Lasso And Elastic-Net

Hugo Henry Sabogal Rodriguez

Bioinformatics

1 Introduction

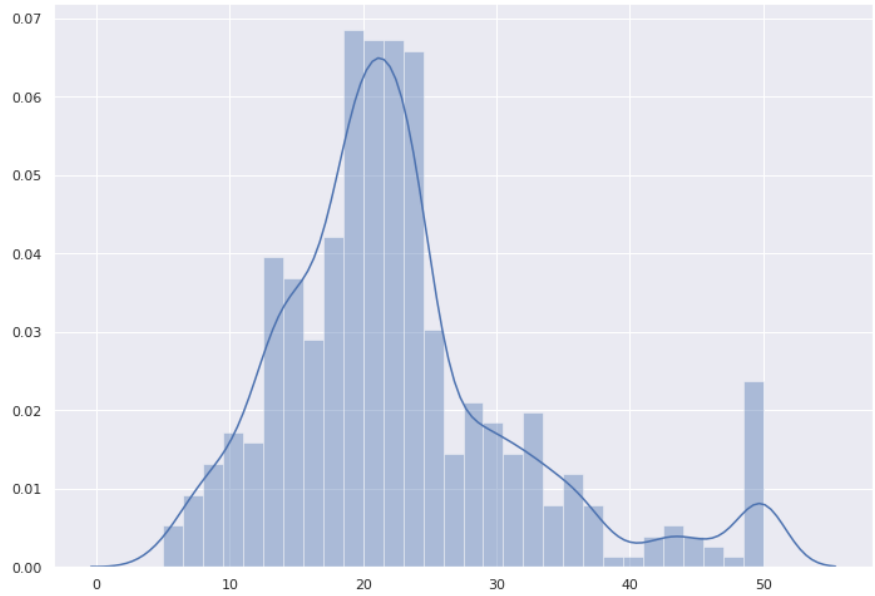
In this document, I show the results to the project Lasso and Elastic Net. First, I work with the Boston dataset which is integrated into SKlearn. The idea is to illustrate the basic concepts of the Lasso and Elastic Net. Second, I try to implement the tow-stages method proposed in the references. The method aims to build a sparse model to extract the most relevant features in a dataset. The two-stage method uses an iterative process with Elastic Net. To test the method, I use two synthetic datasets, each one with a different distribution. Finally, I test the method to select and to visualize a real dataset (the Golub dataset).

2 Results

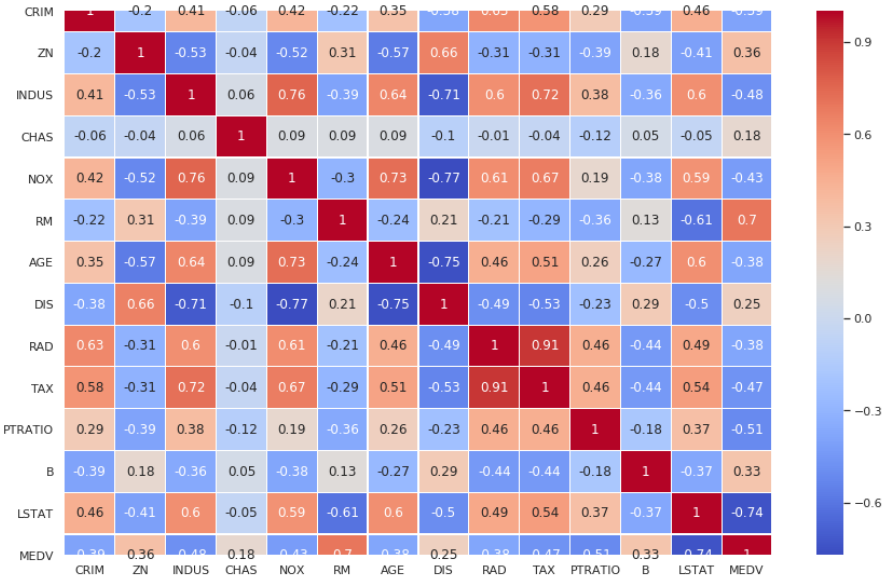
2.1 Lasso And Elastic Net

This section shows how Lasso can select a group of the most relevant features in a dataset. The dataset that I use for this section is the Boston Housing dataset which is integrated into the SKlearn package. The Boston dataset is small and contains 13 features, 506 samples and, the output consist of the median value of owner-occupied homes in $1000's. This dataset is suitable for regression problems. It is a small example, but it is useful to illustrate the concepts involved in Lasso model.

The dataset does not contain null values, also the values of the output are distributed normally and have a few outliers. The figure below shows the output distribution.



