CLARA YOKOCHI DE SOUSA SAMPAIO
Bachelor in Mathematics

# APPLICATION OF REGULARIZATION METHODS TO HIGH-DIMENSIONAL DATA AS TOOL FOR PREDICTING THE GEOGRAPHIC ORIGIN OF THE SALTWATER CLAM *RUDITAPES PHILIPPINARUM*

## A CONTRIBUTION TOWARDS THE FIGHT AGAINST ILLEGAL, UNREPORTED AND UNREGULATED FISHING

MASTER IN MATHEMATICS AND APPLICATIONS

NOVA University Lisbon
October, 2021

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATHEMATICS

# APPLICATION OF REGULARIZATION METHODS TO HIGH-DIMENSIONAL DATA AS TOOL FOR PREDICTING THE GEOGRAPHIC ORIGIN OF THE SALTWATER CLAM *RUDITAPES PHILIPPINARUM*

## A CONTRIBUTION TOWARDS THE FIGHT AGAINST ILLEGAL, UNREPORTED AND UNREGULATED FISHING

CLARA YOKOCHI DE SOUSA SAMPAIO

Bachelor in Mathematics

**Adviser:** Regina Maria Baltazar Bispo
*Assistant Professor, NOVA School of Science and Technology, Universidade NOVA de Lisboa*

MASTER IN MATHEMATICS AND APPLICATIONS

NOVA University Lisbon
October, 2021

**Application of regularization methods to high-dimensional data as tool for predicting the geographic origin of the saltwater clam *Ruditapes philippinarum***

# ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Regina Bispo, for her support and enthusiasm throughout the development of this thesis and other projects. Her passion for statistics was one of my greatest inspirations to follow a path in statistics as well. I would also like to express my sincere gratitude to the *Graduate Research Fellowship, within the research area of Statistics and Risk Management of CMA, Center for Mathematics and Applications*, for providing financial support to develop my research.

To conclude, I cannot forget to thank my family, friends and everyone else who, intentionally or unintentionally, helped shape the dedicated person I am today.

*"So long, and thanks for all the fish."*
*- Douglas Adams, The Hitchhiker's Guide to the Galaxy*

# ABSTRACT

As a consequence of science and technology evolution, data dimensionality has been growing and, alongside, the need to solve problems containing these complex types of data. Generically, a data analysis problem is termed *high-dimensional* when the amount of variables used to explain a certain phenomenon is higher than the number of instances of this same event in a dataset. In the context of *Linear* and *Generalized Linear Models*, high-dimensional datasets provoke the non-invertibility of the *Fisher Information Matrix*, which interferes with the estimation of the model parameters. Most regression models resort to intricate numerical and iterative methods for the assessment of the model coefficients, which often require the non-singularity of the *Fisher Information Matrix*. To tackle the difficulties that emerge when the model's *Fisher Information Matrix* is singular, a series of regularization methods have been used to analyze which predictor variables have significant linkage to the outcome and estimate their coefficients. *Ridge*, Least Absolute Shrinkage and Selection Operator (LASSO) and *Elastic Net* methods were at the outset of regularization techniques and, because of this, they are seen as being eminently linked. The algorithms behind these three methods differ in few aspects, seemingly in such a way that LASSO overcomes *Ridge*'s and *Elastic Net* overcomes LASSO's difficulties.

To counter fraud connected to the mislabeling of product origin, regularization methods were applied to predict the location of origin of *Ruditapes philippinarum*, a species of saltwater clam that is commercially harvested for human consumption. The exploited dataset constitutes 30 clam samples, detailing information on 44 composition features, with the purpose of identifying which features distinguish between three geographic origins: Ria de Vigo, Ria de Aveiro, Estuário do Tejo, i.e, a classical *Multinomial Logistic Regression* problem. However, given the high-dimensionality of the dataset (number of variables higher than the number of observations), the estimation of the model coefficients posses, as explained above, further difficulties. To overcome this problem, the three touched upon regularization methods were applied to model the origin of the clams. Additionally, since datasets of only 30 samples challenge the process of model validation, the re-sampling technique of *Monte Carlo Cross-Validation* was also implemented. We finalize comparing the results between the three methods.

**Keywords:** elastic net, high-dimensional data, LASSO, multinomial logistic regression, ridge

# RESUMO

O crescente desenvolvimento da ciência e tecnologia teve como resultado o aumento da dimensão de dados em diversas áreas científicas, acompanhado das complexidades que dados desta natureza trazem na aplicação de várias técnicas estatísticas. Um problema estatístico é designado como sendo de alta dimensão se o número de variáveis usadas para descrever um certo fenómeno for superior ao número de instâncias deste mesmo evento no conjunto de dados em análise. No contexto de *Modelos Lineares* e *Modelos Lineares Generalizados*, dados de alta dimensão induzem a não-invertibilidade da *Matriz de Informação de Fisher*, interferindo com a estimação dos parâmetros do modelo. Na maioria dos modelos de regressão, a estimação destes parâmetros recorre a métodos numéricos e iterativos intrincados, que implicam a determinação da inversa da *Matriz de Informação de Fisher*. Assim, para combater as complexidades causadas por conjuntos de dados de alta dimensão, foram desenvolvidos métodos de regularização com o objetivo de estimar os coeficientes das variáveis que têm maior ligação com o comportamento do fenómeno em estudo. Os métodos *Ridge*, LASSO e *Elastic Net* foram estabelecidos no início do desenvolvimento de técnicas de regularização e, por esta razão, são vistos como estando conectados. Os algoritmos destes três métodos diferem em poucos aspetos, de tal modo que o método LASSO supera as dificuldades de *Ridge*, e *Elastic Net* as de LASSO.

Para combater a fraude associada à falsificação da localização de origem de produtos, métodos de regularização foram aplicados a um conjunto de dados com o objetivo de prever o local de origem de *Ruditapes philippinarum*, uma espécie de amêijoa comercialmente colhida para consumo. Os dados constituem observações de 30 amêijoas, detalhando informação relativa a 44 elementos de composição, com o propósito de identificar quais os que melhor distinguem entre três origens geográficas: Ria de Vigo, Ria de Aveiro, Estuário do Tejo, i.e., um modelo de *Regressão Logística Multinomial*. Contudo, dado que o problema é de alta dimensão, a estimação dos coeficientes do modelo enfrenta complexidades. Assim, os três métodos de regularização mencionados foram aplicados. Adicionalmente, uma vez que conjuntos de dados com apenas 30 instâncias dificultam a validação do modelo, o método de *Validação Cruzada de Monte Carlo* foi implementado. Por fim, compararam-se os resultados obtidos pelos três métodos.

**Palavras-chave:** alta dimensão, elastic net, LASSO, regressão logística multinomial, ridge

# CONTENTS

# List of Figures

# List of Tables

# ACRONYMS

| | |
|---|---|
| **AUC** | Area Under the Curve |
| **BIC** | Bayesian Information Criterion |
| **CFS** | Correlation Feature Selection |
| **CV** | Cross-Validation |
| **DOB-SCV** | Distribution Optimally Balanced Stratified Cross Validation |
| **ESCV** | Estimation Stability Cross-Validation |
| **ET** | Efficient Tuning |
| **FAIR** | Feature Annealed Feature Selection |
| **FCBF** | Fast Correlation-Based Feature |
| **FCBF** | Iterated Sure Independence Screening |
| **INT** | INTERACT |
| **LASSO** | Least Absolute Shrinkage and Selection Operator |
| **mce** | Misclassification Error |
| **NSC** | Nearest Shrunken Centroids |
| **PLR** | Penalized Logistic Regression |
| **RFE** | Recursive Feature Elimination |
| **ROC** | Receiver Operating Characteristic |

