique count for Scholl district=3158
State=48  count=320
COHORT average count

ALL_COHORT_1112    333.867266
ALL_RATE_1112       83.039090
MAM_COHORT_1112      8.163723
MAS_COHORT_1112     32.824961
MBL_COHORT_1112     88.001750
MHI_COHORT_1112     91.042030
MTR_COHORT_1112     11.228290
MWH_COHORT_1112    187.670722
CWD_COHORT_1112     41.203138
ECD_COHORT_1112    146.774154
dtype: float64

COHORT total count

ALL_COHORT_1112        3307623
ALL_RATE_1112           812538
MAM_COHORT_1112          30965
MAS_COHORT_1112         168589
MBL_COHORT_1112         553003
MHI_COHORT_1112         658507
MTR_COHORT_1112          46676
MWH_COHORT_1112     1.8358e+06
```

```
AMA_COHORT_1112      1.03903e+06
CWD_COHORT_1112           393902
ECD_COHORT_1112     1.43105e+06
dtype: object
```

```
Geography and Population Total for
  Tot_Population_CEN_2010 41,000,304
  RURAL_POP_CEN_2010  19,256,868
  URBANIZED_AREA_POP_CEN_2010 13,311,089
  URBAN_CLUSTER_POP_CEN_2010 8,432,347
Gender Total for
  Males_CEN_2010 20,353,022
  Females_CEN_2010 20,647,282
Age Total for
   Pop_under_5_CEN_2010  2,571,831
   Pop_5_17_CEN_2010  7,327,965
   Pop_18_24_CEN_2010  3,608,431
   Pop_25_44_CEN_2010  9,898,820
   Pop_45_64_CEN_2010  11,477,346
   Pop_65plus_CEN_2010 6,115,911
English Language Speaks Total for
  ENG_VW_ACS_08_12  337,275
  ENG_VW_SPAN_ACS_08_12  225,915
  ENG_VW_INDO_EURO_ACS_08_12  55,547
  ENG_VW_API_ACS_08_12  43,778
  ENG_VW_OTHER_ACS_08_12  12,035
Family Education Total for
   Not_HS_Grad_ACS_08_12  3,730,293.00
   College_ACS_08_12 6,266,894.00
Family Background Total for
  Pov_Univ_ACS_08_12  39,675,431
  Prs_Blw_Pov_Lev_ACS_08_12  5,457,898
  Civ_labor_16plus_ACS_08_12  20,216,951
  Civ_emp_16plus_ACS_08_12  18,543,130
  Civ_unemp_16plus_ACS_08_12  1,673,821
  Civ_labor_16_24_ACS_08_12 2,833,010
Income Total for
  PUB_ASST_INC_ACS_08_12  380,658
  Med_HHD_Inc_ACS_08_12  515,609,190
  Aggregate_HH_INC_ACS_08_12  1,035,306,469,500
Other Total for
  Born_US_ACS_08_12  38,404,354
  Born_foreign_ACS_08_12  2,492,610
  US_Cit_Nat_ACS_08_12  1,062,900
  NON_US_Cit_ACS_08_12  1,429,710
  MrdCple_Fmly_HHD_CEN_2010  8,326,407
  Not_MrdCple_HHD_CEN_2010  7,361,715
  Female_No_HB_CEN_2010 1,728,876
  NonFamily_HHD_CEN_2010 4,890,585
--- load_time:    10.298s
```

In [2]:

```
hs.plot_cls_gradcensus()  # Classification plot gradudation and census data
```

```
'Statistics: dataset has 9907 (rows) samples with  577 (columns) features each'
```

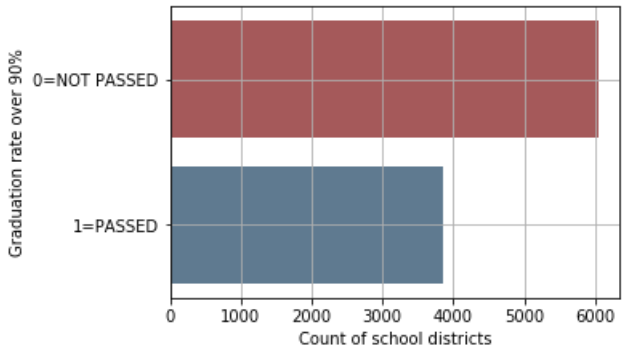
```
'Statistics: dataset has 9907 (rows) samples with  1 (columns) features each'
```

```
Success_Pass_90    int32
dtype: object
```

```
'-Data Summary->'
```

	Success_Pass_90
count	9907.00
mean	0.39
std	0.49
min	0.00
25%	0.00
50%	0.00
75%	1.00
max	1.00

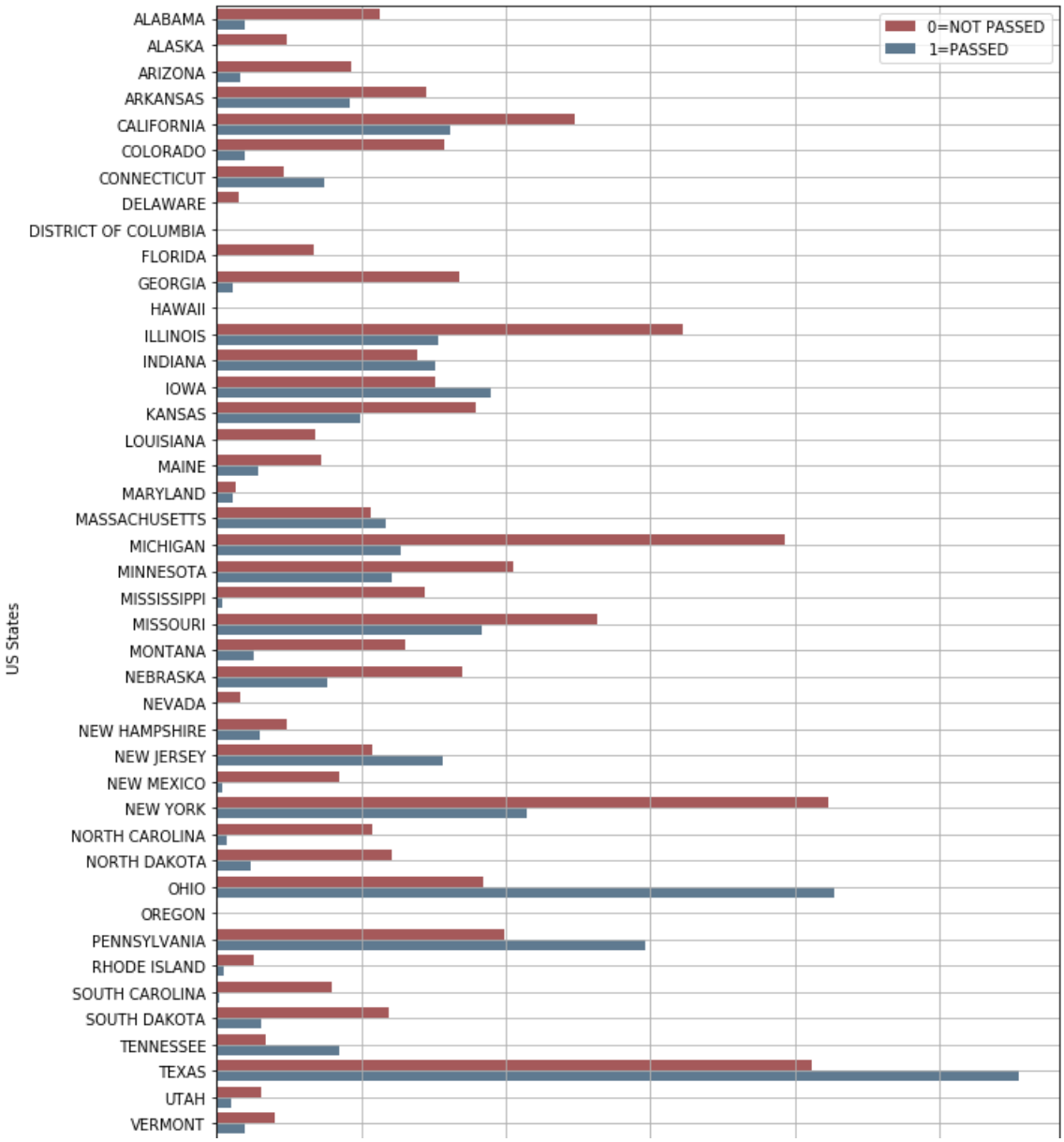
```
Classification dataset target variable (0=NOT PASSED  1=PASSED)
values count=
0    6050
1    3857
Name: Success_Pass_90, dtype: int64
```

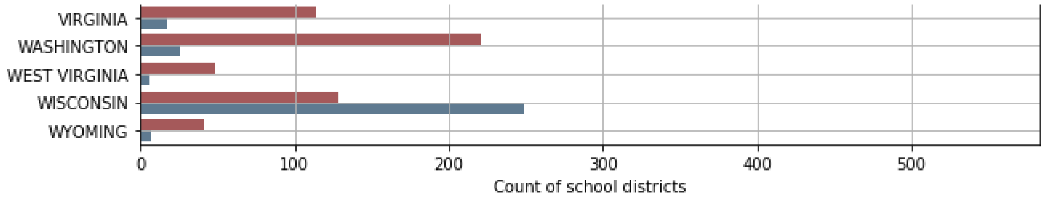


Success_Pass_90	0	1
STNAM		
ALABAMA	113.0	19.0
ALASKA	42.0	1.0
ARIZONA	91.0	16.0
ARKANSAS	145.0	92.0
CALIFORNIA	245.0	162.0
COLORADO	150.0	20.0
CONNECTICUT	46.0	75.0
DELAWARE	15.0	1.0
DISTRICT OF COLUMBIA	1.0	NaN
FLORIDA	67.0	NaN
GEORGIA	168.0	11.0
HAWAII	1.0	NaN
ILLINOIS	322.0	153.0
INDIANA	139.0	151.0
IOWA	131.0	190.0
KANSAS	175.0	99.0
LOUISIANA	68.0	NaN
MAINE	71.0	29.0
MARYLAND	13.0	11.0
MASSACHUSETTS	107.0	117.0
MICHIGAN	390.0	127.0
MINNESOTA	204.0	121.0
MISSISSIPPI	144.0	4.0
MISSOURI	260.0	183.0
MONTANA	115.0	26.0
NEBRASKA	168.0	77.0
NEVADA	16.0	NaN
NEW HAMPSHIRE	42.0	30.0
NEW JERSEY	108.0	156.0
NEW MEXICO	83.0	4.0
NEW YORK	419.0	215.0

Success_Pass_90 STNAM	0	1
NORTH CAROLINA	108.0	7.0
NORTH DAKOTA	111.0	24.0
OHIO	182.0	427.0
OREGON	1.0	NaN
PENNSYLVANIA	199.0	297.0
RHODE ISLAND	26.0	5.0
SOUTH CAROLINA	80.0	2.0
SOUTH DAKOTA	116.0	31.0
TENNESSEE	34.0	85.0
TEXAS	404.0	555.0
UTAH	31.0	10.0
VERMONT	40.0	19.0
VIRGINIA	114.0	17.0
WASHINGTON	207.0	26.0
WEST VIRGINIA	49.0	6.0
WISCONSIN	128.0	249.0
WYOMING	39.0	7.0

Save to a file saved/cls_pivot_count_st_success90.txt





In [3]:

```
hs.preproc_cls_data()  # preprocess classification data
```

```
1. Drop colums regex=ALL_COHORT_1112|MOE_|_FRMS_|Mail|Percentage|County|State|Tract|District|GIDTR|Trac
t|Flag|Response|Delete|Vacant|BILQ|Diff|Leave|Plumb
2. Select colums regex=Inc|INC|_COHORT_|pct_|avg_|_House_|_AREA_|ALL_|Success
3. Filter only datatype float64, int32/int64
4. Drop rows if col has NaN value
5. Get target data for targe column
6. Drop targe column
7. Fill in missing data with zero - impute NaN with zero
feature columns=Index(['MAM_COHORT_1112', 'MAS_COHORT_1112', 'MBL_COHORT_1112',
       'MHI_COHORT_1112', 'MTR_COHORT_1112', 'MWH_COHORT_1112',
       'CWD_COHORT_1112', 'ECD_COHORT_1112', 'LEP_COHORT_1112',
       'URBANIZED_AREA_POP_CEN_2010',
       ...
       'pct_Owner_Occp_HU_CEN_2010', 'pct_Owner_Occp_HU_ACS_08_12',
       'pct_Single_Unit_ACS_08_12', 'pct_MLT_U2_9_STRC_ACS_08_12',
       '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_Recent_Built_HU_ACS_08_12', 'pct_Census_UAA_CEN_2010'],
      dtype='object', length=152)
cls_feature_data

'Statistics: dataset has 9886 (rows) samples with  152 (columns) features each'

cls_target_data

'Statistics: dataset has 9886 (rows) samples with  1 (columns) features each'
```

In [4]:

```
hs.cls_feature_sel()  # feature selection
```

```
Save Classification Feature Select data to a file saved/cls_feature_sel.pkl
Save model to a file saved/cls_feature_sel.pkl
Number of Features: 20
Selected Features Indicator: [ True  True  True  True False  True  True  True  True False False  True
  True False  True False False False False False False False False False
 False False False False False False False False False False False False
 False  True False  True False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False  True  True
 False  True  True False False False False False False False False False
 False False False False False False False False False False False False
 False False  True False  True False False False False False False False
 False False False False False False False  True False False False False
 False False False False False False False False]
Feature Ranking: [  1   1   1   1  21   1   1   1   1  41  17   1   1   3   1  46  82  26
  59  73  76  43  78  74   8  55  58  37  44  40  29  18  31  53  80  75
  83   1  12   1  10   9   7  63  57  33  56  87  91  94  89  68  27  23
  77  93  95 128 127 102  97  99 103 105 107 108 111 113 115 117 119 121
 123 129 131 125 126 130 124 122 120 118 116 114 112 110 106 132 104 101
 100  98 133  22   1   1  66   1   1  24  11  30  50  34  72  51  60  79
  84  48  39  64  71  81  96  86  90  67  69  42  13   4   1  32   1  35
  15  62  45  36   2   5  92 109  47  65  70  49  20   1  25  19  38  52
  14  16  54   6  61  85  88  28]
Save to a file saved/cls_feature_sel_ranking.txt
['Aggr_House_Value_ACS_08_12', 'Aggregate_HH_INC_ACS_08_12', 'CWD_COHORT_1112', 'ECD_COHORT_1112', 'LEP
_COHORT_1112', 'MAM_COHORT_1112', 'MAS_COHORT_1112', 'MBL_COHORT_1112', 'MHI_COHORT_1112', 'MWH_COHORT
_1112', 'Med_HHD_Inc_ACS_08_12', 'pct_Civ_emp_16p_ACS_08_12', 'pct_College_ACS_08_12', 'pct_Female_No_H
B_CEN_2010', 'pct_Hispanic_CEN_2010', 'pct_NH_White_alone_CEN_2010', 'pct_Not_HS_Grad_ACS_08_12', 'pct_
Not_MrdCple_HHD_CEN_2010', 'pct_Prs_Blw_Pov_Lev_ACS_08_12', 'pct_Tot_Occp_Units_ACS_08_12']
Selected Features/columns: ['Aggr_House_Value_ACS_08_12', 'Aggregate_HH_INC_ACS_08_12', 'CWD_COHORT_111
2', 'ECD_COHORT_1112', 'LEP_COHORT_1112', 'MAM_COHORT_1112', 'MAS_COHORT_1112', 'MBL_COHORT_1112', 'MH
I_COHORT_1112', 'MWH_COHORT_1112', 'Med_HHD_Inc_ACS_08_12', 'pct_Civ_emp_16p_ACS_08_12', 'pct_College_A
CS_08_12', 'pct_Female_No_HB_CEN_2010', 'pct_Hispanic_CEN_2010', 'pct_NH_White_alone_CEN_2010', 'pct_No
t_HS_Grad_ACS_08_12', 'pct_Not_MrdCple_HHD_CEN_2010', 'pct_Prs_Blw_Pov_Lev_ACS_08_12', 'pct_Tot_Occp_Un
```

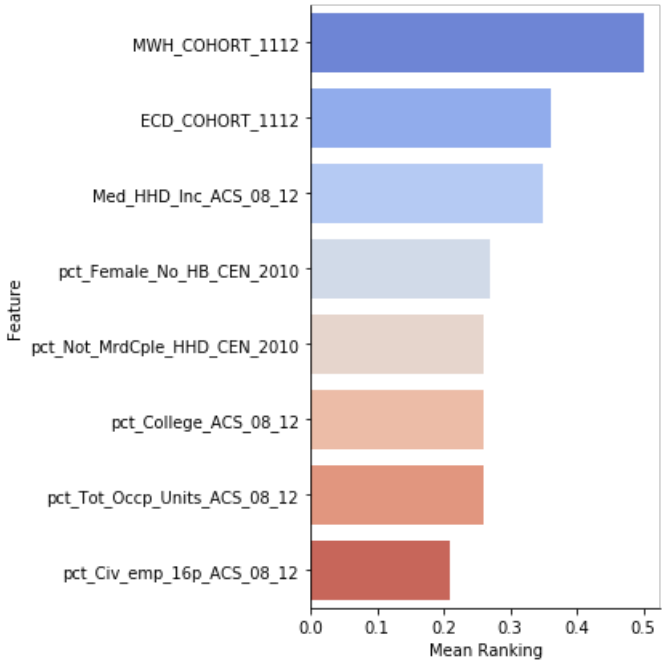
t_HS_Grad_ACS_08_12', 'pct_Not_MrdCple_HHD_CEN_2010', 'pct_Prs_Blw_Pov_Lev_ACS_08_12', 'pct_Tot_Occp_Units_ACS_08_12']

'Statistics: dataset has 9886 (rows) samples with 20 (columns) features each'

In [5]:

```
hs.compare_cls_featranking(minRanking=0.2)
```

```
Save to a file saved/cls_ds_allranking_ds.csv
Features   DT RF Ridge logit Mean
Aggr_House_Value_ACS_08_12  0.0 0.02 0.0 0.0 0.0
Aggregate_HH_INC_ACS_08_12  0.0 0.35 0.0 0.0 0.09
CWD_COHORT_1112  0.0 0.23 0.06 0.01 0.08
ECD_COHORT_1112  0.93 0.41 0.03 0.05 0.36
LEP_COHORT_1112  0.01 0.09 0.01 0.01 0.03
MAM_COHORT_1112  0.01 0.01 0.12 0.0 0.04
MAS_COHORT_1112  0.04 0.03 0.04 0.0 0.03
MBL_COHORT_1112  0.06 0.19 0.02 0.02 0.07
MHI_COHORT_1112  0.12 0.07 0.04 0.02 0.06
MWH_COHORT_1112  1.0 1.0 0.0 0.02 0.5
Med_HHD_Inc_ACS_08_12  0.23 0.18 0.0 1.0 0.35
pct_Civ_emp_16p_ACS_08_12  0.0 0.07 0.75 0.01 0.21
pct_College_ACS_08_12  0.03 0.09 0.91 0.0 0.26
pct_Female_No_HB_CEN_2010  0.0 0.08 1.0 0.0 0.27
pct_Hispanic_CEN_2010  0.0 0.0 0.45 0.0 0.11
pct_NH_White_alone_CEN_2010  0.02 0.03 0.69 0.0 0.18
pct_Not_HS_Grad_ACS_08_12  0.03 0.03 0.02 0.0 0.02
pct_Not_MrdCple_HHD_CEN_2010  0.08 0.09 0.86 0.01 0.26
pct_Prs_Blw_Pov_Lev_ACS_08_12  0.01 0.16 0.43 0.0 0.15
pct_Tot_Occp_Units_ACS_08_12  0.05 0.16 0.83 0.01 0.26
Save to a file saved/cls_ds_meanranking.csv
                        Feature  Mean Ranking
11               MWH_COHORT_1112          0.50
10               ECD_COHORT_1112          0.36
18         Med_HHD_Inc_ACS_08_12          0.35
1      pct_Female_No_HB_CEN_2010          0.27
7   pct_Not_MrdCple_HHD_CEN_2010          0.26
14         pct_College_ACS_08_12          0.26
4   pct_Tot_Occp_Units_ACS_08_12          0.26
0      pct_Civ_emp_16p_ACS_08_12          0.21
```



Selected Features/columns: ['MWH_COHORT_1112', 'ECD_COHORT_1112', 'Med_HHD_Inc_ACS_08_12', 'pct_Female_No_HB_CEN_2010', 'pct_Not_MrdCple_HHD_CEN_2010', 'pct_College_ACS_08_12', 'pct_Tot_Occp_Units_ACS_08_12', 'pct_Civ_emp_16p_ACS_08_12']

In [6]:

```
hs.cls_stats()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Success_Pass_90   R-squared:                       0.463
Model:                            OLS   Adj. R-squared:                  0.463
```

```
Model:                            OLS   Adj. R-squared:                  0.465
Method:                 Least Squares   F-statistic:                     1065.
Date:                Sat, 12 May 2018   Prob (F-statistic):               0.00
Time:                        21:15:56   Log-Likelihood:                -6291.3
No. Observations:                9886   AIC:                         1.260e+04
Df Residuals:                    9878   BIC:                         1.266e+04
Df Model:                           8
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
MWH_COHORT_1112              -3.173e-05   1.56e-05     -2.029      0.043   -6.24e-05   -1.07e-06
ECD_COHORT_1112              -5.833e-05   8.34e-06     -6.991      0.000   -7.47e-05    -4.2e-05
Med_HHD_Inc_ACS_08_12         1.685e-06   3.99e-07      4.227      0.000    9.03e-07    2.47e-06
pct_Female_No_HB_CEN_2010       -0.0035      0.001     -3.053      0.002      -0.006      -0.001
pct_Not_MrdCple_HHD_CEN_2010    -0.0046      0.001     -7.673      0.000      -0.006      -0.003
pct_College_ACS_08_12            0.0043      0.001      7.979      0.000       0.003       0.005
pct_Tot_Occp_Units_ACS_08_12     0.0058      0.000     14.098      0.000       0.005       0.007
pct_Civ_emp_16p_ACS_08_12       -0.0002      0.000     -0.426      0.670      -0.001       0.001
==============================================================================
Omnibus:                      276.301   Durbin-Watson:                   1.696
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1093.084
Skew:                           0.426   Prob(JB):                    4.36e-238
Kurtosis:                       1.612   Cond. No.                     1.57e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.57e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
Save to a file saved/cls_ols_statssummary.txt
Save to a file saved/cls_ols_statssummary.csv
```

In [7]:

```
hs.create_cls_sample()
```

```
cls Training and testing split was successful.  Split using target variable=['Success_Pass_90']
Count of training set is 6920 (70.00%)  testing set is 2966  (30.00%)  in total 9886.
```

```
'Statistics: dataset has 6920 (rows) samples with  8 (columns) features each'
```

	MWH_COHORT_1112	ECD_COHORT_1112	Med_HHD_Inc_ACS_08_12	pct_Female_No_HB_CEN_2010	pct_Not_Mr
7985	10	5	46858	7	40
2589	57	17	50000	3	30
9730	658	9	38761	10	65

```
'Statistics: dataset has 6920 (rows) samples with  1 (columns) features each'
```

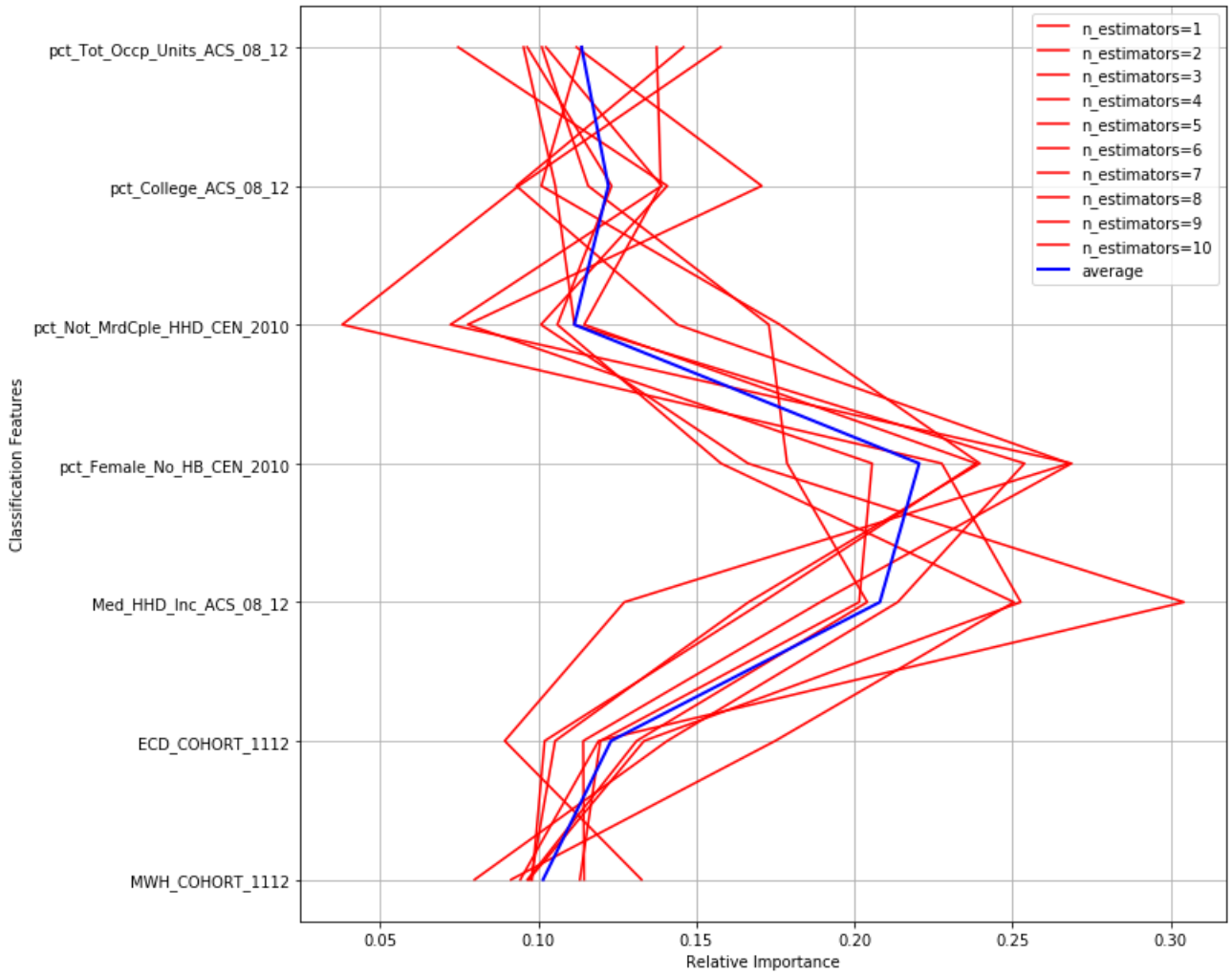
	Success_Pass_90
7985	0
2589	1
9730	1

In [8]:

```
hs.cls_acc_featimportance()
```

```
The mean Accuracy score for features are sorted the highest to lowest
['pct_Female_No_HB_CEN_2010', 'pct_Civ_emp_16p_ACS_08_12', 'pct_Tot_Occp_Units_ACS_08_12', 'pct_Not_Mrd
Cple_HHD_CEN_2010', 'pct_College_ACS_08_12', 'ECD_COHORT_1112', 'Med_HHD_Inc_ACS_08_12', 'MWH_COHORT_11
12']
Sorting Mean Accuracy score, the highest to lowest
#1 pct_Female_No_HB_CEN_2010 Accuracy=  +0.1844
#2 pct_Civ_emp_16p_ACS_08_12 Accuracy=  +0.1298
#3 pct_Tot_Occp_Units_ACS_08_12 Accuracy=  +0.0806
#4 pct_Not_MrdCple_HHD_CEN_2010 Accuracy=  +0.0735
#5 pct_College_ACS_08_12 Accuracy=  +0.0725
#6 ECD_COHORT_1112      Accuracy=  +0.0314
#7 Med_HHD_Inc_ACS_08_12 Accuracy=  +0.0273
#8 MWH_COHORT_1112      Accuracy=  +0.0158
```

```
--> pct_Female_No_HB_CEN_2010 has the highest mean accuracy score 0.1844
Save to a file saved/cls_accuracy.txt
Feature importance -->

[('ECD_COHORT_1112', 0.1229597152342136),
 ('MWH_COHORT_1112', 0.10148253213589394),
 ('Med_HHD_Inc_ACS_08_12', 0.20799175583411103),
 ('pct_College_ACS_08_12', 0.12211245550432097),
 ('pct_Female_No_HB_CEN_2010', 0.22042542558328787),
 ('pct_Not_MrdCple_HHD_CEN_2010', 0.11137556408519289),
 ('pct_Tot_Occp_Units_ACS_08_12', 0.11365255162297974)]
```




```
Sorting feature importance, the hight to lowest

[(0.22042542558328787, 'pct_Female_No_HB_CEN_2010'),
 (0.20799175583411103, 'Med_HHD_Inc_ACS_08_12'),
 (0.1229597152342136, 'ECD_COHORT_1112'),
 (0.12211245550432097, 'pct_College_ACS_08_12'),
 (0.11365255162297974, 'pct_Tot_Occp_Units_ACS_08_12'),
 (0.11137556408519289, 'pct_Not_MrdCple_HHD_CEN_2010'),
 (0.10148253213589394, 'MWH_COHORT_1112')]

Save to a file saved/cls_featimportance.txt
#1  iloc=3 pct_Female_No_HB_CEN_2010  importance= 0.22042542558328787
#2  iloc=2 Med_HHD_Inc_ACS_08_12  importance= 0.20799175583411103
#3  iloc=1 ECD_COHORT_1112  importance= 0.1229597152342136
#4  iloc=5 pct_College_ACS_08_12  importance= 0.12211245550432097
#5  iloc=6 pct_Tot_Occp_Units_ACS_08_12  importance= 0.11365255162297974
#6  iloc=4 pct_Not_MrdCple_HHD_CEN_2010  importance= 0.11137556408519289
#7  iloc=0 MWH_COHORT_1112  importance= 0.10148253213589394
Visualize Features in Correlation Matrix->

