UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Paris 1 Panthéon-Sorbonne University

Economics School of Sorbonne
Econometrics and Statistics Master's Degree

January 2023

Statistical Learning vs Machine Learning

Students :
Cécile Huang
Yoan Jsem
Alice Liu

Under the supervision of :
Philippe De Peretti

Contents

Abstract. In this paper, we first define what is *Machine Learning* and *Statistical Learning*. In a second time we survey different kinds of procedures (*inferential* and *non-inferential*) on different types of data (*Generalized Linear Models, Correlated Data, Correlated Data with Outliers, Outliers only*). For each type of data we've realized a test on simulated data. Each simulated data was built using a *Data Generating Process*, our goal was to find how good were the methods regarding the variable selection problem. At the end we find out which method showed the best results given defined metrics and we test it out on a real dataset.

1 Introduction

Data Science is a very large thematic divided in many fields such as *Machine Learning* and *Statistical Learning*. The aim of this Master's thesis is to explain the differences between these two procedures and evaluate their performances on the variables automatic selection. First, let's define what *Machine Learning* and *Statistical Learning*. Both are very close as they can be used on predictive algorithms that learn on a part of a sample data called *train* and predict on the other part called *test*.

However there happens to be differences, on the one hand, *Statistical Learning* is made of *inferential procedures* which is a procedure that uses mathematics, i.e statistical criteria that can be, theoretically, calculated by ourselves although it is very long.
On the other hand, *Machine Learning* is characterised by *non-inferential procedures* which means that the method makes full use of the computing capacity by modifying the weight of the regressors by changing the coefficients. A recent paper from Thomas Becker (2007)[Bec07] also talks about the variable selection problem we're trying to solve. In his paper, the author speaks of various methods to choose variables from a subset. He runs many experience on a simulated dataset to determine which method works best and how are variables kept or dropped. Finally, he tests the results on a real dataset to see which one performs best. Becker concludes that many methods can be used for variable selection. *Stepwise* selection performs well in comparison with *Backward* and *Forward*. If we consider all the methods *(Forward, Backward, Stepwise, Lars, Lasso)*, the author finds that *Lars* and *Lasso* are the best.

Our goal in this paper will be to see if we can reach the same conclusion as Thomas Becker as we will use the same process. Hence we will ask ourselves: *Considering Machine Learning and Statistical Learning criteria, what is the best method whenever we are approaching the variable selection problem ?*
To begin with we'll introduce and define each procedures to have a better understanding of the subject. Then we will perform several tests on simulated data so as to make our own conclusions regarding the question. Finally we'll test our results on real data.

2 Regressors selection

2.1 Inferential procedures

2.1.1 Stepwise Forward

A *Stepwise Forward*[1] is a variable selection method. We begin with an empty model, containing no predictors and add the most significant variables one by one until all the variables are in the model or until the stop criterion is reached. However, once a variable is added we can't remove it anymore. Then we select a single best model by choosing the model with the lowest RSS (Residual Sum of Squares) or the highest R^2. If the models don't have same numbers of variables, we select the best model by using criteria like C_p, AIC, BIC or adjusted R^2.

2.1.2 Stepwise Backward

We begin with a model with all the variables and remove one by one the variables with the least significant estimator until there are no variables left or until the stop criterion. Once a variable is dropped, we can't add it anymore. Then we select the best model like we did for the *Forward Stepwise*, either by looking for the model with the lowest RSS and the highest R^2 if the models are of same size or by criteria (C_p, AIC, BIC or *adjusted* R^2). We can use *Stepwise Backward*[1] when the number of samples is larger than the number of variables.

2.1.3 Hybrid method or Stepwise method

Hybrid method is[1] a combination of the two previous methods. It starts from an empty model and removes and adds variables. At each step (addition or rejection), we recalculate the significance of each variables and drop those that are useless, we repeat this process until we can no longer add and remove variables.

The *Forward* and *Backward Stepwise* create multiple models with different sizes, i.e. with different variables and number of variables.

The goal is to select the best model that will make the least test error, so we need to estimate the test error using cross-validation (like *K-Fold*, *PRESS*) or a statistical criterion (AIC, BIC, C_p, adjusted R^2, T-stat, F-stat).

2.1.4 Criteria of Statistical Learning

The main criteria of *Statistical Learning* (AIC, BIC, $AICC$, SBC, SL(F statistic), $ADJRSQ$ (adjusted R^2), student, C_p,) are based on the validation set approach. The validation set approach is a simple strategy that randomly split the data in two, a training set and a validation set. The model learn with the training set and predict the observations of the validation set. We use criteria of Statistical Learning on the training set.

2.1.4.1 AIC First we have AIC. The AIC [2] (Akaike Information Criterion) is known for selecting the model that will make the best predictions in the future. AIC is a criteria

[1] These three methods (*Forward*, *Backward* and *Stepwise*) were first mentioned by Efroymson (1960)[Efr60]

[2] Hirotugu Akaike, (1974)[Aka74]

that rely on the likelihood of the model, and it is defined as:

$$AIC = -2log(L) + 2(1+p) \tag{1}$$

With L the maximized likelihood and $1+p$ the number of parameters in the model with the intercept. There is a penalty equal to two times the number of parameters. The best model will be the one with the lowest AIC.

The criterion $AICC$ is a correction of AIC and is used for small sample, and it is defined as :

$$AICC = AIC + \frac{2p(p+1)}{n-p-1} \tag{2}$$

With n the number of observations and p the number of parameters.
When the sample size is small, the $AICC$ offers better results than the AIC. AIC and $AICC$ give similar results when the sample size is significantly bigger than the number of parameters in the model.

2.1.4.2 BIC We also have BIC or SBC. The BIC[3] (Bayesian Information Criterion) is known for selecting the "true" model, i.e the most probable model. BIC penalizes more than AIC and it is defined as:

$$BIC = -2log(L) + (1+p)log(n) \tag{3}$$

With n the number of observations. The best model will be the one with the lowest BIC.

2.1.4.3 T-test A *t-test*[4], is a statistical test in which the test statistic follows a Student's t-distribution under the null hypothesis. It compares the mean of two sample groups. We wonder whether the mean of the two groups are statistically significantly different. Also, the *t-test* is used in a linear regression in order to know if the coefficients are significant or not. At the same time, we can check the p-value : if p-value<0.05 then it is significant, otherwise it isn't.

We consider a linear regression :

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n + \varepsilon_n \tag{4}$$

With the hypothesis : $\mathrm{E}(\varepsilon_n) = 0$, $\mathrm{V}(\varepsilon_n) = \sigma^2$, $cov(\varepsilon_n, \varepsilon_m) \neq 0 \ \forall n \neq m$.

Let's take an example in which we would like to know if β_i is significant :
The null hypothesis (H_0) is : $\beta_i = 0$ which means that β_i is not significant.
The alternative hypothesis (H_1) is : $\beta_i \neq 0$ which means that β_i is significant.

We have $\frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}_{\hat{\beta}_i}} \rightarrow \mathcal{T}(N-(p+1))$ degrees of freedom in which N is the number of the individuals and $(p+1)$ the number of regressors including the intercept. As a consequence, under the null hypothesis, the T-stat is : $\frac{\hat{\beta}_i}{\hat{\sigma}_{\hat{\beta}_i}}$. It follows a Student distribution

[3] Gideon Schwarz (1978) [Sch78]
[4] William Sealy Gosset (1908) also known as "Student", [Stu08] | Brigitte Dormont (1989) [Dor+89]

with $N-(p+1)$ degrees of freedom.

At finite distance : we reject the null hypothesis at the threshold of 5% if $|\frac{\hat{\beta}_i}{\hat{\sigma}_{\hat{\beta}_i}}| > q_{1-\frac{5\%}{2}}^{\mathcal{T}(N-(p+1))}$. We don't reject the null hypothesis at the threshold of 5% if $|\frac{\hat{\beta}_i}{\hat{\sigma}_{\hat{\beta}_i}}| < q_{1-\frac{5\%}{2}}^{\mathcal{T}(N-(p+1))}$.

At asymptotic distance : we reject the null hypothesis at the threshold of 5% if $|\frac{\hat{\beta}_i}{\hat{\sigma}_{\hat{\beta}_i}}| > q_{1-\frac{5\%}{2}}^{\mathcal{N}(0,1)}$.

2.1.4.4 F-test An *F-test*[5] is a statistical test in which the test statistic follows a Fisher distribution under the null hypothesis. The *F-test* is used in order to know whether the regressors are globally significant or not. In this case, we consider two models : one which is the restricted model and the other one which is the unrestricted model.

Let's consider a linear regression :

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n + \varepsilon_n \tag{5}$$

With the same hypothesis : $\mathrm{E}(\varepsilon_n) = 0$, $\mathrm{V}(\varepsilon_n) = \sigma^2$, $cov(\varepsilon_n, \varepsilon_m) \neq 0 \; \forall n \neq m$.
The null hypothesis (H_0) is : $\beta_1 = 0, \ldots, \beta_n = 0$, which means that all the regressors are not significant.
The alternative hypothesis (H_1) is : $\exists i \in \{1, \ldots, n\} \mid \beta_i \neq 0$, which means that at least one β_i is significant.

At finite distance, under the null hypothesis, the F-stat is :

$$F = \frac{(RSS_{restricted\ model} - RSS_{unrestricted\ model})/K}{RSS_{unrestricted\ model}/N-(p+1)} \tag{6}$$

It follows a Fisher's distribution with the parameters $(K, N-(p+1))$.
K : number of constraints.
N : number of individuals.
$p+1$: number of regressors including the intercept.

We reject the null hypothesis at the threshold of 5% if the F-stat $> q_{1-5\%}^{\mathcal{F}(K,N-(p+1)}$.
At infinite distance, under the null hypothesis : $KF \xrightarrow{\mathcal{L}} \chi^2$. Then the F-stat is :

$$KF = \frac{(RSS_{restricted\ model} - RSS_{unrestricted\ model})}{RSS_{unrestricted\ model}/N-(p+1)} \tag{7}$$

We reject the null hypothesis at the threshold of 5% if $KF > q_{1-5\%}^{\chi^2(K)}$.

2.1.4.5 R^2 Statistic After selecting a model that suits our needs, we need a way to verify the validity of our model. For this we could use the R^2 statistic[6],it is a statistic that measures the fit of the model. It's the proportion of variance explained. That way we can see the proportion the model we created can actually explain. It's defined as follows:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} \tag{8}$$

[5] Ronald Aylmer Fisher (1926) [Fis92]
[6] Wright Sewall (1921) [Wri21]

First let's define all terms in this formula. TSS stands for *Total Sum of Squares*

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{9}$$

It measures the variability of Y the response. In other words it is the variance of the response before regressing.

