BRFSS Report

By

Nisha Muthukumaran

Behavioral Risk Factor Surveillance System Data is collected by telephonic means and represents the population health. It contains health related risk-behaviors data. Unlike the claims data that represented individual health cases, the BRFSS data gives a broad understanding of the population data. Every person in the United States is not interviewed for this, rather a smaller population that represents the country's population are interviewed. The data contains majorly categorical data with few columns continuous columns.

We are performing the analysis for New England region including the states of Massachusetts (8415 responses), New Hampshire (6420 responses), Vermont (6540 responses), Rhode Island (5457 responses), Maine (10019) and Connecticut (11041 responses) based on _LLCPWT. Ex: The 6420 responses from NH are interpreted the state's population health (which is 1 Million)

Question 1:

The objective is to understand and explore the idea of comorbidity, which is defined as the simultaneous existence of two or more health conditions. The health conditions considered for comorbidity are:

1) **Diabetes (DIABETE3)**
2) **Asthma (ASTHMA3)**
3) **COPD (CHCCOPD1)**
4) **Cancer (combine CHCSCNCR and CHCOCNCR)**
5) **heart disease / heart attack/ stroke (CVDCRHD4, CVDINFR4 , CVDSTRK3)**
6) **Depression (ADDEPEV2)**
7) **Arthritis (HAVARTH3)**
8) **Kidney Disease (CHCKIDNY)**

The idea is to construct comorbidity severity scores in two different ways and analyzing the differences.

i. **Sum all eight health conditions together and create a score from 0 to 8 indicating the presence of these conditions.**

The data given to us contains values for questions related to chronic conditions like, "Have you been told that you have diabetes?", "Do you have asthma?". If the person being interviewed wishes to be honest, they respond with a Yes or a No which are categorized into value responses. Our aim is to create eight new columns from the **DIABETE3, ASTHMA3, CHCCOPD1, CHCSCNCR, CHCOCNCR, CVDCRHD4, CVDINFR4, CVDSTRK3, ADDEPEV2, HAVARTH3** and **CHCKIDNY** columns.

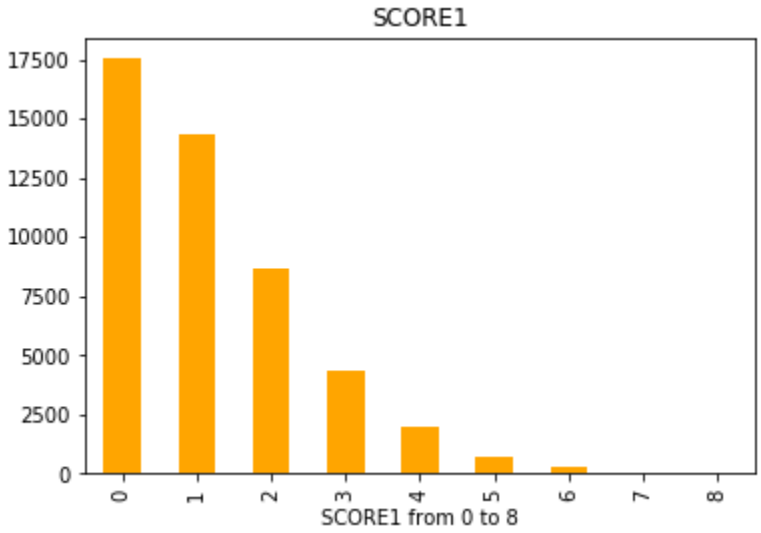
```
table['DIABETES'] = np.where(table['DIABETE3'] == 1, 1, 0)
```

Column **DIABETES** is created which contains the value 1 if the person responded to the question as they had the condition and 0 for any other response that was recorded for that in the **DIABETE3** column. This method is repeated for all the other 8 chronic conditions.

```
table['ASTHMA'] = np.where(table['ASTHMA3'] == 1, 1, 0)
table['COPD'] = np.where(table['CHCCOPD1'] == 1, 1, 0)
table['CANCER'] = np.where((table['CHCSCNCR'] == 1) | (table['CHCOCNCR'] == 1) , 1, 0)
```

```
table['HEART'] = np.where((table['CVDCRHD4'] == 1) | (table['CVDINFR4'] == 1) |
(table['CVDSTRK3'] == 1), 1, 0)
table['DEPRESSION'] = np.where(table['ADDEPEV2'] == 1, 1, 0)
table['ARTHRITIS'] = np.where(table['HAVARTH3'] == 1, 1, 0)
table['KIDNEY'] = np.where(table['CHCKIDNY'] == 1, 1, 0)
```

Finally, a **SCORE1** column was created with a NAÏVE method that sums all the corresponding 1's from the newly created columns to give a whole number score from 0 through 8. The distribution of this column is plotted, and we see the below graph which is skewed to the left.



