Feature Selection Under Multicollinearity: A Comparison of Stepwise and Regularization Approaches

Group 6: Inès Ben Hamza, Inès Battah

1. Introduction

In high-dimensional datasets, selecting relevant features is essential to building interpretable and accurate predictive models. To address this issue, both stepwise and regularization-based feature selection methods are used to improve model performance by identifying the most informative features. Stepwise methods such as Forward and Backward Selection, rely on iterative inclusion or exclusion of features based on their contribution to model accuracy. Regularization methods, including Lasso and Elastic Net, incorporate penalty terms that shrink some feature coefficients towards zero, effectively eliminating irrelevant or redundant features from the model. By doing so, these methods help reduce overfitting and enhance the generalizability of the model to unseen data.

However, when features are highly correlated, some predictors in the model provide redundant information, making it difficult to isolate their individual effects on the dependent variable. In severe cases, multicollinearity can lead to instability in coefficient estimates, increased standard errors, and difficulties in interpreting the model.

Multicollinearity occurs when two or more predictor variables are highly correlated and therefore contain redundant information. This complicates feature selection since it destabilizes coefficient estimates in linear models: the model may struggle to distinguish the unique contribution of each correlated predictor. A small change in the data can therefore lead to large variations in the estimated coefficients, leaving us with an unstable and unreliable model. Moreover, multicollinearity reduces model interpretability. When different features represent similar information, interpreting their individual effects on the dependent variable can become challenging but also misleading, since the model may arbitrarily assign high importance to one correlated feature over another. In feature selection, multicollinearity can lead to biased models by increasing the likelihood of selecting irrelevant features that correlate with the relevant ones.

Hence why we found great importance in running a simulation to study closely the performance of different feature selection methods when impacted by multicollinearity. Our initial hypothesis is that this challenge is particularly pronounced in stepwise methods since they add or remove variables based on marginal contributions without considering the interdependence between predictors.

Therefore stepwise feature selection methods may struggle to distinguish between essential and redundant information, potentially leading to biased models. We compare their performance in recovering our true model with more robust feature selection methods: embedded methods.

1.1 - Description of the Methods Compared

Stepwise Method

Our research involves different methods for feature selection. Stepwise methods are some of the most used for this task, and we use both forward and backward selection to recover our true features methodically. Backward elimination starts with a model with all features and removes, at each iteration, the one that improves performance the least, stopping only when removing more variables doesn't impact performance anymore. Forward selection uses the same logic but starts with an empty model and adds predictor variables at each step following the same method by selecting the one that improves performance the most. Performance is quantified in terms of AIC, a performance metric that accounts for model complexity by accounting for the number of predictors used in the model. These methods are especially helpful in ensuring a balance between complexity and performance.

In the scope of our project, we aim to study how effective they are in recovering our true model, which is the relevant predictors in our linear regression. We implemented these methods using the *stepAIC* function from the

MASS package in R. In forward selection, we started with an initial empty model and specified the full model as the upper scope, allowing the model to select variables that improved AIC the most. In backward elimination, we applied the *stepAIC* function directly to the full model, removing predictors with the least contribution to AIC until no further improvements were possible. As stepwise methods analyze features individually or in stages, frequently using metrics like p-values or AIC, it might result in the inclusion of features that appear important by chance, particularly in small datasets. Because these methods do not account for all variables simultaneously, they may capture noise or correlations, resulting in models that perform well on training data but badly on unseen data, potentially overfitting.

The added difficulty of having to choose from a variety of predictors, combined with the challenge of multicollinearity, may hinder these methods' ability to distinguish between relevant and irrelevant variables. Hence, we have chosen to compare their performance with embedded methods to study how effective each method is in this scenario.

Embedded method

The two other methods that we will study are Lasso and Elastic Net, both part of the embedded methods family. Compared to stepwise methods, they select features based on criteria inherent to the model, such as regularization penalties, which naturally prioritize the most relevant features. Regularization penalizes model complexity by shrinking coefficients toward zero, thereby minimizing overfitting. This process naturally perform feature selection, as some coefficients are reduced to exactly zero, excluding less important variables from the model.

