LSTAT2450 : Examination Project Part 1

Tytgat Alexandre

January 7, 2021

alexandre.tytgat@student.uclouvain.be

Abstract

In this project, several estimation techniques, with special attention to penalized models, are applied in two different cases. The first one is a comparison study of simple linear regression vs. the Lasso method. This comparison is operated on a simulated dataset and the impacts of the following effects on the models are studied : low/high sample sizes, low/high correlation, low/high SNR and low/high dimensional data. In the settings where $p > n$ and a linear model can not be fitted, a Ridge regression is used instead. The performances are then evaluated on a test set by computing the MSE, the recovery of the active set and the MSE of the estimated coefficients. The results show that Lasso obtains better predictions than simple linear regression and Ridge in most cases and that it is always a simpler model (more sparse). In the second part, variable selection procedures are used on a real life dataset to identify the relevant predictors of diabetes in a given population. Beforehand, a visualization analysis is presented to get an idea of the relevant predictors. Then, several models are fitted to the data such as stepwise selection procedures, Elastic Nets, and a classification tree. Finally, the number of times each features were chosen is counted and the active set estimated is defined as the variables that were chosen by at least 60% of the models. The results come close the initial expectations from the previous visualization analysis.

1 Introduction

In this project, the aim is to put into practice several penalized techniques seen during the semester and study their properties. One of the most interesting property of these techniques is the sparsity of their predictions, or variable selection. Indeed, this aspect is especially important in high dimensional setups where it is assumed that most features are irrelevant for prediction. Thus, it becomes obvious that sparse methods are necessary to tackle problem in high dimensions. One of the most well known sparse penalized technique is Lasso or its generalization the Elastic Net. In the first part, Lasso will be compared to simple regression and Ridge in a simulation study to evaluate which methods work best in different cases. In the second part, the focus will be on the aspect of variable selection of Lasso. It will be used over a real life dataset to find the best determinants of diabetes in a given population of women.

2 Exercice 1 : Simulation Study

In this section, the aim is to provide a comparison analysis of unpenalized and penalized (LASSO, ridge) regression techniques. This comparison is based on a linear model build from a simulated dataset where the parameters are tuned to reflect different effects (dimensionality, number of samples, correlation, signal-to-noise ratio). The next subsection details the simulation procedure, then a description of the models is given and finally an analysis of the results is given.

2.1 Description of the simulated dataset

The following linear model was considered to build the response variable Y from the randomly simulated $n \times p$ data matrix X,

$$Y = X^T \boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1}$$

with $\boldsymbol{\epsilon} \sim N(0, \sigma^2)$, and $X \sim N(\mathbf{0}, \boldsymbol{\Sigma})$. Here, $\boldsymbol{\Sigma}$ is the covariance matrix with unit diagonal and off-diagonals elements equals to ρ. The choice of the coefficients values is given by this array :

$$\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, ..., \beta_{p-1}, \beta_p) = (0.4, 7, 3, 3, 3, 0, ..., 0, 0).$$

Only five of the p coefficients are chosen to be non-zero in order to evaluate the active set recovery aspect of Lasso. Note that the first coefficient is purposely close to zero for this reason. Indeed, in some cases it is harder for Lasso to estimate values close to zero.

As mentioned previously, the impact of different combinations of the model's parameters was studied on the estimation procedures. In particular, the effects that were investigated, and their associated values, are listed below :

- low/high sample size : $n = 15$ vs. $n = 100$,
- low/high correlation : $\rho = 0.2$ vs. $\rho = 0.7$,
- low/high SNR[1] : $\sigma = 1$ vs. $\sigma = 20$,
- low/high dimensional data : $p = 10$ vs. $p = 150$.

2.2 Description of the models

The simulation procedure is carried $M = 1000$ times. Each run, a training and a test set of equal sizes n are randomly sampled from the data simulated. Then, two settings are considered separately :

1. if $n \geq p$, a simple linear model and Lasso are fitted,
2. if $n < p$, a Ridge model is fitted instead of the linear model to get around the problem of the non-invertible matrix in OLS. Lasso remains unchanged.