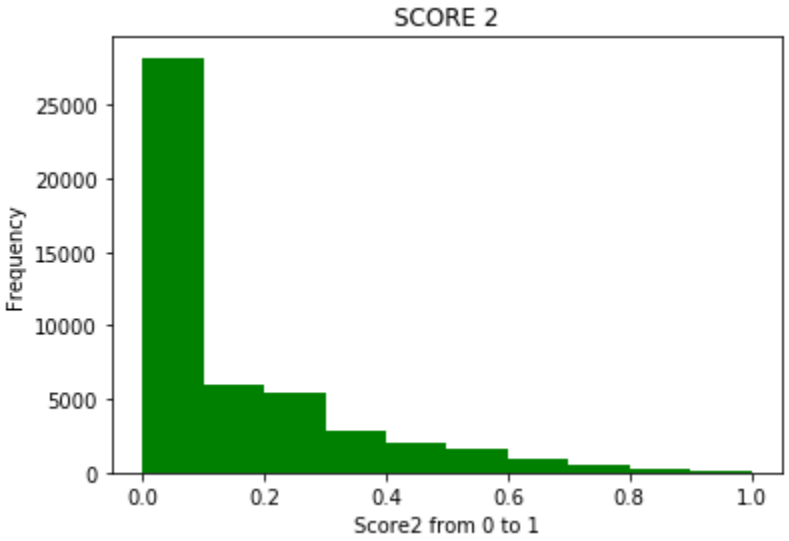
The graph depicts is that large part of the New England population is healthy because a score of 0 shows that the person did not have any of the eight chronic conditions. A score of 1 depicts that the population has one of the 8 conditions (but we cannot tell which one with this grouping), and with more conditions the score increases. The maximum of 8 shows that the population frequency that contains all the 8 chronic conditions. From the analysis we can tell that a very small frequency has 7 out of the 8 conditions and only 15 people have all 8 conditions. Since it is a smaller number on the frequency scale, it doesn't appear.

ii. The above score assumes equality of health conditions together. The score tries to create differentiation based on the severity of health conditions. Create a weighted comorbidity score based on the following weights:

Health Condition	Base Weight	Proportional Weight
Diabetes	245	245 / 1113 = 0.220
Asthma	56	56 / 1113 = 0.050
COPD	32	32 / 1113 = 0.028
Cancer	171	171 / 1113 = 0.154
Heart Disease /Heart Attack / Stroke	386	386 / 1113 = 0.347
Depression	110	110 / 1113 = 0.099
Arthritis	80	80 / 1113 = 0.072
Kidney Disease	33	33 / 1113 = 0.03

For those who had a score value of 1 in the newly created columns in i) , we created other new columns that assigned weights wherever it had a corresponding **SCORE1** value of 1. Ex. If a person said he had ASTHMA, he got a score of 1 in the newly created **ASTHMA** table in i) and a score 0.050 in the now created **ASTHMA_WT** column. This table is a representation of how many millions are spent on each chronic condition. 386 Million is spent on Heart conditions alone. The weights are added together to make it the final number into a proportion of the whole. So, if a person had a high score in Score2 it would mean that he has more of the chronic conditions, but we cannot see how much individually because a score of 0.347 could mean he has a heart condition alone, or that he has a combination of kidney diseases, arthritis, depression and asthma.

```
table['DIABETES_WT'] = np.where(table['DIABETES'] == 1, 0.220, 0)
table['ASTHMA_WT'] = np.where(table['ASTHMA'] == 1, 0.050, 0)
table['COPD_WT'] = np.where(table['COPD'] == 1, 0.028, 0)
table['CANCER_WT'] = np.where(table['CANCER'] == 1 , 0.154, 0)
table['HEART_WT'] = np.where(table['HEART'] == 1, 0.347, 0)
table['DEPRESSION_WT'] = np.where(table['DEPRESSION'] == 1, 0.099, 0)
table['ARTHRITIS_WT'] = np.where(table['HAVARTH3'] == 1, 0.072, 0)
table['KIDNEY_WT'] = np.where(table['CHCKIDNY'] == 1, 0.03, 0)
```