We implemented Lasso and Elastic Net using the cv.glmnet function from the glmnet package in R, allowing for cross-validation to select the optimal regularization parameter (lambda) that minimizes the cross-validated error. For Lasso, we set alpha at 1, applying pure L1 regularization, which shrinks some coefficients to zero, thus excluding certain features. For Elastic Net, we set alpha at 0.5 to balance L1 and L2 regularization, making it more robust to correlated predictors by retaining groups of correlated features. The optimal lambda value was chosen as the one that produced the lowest mean cross-validated error.

Lasso is a shrinkage method that simultaneously selects and regularizes variables by using L1 regularization, while Elastic Net combines both L1 (Lasso) and L2 (Ridge) regularization. This makes Elastic Net more flexible in cases where predictors are highly correlated, as it can retain groups of correlated features together—something Lasso may not achieve alone. However, in our simulation, it is preferable to shrink one feature to zero and retain only the relevant feature in each correlated pair. Both methods are essential in our study, as they incorporate feature selection into the modeling process, enabling us to handle high-dimensional data and account for multicollinearity, ultimately enhancing model performance.

1.2 - Research question

This study aims to evaluate the performance of Forward Selection, Backward Selection, Lasso, and Elastic Net in recovering the true model under conditions of high multicollinearity. Here,'recovering the true model' refers to each method's capacity to accurately choose only relevant variables that have a true influence on the response variable, while rejecting the irrelevant ones. Our goal is to optimize the selection of true predictors (True Positive Rate, or TPR) while minimizing the selection of irrelevant predictors (False Positive Rate, or FPR). Furthermore, we intend to look into how each method handles multicollinearity – the presence of strongly correlated features — which can affect the selection process and result in biased models. In this environment, addressing multicollinearity is critical to ensuring model stability and interpretability.

We therefore aim to understand :

- How well each method recovers the true model by selecting relevant features and avoiding the irrelevant ones.
- Whether some methods are more robust in multicollinearity.
- How does the sample size affect each method's ability to identify the true predictors in the model.

2. Data Generation

2.1 - Data Generation Process

To generate our data, we designed a custom generate function that allows us to generate several samples with the same underlying true model. At each iteration, N data points will be generated, each following a normal distribution. More specifically, it will create a matrix X filled with random numbers drawn from a standard normal distribution with mean 0 and standard deviation 1.

Each synthetic dataset will consist of ten features, each drawn from a normal distribution.

Out of these ten features, five are included in the true model with non-zero coefficients, indicating their effect on the response variable, while the other five have zero coefficients, meaning they have no effect. The response variable Y is generated as a linear combination of the five significant features.

2.2 - Design of True Model and Multicollinearity

True model : $\mathbf{Y = 1 + 3\ X_1 - 2\ X_2 + 4\ X_3 + 2\ X_4 + 3\ X_5 + 0\ X_6 + 0\ X_7 + 0\ X_8 + 0\ X_9 + 0\ X_{10} + e}$

The coefficients in the true model were chosen to represent a range of effect sizes, allowing us to test the effectiveness of feature selection approaches to find both stronger and moderate predictors. These numbers, though arbitrary, simulate a scenario where predictors contribute differently to the outcome, similar to real-world datasets in which not all factors have equal importance. By assigning a mix of positive and negative coefficients, we aim to evaluate how effectively each method handles predictors with opposing relationships to the outcome.

To introduce multicollinearity in the data and explore its potential effect on feature selection methods, two of the ten features will be highly correlated. To do so, we created a covariance matrix where we set specific correlations between features. In particular, we created a high correlation (0.9) between feature 1 (relevant) and feature 6 (irrelevant), as well as between feature 2 (relevant) and feature 7 (irrelevant), simulating a scenario where both relevant and irrelevant features are highly correlated with each other. This high level of correlation allows us to examine how different feature selection methods handle the challenge of identifying relevant variables in the presence of multicollinearity.

For each iteration of the Monte Carlo simulation, 50, 100, 500 and 1000 observations will be generated. This will enable us to evaluate how the performance of the feature selection methods scales with different sample sizes.

3. Monte Carlo Simulation

The purpose of the simulation is to compare the ability of different feature selection methods to correctly identify relevant features in the presence of multicollinearity.