The λ parameter of Ridge and Lasso is obtained by cross-validation. The explored space is defined as the range of 1000 equally spaced values in the interval $]0, 15]$. Regarding the number of folds, more accurate estimations can be obtained with a higher number of fold. However, the number of folds is restricted by the sample size. Thus, in order to balance the accuracy and the complexity of the computations, it was decided to set #folds=5 and #folds=10 for the low sample case and the high sample case respectively.

Given these considerations, the whole procedure was implemented in `R`. The linear model was fitted using the `lm` function, whereas the Lasso and Ridge models were estimated with the `cv.glmnet` function from the `glmnet` package. Note that, it was decided to not include intercepts in the models as it would be no more interesting than adding an additional null component β_0.

2.3 Results analysis

As mentioned in the previous subsection, the Ridge regression is used when it is not possible to fit the usual linear regression (if the data is high dimensional $p > n$). Thus, the analysis is separated into the low and the high dimensionality settings. It should be noted that the Ridge regression is a penalized procedure with a L^2 penalty. However, it is still interesting to use it to compare to the Lasso method. Indeed, both method perform regularization but only Lasso also perform variable selection.

For each simulations, the predictive performance of each model was evaluated as the MSE computed over the test set. Moreover, the complexity aspect of the models was evaluated as the MSE of the estimated coefficients. Indeed, high values of coefficients indicates that the model is not able to understand the relevance of features in predicting the response. A first issue can arise when coefficients values compensate each other by taking either really high or really low values. Moreover, it is often the case that only a few variables are relevant predictors, especially in the high dimensionality setup. Ultimately, the average of these measures is computed over all the simulations. Additionally, the variable selection aspect of Lasso is evaluated by how well the active set is recovered on average. This is done by computing the average TPR, FPR, FNR and FDR.

Beforehand, let's clarify a few definitions. Let $\hat{S}$ be the estimated active set, P the true set of active components and N the true set of non-active components. Then we have :

- $TP = \{j : j \in \hat{S} \text{ and } \beta_j \neq 0\}$ the set of correct hits on active components,
- $FP = \{j : j \in \hat{S} \text{ and } \beta_j = 0\}$ the set of falsely claim hits on non-active components,
- $TN = \{j : j \notin \hat{S} \text{ and } \beta_j = 0\}$ the set of correct hits on non-active components,
- $FN = \{j : j \notin \hat{S} \text{ and } \beta_j \neq 0\}$ the set of falsely claim hits on active components,

Hence, the meaning of the following ratios is clear : $TPR = TP/P$; $FPR = FP/N$; $FNR = FN/P$; $FDR = FP/(FP + TP)$. Note that, the FDR takes undefined values when both FP and TP are equals to zero. In this case, it is removed from the averaging.

[1] SNR= $\frac{\text{Var}(f(x))}{\text{Var}(\epsilon)}$ is the signal to noise ratio. In this case, f(x)=Y.

2.3.1 Low dimensional size $p = 10$

List of the main observations in Table 1 :

- Comparing the impact of the **correlation** when all other parameters values are fixed, one observe that it has a very low effects on both models overall. This is especially true for the linear model's MSE_{test} which is exactly the same with low or high correlation. Even though the differences are low, Lasso's results are mostly improved when the correlation increases : better MSE on the test set and better accuracy of the active set. However, this is not true for the MSE of the coefficients values which increases with the correlation. This might indicate that Lasso tend to overfit more when the correlation increases (better prediction but more complex model).

- Evaluating the impact of the **sample size** when the other parameters values are kept fixed, one can clearly see all aspects of the models improve with a higher number of sample. This is especially true for the linear model when the SNR is very low. This comes as no surprise, as a higher number of observations improves the estimations of the coefficients values and thus all other metrics are improved as well. With more samples, the model can also see more clearly through the noise which is why the improvement is so clear when SNR is low.

- Again, taking all parameters values as fixed and looking at the effect of **SNR**, this effect is definitely the most notable. With more noise, the estimations of the models worsen : higher MSE on the test set, worse active set recovery for Lasso, and higher estimations of the coefficients values. This is expected as more noise implies less precise estimations.

- Overall, for low dimensional data, **Lasso performs better than the simple linear model**. This is especially true when one consider a setup with low sample size and a lot of noise (low SNR). Lasso results over the test set are close to twice as good as the ones from the linear model. However, if the number of samples increases, the difference shrinks and both models have around the same MSE_{test} values. In this case, Lasso can still be considered a better model because it is still sparser.

