In this assigment we will use a lasso regression to choose the most suitable subset from a list of 23 quantitative and categorical predictors. Firstly, we should import all needed dependencies.

```
In [1]: import sklearn as sk
        import pandas as pd
        import numpy as np
        from sklearn.model_selection import train_test_split
        import matplotlib.pyplot as plt
        from sklearn.metrics  import mean_squared_error
        import warnings
        from sklearn.preprocessing import StandardScaler
        from sklearn.linear_model import LassoLarsCV
        warnings.filterwarnings('ignore')
        %matplotlib inline
```

Loading data:

```
In [2]: data=pd.read_csv('tree_add_health.csv')
```

Data transformation:

```
In [3]: recode1 = {1:1, 2:0}
        data['MALE']= data['BIO_SEX'].map(recode1)
```

Replacing missing values in each feature by its mean.

```
In [4]: data.fillna(np.round(data.mean()),inplace=True)
```

Out[4]:

	BIO_SEX	HISPANIC	WHITE	BLACK	NAMERICAN	ASIAN	age	TREG1	\
0	2.0	0.0	0.0	1.0	0.0	0.0	17.000000	0.0	
1	2.0	0.0	0.0	1.0	0.0	0.0	19.427397	1.0	
2	1.0	0.0	1.0	0.0	0.0	0.0	17.000000	0.0	
3	1.0	0.0	0.0	1.0	0.0	0.0	20.430137	1.0	
4	2.0	0.0	0.0	1.0	0.0	0.0	17.000000	0.0	
5	1.0	0.0	0.0	1.0	0.0	0.0	14.509589	0.0	
6	1.0	0.0	0.0	1.0	0.0	0.0	13.676712	0.0	
7	1.0	0.0	1.0	0.0	0.0	0.0	15.178082	1.0	
8	1.0	0.0	0.0	1.0	0.0	0.0	14.673973	0.0	
9	1.0	0.0	1.0	0.0	0.0	0.0	14.926027	0.0	
10	1.0	0.0	0.0	1.0	0.0	0.0	15.591781	0.0	
11	1.0	0.0	0.0	1.0	0.0	0.0	17.342466	0.0	
12	1.0	0.0	1.0	0.0	0.0	0.0	16.342466	0.0	
13	1.0	0.0	1.0	0.0	0.0	0.0	17.000000	1.0	
14	2.0	0.0	0.0	1.0	0.0	0.0	13.509589	0.0	
15	2.0	0.0	1.0	0.0	0.0	0.0	13.673973	0.0	
16	2.0	0.0	1.0	0.0	0.0	0.0	17.180822	0.0	
17	1.0	0.0	1.0	0.0	0.0	0.0	13.673973	0.0	
18	1.0	1.0	0.0	0.0	0.0	0.0	19.594521	0.0	
19	2.0	0.0	1.0	0.0	0.0	0.0	17.000000	1.0	