3.1 - Simulation Design and Setup

We set a seed at the beginning of the simulation function to ensure that all the random number generated within that function, will produce the same results each time we run the function making sure our entire simulation is consistent and reproducible.

The simulation was structured to mimic real-world data characteristics while introducing additional complexity. Using a Monte Carlo simulation, our true model will include some correlated features, as explained in section 2.2, and we will systematically vary the sample size to examine each method's ability to correctly identify relevant features while minimizing the inclusion of irrelevant ones.

Thus, for each sample size (50, 100, 500, 1000) and method, 1000 iterations are run to ensure that the results are robust and reflective of the methods' performance across many different samples. This repetition provides a comprehensive assessment of each method.

Our analysis focuses on the trade-offs these methods face between accuracy, stability, and computational efficiency in a high-dimensional context. By comparing the True Positive Rate (TPR) and False Positive Rate (FPR) of each approach, this study seeks to provide insights into the strengths and limitations of each feature selection method, particularly in environments where there is multicollinearity.

3.2 - Evaluation Metrics

We assess each method's ability to correctly identify relevant features while minimizing the inclusion of irrelevant ones. The primary evaluation metrics used are :

True Positive Rate: TPR is calculated by determining the proportion of relevant features that a method correctly identifies. This process starts by counting the number of true positives, which are features that the method selects and that are indeed relevant. This count is then divided by the total number of actual relevant features present in the dataset. Essentially, TPR measures the method's ability to correctly include all features that belong to the true model. A high TPR indicates that the method effectively captures the important features needed for accurate modeling.

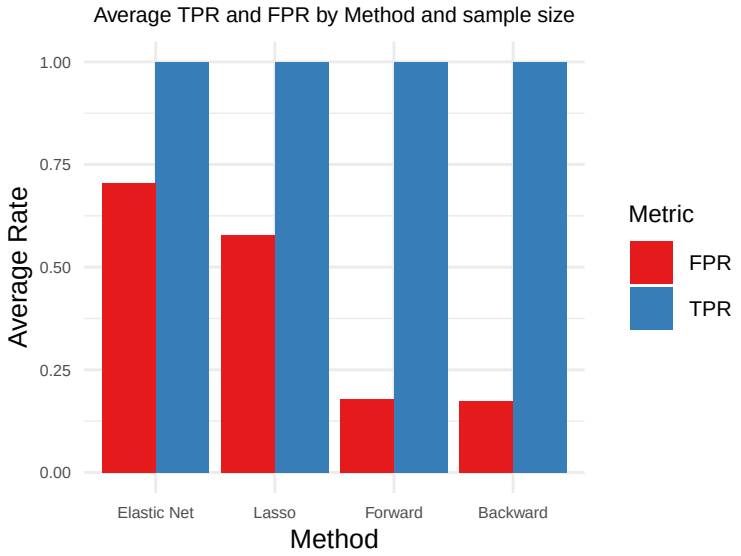
False Positive Rate: FPR quantifies how many irrelevant features the feature selection method mistakenly includes. To calculate this, the number of false positives features that the method selects but are not actually relevant is first counted. This number is then divided by the total number of irrelevant features in the dataset. A lower FPR means the method is better at excluding features that do not contribute meaningful information.

These metrics are calculated for each iteration of the simulation and for each sample size resulting in 16,000 calculations (1000 iterations for each method and each sample size, leading to 16000 iterations in total).