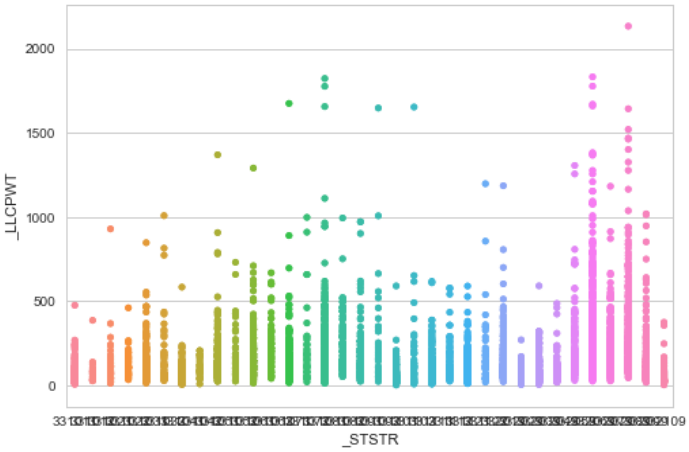
When **SCORE2** was plotted we see the below graph skewed to the left.

This again reiterates that the new England populations' majority is on the healthier side. As the graph shows the score 0 has the highest frequency and with increase in the presence of the condition/conditions the frequency decrease. A very small population has all the conditions.

iii) Analyze the weight variable _LLCPWT on basic descriptive statistics (for NH only), by various categories like _STSTR, Sex, Age, and Income.

_LLCPWT is the weightage an entry is given for Landline cellphone weight.

A subset of the New Hampshire responses is created. And the variables are plotted below

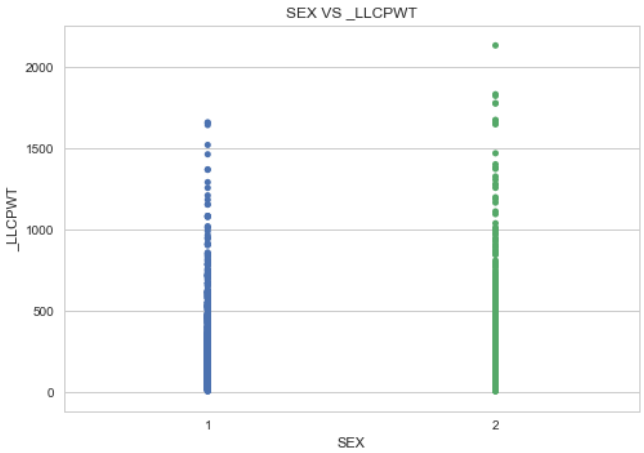


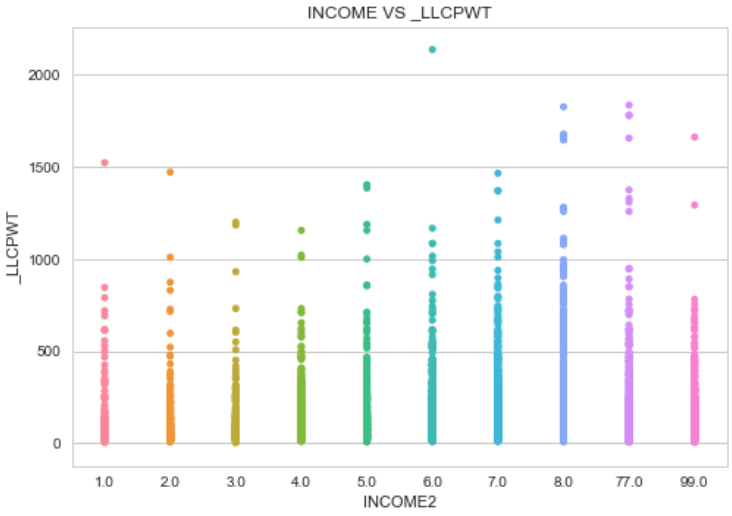
_LLCPWT vs **_STSTR**:

New Hampshire is State 33 and that is why the values starting with 33 appear here. The x axis actually represents categorical variables but since there are too many markings, the numbers are overlapping over each other. **_STSTR(**stratification**)** represents the strata for cellphones in NH and landlines in NH. The numbers that are 3310 are cell phones and if it ends with odd last 2 digits it is for high density and even last 2 digits is for low density. So ideally high density should be in the southern part of NH counties as it has a higher population than the North.