**se** Standard error
**sem** Standard error of the mean
**SIS** Sure Independence Rules
**SVM** Support Vector Machine

**UR** Univariate Ranking

# SYMBOLS

| | |
|---|---|
| $\theta$ | Location parameter of a variable belonging to the exponential family |
| $\phi$ | Dispersion parameter of a variable belonging to the exponential family |
| $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ | Vector of model coefficients in a regression model with $p$ variables |
| $\boldsymbol{\pi} = (p_0, \ldots, p_J)^T$ | Vector of probabilities of class affiliation of a variable with *Multinomial* distribution |
| $\otimes$ | Kronecker product |
| $\lambda$ | Penalization parameter of a regularization model |
| $\alpha$ | Mixing parameter of a regularization model |
| $\hat{\boldsymbol{\beta}}^{EN}$ | Model coefficients' estimates resorting to the *Elastic Net* method of regularization |
| $\hat{\boldsymbol{\beta}}^{L}$ | Model coefficients' estimates resorting to the LASSO method of regularization |
| $\hat{\boldsymbol{\beta}}^{R}$ | Model coefficients' estimates resorting to the *Ridge* method of regularization |
| $\mu_i$ | Expected value of $Y_i \vert \mathbf{x}_i$, $i \in \{1, \ldots, n\}$ |
| $\eta_i$ | Linear predictor for the $ith$ instance, $i \in \{1, \ldots, n\}$ |
| $\pi_{iq}$ | Probability of observation $i$ belonging to class $q$, i $\in \{1, \ldots, n\}$, $q \in \{0, \ldots, J\}$ |
| $a!$ | Factorial of $a$, where $a \in \mathbb{N}_0^+$ |
| $\vert a \vert$ | Absolute value of $a$, $a \in \mathbb{R}$ |
| $h^{-1}(\cdot)$ | Link function of a regression model |
| $\vert\vert \mathbf{v} \vert\vert_1$ | $\ell_1$ norm (Manhattan distance) of vector $\mathbf{v}$ |
| $\vert\vert \mathbf{v} \vert\vert_2$ | $\ell_2$ norm (euclidean norm) of vector $\mathbf{v}$ |
| $\mathbf{M}^{-1}$ | Inverse of matrix $\mathbf{M}$, where $\mathbf{M}$ is non-singular |
| $\mathbf{M}^T$ | Transpose of matrix $\mathbf{M}$ |
| $\mathbf{v}^T$ | Transpose of vector $\mathbf{v}$ |
| $\hat{\boldsymbol{B}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)^T$ | Maximum likelihood estimators coefficients of a model with $p$ variables |

| | |
|---|---|
| $\mathbf{exp}(a)$ | Exponential of $a$ ($e^a$), where $a \in \mathbb{R}$ |
| $\mathbb{E}[\cdot]$ | Expected value of random variable |
| $f'(\cdot)$ | First derivative of function $f(\cdot)$ |
| $f''(\cdot)$ | Second derivative of function $f(\cdot)$ |
| $f_Y^{\beta}(\cdot)$ | Probability density function or probability mass function of random variable $Y$, where $\beta$ is the vector of parameters in study |
| $I(\beta)$ | *Fisher Information Matrix* of a model with a vector of parameters $\beta$ |
| $I_a$ | Identity matrix of dimension $a \times a$ |
| $[I(\beta)]_{j1\,j2}$ | $j1\,th$ row and $j2\,th$ column component of the *Fisher Information Matrix*, $j1 \in \{1,\ldots,p+1\}$, $j2 \in \{1,\ldots,p+1\}$ |
| $J+1$ | Number of classes of a variables with *Multinomial* distribution |
| $L(\cdot)$ | Likelihood function |
| $\mathbf{log}(a)$ | Logarithm of $a$, where $a \in \mathbb{R}_0^+$ |
| $l(\cdot)$ | Log-likelihood function |
| $n$ | Number of data instances/observations in a dataset |
| $p$ | Number of variables in a dataset |
| $p_q$ | Probability of success for the $q\,th$ outcome, in a variable with *Multinomial* distribution, $q \in \{0,1,\ldots,J\}$ |
| $s(\cdot)$ | Score function |
| $\mathbb{V}[\cdot]$ | Variance of random variable |
| $\mathbf{x} = (x_1,\ldots,x_p)^T$ | Vector of predictor variables of a regression model |
| $x_{ik}$ | Value of the $k\,th$ predictor variable in the $i\,th$ instance, $i = 1,\ldots,n$, $k = 1,\ldots,p$ |
| $Y$ | Response variable of a regression model |
| $\mathbf{Y} = (Y_1,\ldots,Y_n)^T$ | Sample of $Y$, *i.e.*, vector of random variables, independent and identically distributed with $Y$ |
| $Y_i$ | Outcome for each observation $i \in \{1,\ldots,n\}$ |
| $(y_i,\mathbf{x}_i)$ | Data resulting from $n$ realizations of $(Y,\mathbf{x})$, $i = 1,\ldots,n$ |
| $\mathbf{Z}_D$ | Design matrix of a regression model with $p$ variables and $n$ observations |

| | |
|---|---|
| $z_{im}$ | $ith$ row and $mth$ column component of the model's design matrix, $i \in \{1,\ldots,n\}$, $m \in \{1,\ldots,p+1\}$ |

# 1

# INTRODUCTION

Numerous scientific fields frequently face the need to manipulate elaborate high-dimensional datasets. Unfortunately, due to the mathematical complexities linked to high-dimensional problems, it has brought up many complications when it comes to implementing what would otherwise be relatively simple tasks. In the context of *Linear* and *Generalized Linear Models*, these datasets interfere with the invertibility of the *Fisher Information Matrix*, thus complicating the process of estimating the model parameters. As consequence, plenty of the required tasks in regular statistical problems become complex and computationally demanding. To deal with these difficulties, a series of regularization methods, such as *Ridge*, LASSO and *Elastic Net*, were assembled to analyze and estimate the coefficients of the predictor variables that have a higher linkage to the phenomenon outcome. LASSO and *Elastic Net* are also known as *Feature Selection Methods* , since they allow to select a subset of relevant features that properly describe the concerned phenomenon. This is done through the algorithm's ability of detecting the significant features and discarding the irrelevant and redundant ones, while taking into account the possible correlation between variables. Although *Ridge* is not technically a feature selection method, it is still a regularization method, since it performs by shrinking the coefficients of less significant features and giving a heavier weight to the most important features.

## 1.1 Motivation

This thesis addresses the issue of modeling high-dimensional data by means of *Generalized Linear Models*, in particular, *Multinomial Logistic Regression* models, that are used to model multi-categorical outcomes. We resort to the regularization methods of *Ridge*, LASSO and *Elastic Net* methods. In order to perform a comparative study of these three methods, we applied them to a set of data with the purpose of creating viable models to predict the location of a marine life species (*Ruditapes philippinarum*). Finding which features are most relevant for predicting the location of origin of this species is essential, because attaining information of each feature is time and cost demanding. Because this species is commercially harvested for consumption, these models are valuable for answering questions regarding product traceability therefore controlling product quality and safeguarding the interest of the consumers, as well as

regulating the growth of product origin forgery issues.

## 1.2 Objectives

We aim to perform an extensive analysis of the *Ridge*, LASSO and *Elastic Net* regularization methods, through applying them to a set of data.