4. Reporting the results

4.1 - Overall results across Methods

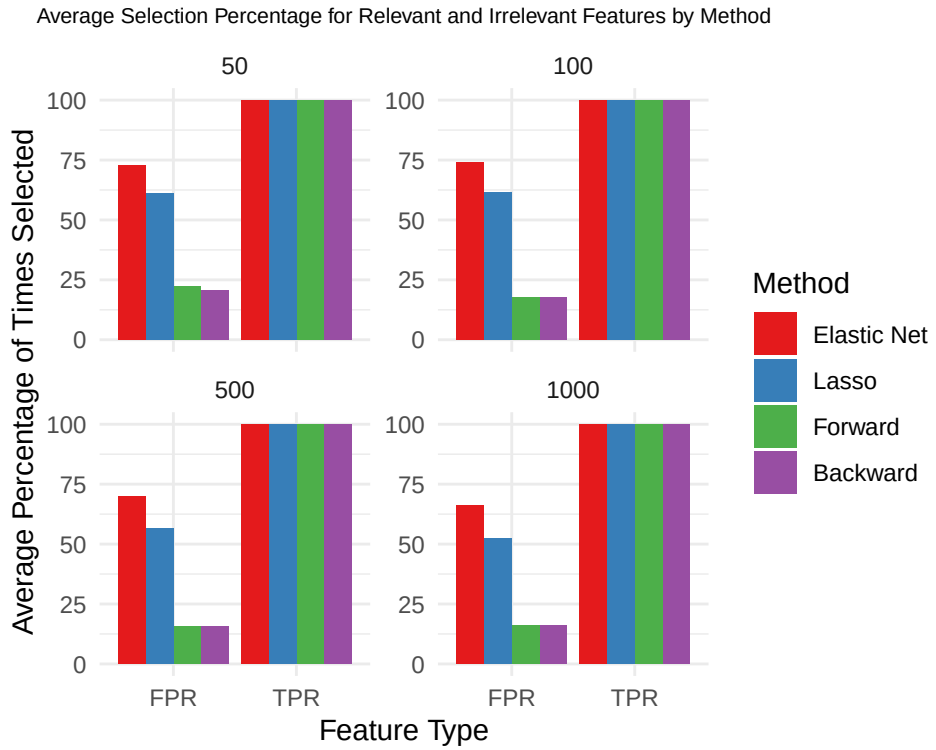
As it can be seen on the below plot, the best methods for minimizing false positives are Backward and Forward selection, making them very reliable when the goal is simply to identify the true features of a model. The two other embedded methods, Elastic Net and Lasso, both have a relatively high false positive average rate.



To gain insight into the results of the simulation, we calculate and display the average True Positive Rate (TPR) and False Positive Rate (FPR) for each feature selection method across different sample sizes by grouping the simulation results by method and sample size, ensuring that the averages are calculated separately for each combination. For each group, we compute the mean TPR and FPR, which represents how well each method correctly identifies relevant features and avoids selecting irrelevant ones, respectively.

4.2 - Method Performance Across Sample Sizes: True Positive and False Positive Rates

After running the simulations, We analyzed the average TPR and FPR for each method and sample size across all 1,000 iterations.



We found that while all methods have a 100% TPR (e.i. all methods correctly identify all true relevant features), all feature selection methods are not all created equal. As False Positive Rates differ greatly amongst methods, we can infer that in our scenario, these feature selection methods perform equally well in recovering the true model (True Positives), but they do not select the wrong features at the same rate (False Positives). Over all iterations of our simulation and sample sizes, stepwise methods consistently outbest embedded methods in terms of FPR.

Average True Positive Rate (TPR) and False Positive Rate (FPR) by Method and Sample Size

Method	Sample Size	Avg_TPR	Avg_FPR
Backward	50	1	0.208
Elastic Net	50	1	0.726
Forward	50	1	0.223
Lasso	50	1	0.610
Backward	100	1	0.177
Elastic Net	100	1	0.740
Forward	100	1	0.179
Lasso	100	1	0.616
Backward	500	1	0.155
Elastic Net	500	1	0.698
Forward	500	1	0.155
Lasso	500	1	0.565
Backward	1000	1	0.160
Elastic Net	1000	1	0.660
Forward	1000	1	0.159
Lasso	1000	1	0.525

Sample size effect

As sample size increases, we notice a consistent decrease in the False Positive Rate (FPR) across all methods. Models have more data to better estimate the relationships between features and the response variable. This increased data

availability enables more reliable statistical inferences, making it easier for feature selection methods to distinguish between relevant and irrelevant features. As a result, all methods tend to select fewer irrelevant features, which improves their overall accuracy in identifying the true underlying model.

• **Lasso**: FPR decreases from 0.61 at sample size 50 to 0.525 at 1000, showing an 8.5% improvement. Despite this reduction, Lasso maintains a relatively high FPR, reflecting its tendency to select more features overall.