Now I plot the correlation matrix to identify the linear relationships between variables and the relationships between variables and output.



It can be seen that the RM feature has the strongest positive correlation (0.7) with the output MEDV. On the other hand, the variable LSTAT shows the strongest negative correlation (-0.74) with the target MEDV. Also, it is

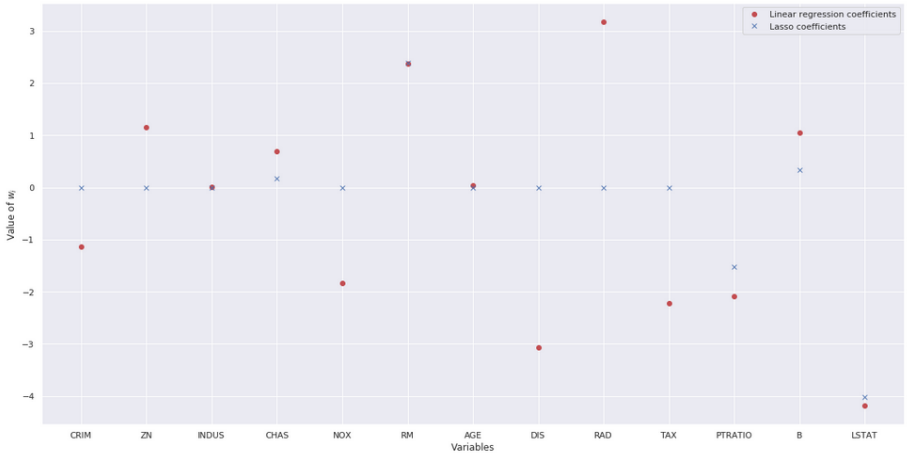
important to note the correlation between features. The features with a strong correlation are RAD and TAX, which have a correlation value of 0.91 and, the variable DIS shows a strong correlation with the variables NOX (-0.77) and AGE (-0.75) respectively.

A sparse model is one in which a small number of predictors play an important role. Lasso applies the l1-norm to create a sparse model but it could be useful to compare the performance in a linear regression model, which use all the predictors in the dataset, with the performance of the Lasso regression model. To make this, I split the data into training and test sets. Then, it is important to standardize the data set. First, I standardize the training set then, I use the same parameters obtained in the training set to standardize the test set.

The table below shows that the performance for both models, the linear model and lasso model is similar.

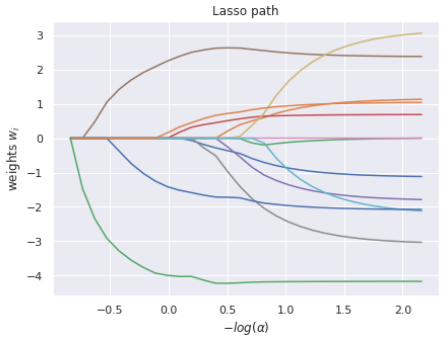
	MSE training set	MSE test set	SCORE training set	SCORE test set
Linear Regression	22.477090	20.869292	0.738339	0.733449
Lasso Regression	22.479623	20.776007	0.738310	0.734641

The graph below shows the values of the coefficients for both linear regression and lasso regression. According with the graph, with alpha = 0.8 the Lasso model sets seven of the thirteen predictors to zero. Also, Lasso tends to shrink the coefficients to the other features towards zero.



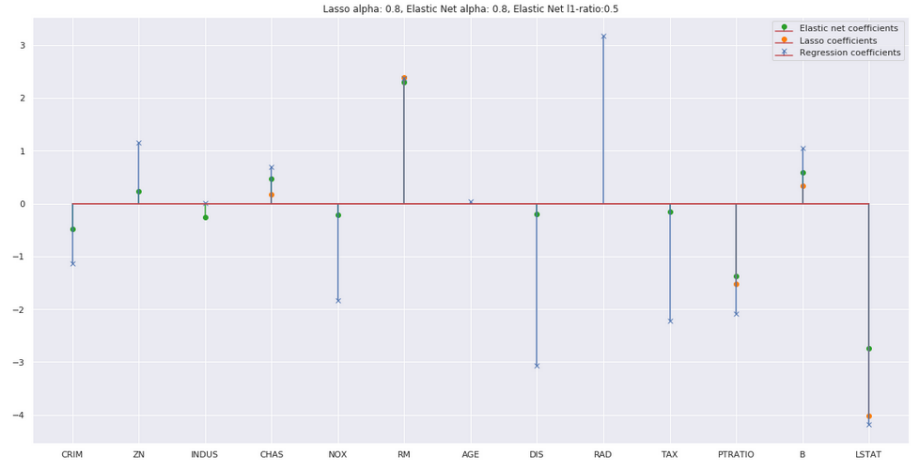
The coefficients with the highest values are the coefficients for LSTAT and RM. This is consistent with the results of the correlation matrix. So, Lasso selects the more important features in the dataset.