Data points considered outliers for the feature --> 'MWH_COHORT_1112'
   Q1=35.0000 Q3= 200.000000 step= 1.5*(Q3-Q1) = 247.5000   Feeature Outlier cnt= 855

Data points considered outliers for the feature --> 'ECD_COHORT_1112'
   Q1=15.0000 Q3= 99.000000 step= 1.5*(Q3-Q1) = 126.0000   Feeature Outlier cnt= 1187

Data points considered outliers for the feature --> 'Med_HHD_Inc_ACS_08_12'
   Q1=38838.5000 Q3= 59707.000000 step= 1.5*(Q3-Q1) = 31302.7500   Feeature Outlier cnt= 641
```

```
   Q1=38838.5000 Q3= 59707.000000 step= 1.5*(Q3-Q1) = 31302.7500   Feeature Outlier cnt= 641

Data points considered outliers for the feature --> 'pct_Female_No_HB_CEN_2010'
   Q1=7.0000 Q3= 13.000000 step= 1.5*(Q3-Q1) = 9.0000   Feeature Outlier cnt= 454

Data points considered outliers for the feature --> 'pct_Not_MrdCple_HHD_CEN_2010'
   Q1=39.0000 Q3= 53.000000 step= 1.5*(Q3-Q1) = 21.0000   Feeature Outlier cnt= 283

Data points considered outliers for the feature --> 'pct_College_ACS_08_12'
   Q1=12.0000 Q3= 26.000000 step= 1.5*(Q3-Q1) = 21.0000   Feeature Outlier cnt= 675

Data points considered outliers for the feature --> 'pct_Tot_Occp_Units_ACS_08_12'
   Q1=81.0000 Q3= 93.000000 step= 1.5*(Q3-Q1) = 18.0000   Feeature Outlier cnt= 595

Data points considered outliers for the feature --> 'pct_Civ_emp_16p_ACS_08_12'
   Q1=89.0000 Q3= 94.000000 step= 1.5*(Q3-Q1) = 7.5000   Feeature Outlier cnt= 530


data size=9886 max_idx=9906  Outliers for all features =5220
Note: Also found duplicate outliers in multiple features =1501

Removed duplicated outlier data -> good datasize=(8387, 8)   target datasize=(8387, 1)


'Before log-transformed, log_data Mean=Average & Median=50% -> '
```

	MWH_COHORT_1112	ECD_COHORT_1112	Med_HHD_Inc_ACS_08_12	pct_Female_No_HB_CEN_2010	pct_Not_M
mean	169.28	124.03	51945.61	10.35	46.49
50%	84.00	34.00	47569.00	9.00	45.00

	MWH_COHORT_1112	ECD_COHORT_1112	Med_HHD_Inc_ACS_08_12	pct_Female_N
MWH_COHORT_1112	1.00	0.51	0.22	0.04
ECD_COHORT_1112	0.51	1.00	0.00	0.18
Med_HHD_Inc_ACS_08_12	0.22	0.00	1.00	-0.35
pct_Female_No_HB_CEN_2010	0.04	0.18	-0.35	1.00
pct_Not_MrdCple_HHD_CEN_2010	0.00	0.13	-0.52	0.62
pct_College_ACS_08_12	0.25	0.04	0.73	-0.30
pct_Tot_Occp_Units_ACS_08_12	0.16	0.05	0.35	0.08
pct_Civ_emp_16p_ACS_08_12	-0.05	-0.09	0.33	-0.30

```
'After log-transformed, log_data Mean=Average & Median=50% -> '
```

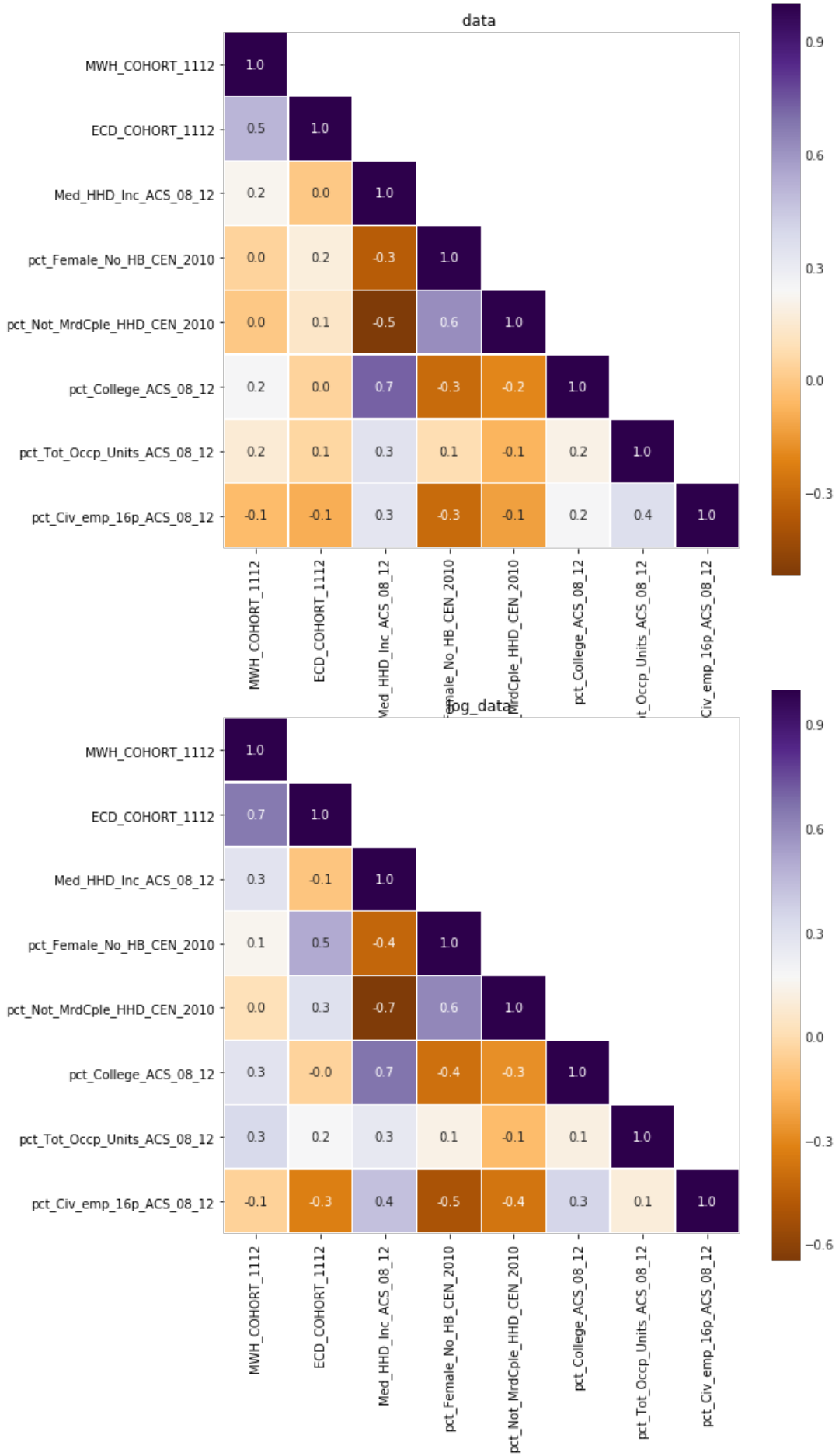
	MWH_COHORT_1112	ECD_COHORT_1112	Med_HHD_Inc_ACS_08_12	pct_Female_No_HB_CEN_2010	pct_Not_M
mean	-inf	-inf	-inf	-inf	-inf
50%	4.430000	3.530000	10.770000	2.200000	3.810000

```
'Statistics: dataset has 8387 (rows) samples with  8 (columns) features each'

log_data in Correlation Matrix ->
```

	MWH_COHORT_1112	ECD_COHORT_1112	Med_HHD_Inc_ACS_08_12	pct_Female_N
MWH_COHORT_1112	1.00	0.65	0.29	0.15
ECD_COHORT_1112	0.65	1.00	-0.10	0.51
Med_HHD_Inc_ACS_08_12	0.29	-0.10	1.00	-0.43
pct_Female_No_HB_CEN_2010	0.15	0.51	-0.43	1.00
pct_Not_MrdCple_HHD_CEN_2010	0.02	0.30	-0.65	0.61
pct_College_ACS_08_12	0.29	-0.04	0.66	-0.39
pct_Tot_Occp_Units_ACS_08_12	0.34	0.18	0.26	0.13

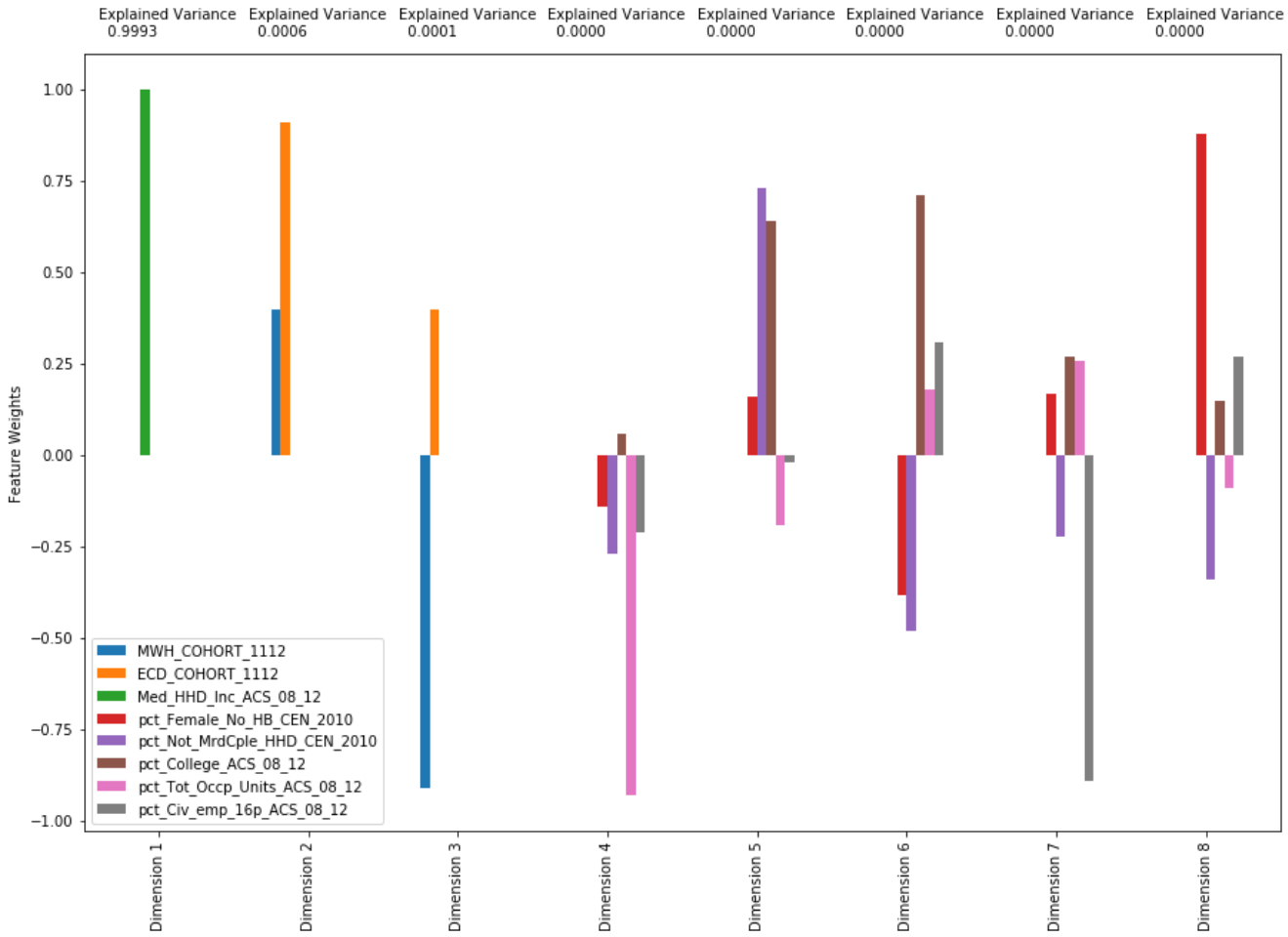
pct_Civ_emp_16p_ACS_08_12	MWH_COHORT_1112	ECD_COHORT_1112	Med_HHD_Inc_ACS_08_12	pct_Female_N
0.05	0.38	0.12	0.51	

Visualize comparing data and log_data in Correlation Matrix



Extracting the top 8 features from 8387 data points

```
 PCA Explained variance ratio=[1. 0. 0. 0. 0. 0. 0. 0.]
[[ 3.08107182e-03  7.11794975e-05  9.99995074e-01 -9.07757219e-05
  -2.76620076e-04  4.62275956e-04  2.08763532e-04  1.09799804e-04]
 [ 4.03540315e-01  9.14949223e-01 -1.30824834e-03  2.03373118e-03
   3.19362834e-03  1.54617801e-03  1.47248681e-03 -1.58973741e-03]
 [-9.14931983e-01  4.03556868e-01  2.79220484e-03  2.45974260e-04
  -1.41953033e-03 -4.03227834e-03 -3.27371494e-03  2.11474416e-03]
 [ 3.30895143e-03  8.30609329e-04  9.02043830e-05 -1.44780245e-01
  -2.70860250e-01  6.04114435e-02 -9.26123861e-01 -2.10491829e-01]
 [-4.07053998e-03 -1.89316504e-03 -2.27338404e-05  1.57073227e-01
   7.30603745e-01  6.35540994e-01 -1.93350399e-01 -1.51421105e-02]
 [-1.42285888e-03  2.20441145e-03 -5.60175727e-04 -3.81485746e-01
  -4.79593069e-01  7.06537383e-01  1.76229221e-01  3.06920356e-01]
 [-3.88938193e-03 -3.03555501e-04 -1.11438164e-04  1.68518250e-01
  -2.17451135e-01  2.65131182e-01  2.56329273e-01 -8.87863802e-01]
 [ 7.19602513e-04 -7.36105811e-04 -9.68280224e-05  8.83421022e-01
  -3.39920716e-01  1.51425303e-01 -9.01989671e-02  2.70101201e-01]]
```



In [9]:

```
hs.cls_visual_benchmark( )
```

```
================================================================================
Gaussian Naive Bayes(NB)
________________________________________________________________________________
Training:
GaussianNB(priors=None)
fit -> cls.score:    0.551
--- train time:    0.000s
--- test time:     0.000s
confusion matrix:
[[ 607 1205]
 [ 126 1028]]
classification report:
                 precision    recall  f1-score   support

