Evaluation of statistical learning and machine learning models

Loana Abou Chacra, Damien Lebas, Cristian Carp and Oscar Stemmelin

2025/2026

Contents

Abstract

This thesis analyzed the bias-variance tradeoff in the context of variable selection and regulation procedures. The objective was to evaluate the performance of the methods (Forward, Backward, Stepwise, LASSO, LARS, and Elastic net) in identifying the correct support according to different scenarios applied to the data (extreme values, multicollinearity, and breakpoints). Through Monte Carlo simulations, we showed that the choice of stopping criteria is essential in order to estimate the right model. However, it turns out that there is no universal stopping rule, since it all depends on the data structure. We also then conducted an empirical study on data with multicollinearity in order to validate our observations. Our results confirmed that AIC, which is accuracy-oriented, favors a slight bias and generates constant overfitting. Conversely, BIC favors parsimony and increases the risk of underfitting.

Introduction

The field of Statistics is constantly challenged by the problems that Science and Industry bring to its door, since vast amounts of data are being generated in many fields. With the emergence of computers and the information age, statistical problems have exploded both in size and complexity, leading to a profound transformation in the statistical sciences where computation plays a key role.

In econometrics and statistical learning, the question of automatic variable selection arises, particularly when the database contains numerous predictors. In such contexts, the core problem in prediction modeling is not the estimation of the variables, but the out-of-sample generalization. Indeed, as the model becomes increasingly complex in its training, it can adapt to more complicated underlying structures. Hence there is a decrease in bias but an increase in variance. The performance of the training model is increasing with its complexity, typically dropping the training error to near to zero if it is increased enough. However, a model with near-zero training error is overfit to the training data, leading to a poor out-of-sample generalization. Conversely, overly constraining the model to reduce variance increases bias, producing and underfitted model that also generalizes poorly. Therefore, the quality of a model is fundamentally determined by the arbitration between its bias and variance.

To this extent, the bias-variance tradeoff is tackled by subset selection and shrinkage methods, which can be perceived as mechanisms that control the complexity of the model. These methods are particularly useful when the number of possible predictors is high or even exceeds the number of observations of the dataset. The prediction accuracy can be improved by shrinking or setting some coefficients to zero. By doing so, we sacrifice a little bit of bias to reduce the variance

of the predicted values. Moreover, we can achieve parsimony by keeping the variables with the strongest signal. Yet, unlike continuous estimators, the selection procedures are discrete, considering that some variables are included whereas others are not. Consequently, minor data perturbations can lead to very different selected supports. The variance therefore does not only relate to the estimated coefficients, but also to the number of selected variables, which raises important questions regarding the stability of these methods.

The performance of these procedures depends heavily on the selection criterion and the objective pursued. In effect, if we use them for an explanatory purpose, then criteria that favor parsimony and interpretability are going to be preferred. In contrast, when the objective is prediction, validation-based criteria such as cross-validation are typically employed. Thus, the same algorithm can produce very different results depending on the criterion used.

Literary survey

The bias-variance interplay has been an extensively documented concept in the estimation theory and statistical learning, as it captures the intrinsic compromise between model flexibility and *"generalization"* on new data.

In predictive modeling, the objective is not only merely to fit the observed data well, but to obtain a model that approximates accurately on new, unseen observations. This performance is commonly measured by the estimation error. As emphasized in the literature, an unbiased estimator may still have a large mean-squared error if the variance is large. Thus, either bias or variance can contribute to poor performance. There is often a tradeoff between their contributions to the estimation error. Indeed, the variance can be reduced through "smoothing", for example by combining the influences of samples that are nearby. However, this will introduce bias, since details of the regression function such as peaks or valleys will be blurred (Geman et al. (1992); Hastie et al. (2009)).

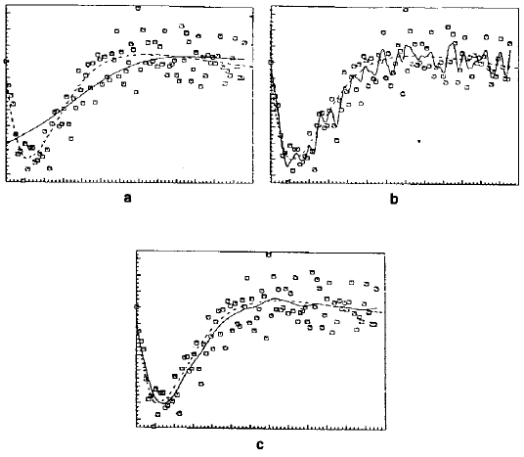

Figure 1: The bias–variance tradeoff. (a) Underfitting: high bias. (b) Overfitting: high variance. (c) An optimal compromise. Source: Wahba and Wold (1975).

This interchange is illustrated in Figure 1. In each panel, the broken curve is the regression and the solid curve is the spline fit. Panel (a) corresponds to an overly simplified model that has very little variance, but fails to capture the underlying structure. Panel (b) depicts an excessively flexible model that matches the noise and decreases the bias, but consequently explodes the variance. In the third panel (c), we can observe an optimal equilibrium between the bias and variance, that results in the lowest generalization error. From a conceptual standpoint, bias measures the systematic error introduced by approximating a potentially complex relation with a simplified model, whereas variance reflects the sensitivity of the estimator to fluctuations in the training sample. These occurrences are well documented and are the cornerstone of the modern predictive strategies (Hastie et al. (2009)).

This tradeoff is especially important when the number of the predictors increases, or even when they are correlated. In this case, the variance of the classical least squares estimators is substantial, even if the underlying structure is correctly specified. Although historically and originally used in prediction, the ordinary least squares (OLS) are now unsatisfactory due to the prediction accuracy (Tibshirani (1996)). Apart from

being optimal only under strong assumptions, the least squares are structurally defined within an inferential framework, criteria which poorly address out-of-sample performance. Moreover, they prioritize unbiased-ness at the expense of precision. The problem is significantly enhanced in high-dimensional datasets, where the number of predictors can be comparable with, or even superior to the number of observations, where the latter implies the impossibility to define the estimator due to the singularity of the design matrix.

To address this issue, the literature has proposed various mechanisms aimed to control the model complexity. Amongst these, variable selection and regularization methods are designed to reduce the variance by restricting the coefficients of the effective model. The concept behind them is that by removing impertinent predictors or shrinking coefficient estimates towards zero, bias is deliberately introduced in order to decrease the complexity, respectively to reduce variance and improve generalization.

These methods trade bias for variance reduction, offering estimators with improved predictive accuracy, especially in high-dimensional settings. As depicted in Figure 2 for linear models, the model space consists of all the linear predictions generated from the inputs and the black dot labeled "closest fit" is the estimator that best approximates the observed truth. The blue-shaded region depicts the noise-related uncertainty in the training sample, while the model bias and variance are represented by the large yellow circle centered at the black dot labeled "closest fit in population."

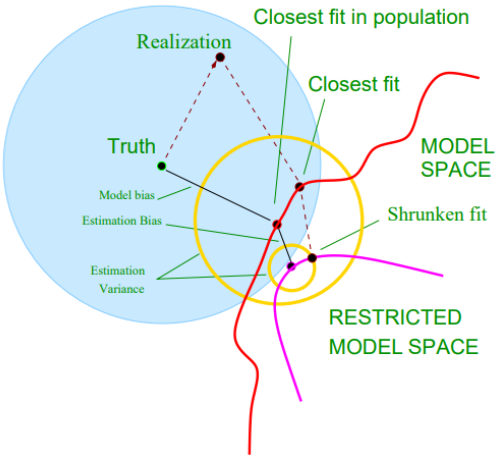

Figure 2: Schematic of the behavior of bias and variance. Source: Hastie et al. (2009).

A regularized or shrunken fit is also shown. Even though it has additional estimation bias, the prediction error is smaller due to its decreased variance. As the literature emphasizes, when a model is fit with fewer predictors, or regularized with shrunk coefficients, the result is the "shrunken fit" illustrated in the figure. The fit has additional bias due to not being the closest fit in the model space, however it has a smaller variance. "As long as the reduction in variance exceeds the increase in bias, the tradeoff is considered to be worthwhile" (Hastie et al. (2009)).

However, unlike continuous estimators, variable selection procedures are discontinuous, since they either exclude or include predictors. This discrete nature introduces an additional variability that is not fully captured by the classical bias-variance decomposition. Furthermore, as Breiman (1996) states in his work on heuristic instability, model uncertainty arises when a small change in the modeled data leads to substantial changes to the selected sequence. In this sense, variance is not only associated with the magnitude of the coefficient estimations, but also with the composition of the selected model itself, leading to higher predictive errors. Solutions such as the stabilization by averaging were proposed, but were not a *"panacea"*, showing that an area that needs exploration is the possibility of stabilization of procedures by changing their structure rather than averaging.

On the other hand, unlike variable selection procedures that enforce sparsity by exclusion or inclusion decisions, the shrinkage methods operate by continuously penalizing the magnitude of the coefficients. Ridge regression introduces an ℓ_2 penalty that shrinks coefficients towards zero without enforcing sparsity, while the LASSO (Tibshirani, 1996) employs an ℓ_1 penalty that induces exact zeros, thus performing variable selection and estimation simultaneously. The Elastic Net (Zou and Hastie, 2005) combines both penalties to address the limitations of the

LASSO in the presence of multicolinearity. These methods are particularly effective in high-dimensional settings, offering improved predictive accuracy. That being said, their performance critically depends on the regularization parameter, which oscillates the complexity of the model.

Therefore, both variable selection processes and shrinkage methods are mere mechanisms that are constrained by limits. As the literature exhaustively mentions, the selection criterion depends on the desired objective. In explanatory settings, criteria favoring parsimony and interpretability, such as inferential tests or information criteria, are commonly made use of. Predictive objective prefer instead criteria that directly target the generalization error, namely the cross-validation. As a consequence, the same procedure may show different behavior depending on the stopping rule employed, that is to say its *structure*.

This master's thesis investigates how variable selection and shrinkage methods manage the bias-variance tradeoff under different data-generating processes. Using Monte Carlo simulations, we evaluate their ability to recover the true model, control overfitting and underfitting - translated into probabilities - and generalize out-of-sample, depending on the stopping criterion employed. Particular attention is paid to model instability, measured through false positives, false negatives, and the variability of selected supports across replications.

In Section 1 we will define the theoretical framework, followed by Monte Carlo simulations in Section 2. The performance is going to be assessed using predictive and selection metrics, illustrated by overfitting and underfitting probabilities. A real data example is given in Section 3, while Section 4 contains a conclusion.

1 The theoretical framework

Let us consider $(X, Y) \in \mathbb{R}^2$ a pair of random variables and (x_i, y_i) a pair of observations. We will consider these observations independent, which implies that every dataset $\mathcal{D}$ may vary both in structure and variability.

In order to perform reliable out-of-sample generalization, the aim is to construct a predictive function $\hat{f}(x)$ based on a training set $((x_1, y_1), ..., (x_n, y_n))$, for the purpose of approximating Y at future observations of X. Considering the dependence of $\hat{f}$ on the data $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$, we will write $\hat{f}(x; \mathcal{D})$ instead of simply $\hat{f}(x)$.