The manipulated dataset contains detailed information on the biochemical and geochemical fingerprints of a species of saltwater clam, *Ruditapes philippinarum*. The dataset is composed of 30 clam samples, containing information on 44 composition features. The phenomenon in study is represented by a multi-class variable that serves as an indicator of the clam's location of origin: Ria de Vigo (Galiza, Spain), Ria de Aveiro (Aveiro, Portugal), Estuário do Tejo (Lisbon, Portugal). In summary, we want to predict of the clam's location of origin, based on their biochemical and geochemical fingerprints, by means of fitting a *Multinomial Logistic Regression*. Because it is a high-dimensional set of data, where there are more features than observations, the three mentioned regularization methods were applied in order to establish which predictor variables have a higher linkage to the geographic origin of the clams. This way, we desire to assemble models to oppose the fraud that is related to the falsification of product origin. Additionally, to study the predictive quality and performance of the three regularization methods, we resorted to *Monte Carlo Cross-Validation* as a re-sampling procedure to establish different training and testing sets for model implementation and validation, respectively. This is done due to the reduced amount of data observations, in order to achieve more statistically relevant conclusions.

## 1.3 Structure

In addiction to this introductory chapter, this thesis is arranged in 5 chapters. Chapter 2 constitutes the *Literature Review*, where we discuss the advantages and disadvantages of a series of different regularization methods. Additionally, we highlight a broad number of previous regularization studies, accompanied by a comparative analysis of the results achieved through the application of these methods. In Chapter 3 we discuss the different methods that are used throughout the data analysis process, including a description of *Generalized Linear Models*, with special elaboration on *Multinomial Logistic Regression* models, measures of *Predictive Quality*, measures of *Variable Importance* and *Residual Analysis*. Moreover, in this chapter we also specify the *Monte Carlo Cross-Validation* re-sampling method, and three different regularization methods: *Ridge*, LASSO and *Elastic Net*. Chapter 4 is where we implement the methods described in the previous chapter to a real high-dimensional dataset containing detailed information of a species of saltwater clam. Finally, in Chapter 5 we finalize comparing the results between the three methods, identifying which has the best predictive performance and comparing the estimation errors in each category of the response.

# 2

# LITERATURE REVIEW

This thesis focuses on analyzing high-dimensional data through the implementation of three different regularization techniques: *Ridge*, LASSO and *Elastic Net*, with the intention of making a comparative study between them. However, there have been developed plenty of other regularization methods to address high-dimensionality. In this chapter, we discuss several studies that compare results between the three implemented methods, as well as plenty of other methods, in classification problems with categorical outcomes (multi-class or binary) and where the number of predictors is higher than the number of observations. These methods include extensions of the *Ridge*, LASSO and *Elastic Net* methods, *Support Vector Machine* and *Penalized Logistic Regression*, using both *Recursive Feature Elimination* and *Univariate Ranking, Nearest Shrunken Centroids, Feature Annealed Independence Rules, Sure Independence Screening* and *Iteratively Sure Independent Screening*. We also elaborate on the advantages and disadvantages of the different regularization methods.

The *Ridge* method generally exhibits good performance measures but, since it only performs coefficient shrinkage, and not variable selection, the derived models are often very dense. The LASSO method overcomes Ridge's disadvantages by performing variable selection, however, this method can only select, at most, a number of variables equal to the number of observations. One of the major difficulties of this method is the inability to perform group selection. This signifies that, if the data contains groups of highly correlated variables, then the LASSO tends to arbitrarily select only one predictor from each group and drop the remainder. To respond to the drawbacks of the LASSO, the *Elastic Net* is not only able to select more variables than the sample size but, if fronted with a group of highly correlated variables, this method stimulates a grouping effect, where highly correlated predictors tend to be in or out of the model together (Zou & Hastie, 2005). Additionally, according these authors, who proposed the *Elastic Net* as a method of regularization, it is particularly useful when the number of variables is much larger than the number of observations.

*Supportive Vector Machine* is recognized as an effective method in high-dimensional spaces, however, considering a classification problem where the response variable is categorical, this method only allows class prediction, not providing any estimate of underlying probability. As an alternative, Zhu and Hastie (2004) proposed the *Penalized Logistic Regression* method, which

in theory often performs similarly to *Supportive Vector Machine*, but provides estimates for probability of affiliation.

Proposed by Tibshirani et al. (2003), the *Nearest Shrunken Centroids* method is identified as an appealing tool for feature selection and classification with high-dimensional data. Likewise, J. Fan and Fan (2008) also suggested the *Feature Annealed Independence Rules* as a method of classification using high-dimensional datasets. Nonetheless, much like the LASSO method, *Nearest Shrunken Centroids* and *Feature Annealed Independence Rules* do not conduct a proper interpretation of highly correlated variables, sometimes yielding misleading feature selection and poor classification (Mai et al., 2012).

J. Fan and Lv (2008) introduced the *Sure Independence Screening* method that, based on correlation learning, reduces the dimensionality of a high-dimensional problem to moderate scale, that is, below the sample size. The authors showed that, under certain regularity conditions, this fast variable selection method has a sure screening property, that is, it retains all of the important variables in the model, with probability close to 1. Consequently, an extension of this method was also proposed. *Iterative Sure Independence Screening* acts as a performance enhancer for the first method in problems where regularity conditions might fail, for example, if a predictor is marginally uncorrelated, but jointly correlated with the response variable (J. Fan et al., 2008). Nevertheless, these methods have been known to be conservative and sometimes ending up including many unimportant variables in the model.

Methods like *Correlation-based Feature Selection* (Hall, 1999) and *Fast Correlation-based Feature Selection* (Yu & Liu, 2003) perform by selecting features that are highly correlated with the response yet uncorrelated with each other. However, these methods have been criticized by their computational complexity (Pino & Morell, 2013).

Similarly to *Iterative Sure Independence Screening*, Zhao and Liu (2007) proposed the *INTERACT* method as a response to the issue that comes with unintentional feature removal when a feature by itself does not show much correlation with the response, but when combined with other features, is highly correlated with the response. Still, this method can only manage nominal features and, if that is not the case, valuable information may be lost in the process of discretization of the data (Zeng et al., 2015).

## 2.1 *Ridge*

This method was proposed by Hoerl and Kennard (1970) to address multicollinearity in the context of *Linear* and *Generalized Linear Modeling*. Multicollinearity is the phenomenon in which one or more predictor variables can be linearly predicted from other variables. As expected, this issue often occurs in high-dimensional datasets, where it is more likely for groups of predictors to be highly correlated (Hilt & Seegrist, 1977). This method has been shown to achieve the same performance as *Support Vector Machine*, with the additional benefit of computing probabilities of class affiliation rather than scores. However, the high density of the models constructed by this regularization method are often inconvenient when solving high-dimensional problems (Aseervatham et al., 2011).

Aseervatham et al. (2011) and Pereira et al. (2016) studied the performance of the *Ridge* method in *Logistic Regression*. The first article covers large-scale text categorization (assigning a text document to one or more relevant categories, according to its content), and proposes *Selected Ridge*, a new method that constitutes a broadening of the *Ridge* method to feature selection. Besides implementing this new method, the authors include in their analysis the regular *Ridge* and LASSO methods, in order to compare results. These three methods were applied to four distinct sets of data, reaching the overall conclusion that, for the most part (three out of the four analyzed datasets), the *Ridge* method either performed equally or outperformed the LASSO method. The *Selected Ridge* method also performed equally or outperformed the regular *Ridge* method. As consequence, the *Selected Ridge* method was opt for as the best method given its better performance and property of feature selection. The second paper covers the prediction of corporate failure, and resorts to the usage of *Ridge* and LASSO methods in four different studies using the same set of data but considering different partitions of *training/testing sets*. They concluded that none of these methods behaved clearly better than the other, but that both of them tended to favor the category of the response variable that appeared with heavier weight in the *training set*.

## 2.2 LASSO

Proposed by Tibshirani (1996), the LASSO regularization technique has proven to be a flexible tool to select relevant features and estimate the model coefficients simultaneously. This method has been widely used in numerous research areas where the number of variables is very large, such as genome studies, finance, risk analysis, biomedical imaging, and others. Despite its notoriety, it is sometimes challenging to guarantee the feature selection consistency of LASSO, especially when the dimension of the data is very large (Yang et al., 2019). This method yields sparse solutions however its performance is generally dominated by the *Ridge* method when the number of features is larger than the number of observations and/or when the features are highly correlated (Aseervatham et al., 2011).

