ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Σχολή Χρηματοοικονομικής και Στατιστικής

Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ

ΣΥΓΚΡΙΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΝΙΚΟΠΟΙΗΜΕΝΩΝ ΤΕΧΝΙΚΩΝ ΚΑΙ ΤΕΧΝΙΚΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Δημήτριος Ραΐσης

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην *Εφαρμοσμένη Στατιστική*

Πειραιάς

Ιούνιος 2026

Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο Π Ε Ι Ρ Α Ι Ω Σ

Σχολή Χρηματοοικονομικής και Στατιστικής

Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ

ΣΥΓΚΡΙΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΝΙΚΟΠΟΙΗΜΕΝΩΝ ΤΕΧΝΙΚΩΝ ΚΑΙ ΤΕΧΝΙΚΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Δημήτριος Ραΐσης

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην *Εφαρμοσμένη Στατιστική*

Πειραιάς

Ιούνιος 2026

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή που ορίσθηκε από τη ΓΣΕΣ του Τμήματος Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς στην υπ' αριθμ. …….. συνεδρίασή του σύμφωνα με τον Εσωτερικό Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών στην Εφαρμοσμένη Στατιστική

Τα μέλη της Επιτροπής ήταν :

- Αναπλ. Καθηγητής Καραγρηγορίου Αλέξανδρος (Επιβλέπων)
- Αναπλ. Καθηγητής Τριανταφύλλου Ιωάννης
- Επίκουρος Καθηγητής Ανδρουλάκης Εμμανουήλ

Η έγκριση της Διπλωματική Εργασίας από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνωμών του συγγραφέα.

UNIVERSITY OF PIRAEUS

School of Finance and Statistics

Department of Statistics and Insurance Science

POSTGRADUATE PROGRAM IN

APPLIED STATISTICS

Comparative Evaluation of Penalized Techniques and Machine Learning Techniques for the Study of Linear Models

By

Dimitrios Raisis

MSc Dissertation

submitted to the Department of Statistics and Insurance Science of the University of Piraeus in partial fulfilment of the requirements for the degree of Master of Science in Applied Statistics

Piraeus, Greece

June 2026

Στους γονείς μου

Μιχάλη και Κατερίνα

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλεπόντα Καθηγητή Αλέξανδρο Καραγρηγορίου για την εμπιστοσύνη και την ανάθεση του θέματος καθώς και την στήριξη του κατά την διάρκεια εκπόνησης της εργασίας. Ακόμη θα ήθελα να ευχαριστήσω την σύντροφο μου Εύα και τους φίλους μου Κ.Κ. & Δ.Ζ. που είναι διπλά μου σε κάθε βήμα μου.

Περίληψη

Η πρόβλεψη συνεχών μεταβλητών αποτελεί σημαντικό αντικείμενο της σύγχρονης στατιστικής ανάλυσης και της μηχανικής μάθησης. Πέρα από τα κλασικά γραμμικά μοντέλα, έχουν αναπτυχθεί μέθοδοι ποινικοποιημένης παλινδρόμησης, όπως οι Ridge, Lasso και Elastic Net, οι οποίες επιτρέπουν την αντιμετώπιση προβλημάτων όπως η πολυσυγγραμμικότητας και η μείωση διάστασης . Παράλληλα, οι μέθοδοι δέντρων παλινδρόμησης αποτελούν μια ευέλικτη μη παραμετρική προσέγγιση που μπορεί να μοντελοποιήσει μη γραμμικές σχέσεις μεταξύ των μεταβλητών, ενώ σύγχρονες τεχνικές όπως τα Random Forests συνδυάζουν πολλαπλά δέντρα με σκοπό τη βελτίωση της προβλεπτικής ακρίβειας.

Στην παρούσα εργασία μελετάται και συγκρίνεται η προβλεπτική ικανότητα των παραπάνω μεθόδων, καθώς και υβριδικών προσεγγίσεων που συνδυάζουν τον αριθμό των μεταβλητών μέσω elastic net με Regression trees. Επιπλέον, εξετάζονται γενικές συναρτήσεις ποινικοποίησης που χρησιμοποιείται στη διαδικασία κλαδέματος (pruning) δέντρων, με στόχο να διερευνηθεί κατά πόσο οι γενικεύσεις αυτές μπορούν να βελτιώσουν την προβλεπτική ικανότητα του μοντέλου. Η αξιολόγηση των μεθόδων πραγματοποιείται μέσω εφαρμογών σε πραγματικά σύνολα δεδομένων και σύγκρισης κατάλληλων μέτρων αξιολόγησης, με σκοπό την ανάδειξη των πλεονεκτημάτων και των περιορισμών κάθε μεθόδου.

Στην εισαγωγική ενότητα συζητούνται τα δύο πραγματικά σύνολα δεδομένων από της R.

Στο Κεφάλαιο 1 γίνεται αναφορά στην πολυσυγγραμικότητα και τη σχέση μεταβλητότητας και μεροληψίας ενώ συ ζητώνται συνοπτικά τεχνικές συρρίκνωσης όπως Ridge, LASSO & Elastic Net οι οποίες μπορούν να αξιοποιηθούν στον περιορισμό της πολυσυγγραμικότητας (εφόσον υφίσταται) καθώς και στη μείωση της διάστασης (dimension reduction).

Το Κεφάλαιο 2 είναι αφιερωμένο στις μεθόδους που βασίζονται σε δένδρα απόφασης

καθώς και στη διαδικασία κλαδέματος.

Στο Κεφάλαιο 3 προτείνονται και συζητώνται δύο γενικευμένες τεχνικές ποινικοποίησης οι οποίες και εφαρμόζονται στα δεδομένα .

Η συγκριτική αξιολόγηση των κλασικών δένδρων απόφασης καθώς και των προτεινομένων τεχνικών βασισμένων αφενός μεν στο μέσο τετραγωνικό σφάλμα και στο μέγεθος του δένδρου. Στη συγκριτική αξιολόγηση για σκοπούς αντιπαραβολής έχει περιληφθεί και η τεχνικών των τυχαίων δασών αν και αφορούν μεγάλο αριθμού (συχνά μη μετρήσιμου) δένδρων.

Abstract

The prediction of continuous response variables is an important topic in modern statistical analysis and machine learning. In addition to classical linear models, penalized regression methods such as Ridge, Lasso, and Elastic Net have been developed to address problems including multicollinearity and dimensionality reduction. At the same time, regression tree methods provide a flexible non-parametric approach capable of modeling nonlinear relationships among variables, while modern techniques such as Random Forests combine multiple trees in order to improve predictive accuracy.

This thesis studies and compares the predictive performance of the aforementioned methods, as well as hybrid approaches that combine variable selection through Elastic Net with regression trees. Furthermore, generalized penalty functions for the tree pruning process are investigated in order to examine whether these generalizations can improve the predictive performance of the resulting models. The methods are evaluated using real-world datasets and appropriate performance measures, with the aim of highlighting the advantages and limitations of each approach.

The introductory chapter presents the two real-world datasets available in the R statistical environment.

Chapter 1 introduces the concepts of multicollinearity and the bias–variance trade-off, and briefly discusses shrinkage techniques such as Ridge, Lasso, and Elastic Net, which can be used to reduce multicollinearity (when present) as well as to perform dimensionality reduction.

Chapter 2 is devoted to regression tree methods and the corresponding tree pruning procedure.

In Chapter 3, two generalized penalty functions are proposed, discussed, and applied to the datasets.

Finally, the classical regression trees and the proposed pruning techniques are comparatively evaluated using the Mean Squared Error (MSE) and tree size as the main evaluation criteria. For comparison purposes, the Random Forest method is also included, although it is based on a large (and often uncountable) number of decision trees.

Table of Contents

List of Tables

List of Figures

INTRODUCTION

In statistical science, we aim to provide answers to problems concerning the relationship between variables, with the ultimate goal of predicting the value of one variable using the observed values of others.
The rapid advancement of technology and computing power has led to the collection of much larger volumes of data compared to the past. Consequently, researchers are now faced with problems of higher complexity, involving a greater number of variables and more intricate interdependencies.

One of the most classical and fundamental methods for studying relationships among variables and predicting the value of a dependent variable is regression analysis. In regression analysis, through a set of independent variables, we attempt to achieve the best possible prediction of the dependent variable by fitting a linear model to the data. The coefficients of this linear model are typically estimated using the least squares method.

When the number of independent variables is large, there is an increased likelihood of encountering multicollinearity, that is, a high correlation among explanatory variables.
The presence of multicollinearity leads to higher standard errors for the estimated coefficients, wider confidence intervals, and reduced reliability of statistical inference. It therefore becomes evident that proper variable selection prior to model estimation is of great importance.

The application of penalized regression techniques constitutes an effective approach to addressing multicollinearity. Methods such as Ridge Regression, LASSO (Least Absolute Shrinkage and Selection Operator), and Elastic Net penalize variables that exhibit high correlation or low predictive contribution, while simultaneously improving the model's generalization ability. Moreover, they can also serve as variable selection techniques, identifying the most relevant predictors.

An alternative approach involves the use of regression trees, which are easy to interpret, construct, and visualize, yet they often underperform in terms of accuracy due to their tendency to overfit the training data.

In the present dissertation, we examine techniques aimed at reducing the overfitting of regression trees by introducing penalty functions that impose a cost on excessively large or complex trees. In addition, we explore the combined use of machine learning–based variable selection methods to investigate whether feature selection contributes to improving the accuracy and stability of regression tree models.

Dataset Description

To evaluate the application and effectiveness of the techniques on real-world data, two well-known datasets from the R datasets library are used: the *Airquality* dataset and the *Boston Housing* dataset.

The *Airquality* dataset contains daily measurements of air quality in New York City from May to September 1973.
It includes 154 observations and six variables, some of which contain missing values.
The response variable is Ozone concentration (Ozone), while the predictors include

Solar.R	Solar radiation, in langleys
Wind	average wind speed, in miles per hour
Temp	maximum daily temperature, in degrees F
Month	month of measurement
Day	day of measurement

Table 1: Airquality Dataset

The *Boston Housing* dataset, originally published by Harrison and Rubinfeld (1978), provides information on housing values in suburbs of Boston, Massachusetts.
It consists of 506 observations and 13 explanatory variables, with the response variable MEDV representing the median value of owner-occupied homes (in $1000s).
The predictors describe socioeconomic, demographic, and environmental characteristics such as the per capita crime rate (CRIM), the average number of rooms per dwelling (RM), the nitric oxides concentration (NOX), and the percentage of lower-status population (LSTAT).

CRIM	Per capita crime rate by town
ZN	Proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town
CHAS	Charles River dummy variable (=1 if tract bounds river, 0 otherwise)
NOX	Nitric oxides concentration
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centers
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil–teacher ratio by town
B	$1000 \times (Bk - 0.63)^2$, where Bk is the proportion of Black residents by town
LSTAT	Percentage of lower-status population

Table 2: Boston Housing Dataset

A descriptive statistical summary of the datasets is presented below.

A] Ozone dataset

Descriptive Statistics					
Statistic	N	Mean	St.Dev	Min	Max
Ozone	116	42.129	32.998	1	168
Solar.R	146	185.932	90.058	7	334
Wind	153	9.958	3.523	1.700	20.700
Temp	153	77.882	9.465	56	97
Month	153	6.993	1.417	5	9
Day	153	15.804	8.865	1	31

Table 3: Airquality Dataset Descriptives

As we can see the variables Solar.R and Ozone contain missing values. These missing observations were handled using mean imputation, where the missing entries were replaced by the corresponding variable means.

From the correlation plot below (Figure 1), we observe that Solar.R and Temp are positively correlated with the dependent variable Ozone, while Wind is negatively correlated with it. The variables Month and Day present correlations close to zero.

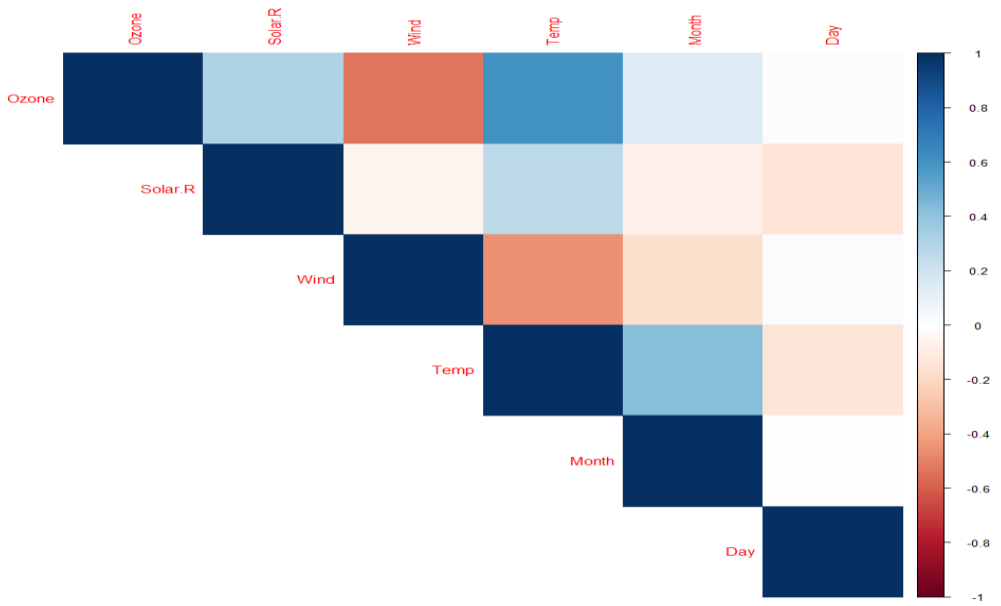

Figure 1:Airquality Dataset Correlation Plot

In Figure 2 below, the histograms of the main variables in the dataset are presented. The response variable *Ozone* exhibits a right-skewed distribution, while *Temp* and *Wind* appear more symmetrically distributed

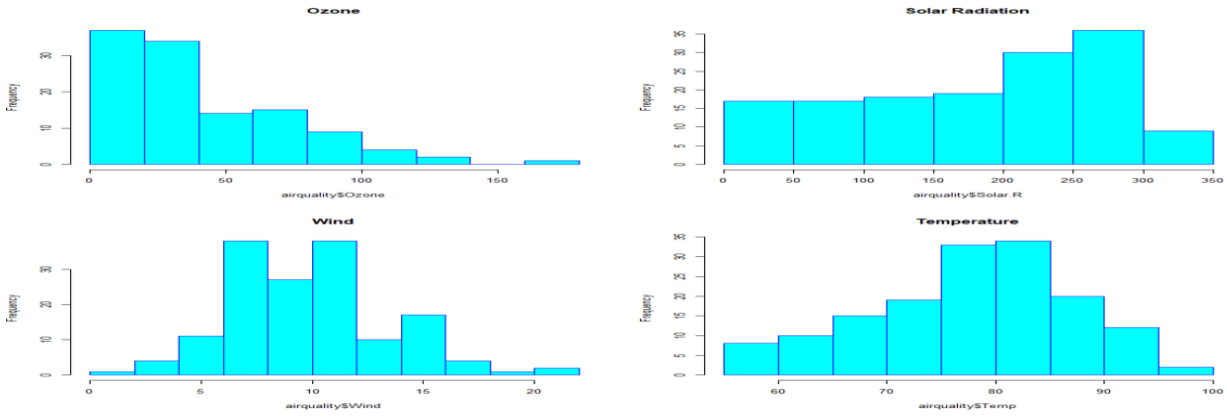

Figure 2: Airquality Dataset Histograms

Figure 3 presents the scatterplots between the response variable Ozone and selected predictors. A positive relationship can be observed between Temp and Ozone, while Solar.R also appears to be

positively associated with Ozone. In contrast, Wind seems to exhibit a negative relationship with the response variable.

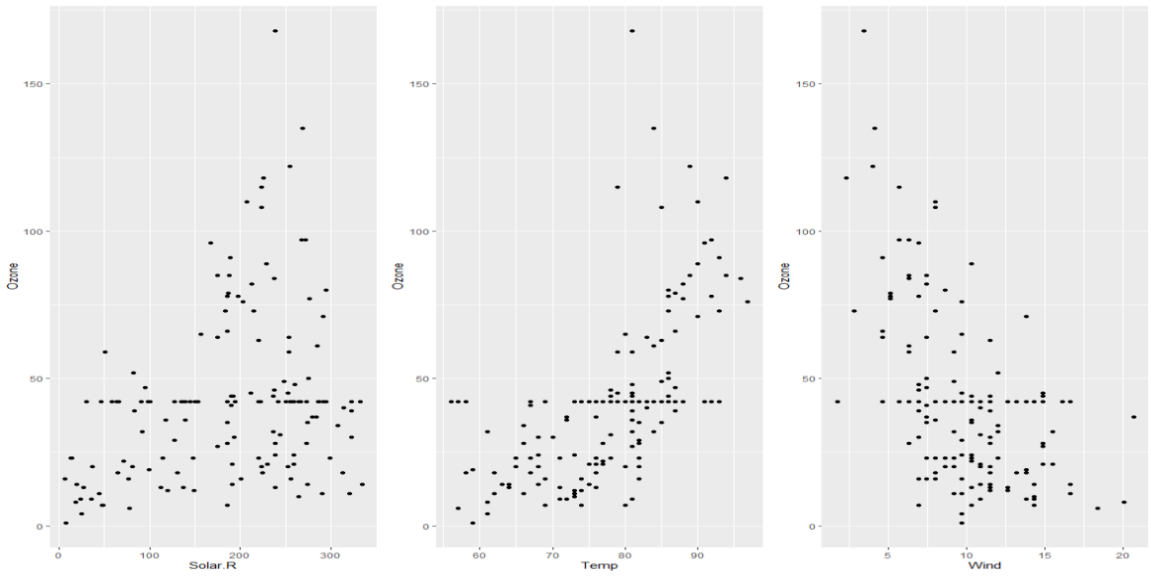

Figure 3: Air quality Dataset Scatterplots

B] Boston Housing dataset

The descriptives statistics appears in Table 4

Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Max
crim	506	3.614	8.602	0.006	88.976
zn	506	11.364	23.322	0.000	100.000
indus	506	11.137	6.860	0.460	27.740
chas	506	0.069	0.254	0	1
nox	506	0.555	0.116	0.385	0.871
rm	506	6.285	0.703	3.561	8.780
age	506	68.575	28.149	2.900	100.000
dis	506	3.795	2.106	1.130	12.126
rad	506	9.549	8.707	1	24
tax	506	408.237	168.537	187	711
ptratio	506	18.456	2.165	12.600	22.000
black	506	356.674	91.295	0.320	396.900
lstat	506	12.653	7.141	1.730	37.970
medv	506	22.533	9.197	5.000	50.000

Table 4: Boston Housing Dataset Descriptives

Figure 4 presents the correlation matrix of the variables in the dataset. The response variable *medv* appears to be positively correlated with *rm* and negatively correlated with *lstat*. In addition, several predictors exhibit notable correlations among themselves, indicating potential multicollinearity.