Changing the value for alpha we can set different coefficients to zero. The figure below shows how the Lasso set to zero the coefficients when alpha changes. With larger values for alpha Lasso shrink more coefficients to zero. When alpha decreases the Lasso allows to set more coefficients with a value different to zero.



Lasso is not good to handle highly correlated variables. The coefficient paths tend to be erratic or show unstable behavior. To overcome these limitations we can use a regularization method which is called Elastic net.

Using Elastic Net predictors that are highly correlated are selected together in their groups and, they share approximately the same value. The elastic net also tends to shrink the values to zero. The figure below compares the coefficients set for the linear regression model, the lasso model and, the elastic net model.

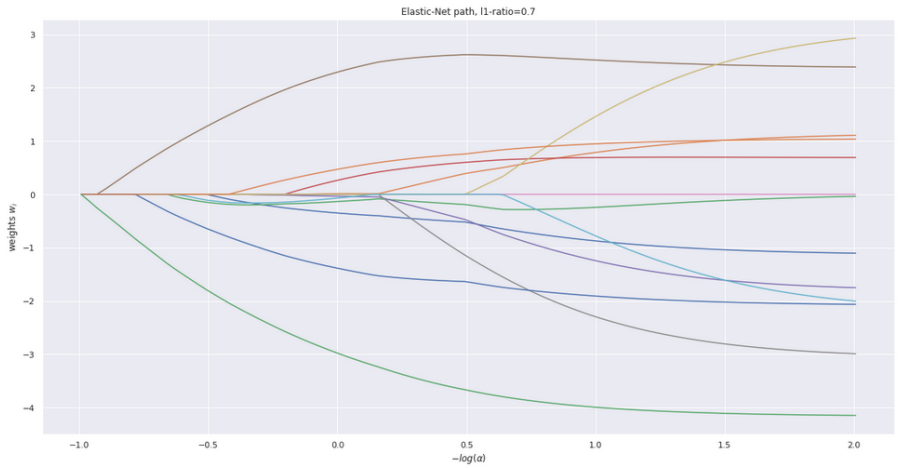


The table below shows the coefficients for each model. With alpha=0.8 for the Lasso model and, with alpha=0.8 and l1-ratio=0.5 for the Elastic net model,

it can be seen that elastic net sets to zero the features AGE and RAD. Also, coefficients tend to shrink to zero for the elastic net while for Lasso and the linear regression models tend to a higher value.

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
Linear Regression	-1.135027	1.158145	0.007371	0.687096	-1.828370	2.362719	0.031665	-3.066328	3.166215	-2.214579	-2.086009	1.044951	-4.176271
Lasso	-0.000000	0.000000	-0.000000	0.177560	-0.000000	2.385886	-0.000000	-0.000000	-0.000000	-0.000000	-1.518516	0.330622	-4.025193
Elastic Net	-0.478728	0.225515	-0.259681	0.471988	-0.216145	2.301519	-0.000000	-0.194887	-0.000000	-0.152487	-1.380558	0.583188	-2.743015

Finally, the figure below shows the elastic net path. The elastic net produces more non-zero coefficients that lasso, and with smaller magnitudes.



2.2 Synthetic dataset

For this section, I built a synthetic dataset in order to test the two stages method proposed in [1]. The experiment shows different results depending on the data distribution in the dataset. First, I built a dataset with a uniform distribution. Then, I built a synthetic dataset with a normal distribution.