Asenso et al. (2020) and Yang et al. (2019) employed the LASSO, as well as some variations of this method, in order to study several high-dimensional problems. While the first publication focuses on the applicability of LASSO in genomics and disease detection/severity, the second article covers a store product management, which helped us understand how diverse high-dimensional problems can be. In both of these publications, the authors resorted to using a variety of other regularization methods besides LASSO and its extensions. Aside from regular LASSO, Asenso et al. (2020) applied the *Elastic Net* method, as well as an extension of LASSO entitled *Pliable* LASSO. The authors applied these methods to two sets of high-dimensional data containing information on gene expression levels to study the severity of breast carcinoma (85 observations, 60 for training and 25 for testing the models, and 456 features) and small, round blue cell tumors of childhood - SRBCT (83 observations, 63 for training and 20 for testing the models, and 2.308 features). In doing so, they discovered that there were no significant differences between the predictive performance of *Pliable* LASSO in relation to the other two

methods. For the breast carcinoma dataset, *Pliable* LASSO selected 65, LASSO 11 and *Elastic Net* 39 features. Although these results were not accompanied by errors, it was indicated that the *Pliable* LASSO method did not perform far better than the other two methods. For the SRBCT data, the authors specified that the misclassification error was the same for the three methods (= 1/20), despite *Pliable* LASSO selecting 25, LASSO 12 and *Elastic Net* 31 features. This signifies that LASSO, although selecting the least amount of features, performs the same as the other two methods in terms of predictive error. For both of these studies notice that if the main objective was to achieve the least dense model with the best predictive quality, LASSO would be opted as the best method of regularization.

We also studied the article published by Yang et al. (2019). The authors resorted to the usage of four distinct LASSO variations that differ in the procedure used for estimating the penalization parameter (that estimates the ideal the amount of penalization that the dataset requires). These procedures include *Efficient Tuning* (ET LASSO), *Cross-Validation* (CV LASSO), *Bayesian Information Criterion* (BIC LASSO) and *Estimation Stability with Cross-Validation* (ESCV LASSO). These methods were applied to a set of data to understand the daily sales of a Chinese supermarket given 6.398 different products in a period of 464 days. It was concluded that ET LASSO led to the smallest error despite selecting only 68 products, followed ESCV with 72 products, CV with 111 products and then BIC with 100 products.

J. Fan et al. (2008) also utilized the LASSO method to study how genetic information influences certain diseases, in this case, the 3-year survival rate of a neuroblastoma patient. The data consisted of 239 patient observations (125 for training and 114 for testing), with information on 10.707 features. Besides LASSO, the authors resorted to the usage of *Nearest Shrunken Centroids*, *Sure Independence Screening*, *Iterative Sure Independence Screening* and an extension of the last two methods. They reached that the method that performed the worst was *Nearest Shrunken Centroids*, selecting 9.413 features but failing to predict 24 out of the 114 test observations. The best performing method was regular *Sure Independence Screening*, selecting only 5 genes with a misclassification rate of 19/114. The remaining models performed in an equal manner with an error of 22/114, although LASSO selected the highest amount of predictors (57 genes) and the extension of *Sure Independence Screening* the least (10 genes).

## 2.3 *Elastic Net*

Zou and Hastie (2005) proposed the *Elastic Net* as a regularization method that out-performs LASSO. Unlike LASSO, this method tends to take into account the high correlation that might exist between groups of predictors. It is known to be particularly useful when the number of predictors is much larger than the number of observations.

Zou and Hastie (2005) and Algamal and Lee (2015) applied the *Elastic Net* to analyze classification of disease severity. Besides *Elastic Net*, Zou and Hastie (2005) resorted to methods like *Support Vector Machine by Recursive Feature Elimination*, *Penalized Logistic Regression by Recursive Feature Elimination* and *Nearest Shrunken Centroids*, and compared the results they obtained with a study made by Golub et al. (1999), to study the influence of gene expression

levels on leukemia disease, a dataset composed of 72 observations (38 for training and 34 for testing), and 7.129 features. In this study, they reached that *Elastic Net* performed the best in terms misclassification error of the testing set (0/34), although it was the method that selected the second highest amount of features (45 genes). The variation of *Support Vector Machine by Univariate Ranking* and *Penalized Logistic Regression by Univariate Ranking* was used by Zhu and Hastie (2004), which reached worse results than when resorting to *Recursive Feature Elimination*. J. Fan and Fan (2008) used this same dataset to analyze the performance of *Feature Annealed Independence Rules*, reaching a total of 11 features, with a misclassification rate of 1/34, concluding that it performed sightly worse than *Elastic Net*.

Algamal and Lee (2015) utilized, in addition to the regular *Elastic Net* method, two other variations of *Elastic Net*: *Adaptive Elastic Net* and *Adjusted Adaptive Elastic Net*, as well as a variation of the *Ridge* method that performs variable selection entitled *AERidge*. These methods were used to study the severity of three different cancer high-dimensional sets of data containing detailed information on gene expression levels: prostate cancer with 102 samples and 5.966 features, diffuse large B-cell lymphoma with 77 observations and 7.129 features, colon cancer with 62 observations and 2.000 features. In all of these studies, the authors concluded that the *Adjusted Adaptive Elastic Net* reached the models with the smallest error and that, considering just regular *Elastic Net* and *AERidge* methods, *AERidge* outperformed *Elastic Net*, with regards to error, while still selecting fewer features.

## 2.4 Other methods

Zhu and Hastie (2004) also applied *Support Vector Machine* and *Penalized Logistic Regression* to genetic data in order to study various problems. Similarly to Asenso et al. (2020), these authors studied the SRBCT data, but resorted to *Support Vector Machine* and *Penalized Logistic Regression* by both *Univariate Ranking* and *Recursive Feature Elimination*. Moreover, they also applied these methods to the data presented in Ramaswamy et al. (2001) (144 training observations, 54 testing observations, 16.063 predictors). Both of these studies led them to conclude that *Recursive Feature Elimination* produced the models with the best predictive quality when compared to *Univariate Ranking*, in both regularization methods used. Notice that for the SRBCT data, *Recursive Feature Elimination* selected a higher number of features than *Univariate Ranking*, but in Ramaswamy data, the opposite occurred.

Through analyzing the articles by J. Fan and Fan (2008) and J. Fan et al. (2008), we compared the performance of *Nearest Shrunken Centroids* and other methods such as LASSO and *Feature Annealed Independence Rules*. Jointly, they applied these methods to four sets of data to study disease classification: leukemia, lung cancer, prostate cancer and neuroblastoma. In section 2.2, we remarked how *Nearest Shrunken Centroids* performed the worst (compared to LASSO, and two variations of *Sure Independence Screening* and *Iterative Sure Independence Screening*), while simultaneously selecting the highest amount of predictors into the model. In J. Fan and Fan (2008) the authors showed that the performance of *Nearest Shrunken Centroids* is oftentimes

worse than *Feature Annealed Independence Rules* even though, in two out of the three datasets they studied, it selected more predictors.

To finalize, Bolón-Canedo (2014) studied the application of *Correlation-based Feature Selection*, *Fast Correlation-based Feature Selection* and *INTERACT* to a series of different datasets. With the purpose of analyzing the adequacy of performance, the author resorted to *Cross-Validation* and *Distribution Optimally Balanced Stratified Cross-Validation*. Though it is not emphatically expressed a comparison analysis between these methods' performance, we see that generally *Fast Correlation-based Feature Selection* selects the fewest amount of features, while not being remarked as having a much worse performance than the other two methods applied. In parallel, *Correlation-based Feature Selection* is usually the method that selects the highest amount of features.