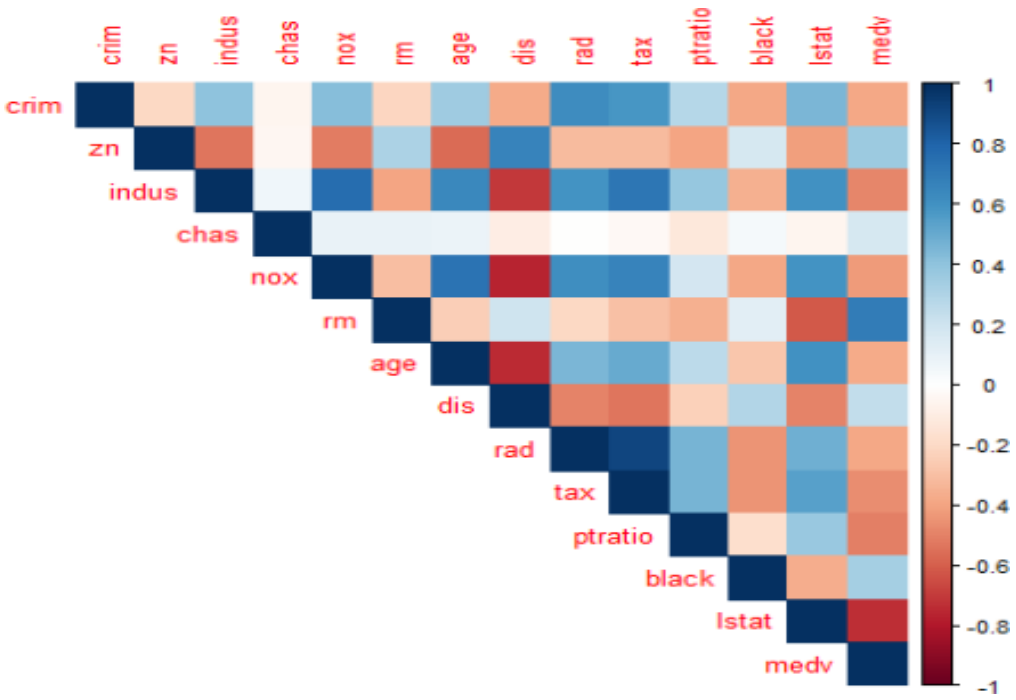

Figure 4: Correlation Plot Boston Housing Dataset

The histograms of the predictors that show the highest correlation with the response variable are presented below (Figure 5)

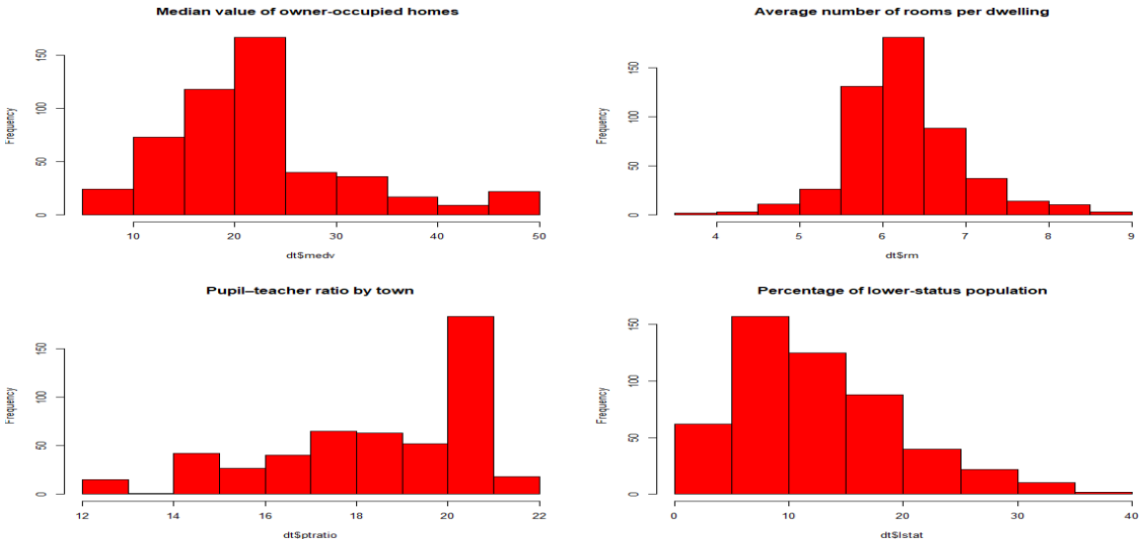

Figure 5: Boston Housing dataset Histograms

As we can see in Figure 6, the variable *rm* shows a relatively symmetric distribution, while *lstat* appears right skewed. The variables *medv* and *ptratio* display moderate variability across the dataset.

The figure below presents the scatterplots between the response variable *medv* and the predictors *lstat*, *rm*, and *ptratio*. A strong negative relationship can be observed between *lstat* and *medv*, while *rm* appears to be positively associated with housing prices. In contrast, *ptratio* shows a weaker negative relationship with the response variable.

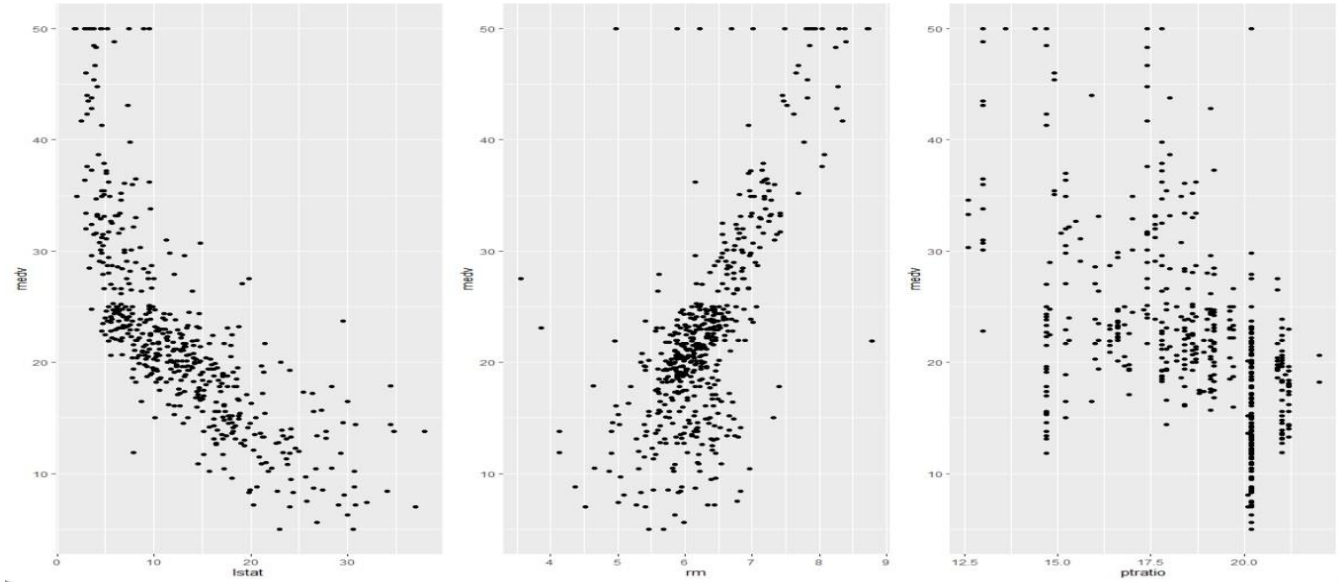

Figure 6: Boston Housing dataset Scatterplots

CHAPTER 1

PENALIZED REGRESSION

1.1 *Feature Selection*

In many cases, when a large set of independent variables is available for predicting a dependent variable, it is desirable mostly for predictive purposes to select a smaller subset of them. Variable selection is of particular practical importance, mostly for predictive purposes, as collecting data for a large number of variables can be both time-consuming and costly. Moreover, model interpretability improves when the number of variables is reduced. Including a large number of variables may also increase the risk of overfitting, where the model captures noise rather than the underlying data structure, leading to poor generalization performance on new data, as well as to poor predictions.

During the variable selection process, it is essential to consider the predictive contribution of each variable, as some variables may act as noise and provide little or no useful information to the model. In addition, the correlation among independent variables should be carefully examined, since high correlations may lead to the problem of multicollinearity, which is discussed in the following paragraph. This issue becomes even more pronounced in high-dimensional settings, where the number of variables may exceed the number of observations ($d > n$), making classical estimation methods unstable or even infeasible.

From a statistical perspective, variable selection is closely related to the bias–variance trade-off, as reducing the number of variables may increase bias but can significantly decrease variance, leading to improved predictive performance, which is frequently the purpose of the analysis.

In this chapter, we examine how penalized regression methods can lead to more parsimonious models by effectively performing automatic feature selection. Penalized regression methods address this problem by introducing a penalty term that shrinks coefficient estimates, reducing model complexity and, in some cases, setting certain coefficients exactly to zero. Among the

most widely used approaches are Ridge regression, Lasso, and Elastic Net, each of which imposes a different type of penalty and leads to different variable selection properties.

The aim is to construct models with high predictive accuracy while maintaining satisfactory interpretability. However, automatic variable selection methods such as stepwise selection, subset regression, backward elimination etc. Through these methods may not always yield the optimal model in terms of predictive performance and interpretability

1.2 *The problem of multicollinearity*

In high-dimensional data, it is common for the variables to exhibit high correlation among themselves. In this case, the phenomenon of multicollinearity arises.

The presence of this phenomenon in linear regression should make us particularly cautious, as it may easily lead to invalid conclusions as well as misleading interpretations.

Moreover, if one of the explanatory variables can be written as a linear combination of a subset of the remaining explanatory variables, then this would lead to design matrices for which the determinant of the matrix $(X'X)$ is nearly zero. In such a case, the computation of the ordinary least squares estimator

$$\hat{\beta} = (X'X)^{-1}X'Y \qquad (1)$$

is subject to rounding errors, resulting in a significant loss of accuracy, where:

- X is the n x d design matrix associated with the d covariates $X_1, X_2, X_3, .., X_d$,
- β is the (d +1) x 1 vector of estimators for the d-dimensional vector of parameters and,
- Y is a n-dimensional vector of responses.

Due to the above problems that the existence of multicollinearity may create, it is necessary to have simple ways in order to detect it. Some indications of multicollinearity related to the linear model are the following: the predictions of the full model are close to the predictions obtained

when a variable is removed; the standard errors of the full model are considerably larger compared to the model where a variable has been removed; and the confidence intervals are wider compared to the model without that variable.

Another way to detect multicollinearity is through the variance inflation matrix (VIF) index

$$\mathrm{VIF}_k = \frac{1}{1 - \mathrm{R}^2{}_k} \; k = 1,2,3,4,..,d - \{k\}$$

where R_k^2 is the coefficient of determination of the model that uses X_k as the dependent variable and the remaining $d - 2$ variables as independent variables.

The larger the value of R_k^2, the larger the corresponding VIF_k will be.

The index VIF_k indicates how much the variance of an estimated coefficient $\mathrm{Var}(\beta_k)$ increases when the corresponding variable exhibits multicollinearity.

Empirically, the following holds:[1]

- If $VIF_k \approx 1$, then X_k does not present a multicollinearity problem.
- If $VIF_k > 10$, then X_k presents a multicollinearity problem

Additionally, we can use the average VIF as a criterion for the absence of multicollinearity in the model under consideration:

$$\bar{VIF} = \frac{1}{d} \sum_{k=1}^{d} VIF_k$$

If $\bar{VIF} = 1$, then it is easy to see that $VIF_k = 1$ for all k, and therefore no variable exhibits multicollinearity with respect to the others.

[1] M. Koutras and C. Evangelaras, Ανάλυση παλινδρόμησης: Θεωρία και εφαρμογές [Regression Analysis: Theory and Applications] (Tsotras, 2018).

However, if $\bar{VIF}$ is significantly greater than 1, this means that the VIF_k for one or more values of k is greater than 1, and thus there is an indication of multicollinearity.

Tables 5 and 6 present the Variance Inflation Factor (VIF) values for the Airquality and Boston Housing datasets of the introductive section. It can be observed that in the Boston Housing dataset some variables exhibit VIF values close to 10, suggesting the presence of moderate multicollinearity. In contrast, the Airquality dataset shows low VIF values, indicating little to no multicollinearity among the predictors.

Table 5 Boston Housing dataset VIF

Variable	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat
VIF	1.792	2.299	3.992	1.074	4.394	1.934	3.101	3.956	7.484	9.009	1.799	1.349	2.941

Table 6 Airquality dataset VIF

Variable	Temp	Wind	Month	Solar.R	Day
VIF	1.685	1.275	1.274	1.144	1.033

1.3 *Tradeoff between Prediction Accuracy and Model Interpretability*

In many applications of statistical learning the models we develop are required to balance two fundamental properties: interpretability and predictive accuracy.

For instance, the classical linear regression model is highly interpretable, as each coefficient represents the marginal effect of an explanatory variable on the response variable, holding all other variables constant. In contrast, more complex models such as neural networks or Support Vector Machines are difficult to directly understand how predictions are generated.

However, complex models frequently achieve higher predictive accuracy, especially when the true relationship between variables is nonlinear.[2]

When interpretability is the primary objective – as for instance, in economic, medical, or regulatory applications — simpler models are typically preferred, even if they are slightly inferior in terms of predictive performance. On the other hand, when the main goal is prediction, emphasis is placed on maximizing predictive accuracy, regardless of model complexity.

[2] A.Karagrigoriou , Διαλέξεις μαθήματος Μηχανικής Μάθησης [Machine Learning course lectures], MSc in Applied Statistics, University of Piraeus

Nevertheless, it is not uncommon for a simpler model to perform better than a more complex one. This seemingly paradoxical result can be explained by the phenomenon of overfitting, where highly flexible models capture not only the underlying structure of the data but also the random noise, thereby reducing their ability to generalize to new observation.

Therefore, the key objective is not to make an absolute choice between accuracy and interpretability, but rather to identify an appropriate balance between model complexity and predictive performance - that is, to find the point that optimally trades off bias and variance.

1.4 *Bias -Variance Trade-off*

To understand the bias–variance trade-off, we must first clarify what is meant by variance and bias in the context of a statistical learning method.

Variance refers to the amount by which the model's prediction would change if we estimated it using different training datasets. In other words, it measures the sensitivity of a method to fluctuations in the training data. Ideally, we would like the estimated function to remain relatively stable across different training sets. However, highly flexible or complex methods tend to exhibit higher variance, since they adapt closely to the specific patterns of the observed data.

Bias, on the other hand, refers to the error introduced by approximating a complex real-world problem with a simplified model and arises from the assumptions imposed by the model. For example, assuming a linear relationship between predictors and the response variable when the true relationship is nonlinear introduces systematic error. Simple models typically have higher bias because they restrict the functional form of the relationship.

A general rule is that as model complexity increases; variance tends to increase while bias decreases. The rate at which these two quantities change determines whether the Mean Squared Error (MSE) increases or decreases:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 .$$

where, Y_i is the i^{th} observation and $\hat{Y}_i$ is the prediction of the i^{th} observation.

Initially, as model complexity grows, the reduction in bias dominates the increase in variance, leading to a decrease in the overall MSE. However, beyond a certain level of complexity, further

reductions in bias become negligible, while variance continues to increase. As a result, the MSE begins to rise. This explains the bath-shaped curve of the test error as a function of model complexity and highlights the existence of an optimal level of complexity that balances bias and variance.

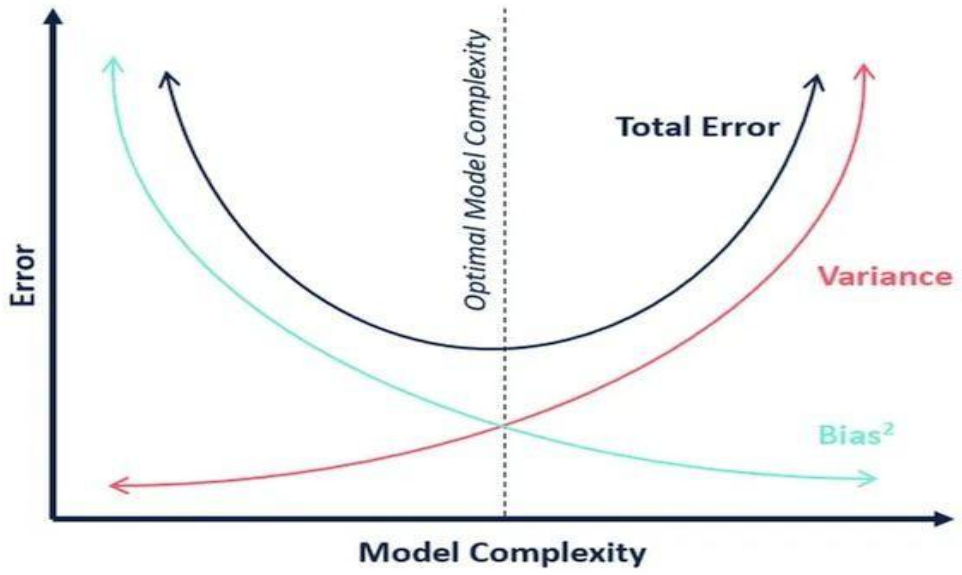

Figure 7: Error vs Complexity

1.5 *Ridge Regression*

In large datasets, it is very common to encounter the phenomenon of multicollinearity. The negative impact of collinearity on the Least Squares (LS) estimator in a regression context is well known, as it leads to unstable estimates and inflated standard errors. To address this problem, Hoerl and Kennard (1970) proposed Ridge Regression, a regularization technique that adds a penalty term to the LS objective function. Unlike variable selection methods, Ridge Regression does not remove predictors from the model but rather shrinks the coefficients of some variables toward zero.

The estimators of Ridge regression are computed by minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{d}\beta_k x_{ik})^2 + \lambda\sum_{k=1}^{d}\beta_k^2 = \text{RSS} + \lambda\sum_{k=1}^{d}\beta_k^2$$

Where λ is a positive tuning parameter; for $\lambda = 0$, the solution coincides with the classical OLS estimators.

An alternative way of expressing the same problem is to minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{d}\beta_k x_{ik})^2$$

under the restriction:

$$\sum_{k=1}^{d}\beta_k^2 \leq t$$

where t depends on λ.

As the parameter λ increases, the regression coefficients are increasingly shrunk toward zero. In this way, the variance of the coefficient estimates decreases, leading to a more stable and accurate model

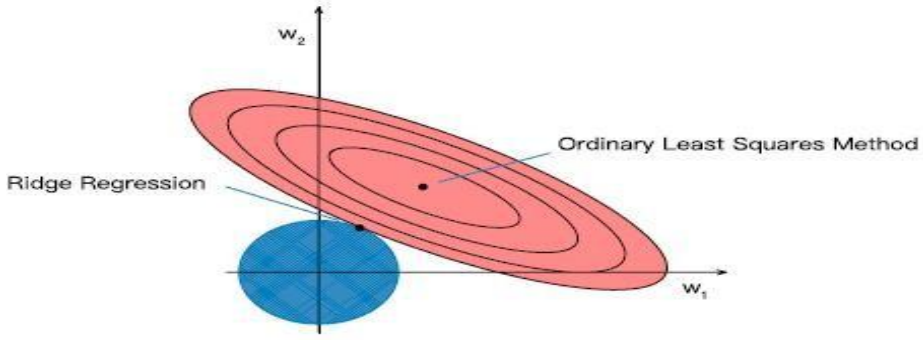

Figure 8: Geometric interpretation of Ridge restriction.