1.1 The Bias-Variance Decomposition of Mean-Squared Error

For a given dataset $\mathcal{D}$ and a particular X, a common measure of the effectiveness of the predictor of Y is the mean-squared error

$$\mathbb{E}\big[(Y - \hat{f}(x; \mathcal{D}))^2 | X, \mathcal{D}\big]$$

where $\mathbb{E}[.]$ means expectation regarding the probability distribution. The dependency of $\hat{f}$ on the fixed data $\mathcal{D}$ can be noted as

$$\mathbb{E}\big[(Y - \hat{f}(x;\mathcal{D}))^2|X,\mathcal{D}\big] = \mathbb{E}\big[(Y - \mathbb{E}[Y|X=x])^2|X,\mathcal{D}\big] + (\hat{f}(x;\mathcal{D}) - \mathbb{E}[Y|X=x])^2$$

where $\mathbb{E}\big[(Y - \mathbb{E}[Y|X=x])^2|X,\mathcal{D}\big]$ does not depend on the data $\mathcal{D}$ or on the estimator, but is the incompressible variance of Y given x, hence the squared distance of the second member measuring the effectiveness of $\hat{f}$ as a predictor of Y. Presumably, for a particular training set $\hat{f}(x;\mathcal{D})$ can be an exceptional approximation of $\mathbb{E}[Y|X=x]$, making it an near-optimal predictor of Y. Yet, it is conceivable that for other realizations of $\mathcal{D}$, the performances of $\hat{f}(x;\mathcal{D})$ can be average or worse, making it a weaker approximation of Y due to its variations. Following the classic decomposition of Geman et al. (1992), the mean-squared prediction error can be derived as follows :

$$\mathbb{E}_{\mathcal{D}}\Big[(\hat{f}(x;\mathcal{D}) - \mathbb{E}[Y \mid X=x])^2\Big] = \mathbb{E}_{\mathcal{D}}\Big[\Big((\hat{f}(x;\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})]) + (\mathbb{E}_{\mathcal{D}}[\hat{f}(x;\mathcal{D})] - \mathbb{E}[Y \mid X=x])\Big)^2\Big]$$

$$= \mathbb{E}_{\mathcal{D}}\big[(f(x;\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})])^2\big] \quad + \mathbb{E}_{\mathcal{D}}\big[(\mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})] - \mathbb{E}[Y \mid X=x])^2\big]$$

$$+2\,\mathbb{E}_{\mathcal{D}}[(f(x;\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})])(\mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})] - \mathbb{E}[Y \mid X=x])]$$

$$= \mathbb{E}_{\mathcal{D}}\big[(\hat{f}(x;\mathcal{D}) - \mathbb{E}[Y|X=x])^2\big] + \mathbb{E}_{\mathcal{D}}\big[(\hat{f}(x;\mathcal{D}) - \mathbb{E}_{\mathcal{D}}(\hat{f}(x;\mathcal{D}))^2\big]$$

$$= \mathrm{Bias}^2(\hat{f}) \quad + \quad \mathrm{Var}(\hat{f})$$

where $\mathbb{E}_{\mathcal{D}}$ describes the expectation of the mean-squared error for a given training set $\mathcal{D}$. When the estimator is systematically different from $\mathbb{E}[Y|X=x]$, then it is considered that $\hat{f}(x;\mathcal{D})$ is biased. We can therefore define the bias as the systematic error induced by the approximation of Y with a poorly specified estimator. Nonetheless, even if $\hat{f}(x;\mathcal{D})$ is unbiased, it can still remain very sensitive to the data, leading to a high mean-squared error due to strong variance. Consequently, the variance reflects the sensitivity of the estimator to oscillations in the sample.

Considering the fact that our main objective is to reduce the predictive error to a maximal degree, we can observe that it can never reach the zero limit. Indeed, there is the irreducible variance σ^2 of Y depending on X that cannot be diminished because of the fundamental uncertainty in the data-generating process. However, it can be adjusted through the $Bias^2$ and the $Variance$ of the estimator $\hat{f}$. The difficulty consists in the fact that minimizing one will necessarily inflate the other, making the tradeoff unavoidable.

Understanding this interchange - translated into model complexity - is critical to better understand the behavior of the prediction models. Mathematically, the optimal model complexity is achieved when the increase in bias is equivalent to the reduction of variance :

$$\frac{\delta \text{ Bias}}{\delta \text{ Complexity}} = -\frac{\delta \text{ Variance}}{\delta \text{ Complexity}}$$

In practice, even though we do not observe the decomposition, the consequences of the bias' or variance's domination can be noted on the predictive error.

1.2 The generalization's obstacles : the overfitting and underfitting

By construction, a prediction function has the objective to minimize a prediction loss, most commonly the the mean-squared error, in other words the squared distance between the real variable and its estimation

$$\mathbb{E}\big[(Y - \hat{f}(x; \mathcal{D}))^2 | X, \mathcal{D}\big]$$

that can also be perceived as a loss function. Therefore, let us note $L(Y, \hat{f}(x; \mathcal{D}))$ the loss function that we want to minimize.

When building a predictive model, we split the available data in two sub-samples: a training set to estimate the function $\hat{f}$, and a test set that is used to assess its accuracy. In order to evaluate the effectiveness of the said model, we typically compare the training error

$$Err_{train} = \frac{1}{N} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i; \mathcal{D}))$$

with the test error, also called the generalization error

$$Err_{test} = \mathbb{E}_{(X,Y)}\big[L(Y, \hat{f}(X; \mathcal{D})\big]$$

Overfitting occurs when the model is too flexible and thoroughly adjusts itself not only to the underlying signal, but also to the noise of the training sample. As a consequence, the training error is very low , while the generalization error remains very large:

$$Err_{train} < Err_{test}$$

According to Hastie et al. (2009), the overfitting can be quantified by calculating the optimism of the training error, which reflects the tendency of the training error to underestimate the true prediction error:

$$Optimism = Err_{test} - Err_{train}$$

A high *Optimism* indicates an excessive model adaptability, and is the sign of overfitting. On the contrary, the **underfitting** corresponds to models that are too constrained or simple to capture the underlying structure, leading to **high** training and test errors:

$$Err_{train} \approx Err_{test}$$

In this case, the training error *Optimism* is very low. To further illustrate these occurrences, we can look at the optimal model complexity from Figure 3. A complex model has low bias because

it learned well the peaks and the valleys of the underlying signal, however the cumulation of the effects explodes its variance, leading to overfitting. Conversely, a simple or overly constrained model does not capture the underlying structure, leading to a very high bias due to the fact that it cannot predict all the true variables, resulting in underfitting.

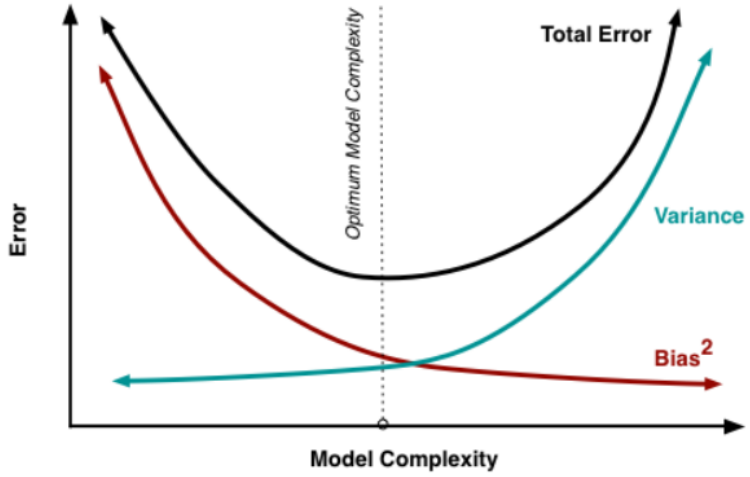

Figure 3: The optimal model complexity. Source: Fortmann-Roe (2012).

These phenomena, along with *Optimism*, prove the reason why the training error cannot be considered as a reliable source of evaluation criterion. In this master's thesis, the overfitting and underfitting will be defined by discrete events, namely the selected support, and will be quantified by probabilities. Indeed, the reason of this work is the evaluation of the methods, specifically their ability to recover the true model. Thus, let us introduce the notions that will be employed in order to quantify the metrics.

1.3 The construction of the metrics

Let S denote the **true support**, defined as the set of predictors with non-zero coefficients that truly explain Y in the data-generating process, and $\hat{S}$ the **selected support**, corresponding to the set of variables chosen by a given method.

The **cardinality** of a given selected support is defined as

$$|\hat{S}|$$

that represents the number of predictors in the selected support. It is a good measure of model complexity.

A **true positive** (TP) is a variable that belongs to the true model S and is correctly selected in $\hat{S}$, that can be defined as follows:

$$TP = |\hat{S} \cap S|$$

A **false positive** (FP) is a predictor that does not belong to the true support S, but is nonetheless incorrectly included in the selected support $\hat{S}$:

$$FP = |\hat{S} \setminus S|$$

A **false negative** (FN) is a relevant predictor in the true support S, but is omitted by the selection procedure in the support $\hat{S}$:

$$FN = |S \setminus \hat{S}|$$

These elements will be maneuvered in Section 2 in order to compute the overfitting and underfitting probabilities considering the employed method.

1.4 Variable Selection and Regularization methods

The complexity of a model determines its bias, variance and the stability of the selected support. Although the following defined methods can be perceived as mere mechanisms that act accordingly to their selection criterion, their *modus operandi* is not the same, therefore it is important to differentiate them.

Suppose that the data set has n observations and p predictors. Let us note $y = (y_1, ..., y_n)$ the variable we want to predict and $X = (x_1|x_2|...|x_p)$ the model matrix, where each $x_j = (x_{1j}, ..., x_{nj})^\top$, $j = 1, ..., p$ are the predictors. We will also assume that the variables we want to predict are centered and the predictors are standardized:

$$\sum_{i=1}^{n} y_i = 0, \qquad \sum_{i=1}^{n} x_{ij} = 0 \quad and \quad \sum_{i=1}^{n} x_{ij}^2 = 1 \quad for \quad j = 1, ..., p.$$

This normalization ensures the fact that the regularization penalty acts uniformly independently of the magnitude of the coefficients.

1.4.1 Subset selection procedures

Subset selection methods aim to identify a subset of predictors from a larger set of of candidates by solving a sequence of comparison problems. At each step, variables are either included or excluded from the model according to a predefined criterion. The discrete nature of these procedures makes them inherently unstable (Breiman, 1996). It is crucial to note that the final model depends on the path taken during the procedure, and the stopping rule plays a central role, determining the complexity of it. According to Hastie et al. (2009), these procedures do not solve a single global optimization problem. They instead rely on greedy, path-dependent decisions.

Forward Selection

The **Stepwise Forward** method starts from a model containing only an intercept. At each iteration, variables are added sequentially based on their contribution to improving the model according to a given criterion (such as the $p - value$, AIC, R^2). The algorithm stops when no remaining variable leads to a significant improvement.

At each step, one candidate variable is added, and its contribution is validated conditionally on the variables already selected. This validation is tested using the Fisher statistic:

$$F = \frac{(RSS_{k_1} - RSS_{k_1+k_2})/k_2}{RSS_{k_1+k_2}/(T - (k_1 + k_2) - 1)}.$$

where $RSS_{k_1+k_2}$ represents the RSS of the augmented model. Although under classical assumptions the exact distribution of this statistic is known, the adaptive nature of the procedures make the $p - values$ non valid. Thus, this approach is "out of fashion" (Hastie et al, 2009), and other criterion are used as a stopping rule. Conversely to the *Backward Selection*, Forward selection is computable even for a p larger than n ($p > n$). However, it is more likely to have an important bias due to the fact that it can miss relevant variables.

Backward Selection

The **Stepwise Backward** method follows the same principle as the forward approach, except that it starts from the full model and removes, at each step, the least significant variables until all the remaining predictors are relevant.

The test statistic is defined as follows:

$$F = \frac{(RSS_{k_1-k_2} - RSS_{k_1})/k_2}{RSS_{k_1}/(T - k_1 - k_2)}.$$

where k_2 represents the set of "non-significant" variables excluded. As for the *Forward Selection*, the distribution of the statistic is theoretically known, yet due its limit, other criteria is used as a stopping rule. A common criterion si the AIC.

Stepwise Selection

The Stepwise Selection combines the Forward inclusion and the Backward exclusion, allowing the variables to enter and to leave the model at different stages. Just as the precedent methods, the results of *stepwise selection* drastically vary depending on the chosen criterion (Cohen, 2006).

1.4.2 The shrinkage methods

The regularization-based methods estimate the regression coefficients by solving a penalized optimization problem, where the model complexity is controlled by the continuous penalty on the magnitude of the coefficients. They are composed of three main components:

- The empirical error $\sum_{i=1}^{n}(y_i - x_i^\top \beta)^2$
- The regularization term λ, also called the complexity parameter
- The penalization $\mathcal{P}(\beta)$.

It should be emphasized that the regularization term λ plays a key role, since it determines the strength of the penalty. When $\lambda \to 0$, the estimation becomes a classic OLS. However, when $\lambda \to \infty$ the function becomes insignificant, because the coefficients are shrunk to zero. In practice, the behavior of these methods depends precisely on how the tuning parameter λ is selected via the criterion ($cross - validation$, AIC, etc.).