For a more detailed analysis of the studies mentioned, see Tables A.1 through A.11.

# 3

# METHODS

## 3.1 Generalized Linear Models

Within the field of *Modeling* in statistics, *Regression models* concern an ensemble of procedures used to model the relationship between a phenomenon, mathematically represented by an outcome variable, and a set of features that can trigger this event, portrayed by explanatory variables. The relationship between the phenomenon and the features that trigger it can be either linear or non-linear. In the linear case, we have the termed *Linear Regression models*, where the relationship is modeled by using a linear predictor function whose parameters are estimated from the data. This type of model relies on the assumption that the conditional mean of the outcome variable, given the explanatory variables, can be derived from a linear function of the explanatory variables.

Let $Y$ be the dependent outcome variable, and $\mathbf{x} = (x_1, \ldots, x_p)^T$ be the set of $p$ explanatory variables (also known as covariates or predictor variables), that are used to explain part of the inherit behavior of $Y$. $Y$ can be continuous, discrete or categorical. The covariates can be continuous, discrete, ordinal or categorical (Turkman & Silva, 2000). We assume that, given an experience of the phenomenon in study in $n$ different instances, we have a dataset with the format,

$$(y_i, \mathbf{x}_i), \quad i \in 1, \ldots, n, \tag{3.1}$$

ensuing the realization of $(Y, \mathbf{x})$ in $n$ instances. The components $Y_i$ of the random vector $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ are independent.

Consider the matricial representations of the data,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix}, \quad \mathbf{Z}_D = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1p} \\ 1 & x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix} \tag{3.2}$$

In a classic *Linear Regression model*, the relationship between the dependent and the explanatory variables is defined as,

$$\mathbf{Y} = \mathbf{Z}_D \, \boldsymbol{\beta} + \epsilon, \tag{3.3}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ is the vector of the model parameters (or coefficients) that require estimation, $\mathbf{Z}_D$ is the $n \times (p+1)$ design matrix of the model, of components $[z_{im}]$, $i \in \{1, \ldots, n\}$, $m \in \{1, \ldots, p+1\}$ (covariate matrix $\mathbf{X}$ combined with a unitary vector as its first column), and $\epsilon$ is a $n \times 1$ vector that represents the noise term of the model, constituting other possible factors that influence $\mathbf{Y}$ (assumed to have a $N(0, \sigma^2\mathbf{I})$ distribution (Turkman & Silva, 2000)). Note that there are terms that are observable and others that are not. The observable factors are $\mathbf{Y}$ and $\mathbf{X}$, that compose the available data. $\epsilon$ is unobserved. The vector of model coefficients $\boldsymbol{\beta}$ is also unobserved. The main task of the model is to estimate the model parameters based on the information provided by the dataset.

On a dataset with $n$ independent instances, for each observation $i \in \{1, \ldots, n\}$, we have an outcome $Y_i$, and a $p \times 1$ vector of covariates $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, where $x_{ik}$ ($k \in \{1, \ldots, p\}$) represents the value for the $k$*th* covariate in observation $i$.

*Generalized Linear Models* are a generalization of the *Linear Models* to an outcome whose distribution belongs to the *Exponential Family*, and whose mean value relates to the explanatory variables by means of a differentiable function.

### 3.1.1 Exponential Family

One of the main assumptions in *Generalized Linear Modeling* is that the distribution of the outcome variable belongs to the exponential family, *i.e.*, that the probability density function (or the probability mass function) be written as,

$$f(y|\theta, \phi) = \exp\Big\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\Big\}, \tag{3.4}$$

where $\theta$ is the location parameter, $\phi$ is the dispersion parameter, $a(\cdot)$, $b(\cdot)$ and $c(\cdot, \cdot)$ are known real-valued regular functions.

#### 3.1.1.1 Expected Value

Let us demonstrate that, for *Generalized Linear Models*, we have,

$$\mathbb{E}[Y] = b'(\theta). \tag{3.5}$$

Given that $\int_{-\infty}^{+\infty} f(y|\theta,\phi)\,dy = 1$ and $\frac{\partial}{\partial\theta}1 = 0$, we have,

$$\begin{aligned}\frac{\partial}{\partial\theta}f(y|\theta,\phi) &= \frac{\partial}{\partial\theta}\exp\Big\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\Big\}\\ &= \Big(\frac{y - b'(\theta)}{a(\phi)}\Big)f(y|\theta,\phi).\end{aligned} \tag{3.6}$$

For the purpose of this thesis, we assume that we are under the regularity condition that allows us to infer that,

$$\frac{\partial}{\partial\theta}\Big(\int_{-\infty}^{+\infty} f(y|\theta,\phi)\Big)\,dy = \int_{-\infty}^{+\infty}\frac{\partial}{\partial\theta}f(y|\theta,\phi)\,dy. \tag{3.7}$$

Consequently we find that the expected value of the outcome variable can be calculated by solving the equation,

$$\begin{aligned}0 &= \frac{\partial}{\partial\theta}\Big(\int_{-\infty}^{+\infty} f(y|\theta,\phi)\Big)\,dy\\ &= \int_{-\infty}^{+\infty}\frac{\partial}{\partial\theta}f(y|\theta,\phi)\,dy\\ &= \int_{-\infty}^{+\infty}\Big(\frac{y - b'(\theta)}{a(\phi)}\Big)f(y|\theta,\phi)\,dy\\ &= \frac{1}{a(\phi)}\Big[\int_{-\infty}^{+\infty} y f(y|\theta,\phi)\,dy - b'(\theta)\int_{-\infty}^{+\infty} f(y|\theta,\phi)\,dy\Big]\\ &= \mathbb{E}[Y] - b'(\theta).\end{aligned} \tag{3.8}$$

Hence $\mathbb{E}[Y] = b'(\theta)$.

#### 3.1.1.2 Variance

Let us prove that, in a *Generalized Linear Model*, we have,

$$\mathbb{V}[Y] = b''(\theta)\,a(\phi). \tag{3.9}$$

Similarly to the deduction made for the expected value of the outcome, the variance of this

variable can be obtained by considering that,

$$\begin{aligned}\frac{\partial^2}{\partial\theta^2} f(y|\theta,\phi) &= \frac{\partial}{\partial\theta}\left[\left(\frac{y-b'(\theta)}{a(\phi)}\right) f(y|\theta,\phi)\right] \\ &= -\frac{b''(\theta)}{a(\phi)} f(y|\theta,\phi) + \left(\frac{y-b'(\theta)}{a(\phi)}\right)\frac{\partial}{\partial\theta} f(y|\theta,\phi) \\ &= -\frac{b''(\theta)}{a(\phi)} f(y|\theta,\phi) + \left(\frac{y-b'(\theta)}{a(\phi)}\right)^2 f(y|\theta,\phi).\end{aligned} \tag{3.10}$$