RSS stands for *Residual Sum of Squares*

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y})^2 \tag{10}$$

The RSS is calculated after we've performed the regression. It measures the variability unexplained.

That way the R^2 which is $TSS - RSS$ is the variance without the unexplained part, in other words: the variance explained. Divided by the TSS it gives us the proportion of variance explained by the model.

As the R^2 is a proportion, it's a statistic that is between 0 and 1, the closer it is to 1 the more the proportion of variance is explained. On the contrary, an R^2 close to 0 means the proportion of the variance explained by the regression is low. With this statistic as a mean to verify how well is our regression we should know how to interpret its results. But then, what is a good R^2 statistic and how much can we trust this indicator of fit? The R^2 stat must not be trusted too much, in case we find a R^2 too close or equal to 1 we should be rather suspicious than happy. Why is it that the R^2 is so high? In rare cases it happens, but most of the time it's due to a major issue in the data or the model. It could be *over-learning* or maybe the data is not linear (unfit to be modelized by a linear regression) or even worse, our data are *non-stationary*.

Stationarity is a process where the statistical properties of a series remain unchanged over the time. Let x_t be X observations over t the time, a stationary process is defined by:

- $Ex_t = \mu$
- $Vx_t = \sigma^2 \; \forall t$

There is no meaning in working with non-stationary data, the basic hypothesis of econometrics are not met.

Another point, the R^2 increase as we add more variables to the model. In fact, adding more variables means fitting the regression more accurately. That's why, even if just slightly, adding more variables will increase the R^2 mechanically. This is a disadvantage of using R^2 as criteria to measure the fit of a model. If we use the R^2 statistic we will just take the biggest model as it will have a bigger proportion of variance explained. However, we can use it as a mean to see if variables must be dropped, as they won't make the R^2 grow significantly.

2.1.4.6 Adjusted R^2 As we saw previously, the R^2 is a statistic bound by its number of variables, the more the variables, the more the R^2 grow. It is because the RSS decreases due to having more variable to fit the model. Let our series have $p+1$ variables (including intercept). *Adjusted* R^2 Statistic[7] is defined as:

$$Adjusted\ R^2 = 1 - \frac{\frac{RSS}{n-(p+1)-1}}{\frac{TSS}{n-1}} \tag{11}$$

The principle behind the *Adjusted* R^2 is the same as the previous, the larger the stat, the better it is. To prevent using too many variables we add a weight to RSS, if we add too many insignificant variables, it might decrease or increase the final stat because of the number of variables in the denominator: $n-(p+1)-1$. That way, if we add a variable that's non necessary (meaning it barely increase the R^2) it will lead to a decrease of *Adjusted* R^2 because RSS will only slightly increase while the number of variable m increase by at least 1 leading to an *Adjusted* R^2 smaller. In the end, if we use the *Adjusted* R^2 criteria, it should only remain significant variables of the model, dropping those with low effect on the R^2.

2.1.4.7 C_p As an estimate of the *Mean Square Error*, the ${C_p}$[8] is defined as follows:

$$C_p = \frac{1}{n}(RSS + 2(p+1)\widehat{\sigma^2}) \tag{12}$$

with $\widehat{\sigma^2}$ the estimate variance of the error ε. The intuition behind this is adding a penalty to the RSS. By adding $\widehat{\sigma^2}$ we make it so that the penalty increase as the number of predictors increase. Hence, the lower the test error (RSS) the lower will C_p be. With the penalty we can guarantee that too much predictors are added. The C_p is to be minimized and can thus be used as a mean to find a good model that explain the series without using too many variables.

If we divide the data sample between train and test, the validation set approach has two disadvantages while working on datasets. The first one is that the test error can be very different if we cut our data differently, that is to say with different observations for our training and test data we can end up with a whole different test error. The second disadvantage is, by cutting our data the estimation of the test error is less precise, because the statistical methods are less efficient when there are fewer observations.

2.2 Non-inferential procedures

2.2.1 The Lasso

The *Lasso*[9] is a shrinkage method like *Ridge* - which will be describe in the next section - with a little difference on the penalty. Indeed, in *Ridge regression*, one disadvantage is that it keeps all p predictors in the final model, as the penalty shrink the coefficients towards zero, but not equal to zero. An alternative to this is called "the Lasso" and it

[7] Introduced by Mordecai Ezekiel (1929) [Eze29]

[8] Introduced by C.L. Mallows (1973)[Mal00]

[9] Robert Tibshirani(1996), [Tib96]

minimizes :

$$\begin{aligned}\hat{\beta}_\lambda^{Lasso} &= \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| \\ &= RSS + \lambda\sum_{j=1}^{p}|\beta_j|\end{aligned} \tag{13}$$

We can see that the penalty in the *Ridge regression* $\lambda\sum_{j=1}^{p}\beta_j^2$ is replaced by $\lambda\sum_{j=1}^{p}|\beta_j|$ in the *Lasso*. The penalty used is called ℓ_1 penalty. Like the *Ridge* regression, the *Lasso* shrinks the estimated coefficients towards zero but the difference is that some coefficients can be forced to be equal to zero. As a result, the *Lasso* will select a subset of the variables, like *AIC* or *BIC*. In this case, the *Lasso's* models are easier to interpret than the *Ridge* one. To select a good value of λ, we need to use a *K-Fold* validation. If $\lambda = 0$, the *Lasso* gives the least squares fit, whereas if λ is large enough, the *Lasso* gives the null model which means that all the estimated coefficients equal zero.

2.2.2 Elastic Net

First, to define what is *Elastic Net*, we need to define the *Ridge regression*.

The Ridge Regression[10] is a mean to shrink coefficients of a regression toward zero. There's no need to use all regressors, in fact giving less significativity to some of them can reduce the variance. Using Ridge Regression we are still minimizing the *RSS* like we have been doing with least squares. Though, unlike least squares we introduce a penalty. With least squares we wanted to estimate $\beta_0 \dots \beta_p$ which minimize:

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y})^2 = \sum_{i=1}^{n}(y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_i j))^2 \tag{14}$$

Here with *Ridge Regression* we want to estimate $\hat{\beta}^{\mathcal{R}}$ which minimize:

$$\sum_{i=1}^{n}(y_i - \hat{y})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{15}$$

Here $\lambda \geq 0$ act as a *tuning parameter*, we will determinate this parameter by cross-validation detailed further in this subsection. Like the least square method we want *RSS* to remain smaller as possible. However this time we carry a penalty ($\lambda\sum_{j=1}^{p}\beta_j^2$), also known as a ℓ_2 penalty, it is a shrinkage penalty. By introducing this penalty we are forcing estimates of β_j towards zero whenever $\beta_1 \dots \beta_p$ are close to zero. Note that we are not taking β_0 into our shrinkage penalty, we only need β related to the response Y. Hence we conserve β_0 the intercept as it is only the measure of the mean of Y. Now, λ is a positive parameter which controls the shrinkage penalty, it regulates how much penalty we want to apply. The closer λ is to 0, the more the Ridge Regression will look like the ordinary least squares. When $\lambda = 0$ Ridge Regression will return least squares estimates. Now, as λ grows the shrinkage penalty will grow further and further making it so the regression

[10] Arthur E. Hoerl and Robert W. Kennard (Feb., 1970)[HK70]

coefficient estimate approaches zero. Note that for each λ there are a different set of $\hat{\beta}^R_\lambda$. Ridge Regression is better than least squares whenever the estimates have a big variance. It happens when number of observations n is not far from number of variables p, a slight change of an observation (*e.g*: measurement error) can highly impact the estimates of least squares.

However, as the *Ridge* isn't a variable selection method we won't be using it. We will be using the *Elastic Net*[11] as a mean to compromise between the *Ridge* and the *Lasso* regression. Like the *Lasso*, *Elastic Net* can select predictors, and like the *Ridge*, it can shrink coefficients of same correlation together. It is defined as follows:

$$\begin{aligned}\hat{\beta}^{ElasticNet} &= \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}(\alpha|\beta_j| + (1-\alpha)\beta_j^2) \\ &= RSS + \lambda\sum_{j=1}^{p}(\alpha|\beta_j| + (1-\alpha)\beta_j^2) \quad \forall\alpha\in[0,1]\end{aligned} \tag{16}$$

As the Lasso's penalty has a limitation regarding the correlated variables, the Elastic net allows to keep variables thanks to the quadratic penality. The first penalty term forces a sparsity over the coefficients while the second term encourages predictors with high correlation to be averaged. Elastic net can be used in many situations of Linear regression or classification, it's flexible to a low number of observations (cases when $p \geq n$).

2.2.3 Least Angle Regression

Least Angle Regression (LARS)[12] is a regression algorithm - that can be adapted for the *Lasso* - for a dataset that has lots of variables and observations. Contrary to the Lasso, this has not any hyperparamenter to choose. This is similar to a *Forward Stepwise* method because at first, all the variables coefficients are equal to zero. At each step, *LARS* selects the regressor x_i which is the most correlated to residuals. Then, it keeps its coefficient β_i and increase it in the direction which is the most correlated with the residuals, until another regressor x_j has the same or more correlation as x_i with the current residuals. Now we have β_i and β_j, we increase them in the direction of $(X'X)^{-1}X'\varepsilon$ (least angle). It keeps going until all the predictors are in the model.

2.2.4 Cross-Validation: criteria of Machine Learning

The main criteria of *Machine Learning* (*K-fold* (*CV*), *PRESS*) are based on cross-validation . An alternative to the validation set approach is the cross validation which is based on several cuts with a different validation and training sample at each step.

2.2.4.1 K-fold The *K-fold* is a cross validation method, and it consists in dividing the data into k groups of same size. The first group of data will be our validation set and the rest, i.e. the other $k-1$ groups, will be our training data. We repeat this procedure

[11] Hui Zou, Trevor Hastie (2005) [ZH05]

[12] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani (2004) [Efr+04]

k times and each time the validation group changes. For each repeated procedure, we obtain the mean square error (MSE), which is the cross-validation error. The *k-fold* is very used in *LASSO* and *Ridge* to select the best λ. By selecting the best λ, we select the best model which should have the lowest test error.

2.2.4.2 Leave-One-out Let's assume that our data are such as:

$$(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n) \tag{17}$$

The $PRESS$(Predicted Residual Error Sum of Squares) criterion divides our data in two groups. The first group is made of one observation (X_1, Y_1), it will be the validation set, the rest of the observations $\{(X_2, Y_2), \dots, (X_n, Y_n)\}$ is our training set. Then, our second observation (X_2, Y_2) becomes our validation set and the rest $\{(X_1, Y_1), (X_3, Y_3), \dots, (X_n, Y_n)\}$ represents our training data, and so on until (X_n, Y_n) becomes our validation data. The $PRESS$ criterion is a special case of the *K-fold* where $k = n$, with n the total number of observations. The $PRESS$ method allows us like the *K-fold* to select the best λ and thus the best model with the least error of test.