Ridge regression

The *Ridge regression* shrinks the regression coefficients by forcing a penalty on their size. The Ridge coefficients minimize a penalized residual sum of squares (RSS):

$$\hat{\beta}^{Ridge} = \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2,$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq t$$

Since $\sum_{i=1}^{n} y_i = 0$ and the estimation of $\hat{\beta}_0 = \bar{y}$, the intercept $\hat{\beta}_0 = 0$ and is omitted in the sequence. The same reasoning will be applied in the other methods. Let us note the estimator as follows:

$$\hat{\beta}^{Ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

Despite the fact that it can outperform the LASSO (Tibshirani, 1996) in scenarios such as a regression with a large number of small effects, the *Ridge regression* never cuts a coefficient to zero, making it incompatible with the main drive of our work. Therefore, it will not be simulated.

LASSO

The *LASSO*, is a regularization method that shrinks the coefficients, with the ability to set some exactly to zero by making them sufficiently small.

The LASSO estimator is defined as:

$$\hat{\beta}^{LASSO} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

Its main distinction from *ridge regression* is the form of penalization. Indeed, the LASSO employs an ℓ_1 penalty, whereas ridge utilizes an ℓ_2 constraint. In Figure 4, the blue areas are the constrained regions subject to $\sum_{j=1}^{2} |\beta_j| \leq t$ and respectively $\sum_{j=1}^{2} \beta_j^2 \leq t^2$, and the red ellipses represent the contours of the least squares error function. The ℓ_1 form ensures that coefficients can be set to zero, since the error crosses the axis.

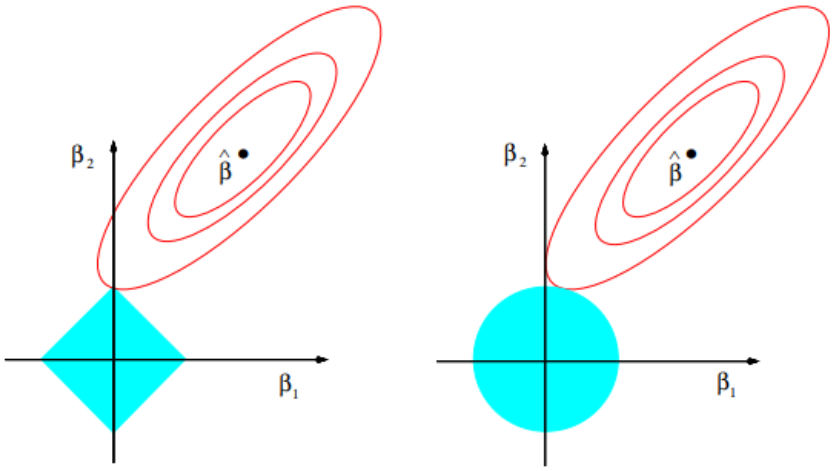

Figure 4: Estimation picture of LASSO (left) and ridge regression (right). Source : Hastie et al. (2009).

Consequently, it performs variable selection and is particularly useful in high-dimensional settings. It performs the best in low signal-to-noise settings (Hastie et al., 2020).

Elastic Net (EN)

The **Elastic Net** regression combines the properties of the two methods mentioned above. It selects variables like the *LASSO* and shrinks together the coefficients of the correlated predictors just as *ridge regression* (Figure 5).

The Elastic Net estimator is defined as:

$$\hat{\beta}^{EN} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda(\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2) \right\}$$

where $\lambda > 0$ controls the overall strength of the penalty and $\alpha \in [0, 1]$ determines the balance between the LASSO and Ridge penalties. When $\alpha = 1$, the Elastic Net estimator reduces to the *LASSO* estimator, whereas for $\alpha = 0$, it corresponds to the *ridge estimator*.

While enjoying a similar sparsity of representation, the empirical and simulation studies show that the *Elastic Net* often outperforms the *LASSO*, especially in cases where $p > n$ (Zou and Hastie, 2005).

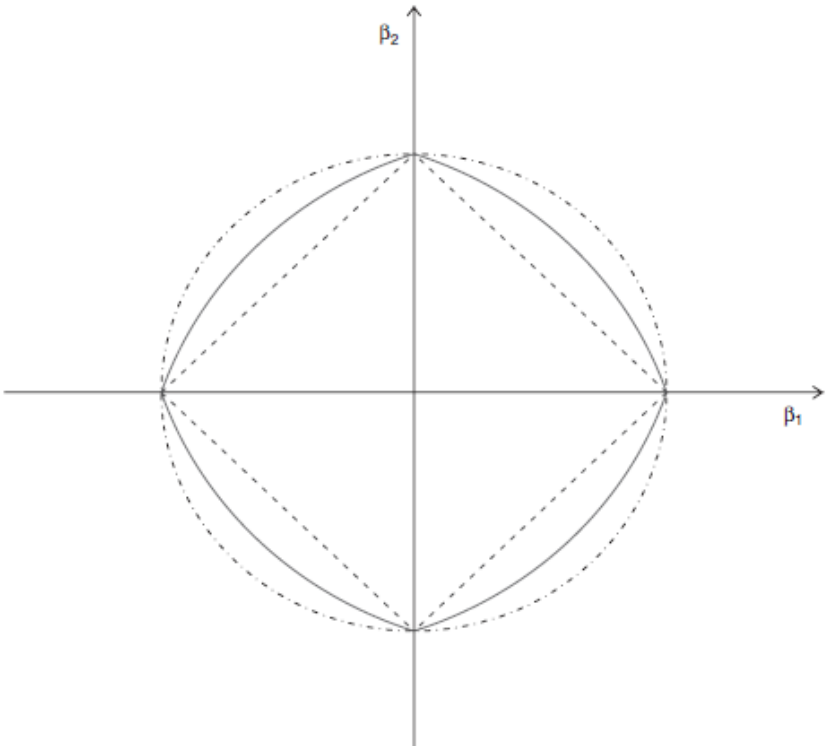

Figure 5: The constrained region of the Elastic Net. In this two-dimensional contour plot, the constrained region of the ridge is delimited by $(- \cdot - \cdot - \cdot -)$, the LASSO penalty is delineated by $(- - - - -)$, and the solid line is the penalty of the Elastic Net. Source : Zou and Hastie (2005).

1.4.3 Path algorithms

Forward Stagewise Regression (FS)

The Forward Stagewise method updates coefficients incrementally. As the *forward-stepwise* regression, it start with an intercept equal to $\bar{y}$, since the predictors are standardized. At each step, the variable most correlated with the residuals is selected, and its coefficient is updated in the direction of the gradient. The algorithm's steps are the following:

Algorithm 1: Forward Stagewise regression

for $i = 1, 2, ..., m$ **do**

1. Initialize with $i = 1$:
$$\beta^{(i)} = (\beta_1^{(i)}, \beta_2^{(i)}, ..., \beta_j^{(i)})^\top = 0, \quad r^i = y - \bar{y}.$$
2. Select x_j, the most correlated variable with the residual r^i, such as $|corr(r^i, x_j)|$ is maximal.
3. Update
$$\beta_j^{(i+1)} = \beta_j^{(i)} + \rho^i, \quad \rho_i = \lambda \cdot \text{sign}(corr(r^i, x_j)),$$
where λ is a fixed positive infinitesimal constant, also called the step size.
4. Update the residuals:
$$r^{i+1} = r^i - \rho^i x_j.$$

The algorithm continues until none of the variables have correlations with the residuals. The core problem of this algorithm is that it fits very slowly. Indeed, unlike *forward-stepwise*, none of the variables adjusts when a new one is added into the model. The corollary is that the computation can take more than p steps. Thus, historically it has been dismissed as being inefficient. Based on this reasoning, we will not model *Forward Stagewise*. Nonetheless, it was showed that a modification to *FS* yields a convergent algorithm for the least squares *LASSO* fit (Freund et al., 2017).

Least Angle Regression (LARS)

Due to the slowness of the *Forward Stagewise* algorithm, the *Least Angle Regression* (Efron et al., 2004) was invented as a better alternative. Although in principle these two path algorithms resemble in their strategies, the main distinction from *FS* is that instead of choosing only one variable, *LARS* "enters as much of a predictor as it deserves". The algorithm is modeled as such:

Algorithm 2: Least Angle Regression

for $i = 1, 2, ..., m$ **do**

1. The predictors are first standardized. Start with $\beta^{(i)} = (\beta_1^{(i)}, \beta_2^{(i)}, ..., \beta_j^{(i)})^\top = 0, \quad$ with $r^i = y - \bar{y}$.
2. Find the predictor x_j most correlated with r^i.
3. Move β_j from zero towards its least-squares coefficients, keeping equal angles with the active predictors, until some other competitor x_k presents as much correlation with r^i as does x_j.
4. Move β_j and β_k in the direction defined by their joint least-squares coefficient of the current residual on (x_j, x_k), until another predictor x_l has as much correlation with the current residual.
5. Continue in this way until all the p predictors have been entered.

The algorithm continues until the least-squares solution is reached. Curiously enough, by construction, *LARS* can be modified to approach the entire *LASSO* path:

Algorithm 3: Least Angle Regression: LASSO modification

1. If a non-zero coefficients becomes null, drop its variable from the current active set of variables and recompute the joint least-squares direction.

1.5 Model selection criteria and stopping rules

1.5.1 Selection procedures and stopping rules in PROC GLMSELECT

In **PROC GLMSELECT**, model selection is controlled by three main arguments: the criterion selection (`select`), the stopping rule (`stop`) and the criterion used to retain the final model (`choose`).

- The `select` argument determines the order in which effects enter or leave at each step of the specified selection method. It is not valid for the *LASSO, Elastic Net* and *LAR* methods, where the solution is defined through a continuous regularization path.
- The `stop` option serves as the "braking" mechanism of the algorithm, since it specifies when the procedure terminates.
- The `choose` argument identifies the model retained as optimal according to the chosen criterion.

1.5.2 Inferential and information-based criteria

The subset selection methods are constructed on inferential criteria. The main problem of these mechanisms is that $F - tests$ and $p - values$ are not valid due to the multiple testing and the adaptive selection. Consequently, $p-value-based$ stopping rules are not designed for prediction, and are expected to perform poorly in model recovery (Cohen, 2006).

In this regard, the information-based criteria offer an alternative, being particularly useful when

comparing different models. The most commonly used measures are the **Akaike Information Criterion** (AIC) (Akaike, 1974) and the **Bayesian Information Criterion** (BIC), which follow the "lowest measure is the best measure" principle.

The **AIC** is given by:

$$\text{AIC} = -2\log(\mathcal{L}) + 2k,$$

where $\mathcal{L}$ denotes the likelihood of the model and k the number of estimated parameters. Thus, this criterion favors models with good predictive performance ($\mathcal{L}$ close to 0), at the cost of a moderate penalization of model complexity (k). This criterion is prediction-oriented, since it favors a lower bias for a higher variance in exchange (Burnham and Anderson (2004)). However, it risks overfitting.

The **BIC** is defined as:

$$\text{BIC} = -2\log(\mathcal{L}) + k\log(n),$$

where n denotes the sample size. Conversely to AIC, the complexity penalty is heftier, as $\log(n)$ is assumed to be greater than 1. Accordingly, the BIC tends to select more parsimonious models, risking underfitting.

1.5.3 Prediction-oriented criteria

Cross-validation methods aim to evaluate the predictive performance of a model on data that were not used during the training. Arguably the most widely used, this criterion directly estimates the expected extra-sample error $Err = \mathbb{E}_{(X,Y)}[L(Y, \hat{f}(X;\mathcal{D}))]$, the average generalization error when $\hat{f}(X;\mathcal{D})$ is applied to an independent test sample.

Since data can be often scarce, we cannot set aside a validation set to assess the performance of the model. In this sense, the **K-fold cross-validation** consists in using a part of the available data to fit the model, and a part to test it.

The data is partitioned into K approximately equal-sized samples. A common split is for example $K = 5$, illustrated in Figure 6. For each fold, one subsample is used as the validation set, while the remaining $K - 1$ subsamples are used for fitting the predictive model. The prediction error is then averaged over the number of folds.

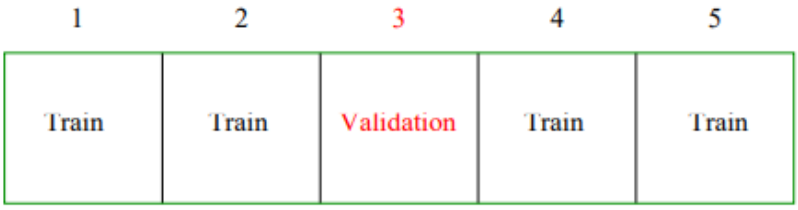

Figure 6: Source : Hastie et al. (2009).