Once more, due to the regularity condition (3.7) and the fact that,

$$\frac{\partial^2}{\partial\theta^2}\int_{-\infty}^{+\infty} f(y|\theta,\phi) = 0, \tag{3.11}$$

we can find a simplified expression for the variance of the outcome variable by solving,

$$\begin{aligned}0 = \frac{\partial^2}{\partial\theta^2}\int_{-\infty}^{+\infty} f(y|\theta,\phi)\,dy &= \int_{-\infty}^{+\infty}\frac{\partial^2}{\partial\theta^2} f(y|\theta,\phi)\,dy \\ &= -\int_{-\infty}^{+\infty}\frac{b''(\theta)}{a(\phi)} f(y|\theta,\phi)\,dy + \int_{-\infty}^{+\infty}\left(\frac{y-b'(\theta)}{a(\phi)}\right)^2 f(y|\theta,\phi)\,dy \\ &= -b''(\theta)\,a(\phi)\int_{-\infty}^{+\infty} f(y|\theta,\phi)\,dy + \int_{-\infty}^{+\infty}(y-b'(\theta))^2 f(y|\theta,\phi)\,dy \\ &= -b''(\theta)\,a(\phi) + \int_{-\infty}^{+\infty} y^2 f(y|\theta,\phi)\,dy + b'(\theta)^2\int_{-\infty}^{+\infty} f(y|\theta,\phi)\,dy - 2b'(\theta)\int_{-\infty}^{+\infty} y f(y|\theta,\phi)\,dy \\ &= -b''(\theta)\,a(\phi) + \mathbb{E}[Y^2] + b'(\theta)^2 - 2b'(\theta)\,\mathbb{E}[Y] \\ &= -b''(\theta)\,a(\phi) + \mathbb{E}[Y^2] + b'(\theta)^2 - 2b'(\theta)^2 \\ &= -b''(\theta)\,a(\phi) + \mathbb{E}[Y^2] - b'(\theta)^2.\end{aligned} \tag{3.12}$$

Meaning that $\mathbb{E}[Y^2] = b''(\theta)\,a(\phi) + b'(\theta)^2$. This way,

$$\mathbb{V}[Y] = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 = b''(\theta)\,a(\phi) + b'(\theta)^2 - b'(\theta)^2 = b''(\theta)\,a(\phi). \tag{3.13}$$

### 3.1.2 Traits of Generalized Linear Models

*Generalized Linear Models* are characterized by three main components:

- Random component
- Systematic component
- *Link function*

#### 3.1.2.1 Random component

Let $\mathbf{x}_i$ be the vector of covariates for the $i\,th$ instance and $Y_i$ be the corresponding outcome. Then the variables $Y_i|\mathbf{x}_i$ are independent and their distribution belongs to the *Exponential Family*, with the expected value,

$$\mathbb{E}[Y_i|\mathbf{x}_i] = \mu_i = b'(\theta_i), \quad i = 1,\ldots,n. \tag{3.14}$$

#### 3.1.2.2 Systematic component

The systematic component defines the linear predictor as a linear combination of the explanatory variables,

$$\eta_i = \mathbf{z}_i^T \boldsymbol{\beta}, \quad i = 1,\ldots,n, \tag{3.15}$$

where $\mathbf{z}_i = (1, x_{i1},\ldots,x_{ip})^T$, $i \in \{1,\ldots,n\}$, is the $i\,th$ design vector.

#### 3.1.2.3 Link function

The *link function* defines the relationship between $\mu_i$ and $\eta_i$ through,

$$\mu_i = h(\eta_i) \Leftrightarrow \eta_i = h^{-1}(\mu_i), \quad i = 1,\ldots,n, \tag{3.16}$$

where $h(\cdot)$ is a monotonous, differentiable and invertible function. $h^{-1}(\cdot)$ is the *link function* of the model.

### 3.1.3 Parameter Estimation

The main focus of any regression model falls on the estimation of the vector of model parameters, $\boldsymbol{\beta}$. Because of its structure, *Generalized Linear Models* enable the usage of the same numeric method for parameter estimation for the various types of fitted models, *i.e.*, considering different *link functions* and response variable distribution.

By definition, suppose that we have a *Generalized Linear Model* characterized by,

$$f(y_i|\theta_i,\phi) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right), \quad i = 1,\ldots,n. \tag{3.17}$$

In statistics, the *likelihood function*, $L(.)$, measures the goodness of fit of a model to a sample of data. It depicts the joint probability distribution of the sample but treating the random variables as fixed at the observed values. Consequently, it is a function of the unknown parameters

of the model only. In a *Generalized Linear Model*, we describe the *likelihood function* as,

$$\begin{aligned} L(\boldsymbol{\beta},\phi) :&= \prod_{i=1}^{n} f(y_i|\theta_i,\phi) \\ &= \prod_{i=1}^{n} \exp\Big\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\Big\} \\ &= \exp\Big\{\frac{1}{a(\phi)}\sum_{i=1}^{n}(y_i\theta_i - b(\theta_i)) + \sum_{i=1}^{n} c(y_i,\phi)\Big\}. \end{aligned} \tag{3.18}$$

The model parameters are estimated as being the values that maximize the *likelihood function*, hence this method's denomination of *Maximum Likelihood Method*. Given the properties of the logarithm, we know that the maximizers (or minizers) of a function are equal to the maximizers (or minizers) of the logarithm of that same function. The logarithm of the *likelihood function* is denominated as the *log-likelihood function*, $l(.)$.

In a *Generalized Linear Model*, we describe the *log-likelihood function* as,

$$\begin{aligned} l(\boldsymbol{\beta},\phi) :&= \log(L(\boldsymbol{\beta},\phi)) \\ &= \frac{1}{a(\phi)}\sum_{i=1}^{n}(y_i\theta_i - b(\theta_i)) + \sum_{i=1}^{n} c(y_i,\phi) \\ &= \sum_{i=1}^{n} l_i(\boldsymbol{\beta}), \quad \text{where} \quad l_i(\boldsymbol{\beta}) = \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi), \quad i = 1,\dots,n. \end{aligned} \tag{3.19}$$

We undertake the parameter $\phi$ as fixed, so that the estimation of $\boldsymbol{\beta}$ does not depend on the value of the dispersion parameter.

The *score function* is the defined as the gradient of the *log-likelihood function*. At any particular component of the parameter vector, the *score* specifies the slope of the *log-likelihood function*, thus being used to measure this function's sensitivity to infinitesimal changes to the parameter values. For the $j\,th$ model parameter, we define the $j\,th$ component of the *score function* as,

$$s_j(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n}\frac{\partial l_i(\boldsymbol{\beta})}{\partial \beta_j}, \quad j = 0,\dots,p. \tag{3.20}$$

The maximum likelihood estimators of $\boldsymbol{\beta}$ are obtained by solving the system of $p+1$ likelihood equations,

$$s_j(\boldsymbol{\beta}) = 0 \quad j = 0,\dots,p. \tag{3.21}$$

Recall that, for each $i \in \{1,\dots,n\}$, $\mu_i = b'(\theta_i)$ and $\mu_i = h(\eta_i)$. In order to find the likelihood equations, we resort to the *Chain Rule* and define the following equality,

$$\frac{\partial l_i(\boldsymbol{\beta})}{\partial \beta_j} = \frac{\partial l_i(\theta_i)}{\partial \theta_i}\frac{\partial \theta_i(\mu_i)}{\partial \mu_i}\frac{\partial \mu_i(\eta_i)}{\partial \eta_i}\frac{\partial \eta_i(\boldsymbol{\beta})}{\partial \beta_j}, \quad i = 1,\dots,n,\ j = 0,\dots,p, \tag{3.22}$$

with:

- $\frac{\partial l_i(\theta_i)}{\partial \theta_i} = \frac{y_i - b'(\theta_i)}{a(\phi)} = \frac{y_i - \mu_i}{a(\phi)}, \quad i = 1, \ldots, n$
- $\frac{\partial \mu_i(\theta_i)}{\partial \theta_i} = b''(\theta_i) = \frac{\mathbb{V}[Y_i]}{a(\phi)} \Longrightarrow \frac{\partial \theta_i(\mu_i)}{\partial \mu_i} = \frac{a(\phi)}{\mathbb{V}[Y_i]}, \quad i = 1, \ldots, n$
- $\frac{\partial \eta_i(\beta)}{\partial \beta_j} = z_{i(j+1)} \quad i = 1, \ldots, n,\ j = 0, \ldots, p.$

Hence,

$$\frac{\partial l_i(\beta)}{\partial \beta_j} = \frac{y_i - \mu_i}{a(\phi)} \frac{a(\phi)}{\mathbb{V}[Y_i]} \frac{\partial \mu_i}{\partial \eta_i} z_{i(j+1)} \quad i = 1, \ldots, n,\ j = 0, \ldots, p, \tag{3.23}$$

and the likelihood equations are,

$$\sum_{i=1}^{n} \frac{(y_i - \mu_i)\, z_{i(j+1)}}{\mathbb{V}[Y_i]} \frac{\partial \mu_i}{\partial \eta_i} = 0, \quad j = 0, \ldots, p. \tag{3.24}$$

These equations are usually solved using numerical optimization due to the fact they can not be solved analytically except in ordinary linear regression with a *Normal* outcome distribution and the identity *link function*.