3 Test on simulated data

In the following section we will further approach the problem of variable selection by testing our methods on simulated data. In fact, some models might work better than others depending on the type of series. Here we will test various methods and see for ourselves what works best. The aim of this section is to determinate the best method to use for each type of series.

How do we intend to proceed ? To find the best method we need to define what does it mean to find the "best" method of variables selection. The best method could be defined as one that suits best the series' profile. To take an example, digging a hole works best with the suitable tool: a shovel. However using a shovel to carry out water from a well would be inefficient. Here we want to proceed the same way: find a suitable tool for each activity. The different tools would be the methods we saw in the first section (AIC ,BIC ,Cp , SL[13]...) and the activity would be the different types of series (normal, correlated, correlated with outliers...). Thus, a good method will find the "true" model with a high success rate.

Firstly we will build a randomly created model as a reference for this section.

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_5 X_5 + \varepsilon \tag{18}$$

The model will act as the true model. With $n = 100$ observations.
Secondly we will test a method to see if, among the 50 variables it can figure out the 5 true variables of the model. In other words the goal for each method will be to find the true model among a set of variables $\{X_1, X_2, \dots, X_{50}\}$.

Let's say the tested method dropped and kept different variables, we want to find: how

[13] We will be using a 5% threshold as it has the best results among the different threshold tested.

often is the true model found ? To measure the success rate of each method we will use different metrics:

- Perfect Fitting (PF): when the used method find the true model without forgetting or adding variables
- Overfitting(OF): when the used method find the true model but adds at least one more variables that doesn't belong to the true model
- Underfitting (UF): when the used method find a model that has forgotten variables
- Semi Failure (SF): when the used method find a model that has forgotten variables from the true model and adds variables that aren't from the true model
- Total Failure (TF): when the used method fails to get any variables from the true model

Furthermore, we will use another metric: the rate at which variables of the true model are discovered. For the true model $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_5 X_5 + \varepsilon$ we will figure the apparition rate of $\{X_1, X_2, \ldots, X_5\}$ depending on the method used.

In this section we will first introduce how do we create theoretical set of data (normal, correlated, correlated with outliers...) by doing a *Data Generating Process*. Then, for each type of series we will try to figure out what method works best among the different tools we have at our disposal.

3.1 Data Generating Process

First and before all, we need to create set of data to work on. We can't test method without a set paradigm, thus we will do a Data Generating Process (DGP). Doing a DGP allows us to create a model sample to test methods on.

3.1.1 DGP for Generalized Linear Models

To create a *Generalized Linear Model*, the model must follows the basic hypothesis of econometrics, so the code to generate it will be as follows:

```
proc iml;
N = 100;
Mean =j(50,1,0);
Cov =I(50);
x = RandNormal( N, Mean, Cov );
eps=normal(j(N,1,0))*0.25;
beta={1,2,0.7,-1.2,3};
y=X[,1:5]*beta+eps;
rnames='y'//('X1':'X50')`;
d=y||X;
```

Here we've created a single *Generalized Linear Model* with $n = 100$ observations. The true model consisting of the response Y and the variables $\{X_1, X_2, \ldots, X_5\}$ with $\beta =$

$\{1, 2, 0.7, -1.2, 3\}$ as coefficients of the true model and ε as residuals. To complete the *DGP* we just need to do a loop, and repeat it as many times as needed. In this section we will set the loop up to 10000 so that we can have significant conclusion over our tests.

3.1.2 DGP for Correlated Variables

Now if we consider correlated models, meaning *Generalized Linear Model* with correlated variables. We must use *Iman Conover*[IC82] algorithm so as to produce a set model that has correlation. It's a transformation which, given a set of variables ($\{X_1, X_2, \ldots, X_{50}\}$) and a covariance matrix can produce correlated variables. In the previous case we didn't want any correlation among variables so we've set the cross-covariance matrix as $cov = I(50)$, meaning no correlation since it's identity matrix. In this case we want $\{X_1, X_2, \ldots, X_5\}$ to be correlated so we will proceed as follows(please refer to annex for the entire code).

First, we will produce correlated models. Using the SAS command *toeplitz*[14] we initialize a cross-covariance matrix. Then, with the *Iman Conover* transformation we can produce a model with $\{X_1, X_2, \ldots, X_5\}$ correlated ($Xcor$) and $\{X_6, X_7, \ldots, X_{50}\}$ unrelated, linear model ($Xlin$). All that's left is to "merge" $Xcor$ and $Xlin$ as one X ready to use. Rest of the *DGP* is as the previous one, building the true model, the model is set to be tested on methods !

3.1.3 DGP for a model with Outliers/Correlation and Outliers

To produce a model with outliers we need to set some observations to be abnormal. We will use another distribution to pick on to create this(refer to the annex).

While generating a model with only outliers we don't need to use the *Iman Conover* transformation. We will use another normal distribution to pick observation in and replace them randomly. We decided to set the outliers' apparition rate up to 5%, the rest of the *DGP* is as always, creating the true model with the response Y using variables $\{X_1, X_2, \ldots, X_5\}$, betas and epsilons.

Now if we need to make a model of correlated variables with outliers, we just need to fuse method from 3.1.2 and the previous code, please see the annex code for the part related to Correlated data with outliers.

3.1.4 Metrics

Earlier we have introduced various metrics that we will use to judge how good a method is, the code to measure it is as follows:

```
proc iml;
PF=0;/*perfect fitting*/
UF=0;/*Underfitting */
OF=0;/*Overfitting*/
SF=0;/*semi failure*/
```

[14] Toeplits named after Otto toeplits is a matrix whose diagonal is constant and allows us to define the target correlation matrix necessary to the algorithm of *Iman Conover*

```
TF=0;/*total failure*/
X1=0;X2=0;X3=0;X4=0;X5=0;
```

First we initialize the variables, to stock the number of apparition. Next is the usual code with the *DGP* and the selection model (we will explain it later on).

```
Xglm=name[2:nrow(name)-1];
Xtrue=('X1':'X5');
D1=setdif(Xglm,Xtrue);
D2=union(Xtrue,Xglm);
D3=setdif(Xtrue,Xglm);
if ncol (D2)=ncol(Xtrue)& ncol(Xtrue)=nrow(Xglm)then /*PERFECT FIT*/
PF=PF+1;else PF=PF;
if ncol(D2)=ncol(Xtrue) & nrow(Xglm)< ncol(Xtrue) then /* Underfitting */
UF=UF+1;else UF=UF;
if ncol(D2)>ncol(Xtrue) & ncol(D3)<5 & ncol(D3)>0 then SF=SF+1;/*Semi Failure*/
else SF=SF;
if ncol(D2)>ncol(Xtrue) & ncol(D3)=0 then OF=OF+1;/*Overfitting*/
else OF=OF;
if ncol(D2)>ncol(Xtrue) & ncol(D3)=5 then TF=TF+1;/*Total Fitting*/
else TF=TF;
if element('X1',Xglm)=1 then X1=X1+1;
else X1=X1;
if element('X2',Xglm)=1 then X2=X2+1;
else X2=X2;
if element('X3',Xglm)=1 then X3=X3+1;
else X3=X3;
if element('X4',Xglm)=1 then X4=X4+1;
else X4=X4;
if element('X5',Xglm)=1 then X5=X5+1;
else X5=X5;
end;
TXPF=(PF/(rep-1))*100;
TXUF=(UF/(rep-1))*100;
TXOF=(OF/(rep-1))*100;
TXSF=(SF/(rep-1))*100;
TXTF=(TF/(rep-1))*100;
TX1=(X1/(rep-1))*100;
TX2=(X2/(rep-1))*100;
TX3=(X3/(rep-1))*100;
TX4=(X4/(rep-1))*100;
TX5=(X5/(rep-1))*100;
```

Using the SAS functions *setdif*, *union* and *element* we can build our metrics as defined in the introduction. *Xglm* stands for the variables generated through the selection process by using the *proc glmselect* command. *Xtrue* stands for the true variables of our model, the reference sample.

In the end, the final code will be composed of, the *DGP*, the selection process and the metrics. For the standard case, with neither correlated variables nor outliers the code will look like this:

```
proc iml;
PF=0;
UF=0;
OF=0;
SF=0;
TF=0;
X1=0;
X2=0;
X3=0;
X4=0;
X5=0;

do rep=1 to 10000;
N = 100;
Mean =j(50,1,0);
Cov =I(50);
x = RandNormal( N, Mean, Cov );
eps=normal(j(N,1,0))*0.25;
beta={1,2,0.7,-1.2,3};
y=X[,1:5]*beta+eps;
rnames='y'//('X1':'X50')`;
d=y||X;
create tableau from d[colname=rnames];
append from d;
close tableau;
submit;
proc glmselect data = tableau outdesign=names noprint ;
model Y= X1-X50 / selection=Lar(choose=CV);
run;
proc contents data=names out=toto noprint;
run;
endsubmit;
use toto; read all;
close toto;
Xglm=name[2:nrow(name)-1];
Xtrue=('X1':'X5');
D1=setdif(Xglm,Xtrue);
D2=union(Xtrue,Xglm);
D3=setdif(Xtrue,Xglm);

if ncol (D2)=ncol(Xtrue)& ncol(Xtrue)=nrow(Xglm)then PF=PF+1;
else PF=PF;

if ncol(D2)=ncol(Xtrue) & nrow(Xglm)< ncol(Xtrue) thenUF=UF+1;
else UF=UF;
```

```
if ncol(D2)>ncol(Xtrue) & ncol(D3)<5 & ncol(D3)>0 then SF=SF+1;
else SF=SF;

if ncol(D2)>ncol(Xtrue) & ncol(D3)=0 then OF=OF+1;
else OF=OF;

if ncol(D2)>ncol(Xtrue) & ncol(D3)=5 then TF=TF+1;
else TF=TF;

if element('X1',Xglm)=1 then X1=X1+1;
else X1=X1;

if element('X2',Xglm)=1 then X2=X2+1;
else X2=X2;

if element('X3',Xglm)=1 then X3=X3+1;
else X3=X3;

if element('X4',Xglm)=1 then X4=X4+1;
else X4=X4;

if element('X5',Xglm)=1 then X5=X5+1;
else X5=X5;
end;

TXPF=(PF/(rep-1))*100;
TXUF=(UF/(rep-1))*100;
TXOF=(OF/(rep-1))*100;
TXSF=(SF/(rep-1))*100;
TXTF=(TF/(rep-1))*100;
TX1=(X1/(rep-1))*100;
TX2=(X2/(rep-1))*100;
TX3=(X3/(rep-1))*100;
TX4=(X4/(rep-1))*100;
TX5=(X5/(rep-1))*100;
print TXPF TXUF TXOF TXSF TXTF TX1 TX2 TX3 TX4 TX5;
```

Doing the other cases will just need to replace the *DGP* like instructed in previous subsections. You can see full codes in annex.