Let us note $k : \{1, ..., K\}$ an indexing function that indicates the partition to which observation i is allocated by the randomization, and $\hat{f}^{-k}(x)$ the fitted model computed with the k^{th} part of the data removed. The cross-validation estimate of the prediction error is defined as such:

$$CV(\hat{f}) = \frac{1}{K} \sum_{k=1}^{K} L(y_i, \hat{f}^{-k(i)}(x)).$$

If we consider that $f(x, \lambda)$ is a set of models indexed by a tuning parameter λ, then the function $CV(\hat{f}, \lambda)$ can provide an estimate of the test error curve, and find the tuning parameter $\hat{\lambda}$ that minimizes it. In practice, this parameter is automatically found by the algorithm.

A particular case of the cross-validation is the **leave-one-out**, which corresponds to a setting where $K = n$. Each observation is successively removed from the sample and used as a validation point. Although this method provides an almost unbiased estimate of the prediction error, it is often computationally expensive and may exhibit high variance.

These criteria reflect structurally different objectives. Whereas information-based measures favor parsimony and interpretability, validation-based criteria focus on predictive accuracy. Thus, the same estimation method can present drastically different behavior depending on the selection rule employed.

In the next section, we will evaluate how these selection criteria impact the conduct of the variable selection and shrinkage methods, under controlled data-generating processes. The performance of each combination will be assessed through their ability to recover the true support, accompanied by the overfitting and underfitting probabilities.

2 Simulations

2.1 The data-generating process

First and foremost, before applying any of the mentioned methods in an empirical experimentation, in this section we will evaluate their behavior in a setting of controlled data-generating processes. The data will be simulated under two hypotheses:

$$\begin{cases} H_0 : \text{The response y is independent of the predictors} \\ H_1 : \text{The response y is linearly dependent of the predictors} \end{cases}$$

$$\iff \begin{cases} H_0 : Y \perp X \\ H_1 : Y = X\beta + \epsilon \end{cases}$$

where:

- $Y \in \mathbb{R}^N$ denotes the vector of the variables we want to predict.
- $X \in \mathbb{R}^{N \times p}$ is the matrix of p predictors.
- $\beta \in \mathbb{R}^p$ is the vector of unknown parameters.
- $\epsilon \in \mathbb{R}^N$ is the vector of error terms, assumed to have zero mean and finite variance.

The base settings

In our simulation, $p = 50$ for $N = 200$ observations, however only 5 variables represent the real effect, wheres the rest is merely noise.

$$\beta^{Real} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_5 \\ \beta_6 \\ \vdots \\ \beta_{20} \\ \vdots \\ \beta_{30} \\ \vdots \\ \beta_{40} \\ \vdots \\ \beta_{50} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0.9 \\ -1 \\ \vdots \\ 1.2 \\ \vdots \\ -0.8 \\ \vdots \\ 1.1 \\ \vdots \\ 0 \end{bmatrix}$$

Consequently, let us set

$$y_i^{True} = x_{5i}\beta_5 + x_{6i}\beta_6 + x_{20i}\beta_{20} + x_{30i}\beta_{30} + x_{40i}\beta_{40}$$

the true underlying structure, and the observed value of y:

$$y^{obs} = y^{True} + \epsilon.$$

Due to its proprieties, notably the facility to manipulate the distribution, the matrix of predictors will follow a multivariate normal law $\boldsymbol{X} \sim \mathcal{N}_p(\boldsymbol{\mu}, \Sigma)$, where Σ is a diagonal matrix of covariance. Moreover, we admit the error homoscedasticity: $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \quad \forall i \in \{1, \ldots, n\}$, and $\sigma^2 = 0.5$.

2.2 The metrics

The model efficiency can be measured through the prediction error $\mathbb{E}\big[(Y - \hat{f}(x; \mathcal{D}))^2 | X, \mathcal{D}\big]$. As depicted earlier, this error can be decomposed in two elements: the bias and the variance. By manipulating the interchange, we can expect to diminish the error.

Nevertheless, in practice, these components cannot be observed. Therefore, a practical approach to assess the model robustness to different case-scenarios is through probabilities. Indeed, by simulating the variables we wish to predict (y) and their predictors (the model matrix X) for

$MC = 300$ times by a Monte Carlo reproduction, we can count the number of occurrences where the true support was either selected or not. By a simple computation of the proportions, we will be able to identify the following events:

P(Exact), which defines the probability that the model recovers exactly the true variables:

$$P(Exact) = \frac{|\text{Exact support}|}{MC}, \quad \text{where FP} = 0 \text{ and FN} = 0$$

P(Overfitting), tracing the probability that the model finds both the entirety of the true variables and at least one noisy variable. This corresponds to a model with good selection but poor precision:

$$P(Overfitting) = \frac{|\text{Overfitted selected support}|}{MC}, \quad \text{where FP} > 0 \text{ and FN} = 0$$

P(Underfitting), which denotes the probability that the model partially finds the true model, without any additional noise:

$$P(Underfitting) = \frac{|\text{Underfitted selected support}|}{MC}, \quad \text{where FP} = 0 \text{ and FN} > 0$$

P(Mixed), outlining the probability that the model recovers variables both from the true support as well as noisy variables. Since mixed models include false exclusions and false inclusions, this case represents the most unstable selection:

$$P(Mixed) = \frac{|\text{Mixed selected support}|}{MC}, \quad \text{where FP} > 0 \text{ and FN} > 0$$

These probabilities must respect the following constraint, otherwise there is an event that was not registered:

$$P(Exact) + P(Overfitting) + P(Underfitting) + P(Mixed) = 1$$

For the rest of this thesis, these probabilities will be used to compare the behavior of the variable selection and shrinkage methods under different stopping rules and scenarios.

2.3 The simulated scenarios

This study deliberately focuses on data-generating processes where the underlying structure is present. Despite the fact that a simulation under the null hypothesis could be very informative for evaluating the false positive (FP) discovery rates, it responds to a different question. Indeed, in such a case there would be no underfitting and the simulations would assess the aggressiveness of the stopping rules rather than the ability of the models to balance the bias and variance. Since the latter requires a signal, and given the already significant set of simulated scenarios, we will restrict the analysis to dependent scenarios alone.

2.3.1 Scenario 1: The simple linear dependency

In this case, we will consider the following structure:

$$y_i = x_{5i}\beta_5 + x_{6i}\beta_6 + x_{20i}\beta_{20} + x_{30i}\beta_{30} + x_{40i}\beta_{40} + \epsilon$$

where the errors are homoscedastic. It will serve as the benchmark - a reference case - where the true underlying signal is relatively easy to recover. The true support is defined by:

$$S = \{5, 6, 20, 30, 40\}$$

and all the remaining coefficients are equal to zero.

In this setting, the main difficulty is the ability of the models to distinguish the pertinent signal from the pure noise, while avoiding the overfitting and the underfitting. Since the values of the true coefficients are close in magnitude, for predictive-oriented criteria we expect average results with high overfitting due to bigger variance, whereas for punitive criteria we anticipate higher underfitting. The inferential criterion ($p - value$) should not work properly.

Les us consider the following results from our simulations:

Table 1: Forward - Linear dependency scenario

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
SL	SL	0	1	0	0
CV	CV	0	1	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.33	0.67	0	0
CV	AIC	0.02333	0.97667	0	0
CV	SBC	0.41333	0.58667	0	0
AIC	SBC	0.37000	0.63000	0	0
AIC	CV	0.050000	0.95000	0	0
SBC	CV	0.056667	0.94333	0	0
SBC	AIC	0	1	0	0

Table 2: Backward - Linear dependency scenario

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
SL	SL	0.01	0.99	0	0
CV	CV	0	1	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.30333	0.69667	0	0
CV	AIC	0	1	0	0
CV	SBC	0.31333	0.68667	0	0
AIC	CV	0	1	0	0
SBC	CV	0	1	0	0
SBC	AIC	0	1	0	0
AIC	SB	0.30333	0.69667	0	0

Table 3: Stepwise - Linear dependency scenario

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
SL	SL	0	1	0	0
CV	AIC	0.02	0.98	0	0
CV	SBC	0.42	0.58	0	0
SBC	AIC	0.32	0.68	0	0
CV	CV	0	1	0	0
AIC	CV	0.07	0.93	0	0
AIC	SBC	0.33667	0.66333	0	0
AIC	AIC	0.00000	1.00000	0	0
SBC	CV	0.34000	0.66000	0	0
SBC	SBC	0.35667	0.64333	0	0

Table 4: LASSO - Linear dependency scenario

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
CV	CV	0.1	0.9	0	0
AIC	AIC	0.123	0.583	0.293	0
SBC	SBC	0.147	0.213	0.64	0
CV	AIC	0.123	0.623	0.253	0
CV	SBC	0.157	0.26	0.583	0
AIC	CV	0.09	0.91	0	0
AIC	SBC	0.197	0.287	0.517	0
SBC	CV	0.117	0.883	0	0
SBC	AIC	0.113	0.62	0.267	0

Table 5: Elastic net - Linear dependency scenario

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
CV	CV	0.073	0.927	0	0
AIC	AIC	0.1	0.7	0.2	0
SBC	SBC	0.137	0.293	0.57	0
CV	AIC	0.09	0.66	0.25	0
CV	SBC	0.153	0.317	0.53	0
AIC	CV	0.09	0.91	0	0
AIC	SBC	0.123	0.297	0.58	0
SBC	CV	0.077	0.923	0	0
SBC	AIC	0.07	0.707	0.223	0

Table 6: LARS - Linear dependency scenario

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixte)**
CV	CV	0.13	0.87	0	0
AIC	AIC	0.1	0.653	0.247	0
SBC	SBC	0.16	0.32	0.52	0
CV	AIC	0.13	0.633	0.237	0
CV	SBC	0.173	0.26	0.567	0
AIC	CV	0.107	0.893	0	0
AIC	SBC	0.173	0.26	0.567	0
SBC	CV	0.08	0.92	0	0
SBC	AIC	0.113	0.637	0.250	0

Although this *trial and error* may seem confusing, it is particularly useful to observe the way these mechanisms adapt according to their stopping rules.

An initial observation would be the fact that overall, every method has a null probability to select a mixed support. In truth, it is expected due to the fact that the model noise is not particularly strong, and the all the true coefficients are not too low, which helps its identifiability.

The variable selection methods are especially dominated by the variance due to their discrete nature, generating instability and altogether very high overfitting. As expected, forward and backward stepwise perform very poorly under an inferential stopping rule, namely because of the multiple testing issue. However, there is no underfitting: all methods find the good support, but add irrelevant noisy predictors.

As we can see, criteria such as AIC or the cross-validation systematically lead to extreme overfitting. This behavior is expected, since these criteria are deliberately favoring low bias at the expense of higher variance. In contrast, punitive criteria such as the SBC (BIC) significantly improve the ability to recover the exact true support, since it penalizes model complexity. That being said, even under this favorable setting, exact recovery remains far from systematic, rarely exceeding 40% depending on the procedure.

The shrinkage methods (LASSO, EN and LARS) reveal a different behavior. Unlike subset selection, these methods explicitly manage the bias-variance tradeoff through regularization, resulting in the apparition of significant underfitting probabilities when restrictive criteria are used. Indeed, SBC notably reduces the overfitting, but often does so at the cost of excluding pertinent variables. Conversely, cross-validation leads to higher variance, hence the increased inclusion of irrelevant predictors. Among the methods, LARS seems to marginally present the best results, although no procedure consistently dominates depending the criteria.

Curiously enough, even in this favorable situation, none of the procedures consistently achieve exact recovery. This proves a fundamental limitation: the permissive criteria tends to overfit, while the restrictive criteria tends to underfit, and no universal choice of stopping rules will lead to consistent model recovery.

This highlights the difficulty of variable selection in high-dimensional settings and motivates the analysis of more complex data-generating processes, where variance-inducing mechanisms, such as the tendency break, the multicollinearity and the extreme values will further stress test the bias-variance tradeoff.