Success_Pass_90       0.83      0.33      0.48      1812

    avg / total       0.69      0.55      0.53      2966

cross_val_score ------>
```

cross_val_score ------>
 Accuracy: mean=   0.546 std=   0.021
 AUC: mean=   0.717 std=   0.013
 precision: mean=   0.458 std=   0.022
 avg precision: mean=   0.636 std=   0.025
 recall: mean=   0.879 std=   0.015
 f1: mean=   0.601 std=   0.019
Scores ------>
accuracy ROC_AUC precision avgprecision recall f1
   0.546   0.717   0.458   0.636   0.879   0.601

==
Logistic Regression(Logistic)

Training:
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
fit -> cls.score: 0.687
--- train time: 0.031s
--- test time: 0.000s
dimensionality/count(non zero of coef_): 8
density: 1.000
coef_-->
Target feature: Success_Pass_90
Top 8 Features

[(0.02975726890549222, 'pct_Tot_Occp_Units_ACS_08_12'),
 (0.024835023803007724, 'pct_College_ACS_08_12'),
 (0.004329614480344245, 'pct_Female_No_HB_CEN_2010'),
 (0.0015162430496520126, 'MWH_COHORT_1112'),
 (3.18707536343517e-06, 'Med_HHD_Inc_ACS_08_12'),
 (-0.004456572483739847, 'ECD_COHORT_1112'),
 (-0.02275619755364649, 'pct_Civ_emp_16p_ACS_08_12'),
 (-0.032510457674538515, 'pct_Not_MrdCple_HHD_CEN_2010')]

confusion matrix:
[[1576 236]
 [693 461]]
classification report:
                 precision   recall f1-score   support

Success_Pass_90   0.69   0.87   0.77   1812

    avg / total   0.68   0.69   0.67   2966

cross_val_score ------>
 Accuracy: mean=   0.701 std=   0.010
 AUC: mean=   0.734 std=   0.013
 precision: mean=   0.690 std=   0.047
 avg precision: mean=   0.662 std=   0.024
 recall: mean=   0.426 std=   0.021
 f1: mean=   0.526 std=   0.019
Scores ------>
accuracy ROC_AUC precision avgprecision recall f1
   0.701   0.734   0.690   0.662   0.426   0.526

==
Random Forest(RF)

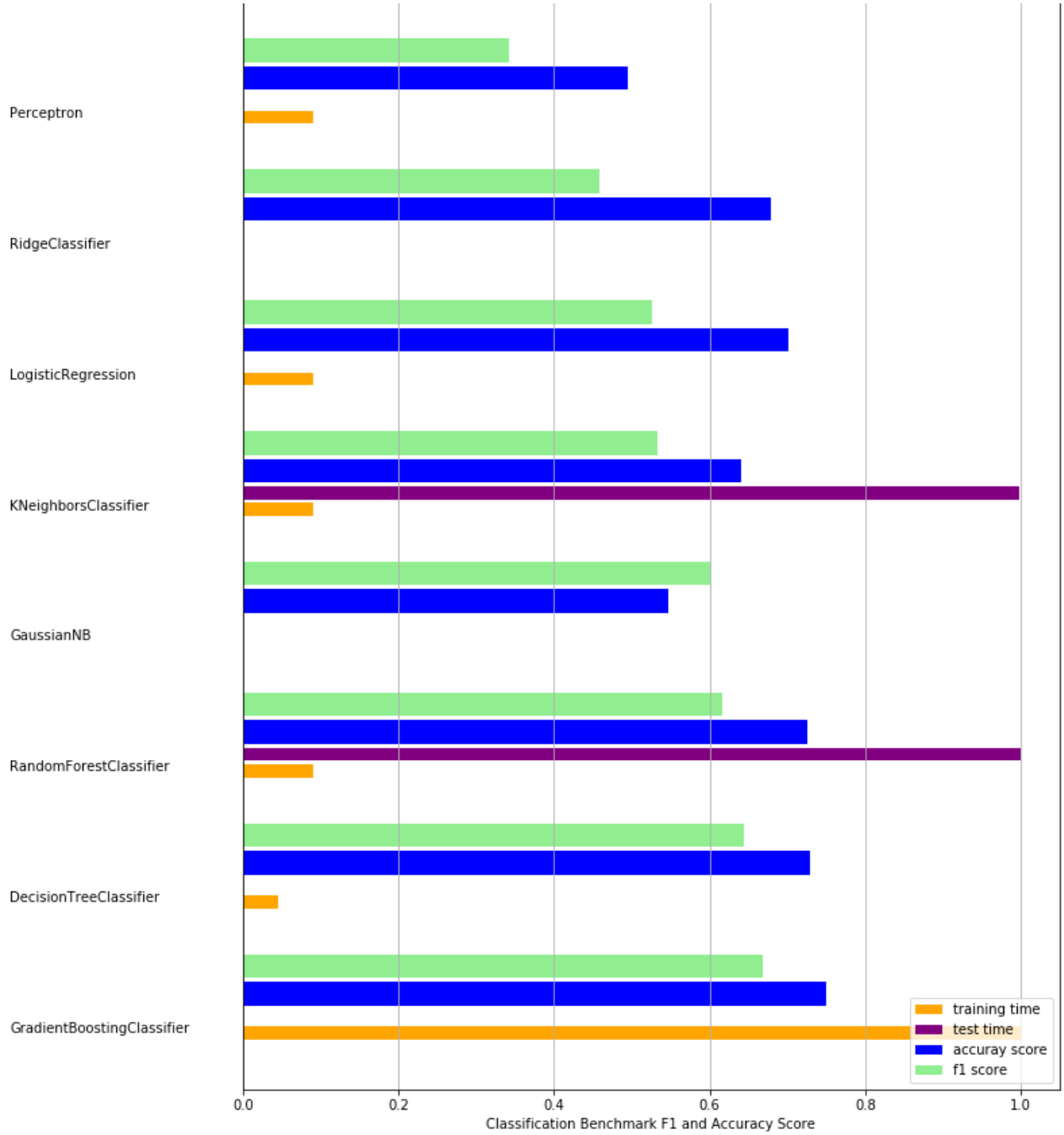
Training:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=5, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
fit -> cls.score: 0.733
--- train time: 0.031s
--- test time: 0.016s
confusion matrix:
[[1521 291]
 [500 654]]
classification report:
                 precision   recall f1-score   support

Success_Pass_90   0.75   0.84   0.79   1812

```
Success_Pass_90       0.75      0.84      0.79      1812

    avg / total       0.73      0.73      0.73      2966

cross_val_score ------>
 Accuracy: mean=   0.726 std=   0.008
 AUC: mean=   0.810 std=   0.015
 precision: mean=   0.694 std=   0.044
 avg precision: mean=   0.727 std=   0.024
 recall: mean=   0.569 std=   0.032
 f1: mean=   0.616 std=   0.013
Scores ------>
accuracy ROC_AUC precision avgprecision recall f1
   0.726    0.810    0.694    0.727    0.569    0.616

================================================================================
K-Nearest Neighbors(KNN)
________________________________________________________________________________
Training:
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=3, p=2,
           weights='uniform')
fit -> cls.score:    0.630
--- train time:    0.031s
--- test time:     0.016s
confusion matrix:
[[1260  552]
 [ 545  609]]
classification report:
                precision    recall  f1-score   support

Success_Pass_90       0.70      0.70      0.70      1812

    avg / total       0.63      0.63      0.63      2966

cross_val_score ------>
 Accuracy: mean=   0.641 std=   0.007
 AUC: mean=   0.654 std=   0.011
 precision: mean=   0.541 std=   0.022
 avg precision: mean=   0.505 std=   0.024
 recall: mean=   0.527 std=   0.029
 f1: mean=   0.533 std=   0.017
Scores ------>
accuracy ROC_AUC precision avgprecision recall f1
   0.641    0.654    0.541    0.505    0.527    0.533

================================================================================
Decision Tree(DT)
________________________________________________________________________________
Training:
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
fit -> cls.score:    0.726
--- train time:    0.016s
--- test time:     0.000s
confusion matrix:
[[1476  336]
 [ 476  678]]
classification report:
                precision    recall  f1-score   support

Success_Pass_90       0.76      0.81      0.78      1812

    avg / total       0.72      0.73      0.72      2966

cross_val_score ------>
 Accuracy: mean=   0.730 std=   0.012
 AUC: mean=   0.803 std=   0.016
 precision: mean=   0.661 std=   0.038
 avg precision: mean=   0.688 std=   0.034
 recall: mean=   0.632 std=   0.054
 f1: mean=   0.644 std=   0.028
Scores ------>
```

```
accuracy ROC_AUC precision avgprecision recall f1
   0.730    0.803    0.661    0.688    0.632    0.644

================================================================================
Ridge Classifier(Ridge)
--------------------------------------------------------------------------------
Training:
RidgeClassifier(alpha=1.0, class_weight=None, copy_X=True, fit_intercept=True,
        max_iter=None, normalize=False, random_state=None, solver='auto',
        tol=0.001)
fit -> cls.score:    0.671
--- train time:    0.000s
--- test time:     0.000s
dimensionality/count(non zero of coef_): 8
density:        1.000
coef_-->
Target feature: Success_Pass_90
Top 8 Features

[(0.011803787382842651, 'pct_Tot_Occp_Units_ACS_08_12'),
 (0.009909162918768442, 'pct_College_ACS_08_12'),
 (2.5071412005240144e-06, 'Med_HHD_Inc_ACS_08_12'),
 (-7.812602252548886e-05, 'MWH_COHORT_1112'),
 (-9.502850153363884e-05, 'ECD_COHORT_1112'),
 (-0.004322844035182331, 'pct_Civ_emp_16p_ACS_08_12'),
 (-0.006484155595645226, 'pct_Female_No_HB_CEN_2010'),
 (-0.011267922757478383, 'pct_Not_MrdCple_HHD_CEN_2010')]

confusion matrix:
[[1606  206]
 [ 771  383]]
classification report:
                precision    recall  f1-score   support

Success_Pass_90       0.68      0.89      0.77      1812

    avg / total       0.67      0.67      0.64      2966

cross_val_score ------>
 Accuracy: mean=   0.679 std=   0.007
 AUC: mean=   0.701 std=   0.013
 precision: mean=   0.674 std=   0.040
 avg precision: mean=   0.621 std=   0.024
 recall: mean=   0.349 std=   0.029
 f1: mean=   0.458 std=   0.018
Scores ------>
accuracy ROC_AUC precision avgprecision recall f1
   0.679    0.701    0.674    0.621    0.349    0.458

================================================================================
Perceptron
--------------------------------------------------------------------------------
Training:
Perceptron(alpha=0.1, class_weight=None, eta0=1.0, fit_intercept=True,
      max_iter=None, n_iter=50, n_jobs=1, penalty=None, random_state=0,
      shuffle=True, tol=None, verbose=0, warm_start=False)
fit -> cls.score:    0.542
--- train time:    0.031s
--- test time:     0.000s
dimensionality/count(non zero of coef_): 8
density:        1.000
coef_-->
Target feature: Success_Pass_90
Top 8 Features

[(957542.0, 'MWH_COHORT_1112'),
 (147685.0, 'pct_College_ACS_08_12'),
 (11040.0, 'Med_HHD_Inc_ACS_08_12'),
 (-278792.0, 'pct_Female_No_HB_CEN_2010'),
 (-1023661.0, 'pct_Tot_Occp_Units_ACS_08_12'),
 (-1127727.0, 'pct_Not_MrdCple_HHD_CEN_2010'),
 (-1446010.0, 'pct_Civ_emp_16p_ACS_08_12'),
 (-3252665.0, 'ECD_COHORT_1112')]

confusion matrix:
[[ 575 1237]
 [ 122 1032]]
classification report:
                precision    recall  f1-score   support
```

```
Success_Pass_90       0.82      0.32      0.46      1812

    avg / total       0.68      0.54      0.51      2966

cross_val_score ------>
 Accuracy: mean=   0.495 std=   0.093
 AUC: mean=   0.586 std=   0.135
 precision: mean=   0.440 std=   0.320
 avg precision: mean=   0.512 std=   0.132
 recall: mean=   0.592 std=   0.483
 f1: mean=   0.342 std=   0.278
Scores ------>
accuracy ROC_AUC precision avgprecision recall f1
   0.495    0.586    0.440    0.512    0.592    0.342

================================================================================
Gradient Boosting Classifier(GB)
________________________________________________________________________________
Training:
GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.1, loss='deviance', max_depth=None,
              max_features=None, max_leaf_nodes=4,
              min_impurity_decrease=0.0, min_impurity_split=None,
              min_samples_leaf=1, min_samples_split=5,
              min_weight_fraction_leaf=0.0, n_estimators=100,
              presort='auto', random_state=99, subsample=1.0, verbose=0,
              warm_start=False)
fit -> cls.score:    0.749
--- train time:    0.344s
--- test time:     0.000s
confusion matrix:
[[1492  320]
 [ 423  731]]
classification report:
                precision    recall  f1-score   support

Success_Pass_90       0.78      0.82      0.80      1812

    avg / total       0.75      0.75      0.75      2966

cross_val_score ------>
 Accuracy: mean=   0.751 std=   0.009
 AUC: mean=   0.827 std=   0.016
 precision: mean=   0.695 std=   0.035
 avg precision: mean=   0.756 std=   0.026
 recall: mean=   0.645 std=   0.015
 f1: mean=   0.668 std=   0.014
Scores ------>
accuracy ROC_AUC precision avgprecision recall f1
   0.751    0.827    0.695    0.756    0.645    0.668

Benchmark Summary for Classification (sorted by f1 score Desc (higher to lower))
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   cls_names  train_time  test_time  accuracy_score  \
7  GradientBoostingClassifier    1.000000   0.000000        0.750723
4      DecisionTreeClassifier    0.045451   0.000000        0.729913
2      RandomForestClassifier    0.090897   1.000000        0.726156
0                  GaussianNB    0.000000   0.000000        0.546243
3        KNeighborsClassifier    0.090935   0.999131        0.641040
1          LogisticRegression    0.090888   0.000000        0.701012
5             RidgeClassifier    0.000000   0.000000        0.679335
6                  Perceptron    0.090905   0.000000        0.495376

    roc_auc  precision  avg_precision    recall  f1_score
7  0.826509   0.694685       0.755856  0.644973  0.668220
4  0.802967   0.661430       0.687977  0.632278  0.644315
2  0.810488   0.693926       0.726633  0.568592  0.616142
0  0.717272   0.457517       0.635967  0.878520  0.601276
3  0.653535   0.540704       0.504605  0.526625  0.532865
1  0.734298   0.690321       0.661984  0.425830  0.525645
5  0.701283   0.674011       0.621319  0.348742  0.457826
6  0.586482   0.439788       0.511647  0.592483  0.342035
Save to a file saved/cls_visual_benchmark.txt
Save to a file saved/cls_visual_benchmark.csv
```



In [10]:

```
exec_time = time() - t1
print("--- Classification exec_time:  {0:8.3f}s".format(exec_time))
t1 = time()
```

```
--- Classification exec_time:    56.643s
```

In [11]:

```
# Regression
rawdata = hs.load_gradcensus()
```

```
Currnet wdir=I:\_github\capstone-report   Loading jtmoogle/data/GRADUATION_WITH_CENSUS.csv
Load dataset path=jtmoogle/data/GRADUATION_WITH_CENSUS.csv
```

In [12]:

```
hs.plot_rgs_gradcensus()
```

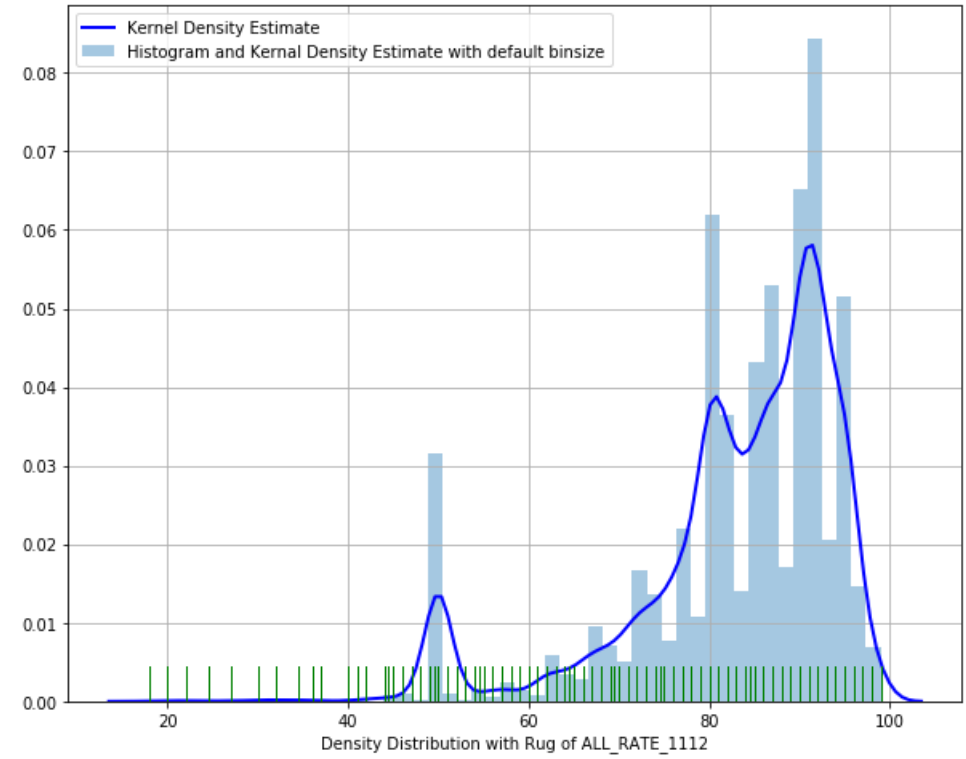
```
Regression dataset feature variables

'Statistics: dataset has 9907 (rows) samples with  576 (columns) features each'

'Statistics: dataset has 9907 (rows) samples with  1 (columns) features each'

ALL_RATE_1112    float64
dtype: object
```

'-Data Summary->'

	ALL_RATE_1112
count	9785.00
mean	83.04
std	11.87
min	18.00
25%	80.00
50%	87.00
75%	92.00
max	99.00

	STNAM	ALL_COHORT_1112	ALL_RATE_1112	MAM_COHORT_1112	MAM_RATE_1112	MAS_COHORT_1112	MAS_
0	ALABAMA	268	83.0	NaN	NaN	NaN	NaN
1	ALABAMA	424	79.0	2.0	PS	1.0	PS
2	ALABAMA	1042	91.0	1.0	PS	71.0	85-89
3	ALABAMA	836	91.0	4.0	PS	44.0	GE90
4	ALABAMA	117	72.0	NaN	NaN	NaN	NaN

5 rows × 576 columns



In [13]:

```
hs.preproc_rgs_data()
```

```
1. Drop colums regex=ALL_COHORT_1112|MOE_|_FRMS_|Mail|Percentage|County|State|Tract|District|GIDTR|Trac
t|Flag|Response|Delete|Vacant|BILQ|Diff|Leave|Plumb
2. Select colums regex=Inc|INC|_COHORT_|pct_|avg_|_House_|_AREA_|ALL_|Success
3. Filter only datatype float64, int32/int64
4. Drop rows if col has NaN value
5. Get target data for targe column
6. Drop targe column
7. Fill in missing data with zero - impute NaN with zero
feature columns=Index(['MAM_COHORT_1112', 'MAS_COHORT_1112', 'MBL_COHORT_1112',
       'MHI_COHORT_1112', 'MTR_COHORT_1112', 'MWH_COHORT_1112',
       'CWD_COHORT_1112', 'ECD_COHORT_1112', 'LEP_COHORT_1112',
```

```
       'URBANIZED_AREA_POP_CEN_2010',
       ...
       'pct_Owner_Occp_HU_CEN_2010', 'pct_Owner_Occp_HU_ACS_08_12',
       'pct_Single_Unit_ACS_08_12', 'pct_MLT_U2_9_STRC_ACS_08_12',
       '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_Recent_Built_HU_ACS_08_12', 'pct_Census_UAA_CEN_2010'],
      dtype='object', length=152)
rgs_feature_data
```

'Statistics: dataset has 9755 (rows) samples with 152 (columns) features each'

rgs_target_data

'Statistics: dataset has 9755 (rows) samples with 1 (columns) features each'

In [14]:

```
hs.rgs_feature_sel()
```

```
Add  Aggregate_HH_INC_ACS_08_12     with p-value 5.2137e-218
Add  pct_NH_AIAN_alone_ACS_08_12    with p-value 4.48068e-84
Add  pct_NH_Blk_alone_CEN_2010      with p-value 4.22215e-67
Add  pct_Tot_Occp_Units_CEN_2010    with p-value 1.08143e-67
Add  pct_RURAL_POP_CEN_2010         with p-value 2.11911e-28
Add  MBL_COHORT_1112                with p-value 3.34109e-23
Add  pct_Hispanic_CEN_2010          with p-value 3.00918e-17
Add  pct_Sngl_Prns_HHD_CEN_2010     with p-value 1.07871e-21
Add  URBANIZED_AREA_POP_CEN_2010    with p-value 1.14009e-10
Add  pct_Pop_Under_5_CEN_2010       with p-value 2.1622e-10
Add  pct_HHD_PPL_Und_18_CEN_2010    with p-value 2.08996e-18
Drop pct_Sngl_Prns_HHD_CEN_2010     with p-value 0.0572376
Add  pct_Pop_45_64_CEN_2010         with p-value 4.50964e-13
Add  pct_PUB_ASST_INC_ACS_08_12     with p-value 3.99887e-08
Add  pct_Census_UAA_CEN_2010        with p-value 1.7931e-07
Add  pct_Rel_Under_6_CEN_2010       with p-value 3.23491e-07
Add  pct_MLT_U10p_ACS_08_12         with p-value 7.81601e-08
Add  pct_Female_No_HB_ACS_08_12     with p-value 1.89118e-05
Add  pct_Prs_Blw_Pov_Lev_ACS_08_12  with p-value 1.32895e-05
Add  Med_House_value_ACS_08_12      with p-value 0.000104158
Add  pct_NO_PH_SRVC_ACS_08_12       with p-value 9.86749e-05
Add  pct_NH_NHOPI_alone_CEN_2010    with p-value 0.000140634
Add  pct_Pop_25_44_ACS_08_12        with p-value 0.000226951
Add  pct_Females_CEN_2010           with p-value 1.18412e-06
Add  pct_Civ_emp_16p_ACS_08_12      with p-value 2.4276e-09
Add  pct_Inst_GQ_CEN_2010           with p-value 4.56674e-06
Add  pct_NonFamily_HHD_CEN_2010     with p-value 0.000129048
Add  pct_Mobile_Homes_ACS_08_12     with p-value 0.000322811
Add  pct_ENG_VW_INDOEURO_ACS_08_12  with p-value 0.00128256
Add  pct_Age5p_WGerman_ACS_08_12    with p-value 0.000596035
Add  pct_Rel_Family_HHDS_CEN_2010   with p-value 0.0020856
Add  pct_Age5p_German_ACS_08_12     with p-value 0.00293546
Save Regression Feature Selection to a file saved/rgs_feature_sel.txt
Number of Features: 30
Regression Selected Features/columns: ['Aggregate_HH_INC_ACS_08_12', 'pct_NH_AIAN_alone_ACS_08_12', 'pc
t_NH_Blk_alone_CEN_2010', 'pct_Tot_Occp_Units_CEN_2010', 'pct_RURAL_POP_CEN_2010', 'MBL_COHORT_1112', '
pct_Hispanic_CEN_2010', 'URBANIZED_AREA_POP_CEN_2010', 'pct_Pop_Under_5_CEN_2010', 'pct_HHD_PPL_Und_18_
CEN_2010', 'pct_Pop_45_64_CEN_2010', 'pct_PUB_ASST_INC_ACS_08_12', 'pct_Census_UAA_CEN_2010', 'pct_Rel_
Under_6_CEN_2010', 'pct_MLT_U10p_ACS_08_12', 'pct_Female_No_HB_ACS_08_12', 'pct_Prs_Blw_Pov_Lev_ACS_08_
12', 'Med_House_value_ACS_08_12', 'pct_NO_PH_SRVC_ACS_08_12', 'pct_NH_NHOPI_alone_CEN_2010', 'pct_Pop_2
5_44_ACS_08_12', 'pct_Females_CEN_2010', 'pct_Civ_emp_16p_ACS_08_12', 'pct_Inst_GQ_CEN_2010', 'pct_NonF
amily_HHD_CEN_2010', 'pct_Mobile_Homes_ACS_08_12', 'pct_ENG_VW_INDOEURO_ACS_08_12', 'pct_Age5p_WGerman_
ACS_08_12', 'pct_Rel_Family_HHDS_CEN_2010', 'pct_Age5p_German_ACS_08_12']
```

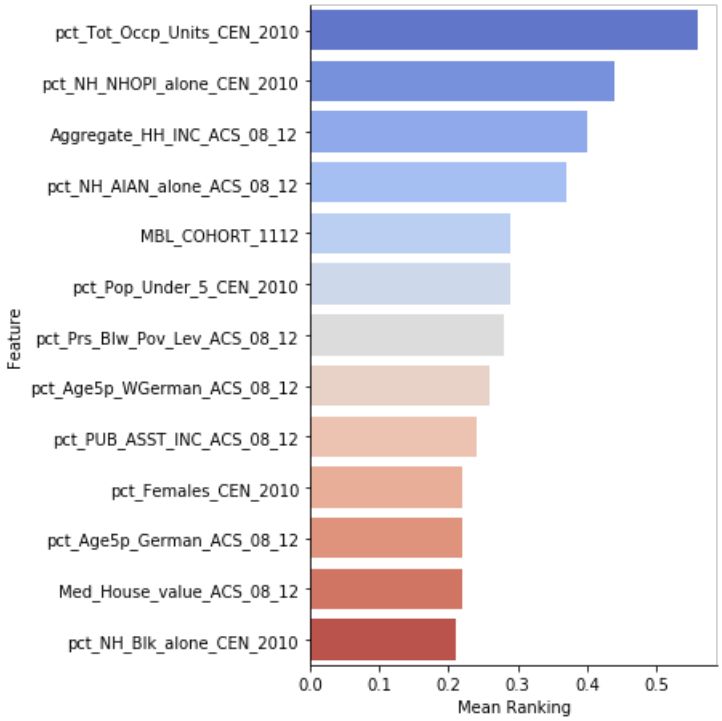
'Statistics: dataset has 9755 (rows) samples with 30 (columns) features each'

In [15]:

```
hs.compare_rgs_featranking(minRanking=0.2)
```

```
Save to a file saved/rgs_ds_allranking_ds.csv
Features   Corr DT Linear RF Ridge Mean
Aggregate_HH_INC_ACS_08_12  1.0 0.54 0.0 0.44 0.0 0.4
pct_NH_AIAN_alone_ACS_08_12  0.46 0.55 0.19 0.46 0.18 0.37
pct_NH_Blk_alone_CEN_2010  0.37 0.22 0.13 0.24 0.11 0.21
pct_Tot_Occp_Units_CEN_2010  0.64 1.0 0.07 1.0 0.08 0.56
pct_RURAL_POP_CEN_2010  0.38 0.0 0.02 0.1 0.02 0.1
MBL_COHORT_1112  0.13 0.68 0.0 0.63 0.0 0.29
pct_Hispanic_CEN_2010  0.03 0.21 0.05 0.25 0.04 0.12
```

```
URBANIZED_AREA_POP_CEN_2010  0.27 0.03 0.0 0.04 0.0 0.07
pct_Pop_Under_5_CEN_2010  0.08 0.09 0.61 0.14 0.51 0.29
pct_HHD_PPL_Und_18_CEN_2010  0.06 0.16 0.17 0.16 0.12 0.13
pct_Pop_45_64_CEN_2010  0.0 0.27 0.17 0.2 0.18 0.16
pct_PUB_ASST_INC_ACS_08_12  0.32 0.25 0.2 0.25 0.2 0.24
pct_Census_UAA_CEN_2010  0.0 0.21 0.04 0.2 0.04 0.1
pct_Rel_Under_6_CEN_2010  0.07 0.19 0.13 0.14 0.11 0.13
pct_MLT_U10p_ACS_08_12  0.03 0.17 0.05 0.19 0.05 0.1
pct_Female_No_HB_ACS_08_12  0.19 0.21 0.08 0.22 0.06 0.15
pct_Prs_Blw_Pov_Lev_ACS_08_12  0.69 0.32 0.04 0.32 0.04 0.28
Med_House_value_ACS_08_12  0.55 0.28 0.0 0.29 0.0 0.22
pct_NO_PH_SRVC_ACS_08_12  0.13 0.15 0.11 0.17 0.1 0.13
pct_NH_NHOPI_alone_CEN_2010  0.0 0.12 1.0 0.09 1.0 0.44
pct_Pop_25_44_ACS_08_12  0.04 0.24 0.1 0.2 0.09 0.13
pct_Females_CEN_2010  0.05 0.28 0.28 0.26 0.21 0.22
pct_Civ_emp_16p_ACS_08_12  0.1 0.33 0.05 0.36 0.06 0.18
pct_Inst_GQ_CEN_2010  0.0 0.06 0.09 0.1 0.07 0.06
pct_NonFamily_HHD_CEN_2010  0.11 0.11 0.13 0.1 0.08 0.11
pct_Mobile_Homes_ACS_08_12  0.34 0.23 0.03 0.23 0.03 0.17
pct_ENG_VW_INDOEURO_ACS_08_12  0.0 0.09 0.23 0.06 0.24 0.12
pct_Age5p_WGerman_ACS_08_12  0.01 0.0 0.66 0.0 0.64 0.26
pct_Rel_Family_HHDS_CEN_2010  0.11 0.13 0.08 0.11 0.02 0.09
pct_Age5p_German_ACS_08_12  0.01 0.05 0.48 0.06 0.48 0.22
Save to a file saved/rgs_ds_meanranking.csv
                          Feature  Mean Ranking
27    pct_Tot_Occp_Units_CEN_2010          0.56
16    pct_NH_NHOPI_alone_CEN_2010          0.44
4      Aggregate_HH_INC_ACS_08_12          0.40
14    pct_NH_AIAN_alone_ACS_08_12          0.37
17               MBL_COHORT_1112          0.29
6        pct_Pop_Under_5_CEN_2010          0.29
3   pct_Prs_Blw_Pov_Lev_ACS_08_12          0.28
25    pct_Age5p_WGerman_ACS_08_12          0.26
9      pct_PUB_ASST_INC_ACS_08_12          0.24
18           pct_Females_CEN_2010          0.22
12     pct_Age5p_German_ACS_08_12          0.22
13      Med_House_value_ACS_08_12          0.22
10      pct_NH_Blk_alone_CEN_2010          0.21
```