2.2.1 Synthetic dataset with uniform distribution

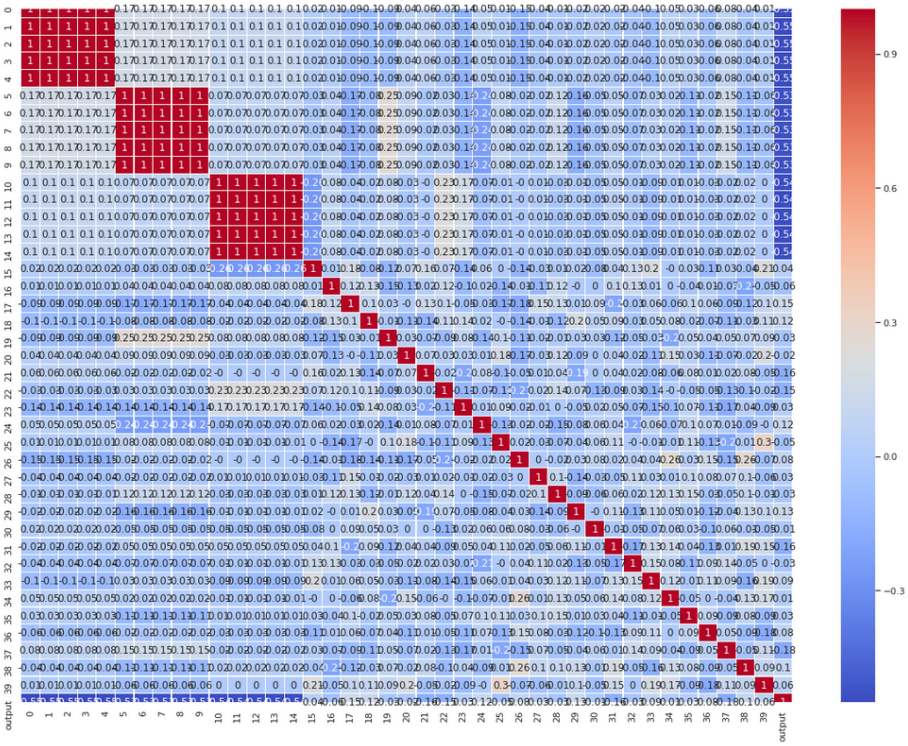
The first synthetic dataset consists in n=100 samples. To each sample there is associated a 40 dimensional vector x_i given by the following rules:

- There will be four groups G1, G2, G3, G4
- x_1, x_6, x_{11} are obtained by a random uniform distribution with values between -0.5 and 0.5.
- The group G1 is given by: $x_{1+i} = s_i * x_i + \sigma_1 * \epsilon$ for i = 1,...,4
- The group G2 is given by: $x_{6+i} = s_i * x_i + \sigma_2 * \epsilon$ for i = 1,...,4

- The group G3 is given by: $x_{11+i} = s_i * x_i + \sigma_3 * \epsilon$ for i = 1,...,4
- Where $s_i = 1, i = 1, ..., 4$ and, $\epsilon \sim N(0, 1)$ and $\sigma_1 = \sigma_2 = \sigma_3 \sim 0.1$
- The group G4, $x_{16}, ..., x_{40}$, are obtained by a random uniform distribution with values between -0.5 and 0.5.
- The output is obtained according with the following function: $y = -sign(X\beta + \epsilon)$ where β is given by:

$$\beta = (\underbrace{1, 1, ..., 1}_{15}, \underbrace{0, 0, ..., 0}_{25}).$$

The figure below shows the matrix correlation for the synthetic data set.



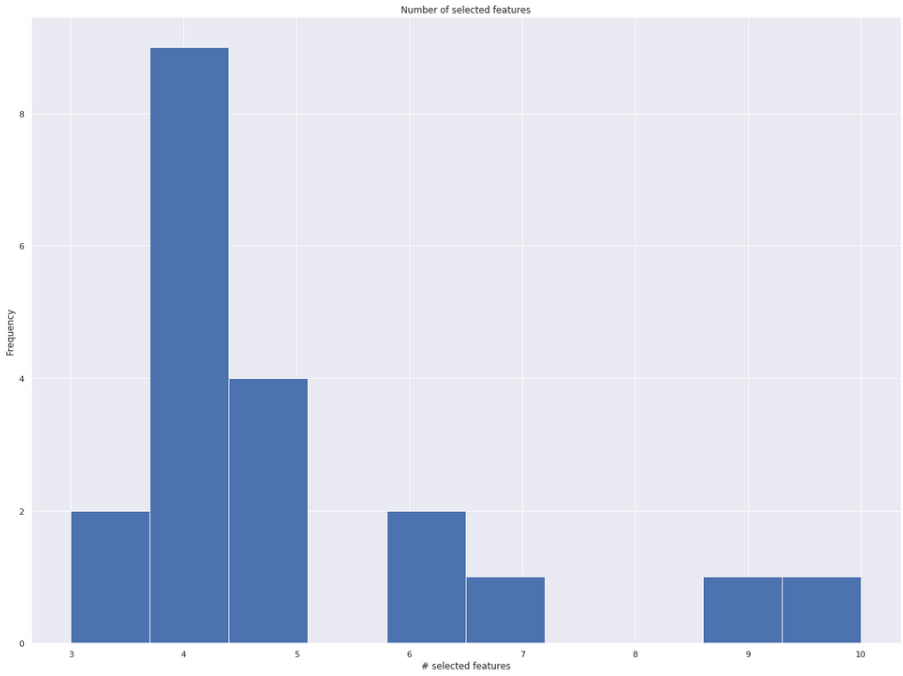
It can be seen that the groups G1 $(X_1, ..., X_5)$, G2$(X_6, ..., X_{10})$ and, G3 $(X_{11}, ..., X_{15})$ show a strong correlation between variables. Also, those are the groups with the higher correlation with the output. The dataset has a mean of -0.001 and the standard deviation has a value of 0.28734.