- In the **special cases of low sample size and low SNR** regardless of the correlation (#2, #6), the results of the Lasso estimators are heavily badly impacted. Indeed, the TPR value is the lowest and its MSE_β is the largest. A low TPR value indicates that most of the non-zero coefficients have not been included in the active set. This is likely caused by the noise because it can be seen that if the number of sample increases (#4, #8) the TPR gets much better.

- Finally, regarding the **estimation of the active set**, Lasso seems to correctly identify at least 75% of its elements in most cases. Furthermore, it is really efficient at detecting non-active components since the FPR values are rarely above 30%. This is also confirmed by the low values obtained for the FDR.

#	ρ	n	σ	SNR	LM MSE_{test}	Lasso MSE_{test}	TPR	FPR	FNR	FDR	LM MSE_β	LASSO MSE_β
1	0.2	15	1	123.928	3.610	2.493	0.922	0.348	0.078	0.236	0.296	0.187
2	0.2	15	10	2.239	360.951	176.468	0.280	0.126	0.720	0.229	29.594	7.026
3	0.2	100	1	116.609	1.114	1.131	0.997	0.224	0.003	0.161	0.013	0.013
4	0.2	100	10	2.154	111.351	108.999	0.752	0.109	0.248	0.101	1.310	1.567
5	0.7	15	1	226.763	3.610	2.225	0.910	0.362	0.090	0.257	0.769	0.334
6	0.7	15	10	3.285	360.951	158.568	0.377	0.202	0.623	0.315	76.875	7.524
7	0.7	100	1	213.628	1.114	1.116	0.987	0.316	0.013	0.222	0.034	0.025
8	0.7	100	10	3.126	111.351	107.517	0.751	0.238	0.249	0.216	3.403	1.941

Table 1: Table of the models performances results for each setup with $p = 10$ (low dimensional data).

2.3.2 High dimensional size $p = 150$

List of the main observations in Table 2 :

- **The observations for Lasso made in the low dimensional** setting on the impact of correlation, sample size and noise are **also true in high dimension**. Plus, they also hold for the Ridge model. That is to say that the impact of correlation is low, the estimations quality increases with larger sample sizes and a lot of noise causes less precise estimations of the coefficients values. The only thing worth

mentioning is that, in the low SNR setting, the results of Ridge improves less than Lasso's when the sample size increases. This can be seen by comparing the MSE_{test} in #2 and #4, #6 and #8).

- Overall, **Lasso performs better than Ridge**. Indeed, in most cases it has a lower MSE_{test} than the Ridge model. This is likely thanks to the sparse property of Lasso. Indeed, even though both method are able to shrink the coefficients values toward zero, Ridge can only tend toward null values whereas Lasso can estimate null values. This hypothesis is reinforced by the fact that all MSE_{β} of Lasso are smaller than the ones of Ridge.
- The only setup where **Ridge performs better** than Lasso (lower MSE_{test}) is **for a low sample size and a low SNR** (#2 and #6). The TPR value is really close to zero meaning that most active components are estimated to be zero. Clearly, one can not hope to get good predictions if almost all relevant predictors are not taken into account. Since Ridge only shrinks down the coefficient, the relevant components are still at least a little bit taken into account which is probably why it performs better than Lasso in this case.

#	ρ	n	σ	SNR	Ridge MSE_{test}	Lasso MSE_{test}	TPR	FPR	FNR	FDR	Ridge MSE_{β}	LASSO MSE_{β}
1	0.2	15	1	121.980	60.273	30.911	0.418	0.031	0.582	0.615	0.459	0.280
2	0.2	15	10	2.200	177.313	198.833	0.062	0.011	0.938	0.769	0.538	0.520
3	0.2	100	1	116.560	23.648	1.260	0.986	0.065	0.014	0.608	0.226	0.002
4	0.2	100	10	2.157	153.169	123.152	0.607	0.030	0.393	0.475	0.416	0.180
5	0.7	15	1	225.103	23.419	13.571	0.523	0.062	0.477	0.752	0.460	0.283
6	0.7	15	10	3.229	139.735	162.283	0.084	0.026	0.916	0.887	0.608	0.666
7	0.7	100	1	213.425	10.964	1.248	0.942	0.080	0.058	0.695	0.264	0.005
8	0.7	100	10	3.130	124.474	117.134	0.499	0.055	0.501	0.738	0.485	0.282

Table 2: Table of the models performances results for each setup with $p = 150$ (high dimensional data).