In the figure above, the constraint $\sum_{k=1}^{d} b_k \leq t$ is represented geometrically in the context of minimizing the least squares expression in two dimensions.

Each ellipse corresponds to points where the Residual Sum of Squares (RSS) takes the same value. Smaller the ellipse smaller the RSS. The minimum RSS is found at the center of the smallest ellipse, which corresponds to the Ordinary Least Squares (OLS) estimators. It can therefore be observed that solution exists to the minimization

$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{d}\beta_k x_{ik})^2$ under the $\sum_{k=1}^{d} b_k \leq t.$

Let $x = [x_1, x_2, \ldots x_n]^T, \|x\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$

Ridge estimators are calculated as

$$\beta^{\text{ridge}} = \min\{\|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2\} = \min\{Y'Y - 2\beta'X'Y + \beta'\beta + \lambda\|\beta\|_2^2\}$$
$$= min\{-2\beta'X'Y + \|\beta\|_2^2 + \lambda\|\beta\|_2^2\}.$$

Differentiating the above expression with respect to β

$$\frac{d\beta^{\text{ridge}}}{d\beta} = -2X'Y + 2\beta + 2\lambda\beta$$

If the vectors of X are orthogonal, the least squares estimators are computed from the following relation: $\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{Y}$ (see equation (1))

$$\frac{d\beta^{\text{ridge}}}{d\beta} = -2\beta_{LSE} + 2\beta + 2\lambda\beta$$

$$\frac{d\beta^{\text{ridge}}}{d\beta} = 0 \Leftrightarrow -2\beta_{LSE} + 2\beta + 2\lambda\beta = 0 \Leftrightarrow \beta_k^{ridge} = \frac{1}{1+\lambda}\beta_{LSE}, \ \lambda \geq 0.$$

The ridge estimators β^{ridge} exhibit lower variance and, consequently, smaller squared errors; however, they are biased. The bias of the model is reflected in the intercept term β_0, which remains unaffected by the ridge penalty constraint.

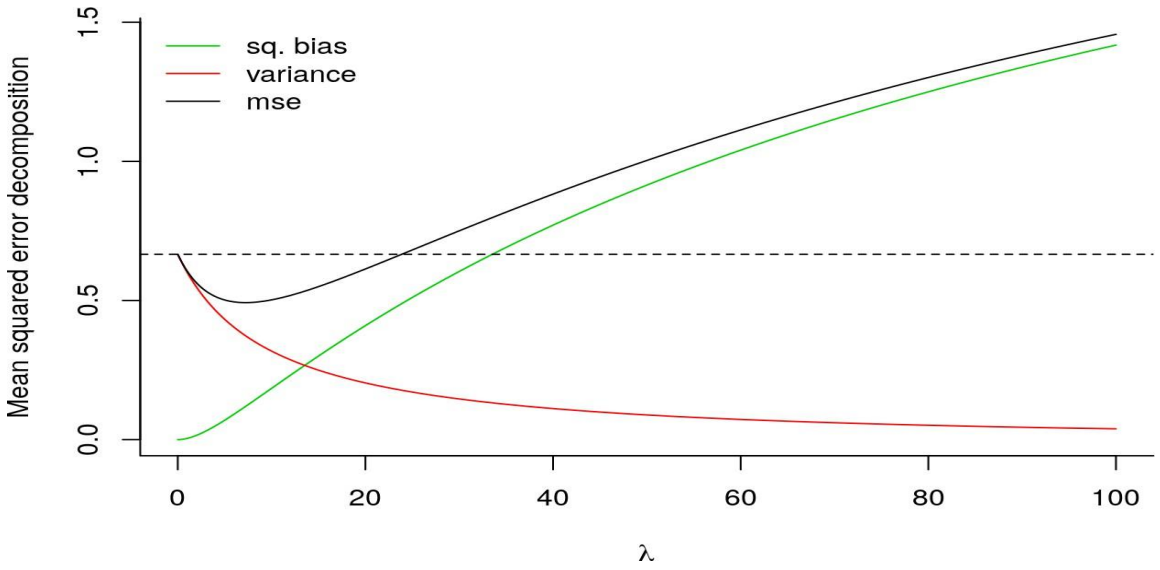

Figure 9: Variation of squared bias, variance, and mean squared error (MSE) with respect to the regularization parameter

In the figure above, the effect of the regularization parameter λ on the bias, variance, and mean squared error (MSE) is illustrated. As λ increases, the bias also increases — and consequently, the estimate of the intercept $\beta_0$tends to increase as well. Moreover, the mean squared error rises according to the relationship

$$\mathrm{MSE}(\beta) = VAR(\beta) + bias(\beta)^2$$

At the same time, we observe a decrease in the variance of the coefficient estimates as λ grows. The objective is to determine the optimal value of λ that minimizes the MSE — achieved through a substantial reduction in variance accompanied by only a moderate increase in bias. The selection of the optimal value of λ constitutes a complex problem. According to Fan and Tang, obtaining the exact optimal $\lambda$is not practically feasible. Fan and Li (2001) proposed evaluating a sequence of candidate λ values, fitting the corresponding models, and then selecting the most appropriate one using well-known information criteria such as AIC or BIC. However, this approach often fails to perform adequately when the number of predictors is large.

Tibshirani (1996) considered cross-validation as a more practical technique for determining the tuning parameter λ. In this method, the dataset is initially divided into two parts: the first is used to estimate the coefficients β^{ridge}, while the second is used to identify the value of λ that minimizes the cross-validation error. Since this approach does not utilize all available data for coefficient estimation and thus results in a loss of information, a generalized version known as K-fold cross-validation is typically preferred. In K fold cross-validation, the sample is randomly partitioned into K approximately equal subsets (folds). Each subset is used once as a validation set, while the remaining $K - 1$ subsets form the training set.

The procedure is repeated K times, yielding an estimated prediction error MSE_i for each iteration.

The overall cross-validation error is given by

$$CV_K = \frac{1}{K}\sum_{i=1}^{K} MSE_i$$

Typical choices for K are 5 or 10.

The optimal value of λ is then selected as the one minimizing the average cross-validation error across all folds.

1.6 *LASSO Regression*

A main limitation of Ridge regression is that, although the estimated coefficients are shrunk towards zero, they never become exactly zero. This characteristic may lead to difficulties in model interpretability, since models with a large number of predictors tend to be complex and less transparent. In cases where many independent variables are available, it is often desirable to identify only those predictors that have the strongest effect on the dependent variable.

To address this issue, the LASSO (Least Absolute Shrinkage and Selection Operator) method was proposed by Robert Tibshirani (1996). In the LASSO approach, the coefficients are shrunk similarly to Ridge regression, but unlike Ridge, some coefficients are forced exactly to zero, effectively performing variable selection in addition to regularization. Therefore, the LASSO method is particularly useful for high-dimensional datasets, where the number of predictors is large and it is important to determine which variables have the most significant influence on the response.

The LASSO estimators are obtained by minimizing the following:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{d}\beta_k x_{ik})^2 + \lambda \sum_{k=1}^{d}|\beta_k| = \text{RSS} + \lambda \sum_{k=1}^{d}|\beta_k|$$

where $\lambda \geq 0$ is a tuning parameter that controls the amount of regularization applied to the model.

An alternative way of expressing the same problem is to minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{d}\beta_k x_{ik})^2$$

under the restriction:

$$\sum_{k=1}^{d}|\beta_k| \leq t$$

where t is dependent by λ.

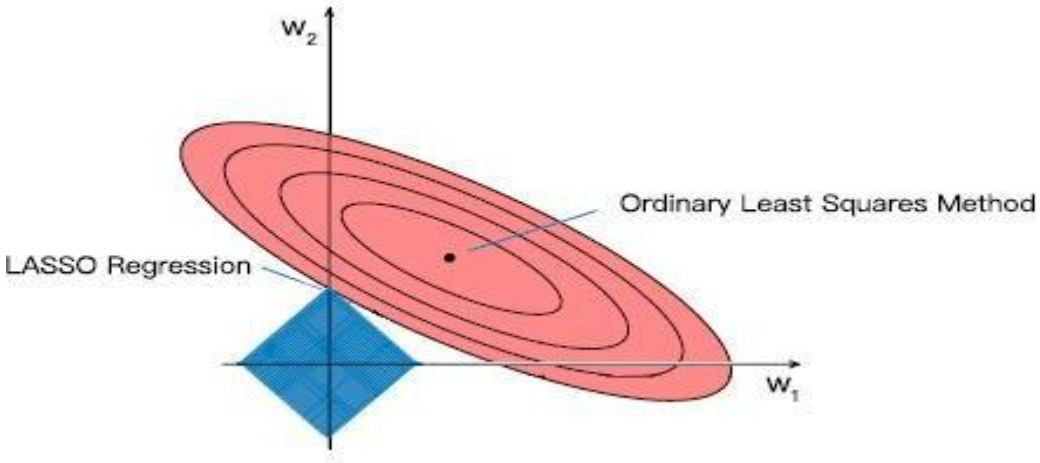

Figure 10: Geometric interpretation of LASSO restriction

In two dimensions, the LASSO constraint is given by
$\sum_{k=1}^{p} |\beta_{\boldsymbol{\kappa}}| \leq t$ which forms a diamond-shaped (or rhombus) region in the coefficient space.

The ellipses represent the contours of the least squares loss function,

$$\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{k}^{d} \beta_{\boldsymbol{\kappa}} x_{ik})^2 .$$

The corners of the diamond correspond to points where some coefficients are exactly zero. Thus, when the minimum of the objective function occurs at one of these corners, the corresponding coefficient becomes zero, illustrating why LASSO tends to produce sparse models and performs variable selection naturally.

LASSO estimators are calculated by

$$\beta^{\text{LASSO}} = \min\{\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1\} = \min\{Y'Y - 2\beta'X'Y + \beta X'X\beta + \lambda\|\beta\|_1\}$$
$$= \min\{-2\beta'X'Y + \beta X'X\beta + \lambda\|\beta\|_1\} .$$

If the vectors of *X* are orthogonal, the least squares estimators are computed from the following relation: $\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y}$ (see equation (1))

$$\beta^{\text{LASSO}} = min \sum_{k=1}^{d} -2\beta_{LSE}\beta_{\boldsymbol{\kappa}} + \beta_{\boldsymbol{\kappa}}^2 + \lambda|\beta_{\boldsymbol{\kappa}}|$$

We aim to minimize the following expression

$$-2\beta_{\text{LSE}}\beta_{\boldsymbol{\kappa}} + \beta_{\boldsymbol{\kappa}}^2 + \lambda|\beta_{\boldsymbol{\kappa}}|$$

Since each term of the above sum corresponds to a distinct coefficient $\beta_k, k = 1,2,..,$d. Consequently, each term can be minimized independently in order to obtain the optimal value for the respective coefficient.

To minimize the above quantity, the least squares estimator and the LASSO estimator must have the same sign. If they have opposite signs, their product becomes negative and the term $-2\beta_{LSE}\beta_k$ increases, which in turn raises the value of the objective function.

In the case where $\beta_{LSE} > 0$ then $\beta_k \geq 0$

$$-2\beta_{LSE}\beta_{kc} + {\beta_k}^2 + \lambda\beta_k.$$

Differentiating the above expression with respect to β

$$\beta^{LASSO} = \beta_{LSE} - \frac{\lambda}{2}, \lambda \geq 0$$

Similarly, in the case where $\beta_{LSE} < 0$ the corresponding LASSO coefficient β_k must also satisfy $\beta_k \leq 0$

$$-2\beta_{LSE}\beta_k + {\beta_k}^2 - \lambda\beta_k.$$

Differentiating the above expression with respect to β we get:

$$\beta^{LASSO} = \beta_{LSE} + \frac{\lambda}{2}.$$

So, the LASSO estimator can be expressed as:

$$\beta^{LASSO} = \begin{cases} \beta_{\text{LSE}} - \frac{\lambda}{2}, \beta_{\text{LSE}} > 0 \\ \beta_{\text{LSE}} + \frac{\lambda}{2}, \beta_{\text{LSE}} < 0 \end{cases}.$$

The main difference between the LASSO and Ridge regression is that, for an appropriate choice of the tuning parameter λ, some coefficients become exactly equal to zero rather than merely approaching zero. This occurs when the absolute value of the ordinary least squares (OLS) estimator is smaller than $\frac{\lambda}{2}$. As a result, the LASSO method performs variable selection, which is particularly desirable when dealing with a large number of predictors. As in Ridge regression, selecting the optimal value of λ in LASSO regression is a complex problem. The tuning parameter λ determines the amount of penalization and hence the degree of shrinkage applied to the coefficients. Small values of λ lead to less regularization and more complex models, whereas larger values impose stronger penalties and yield simpler models. The optimal λ is typically chosen via k-fold cross-validation, where the data are split into *k* folds and the model is refitted *k* times, each time leaving out one fold for validation. The λ that minimizes the mean squared prediction error (MSE) across the validation folds is denoted as λ_{min}.

Alternatively, one may select the λ within one standard error of the minimum (λ_{1se}), which applies a stronger penalty and results in a simpler, more parsimonious model, often with comparable predictive accuracy.

1.7 Elastic Net

Tibshirani (1996) and Fu (1998) compared the predictive performance of the LASSO and Ridge regression methods and concluded that neither method dominates the other.

In high-dimensional datasets, the LASSO method is particularly appealing, as it tends to produce simpler and more interpretable models. On the other hand, despite its attractiveness and its good performance in many situations, there are scenarios in which the LASSO method exhibits certain limitations:

(A) When the number of predictors exceeds the sample size ($p >> n$), the LASSO can select at most *n* variables with nonzero coefficients. This limitation arises from the geometric

nature of the L1 optimization problem. Moreover, the LASSO is well-defined only when there exists an upper bound on the L_1 norm of the coefficients, that is, $\| \beta \|_1 \leq t$ for some finite value of *t*.

(B) In situations where a group of predictors are highly correlated with each other, the LASSO usually selects a single representative variable from the group, without distinguishing which of them carries the actual predictive information.

(C) When the number of observations exceeds the number of predictors ($n > p$) and substantial correlations exist among the predictors, Ridge regression has been empirically shown to outperform LASSO in terms of prediction accuracy (Tibshirani, 1996).

Clearly, scenarios (A) & (B) make LASSO an inappropriate variable selection method.

Zou and Hastie (2005) proposed a penalization method that performs better than LASSO, known as the Elastic Net. Their objective was to develop a technique that retains the advantages of LASSO while overcoming its limitations under scenarios (A) and (B). The Elastic Net combines the properties of Ridge and LASSO regression:

- as it achieves continuous shrinkage
- performs automatic variable selection and,
- is capable of selecting groups of correlated predictors.

Zou and Hastie metaphorically described the Elastic Net as a *"*stretchable fishing net that retains all the big fish." [3]Both simulation experiments and empirical data analyses have demonstrated that the Elastic Net often outperforms LASSO in terms of predictive accuracy.

Suppose that we have a dataset with n observations and p independent variables with:

- Y= $(y_1, y_2, \ldots, y_n)$ is the response variable,
- **X=** $(X_1, X_2, X_{3,\ldots,} X_p)$ the model matrix and,
- $x_j = (x_{1j}, x_{2j}, \ldots, x_{nj})^T$ the predictors after scaling $\sum_{i=1}^{n} y_i = 0, \sum_{i=1}^{n} x_{ij} = 0, \sum_{i=1}^{n} x_{ij}^2 = 1, \ j = 1, .., p.$

[3] Zou, H., & 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. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Zou and Hastie (2005) defined the naive elastic net criterion as follows:

$$L(\lambda, \alpha, \beta) = |\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|^2 + \lambda(1-\alpha)|\beta|^2 + \lambda\alpha|\beta|_1 \quad (2)$$

where $|\beta|^2 = \sum_{j=1}^{d} \beta_j{}^2$ and $|\beta|_i = \sum_{j=1}^{d} |\beta_j|$.

The naive elastic net estimator β is the minimizer of equation (2).

The proposed procedure can be regarded as a regularized least squares framework.

$$\beta = argmin\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2 \text{ subject to } \lambda_1|\beta|_1 + \lambda_2|\beta|^2 \leq t \; for \; t > 0 .$$

The term $\lambda_1|\beta|_1 + \lambda_2|\beta|^2$is known as elastic net penalty which represents a combination of the LASSO and Ridge penalties. The parameter $\alpha \in [0,1]$controls the relative contribution of the Ridge and LASSO penalties. Setting $\alpha = 1$ yields the LASSO solution, while $\alpha = 0$ leads to Ridge regression.

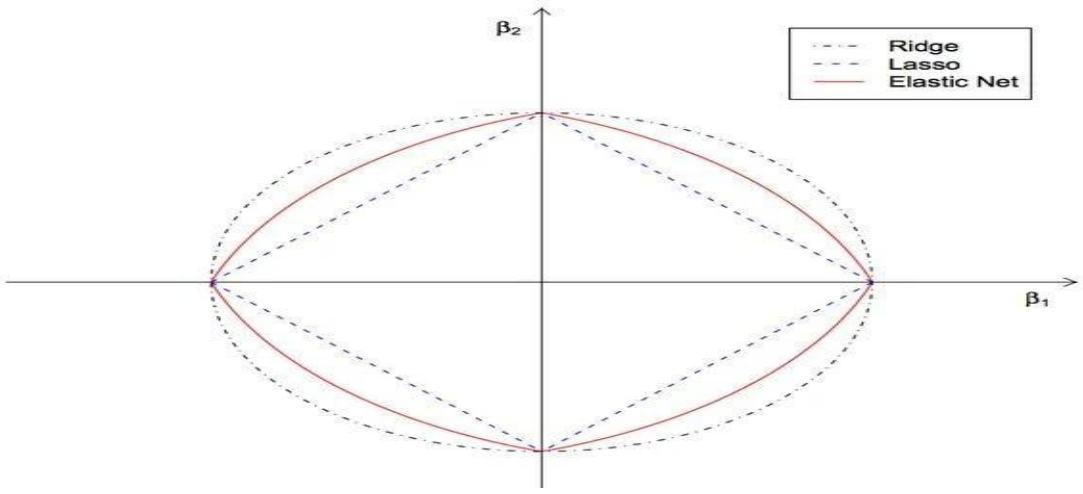

Figure 11: Geometric comparison of Ridge, LASSO, and Elastic Net in a two-dimensional space

It should be noted that the selection of the most appropriate method cannot be known as a priori, as it depends on the structure and nature of the data. LASSO and Elastic Net techniques tend to produce simpler models with fewer parameters through the process of variable selection, which results in models that are more interpretable. However, when all predictors are relevant, setting

some coefficients exactly to zero may lead to information loss, making the LASSO less effective compared to Ridge regression. The Elastic Net is more computationally demanding, since for each possible value of the parameter λ_1, it must search for the optimal value of λ_2.
The Ridge regression shrinks all coefficients by the same proportion, providing more stable estimates and better predictive performance, whereas the LASSO shrinks the coefficients toward zero by varying amounts, setting some exactly to zero, thus producing models with fewer variables and greater interpretability.