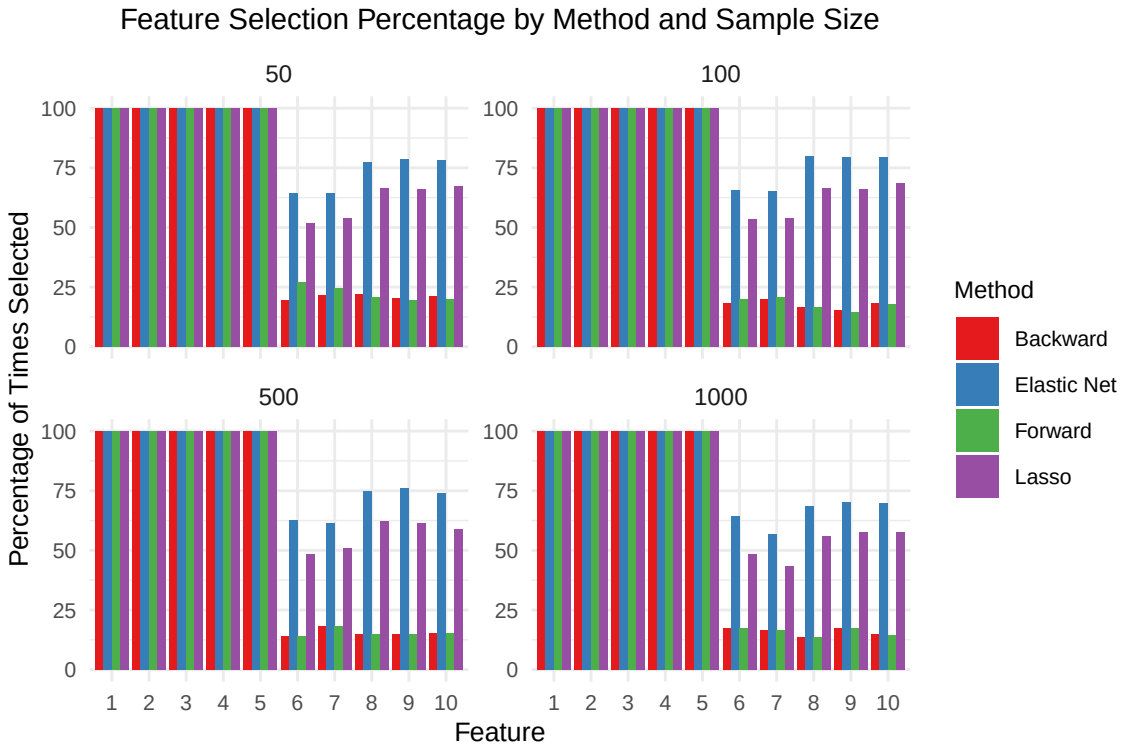
• **Elastic Net**: FPR decreases from 0.726 to 0.66 (around 6.6% decrease), but remains relatively high across sample sizes

• **Backward Selection**: FPR drops from 0.208 at sample size 50 to 0.16 at 1000, a 4.8% decrease. This highlights Backward Selection's conservative approach in avoiding irrelevant features, with small improvements as sample size grows.

• **Forward Selection**: Similar to Backward Selection, Forward Selection's FPR decreases from 0.223 to 0.159, a 6.4 % reduction, indicating improved feature selection accuracy with larger samples.

With an FPR about three times lower than Elastic Net's – as we can see in the table– both stepwise methods Backward and Forward Selection show a lower propensity to select irrelevant variables, demonstrating their robustness and reliability in cases where avoiding false positives is more important. This implies that, in contexts with no multicollinearity, stepwise methods may be more effective than embedded methods for precise feature selection.

Addressing multicollinearity



A primary focus of this study is how feature selection methods handle multicollinearity, as correlated features like X6 and X7 present challenges in accurately identifying relevant variables. Across all sample size, we can see that X6 and X7 are selected less frequently compared to other irrelevant features like X8, X9, and X10 by Lasso and Elastic Net. Therefore they effectively handle multicollinearity by prioritizing relevant features and excluding redundant ones. Respectively :

Elastic Net Selection Frequencies (%) of Irrelevant Features

Sample Size	X6	X7	X8	X9	X10
50	*64.4*	*64.5*	77.3	78.8	78.1
100	*65.7*	*65.2*	79.9	79.4	79.6
500	*62.6*	*61.3*	74.7	76.2	74.2
1000	*64.5*	*56.9*	68.4	70.4	69.7