20	2.0	0.0	1.0	0.0	0.0	0.0	16.931507	0.0
21	2.0	1.0	0.0	0.0	0.0	0.0	15.591781	0.0
22	2.0	0.0	1.0	0.0	0.0	0.0	18.342466	0.0
23	2.0	0.0	0.0	0.0	0.0	1.0	14.339726	0.0
24	1.0	0.0	0.0	1.0	0.0	0.0	15.339726	0.0
25	1.0	0.0	1.0	0.0	0.0	0.0	17.000000	0.0
26	1.0	0.0	0.0	1.0	0.0	0.0	16.594521	0.0
27	1.0	1.0	1.0	0.0	0.0	0.0	17.509589	0.0
28	1.0	0.0	1.0	0.0	0.0	0.0	14.928767	0.0
29	2.0	0.0	1.0	0.0	0.0	0.0	14.339726	0.0
...	...	...	...	...	...	...	...	...
6474	2.0	0.0	0.0	1.0	0.0	0.0	14.926027	0.0
6475	2.0	0.0	1.0	0.0	0.0	0.0	15.345205	0.0
6476	2.0	0.0	1.0	0.0	0.0	0.0	14.339726	0.0
6477	2.0	0.0	0.0	1.0	0.0	0.0	17.000000	0.0
6478	1.0	0.0	0.0	1.0	0.0	0.0	17.000000	0.0
6479	1.0	0.0	0.0	1.0	0.0	0.0	14.424658	0.0
6480	1.0	0.0	1.0	0.0	0.0	0.0	15.095890	0.0
6481	2.0	0.0	1.0	0.0	0.0	0.0	15.178082	0.0
6482	1.0	0.0	0.0	1.0	0.0	0.0	13.506849	0.0
6483	1.0	0.0	0.0	1.0	0.0	0.0	15.843836	0.0
6484	1.0	0.0	1.0	0.0	0.0	0.0	14.178082	0.0
6485	1.0	0.0	0.0	1.0	0.0	0.0	17.000000	1.0
6486	2.0	0.0	0.0	1.0	0.0	0.0	14.010959	0.0
6487	1.0	0.0	0.0	1.0	0.0	0.0	15.263014	0.0
6488	1.0	0.0	0.0	1.0	0.0	0.0	14.591781	0.0
6489	2.0	0.0	0.0	1.0	0.0	0.0	14.509589	0.0
6490	1.0	0.0	0.0	1.0	0.0	0.0	15.424658	0.0
6491	1.0	0.0	1.0	0.0	0.0	0.0	15.339726	0.0
6492	2.0	0.0	1.0	0.0	0.0	0.0	14.673973	0.0
6493	2.0	0.0	0.0	1.0	0.0	0.0	14.172603	0.0
6494	2.0	0.0	0.0	1.0	0.0	0.0	14.172603	0.0
6495	2.0	1.0	0.0	0.0	0.0	0.0	15.093151	0.0
6496	1.0	0.0	1.0	0.0	0.0	0.0	16.178082	0.0
6497	1.0	0.0	1.0	0.0	0.0	0.0	15.345205	0.0
6498	2.0	0.0	1.0	0.0	0.0	0.0	14.424658	0.0
6499	2.0	0.0	0.0	1.0	0.0	0.0	14.260274	0.0
6500	1.0	0.0	0.0	1.0	0.0	0.0	17.000000	0.0
6501	1.0	0.0	1.0	0.0	0.0	0.0	15.093151	1.0
6502	2.0	0.0	1.0	0.0	0.0	0.0	15.509589	0.0
6503	1.0	0.0	1.0	0.0	0.0	0.0	17.000000	0.0

	ALCEVR1	ALCPROBS1	...	VIOL1	PASSIST	DEVIANT1	SCHCONN1	GPA1	\
0	1.0	2	...	4.0	0	5.0	28.0	3.000000	
1	1.0	1	...	1.0	0	5.0	22.0	2.333333	
2	0.0	0	...	0.0	0	1.0	30.0	2.250000	
3	0.0	0	...	4.0	1	4.0	19.0	2.000000	
4	1.0	0	...	0.0	0	5.0	32.0	3.000000	

5	0.0	0	...	3.0	0	0.0	27.0	2.666667
6	0.0	0	...	5.0	0	7.0	18.0	2.500000
7	1.0	0	...	8.0	1	6.0	20.0	1.500000
8	1.0	0	...	0.0	0	5.0	24.0	2.250000
9	0.0	0	...	1.0	0	2.0	25.0	2.500000
10	0.0	0	...	1.0	0	2.0	27.0	3.250000
11	0.0	0	...	0.0	0	1.0	30.0	2.000000
12	0.0	0	...	0.0	0	0.0	29.0	2.250000
13	0.0	0	...	11.0	0	9.0	28.0	3.750000
14	0.0	0	...	0.0	0	1.0	30.0	3.000000
15	0.0	0	...	0.0	0	1.0	28.0	3.750000
16	0.0	0	...	0.0	0	1.0	32.0	2.250000
17	0.0	0	...	0.0	0	0.0	27.0	4.000000
18	1.0	1	...	0.0	0	3.0	31.0	2.666667
19	1.0	0	...	0.0	0	1.0	28.0	1.666667
20	1.0	1	...	0.0	1	1.0	29.0	3.500000
21	1.0	0	...	3.0	1	8.0	31.0	2.750000
22	1.0	0	...	4.0	0	4.0	27.0	1.750000
23	1.0	0	...	0.0	0	0.0	34.0	3.500000
24	0.0	0	...	8.0	0	0.0	25.0	1.000000
25	0.0	0	...	0.0	1	6.0	22.0	2.750000
26	0.0	0	...	0.0	0	0.0	26.0	3.000000
27	1.0	0	...	3.0	1	2.0	28.0	2.250000
28	0.0	0	...	0.0	0	0.0	31.0	1.750000
29	0.0	0	...	2.0	0	1.0	23.0	3.500000
...	...	...	...	...	...	...	...	...
6474	0.0	0	...	1.0	0	3.0	36.0	3.000000
6475	1.0	0	...	3.0	0	0.0	27.0	3.500000
6476	0.0	0	...	0.0	0	0.0	38.0	4.000000
6477	1.0	0	...	5.0	0	4.0	31.0	2.666667
6478	1.0	0	...	6.0	0	4.0	27.0	3.000000
6479	0.0	0	...	0.0	0	1.0	29.0	2.750000
6480	0.0	0	...	0.0	0	0.0	36.0	4.000000
6481	0.0	0	...	1.0	0	0.0	34.0	3.250000
6482	1.0	0	...	0.0	0	0.0	28.0	2.750000
6483	0.0	0	...	3.0	0	0.0	22.0	1.750000
6484	1.0	0	...	0.0	0	1.0	16.0	2.500000
6485	0.0	0	...	3.0	0	2.0	31.0	2.750000
6486	0.0	0	...	0.0	0	3.0	31.0	2.750000
6487	1.0	0	...	4.0	0	1.0	29.0	2.250000
6488	0.0	0	...	0.0	0	0.0	34.0	3.250000
6489	1.0	0	...	2.0	1	2.0	34.0	2.250000
6490	0.0	0	...	1.0	0	1.0	32.0	3.250000
6491	0.0	0	...	0.0	0	1.0	36.0	3.750000
6492	1.0	0	...	1.0	0	0.0	30.0	3.250000
6493	0.0	0	...	0.0	0	0.0	27.0	3.500000
6494	0.0	0	...	0.0	0	0.0	17.0	3.750000
6495	0.0	0	...	0.0	0	0.0	36.0	3.250000