1.8 Applications

We proceed with the application of the methods presented in this chapter on Airquality and Boston Housing datasets. We begin with the Airquality dataset, which contains a relatively small number of variables that exhibit moderate correlations among them.

The following table presents the selected values of the hyperparameters λ and α for the Ridge, Lasso, and Elastic Net models on the *Airquality* dataset. These values were obtained through a cross-validation procedure with $k = 5$

Model	α	λ
Ridge	0	47.387
Lasso	1	8.675
Elastic Net	0.700	9.375

Table 7: Models Hyperparameters

Figure 12 illustrates the coefficient paths of the ridge regression model applied to the Airquality dataset as a function of the L2 norm of the coefficients. Each curve represents a regression coefficient and shows how it evolves as the L2 norm increases. As the L2 norm becomes larger, the coefficients are allowed greater flexibility and move further away from zero, approaching their ordinary least squares estimates. Conversely, for smaller values of the L2 norm, the coefficients are more constrained and shrink closer to zero. This behavior reflects the trade-off between model flexibility and regularization imposed by ridge regression.

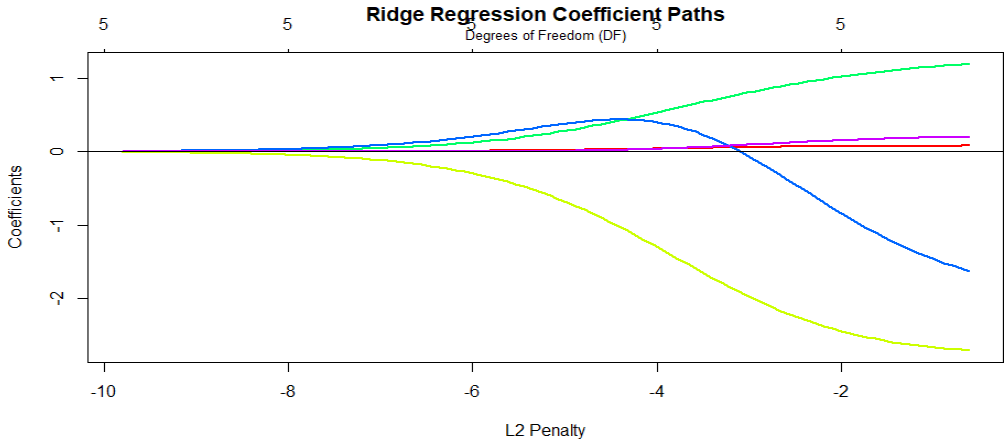

Figure 12: Ridge Coefficient Paths

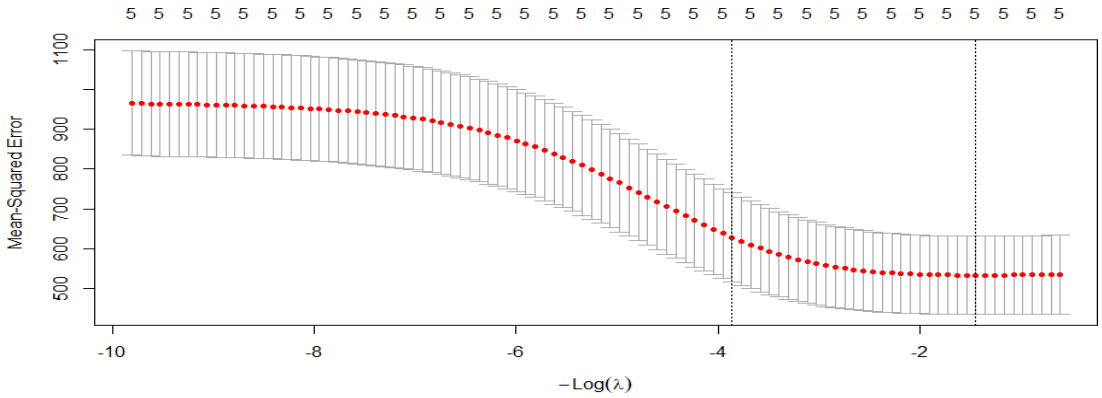

Figure 13: MSE vs Log(λ)

Figure 13 presents the cross-validation results for the ridge regression model. The horizontal axis represents the values of the regularization parameter λ on a logarithmic scale, while the vertical axis shows the corresponding mean squared prediction error. Each red point corresponds to the average cross-validation error, with the grey bars indicating one standard error. The right line represents the value of λ that minimizes the prediction error, while the left dashed line represents a more regularized model within one standard error from the minimum. This plot allows us to select an appropriate level of regularization that balances predictive performance and model stability.

We now proceed with the Lasso regression model, which introduces an alternative form of regularization that can lead to sparse solutions.

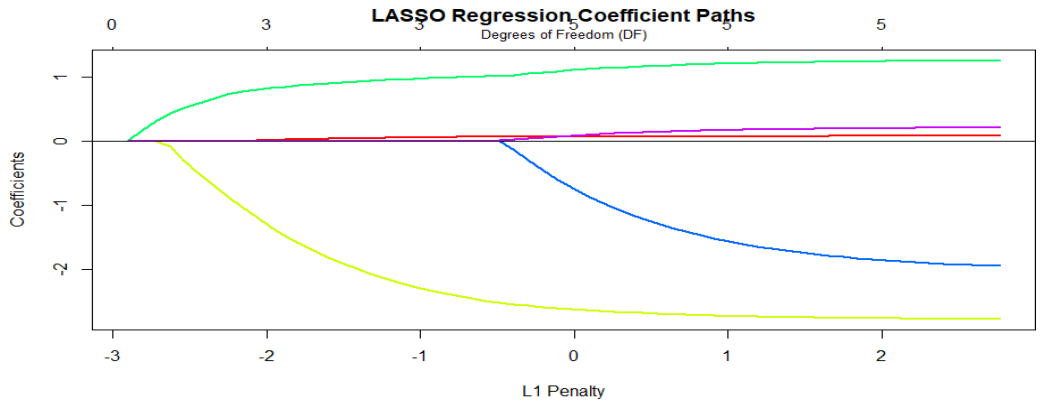

Figure 14: LASSO coefficient paths

Figure 14 illustrates the coefficient paths of the lasso regression model applied to the Airquality dataset as a function of the L1 norm of the coefficients. As the L1 norm becomes larger, the coefficients are allowed greater flexibility and move further away from zero, approaching their ordinary least squares estimates. Conversely, for smaller values of the L-norm, the coefficients are more constrained and shrink closer to zero and some of them are becoming equal to zero. This property enables lasso regression to perform variable selection by effectively removing less important predictors from the model.

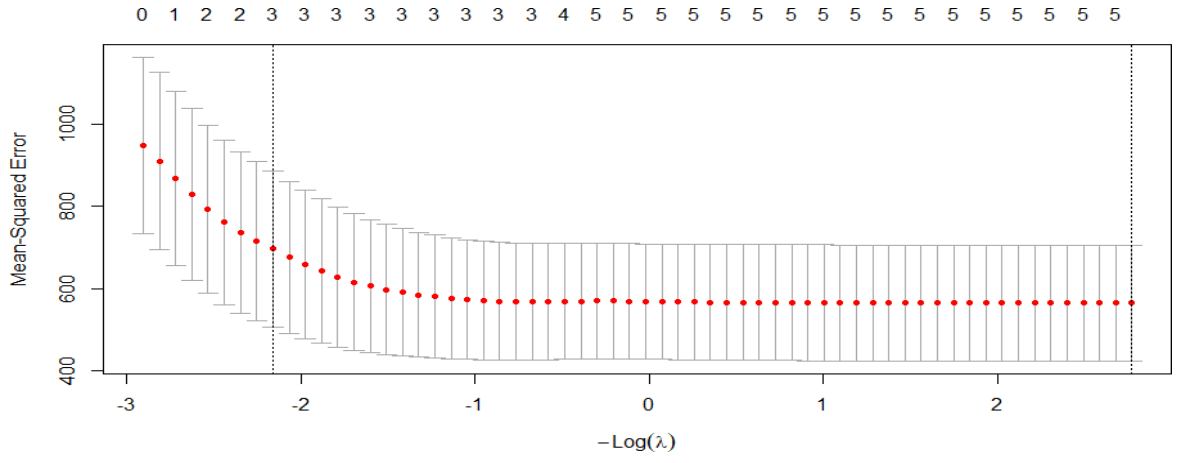

Figure 15: MSE vs Log(λ)

In this case, the value of λ_{1se}results in a sparser model by selecting 3 predictors, where some coefficients are shrunk exactly to zero, effectively removing less important predictors.

Having examined Ridge and Lasso regression, we now proceed to Elastic Net, a regularization method that combines the shrinkage property of ridge regression with the variable selection ability of Lasso.

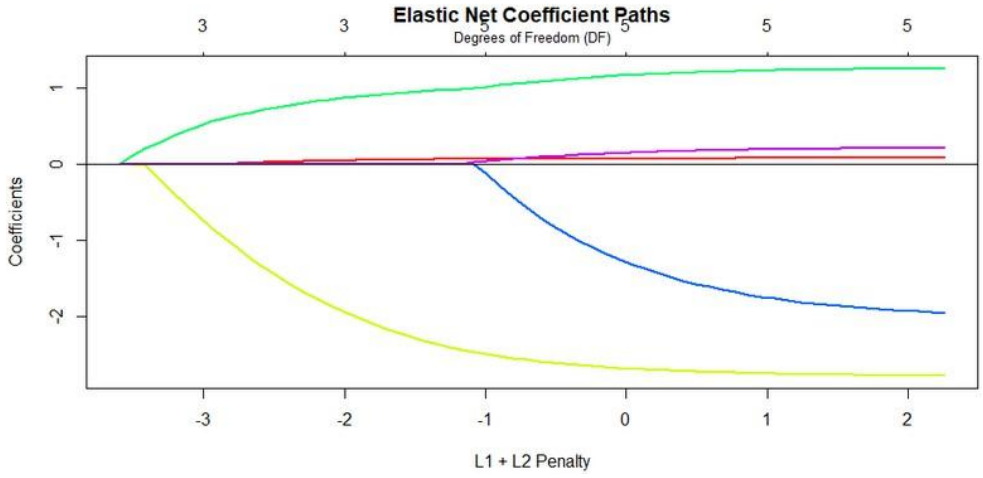

Figure 16: Elastic Net coefficients paths

Figure 16 illustrates the coefficient paths of the elastic net regression model applied to the Airquality dataset as a function of the combined L1 and L2 norm of the coefficients. As the norm increases, the coefficients are allowed greater flexibility and move further away from zero, corresponding to weaker regularization. Conversely, for smaller values of the norm, the coefficients are more constrained and shrink towards zero, with some of them becoming exactly zero. This demonstrates that elastic net combines the shrinkage effect of ridge regression with the variable selection property of lasso, leading to a balanced and interpretable model.

The same modeling procedure was applied to the Boston Housing dataset. Since the methodology is identical to that presented above, we focus here on the main results and the comparative performance of ridge, lasso and elastic net.

The table above presents the selected values of the hyperparameters λ and α for the Ridge, Lasso, and Elastic Net models on the *Boston Housing* dataset. These values were obtained through a cross-validation procedure.

Model	α	λ
Ridge	0	4.599
Lasso	1	0.303
Elastic Net	0.700	0.571

Table 8: Model Hyperparameters

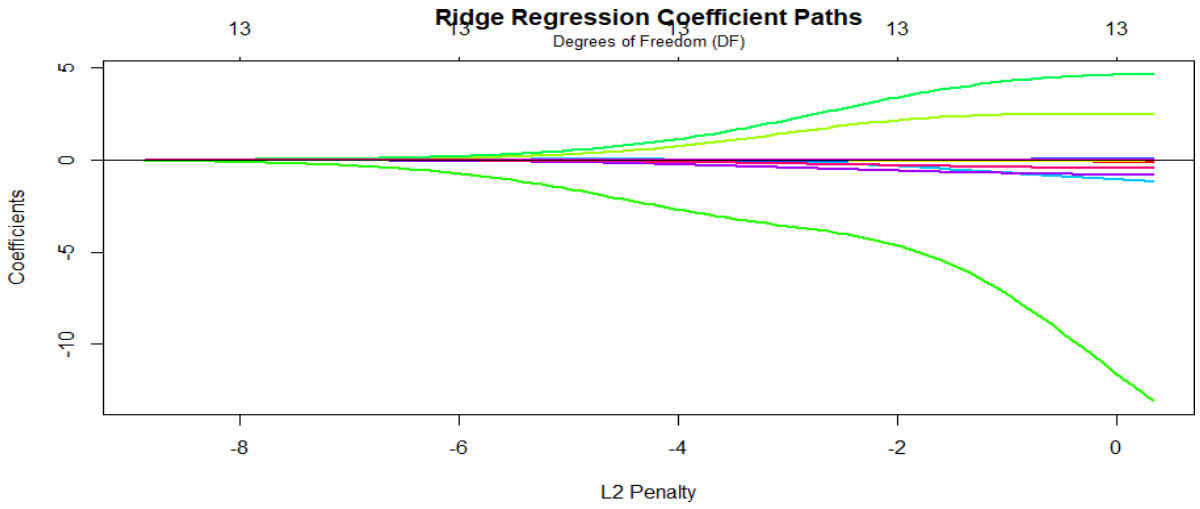

Figure 17: Ridge coefficients path

Figure 17 illustrates the ridge regression coefficient paths for the Boston Housing dataset.

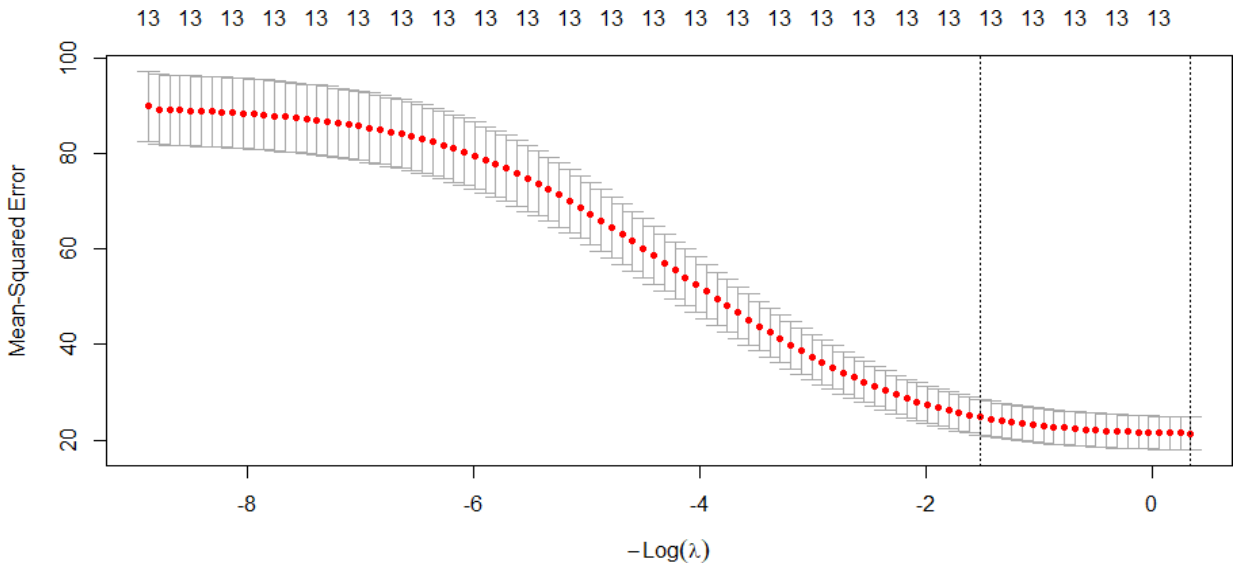

Figure 18: MSE vs Log(λ)

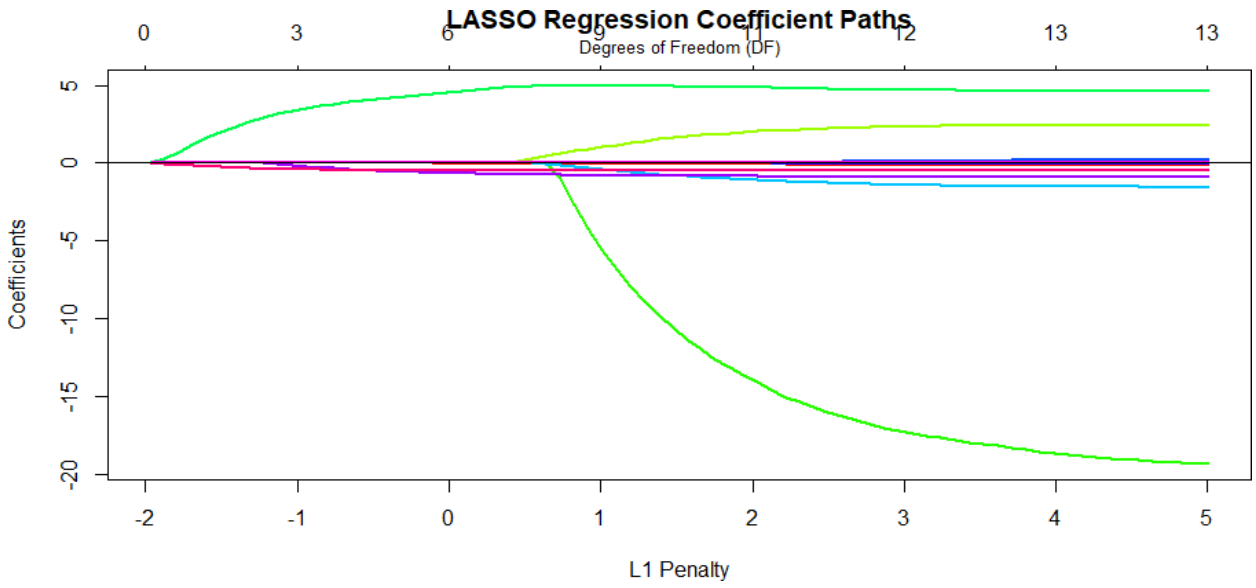

Figure 19: Lasso coefficient path

It is observed that several variables tend to become exactly zero as the L1 norm decreases, highlighting the sparsity induced by the lasso regression method (see fig.19)

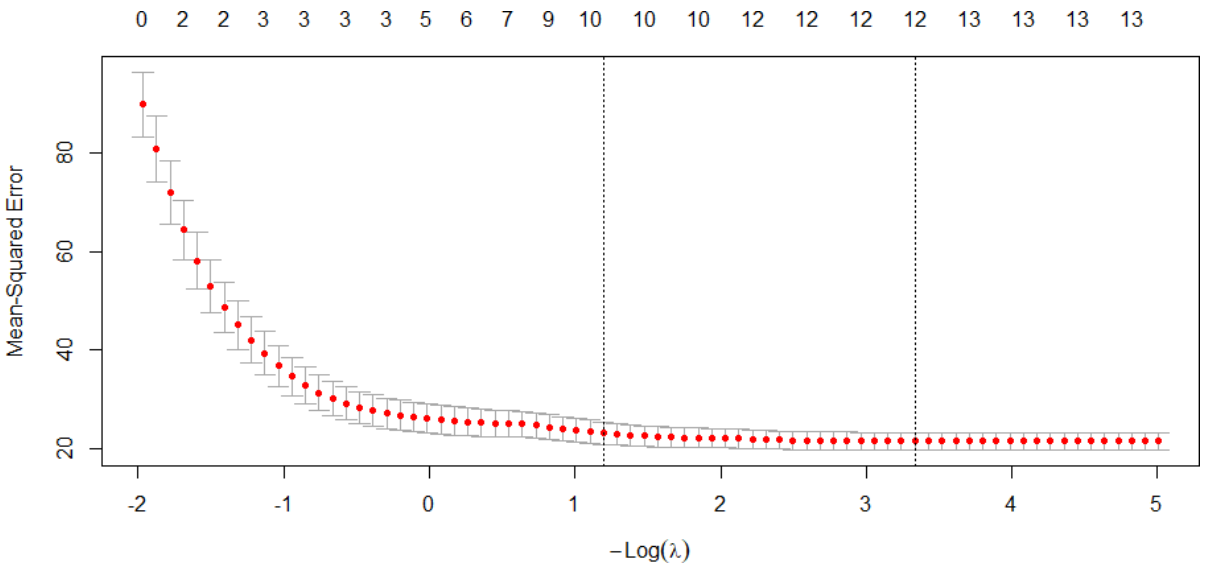

Figure 20: LASSO Cross-Validation Curve

The more conservative λ_{1se} rule results in a simpler model, selecting approximately 10 predictors, thus reducing complexity while maintaining good predictive performance.