_LLCPWT vs **SEX**:

Here 2 represents females and 1 represents males. This shows that more women have taken the survey than men in the state of NH. This is a representation of the whole population of NH as it contains men and women (1 Million people). Higher weights were given to women which show that more women are represented by the ones who took the survey.





INCOME2 vs **_LLCPWT**:

This shows the income levels. Until 8 the levels are increasing showing more weights were given to people who earned an income of 75K or more. The level for 77 shows that many people didn't know or weren't sure their incomes. A relatively big population refused to reveal their incomes too which is seen against 99.

Question 2:

For comparison of risk scores in the above question, perform the following analysis and describe in your own words:

a) Perform ANOVA on the weighted risk score (part ii) on the eight risk score categories (except 0) created in part i

Here Score1 is the Naïve score, Score2 is the weighted score. When we group the data into eight categories, all the data that have Naïve Score =1 as group1, and similarly for all 8 score.

```
group1 = table[table['SCORE1'].astype(int) == 1]
group2 = table[table['SCORE1'].astype(int) == 2]
group3 = table[table['SCORE1'].astype(int) == 3]
group4 = table[table['SCORE1'].astype(int) == 4]
group5 = table[table['SCORE1'].astype(int) == 5]
group6 = table[table['SCORE1'].astype(int) == 6]
group7 = table[table['SCORE1'].astype(int) == 7]
group8 = table[table['SCORE1'].astype(int) == 8]
```

We now have 8 groups, who's means we are calculating by the weighted Score2. We need to find if there is any statistical difference in the means of the groups. The one-way ANOVA test cannot tell which specific groups were statistically significantly different from each other; it only tells that at least two groups were different.

Null hypothesis: states that the means of all eight groups are equal.
Alternative hypothesis: states that the means of at least two of the groups is different.

There is NO significant difference in groups, or more generally, that there is no association between 8 groups. In other words, it is describing an outcome that is the opposite of the research hypothesis. The original speculation is not supported. The p value that came out of the ANOVA test was 0. Therefore, we can reject the null hypothesis which was where we started i.e. assuming the means might be equal amongst the groups.

```
fvalue:  7749.8973726
pvalue:  0.0
```

The ANOVA test does not give specifics, but we can conclude that the group means are not equal.

b) Identify top 50 and bottom 50 patients in each risk category (part i: 1 through 8) and compare average mean scores obtained in part ii

	SCORE1	SCORE2
0	1	0.111077
1	2	0.240580
2	3	0.375199
3	4	0.499402
4	5	0.630468
5	6	0.761559
6	7	0.883569
7	8	1.000000

The overall mean calculated for the eight groups (grouped by the Naïve Score), the weighted score columns are in the table.

Ex: For all entries that had a Naïve Score 1, that group had a weighted score of 0.11.

This mean was compared with the top 50 and bottom 50 of the same group and its weighted mean was calculated. The results turned out to be approximately the same.

Group 1 results:

```
Mean of weighted scores for only top 50 and last 50 entries, grouped by Naive score as 1: 0.0980199
Mean of weighted scores for all entries grouped by Naive Score as 1: 0.111077
```

Group 2 results:

```
Mean of weighted scores for only top 50 and last 50 entries, grouped by Naive score as 2: 0.23482
Mean of weighted scores for all entries grouped by Naive Score as 2: 0.240580
```

Group 3 results:

```
Mean of weighted scores for only top 50 and last 50 entries, grouped by Naive score as 3: 0.3945599
Mean of weighted scores for all entries grouped by Naive Score as 3: 0.375199
```

Group 4 results:

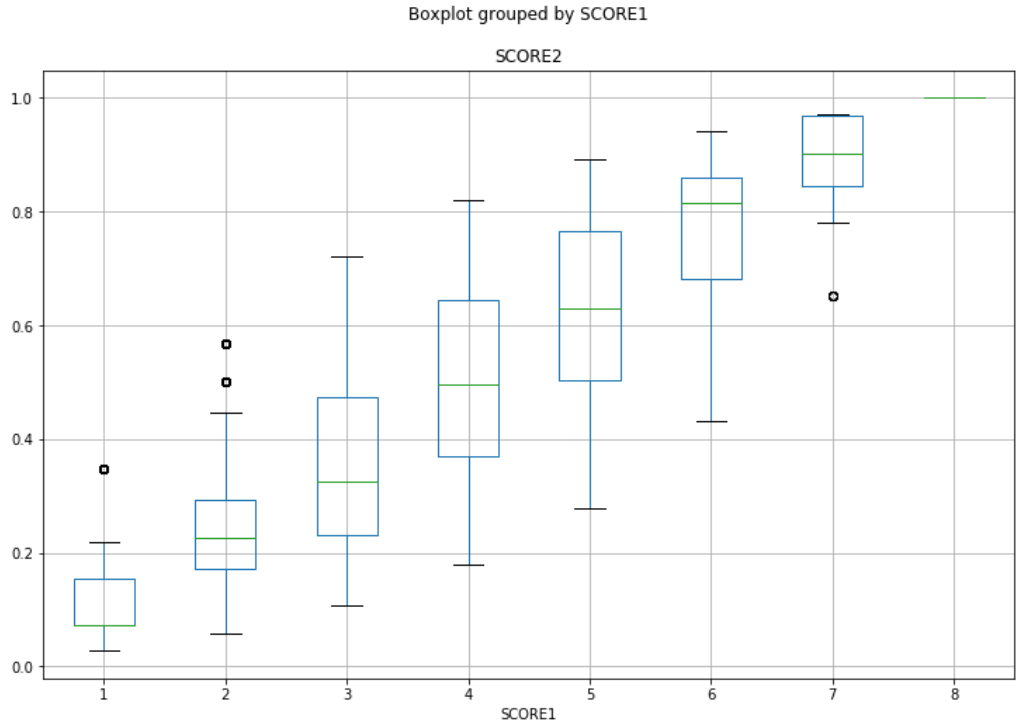
```
Mean of weighted scores for only top 50 and last 50 entries, grouped by Naive score as 4: 0.4909199
Mean of weighted scores for all entries grouped by Naive Score as 4: 0.499402
```

Group 5 results:
Mean of weighted scores for only top 50 and last 50 entries, grouped by Naive score as 5: **0.6852000**
Mean of weighted scores for all entries grouped by Naive Score as 5: **0.630468**

Group 6 results:
Mean of weighted scores for only top 50 and last 50 entries, grouped by Naive score as 6: **0.7742799**
Mean of weighted scores for all entries grouped by Naive Score as 6: **0.761559**

Group 7 results:
Mean of weighted scores for only top 50 and last 50 entries, grouped by Naive score as 7: **0.8851999**
Mean of weighted scores for all entries grouped by Naive Score as 7: **0.883569**

Group 8 results:
Mean of weighted scores for only top 50 and last 50 entries, grouped by Naive score as 8: **1.0**
Mean of weighted scores for all entries grouped by Naive Score as 8: **1.0**