Lasso Selection Frequencies (%) of Irrelevant Features

Sample Size	X6	X7	X8	X9	X10
50	*51.8*	*53.8*	66.3	66.2	67.1
100	*53.5*	*53.7*	66.5	66.0	68.5
500	*48.3*	*50.8*	62.1	61.6	59.1
1000	*48.3*	*43.2*	56.0	57.6	57.5

By selectively penalizing correlated predictors, these methods focus on features that provide independent predictive contributions. The reduced selection of X6 and X7 in the tables indicates that they are recognized as redundant due to their high correlation with relevant variables, showcasing the ability of these regularization methods to improve model robustness. This successful avoidance of correlated irrelevant variables is not observed in stepwise methods which proves that while they have less of a tendency to select irrelevant variables, they lack the capacity to handle multicollinearity effectively. In fact for small sample size (e.i. 50 and 100), we can see that forward selection tends to include X6 and X7 more often compared to X8, X9 and X10, respectively :

Forward Selection Frequencies (%) of Irrelevant Features

Sample Size	X6	X7	X8	X9	X10
50	*27*	*24.5*	20.9	19.3	19.8
100	*19.9*	*20.7*	16.5	14.6	17.8

Because stepwise approaches do not use regularization to penalize correlated predictors, they may mistakenly include redundant features. This illustrates a trade-off: whereas stepwise methods effectively minimize false positives, embedded methods such as Elastic Net and Lasso better address multicollinearity, improving model stability in the presence of correlated features.

4.3 - Statistical Analysis of Feature Selection Methods

To understand the impact of method choice and sample size on FPR, we used ANOVA to detect significant overall differences, followed by Tukey's HSD test for pairwise comparisons. The ANOVA test confirms that selection method has a highly significant impact on performance ($F(3, 15995) = 6849.6$, $p < 2e\text{-}16$), with pairwise comparisons via Tukey's HSD test showing substantial differences between methods. This underscores that methods like Backward and Forward Selection, which exhibit lower FPR, consistently outperform embedded methods in terms of avoiding irrelevant features. Sample size also plays a role ($F(1, 15995) = 235.3$, $p < 2e\text{-}16$), but its effect is less pronounced, suggesting that all methods tend to improve as more data becomes available.

We performed a Tukey post-hoc test to identify specific pairwise differences in FPR between methods. The test reveals that Elastic Net and Lasso both have significantly higher FPRs than Backward Selection (differences of 0.5309 and 0.4039, respectively, both $p < 0.0001$), while Forward Selection does not significantly differ from Backward in FPR (difference = 0.0041, $p = 0.8164$). This underlines that stepwise methods (Backward and Forward Selection) are more conservative in feature selection, focusing on minimizing false positives compared to embedded methods.

These findings support our hypothesis that regularization-based methods, while useful for managing multicollinearity, tend to include more irrelevant features, whereas stepwise methods were proven advantageous in high-dimensional settings. Method choice is a major determinant of the False Positive Rate, and Sample Size also influences FPR.

Conclusion

In conclusion, we've found that both stepwise and embedded methods perform fairly well in consistently recovering relevant features (1 to 5) across different sample sizes.

Elastic Net seems more prone to selecting irrelevant features, while Backward and Forward Selection remain more conservative. Lasso was the second most prone to falsely select the wrong features, showing that in this scenario, embedded methods tend to perform less well than stepwise ones. This trend holds throughout our simulation and across different sample sizes.

As the sample size increases, we notice that all methods improve their feature selection accuracy by reducing their selection of irrelevant features, but the general trend remains the same, indicating that stepwise methods are the most robust feature selection method in this environment.

We also studied how well each method handled multicollinearity, where Elastic Net was the best method due to the combined effects of Lasso and Ridge penalties, which allows for some feature selection while retaining correlated predictors, but Lasso didn't fall behind either. These 2 methods worked better at handling correlations in comparison to stepwise methods which did not really consider the combined effect of the features.

Overall, our findings suggest that while embedded methods may offer advantages in handling multicollinearity, stepwise methods provide a more conservative and reliable approach in scenarios where avoiding false positives is critical, like in small sample sizes.