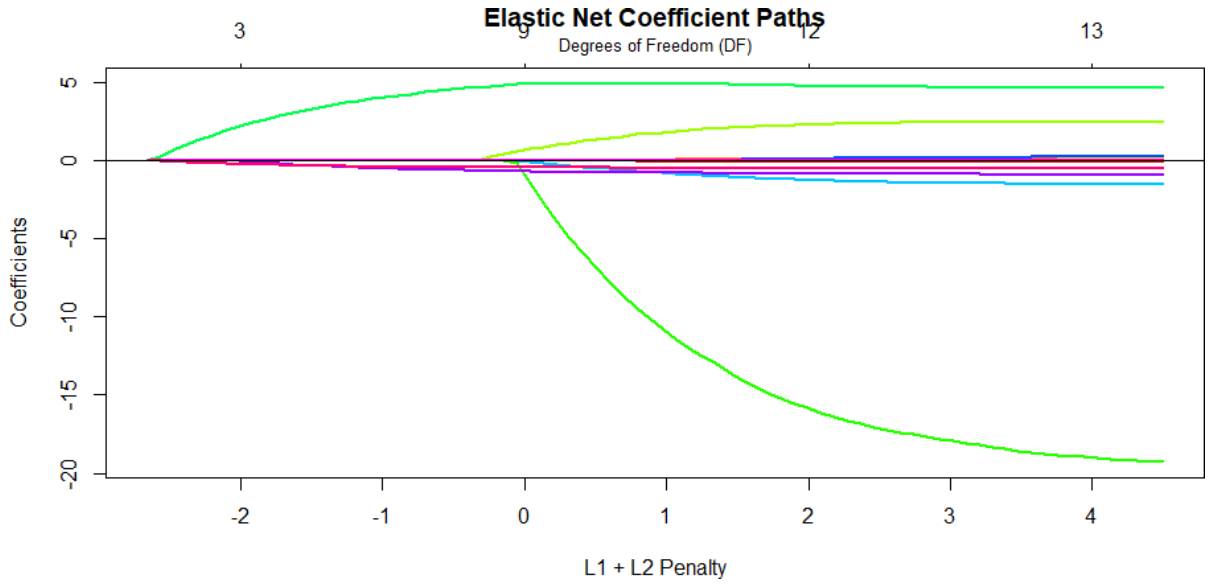

Figure 21

Figure 21 illustrates the coefficient paths of the elastic net regression model applied to the Boston housing dataset.

CHAPTER 2

Tree-Based Models

2.1 Introduction to tree-based methods

In this chapter, we describe tree-based models. These models allow us to make decisions and generate predictions regarding our data in a simple and intuitive manner. In contrast to other statistical methods, tree-based models could not have been developed without the use of computers, as their implementation requires substantial computational power.

The need for such methods arose primarily from social scientists who are often required to address various problems related to the data structure, similarly to the development of other techniques such as factor analysis. Although early versions of these methods existed as early as the 1960s, the formal development of the CART (Classification and Regression Trees) methodology was introduced in 1984 by Breiman et al. The CART algorithm was a milestone in the development of machine learning and tree-based modeling.

Tree-based models are divided into two main categories: classification trees and regression trees. Both approaches describe a mechanism through which the observed data lead to a specific predictive outcome, regardless of whether the task concerns regression or classification into categories. These models are highly interpretable, as they are based on discrete steps, with each step defined by a simple decision rule. Furthermore, they do not require strict distributional assumptions about the data, which makes their implementation particularly attractive.

The structure of a tree-based model is well-defined and remains consistent regardless of its specific type. The construction process begins with an initial node, the so-called root node, which contains all available observations. The root node is subsequently partitioned into two child nodes according to a decision rule based on one of the available explanatory variables. In each node the algorithm selects the split that maximizes an information criterion. This recursive splitting procedure continues until a predefined stopping criterion is satisfied. The resulting final nodes,

commonly referred to as terminal nodes or leaves, correspond to the predictive outcome of the model.

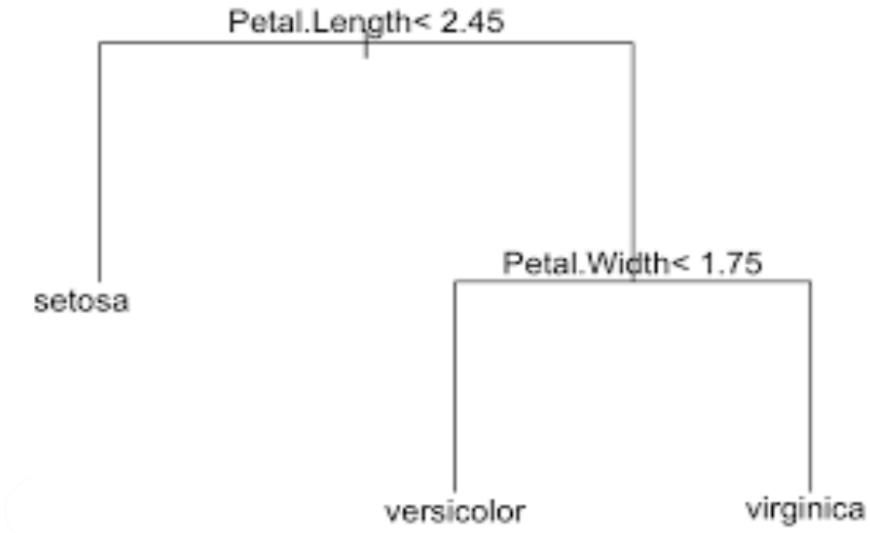

Figure 22: Classification Tree on IRIS dataset [4]

The fundamental idea underlying tree-based methods is the partitioning of the feature space of the original variables into disjoint regions $R_1, R_2, R_3, \dots, R_K$. For every observation that falls within the same region, the model assigns the same estimate of the response variable.

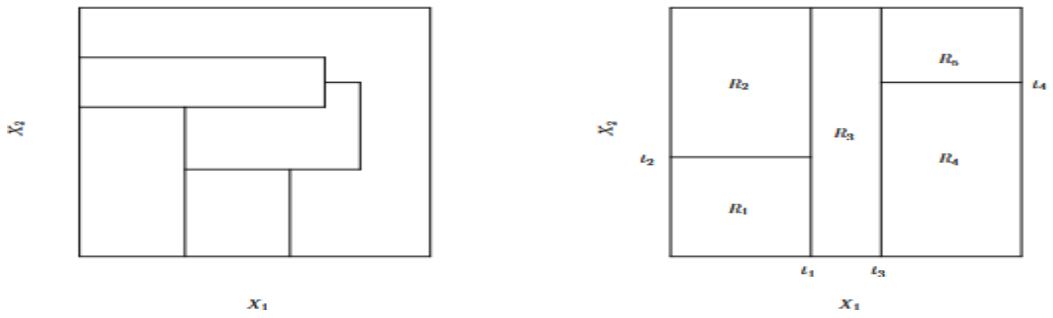

Figure 23: Partitioning of the two-dimensional feature space

[4] A.Karagrigoriou , Διαλέξεις μαθήματος Μηχανικής Μάθησης [Machine Learning course lectures], MSc in Applied Statistics, University of Piraeus

Despite their simplicity and interpretability, single tree models often exhibit lower predictive accuracy compared to other regression or classification methods. However, ensemble approaches such as random forests combine a large number of trees and achieve significantly improved predictive performance.

2.2 Regression Tree Models

In this thesis, we will primarily focus on regression trees. Regression trees are tree-based models that are used for the prediction of a dependent variable that is numerical. As already mentioned, they are highly interpretable and do not require strong assumptions about the data, such as the existence of a linear relationship.

The structure of a regression tree is similar to the general tree structure described above. However, there are certain differences. First, each terminal node is associated with a specific value, which represents the prediction of the dependent variable. Specifically, the prediction assigned to each terminal node is equal to the mean response observations that in that region.

The criterion used for partitioning the initial feature space into subregions is the minimization of the residual sum of squares (RSS)

$$RSS = \sum_{k=1}^{K} \sum_{i \in R_k} (y_i - \hat{y}_{R_k})^2$$

Where $\hat{y}_{R_k}$ notes the estimate of the response variable for the observations that belong to region R_k , $k = 1,2,..K$. For this estimation, the mean of the response variable is calculated using the training data.

It is infeasible to consider all possible subregions that could be formed within the feature space. For this reason, a top-down approach is adopted, known as recursive binary splitting, in which the process begins at the top of the tree, where the feature space contains all observations, and proceeds through successive partitions of the explanatory variable space. Each split is represented by two new branches that extend further down the structure of the tree.

This approach is characterized as greedy because, at each stage, the best possible split is selected for that particular step without taking into account the possibility that an alternative split might lead to a better overall tree at a subsequent stage.

The partitioning is performed by selecting one variable X_i from the set of explanatory variables and a cutpoint s. The feature space is then divided into the two regions $\{ X_i < s \}$and $\{ X_i \geq s \}$. All explanatory variables $X_i's$ and all possible cutpoints s are examined, and the pair $(X_{i'}, s)$that yields the smallest possible RSS is selected.

Specifically let:

$$R_1(i,s) = \{X|X_i > s\} \,\&\, R_2(i,s) = \{X|X_i \leq s\}$$

and look for the values of i and s that minimize

$$\sum_{xi\,\epsilon R_1} (y_i - \hat{y}_{R1})^2 + \sum_{xi\epsilon R_2} (y_i - \hat{y}_{R2})^2$$

where $\hat{y}_{R_1}$is the mean response of the training observations that belong to region $R_1(i,s)$, and $\hat{y}_{R_2}$ is the mean response of the training observations that belong to region $R_2(i, s)$. When the number of explanatory variables is not very large, the determination of the optimal i and s can be carried out relatively quickly.

The procedure is then repeated by searching for the predictor and the new cutpoint that further minimizes the RSS within the newly formed regions. However, instead of partitioning the original feature space again, one of the previously created regions is split into two subregions. As a result, four regions are obtained, and the algorithm evaluates whether further partitioning is possible. The process continues until a stopping criterion is satisfied; for example, it may terminate when no region contains more than five observations.

Once the regions $R_1, \ldots, R_K$ have been constructed, the prediction for a new observation (test observation) is given by the mean response of the training observations that belong to the region in which the new observation falls.

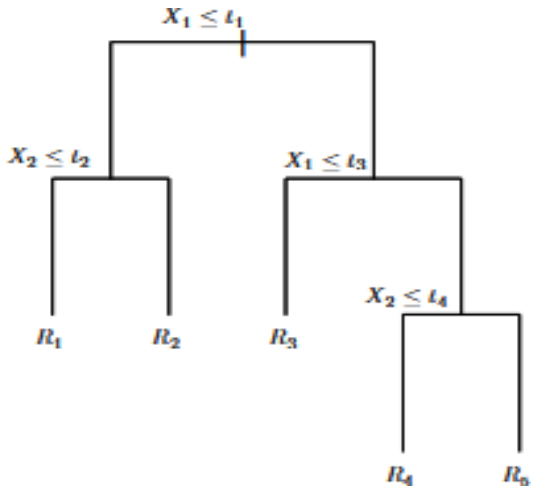

Figure 24: Regression tree model structure

Although this procedure may produce satisfactory results, a common limitation of tree-based methods is their tendency to overfit the training data.

2.3 Tree versus Linear Models

Regression Trees are very different from the classical approaches of linear regression. Linear regression assumes a model of the form

$$Y = \beta_0 + \sum_{i=1}^{p} X_i \beta_i$$

where β_i are the classical OLS estimators. On the other hand, regression trees assume a model of the form

$$Y = \sum_{m=1}^{M} c_m 1_{X \in R_m}$$

where R_m is the partion of the future space and c_m is the mean of y_i in R_m region.

The selection of the model depends on the data. If the relationship between variables is linear then a linear model will outperform a regression tree model that doesn't take into concern the linear structure.

On the other hand, if data structure is nonlinear and high complexity tree model may perform better than a linear model .

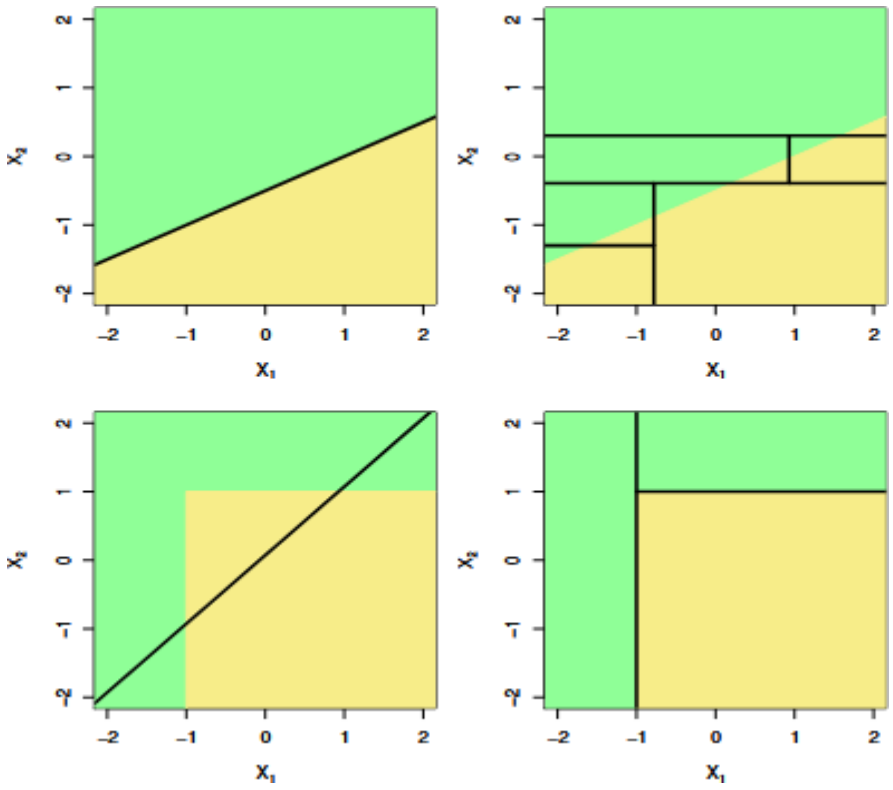

Figure 25: Linear vs nonlinear structure

In Figure 25, we observe different configurations of a two-dimensional predictor space based in two variables X_1,X_2 .In the first case and the second case (top left and top right), there is a clear linear relationship between the two variables, as the decision boundary is represented by a straight line .Therefore, we expect the classical linear regression model to outperform a regression tree, since the tree model does not explicitly exploit the linear structure of the data.

In the remaining cases, however, the decision boundaries are not linear. Instead, we observe rectangular regions, suggesting that the true relationship between the variables is nonlinear. In such situations, a linear model will not perform well.

By contrast, a regression tree, through recursive partitioning of the predictor space, can adapt to complex and nonlinear patterns. Consequently, when the underlying relationship is nonlinear or

exhibits local structural variations, we expect the tree-based model to achieve superior predictive performance compared to the classical linear model.

Regression trees have a number of advantages over the classic linear model

1) Trees are very easy to explain to people. In fact, they are even to explain than linear regression.
2) Trees can be displayed graphically and are easily interpreted even by non-experts.
3) Trees exhibit robustness to Outliers.
4) Trees can easily handle qualitive predictors without the use of dummy variables.
5) Variable Selection only high correlated predictors appear to tree structure. Predictors with stronger correlation with the response variable tend to appear in the first levels of the tree.

However, regression trees also suffer from important disadvantages:

1) Regression trees are unstable: Small changes in the data may lead to different tree structures which reflects their high variance nature.
2) Lower predictive accuracy: A single regression tree often does not achieve the same predictive accuracy as more structured models.

2.4 Overfitting

Overfitting refers to the situation in which a statistical model is overadapting in the training dataset, capturing not only the relationship of the data but also random noise. Although such a model may achieve very low training error, its predictive accuracy on new observations is often low so the model is unable to reach a good predictive performance .This situation is closely connected to the bias–variance trade-off, since models with increased flexibility typically exhibit reduced bias but substantially higher variances .On other hand, underfitting occurs when the model is very simple and is unable to capture the relationship on the dataset leading to poor performance.

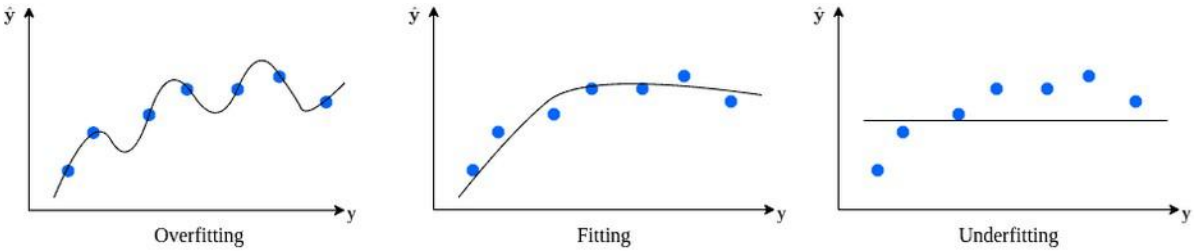

Figure 26: Overfitting vs Underfitting

Regression trees tend to overfit due to their binary splitting procedure, which greedily selects splits that minimize training error at each step. When allowed to grow without restrictions, trees may become excessively deep, generating many small terminal nodes that reflect irregularities of the dataset rather than true relationships. Consequently, without controlling the size of the tree with mechanisms such as pruning, fully grown trees tend to display high variance and limited generalization ability.