2.3.3 Comparison of Lasso results in low dimensional vs. high dimensional data

First, let us analyze Lasso's prediction error. At first sight, it seems like it gets worse in higher dimensional setting when looking at the MSE results over the test set. However, in most cases they are still close to the results in lower dimension. Plus, it is almost inevitable that the prediction error increases with more parameters since the errors made on all the estimated coefficients add up. Thus, it is safe to say that Lasso is almost as efficient regardless of the dimension of the data and in all cases is more efficient than Ridge or OLS in most cases.

Other than that, the MSE obtained on the estimated components is much lower in high dimension than in low dimension. This can simply be explained by the fact that Lasso is very often able to correctly identify non-active components (low FPR values). Hence, much more components equals to zero are considered in the computation of the MSE, resulting in lower values.

Finally, Lasso's ability to recover the active set is highly impacted by the dimensionality of the data. Indeed, lower TPR values are found in the high dimensional setting meaning it is less able to discover the active components. However, its capacity to detect non-active components remains very effective in high dimension (low FPR). Also, its FDR values are badly impacted (sometimes heavily) in higher dimension. This signifies that most of the estimated actives components are actually non active components. This is likely caused by the fact that there are non-active in this setting so there are more errors to be made.

2.4 Conclusion

This simulation study provided a convenient framework to study the impact of common effects on penalized and unpenalized regression techniques. It was observed that Lasso is often more efficient than Ridge and OLS. Its property of sparsity is its main advantage in comparison to the Ridge method. Additionaly, it was noted that Lasso performances are not heavily impacted by correlation nor the dimensionality of the data. It is especially performant in comparison to other method in the case $p > n$. Furthermore, it is able to recover accurately the active components in many cases. The only setup where Lasso does a poorer job than the other techniques is for low SNR and low sample size.

3 Exercise 2 : Variable Selection on a Real-Life Dataset

In this section, a variable selection procedure is operated to detect the predictors of the presence of diabetes in women of at least 21 years old, from Indian heritage and living in the US. For this purpose, data collected by the US National Institute of Diabetes and Digestive and Kidney Diseases is used. Figure 1 gives a summary of the dataset. The procedure implemented is to fit to the data several models able to perform variable selection and then choose the determinants as the most recurrent selected features. For this task, it is preferable to have models with clear interpretation of the features and their relative importance. For this reason, it was decided to use both a stepwise AIC and BIC procedure, Lasso and Elastic Net and finally a classification tree. Furthermore, to make sure that the selected features are relevant for the prediction task, a training and a test set were randomly sampled from the full dataset to evaluate the performances of each models.

Variable	Description
NrPregnant	Number of times pregnant
Glucose	Plasma glucose concentration (glucose tolerance test)
Pressure	Diastolic blood pressure (mm Hg)
Triceps	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI	Body mass index (weight in kg/(height in m)$\hat{2}$)
Pedigree	Diabetes pedigree function
Age	Age (years)
Diabetes	Class variable (test for diabetes: pos/neg)

Figure 1: Description of the dataset from the US National Institute of Diabetes and Digestive and Kidney Diseases.

3.1 Dataset description

In this section, a detailed analysis of the dataset is given. Beforehand, it should be noted that they are samples with missing entries and that they were removed. Another possible solution to this issue would be to replace the missing values by the averages of the available values. However, 392 observations remains after the removal of the incomplete samples for only 8 features so it was decided that more observations were not necessary for the estimations quality.

First, a visualization analysis is given to get an idea of which features might be good discriminants. Figure 2 shows the density plot of each variables where each density is colored with respect to its class membership. In Figure 3, the box plots of each variables are presented. For a given feature, the less overlap between the two densities and between the boxes is a good indicator for it to be a good discriminant. With these considerations in mind and a quick look at both Figures 2 and 3, it seems that the best discriminants are the features `Insulin, BMI, Triceps` and in particular `Age` and `Glucose`.