#### 3.1.3.1 Fisher Scoring Optimization

In the majority of *Linear Models*, the likelihood equations can not be solved analytically. Turkman and Silva (2000) describe the *Fisher Scoring* method as a numerical iterative procedure to obtain the *maximum likelihood estimators* of the model coefficients, using the update,

$$\beta^{(t+1)} = \beta^{(t)} + \left[ I(\beta^{(t)}) \right]^{-1} s(\beta^{(t)}), \tag{3.25}$$

where $I(\beta) = \mathbb{E}\left[ -\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right] = \mathbb{E}\left[ -\frac{\partial s(\beta)}{\partial \beta} \right]$ is called the *Fisher Information Matrix*, and $s(\beta) = \frac{\partial l(\beta)}{\partial \beta}$ is the *score function* for $\beta$, of components $s_j(\beta)$, $j = 0, \ldots, p$.

### 3.1.4 Fisher Information Matrix

As stated before, $I(\beta) = \mathbb{E}\left[ -\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right] \left( = -\mathbb{E}\left[ \frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right] \right)$ represents the *Fisher Information Matrix*. Thus, to compute this matrix we need to calculate the expected value of the second order derivatives of $l_i(\beta)$, $i = 1, \ldots, n$. To do that, we will first prove that, under regularity condition (3.7), we have,

$$-\mathbb{E}\left[ \frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right] = \mathbb{E}\left[ \left( \frac{\partial l(\beta)}{\partial \beta} \right)^2 \right]. \tag{3.26}$$

Remark that, for any random variable of configuration $\Phi(H_1,\dots,H_m)$, where $\Phi$ is a regular function, we have the following definition,

$$\mathbb{E}_\theta[\Phi(H_1,\dots,H_m)] = \int_{IR}\dots\int_{IR}\Phi(h_1,\dots,h_m)\prod_{k=1}^{m} f_H^\theta(h_k)\,d_{h_1}\dots d_{h_m}, \tag{3.27}$$

where $(H_1,\dots,H_m)$ is a sample of the random variable $H$, *i.e.*, a succession of random variables that are independent and identically distributed with $H$.

Notice that,

$$\begin{aligned}
\mathbb{E}\left[\frac{\partial}{\partial\beta}l(\beta)\right] &\equiv \mathbb{E}_\beta\left[\frac{\partial}{\partial\beta}l(\beta;Y_1,\dots,Y_n)\right]\\
&= \int_{IR}\dots\int_{IR}\frac{\partial}{\partial\beta}\log\left[\prod_{k=1}^{n} f_Y^\beta(y_k)\right]\prod_{k=1}^{n} f_Y^\beta(y_k)\,d_{y_1}\dots d_{y_n}\\
&= \int_{IR}\dots\int_{IR}\frac{\frac{\partial}{\partial\beta}\prod_{k=1}^{n} f_Y^\beta(y_k)}{\prod_{k=1}^{n} f_Y^\beta(y_k)}\prod_{k=1}^{n} f_Y^\beta(y_k)\,d_{y_1}\dots d_{y_n}\\
&= \int_{IR}\dots\int_{IR}\frac{\partial}{\partial\beta}\prod_{k=1}^{n} f_Y^\beta(y_k)\,d_{y_1}\dots d_{y_n}\\
&= \frac{\partial}{\partial\beta}\int_{IR}\dots\int_{IR}\prod_{k=1}^{n} f_Y^\beta(y_k)\,d_{y_1}\dots d_{y_n}\\
&= \frac{\partial}{\partial\beta}\left[\left(\int_{IR} f_Y^\beta(y_1)\,d_{y_1}\right)\dots\left(\int_{IR} f_Y^\beta(y_n)\,d_{y_n}\right)\right]\\
&= \frac{\partial}{\partial\beta}(1\times\dots\times 1) = 0.
\end{aligned} \tag{3.28}$$

Therefore, to prove the equality in (3.26),

$$
\begin{aligned}
0 &= \frac{\partial}{\partial\beta}\left(\mathbb{E}_\beta\left[\frac{\partial}{\partial\beta}l(\beta;Y_1,\ldots,Y_n)\right]\right) \equiv \frac{\partial}{\partial\beta}\left(\mathbb{E}_\beta\left[\frac{\partial}{\partial\beta}l^\beta(Y_1,\ldots,Y_n)\right]\right) \\
&= \frac{\partial}{\partial\beta}\left(\int_{IR}\ldots\int_{IR}\left(\frac{\partial}{\partial\beta}l^\beta(y_1,\ldots,y_n)\right)L^\beta(y_1,\ldots,y_n)\,d_{y_1}\ldots d_{y_n}\right) \\
&= \int_{IR}\ldots\int_{IR}\left(\frac{\partial^2}{\partial\beta^2}l^\beta(y_1,\ldots,y_n)\right)L^\beta(y_1,\ldots,y_n)\,d_{y_1}\ldots d_{y_n} \\
&\qquad + \int_{IR}\ldots\int_{IR}\left(\frac{\partial}{\partial\beta}l^\beta(y_1,\ldots,y_n)\right)\left(\frac{\partial}{\partial\beta}L^\beta(y_1,\ldots,y_n)\right)d_{y_1}\ldots d_{y_n} \\
&= \int_{IR}\ldots\int_{IR}\left(\frac{\partial^2}{\partial\beta^2}l^\beta(y_1,\ldots,y_n)\right)L^\beta(y_1,\ldots,y_n)\,d_{y_1}\ldots d_{y_n} \\
&\qquad + \int_{IR}\ldots\int_{IR}\left(\frac{\partial}{\partial\beta}l^\beta(y_1,\ldots,y_n)\right)\left(\frac{\partial}{\partial\beta}l^\beta(y_1,\ldots,y_n)\right)L^\beta(y_1,\ldots,y_n)\,d_{y_1}\ldots d_{y_n} \\
&= \int_{IR}\ldots\int_{IR}\left(\frac{\partial^2}{\partial\beta^2}l^\beta(y_1,\ldots,y_n)\right)L^\beta(y_1,\ldots,y_n)\,d_{y_1}\ldots d_{y_n} \\
&\qquad + \int_{IR}\ldots\int_{IR}\left(\frac{\partial}{\partial\beta}l^\beta(y_1,\ldots,y_n)\right)^2 L^\beta(y_1,\ldots,y_n)\,d_{y_1}\ldots d_{y_n} \\
&= \mathbb{E}_\beta\left[\frac{\partial^2}{\partial\beta^2}l^\beta(Y_1,\ldots,Y_n)\right] + \mathbb{E}_\beta\left[\left(\frac{\partial}{\partial\beta}l^\beta(Y_1,\ldots,Y_n)\right)^2\right],
\end{aligned}
\tag{3.29}
$$

concluding the demonstration of equality (3.26).