2.5 Pruning

The size of the tree must be investigated, since large trees have greater complexity and tend to overfit the data. RSS is not a good measure for selecting a subtree because it favors bigger trees.

However, it is impossible to know in advance the appropriate tree size. Therefore, our strategy is to allow the tree to grow until a certain criterion is satisfied, usually when no significant contribution is offered in the reduction of the RSS and then proceed to the pruning process.

Through pruning, our goal is to find the tree with the smallest possible size so that we gain simplicity, while at the same time achieving the smallest possible error in order to gain accuracy.

This procedure is carried out using a method that employs a cost function so as to "penalize" large trees. This procedure is known as Cost Complexity Pruning.

Let $|T|$ denote the number of terminal nodes in each tree, and let

$$N = \#\{x_i \in R_k\}.$$

Then, the cost C_a function is defined as: $C_a(T) = \sum_{i=1}^{|T|} \sum_{i \in R_m} (y_i - \hat{y}_{R_m})^2 + a|T|$.

The parameter $\alpha \geq 0$ expresses how strict the penalty will be for the size of the tree represented by |T|. In case of $\alpha = 0$ this essentially means that no penalty is imposed, whereas as α increases, the size of the tree tends to become smaller as α approaches infinity. For each value of α, it can be shown that there exists a unique subtree that minimizes $C_\alpha(T)$.

To determine the ideal T_α, we use the weakest link pruning algorithm. The main idea of this procedure is described in the following algorithm.

Let T_0 be the full tree and k=0.

Step 1: In Tree $T^{(K)}$ for each internal node we calculate $a(t) = \frac{R(t)-R(T_t)}{|T_t|-1}$ so a(t) is the increase in the error when we remove the subtree T_t with root at node t.

Step 2: Find the weakest link. We are looking for the node t that minimizes t*

$$\boldsymbol{t *= min(a(t))} \text{ and } a_{k+1} = a(t *).$$

Step 3: Create a new tree removing the subtree which star from node t*.

Step 4: Repeat Steps 1-3 until a single root tree remains.

The above algorithm returns two sequences. The first is a decreasing sequence of subtrees $T^{full} \supset T^{full-1} \supset T^{full-2} \supset \ldots \supset t$ while the second is an increasing sequence of penalty parameters α, with $\alpha_1 = 0$.

It can be shown that there exists a one-to-one correspondence between the sequence of α values and the corresponding sequence of subtrees.

In the initial stages, the algorithm tends to prune subtrees that contain a large number of internal nodes. As the tree becomes progressively smaller, the number of nodes removed at each step decreases.

Tree	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}
$\lvert\widetilde{T}_k\rvert$	71	63	58	40	34	19	10	9	7	6	5	2	1

Table 9: Number of Terminal Nodes in Each Pruned Tree

The algorithm does not return a single tree, but rather a sequence of subtrees. Through the cross-validation procedure, we can then select the optimal tree.

More specifically, for each fold of the training set, the tree is first allowed to grow to its full size (a full tree), and the pruning algorithm described above is applied to generate a sequence of values of the penalization parameter $a$along with the corresponding subtrees.

For each value a in this sequence, we compute the prediction error on the validation fold. Then, we calculate the mean cross-validation error (CV error) across all folds and select the value of a that minimizes it. Formally,

$$\hat{a} = \arg\min_{a} CV(a).$$

The final selected tree is the subtree corresponding to the value $\hat{a}$.In practice, this entire procedure is implemented through statistical packages, which automatically perform both the pruning algorithm and the cross-validation process in order to return the final selected tree.

2.6 Applications

In this section, we apply tree-based models to both datasets considered in this Thesis, in order to investigate the behavior of the pruning algorithm, for understanding how pruning affects model structure and prediction accuracy. We begin with the *Airquality* dataset, with the aim of predicting ozone levels. The initial full tree has 10 nodes (see Figure 27).

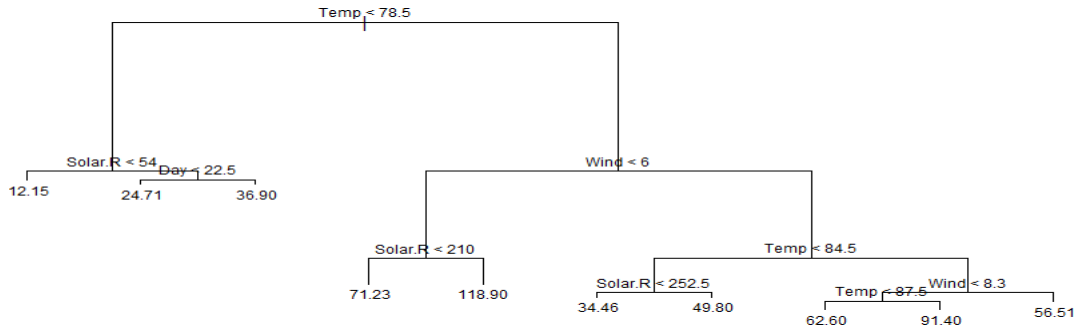

Figure 27: Full tree on Airquality dataset

We first examine the order in which variables appear in the tree. For each independent variable, we fitted a simple linear regression model with *Ozone* as the response variable and the corresponding variable as the sole predictor. We then computed the coefficient of determination (R^2) for each model. We observed that the tree tends to split first on variables associated with higher R^2 values. This is consistent with the tree-building procedure, as variables that explain a larger proportion of the variability in the response lead to greater reductions in the residual sum of squares (RSS) and are therefore selected at earlier splits.

We then proceed with the pruning procedure. In the following diagram, we observe how the residual sum of squares (RSS) changes as a function of the tree size.

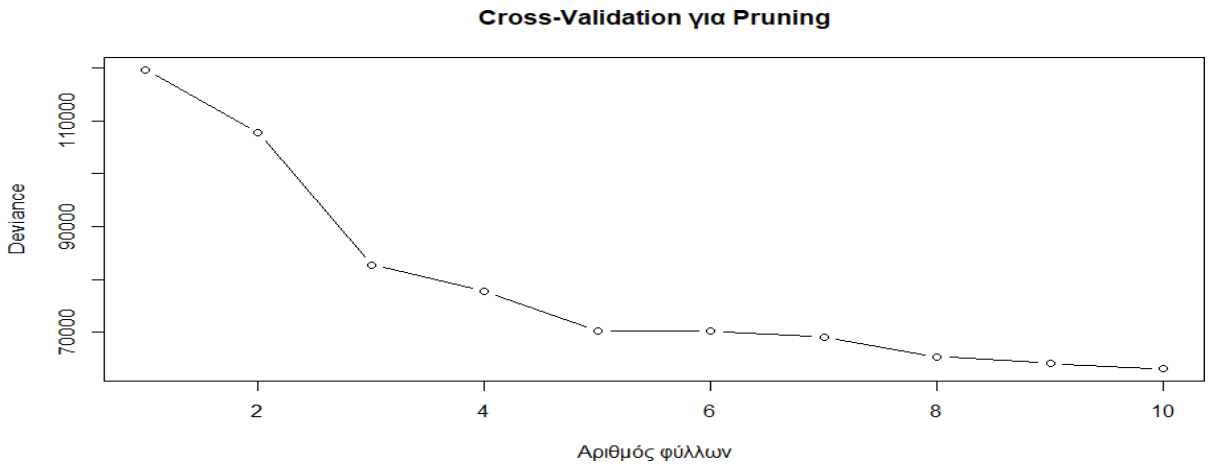

Figure 28: Tree Size vs RSS

From the cross-validation plot (Figure 28), we observe that the RSS decreases rapidly up to around 5 terminal nodes and then stabilizes, with only marginal/minimal improvements thereafter. Therefore, we conclude that selecting a tree with 5 leaves provides a good balance between model complexity and predictive performance.

Figure 29 presents the pruned regression tree. Despite its reduced complexity, the increase in RSS compared to the full tree is considered to be minimal (approximately 20 units), indicating that pruning leads to a simpler model without a significant loss/cost in predictive performance.

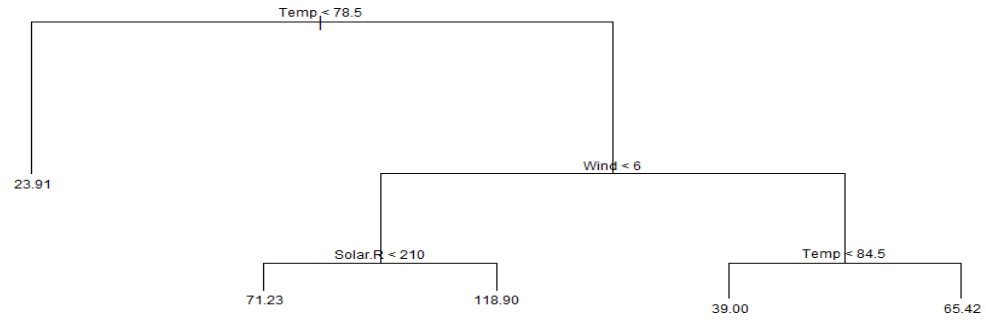

Figure 29: Pruned Tree on Airquality dataset

We have also applied the regression tree model to the Boston Housing dataset with a view to predict the median housing value. The full tree consists of 10 terminal nodes (see Figure 30).

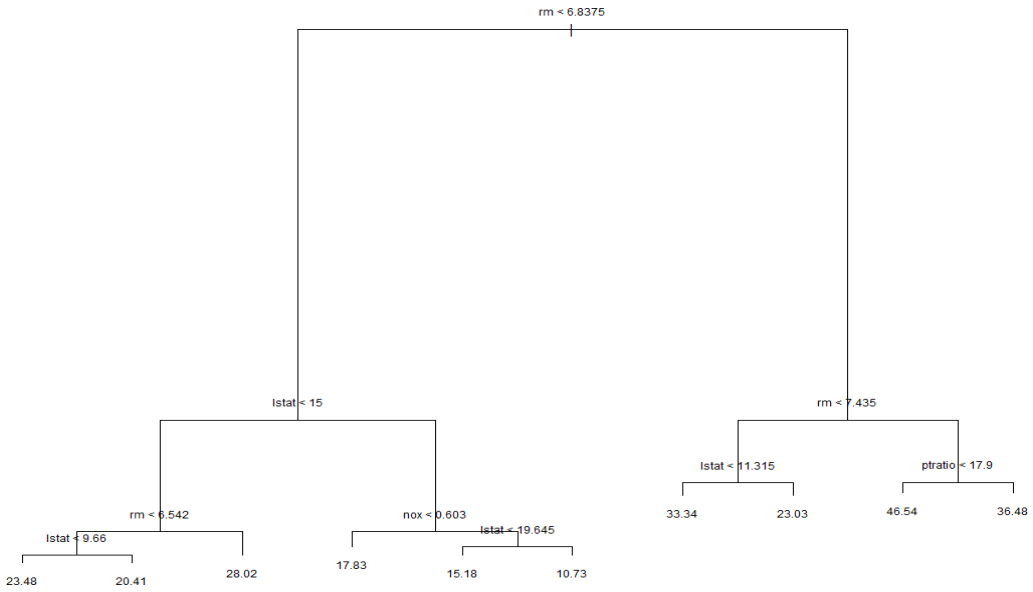

Figure 30: Full tree on Boston Housing dataset

We proceed with the pruning procedure. In the following diagram, we observe how the residual sum of squares (RSS) changes as a function of the tree size.

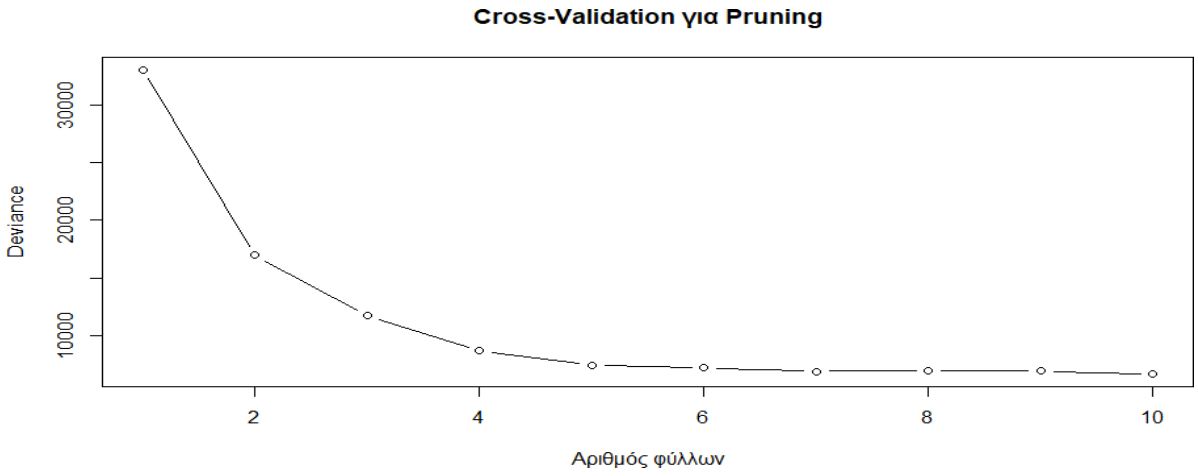

Figure 31: Tree Size vs RSS

More specifically, the cross-validation curve in Figure 31 shows that the deviance decreases sharply as the number of terminal nodes increases from 1 to around 4–5, indicating a significant improvement in model fit. Beyond this point, the reduction in deviance becomes marginal/minimal, suggesting diminishing returns from adding more complexity. This implies that a smaller tree with 5 or 6 terminal nodes achieves a good balance between model accuracy and complexity, while larger trees do not provide substantial additional predictive benefit.

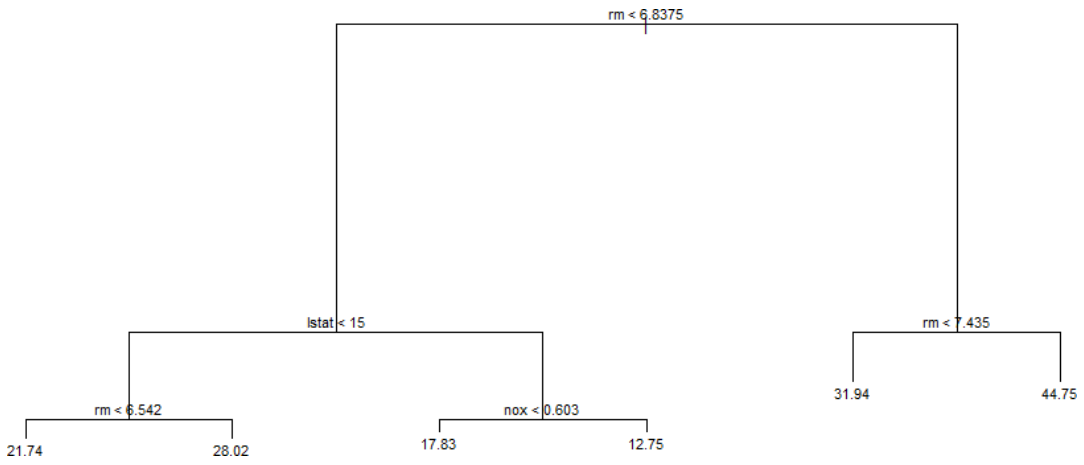

Figure 32: Pruned tree on Boston dataset

Figure 32 presents the pruned regression tree. Despite its reduced complexity, the increase in RSS compared to the full tree is relatively small (approximately 9 units), indicating that pruning is more interpretable model (as compared to the original tree) without a great loss/cost in predictive performance.

CHAPTER 3

Advanced Techniques

In this chapter, we propose two modified penalization functions for regression trees in order to examine how the structure of the penalty and the number of variables involved influence the final tree size and overall model complexity.

Furthermore, we introduce more advanced machine learning techniques, random forests, which extend the idea of tree-based methods by combining multiple trees to improve predictive performance and reduce variance.

Finally, we conduct a comprehensive comparative analysis of all methods presented in the previous chapters, evaluating their predictive performance on the considered datasets. The results aim to highlight the strengths and limitations of each approach, as well as to assess whether the proposed penalization schemes can lead to improved model selection and generalization ability.

3.1 Advanced Penalized Techniques

The classical penalization scheme in regression trees is linear with respect to the size of the tree. In this section, we extend this framework by introducing an exponent on the tree size, aiming to examine how such a modification affects both the behavior of the algorithm and the resulting tree complexity. In particular, we investigate the impact of transforming the penalty function from a linear to a polynomial form.

To this end, we redefine the cost function by incorporating an exponent b on the tree size.

$$C(T) = RSS + \alpha \mid T \mid^{b}$$

Figure 33 illustrates the behavior of the proposed penalization scheme as a function of the exponent b for different tree sizes. As b increases, the cost associated with larger trees increases

at a significantly faster rate compared to smaller trees. This results in a stronger penalization of complex models, effectively favoring simpler tree structures for higher values of b.

In contrast, for lower values of b, the differences in cost across tree sizes are less pronounced, allowing larger trees to remain competitive. This emphasizes the role of the exponent as a tuning mechanism that controls the trade-off between model complexity and goodness-of-fit.