```
Selected Features/columns: ['pct_Tot_Occp_Units_CEN_2010', 'pct_NH_NHOPI_alone_CEN_2010', 'Aggregate_HH
_INC_ACS_08_12', 'pct_NH_AIAN_alone_ACS_08_12', 'MBL_COHORT_1112', 'pct_Pop_Under_5_CEN_2010', 'pct_Prs
_Blw_Pov_Lev_ACS_08_12', 'pct_Age5p_WGerman_ACS_08_12', 'pct_PUB_ASST_INC_ACS_08_12', 'pct_Females_CEN_
2010', 'pct_Age5p_German_ACS_08_12', 'Med_House_value_ACS_08_12', 'pct_NH_Blk_alone_CEN_2010']
```

In [16]:

```
hs.rgs_stats()
```

```
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:          ALL_RATE_1112   R-squared:                       0.980
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                 3.647e+04
Date:                Sat, 12 May 2018   Prob (F-statistic):               0.00
Time:                        21:19:10   Log-Likelihood:                -38004.
No. Observations:                9755   AIC:                         7.603e+04
Df Residuals:                    9742   BIC:                         7.613e+04
Df Model:                          13
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
pct_Tot_Occp_Units_CEN_2010       0.3504      0.011     31.446      0.000       0.329       0.372
pct_NH_NHOPI_alone_CEN_2010      -0.5520      0.495     -1.114      0.265      -1.523       0.419
Aggregate_HH_INC_ACS_08_12     1.664e-08   2.43e-09      6.843      0.000    1.19e-08    2.14e-08
pct_NH_AIAN_alone_ACS_08_12      -0.2153      0.017    -12.750      0.000      -0.248      -0.182
MBL_COHORT_1112                  -0.0033      0.000     -8.491      0.000      -0.004      -0.003
pct_Pop_Under_5_CEN_2010         -0.3329      0.081     -4.130      0.000      -0.491      -0.175
pct_Prs_Blw_Pov_Lev_ACS_08_12     0.0165      0.016      1.019      0.308      -0.015       0.048
pct_Age5p_WGerman_ACS_08_12       0.9944      0.261      3.809      0.000       0.483       1.506
pct_PUB_ASST_INC_ACS_08_12       -0.2389      0.054     -4.409      0.000      -0.345      -0.133
pct_Females_CEN_2010              1.0530      0.020     53.583      0.000       1.015       1.092
pct_Age5p_German_ACS_08_12       -0.7203      0.265     -2.719      0.007      -1.240      -0.201
Med_House_value_ACS_08_12      8.88e-06   1.26e-06      7.039      0.000    6.41e-06    1.14e-05
pct_NH_Blk_alone_CEN_2010        -0.1202      0.010    -12.347      0.000      -0.139      -0.101
==============================================================================
Omnibus:                     1646.451   Durbin-Watson:                   1.818
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            27211.480
Skew:                           0.294   Prob(JB):                         0.00
Kurtosis:                      11.161   Cond. No.                     5.18e+08
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.18e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
Save to a file saved/rgs_ols_statssummary.csv
Save to a file saved/rgs_ols_statssummary.txt
```

In [17]:

```
hs.create_rgs_sample()
```

```
rgs Training and testing split was successful.  Split using target variable=['ALL_RATE_1112']
Count of training set is 6828 (69.99%)  testing set is 2927  (30.01%)  in total 9755.

'Statistics: dataset has 6828 (rows) samples with  13 (columns) features each'
```

	pct_Tot_Occp_Units_CEN_2010	pct_NH_NHOPI_alone_CEN_2010	Aggregate_HH_INC_ACS_08_12	pct_NH_AIAN_
2890	89.84	0.03	127415500.0	1.027717
8591	87.39	0.07	16097700.0	0.000000
6	93.67	0.11	160099900.0	0.000000

```
'Statistics: dataset has 6828 (rows) samples with  1 (columns) features each'
```

	ALL_RATE_1112
2890	62.0
8591	98.0
6	93.0

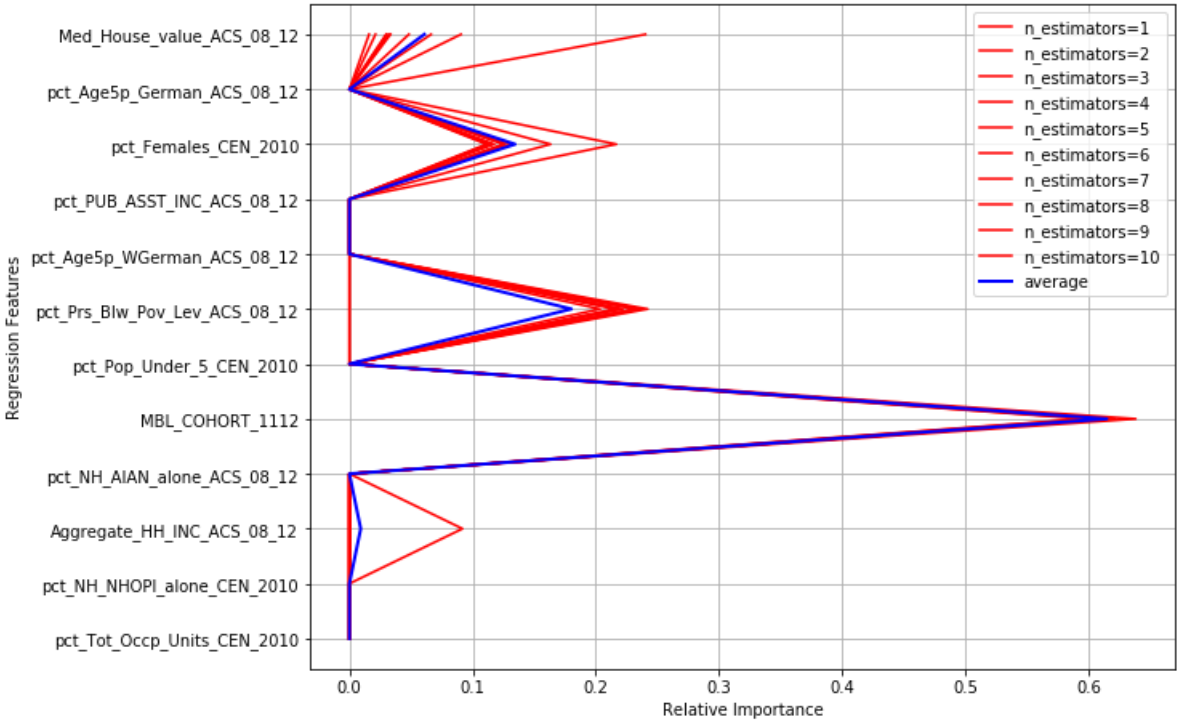
In [18]:

```
hs.rgs_r2_featimportance()
```

```
The R^2 score for features are sorted the highest to lowest
Sorting R2 score, the highest to lowest
# 1 pct_NH_Blk_alone_CEN_2010 R2=  +0.5840
# 2 pct_Females_CEN_2010 R2=  +0.4928
# 3 Aggregate_HH_INC_ACS_08_12 R2=  +0.4643
# 4 Med_House_value_ACS_08_12 R2=  +0.4563
# 5 pct_Prs_Blw_Pov_Lev_ACS_08_12 R2=  +0.4075
```

```
# 6 pct_NH_AIAN_alone_ACS_08_12 R2=  +0.3844
# 7 pct_Tot_Occp_Units_CEN_2010 R2=  +0.3173
# 8 pct_PUB_ASST_INC_ACS_08_12 R2=  +0.2715
# 9 pct_Pop_Under_5_CEN_2010 R2=  +0.1978
# 10 pct_Age5p_German_ACS_08_12 R2=  +0.1282
# 11 pct_Age5p_WGerman_ACS_08_12 R2=  +0.1123
# 12 MBL_COHORT_1112      R2=  +0.1036
# 13 pct_NH_NHOPI_alone_CEN_2010 R2=  +0.0133

--> pct_NH_Blk_alone_CEN_2010 has the highest R2 score 0.5840
Save to a file saved/rgs_r2.txt
Feature importance -->
```

```
[('Aggregate_HH_INC_ACS_08_12', 0.009182427923730713),
 ('MBL_COHORT_1112', 0.6151117879331575),
 ('Med_House_value_ACS_08_12', 0.06085671437268603),
 ('pct_Age5p_German_ACS_08_12', 0.0),
 ('pct_Age5p_WGerman_ACS_08_12', 0.0),
 ('pct_Females_CEN_2010', 0.13454662078133844),
 ('pct_NH_AIAN_alone_ACS_08_12', 0.0),
 ('pct_NH_NHOPI_alone_CEN_2010', 0.0),
 ('pct_PUB_ASST_INC_ACS_08_12', 0.0),
 ('pct_Pop_Under_5_CEN_2010', 0.0),
 ('pct_Prs_Blw_Pov_Lev_ACS_08_12', 0.18030244898908734),
 ('pct_Tot_Occp_Units_CEN_2010', 0.0)]
```




```
Sorting feature importance, the hight to lowest
```

```
[(0.6151117879331575, 'MBL_COHORT_1112'),
 (0.18030244898908734, 'pct_Prs_Blw_Pov_Lev_ACS_08_12'),
 (0.13454662078133844, 'pct_Females_CEN_2010'),
 (0.06085671437268603, 'Med_House_value_ACS_08_12'),
 (0.009182427923730713, 'Aggregate_HH_INC_ACS_08_12'),
 (0.0, 'pct_Tot_Occp_Units_CEN_2010'),
 (0.0, 'pct_Pop_Under_5_CEN_2010'),
 (0.0, 'pct_PUB_ASST_INC_ACS_08_12'),
 (0.0, 'pct_NH_NHOPI_alone_CEN_2010'),
 (0.0, 'pct_NH_AIAN_alone_ACS_08_12'),
 (0.0, 'pct_Age5p_WGerman_ACS_08_12'),
 (0.0, 'pct_Age5p_German_ACS_08_12')]
```

```
Save to a file saved/rgs_featimportance.txt
Visualize Features in Correlation Matrix->

Data points considered outliers for the feature --> 'pct_Tot_Occp_Units_CEN_2010'
   Q1=83.5900 Q3= 93.335000 step= 1.5*(Q3-Q1) = 14.6175   Feeature Outlier cnt= 735

Data points considered outliers for the feature --> 'pct_NH_NHOPI_alone_CEN_2010'
   Q1=0.0000 Q3= 0.040000 step= 1.5*(Q3-Q1) = 0.0600   Feeature Outlier cnt= 1082

Data points considered outliers for the feature --> 'Aggregate_HH_INC_ACS_08_12'
```

```
   Q1=60407100.0000 Q3= 130396800.000000 step= 1.5*(Q3-Q1) = 104984550.0000   Feeature Outlier cnt= 488

Data points considered outliers for the feature --> 'pct_NH_AIAN_alone_ACS_08_12'
   Q1=0.0000 Q3= 0.512289 step= 1.5*(Q3-Q1) = 0.7684   Feeature Outlier cnt= 1079

Data points considered outliers for the feature --> 'MBL_COHORT_1112'
   Q1=0.0000 Q3= 18.000000 step= 1.5*(Q3-Q1) = 27.0000   Feeature Outlier cnt= 1616

Data points considered outliers for the feature --> 'pct_Pop_Under_5_CEN_2010'
   Q1=5.1400 Q3= 7.010000 step= 1.5*(Q3-Q1) = 2.8050   Feeature Outlier cnt= 432

Data points considered outliers for the feature --> 'pct_Prs_Blw_Pov_Lev_ACS_08_12'
   Q1=7.3996 Q3= 18.544776 step= 1.5*(Q3-Q1) = 16.7178   Feeature Outlier cnt= 389

Data points considered outliers for the feature --> 'pct_Age5p_WGerman_ACS_08_12'
   Q1=0.0000 Q3= 0.000000 step= 1.5*(Q3-Q1) = 0.0000   Feeature Outlier cnt= 537

Data points considered outliers for the feature --> 'pct_PUB_ASST_INC_ACS_08_12'
   Q1=0.8638 Q3= 3.353312 step= 1.5*(Q3-Q1) = 3.7343   Feeature Outlier cnt= 528

Data points considered outliers for the feature --> 'pct_Females_CEN_2010'
   Q1=49.4700 Q3= 51.750000 step= 1.5*(Q3-Q1) = 3.4200   Feeature Outlier cnt= 511

Data points considered outliers for the feature --> 'pct_Age5p_German_ACS_08_12'
   Q1=0.0000 Q3= 0.000000 step= 1.5*(Q3-Q1) = 0.0000   Feeature Outlier cnt= 1970

Data points considered outliers for the feature --> 'Med_House_value_ACS_08_12'
   Q1=85900.0000 Q3= 180850.000000 step= 1.5*(Q3-Q1) = 142425.0000   Feeature Outlier cnt= 826

Data points considered outliers for the feature --> 'pct_NH_Blk_alone_CEN_2010'
   Q1=0.3000 Q3= 3.980000 step= 1.5*(Q3-Q1) = 5.5200   Feeature Outlier cnt= 1535


data size=9755 max_idx=9906  Outliers for all features =11728
Note: Also found duplicate outliers in multiple features =3350

Removed duplicated outlier data -> good datasize=(6444, 13)   target datasize=(6444, 1)


'Before log-transformed, log_data Mean=Average & Median=50% -> '
```

	pct_Tot_Occp_Units_CEN_2010	pct_NH_NHOPI_alone_CEN_2010	Aggregate_HH_INC_ACS_08_12	pct_NH_AIAN_
mean	86.47	0.04	1.055161e+08	1.13
50%	89.78	0.00	8.914955e+07	0.05

	pct_Tot_Occp_Units_CEN_2010	pct_NH_NHOPI_alone_CEN_2010	Aggregate_HH_IN
pct_Tot_Occp_Units_CEN_2010	1.00	0.03	0.32
pct_NH_NHOPI_alone_CEN_2010	0.03	1.00	0.00
Aggregate_HH_INC_ACS_08_12	0.32	0.00	1.00
pct_NH_AIAN_alone_ACS_08_12	-0.07	0.00	-0.10
MBL_COHORT_1112	0.03	0.04	0.00
pct_Pop_Under_5_CEN_2010	0.24	0.11	-0.06
pct_Prs_Blw_Pov_Lev_ACS_08_12	-0.15	0.02	-0.44
pct_Age5p_WGerman_ACS_08_12	0.01	-0.01	-0.02
pct_PUB_ASST_INC_ACS_08_12	-0.04	0.04	-0.22
pct_Females_CEN_2010	0.35	-0.03	0.14
pct_Age5p_German_ACS_08_12	-0.01	-0.01	-0.03
Med_House_value_ACS_08_12	0.15	0.05	0.63
pct_NH_Blk_alone_CEN_2010	-0.01	0.01	-0.15