3.2 Generalized Linear Model

In this subsection we look for the best selection method in the case of perfect data without correlation between explanatory variables and with a model that follows a normal distribution. To do this we need to create our simulated reference data as described in section 3.1.1 *DGP* for generalized linear models with an identity matrix to force this absence of correlation between explanatory variables.

In a second step, we went through the *glmselect* procedure to test the selection models as well as the stopping criteria. We used two options in *glmselect*, "Selection" to choose the selection method ($Forward$, $Backward$, $Stepwise$, $Lasso$, $Lars$, $ElasticNet$) and "Choose" so that the procedure chooses the model according to the stopping criterion (AIC, BIC, $AICC$, SBC, SL, C_p, $PRESS$, CV, *Adjusted* R^2).

For example, if we set AIC for “Choose”, then the model kept will be the one with the lowest AIC.

Let’s have a global look on the average of every method : from the *Generalized Linear Model* if we consider all the criteria, we could see that the $Forward$ method has a high *Overfitting* rate, about 83%.

Method	PF	UF	OF	SF	TF
Forward	16,13	0	83,86	0	0
Backward	2,02	0	97,98	0	0
Stepwise	16,24	0	83,76	0	0
Elastic Net	23,13	5,90	70,21	0,11	0,65
Lars	28,35	12,22	28,77	0	0,65
Lasso	28,01	12,19	59,19	0,02	0,59

Like the $Forward$ method, the $Backward$ method does mostly *Overfitting*, the rate is about 97% for 2% of *Perfect Fitting*.

The *Stepwise* method, another method that tends to do *Overfitting* with a rate of about 83% and 16% of *Perfect Fitting*.

The $Lars$ method on the other hand finds more *Perfect Fitting* (28%) than the methods seen previously. In addition to *Overfitting* 29% of the time, there can be *Underfitting* with a rate of 12% and *Total Failure* with a rate of 0.6%.

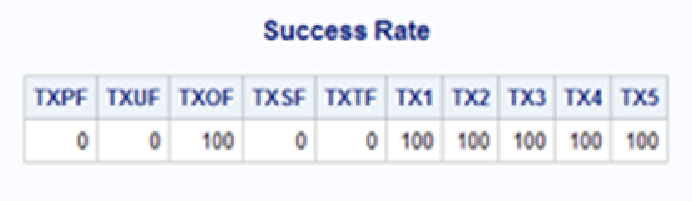
The $Lasso$ method, like the $Lars$ method, finds our reference model 28% of the time, but mostly does *Overfitting* 60% of the time, *Underfitting* 12% of the time, and also does *Total Failures* with similar rate to the $lasso$ case. With the $Lasso$ method we can also find *Semi Failure* situation $0,02\%$ of the time.

Finally, about the *Elastic Net* method, it finds about 23% of the time the reference model, it makes *Overfitting* 70% of the time, *Underfitting* 6% of the time, *total failure* 0.6% of the time and *Semi Failure* 0.11% of the time.

Thus in the case of *Generalized Linear Model* we could see that the worst method is the $Backward$ method with an average rate of *Perfect Fitting* close to 0% ($2,02\%$). In order to have a better look on which is the worst model for generalized linear model, we do a top 3 ranking :

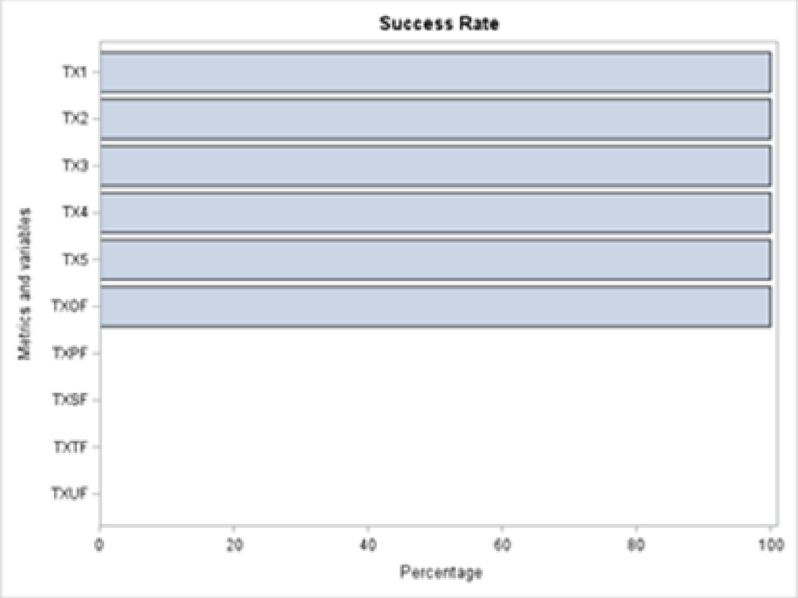
- Worst model : *Backward* method with *Adjusted* R^2 criteria : 0% of *Perfect Fitting*.
- 2nd worst model : *Backward* method with AIC criteria : 0,02% of *Perfect Fitting*.
- 3rd model : *Backward* method with *Press* criteria : 0,03% of $PerfectFitting$.

Here is a graphic of the worst model : *Backward* method with *Adjusted* R^2 criteria :



Success Rate

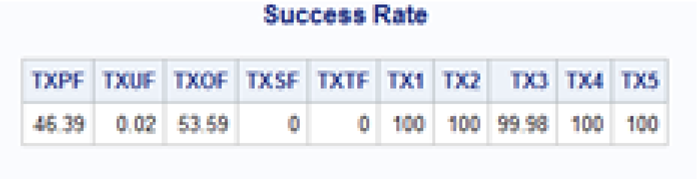
TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
0	0	100	0	0	100	100	100	100	100



The best model to use in the case of *Generalized Linear Model* seems to be the *Lars* and *Lasso* method. Looking by criteria, we see that the best method is the *Lars* method with the stopping criterion $PRESS$. Indeed, we find the reference model 46% of the time. Other methods can be taken into consideration like the *Lasso* with the criterion $PRESS$ and the *Elastic Net* but with the C_p criterion. For the selection methods without inference the $PRESS$ criterion seems to be very efficient. In order to have a better look on which is the best model for *Generalized Linear Model*, we do a top 3 ranking :

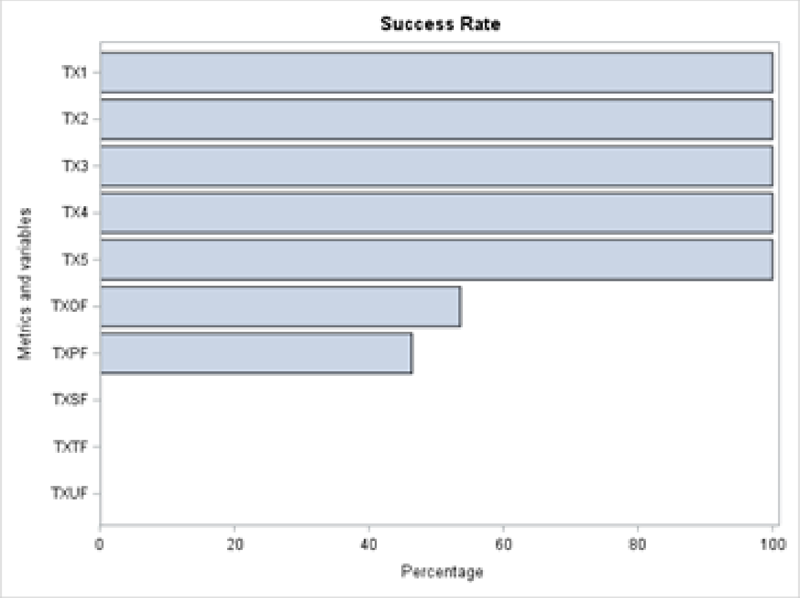
- best model : *Lars* method with $PRESS$ criteria : 46,39% of *Perfect Fitting*.
- 2nd best model : *Lasso* method with $PRESS$ criteria : 46,17% of *Perfect Fitting*.
- 3rd model : *Elastic Net* method with C_p criteria : 37,64% of *Perfect Fitting*.

Here is a graphic of the best model : *Lars* method with $PRESS$ criteria :



Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
46.39	0.02	53.59	0	0	100	100	99.98	100	100



This model has about 100% rate for $\{X_1, X_2, X_3, X_4, X_5\}$, which means that it almost always finds all the variables of our true model. We can explain it because there is a one out of two chance to find *Overfitting*. So having a perfect rate of variables does not mean that it is a *Perfect Fitting*. Moreover, for the variable X_3 the rate is 99.98% which explains the *Underfitting* rate at 0.02%.

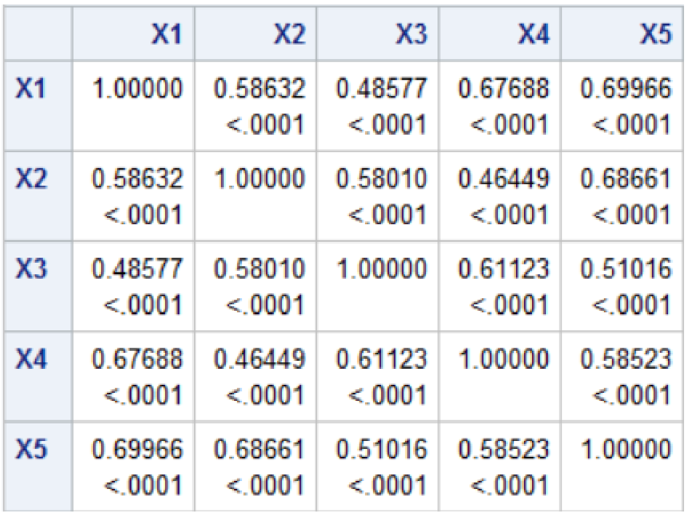
We can notice that in the *Statistical Learning* procedures, what works best is the *Stepwise* method. Indeed *Stepwise* is better than *Backward* and *Forward*. On average, we find *Perfect Fitting* rate is of 16.24% against 16.13% for *Forward* and 2.02% for *Backward*. This result matches what Thomas Becker found. When it comes to the *Machine Learning* procedures, the *Lars* and *Lasso* method works the best. Furthermore, *Machine Learning's* methods outperform *Statistical Learning's* when it comes to find the true model.