c) Perform ML techniques to predict Weighted Risk Score created in part ii

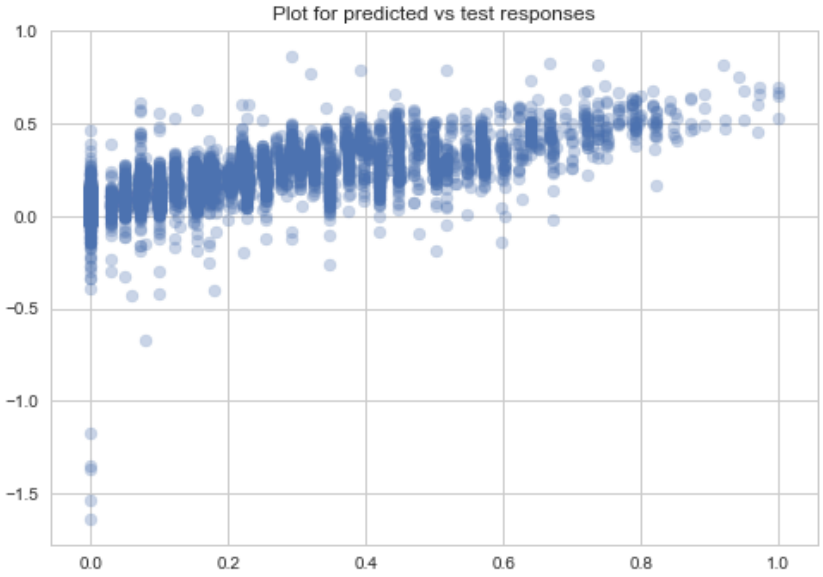
Data Cleaning:

There were many columns that contained more than 47K NA's, hence these columns were dropped. There were more columns with 20K NA's which were also dropped. Eventually we ended up with columns containing a few hundred NA's which we dropped as rows. Out of the total 47,891 rows and 275 columns, the final model had 47174 rows and 106 columns. The data was further standardized. The data was split to test and train after. We remove all the columns that we construct earlier for the previous questions to avoid an R Square value of 1.

Data Modelling:

1. **Linear Regression –**

The number of predictors used is 78 and we predict the response of the '**SCORE2'** column. After fitting the model, we observe the predicted response vs the actual response.



Mean squared error: 0.01
Variance score: **0.60**

The graph appears to be in a linear fashion which suggest that our model is relatively good. And the variance is at 60% which tells that the model can be improved upon.

2. **Lasso Regression –**

Lasso method is used for subset selection and it also helps in reducing the coefficients of unnecessary variables to zero. The model coefficients are as listed below.

{'GENHLTH': 0.028186866741156839, 'PHYSHLTH': -0.0067136871944270936, 'MENTHLTH': -0.0052741237197355016, 'HLTHPLN1': -0.001069
1176603471761, 'PERSDOC2': -0.00082766233659446753, 'MEDCOST': 0.0011634469890903852, 'CHECKUP1': -0.0054251373224861246, 'EXER
ANY2': 0.0057484877717761725, 'SLEPTIM1': 6.7727956323085551e-05, 'CVDINFR4': -0.019272504042560572, 'CVDCRHD4': -0.01333201480
3560747, 'CVDSTRK3': -0.013125831794775118, 'ASTHMA3': 0.0060432548176405218, 'CHCSCNCR': -0.015231685346631186, 'CHCOCNCR':
-0.017604551296283407, 'CHCCOPD1': 0.004718895953499419, 'HAVARTH3': -0.020170387775418897, 'ADDEPEV2': -0.018754421573464453,
'CHCKIDNY': 0.011498964208348013, 'DIABETE3': -0.070404116919333237, 'LASTDEN3': 0.0046724751172041541, 'RMVTETH3': -0.00056075
110620517439, 'SEX': -0.006120833157428676, 'MARITAL': -0.00082475138284459845, 'EDUCA': -0.0037743646506468572, 'RENTHOM1': 0.
006103174366146766, 'VETERAN3': -0.0019049613590722972, 'EMPLOY1': 0.013865112828732082, 'CHILDREN': 0.00085529157490321975, 'I
NCOME2': 0.00053667426557754032, 'INTERNET': 0.0011097354110700054, 'WEIGHT2': -0.00081491397991398167, 'HEIGHT3': 0.0002224300
9948825381, 'QSTVER': 0.001092289876674244, 'QSTLANG': -0.0020669278005683283, '_STSTR': 0.0010715679742865048, '_STRWT': 0.0,
'_RAWRAKE': -0.00058315581633658223, '_WT2RAKE': 0.00086796326670097426, '_DUALUSE': -0.0010221968067200577, '_LLCPWT2': -0.001
4934086089529726, '_LLCPWT': 0.0014556437974404029, '_PHYS14D': 0.0028982793174340731, '_MENT14D': 0.0071500338427602375, '_HCV
U651': 0.0037796874221282459, '_TOTINDA': -0.0025063520628604134, '_LTASTH1': 0.019152558551093036, '_CASTHM1': 0.0123334657082
15042, '_ASTHMS1': -0.016918404337713064, '_EXTETH2': 0.0026158093673453202, '_DENVST2': -0.0013177444385362998, '_PRACE1': 0.
0, '_MRACE1': 0.0010473005889629914, '_HISPANC': 0.0060462085196840189, '_RACE': 0.001982187713930667, '_RACEG21': -0.003757415
6318353935, '_RACEGR3': 0.0, '_AGEG5YR': 0.0082675221850365757, '_AGE65YR': 0.0021662304031867799, '_AGE80': 0.0347298052628345
21, '_AGE_G': -0.013001975511825819, '_RFBMI5': 0.0, '_CHLDCNT': -0.001267757211028209, '_EDUCAG': 0.0073448112139566733, '_INC
OMG': -0.003855186541037268, '_SMOKER3': -0.0058165516029868121, '_RFSMOK3': 0.0036103229678129006, '_ECIGSTS': -0.002476173808
6515228, '_CURECIG': 0.0034824825758250281, 'DRNKANY5': 0.0095617999380729485, 'DROCDY3_': -0.0065210815684650179, '_RFBING5':
-0.0020793081026186525, '_DRNKWEK': 0.0, '_RFDRHV5': 0.00090194570658202924, '_RFSEAT2': -0.0019268498979659839, '_RFSEAT3':
-0.0, '_DRNKDRV': -0.00120436452823006, '_RFHLTH': 0.0024496518598136 61}