2.3.2 Scenario 2: Linear dependency with a tendency break

In this scenario, the structural break will be introduced on the regression coefficients, inducing non-stationarity in the data-generating process.

$$\begin{cases} y_i = x_{5i}\beta_5^{(1)} + x_{6i}\beta_6^{(1)} + x_{20i}\beta_{20}^{(1)} + x_{30i}\beta_{30}^{(1)} + x_{40i}\beta_{40}^{(1)} + \epsilon, & \text{for i} \leq \text{t} \\ y_i = x_{5i}\beta_5^{(2)} + x_{6i}\beta_6^{(2)} + x_{20i}\beta_{20}^{(2)} + x_{30i}\beta_{30}^{(2)} + x_{40i}\beta_{40}^{(2)} + \epsilon, & \text{for i} > \text{t} \end{cases}$$

where t=0.5, meaning that halfway through the period there happened a tendency break. It is noteworthy to mention that the break effect is not potent. The true support still remains:

$$S = \{5, 6, 20, 30, 40\}$$

However, their magnitude is not the same anymore, meaning that some coefficients will be easier to spot due to their increase, whereas other might become less significant and harder to observe. Although the variables remain relevant in the structure, the escalated variance due to non-stationarity should decrease the effectiveness of the chosen criteria. We anticipate a significant increase in overfitting.

Table 7: Forward - Tendency break

Choose	**Stop**	**P(Exact)**	**P(Overfit)**	**P(Underfit)**	**P(Mixed)**
CV	CV	0	1	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.34667	0.65333	0	0
AIC	SBC	0.3	0.7	0	0
AIC	CV	0.05	0.95	0	0
SBC	AIC	0	1	0	0

Table 8: Backward - Tendency break

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
SL	SL	0.01	0.99	0	0
CV	CV	0	1	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.29667	0.70333	0	0
CV	AIC	0.03	0.97	0	0
CV	SBC	0.47667	0.52333	0	0
AIC	CV	0.06667	0.93333	0	0
AIC	SBC	0.36333	0.63667	0	0
SBC	CV	0.1	0.9	0	0
SBC	AIC	0.00333	0.99667	0	0

Table 9: Stepwise - Tendency break

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
AIC	SBC	0.32667	0.67333	0	0
SBC	AIC	0	1	0	0
AIC	CV	0	1	0	0
CV	AIC	0	1	0	0
CV	SBC	0.21	0.79	0	0
AIC	CV	0	1	0	0
AIC	SBC	0.30333	0.69667	0	0
SBC	CV	0	1	0	0
SBC	AIC	0	1	0	0

Table 10: LASSO - Tendency break

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.11	0.89	0	0
AIC	AIC	0.12667	0.67333	0.2	0
SBC	SBC	0.19667	0.4	0.40333	0
CV	AIC	0.10667	0.68	0.21333	0
CV	SBC	0.19333	0.31	0.49667	0
AIC	CV	0.11	0.89	0	0
AIC	SBC	0.22	0.42	0.36	0
SBC	CV	0.1	0.9	0	0
SBC	AIC	0.16	0.68667	0.15333	0

Table 11: Elastic Net - Tendency break

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.08	0.92	0	0
AIC	AIC	0.1	0.77667	0.12333	0
SBC	SBC	0.17	0.39333	0.43667	0
CV	AIC	0.11333	0.75	0.13667	0
CV	SBC	0.15333	0.34667	0.5	0
AIC	CV	0.10333	0.89667	0	0
AIC	SBC	0.19333	0.38667	0.42	0
SBC	CV	0.07333	0.92667	0	0
SBC	AIC	0.08	0.72667	0.19333	0

Table 12: LARS - Tendency break

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.11333	0.88667	0	0
AIC	AIC	0.15667	0.66	0.18333	0
SBC	SBC	0.25	0.32	0.43	0
CV	AIC	0.15333	0.67333	0.17333	0
CV	SBC	0.22	0.25667	0.52333	0
AIC	CV	0.11667	0.88333	0	0
AIC	SBC	0.19333	0.35	0.45667	0
SBC	CV	0.14	0.86	0	0
SBC	AIC	0.11667	0.71	0.17333	0

With these results, we can say that the coefficients of the explanatory variables have generally increased rather than decreased in magnitude.
In fact, for all methods without exception, we observed an increase in overfitting and a decrease in underfitting.
Another reason could be the adjustment capacity of the methods: the presence of a tendency break accentuates the fluctuations of the different explanatory variables, and methods confronted with this type of scenario tend to overfit their estimates in order to capture the effects on Y. In short, the methods seek to adjust to the excessive complexity of the model, which is in fact "simulated" by the break. They therefore select the right variables, but at the expense of excessive overfitting.

Since overfitting has increased overall, we will instead focus on the performance of restrictive models and criteria.
Among the results, we can see that the LASSO method and the SBC criterion provide us with the best exact model probabilities, but at the cost of higher underfitting. It makes sense to compensate for variance with restrictive methods and criteria that will limit variable selection, even if in some cases underfitting reaches quite significant levels of up to 50%.

As for LARS, it is difficult to determine why it has the best predictions among all the models, but here is one explanation that could justify this:
LARS is an algorithm that enters variables one by one into its estimation model according to their linear coefficients. However, when there is a tendency break, these coefficients change along the way, but LARS does not take them into account and continues on its linear path to find its variables. Thus, we can assume that the break affects it significantly less than the other methods, which explains its performance in the presence of such a tendency break.

2.3.3 Scenario 3: Linear dependency with multicollinearity

In this plan, the model matrix $\boldsymbol{X} \sim \mathcal{N}_p(\boldsymbol{\mu}, \Sigma)$, but instead Σ is not diagonal anymore. Though unrealistic, we will apply the Toeplitz method: variables with nearby indices will be strongly correlated, whereas distant predictors will be nearly independent. In order to mathematically illustrate this, consider the following matrix for four variables:

$$\begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$$

where ρ is the correlation between two variables. The farther apart the predictors are from each other, the smaller is the statistical association. The true support stands as follows:

$$S = \{5, 6, 20, 30, 40\}.$$

Due to the correlation between the predictors, we anticipate a slight increase in model instability, taking form in mixed selected supports.

Let us consider the following results:

Table 13: Forward - Multicollinearity

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
CV	CV	0	1	0	0
CV	AIC	0.05667	0.94333	0	0
CV	SBC	0.53667	0.46333	0	0
AIC	CV	0.1	0.9	0	0
AIC	SBC	0.46667	0.53333	0	0
SBC	CV	0.10333	0.89667	0	0
SBC	AIC	0.01	0.99	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.44333	0.55667	0	0

Table 14: Backward - Multicollinearity

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
CV	CV	0	1	0	0
CV	AIC	0	1	0	0
CV	SBC	0.22333	0.77667	0	0
AIC	CV	0	1	0	0
AIC	SBC	0.32	0.68	0	0
SBC	CV	0	1	0	0
SBC	AIC	0	1	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.330	0.67	0	0

Table 15: Stepwise - Multicollinearity

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0	1	0	0
CV	AIC	0.06	0.94	0	0
CV	SBC	0.53333	0.46667	0	0
AIC	CV	0.12	0.88	0	0
AIC	SBC	0.45667	0.54333	0	0
SBC	CV	0.45667	0.54333	0	0
SBC	AIC	0.49667	0.50333	0	0
AIC	AIC	0.01333	0.98667	0	0
SBC	SBC	0.50667	0.49333	0	0

Table 16: LASSO - Multicollinearity

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.22	0.583	0.147	0.05
AIC	AIC	0.013	0.827	0.13	0.03
SBC	SBC	0.013	0.603	0.333	0.05
CV	AIC	0.003	0.853	0.11	0.033
CV	SBC	0.033	0.633	0.273	0.06
AIC	CV	0.263	0.55	0.127	0.06
AIC	SBC	0.03	0.593	0.317	0.06
SBC	CV	0.21	0.617	0.127	0.047
SBC	AIC	0.013	0.84	0.12	0.027

Table 17: Elastic Net - Multicollinearity

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.22	0.627	0.083	0.07
AIC	AIC	0	0.927	0.073	0
SBC	SBC	0.013	0.767	0.2	0.02
CV	AIC	0	0.933	0.067	0
CV	SBC	0.003	0.793	0.18	0.023
AIC	CV	0.173	0.64	0.117	0.07
AIC	SBC	0.027	0.753	0.203	0.017
SBC	CV	0.2	0.603	0.143	0.053
SBC	AIC	0.003	0.91	0.087	0

Table 18: LARS - Multicollinearity

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.213	0.58	0.143	0.063
AIC	AIC	0.003	0.81	0.157	0.03
SBC	SBC	0.03	0.67	0.253	0.047
CV	AIC	0.017	0.82	0.12	0.043
CV	SBC	0.01	0.61	0.317	0.063
AIC	CV	0.197	0.58	0.167	0.057
AIC	SBC	0.01	0.65	0.28	0.06
SBC	CV	0.253	0.52	0.15	0.077
SBC	AIC	0.01	0.833	0.097	0.06

Compared to the benchmark with the independent predictors, this setting affects the variance of the estimators and the stability of the selected supports due to higher variance.

Concerning the subset selection, the behavior observed in the benchmark largely persists: overfitting remains the dominant outcome, whereas the underfitting is absent. This confirms that the multicollinearity does not prevent the variable selection methods to find the pertinent predictors, but prevents the ability of discrete procedures to exclude the irrelevant ones.

We can see that punitive criteria such as SBC slightly improve the probability of exact recovery for forward and stepwise procedures, occasionally exceeding 50%. However, this must be interpreted with caution: the results are extremely sensitive to the stopping rules, and small changes will lead to sharp variations. Backward selection remains particularly instable and performs poorly overall.

In contrast with the subset selection, the shrinkage methods present a more informative behavior. Just as in the benchmark, the LASSO, EN and LARS do not concentrate their error solely in the overfitting, but instead distribute it across the underfitted as well as the mixed selections. The emergence of non-negligible mixed probabilities reflects the difficulty to identify the real predictors when the variables are correlated. This is especially expected of Elastic net: since it regulates like LASSO but shrinks variables in groups as Ridge, the correlations act as substitutes, leading to the partial recovery of the support. Moreover, as anticipated, we can observe that Elastic Net slightly outperforms LASSO, since it has less underfitting and a better tendency to recover the exact support. Though the overfitting is superior to LASSO, the penalty zone of Elastic Net explicitly tackles correlated variables in groups, leading to the inclusion of irrelevant predictors. (Schreiber-Gregory (2018), Zou and Hastie (2005))

Interestingly enough, none of the methods achieve consistent recovery. It turns out that due to the variable correlations, punitive criteria such as SBC will strongly penalize the complexity, excluding by construction even the suitable predictors. On the other hand, permissive criteria will further accentuate the complexity, considering the fact that even the non-proper coefficients will be counted.

On this account, multicollinearity increases the inherent model instability and leads to the emergence of mixed selections for the regularization methods. Although Elastic Net and LARS demonstrated relatively better robustness, the results prove that multicollinearity fundamentally limits the identifiability of the true support, independently of the employed method.

2.3.4 Scenario 4: Linear dependency with outliers

We will generate outliers through a gaussian process, where:

$$\begin{cases} \epsilon^{(0)}, & \text{with a probability of } 95\% \\ \epsilon^{Outliers}, & \text{with a probability of } 5\% \end{cases}$$

where $\epsilon^{Outliers}$ is an amplified noise. The new values, also called "innovative" outliers, influence solely the response Y, generating measurement errors. The true support matches the original:

$$S = \{5, 6, 20, 30, 40\}$$

Without altering the design matrix, we can isolate the effect of the high-magnitude noise on the estimations, and assess the ability of the methods to distinguish the true signal among spurious variance.