3.3 Correlated Data

In this part we will see what method works best on correlated data. Correlated Data is a case were the cross-covariance matrix isn't equal to the identity matrix, we are talking about a matrix positive-definite and symmetric (thus invertible). In the theoretical case of econometrics we usually face perfectly independent variables with no correlation, no heteroscedasticity or whatever. But in reality it's unlikely to be the case, and that makes this subsection quite relevant for the analysis of what method is the best.
For this subsection we have set a correlation among $\{X_1, X_2, \ldots, X_5\}$ of Σ:

$$\Sigma = \begin{bmatrix} 1 & 0.6 & 0.5 & 0.7 & 0.7 \\ 0.6 & 1 & 0.6 & 0.7 & 0.7 \\ 0.5 & 0.6 & 1 & 0.6 & 0.5 \\ 0.7 & 0.5 & 0.6 & 1 & 0.6 \\ 0.7 & 0.7 & 0.5 & 0.6 & 1 \end{bmatrix} \tag{19}$$

Using *Iman Conover's* transform, we can see a correlation between $\{X_1, X_2, \ldots, X_5\}$:



	X1	X2	X3	X4	X5
X1	1.00000	0.58632 <.0001	0.48577 <.0001	0.67688 <.0001	0.69966 <.0001
X2	0.58632 <.0001	1.00000	0.58010 <.0001	0.46449 <.0001	0.68661 <.0001
X3	0.48577 <.0001	0.58010 <.0001	1.00000	0.61123 <.0001	0.51016 <.0001
X4	0.67688 <.0001	0.46449 <.0001	0.61123 <.0001	1.00000	0.58523 <.0001
X5	0.69966 <.0001	0.68661 <.0001	0.51016 <.0001	0.58523 <.0001	1.00000

As explained in the previous section we used different selection methods and different choosing criteria to determinate the success rate of each method. The average results for each method are below.

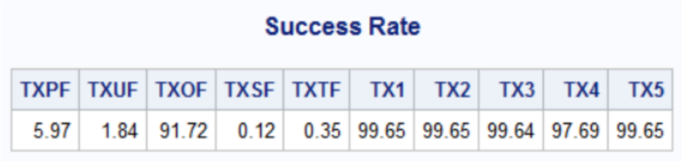
Method	PF	UF	OF	SF	TF
Forward	16,6	0	83,4	0	0
Backward	2,13	0	97,87	0	0
Stepwise	16,45	0	83,55	0	0
Elastic Net	8,61	5,40	84,11	1,48	0,38
Lars	18,1	10,5	61,13	9,93	0,34
Lasso	18,22	10,18	71,04	0,2	0,36

If we judge solely on the selection method, all choosing criteria taken into account: the best average selection method would be the *Lasso* method with a $18,22\%$ of *Perfect Fitting*. Meaning that out of every method, if we select a random choosing criteria we have the average best results using a *Lasso* selection.
If we judge the average methods, the *backward* selection is the worst, with a $2,13\%$ of *Perfect Fitting*. This method fails to find the method most of the time, ending up in *Overfitting* 97% of the time.
The various *Forward*, *Stepwise* and *Lars* selection are rather good since they are all on average near the best average method. Then again, average are often biased by high values and shouldn't be trusted too much so let's have a look into further details.

If we start by the worst methods, then the *Backward* selection is making the entire ranking by itself. The *Perfect Fitting* rate of the *Backward* selection is really low no matter the criteria, it's at best at $13,47\%$ with an average of $2,13\%$. We can already conclude that the *Backward* selection isn't recommended at all as we see the results regarding the methods below.

Rate	AIC	BIC	AICC	SBC	ADJRSQ	Cp	PRESS	K-fold
PF(%)	0	1,97	0,19	13,47	0	0,56	0,09	0,72

Setting the *Backward* selection aside, the worst selection method is the *Elastic Net* with the *Adjusted* R^2 criteria:



Success Rate

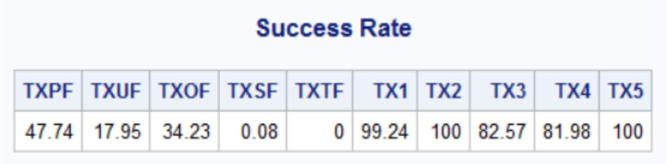
TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
5.97	1.84	91.72	0.12	0.35	99.65	99.65	99.64	97.69	99.65

At this point, regarding correlated data we could make a remark: the less performing criteria usually are *Statistical Learning's* criteria. Regarding each method the criteria which brings less success rate of *Perfect Fitting* would be *BIC*, *AIC* and *Adjusted* R^2.

The 3 best ranking method selection for correlated data are:

- 1. *Lars* method with the choosing criteria *PRESS*: $47,74\%$ of *Perfect Fitting*
- 2. *Lasso* method with the choosing criteria *PRESS*: $46,97\%$ of *Perfect Fitting*
- 3. *Lars* method with the choosing criteria *K-fold*: $25,34\%$ of *Perfect Fitting*

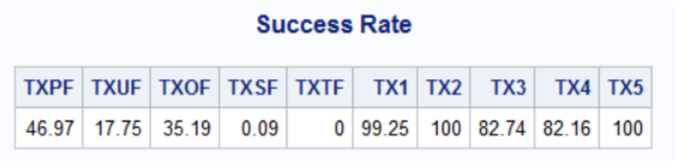
Now we can observe that *Lars* and *Lasso* produce almost the same results with the *PRESS* Validation criteria. The test was repeated a total of 10000 times, so we can say with almost certainty that both *Lars* and *Lasso* methods are equivalent given correlated data.
Aside from the results we can see that Leave-One-Out criteria (*PRESS*) is performing quite well on correlated data and so are *Machine Learning* criteria. Out of 6 selection methods, the *K-fold* and *PRESS* criteria holds the best results in 5 of them.
Here is a graphic for *Lars* with *PRESS*:



Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
47.74	17.95	34.23	0.08	0	99.24	100	82.57	81.98	100

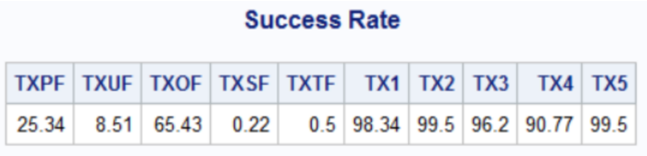
Here for *Lasso* with *PRESS*:



Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
46.97	17.75	35.19	0.09	0	99.25	100	82.74	82.16	100

And here for *Lars* with *K-fold*:



Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
25.34	8.51	65.43	0.22	0.5	98.34	99.5	96.2	90.77	99.5

As we can see, there's a big chance of *Underfitting* with the *Lasso* and *Lars* selection. That's why we would advise to take the *Forward* method with *K-fold* criteria (19, 26% *Perfect Fitting* for 0% *Underfitting* and 80, 74% *Overfitting*) if you don't mind having less *Perfect Fitting* and more *Overfitting*. If you're prepared to have *Underfitting* then you can go for the former methods.
In the case of the best model, X_2 and X_5 are found 100% of the time and other variables are found about 80% of the time. However, the *Perfect Fitting* is only 47%, so having high rates for the variables is not enough to say that it is a good method. In fact, having less than 90% apparition rate on X_3 and X_4 is the reason why the Under Fitting rate is so high.

Another point would be: there's a chance -in both methods- that *Semi Failure* or *Total Failure* appear. It's less than *a percent* but it's still a risk you could be taking. That's why if you don't want no *Semi Failure* or *Total Failure*, taking into account the *Perfect Fitting* rate as the judge of the method, we would advise to take the *Forward* selection with the *K-fold* criteria again.

The method isn't suffering of *Underfitting* problems, nor *Semi Failure* and *Total Failure*. It could be a great deal if you are really afraid of those problems.

To conclude this subsection, if only looking at results and if you're willing to take the risk (less than 1%) of *Semi Failure* or *Total Failure*: we would advise to choose the *PRESS* criteria with the selection *Lars* or *Lasso*. Now if you want absolutely no *Underfitting*, *Semi Failure* or *Total Failure* and are ready to accept bigger chances of *Overfitting* then you could go for the *Forward* selection and *K-fold*.

3.4 Correlated Data with Outliers

In this section, we will see which method is the best on correlated data with outliers. But first, we need to define what is an *outlier*. An outlier is an observation that is significantly different from the other observations, and which the mean is very sensible to this. This may come from error measurement, the variability of the data. For example, if you take a sample of 100 kids of 6 years old and 1 meter tall and someone is 1.35 meter, then it is an outlier.

Now we can start to see how good or how bad *glmselect* will find the true model, as mentioned in the previous section.

As we did with generalized linear model and correlated data, we will use every method (*Forward*, *Backward*, *Stepwise*, *Elastic Net*, *Lars* and *Lasso*). And for each of them, we will use every criteria (*Statistical Learning* and *Machine Learning*) by only changing this line:

```
model Y= X1-X50/selection=forward(choose=aic);
```

Let's have a global look on the average of every rate of every method :

Method	PF	UF	OF	SF	TF
Forward	16,10	0	83,89	0	0
Backward	2,06	0	97,94	0	0
Stepwise	16,18	0	82,82	0	0
Elastic Net	14,98	0,25	84,73	0,043	0,001
Lars	19,63	19,22	61,03	0,11	0,001
Lasso	19,32	19.37	61,23	0,07	0,001

Considering the *Forward* method and every criteria, we have an average of $16,10\%$ of *perfect fitting* rate and $83,89\%$ of *Overfitting* rate. This method doesn't find any *Underfitting* model, but adds lots of variables that isn't in the true model. There are no *Semi Failure* or *Total Failure* models either.

Then, considering the *Backward* method, this has an average of $2,06\%$ of *perfect fitting* rate only and 97.94% of $Overfitting$ rate, which means this is the worst method to use for the correlated data with outliers, and by far, even if it did not find any *Semi Failure* or *Total Failure* models.

About the *Stepwise* method, this finds $16,18\%$ of the time the true model and $83,82\%$ an $Overfitting$ model. Percentages of *Semi Failure* and *Total Failure* are equal to zero.

However, for the *Elastic Net* method, this is not significant to talk about the mean because every rate of every criteria is very variable, in opposition to $Forward$, $Backward$ and *Stepwise* methods. *Perfect fitting* rate is from $6,19\%$ to $40,06\%$; *Overfitting* rate is from $59,68\%$ to $93,49\%$ and *Underfitting* rate is from $0,09\%$ to $0,36\%$ depending on the selected criteria. For the first time, a method has $0,043\%$ of *Semi Failure* models and $0,001\%$ of *Total Failure* models.