R SQUARE: 0.58658579234
MEAN SQUARED ERROR: 0.0138111048175

This method still gives a poor R Square value which is hoped to improve in a MARS and Random forest model.

3. MARS

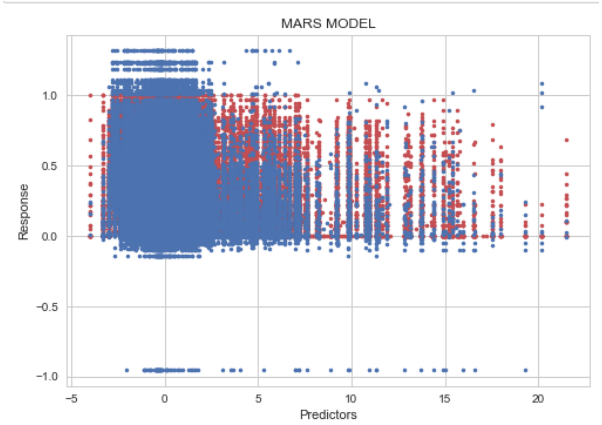
A MARS model with a higher degree of freedom gives a higher R square value as expected.

Max_degree = 3

Forward Pass

iter	parent	var	knot	mse	terms	gcv	rsq	grsq
0	-	-	-	0.034201	1	0.034	0.000	0.000
1	0	19	-1	0.024177	2	0.024	0.293	0.293
2	0	0	-1	0.020954	3	0.021	0.387	0.387
3	0	59	-1	0.018949	4	0.019	0.446	0.446
4	0	17	-1	0.018037	5	0.018	0.473	0.472
5	4	17	-1	0.016461	6	0.016	0.519	0.518
6	0	9	-1	0.015486	7	0.015	0.547	0.547
7	6	9	-1	0.011507	8	0.012	0.664	0.663
8	0	14	-1	0.010521	9	0.011	0.692	0.692
9	8	14	-1	0.009810	10	0.010	0.713	0.713
10	0	16	-1	0.009014	11	0.009	0.736	0.736
11	0	13	-1	0.008328	12	0.008	0.756	0.756
12	11	13	-1	0.007711	13	0.008	0.775	0.774
13	0	11	-1	0.006976	14	0.007	0.796	0.796
14	13	11	-1	0.005829	15	0.006	0.830	0.829
15	10	16	-1	0.005446	16	0.005	0.841	0.840
16	0	10	-1	0.005044	17	0.005	0.853	0.852
17	16	10	-1	0.003660	18	0.004	0.893	0.893
18	11	11	-1	0.003271	19	0.003	0.904	0.904
19	1	19	-1	0.002933	20	0.003	0.914	0.914
20	0	12	-1	0.002686	21	0.003	0.921	0.921
21	0	46	-1	0.002563	22	0.003	0.925	0.925
22	8	13	-1	0.002512	23	0.003	0.927	0.926
23	6	10	-1	0.002473	24	0.002	0.928	0.927
24	0	15	-1	0.002452	25	0.002	0.928	0.928