Then, I run the stage I of the method proposed in [1] by setting $\mu = 0$. I repeat the experiment 20 times over the same dataset. The table below shows the results of the experiment.

	Number of selected features	Selected features	Score	Tau
0	5	([0, 5, 6, 7, 10],)	0.446797	0.1
1	4	([0, 1, 5, 10],)	0.439333	0.1
2	4	([0, 5, 6, 10],)	0.461569	0.1
3	6	([0, 5, 6, 10, 11, 12],)	0.431389	0.1
4	10	([0, 1, 2, 5, 6, 7, 8, 10, 11, 12],)	0.414412	0.1
5	4	([0, 5, 6, 10],)	0.416728	0.1
6	4	([0, 1, 5, 10],)	0.457604	0.1
7	5	([0, 1, 5, 6, 10],)	0.485935	0.1
8	3	([0, 5, 10],)	0.442333	0.1
9	4	([0, 5, 10, 11],)	0.413106	0.1
10	4	([0, 5, 10, 11],)	0.444061	0.1
11	9	([0, 1, 2, 5, 6, 7, 8, 10, 12],)	0.436401	0.1
12	6	([0, 1, 5, 6, 10, 11],)	0.414846	0.1
13	5	([0, 5, 6, 10, 11],)	0.450855	0.1
14	7	([0, 1, 5, 6, 7, 10, 11],)	0.406905	0.1
15	5	([0, 5, 7, 10, 11],)	0.450330	0.1
16	4	([0, 5, 10, 11],)	0.421060	0.1
17	4	([0, 5, 10, 11],)	0.416092	0.1
18	4	([0, 5, 6, 10],)	0.405390	0.1
19	3	([0, 5, 10],)	0.441126	0.1

After I run the stage I the method is never selecting redundant variables (Group 4). However, most of the time the method select more than one variable of the same Group. Also, sometimes the method leave out all variables that belong to a relevant group.

The figure below shows a histogram which represent the number of variables of the obtained model over 20 trials.



Then, I run the stage II by setting $\mu = 1000 * \tau$. The table below shows the results.

	Number of variables selected	Predictors selected	Score	Tau
0	15	([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13...	0.000707	0.1

The model selects 15 variables which is the maximum number of relevant features.

2.2.2 Synthetic dataset with normal distribution

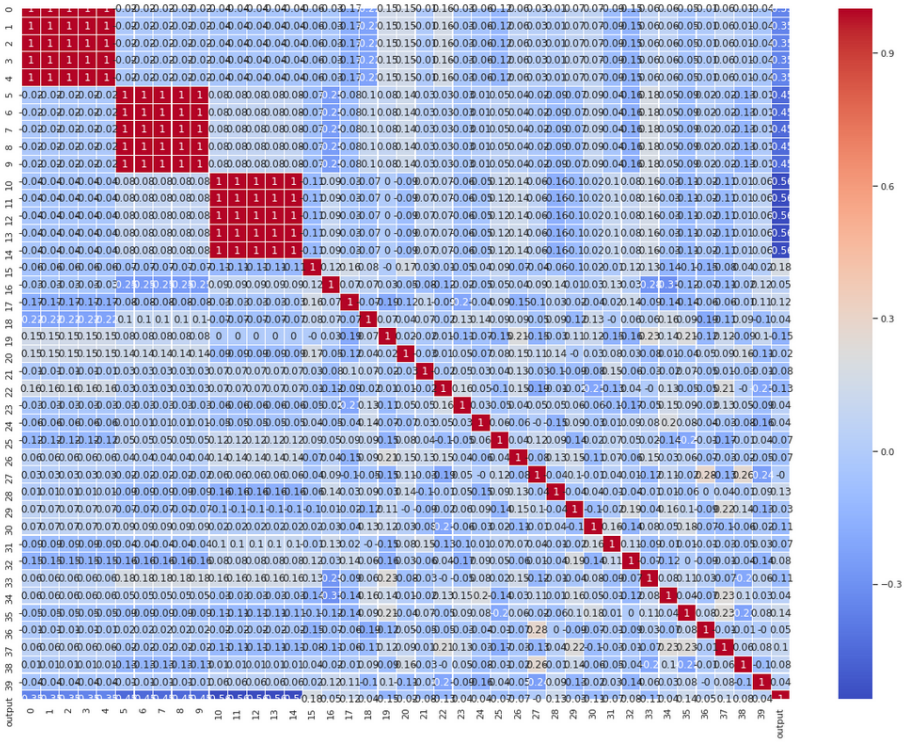
The second synthetic dataset is similar to the first dataset and consists in n=100 samples. To each sample there is associated a 40 dimensional vector x_i given by the following rules:

- There will be four groups G1, G2, G3, G4
- x_1, x_6, x_{11} are obtained by a normal uniform distribution with mean=0 and std=1.
- The group G1 is given by: $x_{1+i} = s_i * x_i + \sigma_1 * \epsilon$ for i = 1,...,4
- The group G2 is given by: $x_{6+i} = s_i * x_i + \sigma_2 * \epsilon$ for i = 1,...,4
- The group G3 is given by: $x_{11+i} = s_i * x_i + \sigma_3 * \epsilon$ for i = 1,...,4

- Where $s_i = 1, i = 1, ..., 4$ and, $\epsilon \sim N(0, 1)$ and $\sigma_1 = \sigma_2 = \sigma_3 \sim 0.1$
- The group G4, $x_{16}, ..., x_{40}$, are obtained by a normal uniform distribution with mean=0 and std=1.
- The output is obtained according with the following function: $y = -sign(X\beta + \epsilon)$ where β is given by:

$$\beta = (\underbrace{1, 1, ..., 1}_{15}, \underbrace{0, 0, ..., 0}_{25}).$$

The figure below shows the matrix correlation.



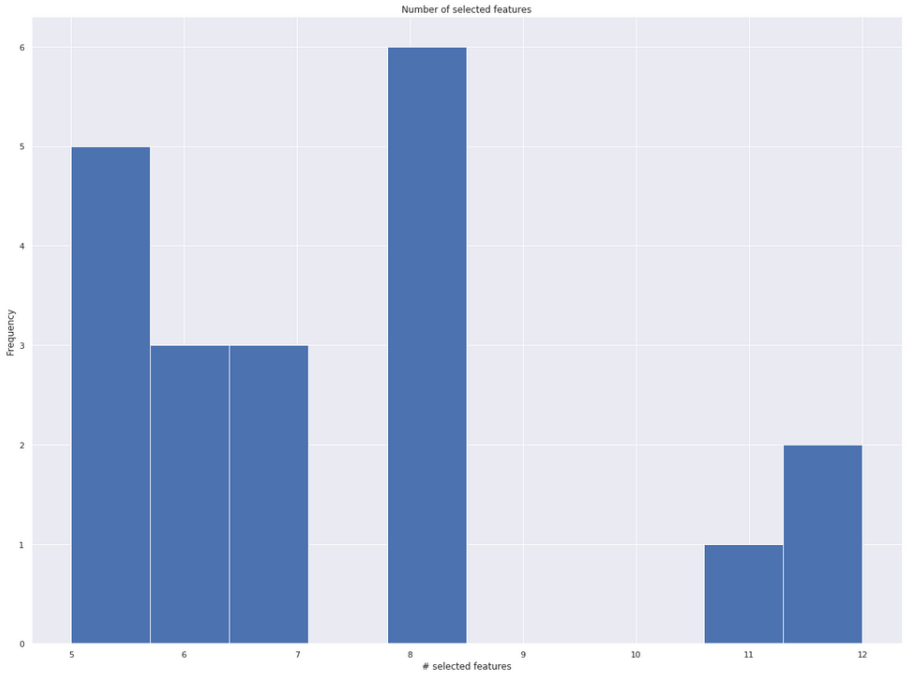
As in the first synthetic dataset groups G1 $(X_1, ..., X_5)$, G2$(X_6, ..., X_{10})$ and, G3 $(X_{11}, ..., X_{15})$ show a strong correlation between variables. Also, those are the groups with the higher correlation with the output.