About the *Lars* method, this has a average of $19,63\%$ of *perfect fitting* rate, $61,03\%$ of *Overfitting* rate and $19,22\%$ of *Underfitting* rate. Nevertheless, one of the criteria is much better than the others. Moreover, there is $0,11\%$ *Semi Failure* models and $0,001\%$ *Total Failure* models. This is the method that founds the most *Semi Failure's* models for correlated data with outliers.

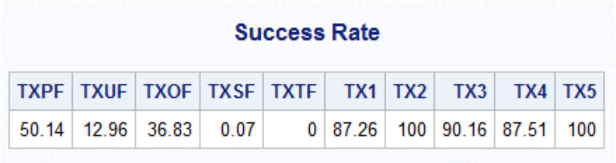
Finally, considering the *Lasso* method, the average of *perfect fitting* is $19,32\%$, $Overfitting$ is about $61,23\%$ and $Underfitting$ is $19,37\%$. There is $0,07\%$ *Semi Failure's* models and $0,001\%$ *Total Failure's* models. Like the *Lars* method, one criteria is better than the others and we will see it below.

In order to have a better look on which is the best model for correlated data with outliers, we do a top 3 ranking :

- Best model : *Lars* method with *PRESS* criteria : $50,14\%$ of *Perfect Fitting.*

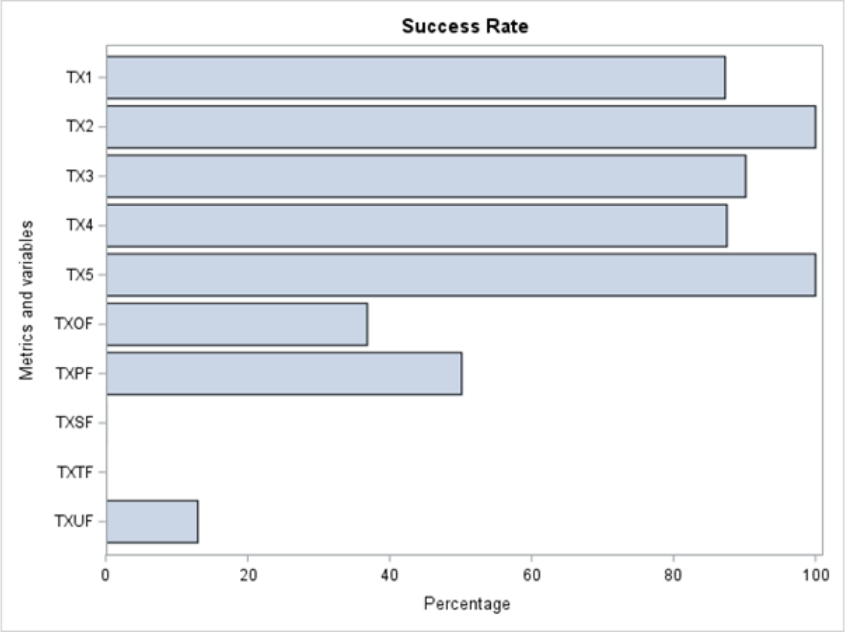
- 2nd best model : *Lasso* method with *PRESS* criteria : 48, 84% of *Perfect Fitting*.
- 3rd best model : *Elastic Net* method with C_p criteria : 40, 06% of *Perfect Fitting*.

Here is a graphic of the best model :



Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
50.14	12.96	36.83	0.07	0	87.26	100	90.16	87.51	100

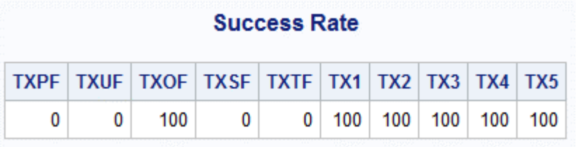


For the best model (*Lars* method with *PRESS* criteria), we can see that for X_2 and X_5, it finds them all of the time. However, this doesn't mean that it is always a *Perfect Fitting* case as there is also *Overfitting* (36, 83%). But for X_1, the model doesn't find it every time, as the rate is 87, 26% which means that there is about one out of ten chances that *glmselect* doesn't select X_1. We can say the same thing about X_3 and X_4.

We can also see the worst model :

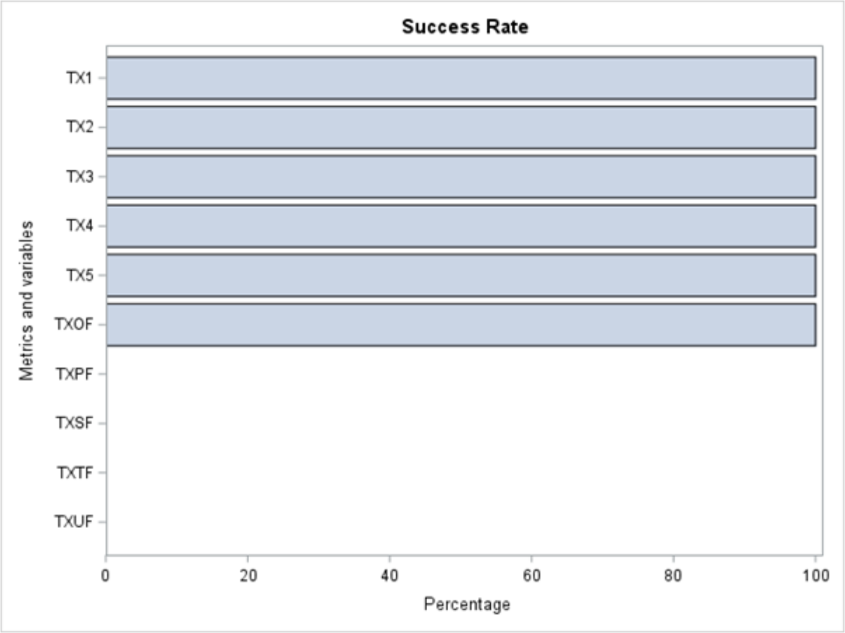
- Worst model : $Backward$ method with AIC and $Adjusted\,R^2$ criterias : 0% of $Perfect\,Fitting$.
- 2nd worst model : $Backward$ method with $PRESS$ criteria : 0, 17% of $Perfect\,Fitting$.
- 3rd worst model : $Backward$ method with $AICC$ criteria : 0, 18% of $Perfect\,Fitting$.

Here is a graphic of the worst model : $Backward$ method with AIC criteria :



Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
0	0	100	0	0	100	100	100	100	100



To conclude about this part, *Lars* method is the best with *PRESS* criteria and finds a *Perfect Fitting* half of the time. In the other half, there is twice as much *Overfitting* than *Underfitting*, which is good because it is better to have all variables plus variables that are not in our real model than not having all variables of our real model.
PRESS criteria is very efficient in *Machine Learning* models (*Elastic Net, Lars and Lasso*) whereas in *Statistical Learning* models, *PRESS Perfect Fitting* rates are very closed to *Statistical Learning* criteria like *AIC, BIC, AICC.*

3.5 Outliers only

In this subsection we're exploring the last case which is about, data with outliers only. We decided in 3.1.3 that the outlier ratio would be of 5%.

The average results for each methods are:

Method	PF	UF	OF	SF	TF
Forward	16,08	0	83,92	0	0
Backward	2.04	0	97,96	0	0
Stepwise	16,10	0	83,89	0	0
Elastic Net	20,05	10,64	68,77	0,42	0,12
Lars	26,27	18,69	55,03	0	0
Lasso	26,54	18,2	55,24	0	0

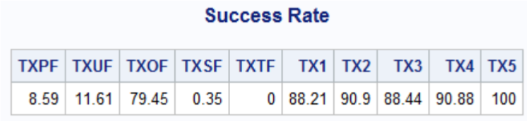
As we can observe, choosing the *Lasso* method has the best results on average.

Then again, the *Backward* method is the worst with only $2,04\%$ of *Perfect Fitting* while *Lars* has $26,54\%$ on average.

Looking into further details we can see that the worst methods are from the *Backward* selection. No matter the criteria used the *Perfect Fitting* rate won't go above $13,13\%$.

Rate	AIC	BIC	AICC	SBC	ADJRSQ	Cp	PRESS	K-fold
PF(%)	0,04	1,78	0,16	13,13	0	0,5	0,13	0,6

Once again the *Backward* selection isn't recommended. The same goes for the *Elastic Net* with *BIC* criterion:

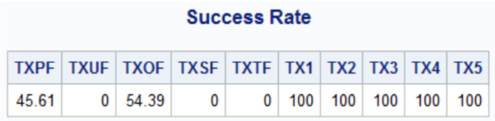

Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
8.59	11.61	79.45	0.35	0	88.21	90.9	88.44	90.88	100

Now if we look at the top 3 method for data with outliers:

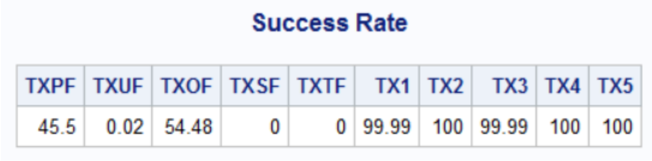
- 1. *Lasso* with *PRESS*:$45,61\%$ of *Perfect Fitting*
- 2. *Lars* with *PRESS*:$45,5\%$ of *Perfect Fitting*
- 3. *Elastic Net* with C_p: $38,44\%$ of *Perfect Fitting*

There's a good similarity with *Lasso* and *Lars* method with *PRESS* criteria, they both have around 45% of *Perfect Fitting*. Meanwhile the *Elastic Net* with C_p method is behind with $38,44\%$.
Looking at the overall results on data with outlier, *Machine Learning* criteria are usually performing better than *Statistical Learning* criteria. Here is a graphic for *Lasso* with *PRESS*:


Success Rate

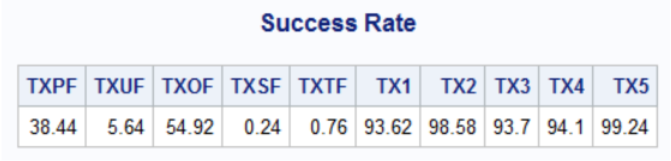
TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
45.61	0	54.39	0	0	100	100	100	100	100

here for *Lars* with *PRESS*:


Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
45.5	0.02	54.48	0	0	99.99	100	99.99	100	100

and here for *Elastic Net* with C_p:


Success Rate

TXPF	TXUF	TXOF	TXSF	TXTF	TX1	TX2	TX3	TX4	TX5
38.44	5.64	54.92	0.24	0.76	93.62	98.58	93.7	94.1	99.24

The results show a really good *Perfect Fitting* rate, even more there's less *Underfitting*. It's a good thing since we want to avoid *Underfitting* more than *Overfitting*. We wouldn't want our model to lack variables. The reason is simple, for an *Overfitting* model we could always try again to figure out which variables are unfit to belong to the model. But what if it's variables that are missing ? We can't really try out all variables one by one. Hence, choosing *Overfitting* over *Underfitting* is likely to be a better choice.
This is the reason why we advise to choose, either the *Lasso* or the *Lars* method with the Leave-One-Out criteria (*PRESS*). The *Elastic Net* with C_p method is not recommended for the reasons above.