6496	0.0	0	...	0.0	0	0.0	29.0	2.500000
6497	0.0	0	...	0.0	0	0.0	33.0	3.250000
6498	0.0	0	...	5.0	1	2.0	32.0	3.666667
6499	0.0	0	...	2.0	0	2.0	27.0	3.500000
6500	0.0	0	...	0.0	1	0.0	32.0	3.000000
6501	1.0	4	...	13.0	0	2.0	14.0	1.000000
6502	0.0	0	...	1.0	1	6.0	26.0	3.000000
6503	1.0	1	...	15.0	1	14.0	25.0	1.250000

	EXPEL1	FAMCONCT	PARACTV	PARPRES	MALE
0	0.0	24.3	8.0	15.0	0.0
1	0.0	23.3	9.0	15.0	0.0
2	0.0	24.3	3.0	15.0	1.0
3	0.0	18.7	6.0	14.0	1.0
4	0.0	20.0	9.0	6.0	0.0
5	0.0	23.7	3.0	13.0	1.0
6	0.0	24.7	6.0	13.0	1.0
7	0.0	22.3	10.0	14.0	1.0
8	0.0	19.0	8.0	15.0	1.0
9	0.0	24.3	3.0	11.0	1.0
10	0.0	20.0	3.0	13.0	1.0
11	0.0	23.0	9.0	11.0	1.0
12	0.0	24.0	8.0	14.0	1.0
13	0.0	21.3	7.0	8.0	1.0
14	0.0	23.7	8.0	9.0	0.0
15	0.0	24.0	5.0	15.0	0.0
16	0.0	24.3	14.0	14.0	0.0
17	0.0	24.0	12.0	14.0	1.0
18	0.0	7.0	4.0	13.0	1.0
19	0.0	19.3	4.0	15.0	0.0
20	0.0	24.0	5.0	13.0	0.0
21	0.0	23.3	8.0	10.0	0.0
22	0.0	18.0	2.0	15.0	0.0
23	0.0	24.7	6.0	14.0	0.0
24	0.0	23.0	1.0	13.0	1.0
25	0.0	23.7	4.0	11.0	1.0
26	0.0	22.3	0.0	12.0	1.0
27	0.0	24.0	6.0	15.0	1.0
28	0.0	23.3	8.0	12.0	1.0
29	0.0	24.3	8.0	13.0	0.0
...	...	...	...	...	...
6474	0.0	24.3	3.0	9.0	0.0
6475	0.0	23.0	1.0	15.0	0.0
6476	0.0	25.0	6.0	15.0	0.0
6477	0.0	24.0	9.0	15.0	0.0
6478	0.0	22.3	8.0	13.0	1.0
6479	0.0	23.7	6.0	15.0	1.0
6480	0.0	24.3	10.0	11.0	1.0

```
6481     0.0      22.0      6.0     15.0   0.0
6482     0.0      25.0      9.0     15.0   1.0
6483     0.0      24.7     11.0     15.0   1.0
6484     0.0      24.3      3.0     14.0   1.0
6485     1.0      19.7      3.0     12.0   1.0
6486     0.0      23.3      7.0     11.0   0.0
6487     0.0      19.0      2.0     12.0   1.0
6488     0.0      24.3     12.0     14.0   1.0
6489     0.0      24.0      5.0     15.0   0.0
6490     0.0      24.7     12.0     13.0   1.0
6491     0.0      25.0     11.0     15.0   1.0
6492     0.0      23.3      6.0     11.0   0.0
6493     0.0      25.0      7.0     12.0   0.0
6494     0.0      23.3      9.0     11.0   0.0
6495     0.0      25.0      6.0     15.0   0.0
6496     0.0      25.0      6.0     11.0   1.0
6497     0.0      21.7      2.0     11.0   1.0
6498     0.0      25.0      6.0     15.0   0.0
6499     0.0      23.0      3.0     14.0   0.0
6500     0.0      23.3      9.0     15.0   1.0
6501     0.0      23.7      9.0     13.0   1.0
6502     0.0      21.3      1.0     12.0   0.0
6503     0.0      17.0      5.0      8.0   1.0

[6504 rows x 26 columns]
```