After run the stage I with $\mu = 0$ for 20 times the result are the following:

	Number of selected features	Selected features	Score	Tau
0	8	([0, 5, 6, 7, 10, 11, 12, 19],)	0.661787	0.1
1	8	([0, 1, 2, 3, 5, 10, 15, 29],)	0.634516	0.1
2	12	([0, 1, 2, 5, 6, 9, 10, 11, 12, 13, 19, 21],)	0.650305	0.1
3	7	([0, 1, 5, 6, 10, 19, 31],)	0.631372	0.1
4	8	([0, 5, 6, 10, 19, 29, 35, 37],)	0.652589	0.1
5	5	([0, 5, 10, 15, 29],)	0.609229	0.1
6	6	([0, 1, 2, 5, 10, 11],)	0.673918	0.1
7	8	([0, 5, 6, 7, 8, 9, 10, 29],)	0.671999	0.1
8	6	([0, 1, 2, 5, 10, 11],)	0.627931	0.1
9	11	([0, 5, 6, 7, 8, 9, 10, 11, 13, 29, 31],)	0.664924	0.1
10	7	([0, 1, 2, 5, 6, 10, 14],)	0.597314	0.1
11	5	([0, 5, 6, 10, 19],)	0.661885	0.1
12	5	([0, 5, 6, 10, 19],)	0.695415	0.1
13	8	([0, 5, 6, 7, 10, 11, 12, 13],)	0.731913	0.1
14	5	([0, 5, 6, 10, 39],)	0.666946	0.1
15	5	([0, 5, 10, 29, 36],)	0.643005	0.1
16	6	([0, 1, 5, 10, 19, 39],)	0.648799	0.1
17	8	([0, 1, 5, 10, 11, 19, 29, 39],)	0.664735	0.1
18	12	([0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 29, 37],)	0.661096	0.1
19	7	([0, 1, 2, 5, 6, 10, 11],)	0.678842	0.1

After the stage I the method selects a large amount of variables of the relevant groups. Also, the method selects some variables of the Group 4 (the irrelevant group).

The figure below shows a histogram which represent the number of variables of the obtained model over 20 trials.



Then, I run the stage II by setting $\mu = 1000 * \tau$. The table below shows the results.

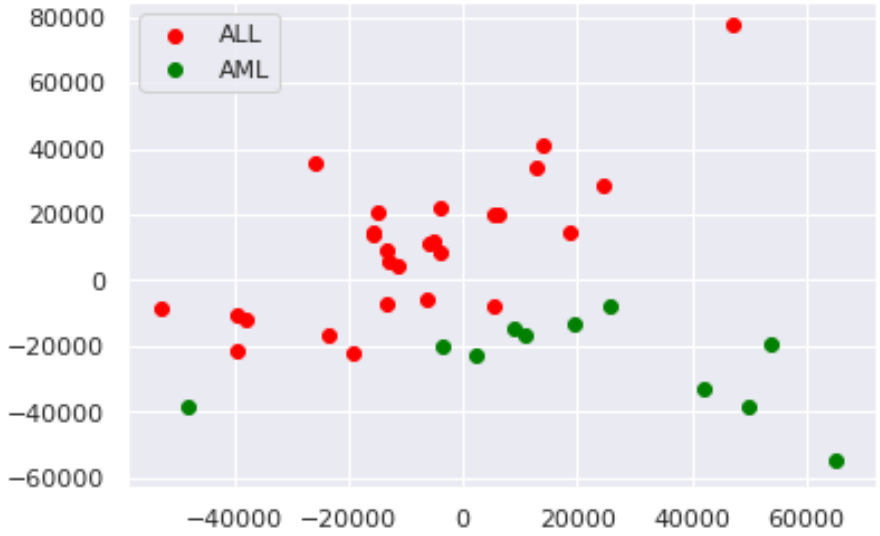
	Number of variables selected	Predictors selected	Score	Tau
0	23	([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13...	0.026232	0.1

It can be seen that the model selects all the relevant features and, also some features in the group 4.

2.3 Real data

In this section, I implemented the visualization method which was proposed in reference [2]. I used the Golub Dataset. The dataset contains data for two different types of cancer, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL).

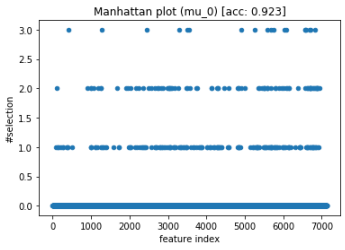
The dataset contains 38 samples and 7071 relevant features. I used PCA to visualize the data.

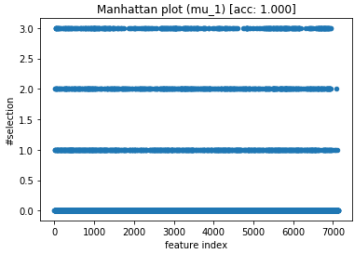


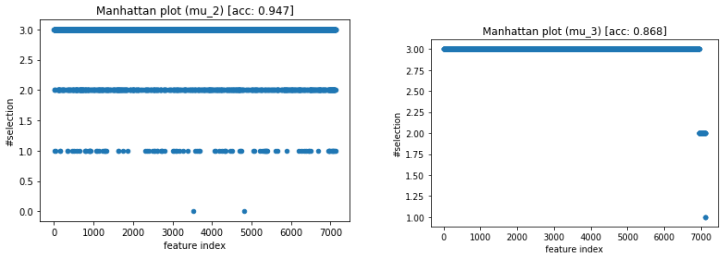
Then, I applied the two stage method setting 4 different values to μ keeping track of the all variables selected and the score on each iteration. The values for μ were $\mu_0 = 1e4, \mu_1 = 1e5, \mu_2 = 1e7, \mu_3 = 1e8$. The table below shows a sample of the lists obtained.