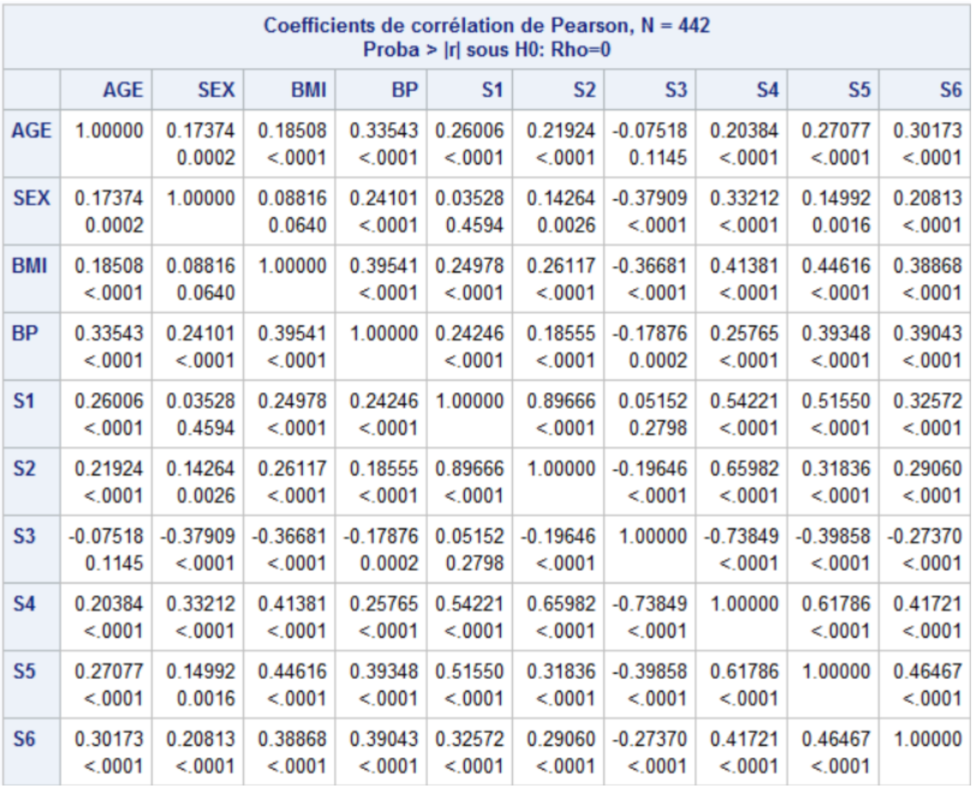
To conclude this subsection, the method producing the best results are *Lasso*/*Lars* with *PRESS*. Not only does this method produce significant rate of *Perfect Fitting* but it also possess a low rate of *Underfitting* which is best for our subject. In the four cases of data, the procedures without inference were generally more efficient than the procedures with inference.

3.6 Test on real data

In this part we want to apply our conclusion on a data set we found. We might not know if the model found with the method is the best model (the true model) but it's a good way to see how we should proceed.

First we will analyse the series profile: Are the variables correlated ? Is there any heteroscedasticity ? Are we in a case of Outliers series ? Determining the series profile is the first step before applying a model.

For this subject we decided to use the data set *Diabetes* (see the annex). It's made of 11 columns, 1 for the response Y and 10 columns for variables. It has 442 observations and the 10 predictors are: *Age*, *Sex*, *BMI* (Body Mass Index), *BP* (Average Blood Pressure), and $S1, S2, \dots, S6$ (results of six blood serum measurement).
To begin with the profile analysis, we look at the correlation between variables. Using the *proc corr* we have:

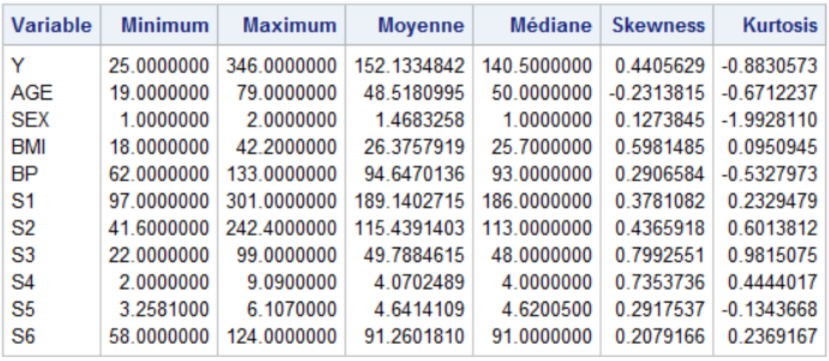


Coefficients de corrélation de Pearson, N = 442
Proba > |r| sous H0: Rho=0

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
AGE	1.00000	0.17374 0.0002	0.18508 <.0001	0.33543 <.0001	0.26006 <.0001	0.21924 <.0001	-0.07518 0.1145	0.20384 <.0001	0.27077 <.0001	0.30173 <.0001
SEX	0.17374 0.0002	1.00000	0.08816 0.0640	0.24101 <.0001	0.03528 0.4594	0.14264 0.0026	-0.37909 <.0001	0.33212 <.0001	0.14992 0.0016	0.20813 <.0001
BMI	0.18508 <.0001	0.08816 0.0640	1.00000	0.39541 <.0001	0.24978 <.0001	0.26117 <.0001	-0.36681 <.0001	0.41381 <.0001	0.44616 <.0001	0.38868 <.0001
BP	0.33543 <.0001	0.24101 <.0001	0.39541 <.0001	1.00000	0.24246 <.0001	0.18555 <.0001	-0.17876 0.0002	0.25765 <.0001	0.39348 <.0001	0.39043 <.0001
S1	0.26006 <.0001	0.03528 0.4594	0.24978 <.0001	0.24246 <.0001	1.00000	0.89666 <.0001	0.05152 0.2798	0.54221 <.0001	0.51550 <.0001	0.32572 <.0001
S2	0.21924 <.0001	0.14264 0.0026	0.26117 <.0001	0.18555 <.0001	0.89666 <.0001	1.00000	-0.19646 <.0001	0.65982 <.0001	0.31836 <.0001	0.29060 <.0001
S3	-0.07518 0.1145	-0.37909 <.0001	-0.36681 <.0001	-0.17876 0.0002	0.05152 0.2798	-0.19646 <.0001	1.00000	-0.73849 <.0001	-0.39858 <.0001	-0.27370 <.0001
S4	0.20384 <.0001	0.33212 <.0001	0.41381 <.0001	0.25765 <.0001	0.54221 <.0001	0.65982 <.0001	-0.73849 <.0001	1.00000	0.61786 <.0001	0.41721 <.0001
S5	0.27077 <.0001	0.14992 0.0016	0.44616 <.0001	0.39348 <.0001	0.51550 <.0001	0.31836 <.0001	-0.39858 <.0001	0.61786 <.0001	1.00000	0.46467 <.0001
S6	0.30173 <.0001	0.20813 <.0001	0.38868 <.0001	0.39043 <.0001	0.32572 <.0001	0.29060 <.0001	-0.27370 <.0001	0.41721 <.0001	0.46467 <.0001	1.00000

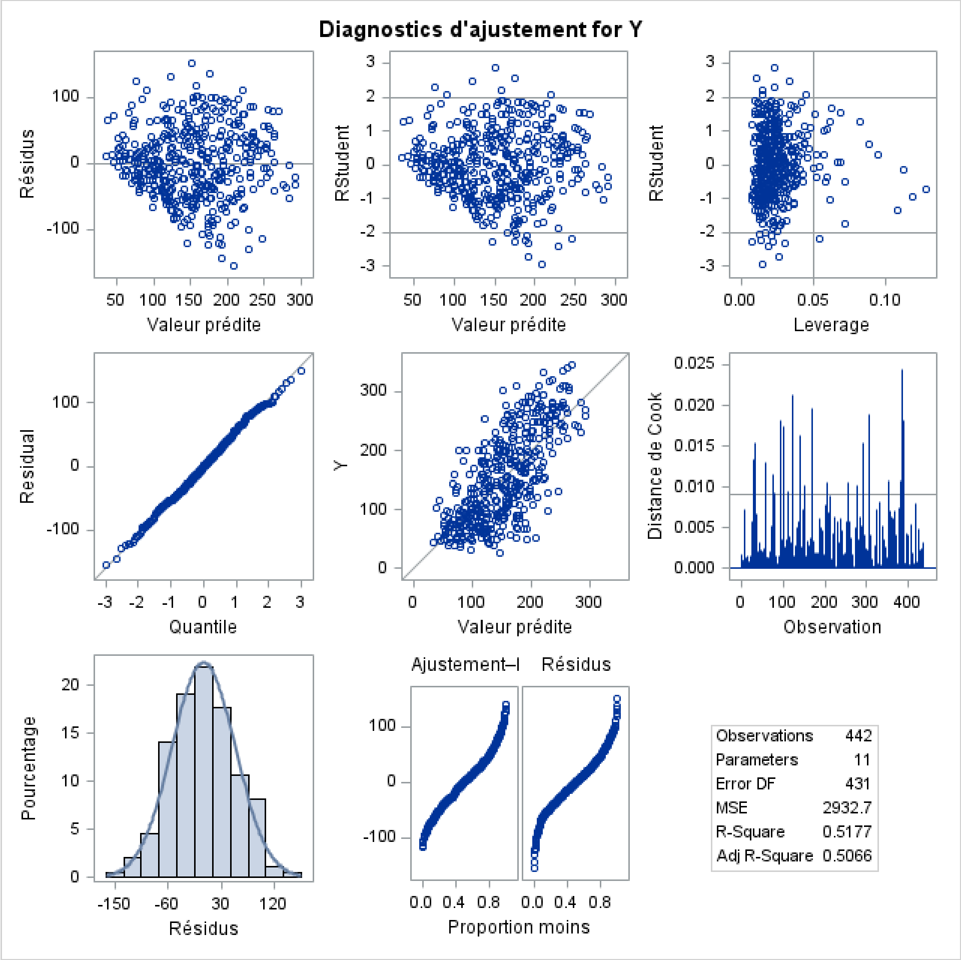
Looking at the results we can observe a correlation between variables. For example we can see that $S1$ and $S2$ are correlated. That makes sense since the two variables are blood measurement. Hence we will consider the data set to be of correlated variables.