Another important aspect is to check the class balance. One can see on Figure 4 that the classes are heavily imbalanced. In fact, there is about half as many positive cases as they are negative cases. This introduces a bias in the learning procedure which can results in poor performances in classification, especially for samples from the minority class.

3.2 Description of the models

In this subsection, a description of each models used in the learning procedure is given. As mentioned in the beginning, a training set and a test set have been randomly sampled from the full data set with around 2/3 and 1/3 of the observations respectively. Again, the problem of class imbalance is prevalent in both sets as can be observed in Figure 5.

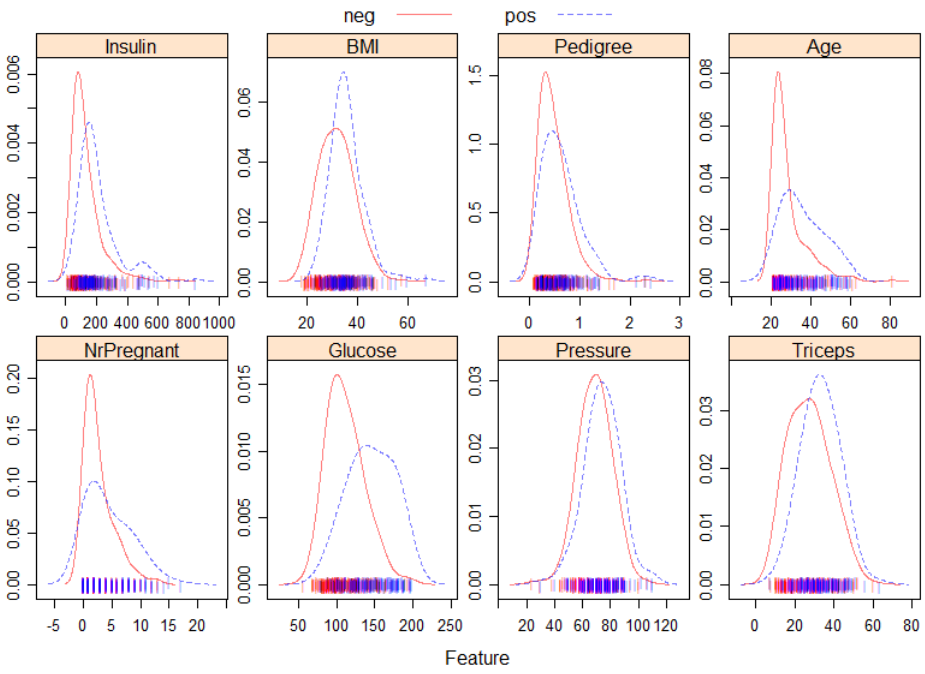

Figure 2: Densities graph of class memberships for each features. In red, individuals who have tested negative to diabetes and in blue who tested positive

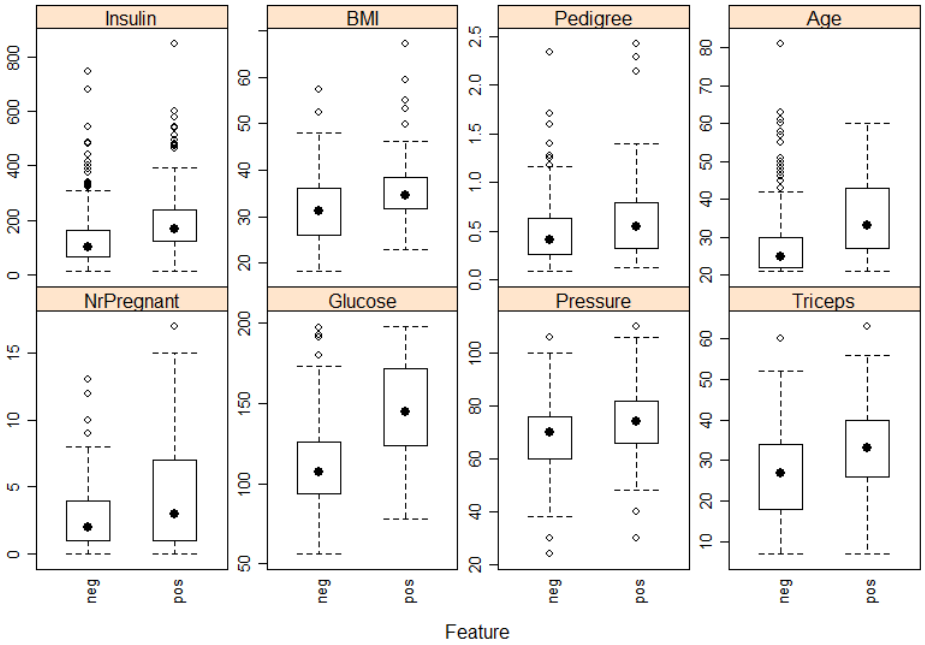

Figure 3: Box plots of class memberships for each features. pos. and neg. classes correspond respectively to individual who tested positive and negative to