	Selection frequency_mu_0	Selection frequency_mu_1	Selection frequency_mu_2	Selection frequency_mu_3
M97016_s_at	0.0	0.0	3.0	3
HG2797-HT2905_s_at	0.0	0.0	3.0	3
U33203_s_at	0.0	0.0	3.0	3
X95239_at	0.0	0.0	3.0	3
U34301_r_at	0.0	0.0	3.0	3
U25138_at	0.0	0.0	3.0	3
U46461_at	0.0	0.0	3.0	3
M34516_at	1.0	3.0	3.0	3
HG3998-HT4268_at	0.0	0.0	3.0	3
D87452_at	0.0	0.0	3.0	3

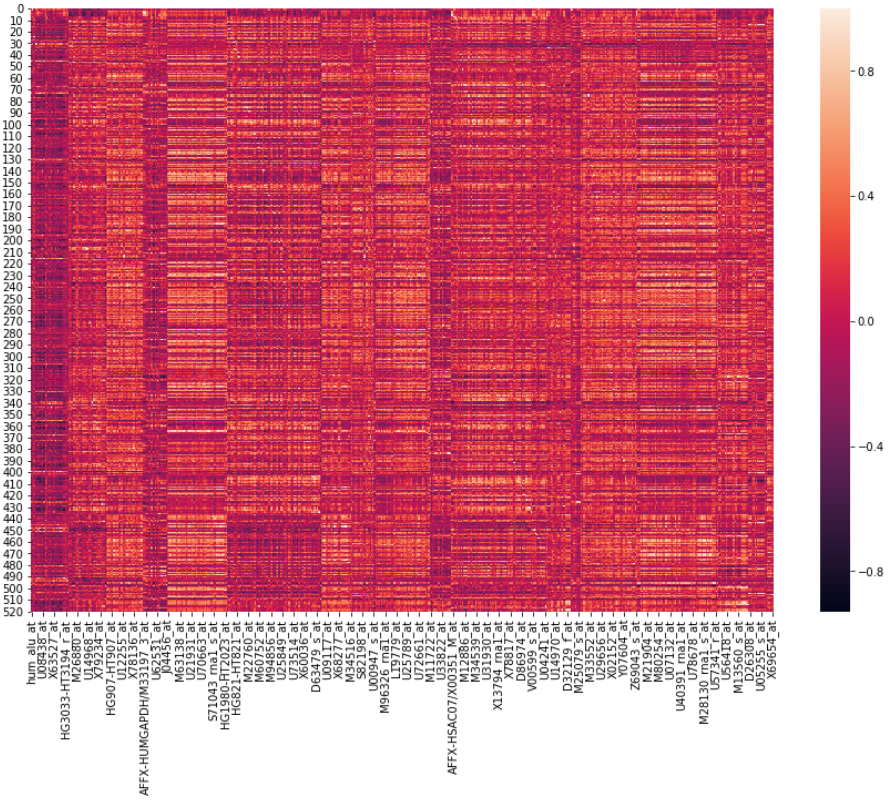
Each column in the table represents the frequency of selection of each variable for a different value of μ. To visualize the lists I plot a Manhattan chart for each value of μ.







Taking the features in μ_0 with a frequency equal or greater than 3 as the minimal list and taking the features in μ_1 with a frequency equal or greater than 3 as the maximal list. I compute the Pearson correlation between lists and get groups of features correlated. To visualize this I plot the matrix correlation between groups.



The final result shows that there are 521 relevant features in the dataset. The heatmap allows to identify different groups of correlated features in the data set.

3 References

1. C. De Mol, S. Mosci, M. Traskine, and A. Verri. A regularized method for selecting nested groups of relevant genes from microarray data. Technical report, DISI, 2007.

2. Mosci S, Barla A, Verri A, Rosasco L. Finding structured gene signatures. IEEE Proceedings BIBM 2008. 2008. p. 8.

3. T. Hastie, R. Tibshirani, M. Wainwright. Statistical Learning with Sparsity: The Lasso and Generalizations. Chapman and Hall CRC 2015