Stopping Condition 2: Improvement below threshold



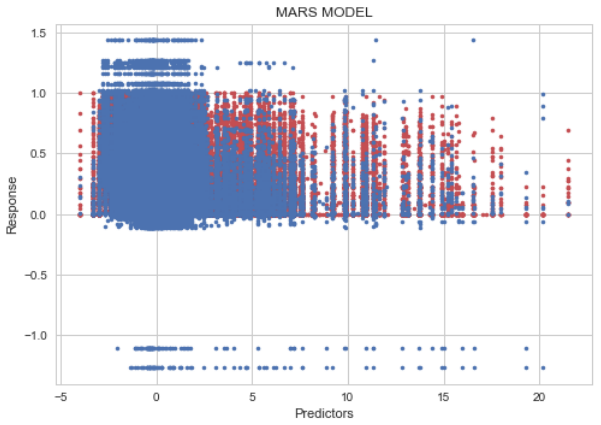
MSE: 0.0025, GCV: 0.0025, RSQ: 0.9283, GRSQ: 0.9281

Max_degree=2

Forward Pass

iter	parent	var	knot	mse	terms	gcv	rsq	grsq
0	-	-	-	0.034201	1	0.034	0.000	0.000
1	0	19	-1	0.024177	2	0.024	0.293	0.293
2	0	0	-1	0.020954	3	0.021	0.387	0.387
3	0	59	-1	0.018949	4	0.019	0.446	0.446
4	0	17	-1	0.018037	5	0.018	0.473	0.472
5	4	17	-1	0.016461	6	0.016	0.519	0.518
6	0	9	-1	0.015486	7	0.015	0.547	0.547
7	6	9	-1	0.011507	8	0.012	0.664	0.663
8	0	14	-1	0.010521	9	0.011	0.692	0.692
9	8	14	-1	0.009810	10	0.010	0.713	0.713
10	0	16	-1	0.009014	11	0.009	0.736	0.736
11	0	13	-1	0.008328	12	0.008	0.756	0.756
12	11	13	-1	0.007711	13	0.008	0.775	0.774
13	0	11	-1	0.006976	14	0.007	0.796	0.796
14	13	11	-1	0.005829	15	0.006	0.830	0.829
15	10	16	-1	0.005446	16	0.005	0.841	0.840
16	0	10	-1	0.005044	17	0.005	0.853	0.852
17	16	10	-1	0.003660	18	0.004	0.893	0.893
18	17	12	-1	0.003263	19	0.003	0.905	0.904
19	1	19	-1	0.002933	20	0.003	0.914	0.914
20	0	12	-1	0.002710	21	0.003	0.921	0.921
21	0	46	-1	0.002589	22	0.003	0.924	0.924
22	19	19	-1	0.002481	23	0.002	0.927	0.927
23	7	9	-1	0.002386	24	0.002	0.930	0.930
24	17	10	-1	0.002316	25	0.002	0.932	0.932
25	6	10	-1	0.002270	26	0.002	0.934	0.933
26	17	9	-1	0.002027	27	0.002	0.941	0.941
27	7	10	-1	0.001944	28	0.002	0.943	0.943
28	8	13	-1	0.001895	29	0.002	0.945	0.944
29	12	14	-1	0.001776	30	0.002	0.948	0.948
30	28	15	-1	0.001677	31	0.002	0.951	0.951
31	5	17	-1	0.001650	32	0.002	0.952	0.952

Stopping Condition 2: Improvement below threshold



MSE: 0.0017, GCV: 0.0017, RSQ: 0.9518, GRSQ: 0.9516

4. RANDOM FOREST

The random forest constructs multiple trees instead of one big tree. Error estimated on these out of bag samples is known as out of bag error. Therefore, using the out-of-bag error estimate removes the need for a set aside test set. This model gives the best (probably incorrect) prediction of 99% which means that it can exactly predict.

Out-of-bag R-2 score estimate: 0.984
Test data R-2 score: 0.999
Test data Spearman correlation: 1.0