3.2.1 Stepwise AIC/BIC

The first two models for variable selection are a stepwise procedure with AIC and BIC criterions. Since the response variable Y follows a Binomial distribution of mean μ, a logistic model is considered, such that,

$$\log(\frac{\mu}{1-\mu}) = X^T\boldsymbol{\beta}$$

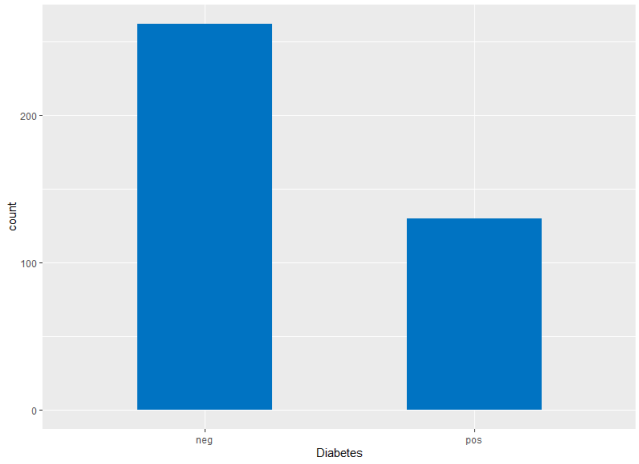

Figure 4: Histogram of the classes in the dataset.

(a) (b)

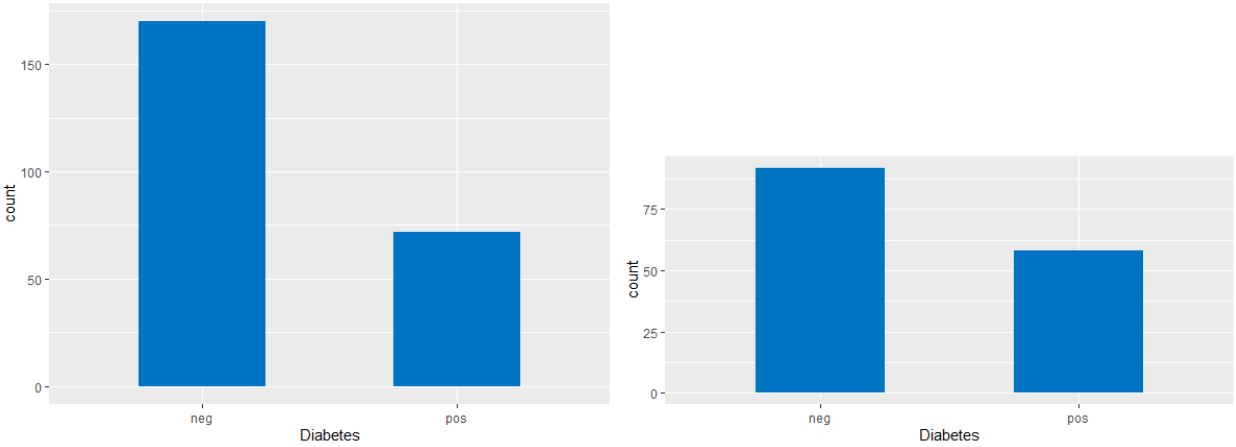

Figure 5: Histograms of the classes in the (a) training set, (b) test set.

where X is the matrix of samples and $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8)$[2]. First, the function `glm` from the `stats` package in `R` is used to fit the model. Then the `stepAIC` from the `MASS` package is used to perform the variable selection procedure. Note that, the backward direction is used but it is equivalent to the forward procedure. The AIC criterion is chosen by setting $k = 2$ whereas it is fixed as $k = log(n)$ for the BIC criterion :

$$\text{Criterion}(k) = -2\text{log-likelihood} + kp. \tag{2}$$

3.2.2 Lasso and elastic net

Both the Lasso and the elastic net models are fitted using the function `cv.glmnet` function from the `glmnet` package. This package uses the following penalty :

$$\text{Pen}(\beta) = \lambda \left((1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right)$$