Let us observe the following simulation results:

Table 19: Forward - Outliers

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
CV	CV	0	1	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.32667	0.67333	0	0
CV	AIC	0.04333	0.95667	0	0
CV	SBC	0.47	0.53	0	0
AIC	CV	0.10333	0.89667	0	0
AIC	SBC	0.34333	0.65667	0	0
SBC	CV	0.08	0.92	0	0
SBC	AIC	0.00333	0.99667	0	0

Table 20: Backward - Outliers

Choose	**Stop**	**P(Exact)**	**P(Overfitting)**	**P(Underfitting)**	**P(Mixed)**
CV	CV	0	1	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.30667	0.69333	0	0
CV	AIC	0	1	0	0
CV	SBC	0.26667	0.73333	0	0
AIC	CV	0	1	0	0
AIC	SBC	0.31333	0.68667	0	0
SBC	CV	0	1	0	0
SBC	AIC	0	1	0	0

Table 21: Stepwise - Outliers

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0	1	0	0
AIC	AIC	0	1	0	0
SBC	SBC	0.31	0.69	0	0
CV	AIC	0.05	0.95	0	0
CV	SBC	0.44667	0.55333	0	0
AIC	CV	0.11	0.89	0	0
AIC	SBC	0.34	0.66	0	0
SBC	CV	0.36	0.64	0	0
SBC	AIC	0.32333	0.67667	0	0

Table 22: LASSO - Outliers

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.163	0.837	0	0
AIC	AIC	0.12	0.593	0.287	0
SBC	SBC	0.123	0.223	0.653	0
CV	AIC	0.123	0.563	0.313	0
CV	SBC	0.173	0.25	0.577	0
AIC	CV	0.173	0.827	0	0
AIC	SBC	0.15	0.237	0.613	0
SBC	CV	0.163	0.837	0	0
SBC	AIC	0.097	0.647	0.257	0

Table 23: Elastic Net - Outliers

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.12333	0.87667	0	0
AIC	AIC	0.08	0.66333	0.25667	0
SBC	SBC	0.11333	0.27	0.61667	0
CV	AIC	0.05	0.68	0.27	0
CV	SBC	0.13	0.25	0.62	0
AIC	CV	0.11	0.89	0	0
AIC	SBC	0.15	0.26	0.59	0
SBC	CV	0.13333	0.86667	0	0
SBC	AIC	0.05333	0.69	0.25667	0

Table 24: LARS - Outliers

Choose	Stop	P(Exact)	P(Overfitting)	P(Underfitting)	P(Mixed)
CV	CV	0.20667	0.79333	0	0
AIC	AIC	0.12333	0.56667	0.31	0
SBC	SBC	0.11667	0.27333	0.61	0
CV	AIC	0.10667	0.61667	0.27667	0
CV	SBC	0.15333	0.23	0.61667	0
AIC	CV	0.19	0.81	0	0
AIC	SBC	0.16667	0.27	0.56333	0
SBC	CV	0.17667	0.82333	0	0
SBC	AIC	0.14	0.55333	0.30667	0

Unlike the multicollinearity, the core challenge of this scenario is not the support identifiability, but rather the artificial inflation of the variance of the estimators. As such, this case represents a stress test of the model robustness of selection and regularization methods to noise pollution.

Comparing to the benchmark, the discrete selection (forward, backward and stepwise) is accompanied by systematic overfitting. As it can be observed, the variance induced by the outliers worsens the already known effects of permissive criteria such as AIC and cross-validation. Because of the increased complexity, the overfitting probability is close to one, while the exact recovery close to zero. Conversely, strongly penalizing rules such as SBC-SBC have a positive effect on their ability of identifying the true underlying structure, resulting in a 30% average success rate. Nonetheless, the high overfitting proportion indicates the fact that SBC acts rather as a brake from the complexity than a solution.

Shrinkage methods display different patterns. Contrarily to the benchmark, the overfitting is no longer systematic, and is now shared along with the underfitting. This reflects an interesting bias-variance tradeoff: since the regularization methods dampen the extreme observations due to their variance, the bias is increased. With prediction-oriented criteria such as the cross-validation, the LASSO, Elastic Net and LARS still overfit, but in a lesser manner than the subset selection methods. Conversely, restrictive criteria such as SBC lead to dominant underfitting, indicating that the combination of strong penalties will exclude genuinely relevant predictors. LARS seems to demonstrate the best relative robustness due to the higher probabilities of exact recovery.

An interesting observation is the absence of mixed probabilities: the outliers do not introduce substitutability between predictors.

We can conclude that the presence of extreme values systematically degrades the performance of the cited methods. While the benchmark already shows strong tendencies to overfitting, the outliers amplifies this effect in the discrete selection, and shifts the error toward underfitting for shrinkage approaches.

These methods do not seem to be robust enough.

3 Empirical application

3.1 Purpose of the research

For the moment, our theoretical hypotheses have only been validated using simulated data. This framework, which we were able to control, allowed us to precisely define the scenarios studied in order to evaluate the performance of the different methods and criteria, thanks to the interpretation of bias and variance via the calculation of overfitting and underfitting probabilities in relation to the true model. It is now necessary to test these hypotheses against real data in order to evaluate the robustness of the methods studied and draw conclusions that are valid in practice.

First, we will provide a detailed description of the dataset. We will then analyse its structure to determine the type of scenario we are dealing with, and then apply the methods and criteria that will yield the best estimates in this context.

3.2 Diabetes Data

For this part, we will use a dataset that was first used by Efron et al. in their paper on the algorithm of *Least Angle Regression* (2004) and later used by many other authors. It is presented in the form below,

Table 25: Diabetes Data

Patient	**AGE**	**SEX**	**BMI**	**BP**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**Y**
1	59	2	32.1	101	157	93.2	38	4	4.8598	87	151
2	48	1	21.6	87	183	103.2	70	3	3.8918	69	75
3	72	2	30.5	93	156	93.6	41	4	4.6728	85	141
...	...	...	...	...	...	...	...	...	...	...	
442	36	1	19.6	71	250	133.2	97	3	4.5951	92	57

The dataset contains a sample size of $n = 442$ for 10 explanatory variables and Y the explained variable, which represents the progression of the disease in a patient one year after the values were measured.
Among all the explanatory variables, SEX is the only categorical variable with a value of 1 for male and 2 for female; the rest are quantitative variables. The variables can be divided into three distinct categories:

- **Demographic variables** :
  - `AGE` : the patient's age;
  - `SEX` : the patient's gender (qualitative variable, numerical).
- **medical / physiological variables** :
  - `BMI` : body mass index;
  - `BP` : mean arterial pressure.
- **Biological variables / health indicators** :
  - `S1` : total cholesterol (TC);
  - `S2` : low-density lipoproteins (LDL);

- `S3` : high-density lipoproteins (HDL);
- `S4` : total cholesterol to HDL ratio (TC/HDL);
- `S5` : logarithm of triglycerides;
- `S6` : blood sugar level.

3.3 Data structure

To find the method with the best predictive performance on this data, we must first determine its nature, i.e. identify the different scenarios we studied previously, namely multicollinearity, breaks and extreme values.

Multicollinearity

The most likely scenario in our case is a high degree of correlation between certain variables. Indeed, the nature of the variables suggests that we are dealing with structural correlation. This intuition comes from the group of biological variables:

- Closely related metabolic data
- S4 = S1 / S3 with an approximation
- S1 approximately equal to S2 + S3

This hypothesis is confirmed by analysing the data:

Table 26: Pearson correlation matrix of explanatory variables ($N = 442$)

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
AGE	1.000	0.173	0.185	0.335	0.260	0.219	-0.075	0.204	0.271	0.302
SEX	0.173	1.000	0.088	0.241	0.035	0.143	-0.379	0.332	0.150	0.208
BMI	0.185	0.088	1.000	0.395	0.250	0.261	-0.367	0.414	0.446	0.389
BP	0.335	0.241	0.395	1.000	0.242	0.186	-0.179	0.258	0.393	0.390
S1	0.260	0.035	0.250	0.242	1.000	0.897	0.052	0.542	0.516	0.326
S2	0.219	0.143	0.261	0.186	0.897	1.000	-0.196	0.660	0.318	0.291
S3	-0.075	-0.379	-0.367	-0.179	0.052	-0.196	1.000	-0.738	-0.399	-0.274
S4	0.204	0.332	0.414	0.258	0.542	0.660	-0.738	1.000	0.618	0.417
S5	0.271	0.150	0.446	0.393	0.516	0.318	-0.399	0.618	1.000	0.465
S6	0.302	0.208	0.389	0.390	0.326	0.291	-0.274	0.417	0.465	1.000

There is indeed a fairly strong correlation between the biological variables, particularly between variables:

- S1 and S2
- S3 and S4

This confirms our hypotheses about the relationships between these variables.

Tendency break

The dataset is not time-series data and does not provide any information on the existence of a threshold that could cause a break. Nevertheless, we propose applying a Chow test, which could detect a possible break and, if so, identify its origin and guide us in its interpretation.

When we plot the smoothing curves between Y and each other variable, we see that all variables have a linear relationship with Y. Based on these curves, the variable most likely to exhibit a break is the BMI variable.

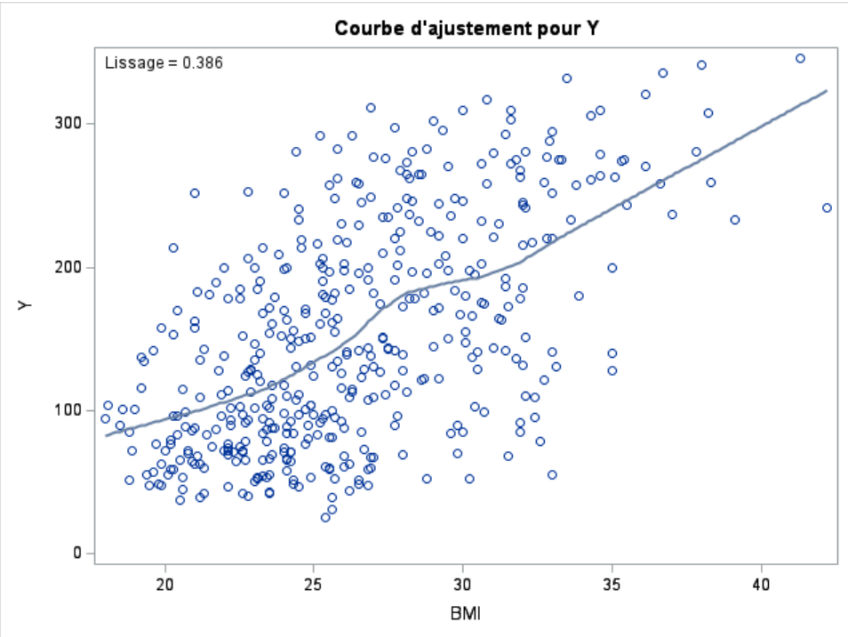

Figure 7: Y / BMI relation

Furthermore, when performing a simple regression on the data, we see that BMI is the most significant variable in the model. This is another reason to test for a break.
We will therefore perform a Chow test on the following model:

$$\iff \begin{cases} Y_{1,n_1} = BMI_{1,n_1}\beta_1 + \epsilon_{1,n_1} \\ Y_{n_2,N} = BMI_{n_2,N}\beta_2 + \epsilon_{n_2,N} \end{cases}$$

Since we do not know the exact breakpoint, we will set n_1 and n_2 to 1/3 and 2/3 of the observations.

Thus, after sorting the BMI observations in ascending order, the following test is performed:

Hypothesis:

$$\begin{cases} H_0 : \beta_1 = \beta_2 \\ H_1 : \beta_1 \neq \beta_2 \end{cases}$$

Test statistic under H_0:

$$F = \frac{(SCR_C - (SCR_1 + SCR_2))/k}{(SCR_1 + SCR_2)/(n_1 + n_2 - 2k)} \sim \mathcal{F}\big(k,\ n_1 + n_2 - 2k\big).$$

Decision rule:

At threshold $\alpha = 5\%$:

$$\text{We reject } H_0 \quad \text{if} \quad F > F_{k,\ n_1+n_2-2k}(\alpha),$$

with $Q_F(1-\alpha)$ the $1-\alpha$ quantile of $F \sim \mathcal{F}\big(k,\ n_1+n_2-2k\big)$.

The results are as follows:

Table 27: Test results

F-value	**Pr > F**
1.40	0.2481

Since the p-value is greater than 5%, we do not reject H_0 and conclude that there is no break for the BMI variable. Since this was the variable most likely to break, we therefore consider that there is no significant break in our overall model.