Now, let's try to figure out: is there any outliers ?



Variable	Minimum	Maximum	Moyenne	Médiane	Skewness	Kurtosis
Y	25.0000000	346.0000000	152.1334842	140.5000000	0.4405629	-0.8830573
AGE	19.0000000	79.0000000	48.5180995	50.0000000	-0.2313815	-0.6712237
SEX	1.0000000	2.0000000	1.4683258	1.0000000	0.1273845	-1.9928110
BMI	18.0000000	42.2000000	26.3757919	25.7000000	0.5981485	0.0950945
BP	62.0000000	133.0000000	94.6470136	93.0000000	0.2906584	-0.5327973
S1	97.0000000	301.0000000	189.1402715	186.0000000	0.3781082	0.2329479
S2	41.6000000	242.4000000	115.4391403	113.0000000	0.4365918	0.6013812
S3	22.0000000	99.0000000	49.7884615	48.0000000	0.7992551	0.9815075
S4	2.0000000	9.0900000	4.0702489	4.0000000	0.7353736	0.4444017
S5	3.2581000	6.1070000	4.6414109	4.6200500	0.2917537	-0.1343668
S6	58.0000000	124.0000000	91.2601810	91.0000000	0.2079166	0.2369167

Proc means tells us the basic information we need. To sum up, the mean and the median are pretty much the same for all variables which seems to say we have no outliers. But if we take a look at the *skewness* and the *kurtosis* it seems evident that it appears to be outliers. In fact the *skewness* which measures the symmetry of a distribution isn't equal to zero like it should. The *kurtosis* (SAS subtract 3 to the real value) measures the "flatness" of the probability distribution. We can also see that it's a bit far from 3. The variables *Sex* and $S4$ could be said to have outliers, but then again we must not conclude too fast so we need more proofs.
If we look at the *Cook Distance* we can try to figure out if there's any outliers.

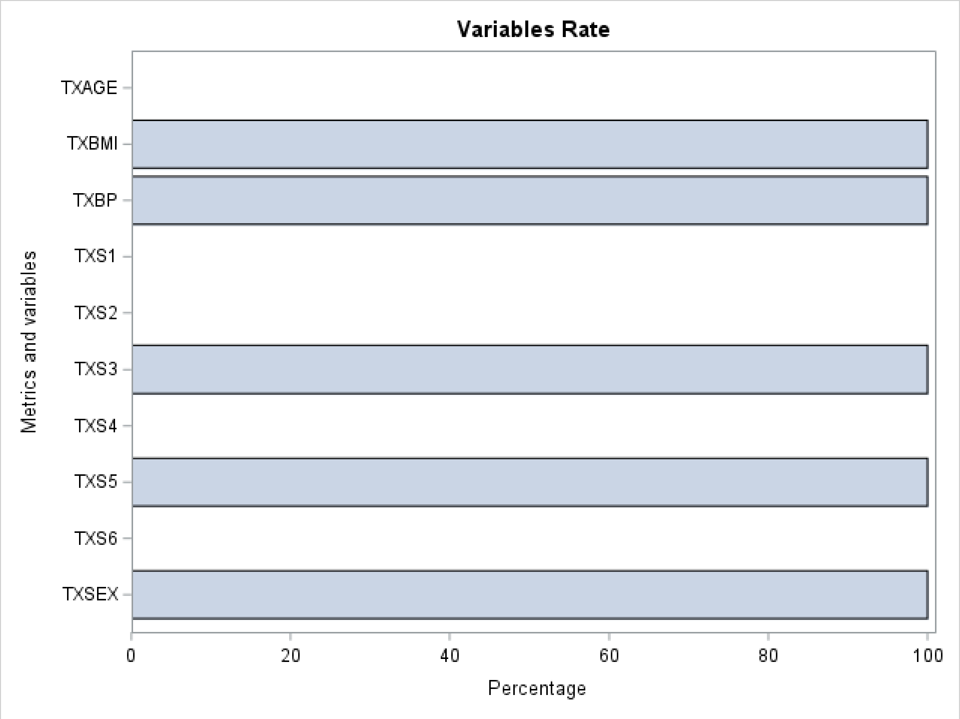


Observations	442
Parameters	11
Error DF	431
MSE	2932.7
R-Square	0.5177
Adj R-Square	0.5066

Looking at the graphic of the *Cook Distance* we can see points going over the threshold. Meaning that there's many outliers in this series. The same thing can be said for residuals, there's many points that are straying from the line inducing outliers, the series isn't following a Gaussian distribution.

With that said, we've determined that the data set is of correlated variables with outliers. Now as we've conclude in the previous sections, the best model we could found would be with the *Lasso* and *PRESS* method.

```
    proc glmselect data = diabetes outdesign=names;
model Y= AGE SEX BMI BP S1-S6 / selection=LASSO(choose=press lscoeff);
run;
```



The model that's proposed by the *Lasso* and *PRESS* method is:

$$Y = \beta_0 + \beta_1 SEX + \beta_2 BMI + \beta_3 BP + \beta_4 S3 + \beta_5 S5 + \varepsilon \tag{20}$$

We can compare it with the worst method we've found on correlated variables with outliers: *Backward* with *Adjusted* R^2 method. After running the code:

```
    proc glmselect data = diabetes outdesign=names ;
model Y= AGE SEX BMI BP S1-S6 / selection=Backward(choose=adjrsq);
run;
```

We can look at the results, and compare. The *Backward* method choose to kept the following model:

$$Y = \beta_0 + \beta_1 SEX + \beta_2 BMI + \beta_3 BP + \beta_4 S1 + \beta_5 S2 + \beta_6 S4 + \beta_7 S5 + \beta_8 S6 + \varepsilon \tag{21}$$

While this model might have a slightly better *Adjusted* R^2 (0.5086 vs 0.5030) it has a lower MSE. Our *Lasso* model has a MSE of 2953.85586 while the *Backward* model has $MSE = 2920.81889$. Moreover, out of 10 regressors of the *Backward* method kept 8 variables while the *Lasso* method kept 5 predictors. The aim of the variable selection is to lower the number of variables we are working with, if we end up with almost the same number of variables as the basic model there would be no using it. Here, using *Lasso* with *PRESS* method significantly reduce the size of the model.
Then again, this method is only the first step to variable selection. There's still a big chance to have *Overfitting*, so doing another regression could be a good idea.

4 Conclusion

To end this paper, we must conclude what results we have gathered so far. We have been testing tons of methods over tons of *DGP* and tons of series profiles...But what stands out is:

- For Generalized Linear data (unrelated and with no outliers) the best method would be: *Lars* method with *PRESS* criteria (46, 39% of *Perfect Fitting*)
- For Correlated data (no outliers but related variables) the best method would be: *Lars*/*Lasso* method with *PRESS* criteria (47% of *Perfect Fitting*)
- For Correlated data with Outliers the best method would be: *Lars* method with *PRESS* criteria (50, 14% of *Perfect Fitting*)
- For data with Outliers (unrelated) the best method would be: *Lasso*/*Lars* with *PRESS* criteria (45% of *Perfect Fitting*)

Given the conclusion it seems easy to see that the *Lars*/*Lasso* and *PRESS* method performs the best no matter the type of series. In fact, looking at overall results we can observe that *Machine Learning* Criteria tends to work better than *Statistical Learning* Criteria.

The *Backward* method is to avoid at all cost, no matter the type of data. It is, without exaggeration, never finding the true model. More than that, it's usually doing only *Over fitting*, we usually try to lessen the variables to a significant number so as to make it easier to work with the models, but most of the time *Backward* won't help at all. Avoiding this method seems a really wise choice.

As we've been through this subject we could observe and now conclude that: *Machine Learning* wins over *Statistical Learning.* No matter the type of data, *inferential procedures* were always behind *non-inferential procedures* in terms of *Perfect Fitting.* Now to answer the article, Thomas Becker stated that *Random Forest*, *Lars* and *Lasso* methods were best performing over the variable selection problem. We didn't explore the *tree-based* solutions but we have also reach the same conclusion regarding the *Lasso* and *Lars* method. We also agree when he says that, compared to *Forward* and *Backward*, *Stepwise* works better on simple data as we have seen in 3.2.

Even though we said the *Lars* and *PRESS* method are best for finding the true model in every situation we must take it at face value. It's merely a first step for our variable selection problem. In fact, we saw that even if the *Perfect Fitting* rate was high, the *Overfitting* rate was still high and significant. As there is no means to determinate if the found model is the true model, it could be wise to run another regression after applying the *Lars* with *PRESS* method. This first method could lessen significantly the number of variables and a second regression would most certainly find the true model though this is not the subject here.

5 Bibliography

References

[Aka74] Hirotugu Akaike. “A new look at the statistical model identification”. In: *IEEE transactions on automatic control* 19.6 (1974), pp. 716–723.

[Bec07] Thomas Becker. “Variable Selection”. MA thesis. Delft University of Technology, 2007.

[Dor+89] Brigitte Dormont et al. *Introduction à l’économétrie des données de panel. Théorie et applications à des échantillons d’entreprises*. Tech. rep. 1989.

[Efr+04] Bradley Efron et al. “Least angle regression”. In: *The Annals of statistics* 32.2 (2004), pp. 407–499.

[Efr60] Michael Alin Efroymson. “Multiple regression analysis”. In: *Mathematical methods for digital computers* (1960), pp. 191–203.

[Eze29] Mordecai Ezekiel. “Meaning and significance of correlation coefficients”. In: *The American Economic Review* 19.2 (1929), pp. 246–250.

[Fis92] Ronald Aylmer Fisher. “Statistical methods for research workers”. In: *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.

[HK70] Arthur E Hoerl and Robert W Kennard. “Ridge regression: Biased estimation for nonorthogonal problems”. In: *Technometrics* 12.1 (1970), pp. 55–67.

[IC82] Ronald L Iman and William-Jay Conover. “A distribution-free approach to inducing rank correlation among input variables”. In: *Communications in Statistics-Simulation and Computation* 11.3 (1982), pp. 311–334.

[Mal00] Colin L Mallows. “Some comments on Cp”. In: *Technometrics* 42.1 (2000), pp. 87–94.

[Sch78] Gideon Schwarz. “Estimating the dimension of a model”. In: *The annals of statistics* (1978), pp. 461–464.

[Stu08] Student. “The probable error of a mean”. In: *Biometrika* (1908), pp. 1–25.

[Tib96] Robert Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[Wri21] Sewall Wright. “Correlation and causation”. In: (1921).

[ZH05] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2 (2005), pp. 301–320.