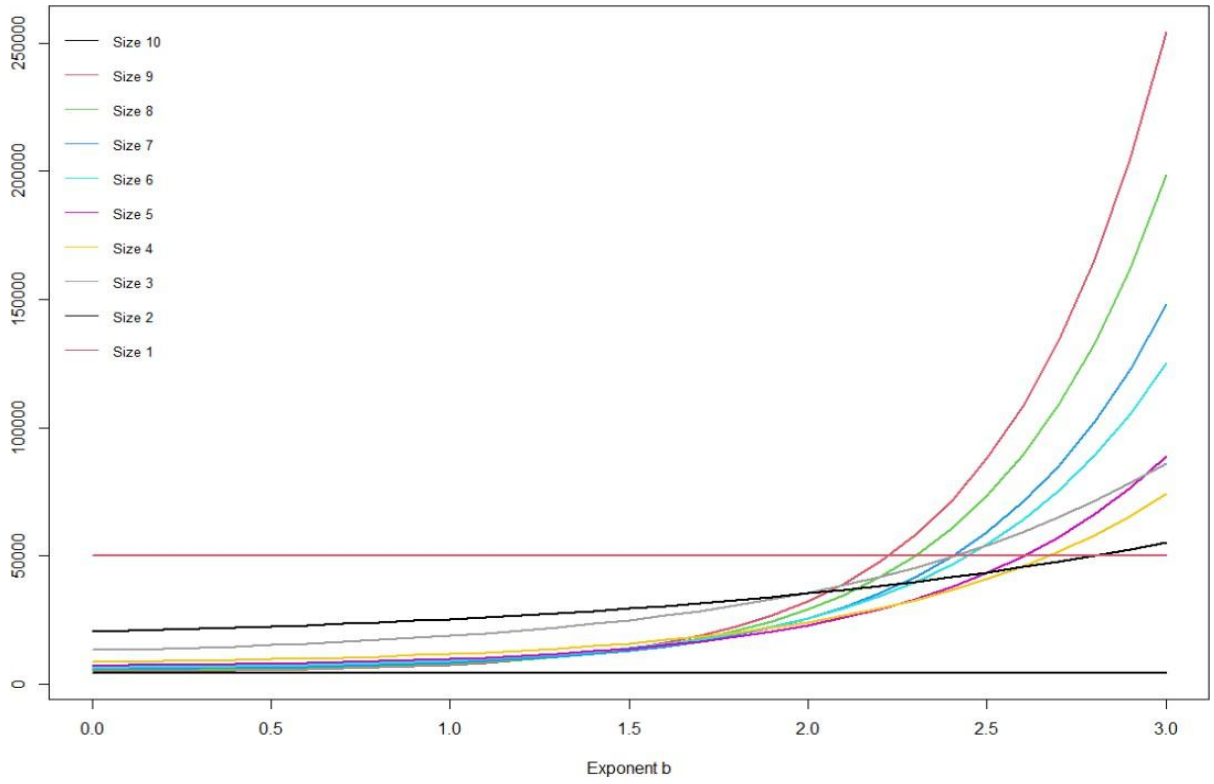

Figure 33: Cost vs Tree Size

The second modified cost function presented in this section aims to combine the classical penalization framework used in regression trees with the regularization approach of the elastic net in linear models. The main objective of this modification is to incorporate information regarding the number of variables utilized by the tree into the penalization process. Specifically, this approach seeks to account not only for the size of the tree, but also for its structural complexity in terms of the predictors involved. We introduce a new linear penalty function, defined as follows:

$$C(T) = RSS/\mathrm{n} + \mathrm{c}(\alpha_{EN} \mid T \mid + \lambda \mathrm{V}(T))$$

The constant c, as well as the normalization of the RSS by the number of observations, are introduced for scaling purposes. V(T) is the number of variables participating in the tree. The parameters α_{EN} and λ are selected via cross-validation when applying the elastic net to the full set of predictors, allowing the procedure to account for both the correlation structure and the relative importance of the variables.

Tables 10 and 11 report the values of the cost function across the considered datasets.

	c	Size	RSS	Penalty	Cost
1	0.01	10	4300.4	0.09	11.81
2	0.03	10	4300.4	0.29	12.00
3	0.10	10	4300.4	0.91	12.62
4	0.32	10	4300.4	2.87	14.59
5	1.00	10	4300.4	9.07	20.79
6	3.16	5	6645.4	14.34	32.45
7	10.00	4	7684.7	38.34	59.28
8	31.62	2	15473.4	60.62	102.78
9	100.00	1	32888.8	70.00	159.62
10	316.23	1	32888.8	221.36	310.97
11	1000.00	1	32888.8	700.00	789.62

Table 10: Cost Function (Boston Housing dataset)

	c	Size	RSS	Penalty	Cost
1	0.01	10	22885.4	0.11	200.86
2	0.03	10	22885.4	0.34	201.09
3	0.10	10	22885.4	1.08	201.83
4	0.32	10	22885.4	3.43	204.18
5	1.00	10	22885.4	10.84	211.59
6	3.16	10	22885.4	34.28	235.03
7	10.00	9	23991.4	93.30	303.75
8	31.62	5	32160.4	193.85	475.96
9	100.00	3	47445.7	382.00	798.19
10	316.23	1	107290.8	252.98	1194.13
11	1000.00	1	107290.8	800.00	1741.15

Table 11 Cost Function (Airquality dataset)

Table 10 corresponds to the Boston Housing dataset, while table 11 refers to the Airquality (Ozone) dataset. The parameter c is introduced as a scaling parameter in order to account for the substantial difference in magnitude between the RSS and the penalty term, ensuring that both components contribute meaningfully to the overall cost function.

More specifically, in the Boston Housing dataset, when c approaches values around 3, the influence of the penalty term becomes noticeable, leading to a deviation from the purely RSS-driven selection of large trees. In contrast, for the Airquality dataset, the value of c must be significantly larger with values around 30 for the penalty to present an effect. This difference can be attributed to the much larger scale of the RSS values in this dataset.

This observation highlights that the RSS term drives the behavior of the cost function. As a result, higher RSS values require stronger penalization aiming to balance the trade-off between good fit and model complexity.

Importantly, once the penalty term becomes sufficiently influential, the selected tree sizes begin to align with those obtained via the classical cost-complexity pruning algorithm, indicating consistency with established pruning methodologies.

Based on this analysis, we select a tree of size 5 for both the Boston Housing and the Airquality datasets, as these correspond to the point where the penalty effectively balances model fit and complexity without leading to over-penalization.

For the Boston Housing dataset, we examine the behavior of the proposed penalty under two limiting cases: when $\alpha_{EN} \to 0$, where Elastic Net approximates Ridge regression, and when $\alpha_{EN} \to 1$, where it approaches LASSO.

We begin with the case $\alpha_{EN} \to 0$. Keeping the parameter $c = 3.16$ fixed, we observe from the corresponding plot that as α_{EN} decreases (from values around 0.7 towards 0), the imposed penalty gradually diminishes and eventually becomes zero. As a result, the model converges to the full tree, with no regularization applied.

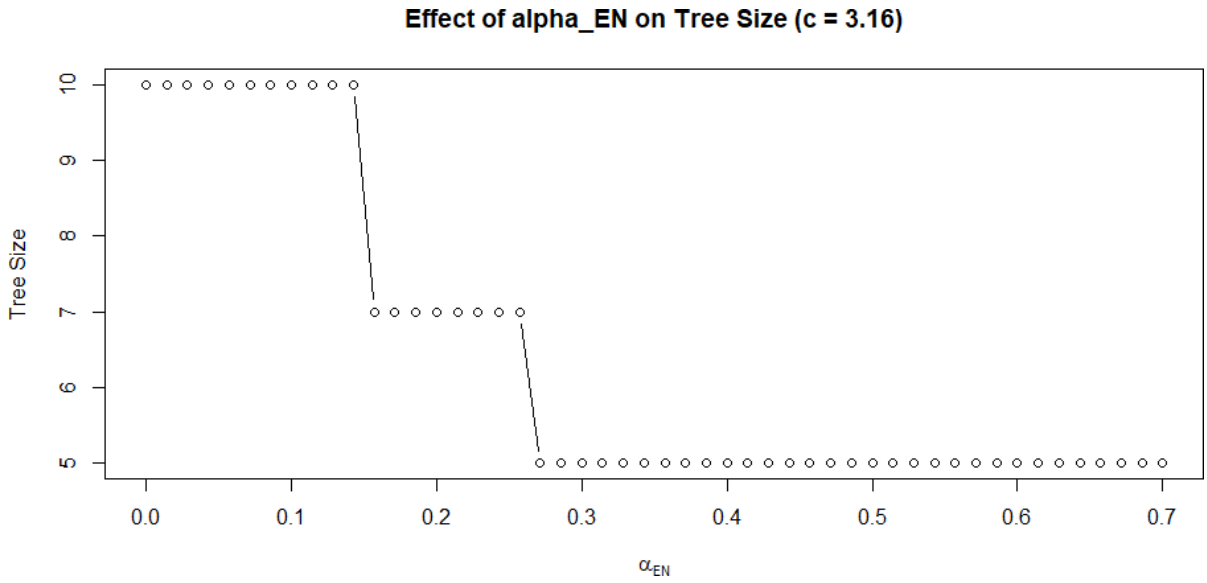

Figure 34: Tree size vs alpha hyperparameter

Next, we consider the case $\alpha_{EN} \to 1$. In this region, as α_{EN}increases, the size of the tree decreases, indicating a stronger penalization effect. In particular, a noticeable transition point appears around $\alpha_{EN} \approx 0.9$, beyond which the tree becomes simpler.

These findings suggest that the parameter α_{EN} plays a crucial role in controlling tree complexity, acting as a bridge between Ridge-type and LASSO-type regularization.

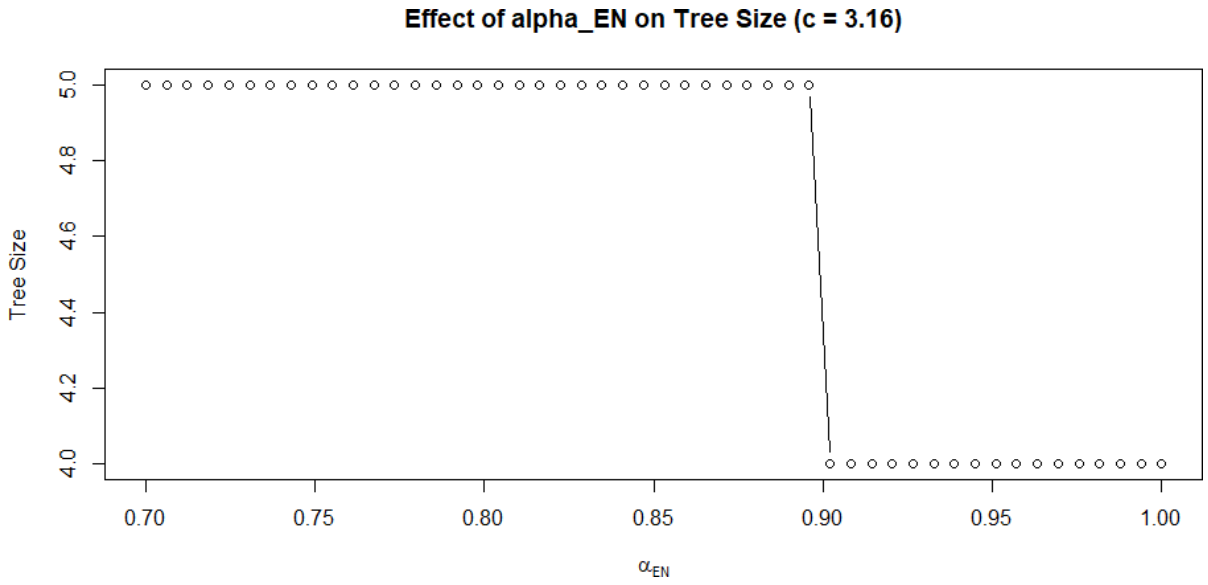

Figure 35: Tree size vs alpha hyperparameter

3.2 Random Forests

Random Forests constitute one of the most widely used machine learning algorithms. They were introduced in 2001 by Leo Breiman and Adele Cutler. They belong to the class of ensemble methods, as they are composed of multiple decision trees and combine the predictions of individual trees in order to improve overall performance. They can be applied to both regression and classification problems.

The Random Forest algorithm can be seen as an extension of the bagging method, which was also introduced by Breiman. The main idea is to construct a significant number of trees, typically ranging from 200 to 1000. For each tree, a bootstrap sample is drawn from the training dataset. It is worth noting that approximately one-third of the observations are not included in this process.

In addition, for each split in a tree, a random subset of predictor variables is considered. These two sources of randomness—bootstrap sampling and random feature selection—ensure that the

trees in the forest are less correlated with each other, which is crucial for improving predictive performance.

Depending on whether the problem is classification or regression, the final prediction of the forest is produced differently. In classification problems, the output is determined by majority voting among the trees, while in regression problems, it is given by the average of the individual tree predictions.

The observations that are not included in the bootstrap sample (out-of-bag observations) are used to estimate the generalization error of the model, providing an internal validation mechanism without the need for a separate test set.

A key advantage of Random Forests is that they significantly reduce overfitting and are robust due to the randomness introduced in their construction. As the number of trees increases, the generalization error tends to stabilize, which aligns with the intuition behind the law of large numbers.

Furthermore, the algorithm provides a measure of variable importance. Variables that lead to a significant increase in the out-of-bag error when permuted are considered important, offering an additional tool for feature selection.

A key disadvantage of Random Forests is that they are less interpretable than simpler models. Although they provide strong predictive performance, it is harder to understand how each variable affects the final prediction

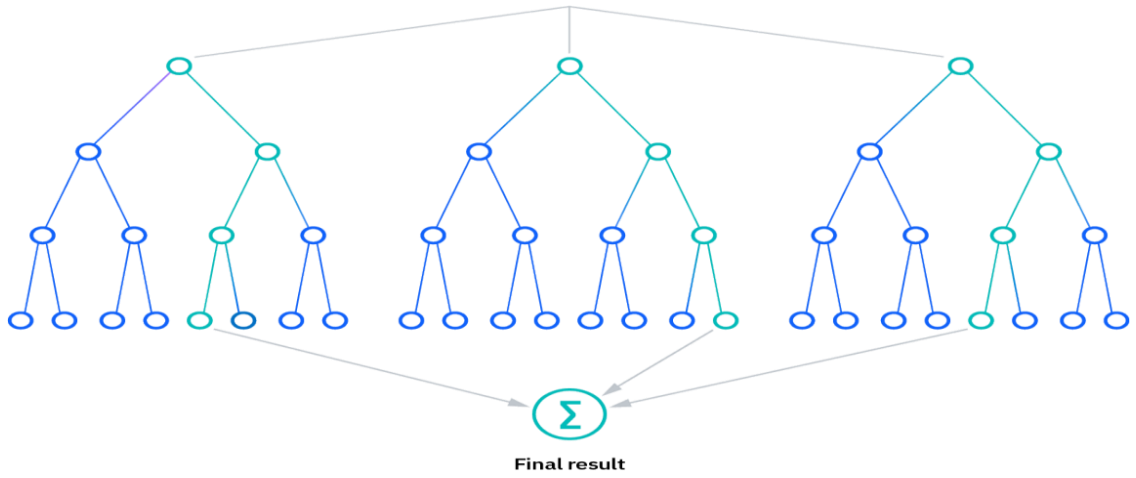

Figure 36: Schematic representation of a Random Forest model

We present a set of plots obtained from applying the Random Forest algorithm to the two datasets, starting in Figure 37 with the Boston dataset in the left plot and the Airquality dataset following in the right plot. Figure 37 illustrates the importance of each variable by showing the percentage increase in MSE when the values of a variable are randomly shuffled and the plot presents how each variable contributes to tree purity. This provides a measure of the importance of each predictor. It can also be used as a tool for variable selection in other models.

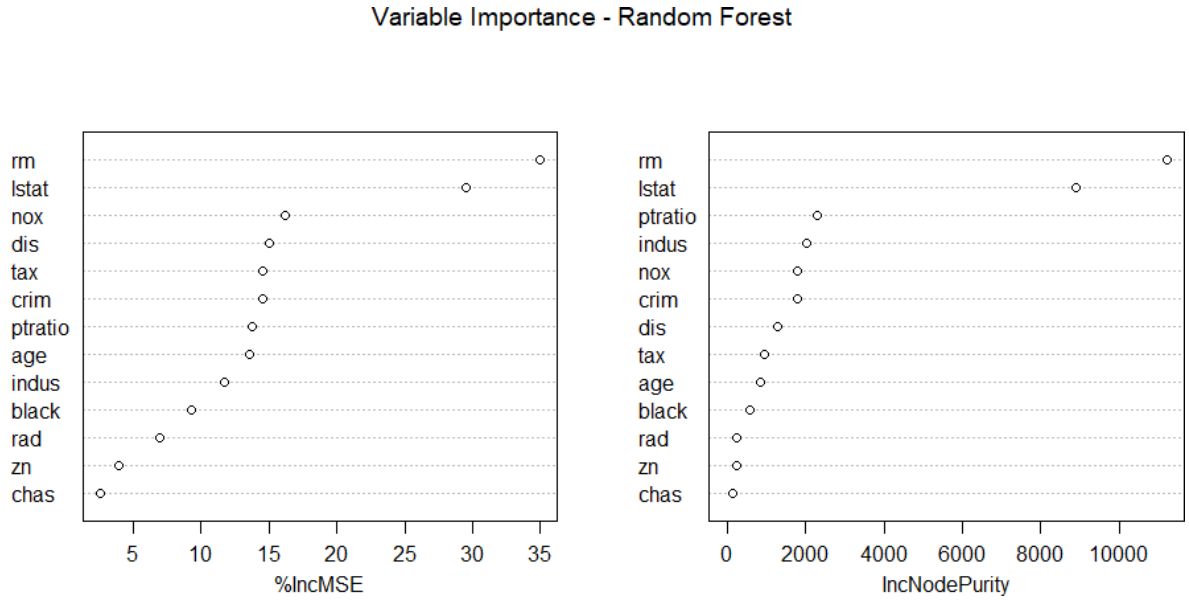

Figure 37: Variance Importance Plot on Boston Dataset

As can be observed from the above plots for the Boston Housing dataset, the variables rm, lstat and nox are the most influential predictors, while the remaining variables have a smaller contribution. Overall, the results suggest that a few key variables play a dominant role in determining the predictive performance of the model.

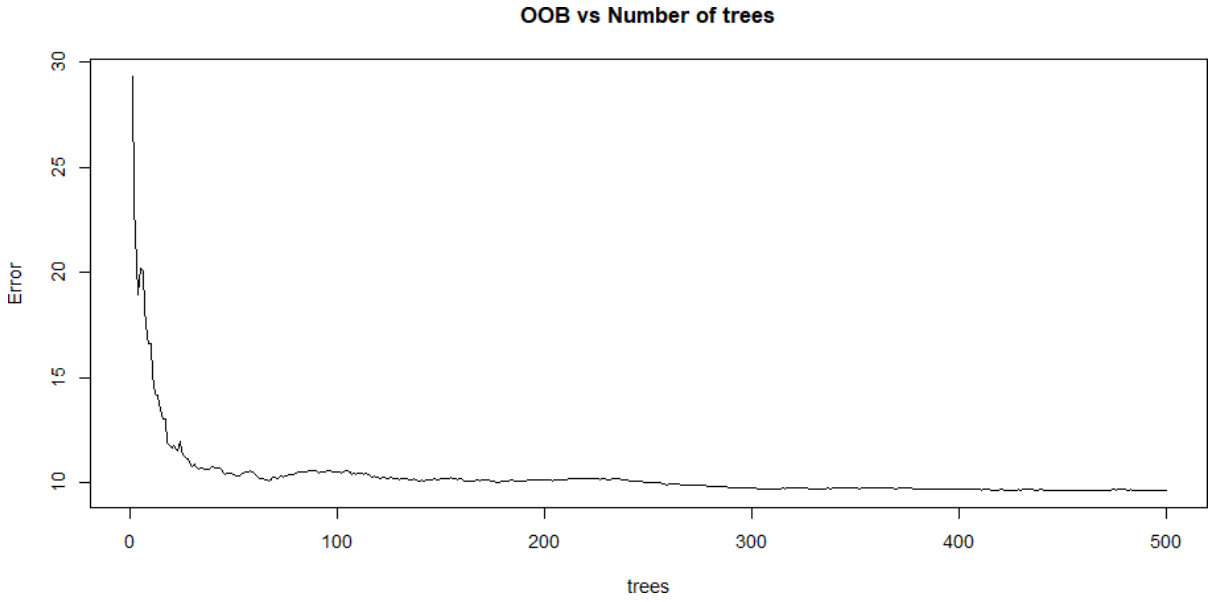

Figure 38: Out of Bag error vs Number of trees

Figure 38 shows how the out-of-bag (OOB) error changes as the number of trees increases. The error drops quickly at the beginning, indicating that adding more trees improves the model. After a certain point around 100 trees, the error stabilizes, suggesting that additional trees do not significantly improve performance.