Extreme Values

To check for extreme values, we will simply analyse the distribution of the different variables in search of outliers.
The only distribution containing outliers is that of S4 (total cholesterol to HDL ratio):

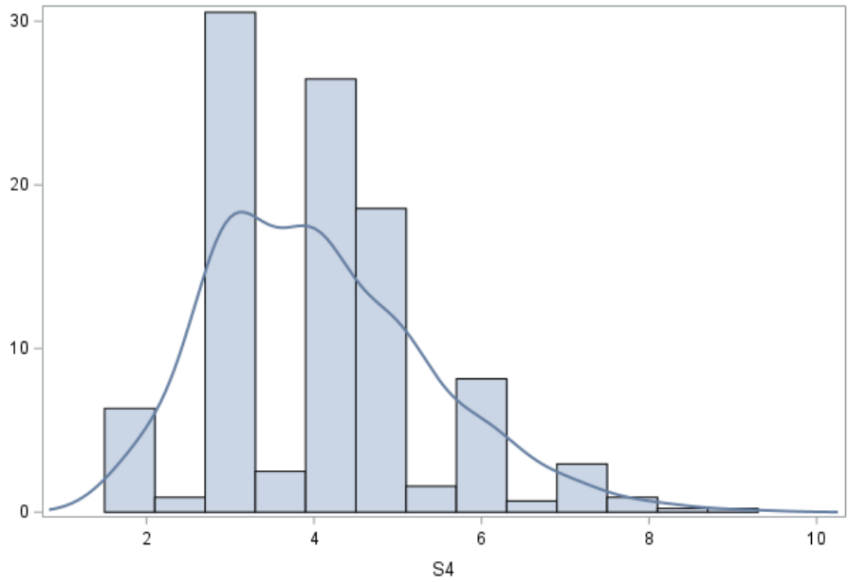

Figure 8: Distribution of S4

This is surely due to the calculation of the variable, since it represents the ratio of total cholesterol to HDL (if total cholesterol is high and HDL is low, there will be an accentuated effect on the S4 ratio). It should also be noted that the S4 variable is calculated in a strange way: a large part of its observations are approximations of the ratio, which makes the variable difficult to interpret.

3.4 Model estimation

Based on the analysis procedure carried out previously, we hypothesise that the variables are strongly correlated. As regards potential breaks and extreme values, these are not significant enough to be included in our study. We are therefore leaning towards the Elastic Net method, which is better suited to this type of scenario. We will nevertheless compare it with other

methods, which are not, a priori, the most effective in this context, in order to confirm our hypotheses. For the criteria, we will favour prediction criteria such as cross-validation, which is more robust in this type of scenario and can therefore be a good criterion for choosing the most effective final model. Another criterion that could prove effective with Elastic Net is BIC: the Elastic Net method tends to keep all variables correlated, and the BIC stop criterion will limit this overfitting effect.

Assessment method

During our simulations, we simply compared our estimate to the actual model to determine the predictive performance of a method. Since we cannot use this approach on an empirical model, we are forced to divide our observations into two groups:
a training group, on which we will perform our estimation
a validation group, which will allow us to determine the accuracy of our estimation.

Metric

Without knowledge of the true model, it is impossible for us to calculate the probabilities of overfitting and underfitting. We therefore decide to use ASE on the validation data, which is a good indicator of predictive performance.

$$\text{ASE} = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

3.5 Results

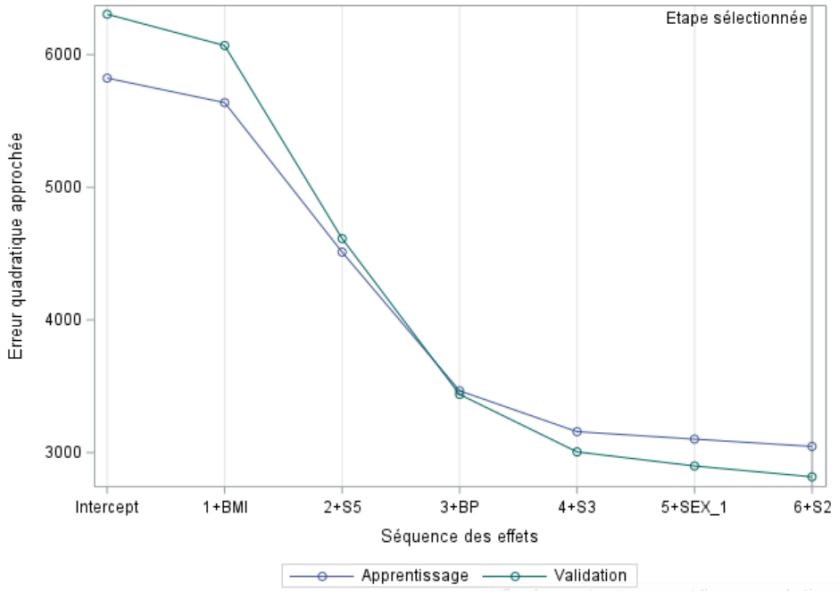

Figure 9: Elastic Net with choose=CV and stop=SBC

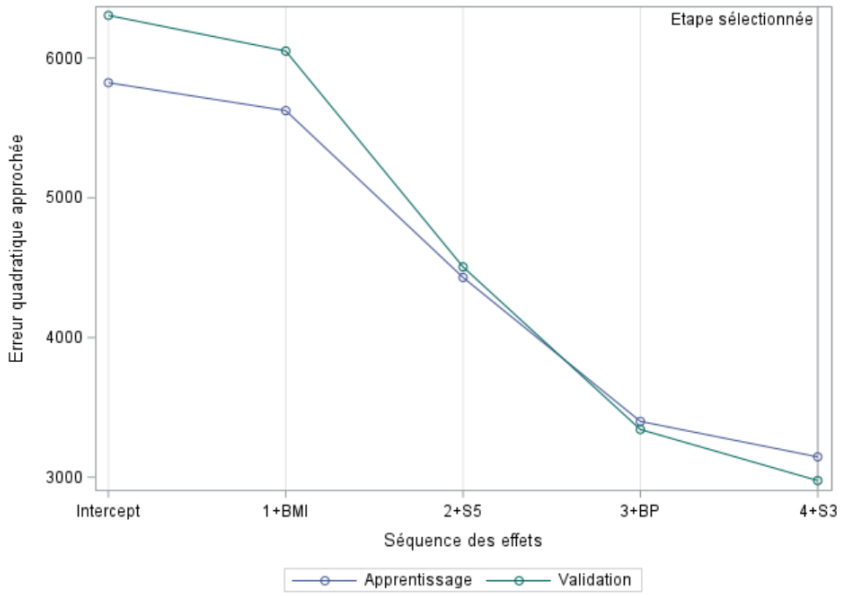

Figure 10: LASSO with choose=CV and stop=SBC

By comparing the LASSO and Elastic Net methods using the same criteria, we can see that LASSO stops earlier in its variable selection, resulting in a higher ASE than Elastic Net, which, in comparison, selects two additional variables.

By replacing the BIC stopping criterion with AIC, we obtain similar comparisons:

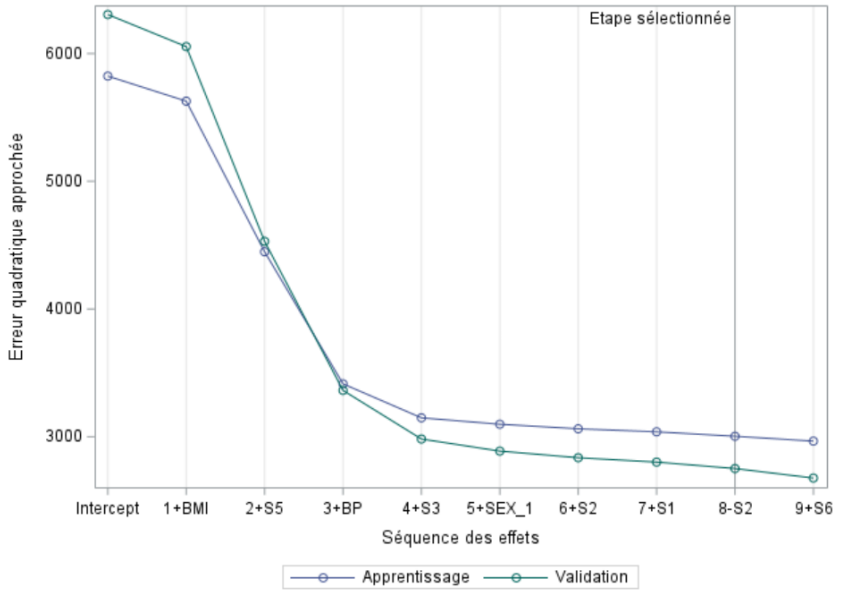

Figure 11: Elastic Net with choose=CV and stop=AIC

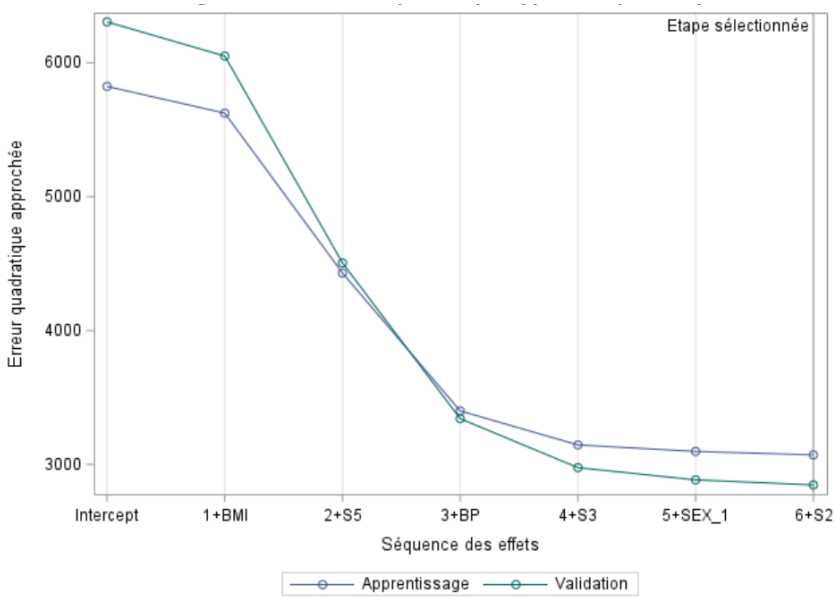

Figure 12: LASSO with choose=CV and stop=AIC

Once again, we have a more advanced selection of variables and a lower ASE with Elastic Net compared to LASSO.

However, we may wonder whether this excessive selection of variables will not introduce too much variance into the Elastic Net estimate, a variance that is invisible on this graph but which could be observed with a Monte Carlo simulation if we had enough observations. We will consider that this hypothesis has already been validated by our simulations: we have indeed observed a significant increase in overfitting when comparing the ElasticNet method to LASSO across various scenarios.

Conclusion and discussions

By conducting our study on the bias-variance trade off, its link to overfitting and underfitting, variable selection methods, and variable selection and stopping criteria, this thesis has allowed us to highlight the unstable nature of prediction models.
In the context of simple linear dependence, selection and shrinkage methods and selection and stopping criteria generally behave as expected in theory. However, as models become more complex, particularly in scenarios such as linear dependence with trend breaks, the presence of extreme values, and internal and external multicollinearity (correlation between the explanatory variables of the true model and variables that are not included in it), their behavior differs.

In most cases, the models tended to over-select or under-select variables. It is at these moments that the selection and stopping criteria are most decisive: restrictive explanatory criteria allow us to limit over-selection, while more flexible explanatory criteria allow us to do the opposite. Predictive criteria, while not excelling in any particular scenario, still managed to provide a certain robustness to the models to cope with all scenarios.
It should be noted, however, that in complex scenarios, regardless of the method and criteria used, there is a significant deterioration in estimates, proving that statistical learning and machine learning models still have limitations.

The empirical application to our *Diabetes* dataset also showed us several things.
Firstly, although our models were theoretically validated by our simulations, the same cannot be said for real-life cases. Even after determining the structure of our *Diabetes* dataset, namely high multicollinearity, we found it difficult to obtain satisfactory results. The prediction quality of Elastic Net, which is supposed to be suited to this type of situation, was only slightly better than that of LASSO.
We also found that the choice of selection and stopping criteria had a significant effect, and that it is therefore important to understand the data structure and the method used in order to obtain satisfactory performance.

In conclusion, this master's thesis highlights the shortcomings of various statistical learning and machine learning models, and the fact that there is no universal method for obtaining accurate predictions. Interpreting and understanding data, as well as having a good knowledge of the tools used, is key to building an accurate and stable prediction model for data that is becoming increasingly complex in a world now centered on AI and data.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6):2350–2383.

Burnham, K. P. and Anderson, D. R. (2004). Understanding aic and bic in model selection. *Sociological Methods & Research*, 33(2):261–304.

Cohen, R. A. (2006). Introducing the GLMSELECT procedure for model selection. In *Proceedings of the Thirty-First Annual SAS Users Group International Conference*, pages 4770–4792.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Fortmann-Roe, S. (2012). Understanding the bias–variance tradeoff. https://scott.fortmann-roe.com/docs/BiasVariance.html. Accessed: January 21, 2026.