The lambda space chosen is the interval $]0, 20]$ divided into 1000 equally spaced values. For Lasso, $\alpha = 1$ is set and $\alpha = 0.5$ for the elastic net. Furthermore, 10 folds are used to perform the cross validation and the distribution family of the response variable is set as a Binomial.

[2]Here the index of the coefficients is associated to the order of the variables presented in Figure 1

3.2.3 Classification tree

Lastly, a classification tree is estimated using the `tree` function from the `R` package of the same name. The default values for the parameters are used (`mincut = 5, minsize = 10, mindev = 0.01, split = 'deviance'`) but better values might be found using cross validation for instance. Then, the tree is pruned to only keep the five most optimal terminal nodes otbained from a 10-folds cross validation according to the misclassification error objective. This procedure is performed using the functions `prune.tree` and `cv.tree` again from the `tree` package.

3.3 Results analysis

Table 3 shows the estimated coefficients (thus the selected variables) for the stepwise AIC/BIC and Lasso/Elastic Net and the most relevant features for the classification tree along with the test set accuracy of all the models.

As expected, `Age` and `Glucose` are among the best determinants of the Diabetes diagnostic. They have been chosen by all the models and they are also among the most important features selected by each one. It is especially true for the classification tree which determined that they were the two most important features. Moreover, the importance of the variables in the regression models can be interpreted by a high absolute value for their associated coefficient. Clearly, when compared to the values of the other coefficients, their order of magnitude is around 10^{-2} whereas most other coefficients are either of the same order or less. The only notable exception is in the stepwise AIC model which estimated a coefficient of order 10^{-1} for the `NrPregnant` feature.

Another important feature that was selected by all the models is the `BMI`. It was also identified earlier as an important feature when visualizing the density and box plots. Unsurprisingly, is was chosen as the third best feature by the classification tree and all its associated coefficients estimated are one order of magnitude below the ones for `Age` and `Glucose`.

The fourth most important feature by the number of times it was selected is `NrPregnant`. This one comes more as a surprise as it was not as clear from the visualization analysis that it was a good determinant. However, it might not matter as much as the three best features. Indeed, if one look at the stepwise AIC and BIC model, it can be seen that the latter did not select `NrPregnant` as opposed to the former and has a better accuracy on the test set. Moreover, the coefficients values estimated by the two models are pretty close. In summary, the inclusion of `NrPregnant` might not be as relevant as it would seem from its count number.

Additionally, it might come to a surprise that `Pedigree` was chosen one time even though it was initially not deemed to be a good discriminator whereas `Insulin` was not chosen by any models and was thought to be among the best features for diabetes prediction.

Finally, `Triceps` was chosen two times and was among the expected good predictors and `Pressure` did not get selected by any models which corresponds to our initial expectations that is was not a good predictor.

	Intercept	NrPregnant	Glucose	Pressure	Triceps	Insulin	BMI	Pedigree	Age	Accuracy
Stepwise AIC	-9.559	0.118	0.035	/	/	/	0.072	/	0.044	0.74
Stepwise BIC	-9.718	/	0.035	/		/	0.064	/	0.070	0.75
Lasso	-5.242	0.031	0.023	/	0.005	/	0.008	/	0.028	0.753
EN	-3.486	0.021	0.015	/	0.001	/	0.002	/	0.019	0.747
Class. Tree	/	/	#1	/	/	/	#3	#4	#2	0.713
Count	/	60%	100%	0%	40%	0%	100%	20%	100%	/

Table 3: A table as an example

3.4 Conclusion

Overall, the results come close to our initial expectations from the visualization analysis. Our model indicates that the best determinants for the prediction of diabetes in the given population of women are the concentration of glucose, the age of the individual and its BMI. This is not surprising, since it is well known that diabetes is heavily linked to glucose and is most common in older individual. Maybe more surprisingly, it would seem that the number of pregnancy also has an impact on the diabetes diagnostic.