```
'After log-transformed, log_data Mean=Average & Median=50% -> '
```

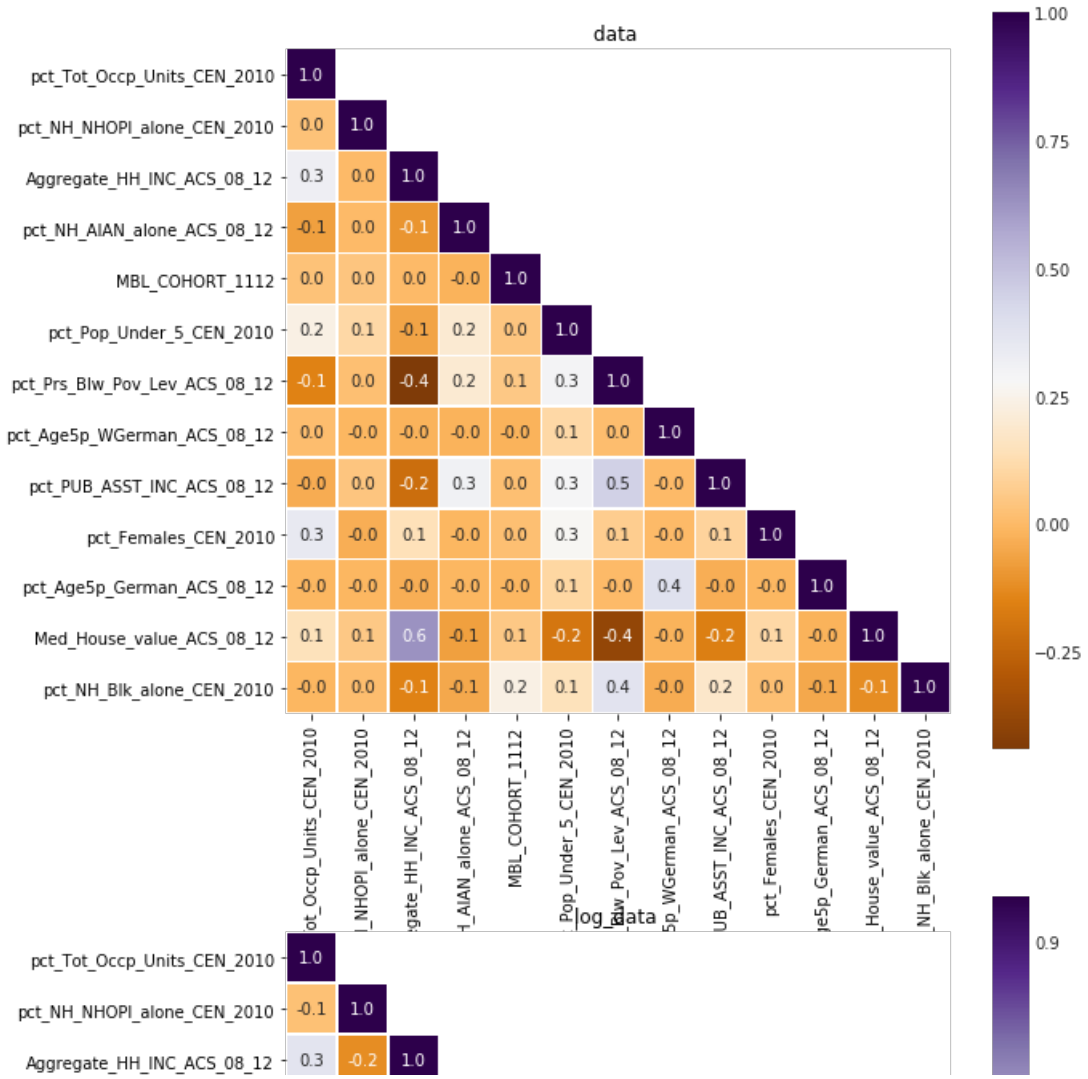
	pct_Tot_Occp_Units_CEN_2010	pct_NH_NHOPI_alone_CEN_2010	Aggregate_HH_INC_ACS_08_12	pct_NH_AIAN_
mean	-inf	-inf	-inf	-inf
50%	4.500000	-inf	18.310000	-3.090000

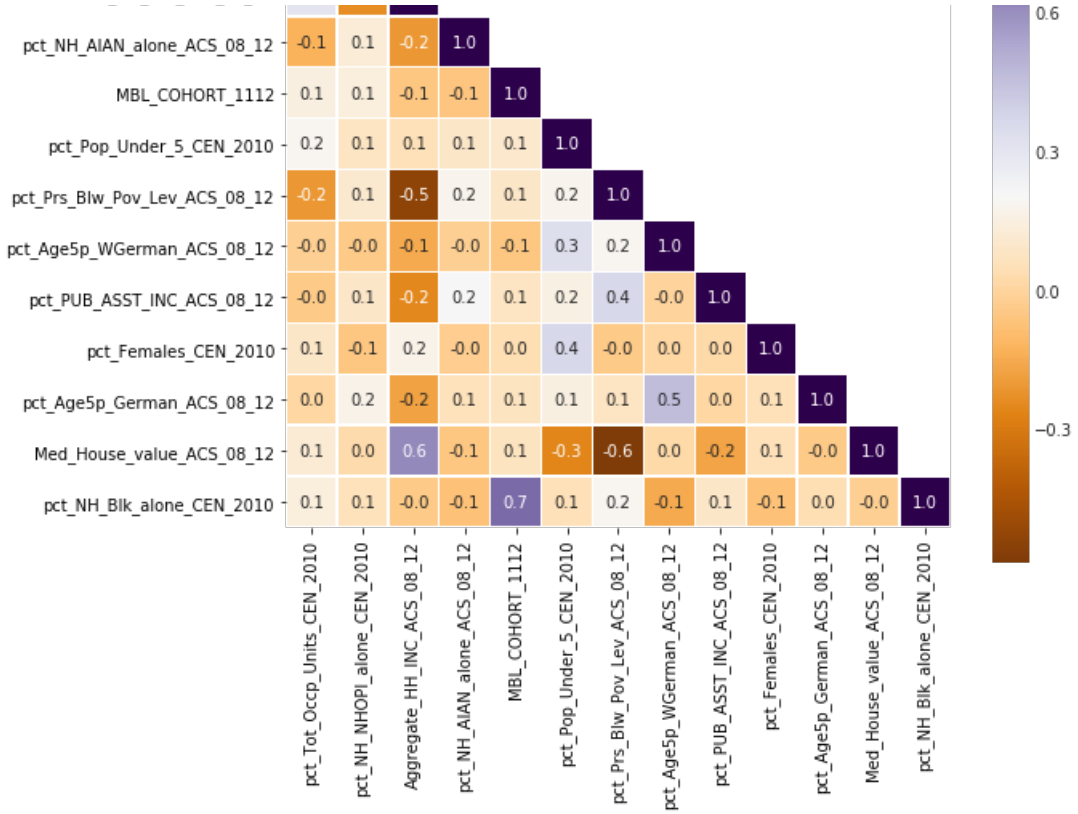
'Statistics: dataset has 6444 (rows) samples with 13 (columns) features each'

log_data in Correlation Matrix ->

	pct_Tot_Occp_Units_CEN_2010	pct_NH_NHOPI_alone_CEN_2010	Aggregate_HH_IN
pct_Tot_Occp_Units_CEN_2010	1.00	-0.08	0.31
pct_NH_NHOPI_alone_CEN_2010	-0.08	1.00	-0.24
Aggregate_HH_INC_ACS_08_12	0.31	-0.24	1.00
pct_NH_AIAN_alone_ACS_08_12	-0.13	0.13	-0.21
MBL_COHORT_1112	0.15	0.14	-0.05
pct_Pop_Under_5_CEN_2010	0.19	0.08	0.05
pct_Prs_Blw_Pov_Lev_ACS_08_12	-0.21	0.11	-0.54
pct_Age5p_WGerman_ACS_08_12	-0.02	-0.04	-0.15
pct_PUB_ASST_INC_ACS_08_12	-0.04	0.09	-0.25
pct_Females_CEN_2010	0.09	-0.05	0.17
pct_Age5p_German_ACS_08_12	0.01	0.17	-0.18
Med_House_value_ACS_08_12	0.12	0.03	0.62
pct_NH_Blk_alone_CEN_2010	0.14	0.06	-0.04

Visualize comparing data and log_data in Correlation Matrix

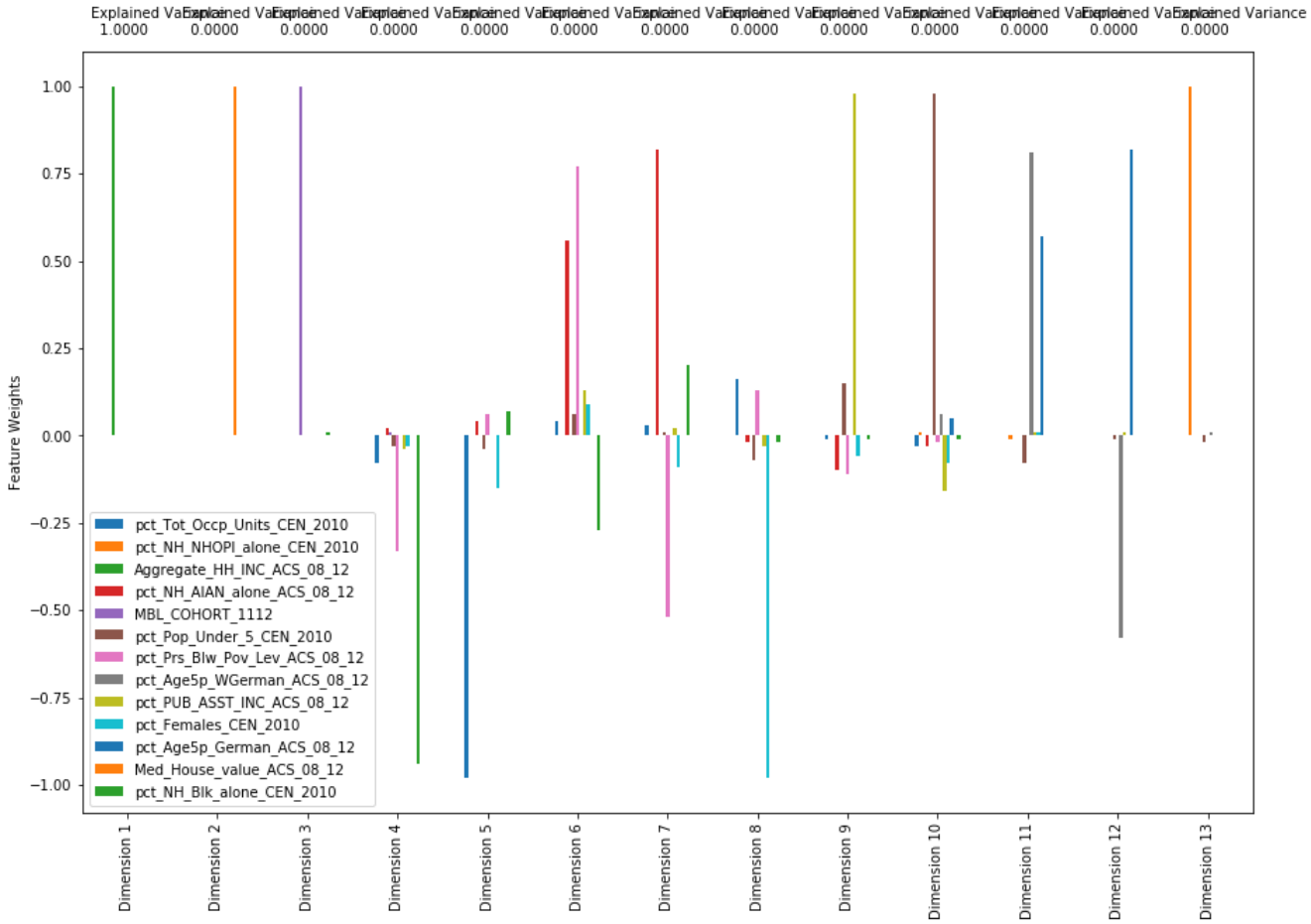




Extracting the top 13 features from 6444 data points

```
 PCA Explained variance ratio=[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[[ 5.47207107e-08  9.67892433e-13  9.99999393e-01 -1.05701858e-08
   8.71103904e-11 -1.58046241e-09 -6.03838971e-08 -1.88367248e-10
  -8.11034743e-09  8.87658649e-09 -2.37793157e-10  1.10197745e-03
  -2.88460109e-08]
 [-8.41541146e-06  1.25846715e-07 -1.10197742e-03 -5.97947984e-08
   2.12688860e-04 -3.63439696e-06 -1.33976759e-05 -9.15831793e-09
  -5.57509713e-07  1.15916625e-06 -2.94516054e-08  9.99999370e-01
  -4.26075657e-06]
 [ 1.30887892e-03  1.99216218e-05  2.34506673e-07 -3.80244644e-04
   9.99948775e-01  2.42896910e-04  1.70696138e-03 -2.75466682e-05
   1.93571103e-04  2.56429028e-04 -1.90472878e-05 -2.12601224e-04
   9.87255327e-03]
 [-7.72986727e-02 -1.49218417e-04 -3.01461320e-08  1.50174592e-02
   9.96771696e-03 -2.51995138e-02 -3.26460419e-01  1.56570638e-03
  -4.18429143e-02 -3.42475034e-02  1.95649735e-03 -1.12236166e-05
  -9.39977974e-01]
 [-9.83318985e-01 -5.12280622e-04  6.91487883e-08  3.98745410e-02
   5.89358086e-04 -3.81011524e-02  6.28334444e-02 -1.21124489e-03
  -1.39213343e-03 -1.47364453e-01 -3.35641497e-04 -7.24271419e-06
   6.61335476e-02]
 [ 3.64225713e-02  6.21221811e-04  3.23428102e-08  5.56052071e-01
   1.46689846e-03  6.33147266e-02  7.65282151e-01  8.24474132e-04
   1.32226948e-01  9.45243569e-02  5.29504313e-06  9.32072030e-06
  -2.70909447e-01]
 [ 2.60019105e-02 -4.04592216e-04 -1.15723436e-08  8.23723900e-01
  -7.35785318e-04  9.50821094e-03 -5.23794019e-01 -2.85044841e-03
   1.91353590e-02 -8.90271996e-02 -1.77100272e-03 -5.61357768e-06
   1.95059696e-01]
 [ 1.55347898e-01  2.52215222e-03  2.75183764e-09 -1.99603184e-02
   3.56048331e-05 -7.39517660e-02  1.32103283e-01  1.30389378e-03
  -3.19048954e-02 -9.75254342e-01  5.27895772e-05  3.82668321e-06
  -2.00363637e-02]
 [-1.00363923e-02  4.53175618e-03  1.92480020e-09 -9.62096429e-02
   3.67262995e-05  1.52598935e-01 -1.06972432e-01  2.76856377e-03
   9.75952454e-01 -5.74183591e-02 -4.90098164e-03 -4.05012367e-07
  -9.00872399e-03]
 [-2.94611376e-02  1.47109055e-02 -2.63999621e-09 -2.81399640e-02
  -5.87384431e-05  9.78935547e-01 -2.36859615e-02  6.15362975e-02
  -1.63229125e-01 -7.59738881e-02  5.48802761e-02  2.97608078e-06
  -5.79653622e-03]
 [ 1.33135451e-03 -1.09339061e-02  5.80314640e-10  4.67381025e-03
   1.17461645e-05 -8.01429008e-02  3.26025835e-04  8.14745599e-01
   1.40963582e-02  6.79301295e-03  5.73903357e-01 -2.36517573e-07
```