Next, we apply the Random Forest algorithm to the Airquality dataset.

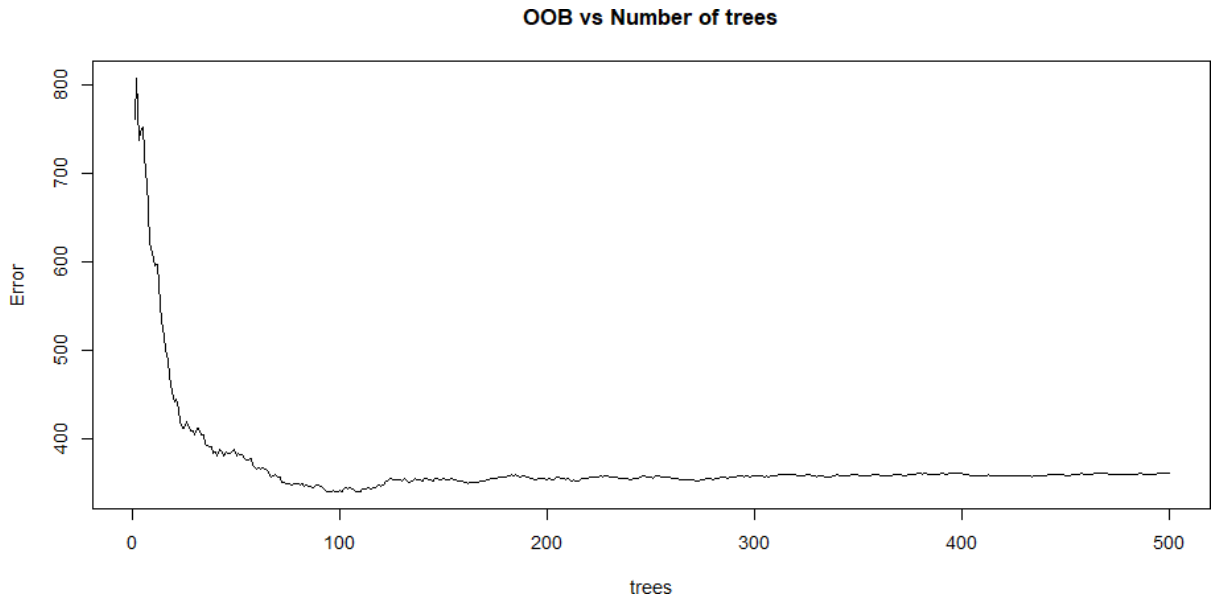

Figure 39: Out of bag error vs Number of trees

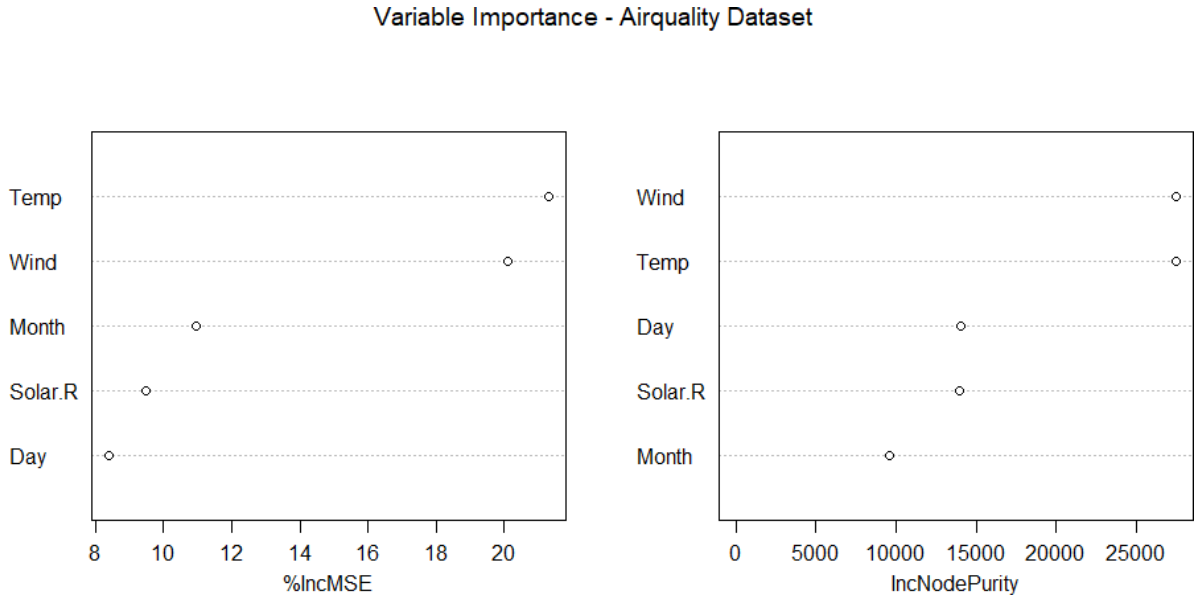

Figure 40: Variance Importance Plot on Airquality Dataset

Figure 40 presents the OOB error versus the number of trees to the second dataset. The OOB error decreases rapidly as the number of trees increases and then stabilizes. The variable importance plots show that Temp and Wind are the most influential predictors, while the remaining variables contribute less to the model. Overall, the results suggest that a small number of key variables drive the predictive performance in this dataset.

3.3 Results

In this section, we proceed with the comparative evaluation of the models described above. We recall that both datasets have been split into training and test sets using a 70–30 ratio.

To assess the predictive performance of the models, we employ the Mean Squared Error (MSE) and the Root Mean Squared Error (RMSE), as these metrics penalize large deviations more heavily and, in the case of RMSE, are expressed in the same units as the response variable.

In what follows, we present how the test error varies as a function of tree size and the number of variables used, for both datasets. We begin with the Boston Housing dataset. From the previous analysis, it was shown that the classical pruning algorithm selects a tree of size 6, using the variables *rm*, *lstat*, *nox*, and *ptratio*.

In contrast, the proposed cost function leads to a simpler tree of size 5, involving only 2 variables (*rm* and *lstat*). It is worth to mention that the predictive performance of the model does not degrade significantly despite the reduction in complexity, suggesting that comparable accuracy can be achieved with a more interpretable model.

Model Complexity vs Performance

	Size	Variables_Number	Variables	RSS	Test_MSE
1	10	4	rm, lstat, nox, ptratio	4,300.44	29.03
2	9	4	rm, lstat, nox, ptratio	4,642.99	30.96
3	8	4	rm, lstat, nox, ptratio	5,021.28	32.30
4	7	3	rm, lstat, nox	5,437.23	32.53
5	6	3	rm, lstat, nox	5,988.19	33.59
6	5	2	rm, lstat	6,645.38	35.34
7	4	2	rm, lstat	7,684.74	37.44
8	3	2	rm, lstat	10,491.64	44.33
9	2	1	rm	15,473.40	57.92
10	1	0	None	32,888.75	71.42

Table 12 Model Complexity vs Performance

Table 12 illustrates the relationship between tree size number of variables, and the corresponding test error. Each point represents a different regression tree, while the labels next to the points indicate the tree size. As the number of variables increases, the test MSE decreases substantially in the initial stages, suggesting that the inclusion of relevant predictors improves predictive performance. Overall, the figure highlights that smaller trees with fewer variables can achieve performance comparable to larger and more complex models, emphasizing the importance of model simplicity.

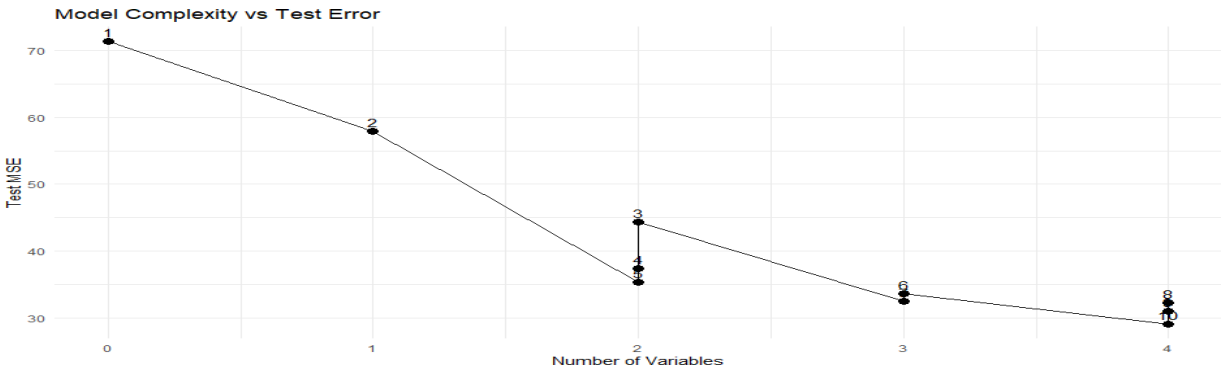

Figure 41: Model Complexity vs Test Error

Table 13 summarizes the predictive performance of the different models on the test set for the Boston Housing dataset. It can be observed that the penalized regression models (Ridge, Lasso, and Elastic Net) exhibit very similar performance, with only minor differences in both MSE and RMSE with Ridge performing slightly better, likely due to the presence of moderate multicollinearity.

Among all models, the Random Forest clearly outperforms the rest, achieving the lowest test error (MSE = 19.316, RMSE = 4.395), which is consistent with its ability to capture complex nonlinear relationships and interactions between variables.

Regarding tree-based methods, the Full Tree achieves the lowest test error among single-tree models. However, both the Pruned Tree and the Custom Pruned Tree result in only a moderate increase in error, while offering reduced complexity and improved interpretability. This further

supports the idea that simpler models can achieve competitive performance without a substantial loss in predictive accuracy.

Model	MSE	RMSE	Number of Variables
Full Tree	29.026	5.388	4
Pruned Tree	33.589	5.796	3
Custom Pruned Tree	35.337	5.944	2
Ridge	33.272	5.768	13
Lasso	33.863	5.819	10
Elastic Net	34.407	5.866	9
Random Forest	19.316	4.396	-

Table 13: Model Comparison Based on Test Set

The results for the second dataset are summarized in Figure 42, which is characterized by a smaller number of variables and a smaller sample size.

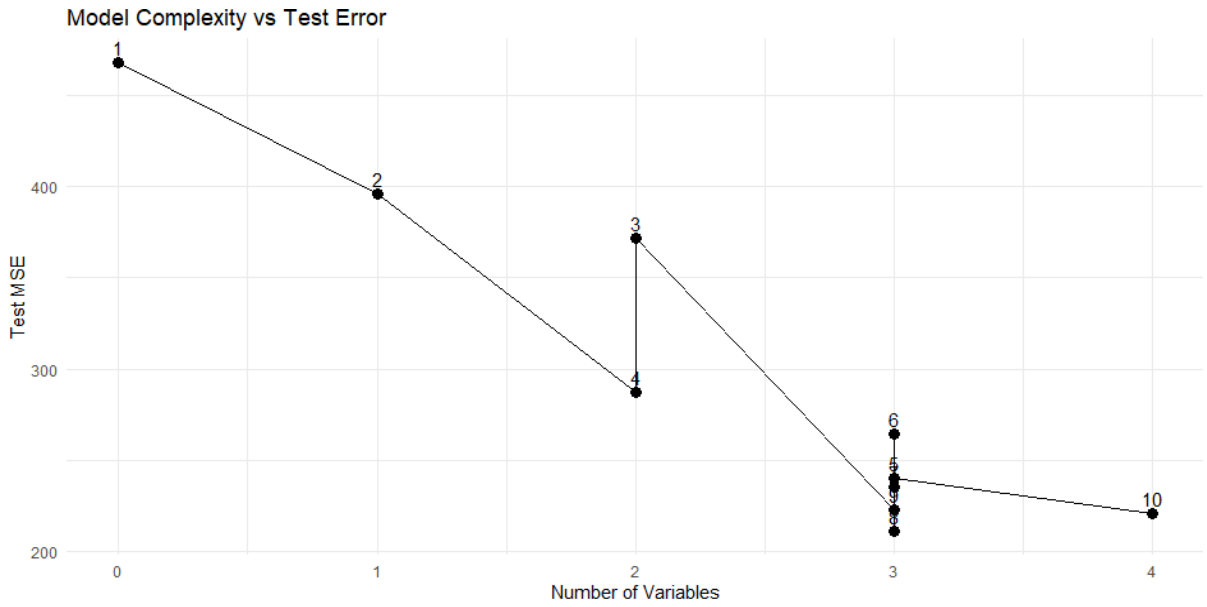

Figure 42 Model Complexity vs Test Error

Model Complexity vs Performance

Size	Variables_Number	Variables	RSS	Test_MSE
10	4	Temp, Solar.R, Day, Wind	22,885.360	220.962
9	3	Temp, Solar.R, Wind	23,991.390	222.662
8	3	Temp, Solar.R, Wind	25,316.720	210.878
7	3	Temp, Solar.R, Wind	27,390.320	235.299
6	3	Temp, Solar.R, Wind	29,763.030	264.207
5	3	Temp, Wind, Solar.R	32,160.440	240.174
4	2	Temp, Wind	38,777.590	287.532
3	2	Temp, Wind	47,445.730	372.088
2	1	Temp	69,610	396.162
1	0	None	107,290.800	468.264

Table 14: Model Complexity vs Performance

Table 14 shows that, due to the limited number of available predictors in the dataset, the set of variables remains largely unchanged across different tree sizes. As a result, the complexity of the model is mainly driven by the size of the tree rather than the number of variables.

Model	MSE	RMSE	Number of Variables
Full Tree	220.962	14.865	5
Pruned Tree	264.206	16.254	3
Custom Pruned Tree	264.206	16.254	3
Ridge	305.573	17.481	5
Lasso	271.865	16.488	3
Elastic Net	265.920	16.307	3
Random Forest	157.920	12.567	-

Table 15: Model Comparison Based on Test Set

In the table above, it can be observed that Elastic Net and Lasso slightly outperform Ridge regression. This can be attributed to the presence of predictors that are weakly or not significantly related to the response variable (Ozone). In such cases Lasso and Elastic Net are able to perform variable selection. On the other hand, Ridge regression retains all variables in the model, which may introduce unnecessary noise. Additionally, the relatively low level of multicollinearity in the dataset reduces the advantage of Ridge, allowing Lasso and Elastic Net to perform better. Additionally, the modified penalty function seems to agree with the classical penalty function regarding both the final tree size and the variables used in the tree.Finally, Random Forest clearly outperforms

all other methods, achieving the lowest test MSE and RMSE, highlighting the advantage of ensemble methods in capturing complex relationships.

3.4 Conclusion

In this project, an attempt was made to combine penalized regression methods with the penalization function used in regression trees. More specifically, the Elastic Net methodology and its parameters were used in a modified cost function, with the aim of accounting not only for the size of the tree, but also for the variables involved in its construction.

This approach enables a more comprehensive control of model complexity, as it combines information related both to the structural characteristics of the tree and to the composition of the predictor set. In this sense, it establishes a bridge between linear penalization techniques and tree-based modeling.

For the Boston Housing dataset, the proposed penalty function produced a regression tree with five terminal nodes and two explanatory variables, compared to the classical pruning algorithm, which resulted in a tree with six terminal nodes and three explanatory variables. Thus, the proposed approach achieved a simpler and more interpretable tree structure, at the cost of only a modest increase in prediction error of approximately 6%.

For the Airquality dataset, the proposed penalty function produced the same tree structure as the classical pruning algorithm.

As a direction for future research, particular interest lies in the development of automated procedures for selecting the penalty parameters, for example through cross-validation techniques. Additionally, the adoption of prior information regarding the importance of variables could further enhance the model, allowing certain predictors to be favored during the tree construction process.

Moreover, it is important to evaluate the proposed methodology using simulated datasets with varying levels of noise, in order to systematically assess its robustness and to examine whether the embedded variable selection mechanism effectively contributes to improved predictive performance. Overall, this work highlights the potential of hybrid penalization frameworks as a

promising direction for enhancing both the interpretability and the predictive capability of tree-based models.

REFERENCES

Ξενόγλωσση Βιβλιογραφία

i. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

ii. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth.

iii. Emmert-Streib, F., Moutari, S., & Dehmer, M. (2016). *Elements of data science, machine learning, and artificial intelligence using R*. Springer.

iv. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55–67.

v. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. Springer.

vi. Ma, X. (2018). *Using classification and regression trees: A practical primer*. Chapman and Hall/CRC.

vii. Speybroeck, N. (2012). Classification and regression trees. *International Journal of Public Health, 57*(1), 243–246.

viii. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288.
7

ix. Zou, H., & 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.

Ελληνική Βιβλιογραφία

i. Διαμαντάρας, Κ., & Μπότσης, Δ. (2019). *Μηχανική μάθηση*. Εκδόσεις Κλειδάριθμος.

ii. Καραγρηγορίου, Α. (2024). *Στατιστική μηχανική μάθηση* (Σημειώσεις μαθήματος). Πανεπιστήμιο Πειραιώς.

iii. Κουτράς, Μ., & Ευαγγελαράς, Χ. (2018). *Ανάλυση παλινδρόμησης: Θεωρία και εφαρμογές*. Εκδόσεις Τσότρας.

R Packages

i. Friedman, J., Hastie, T., & Tibshirani, R. (2023). glmnet: Lasso and elastic-net regularized generalized linear models (R package). https://CRAN.R-project.org/package=glmnet

ii. Ripley, B. D. (2023). tree: Classification and regression trees (R package). https://CRAN.R-project.org/package=tree

iii. Liaw, A., & Wiener, M. (2023). randomForest: Breiman and Cutler's random forests for classification and regression (R package). https://CRAN.R-project.org/package=randomForest

iv. Wickham, H. (2016). ggplot2: Elegant graphics for data analysis. Springer.

v. Hlavac, M. (2022). stargazer: Well-formatted regression and summary statistics tables (R package). https://CRAN.R-project.org/package=stargazer

vi. Schelling, C. (2019). catTools: Tools for categorization, splitting and labeling data (R package). https://CRAN.R-project.org/package=catTools

R Datasets

i. R Core Team. (2025). *airquality: New York air quality measurements*. In *R datasets*. R Foundation for Statistical Computing.

ii. Venables, W. N., & Ripley, B. D. (2002). *Boston: Housing values in suburbs of Boston*. In *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. R package version 7.3