Selecting variables:

```
In [14]: data_clean= data[['MALE','HISPANIC','WHITE','BLACK','NAMERICAN','ASIAN',
         'age','ALCEVR1','ALCPROBS1','marever1','cocever1','inhever1','cigavail','DEP1',
         'ESTEEM1','VIOL1','PASSIST','DEVIANT1','GPA1','EXPEL1','FAMCONCT','PARACTV',
         'PARPRES']]
         data_clean=pd.DataFrame(StandardScaler().fit_transform(data_clean.values),\
                                 index=data_clean.index, columns=data_clean.columns)

         data_clean.head()
```

```
Out[14]:        MALE  HISPANIC     WHITE     BLACK  NAMERICAN     ASIAN       age  \
         0 -0.968217 -0.359125 -1.403035  1.737036   -0.19404 -0.208113  0.247290
         1 -0.968217 -0.359125 -1.403035  1.737036   -0.19404 -0.208113  2.007879
         2  1.032826 -0.359125  0.712741 -0.575693   -0.19404 -0.208113  0.247290
         3  1.032826 -0.359125 -1.403035  1.737036   -0.19404 -0.208113  2.735166
         4 -0.968217 -0.359125 -1.403035  1.737036   -0.19404 -0.208113  0.247290

            ALCEVR1  ALCPROBS1  marever1     ...          DEP1   ESTEEM1     VIOL1  \
         0  0.892572   1.680467  1.617202     ...      0.181682  1.140494  0.926729
         1  0.892572   0.627808 -0.618352     ...      1.214601 -1.075145 -0.233287
         2 -1.120358  -0.424851 -0.618352     ...     -0.998797  0.771221 -0.619959
```

```
3 -1.120358  -0.424851  1.617202    ...     1.509720  1.140494  0.926729
4  0.892572  -0.424851 -0.618352    ...    -0.408557 -0.336598 -0.619959

    PASSIST  DEVIANT1      GPA1    EXPEL1  FAMCONCT   PARACTV   PARPRES
0 -0.335209  0.642898  0.253155 -0.219515  0.626122  0.561220  0.774191
1 -0.335209  0.642898 -0.630823 -0.219515  0.333122  0.858708  0.774191
2 -0.335209 -0.498550 -0.741320 -0.219515  0.626122 -0.926220  0.774191
3  2.983210  0.357536 -1.072812 -0.219515 -1.014681 -0.033756  0.305351
4 -0.335209  0.642898  0.253155 -0.219515 -0.633781  0.858708 -3.445365

[5 rows x 23 columns]
```

Splitting all dataset into train and test parts as relation 3 to 1.

```
In [6]: X_train, X_test, y_train, y_test = train_test_split(data_clean, data['SCHCONN1'], test_s
```

I've used the least angle regression algorithm for model estimation:

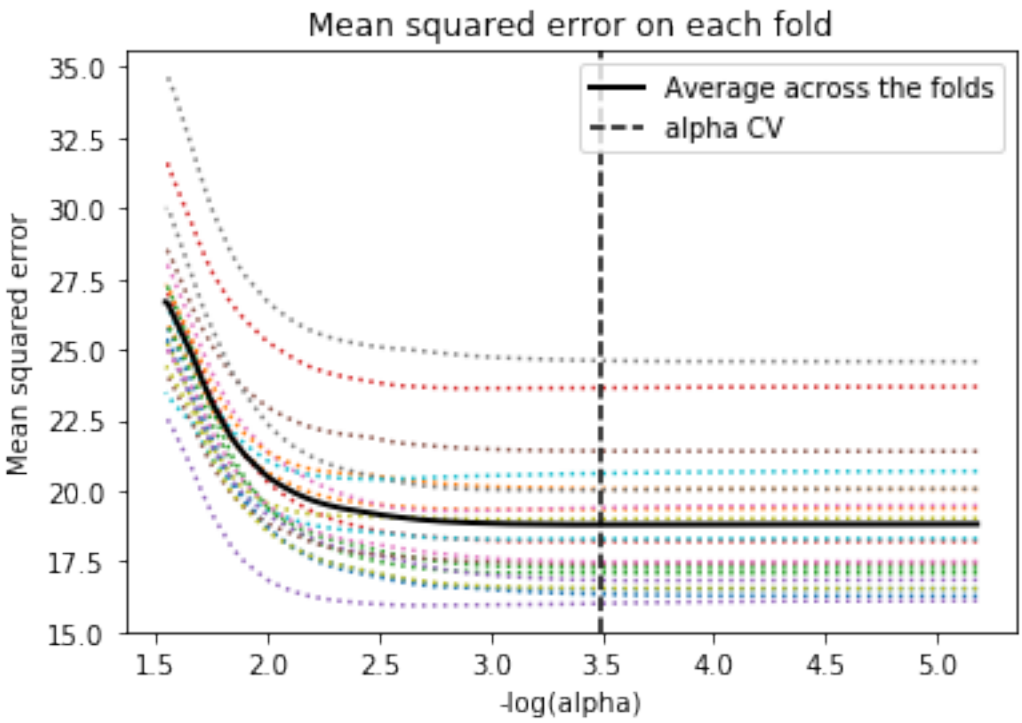
```
In [7]: model=LassoLarsCV(cv=20, precompute=False).fit(X_train,y_train)

In [8]: model

Out[8]: LassoLarsCV(copy_X=True, cv=20, eps=2.2204460492503131e-16,
             fit_intercept=True, max_iter=500, max_n_alphas=1000, n_jobs=1,
             normalize=True, positive=False, precompute=False, verbose=False)

In [10]: m_log_alphascv = -np.log10(model.cv_alphas_)
         plt.figure()
         plt.plot(m_log_alphascv, model.cv_mse_path_, ':')
         plt.plot(m_log_alphascv, model.cv_mse_path_.mean(axis=-1), 'k',
                  label='Average across the folds', linewidth=2)
         plt.axvline(-np.log10(model.alpha_), linestyle='--', color='k',
                     label='alpha CV')
         plt.legend()
         plt.xlabel('-log(alpha)')
         plt.ylabel('Mean squared error')
         plt.title('Mean squared error on each fold')

Out[10]: Text(0.5,1,'Mean squared error on each fold')
```



MSE

```
In [11]: error_train = mean_squared_error(y_train, model.predict(X_train))
         error_test = mean_squared_error (y_test, model.predict(X_test))
         print ('training MSE')
         print(error_train)
         print ('test MSE')
         print(error_test)
```

```
training data MSE
18.6168539056
test data MSE
17.4642321746
```

R-square

```
In [12]: rsquared_train=model.score(X_train,y_train, )
         rsquared_test=model.score (X_test,y_test)
         print ('training R-square')
         print(rsquared_train)
         print ('test R-square')
         print(rsquared_test)
```

```
training data R-square
0.302014887231
```

```
test data R-square
0.307626126173
```

Coefficients estimations:

```
In [13]: dict(zip(data_clean.columns, model.coef_))
```

```
Out[13]: {'ALCEVR1': -0.24921749531225018,
          'ALCPROBS1': 0.0,
          'ASIAN': 0.21983231321220215,
          'BLACK': -0.19836232150286651,
          'DEP1': -0.941310723270814,
          'DEVIANT1': -0.31497503047867426,
          'ESTEEM1': 1.0441471822649511,
          'EXPEL1': -0.075777411971217026,
          'FAMCONCT': 0.30136597109987656,
          'GPA1': 0.75593081357937375,
          'HISPANIC': 0.25661430290401122,
          'MALE': -0.11864466692060359,
          'NAMERICAN': -0.046140325344907765,
          'PARACTV': 0.35213080170504368,
          'PARPRES': 0.075856131068225752,
          'PASSIST': 0.0,
          'VIOL1': -0.62782544578904809,
          'WHITE': 0.0,
          'age': 0.24285110507646082,
          'cigavail': -0.086046230991061165,
          'cocever1': -0.026578372137668709,
          'inhever1': -0.10205208974536151,
          'marever1': -0.20324715070886556}
```

As it clearly seen, three parameters was not been included into model.