Freund, R. M., Grigas, P., and Mazumder, R. (2017). Incremental forward stagewise regression: Computational complexity and connections to the LASSO. *SIAM Journal on Optimization*, 27(3):1702–1746.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York.

Hastie, T., Tibshirani, R., and Tibshirani, R. (2020). Best subset, forward stepwise or LASSO? analysis and recommendations based on extensive comparisons. *Statistical Science*, 35(4):579–592.

Schreiber-Gregory, D. N. (2018). Ridge regression and multicollinearity: An in-depth review. *Model Assisted Statistics and Applications*, 13(4):359–365.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288.

Wahba, G. and Wold, S. (1975). A completely automatic french curve: Fitting spline functions by cross-validation. *Communications in Statistics – Theory and Methods*, 4(1):1–17.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

4 Annex

Listing 1: Code Simulation

```
proc iml;

/* Si l'on veut Y indpendant des X */
start independance(n_samples);
    /* Y est juste du bruit pur, indpendant de toute variable X */
    new_y = normal(j(n_samples, 1, 0));
    return(new_y);
finish;
store module=(Independance);

/* Pour simuler de la multicolinarit avec Toeplitz, Iman-Conover permet de conserver
    les outliers (pas utile dans notre cas) */
start ImanConoverToeplitz(X, rho); /* Le rho est le degr de corrlation de Toeplitz */
   p=ncol(X);
   /* Toeplitz */
   C = j(p,p,0);
   do i = 1 to p;
      do j = 1 to p;
         C[i,j] = rho##abs(i-j);
      end;
   end;
   /* Iman-Conover */
   N = nrow(X);
   S = J(N, ncol(X));
   do i = 1 to ncol(X);
      ranks = ranktie(X[,i], "mean");
      S[,i] = quantile("Normal", ranks/(N+1));
   end;
   CS = corr(S);
   Q = root(CS);
   P = root(C);
   T = solve(Q,P);
   Y = S*T;
   W = X;
   do i = 1 to ncol(Y);
      rank = rank(Y[,i]);
      tmp = W[,i]; call sort(tmp);
      W[,i] = tmp[rank];
   end;
   return(W);
finish;
store module=(ImanConoverToeplitz);

/* Pour simuler une rupture au point break*n, size va dfinir la variation de nos beta
    */
start Rupture(break, size, Y, X, beta, eps);
    beta2=j(nrow(beta),1,0);
    do i=1 to nrow(beta);
        beta2[i]=beta[i]+size*(2*rand("uniform")-1);
    end;
    idx=int(break*nrow(Y));
    do i=idx to nrow(Y);

    Y[i]=X[i,5]*beta2[1]+X[i,6]*beta2[2]+X[i,20]*beta2[3]+X[i,30]*beta2[4]+X[i,40]*beta2[5]+eps[i];
    end;
```

```
    return(Y);
finish;
store module=(Rupture);

/* Simule des outliers sur les rsidus de frquence freq ce qui va dcaler leur moyenne
    droite de size*/
start Extremes(eps, freq, size);
    u=ranuni(j(1,nrow(eps),1));
    do i=1 to nrow(eps);
        if u[i]<freq then do;
            eps[i]=eps[i]+size;
        end;
    end;
    return(eps);
finish;
store module=(Extremes);

%macro MCglm(n, MC, select, choose, stop, inde, rupture, multi, extreme);


/* ===================== Clean ===================== */
proc datasets lib=work nolist;
    delete bigtrain /*bigtest*/;

quit;



proc iml;
    load module=(Extremes Rupture ImanConoverToeplitz Independance);
    /* On dcide de Gnrer un X de taille n*MC puis on le dcoupera en MC bloc n */
    n = &n.*&MC.;
    p = 50;

    beta = j(p,1,0);
    beta[1]=0.9; beta[2]=-1.0; beta[3]=1.2; beta[4]=-0.8; beta[5]=1.1;

    X = normal(j(n,p,0));
    eps = normal(j(n,1,0))*0.25;

    /* ===================== Multvari ===================== */
    /* donc X suit une loi normal(m,var) avec m un vecteur et var une matrice
    diagonale */
    m = 2*ranuni(j(1,p,1))-1;
    var = j(p,p,0);
    do i = 1 to p;
        var[i,i] = 3*ranuni(0)+1;
    end;
    X = X*sqrt(var) + m;

    /* ===================== Valeurs extrmes ===================== */
    %if &extreme.=oui_extreme %then %do;
        eps = Extremes(eps, 0.05, 4);
    %end;

    /* ===================== Multi-colinarit ===================== */
    %if &multi.=oui_multi %then %do;
        X = ImanConoverToeplitz(X, 0.8);
```

```
109     %end;
110
111     /* Modle */
112     Y = X[,5]*beta[1] + X[,6]*beta[2] + X[,20]*beta[3] + X[,30]*beta[4] +
        X[,40]*beta[5] + eps;
113
114     /* ==================== Indpendance ==================== */
115     %if &inde.=oui_inde %then %do;
116         Y = Independance(n);
117     %end;
118
119     /* ==================== Rupture ==================== */
120     %if &rupture.=oui_rupture %then %do;
121         Y = Rupture(0.5, 1, Y, X, beta, eps);
122     %end;
123
124     train = {};
125     test = {};
126     %do r = 1 %to &MC.;
127
128         /* ==================== Split train/test ==================== */
129         /*u = ranuni(j(1,&n.,0));
130         idxTrain = loc(u < 2/3)+&n.*(&r.-1);
131         idxTest  = loc(u >= 2/3)+&n.*(&r.-1);
132
133         train = train // (j(ncol(idxTrain),1,&r.) || Y[idxTrain] || X[idxTrain,]);
134         test  = test // (j(ncol(idxTest),1,&r.) || Y[idxTest] || X[idxTest,]);*/
135         train = train // (j(&n.,1,&r.) || Y[1+&n.*(&r.-1):&n.*&r.] ||
        X[1+&n.*(&r.-1):&n.*&r.,]);
136
137     %end;
138
139     colnames = "rep" ||"Y" || ( "X1":"X50");
140
141     create bigtrain from train[colname=colnames];
142     append from train;
143     close bigtrain;
144
145     /*create bigtest from test[colname=colnames];
146     append from test;
147     close bigtest;*/
148
149
150 quit;
151
152 ods exclude all;
153 ods output ParameterEstimates = ParEst;
154
155 proc glmselect data=bigtrain /*testdata=bigtest*/ plots=all;
156     by rep;
157     model Y = X1-X50
158         / selection=&select choose=&choose stop=&stop;
159
160
161 run;
162 ods exclude none;
163 /* ==================== MODELE CHOISI ==================== */
164
```

```
165
166 data ModeleChoisi;
167     set ParEst;
168     where Parameter ne "Intercept";
169 run;
170
171 /* ==================== FP FN ==================== */
172
173
174 proc iml;
175
176     vrai={X5 X6 X20 X30 X40};
177     use ModeleChoisi; read all var {"rep"} into rep; close;
178     use ModeleChoisi; read all var {"Parameter"} into ChoisiMC; close;
179
180     /* Si ModeleChoisi est vide */
181     if nrow(rep)=0 then do;
182         rep = j(1,1,.);
183         ChoisiMC = j(1,1,"");
184     end;
185
186     results = j(&MC., 3, .);      /* rep, FP, FN */
187
188     do i = 1 to &MC.;
189         idx = loc(rep=i);
190         choisi = {};
191         if ncol(idx)^=0 then do;
192             choisi = ChoisiMC[idx]‘;
193         end;
194
195         /* FP = choisi \ vrai */
196         FP = setdif(choisi, vrai);
197
198         /* FN = vrai \ choisi */
199         FN = setdif(vrai, choisi);
200
201         results[i,] = i || ncol(FP) || ncol(FN);
202
203     end;
204
205     create Indicateurs from results[colname={"rep" "FP" "FN"}];
206     append from results;
207     close Indicateurs;
208
209
210 quit;
211
212 /* ==================== INDICATEURS ==================== */
213
214 data Indicateurs;
215     set Indicateurs;
216     methode  = "&select.";
217     choose   = "&choose.";
218     stop     = "&stop.";
219     Exact    = (FP=0 and FN=0);
220     Overfit  = (FP>0 and FN=0);
221     Underfit = (FP=0 and FN>0);
222     Mixed    = (FP>0 and FN>0);
```

```
223
224 run;
225
226
227 proc sql;
228     create table ResumeIndicateurs as
229     select
230         methode,
231         choose,
232         stop,
233         mean(Exact)    as ExactM,
234         mean(Overfit)  as OverfitM,
235         mean(Underfit) as UnderfitM,
236         mean(Mixed)    as MixedM
237     from Indicateurs
238     group by methode, choose, stop;
239
240 quit;
241
242
243 proc append base=Indicateurs_All data=ResumeIndicateurs force; run;
244
245 %mend;
246
247 proc datasets lib=work nolist;
248     delete Indicateurs_All;
249
250 quit;
251
252 title "Dpendance␣linaire";
253 %MCglm(200,300, elasticnet, cv, cv, non_inde, non_rupture, non_multi, oui_extreme);
254 proc print data=Indicateurs_All; run;
255
256 proc datasets lib=work nolist;
257     delete Indicateurs_All;
258
259 quit;
260
261 title "Multi_colinarit";
262 %MCglm(200,300, lasso, cv, cv, non_inde, non_rupture, oui_multi, non_extreme);
263 proc print data=Indicateurs_All; run;
264
265 proc datasets lib=work nolist;
266     delete Indicateurs_All;
267 quit;
268
269 title "Valeurs␣Extremes";
270 %MCglm(200,300, lasso, cv, cv, non_inde, non_rupture, non_multi, oui_extreme);
271 proc print data=Indicateurs_All; run;
272
273 proc datasets lib=work nolist;
274     delete Indicateurs_All;
275 quit;
276
277 title "Rupture";
278 %MCglm(200,300, lasso, cv, cv, non_inde, oui_rupture, non_multi, non_extreme);
279 proc print data=Indicateurs_All; run;
280
```

```
proc datasets lib=work nolist;
    delete Indicateurs_All;
quit;
```

Listing 2: Cas Empirique

```
data diabetes;
    set sashelp.diabetes;
run;


%macro MCglm(diabetes, select, choose, stop);

ods select ASEPlot;
proc glmselect data=&diabetes. plots=all seed=12345;
    partition fraction(validate=0.3);
    class SEX;
    model Y = AGE SEX BMI BP S1 S2 S3 S4 S5 S6
        / selection=&select choose=&choose stop=&stop
         cvmethod=random(10);
run;
ods select all;

%mend;

title "Lar";
%MCglm(diabetes, lar, cv, cv);
title "lasso";
%MCglm(diabetes, lasso, cv, cv);
title "elasticnent";
%MCglm(diabetes, elasticnet, cv, cv);

/* ====================== Dtection corrlation ====================== */
proc reg data=diabetes;
    model Y = AGE SEX BMI BP S1 S2 S3 S4 S5 S6 / vif tol collin;
run;
quit;
proc corr data=diabetes out=corr_mat;
    var AGE SEX BMI BP S1 S2 S3 S4 S5 S6;
run;
quit;

/* ====================== Dtection valeurs extrmes ====================== */
proc sgplot data=diabetes;
    histogram S4;
    density S4 / type=kernel;
    title "Distribution de S4";
run;


/* ====================== Dtection rupture ====================== */
proc loess data=diabetes;
    model Y = BMI; /* Teste sur la variable la plus corrle */

```

```
run;

/* Exemple : On teste si la structure change  la 221me observation (milieu du
    dataset) */
proc sort data=diabetes out=diabetes_trie;
    by BMI;
run;

data diabetes_chow;
    set diabetes;
    if BMI <= 25 then class = 0;
    else class = 1;
run;

data diabetes_chow;
    set diabetes_chow;
    classBMI = class*BMI;
run;

proc reg data=diabetes_chow;
    model Y = BMI class classBMI;
    test class = 0, classBMI = 0;
run;
quit;
```

Listing 3: Parallel Computing

```

options sascmd="sas";
%let _start_dt = %sysfunc(datetime());
signon task1;
rsubmit task1 wait=no;

    libname SASDL 'C:\temp';

    /* First task here*/

endrsubmit;


signon task2;
rsubmit task2 wait=no;

    libname SASDL 'C:\temp';

    /* Second task here*/

endrsubmit;

/* Etc.*/
```