```
   3.67898980e-03]
 [ 8.12841614e-04  1.89678262e-03  1.12345358e-10 -2.10976515e-04
  -5.91726289e-06 -8.39944716e-03  3.80001412e-04 -5.76453252e-01
   7.06799168e-03 -1.39933954e-04  8.17053528e-01  7.23125514e-09
   4.53540613e-04]
 [-4.27393971e-04  9.99816337e-01  1.68528190e-10  9.62501928e-04
  -1.85962067e-05 -1.58280711e-02 -2.01055136e-04  9.08028461e-03
  -1.88204617e-03  3.73746889e-03  3.94009505e-03 -1.89289238e-07
   3.56698095e-04]]
```



In [19]:

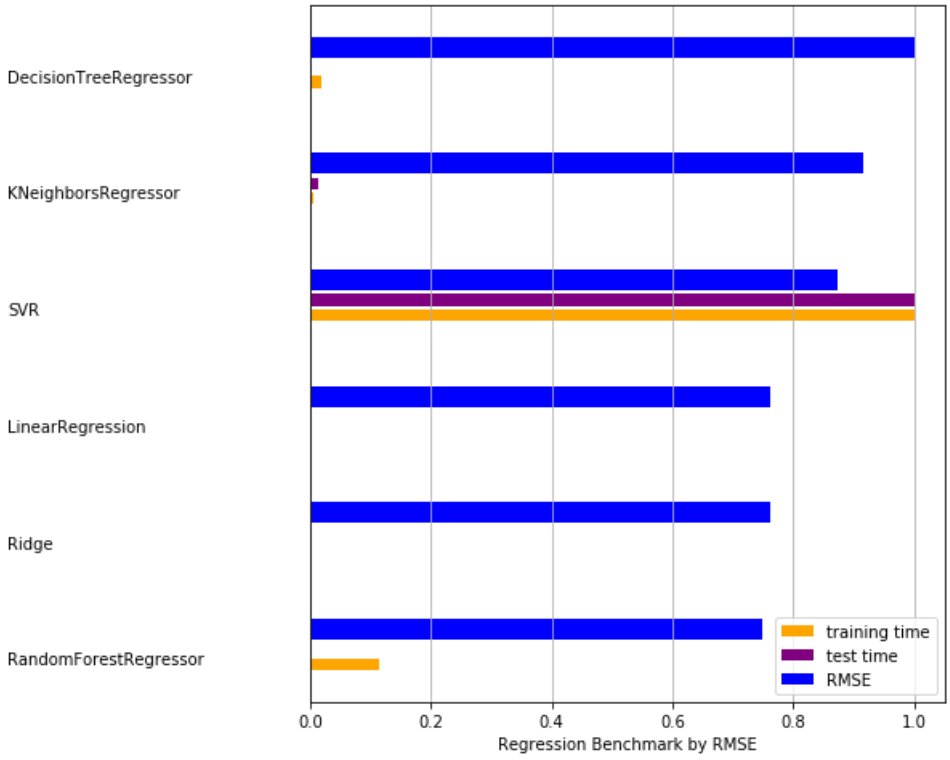
```
hs.rgs_visual_benchmark( )
```

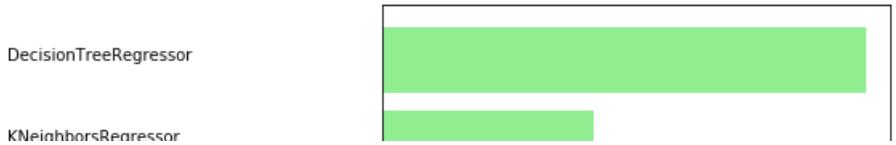
```
================================================================================
Linear Regression(Linear)
________________________________________________________________________________
Training:
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
fit -> rgs.score:    0.204
--- train time:    0.000s
--- test time:     0.000s
dimensionality: 1
coef_: [[ 1.85562663e-01 -1.07052482e+00  2.43683013e-08 -2.58847533e-01
  -3.13191373e-03 -3.05580911e-01 -2.83920776e-02  8.43017899e-01
  -1.69321124e-01  1.01946826e-02 -9.84110660e-01  7.49639646e-06
  -1.04775291e-01]]
density:        1.000
dimensionality/count(non zero of coef_): 13
density:        1.000
coef_---->
Target feature: ALL_RATE_1112
Top 13 Features
[(0.8430178990103359, 'pct_Age5p_WGerman_ACS_08_12'),
 (0.18556266329818472, 'pct_Tot_Occp_Units_CEN_2010'),
 (0.01019468260459322, 'pct_Females_CEN_2010'),
 (7.4963964615815115e-06, 'Med_House_value_ACS_08_12'),
 (2.436830133903026e-08, 'Aggregate_HH_INC_ACS_08_12'),
 (-0.003131913730710853, 'MBL_COHORT_1112'),
 (-0.028392077586106013, 'pct_Prs_Blw_Pov_Lev_ACS_08_12'),
 (-0.10477529092125848, 'pct_NH_Blk_alone_CEN_2010'),
```

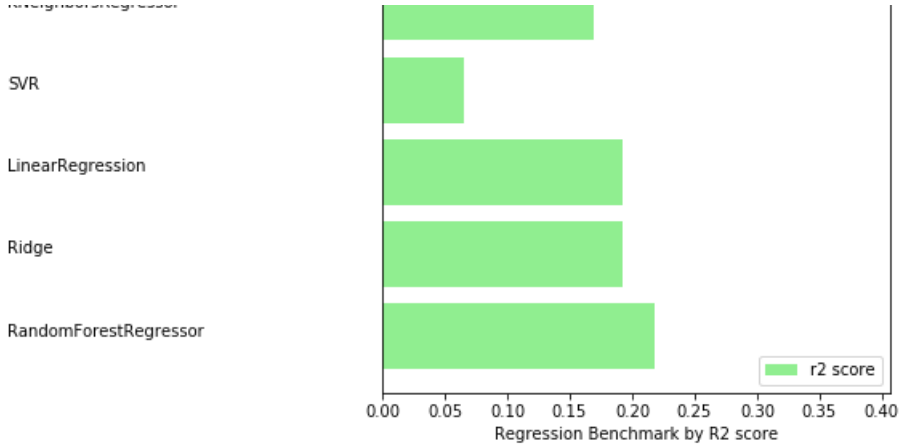
```
 (-0.16932112350110207, 'pct_PUB_ASST_INC_ACS_08_12'),
 (-0.25884753294359286, 'pct_NH_AIAN_alone_ACS_08_12'),
 (-0.3055809108138205, 'pct_Pop_Under_5_CEN_2010'),
 (-0.9841106604952825, 'pct_Age5p_German_ACS_08_12'),
 (-1.07052481683553, 'pct_NH_NHOPI_alone_CEN_2010')]
cross_val_score ------>
ExplainedVariance: mean=       0.193 std=   0.012
MeanAbsError/MAE: mean=       7.717 std=   0.197
MeanSqrErr/MSE: mean=     114.707 std=   6.394
Accuracy / RMSE=       10.710
MedianAbsErr/Median SE: mean=       5.960 std=   0.166
R^2: mean=       0.193 std=   0.012
Scores ------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.193    7.717  114.707   10.710    5.960    5.960
metric------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.204    7.598  109.450   10.462    5.984    5.984

================================================================================
Random Forest(RF)
--------------------------------------------------------------------------------
Training:
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=30,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
           oob_score=False, random_state=99, verbose=0, warm_start=False)
fit -> rgs.score:    0.194
--- train time:    0.531s
--- test time:     0.000s
cross_val_score ------>
ExplainedVariance: mean=       0.221 std=   0.036
MeanAbsError/MAE: mean=       7.673 std=   0.263
MeanSqrErr/MSE: mean=     111.035 std=   8.349
Accuracy / RMSE=       10.537
MedianAbsErr/Median SE: mean=       5.668 std=   0.144
R^2: mean=       0.219 std=   0.037
Scores ------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.221    7.673  111.035   10.537    5.668    5.668
metric------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.198    7.725  110.813   10.527    5.750    5.750

================================================================================
k-Nearest Neighbors(KNN)
--------------------------------------------------------------------------------
Training:
KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='minkowski',
          metric_params=None, n_jobs=1, n_neighbors=3, p=2,
          weights='uniform')
fit -> rgs.score:   -0.133
--- train time:    0.031s
--- test time:     0.016s
cross_val_score ------>
ExplainedVariance: mean=       0.169 std=   0.023
MeanAbsError/MAE: mean=       9.529 std=   0.157
MeanSqrErr/MSE: mean=     166.070 std=   7.033
Accuracy / RMSE=       12.887
MedianAbsErr/Median SE: mean=       7.067 std=   0.170
R^2: mean=       0.170 std=   0.023
Scores ------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.169    9.529  166.070   12.887    7.067    7.067
metric------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
  -0.133    9.270  155.805   12.482    7.000    7.000

================================================================================
Decision Tree(DT)
--------------------------------------------------------------------------------
Training:
DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=1,
```

```
           min_impurity_split=None, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           presort=False, random_state=None, splitter='best')
fit -> rgs.score:   -0.472
--- train time:    0.094s
--- test time:     0.000s
cross_val_score ------>
ExplainedVariance: mean=       0.367 std=   0.089
MeanAbsError/MAE: mean=       9.745 std=   0.421
MeanSqrErr/MSE: mean=     198.050 std=  12.612
Accuracy / RMSE=       14.073
MedianAbsErr/Median SE: mean=       6.800 std=   0.400
R^2: mean=       0.387 std=   0.089
Scores ------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.367    9.745  198.050   14.073    6.800    6.800
metric------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
  -0.466   10.016  202.352   14.225    7.000    7.000

================================================================================
Ridge
________________________________________________________________________________
Training:
Ridge(alpha=0.05, copy_X=True, fit_intercept=True, max_iter=None,
   normalize=True, random_state=None, solver='auto', tol=0.001)
fit -> rgs.score:    0.204
--- train time:    0.000s
--- test time:     0.000s
dimensionality: 1
coef_: [[ 1.76503211e-01 -9.99832948e-01  2.35070460e-08 -2.45776598e-01
  -3.02419378e-03 -2.88680630e-01 -3.72317870e-02  7.89631146e-01
  -1.74527547e-01  1.85340143e-02 -9.21635818e-01  7.41414353e-06
  -9.87846013e-02]]
density:        1.000
dimensionality/count(non zero of coef_): 13
density:        1.000
coef_---->
Target feature: ALL_RATE_1112
Top 13 Features

[(0.7896311462077994, 'pct_Age5p_WGerman_ACS_08_12'),
 (0.17650321137998895, 'pct_Tot_Occp_Units_CEN_2010'),
 (0.01853401428830525, 'pct_Females_CEN_2010'),
 (7.414143528883721e-06, 'Med_House_value_ACS_08_12'),
 (2.3507046007522415e-08, 'Aggregate_HH_INC_ACS_08_12'),
 (-0.003024193779588867, 'MBL_COHORT_1112'),
 (-0.0372317869830558, 'pct_Prs_Blw_Pov_Lev_ACS_08_12'),
 (-0.0987846013402592, 'pct_NH_Blk_alone_CEN_2010'),
 (-0.17452754662748285, 'pct_PUB_ASST_INC_ACS_08_12'),
 (-0.2457765976278551, 'pct_NH_AIAN_alone_ACS_08_12'),
 (-0.28868062985684295, 'pct_Pop_Under_5_CEN_2010'),
 (-0.9216358182032648, 'pct_Age5p_German_ACS_08_12'),
 (-0.9998329476371323, 'pct_NH_NHOPI_alone_CEN_2010')]

cross_val_score ------>
ExplainedVariance: mean=       0.194 std=   0.011
MeanAbsError/MAE: mean=       7.720 std=   0.197
MeanSqrErr/MSE: mean=     114.660 std=   6.432
Accuracy / RMSE=       10.708
MedianAbsErr/Median SE: mean=       5.973 std=   0.163
R^2: mean=       0.193 std=   0.012
Scores ------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.194    7.720  114.660   10.708    5.973    5.973
metric------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.204    7.602  109.428   10.461    5.974    5.974

================================================================================
Support Vector Regression(SVR)
________________________________________________________________________________
Training:
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='auto',
  kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
fit -> rgs.score:   -0.057
--- train time:    4.687s
--- test time:     1.094s
```

```
support_vectors_: mean= 8073191.655  std=33684049.654
cross_val_score ------>
ExplainedVariance: mean=       0.007 std=   0.001
MeanAbsError/MAE: mean=       8.502 std=   0.191
MeanSqrErr/MSE: mean=     151.306 std=   8.361
Accuracy / RMSE=       12.301
MedianAbsErr/Median SE: mean=       5.813 std=   0.068
R^2: mean=       0.065 std=   0.013
Scores ------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.007    8.502  151.306   12.301    5.813    5.813
metric------>
varianceScore meanAbsErr MeanSqrErr RMSE  medianAbsErr r2Score
   0.006    8.386  145.312   12.055    5.813    5.813

Save to a file saved/rgs_visual_benchmark.txt
Save to a file saved/rgs_visual_benchmark.csv
Benchmark Summary for Regression (sorted by RMSE Asc (low to high))
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               rgs_names  train_time  test_time  explained_variance_score  \
1  RandomForestRegressor    0.113333   0.000000                  0.221048
4                  Ridge    0.000000   0.000000                  0.193509
0       LinearRegression    0.000000   0.000000                  0.193160
5                    SVR    1.000000   1.000000                  0.006626
2    KNeighborsRegressor    0.006667   0.014284                  0.168720
3  DecisionTreeRegressor    0.019999   0.000000                  0.367086

   mean_absolute_error  mean_squared_error  root_mean_squared_error  \
1             7.673474          111.035467                10.537337
4             7.719673          114.659658                10.707925
0             7.717334          114.707197                10.710145
5             8.502477          151.306230                12.300660
2             9.528559          166.070441                12.886832
3             9.744881          198.049744                14.073015

   median_absolute_error  r2_score
1               5.667667  0.218501
4               5.973336  0.192867
0               5.959865  0.192513
5               5.813294  0.065085
2               7.066667  0.169889
3               6.800000  0.387051
```







In [20]:

```
exec_time = time() - t1
print("--- Regression exec_time:  {0:8.3f}s".format(exec_time))

complete_time = time() - t0
print("--- complete_time:  {0:8.3f}s".format(complete_time))
```

```
--- Regression exec_time:   258.430s
--- complete_time:   325.372s
```