Let us calculate the second derivatives of $l_i(\beta)$, $i = 1,\ldots,n$. Using the equality in (3.26), we know that,

$$
\begin{aligned}
\mathbb{E}\left[-\frac{\partial^2 l_i}{\partial\beta_{j1}\partial\beta_{j2}}\right] &= \mathbb{E}\left[\left(\frac{\partial l_i}{\partial\beta_{j1}}\right)\times\left(\frac{\partial l_i}{\partial\beta_{j2}}\right)\right] \\
&= \mathbb{E}\left[\left(\frac{(Y_i-\mu_i)\,z_{i(j1+1)}}{\mathbb{V}[Y_i]}\,\frac{\partial\mu_i}{\partial\eta_i}\right)\left(\frac{(Y_i-\mu_i)\,z_{i(j2+1)}}{\mathbb{V}[Y_i]}\,\frac{\partial\mu_i}{\partial\eta_i}\right)\right] \\
&= \mathbb{E}\left[\frac{(Y_i-\mu_i)^2\,z_{i(j1+1)}z_{i(j2+1)}}{\mathbb{V}[Y_i]^2}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2\right] \\
&= \frac{z_{i(j1+1)}z_{i(j2+1)}}{\mathbb{V}[Y_i]^2}\,\mathbb{E}\left[(Y_i-\mu_i)^2\right]\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2 \\
&= \frac{z_{i(j1+1)}z_{i(j2+1)}}{\mathbb{V}[Y_i]}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2, \quad i = 1,\ldots,n,\ j1 = 0,\ldots,p,\ j2 = 0,\ldots,p.
\end{aligned}
\tag{3.30}
$$

Thus, the component of index $[j1+1, j2+1]$ of the *Fisher Information Matrix* is,

$$\big[I(\boldsymbol{\beta})\big]_{(j1+1)(j2+1)} = \sum_{i=1}^{n} \mathbb{E}\Big[-\frac{\partial^2 l_i}{\partial\beta_{j1}\partial\beta_{j2}}\Big] = \sum_{i=1}^{n} \frac{z_{i(j1+1)}z_{i(j2+1)}}{\mathbb{V}[Y_i]}\Big(\frac{\partial\mu_i}{\partial\eta_i}\Big)^2 \quad j1=0,\ldots,p,\ j2=0,\ldots,p. \tag{3.31}$$

This signifies that the *Fisher Information Matrix* can also be written in matricial form as (Turkman & Silva, 2000),

$$I(\boldsymbol{\beta}) = \mathbf{Z}_D^T \mathbf{W} \mathbf{Z}_D, \tag{3.32}$$

where $\mathbf{W}$ is a $n \times n$ diagonal matrix $\text{diag}(w_1,\ldots,w_n)$ such that,

$$w_i = \frac{\Big(\frac{\partial\mu_i}{\partial\eta_i}\Big)^2}{\mathbb{V}[Y_i]}, \quad i=1,\ldots,n. \tag{3.33}$$

## 3.2 Multinomial Logistic Regression

### 3.2.1 Multinomial Distribution

The *Multinomial* distribution is a generalization of the *Binomial* distribution (sum of *Bernoulli* variables with 2 categories, such as true/false or yes/no) to $J+1 \in \mathbb{N}$ possible outcomes. This distribution is characterized by a vector of probabilities of class affiliation, $\boldsymbol{\pi} = (p_0, p_1, \ldots, p_J)^T$, where $\sum_{q=0}^{J} p_q = 1$, and $p_q > 0$ represents the probability of success for the $qth$ outcome. If $J=1$ then the distribution is *Binomial*.

Let $Y_k$ constitute the number of times that the $kth$ outcome occurs in a sample of $n$ trials, that is $\sum_{k=0}^{J} Y_k = n$. The joint probability of $\mathbf{y} = (y_0, y_1, \ldots, y_J)^T$ is defined as

$$P[Y_0 = y_0, Y_1 = y_1, \ldots, Y_J = y_J] := \frac{n!}{y_0! \cdot y_1! \cdots y_J!}\, p_0^{y_0} \cdot p_1^{y_1} \cdots p_J^{y_J}. \tag{3.34}$$

#### 3.2.1.1 Exponential Family

We can easily demonstrate that the *Multinomial* distribution of parameters $\boldsymbol{\pi} = (p_0, p_1, \ldots, p_J)^T$ belongs to the *Exponential Family*. Considering the probability mass function in (3.34),

$$\begin{aligned} P[Y_0 = y_0, Y_1 = y_1, \ldots, Y_J = y_J] :&= \frac{n!}{y_0! \cdot y_1! \cdots y_J!} p_0^{y_0} \cdot p_1^{y_1} \cdots p_J^{y_J} \\ &= \exp\left\{\log\left(\frac{n!}{y_0! \cdot y_1! \cdots y_J!} p_0^{y_0} \cdot p_1^{y_1} \cdots p_J^{y_J}\right)\right\} \\ &= \exp\left\{\log\left(\frac{n!}{y_0! \cdot y_1! \cdots y_J!}\right) + \log\left(p_0^{y_0} \cdot p_1^{y_1} \cdots p_J^{y_J}\right)\right\} \\ &= \exp\left\{\log\left(\frac{n!}{y_0! \cdot y_1! \cdots y_J!}\right) + \sum_{k=0}^{J} y_k \log(p_k)\right\} \\ &= \exp\left\{\mathbf{y}^T \log(\boldsymbol{\pi}) + \log\left(\frac{n!}{y_0! \cdot y_1! \cdots y_J!}\right)\right\} \\ &= \exp\left\{\frac{\mathbf{y}^T \boldsymbol{\theta} - b(\boldsymbol{\theta})}{a(\phi)} + c(\mathbf{y}, \phi)\right\}, \end{aligned} \tag{3.35}$$

where

- $\boldsymbol{\theta} = \log(\boldsymbol{\pi}) = \big(\log(p_0), \log(p_1), \ldots, \log(p_J)\big)^T$
- $b(\boldsymbol{\theta}) = 0$
- $a(\phi) = 1$
- $c(\mathbf{y}, \phi) = \log\left(\frac{n!}{y_0! \cdot y_1! \cdots y_J!}\right)$.

### 3.2.2 Multinomial Logistic Regression

*Multinomial Logistic Regression* is the classical statistical procedure used to predict a sample's category placement (or the probability of such), using a dataset containing the information of $n$ previously established samples to which we already know the category affiliation. This way we have an outcome (or response) variable, representative of the class to which each sample belongs to, $Y$, and multiple explanatory variables (or covariates/predictors) $\mathbf{x}_i, i \in 1, \ldots, p$, considered mainly responsible for $Y$'s behavior (Starkweather and Moske, 2011). Unlike the classic *Logistic Regression* model, in which the outcome variable is binary (e.g., presence/absence of a characteristic), the response variable in a *Multinomial Logistic Regression* model can have more than two classes that are coded categorically. Additionally, one of these categories is taken as the reference category, normally, but not necessarily, chosen as being the category with the largest number of associated observations (Y. Wang, 2005). The predictors can be either categorical or continuous.

Suppose that the outcome variable, $Y$, has $J+1$ categories, denoted as $(0, 1, \ldots, J)$. Without loss of generality, let "0" be the reference category value. Assume that we have a collection of

$p$ predictor variables, $\mathbf{x} = (x_1, \ldots, x_p)$. We are interested in estimating the probabilities of each outcome conditioned to a given realization of $\mathbf{x}$. These probabilities can be defined as,

$$\begin{aligned} P[\,Y = 0\,|\,\mathbf{x}\,] &= \frac{1}{1 + e^{g_1(\mathbf{x})} + \cdots + e^{g_J(\mathbf{x})}} \\ P[\,Y = 1\,|\,\mathbf{x}\,] &= \frac{e^{g_1(\mathbf{x})}}{1 + e^{g_1(\mathbf{x})} + \cdots + e^{g_J(\mathbf{x})}} \\ &\vdots \\ P[\,Y = J\,|\,\mathbf{x}\,] &= \frac{e^{g_J(\mathbf{x})}}{1 + e^{g_1(\mathbf{x})} + \cdots + e^{g_J(\mathbf{x})}}, \end{aligned} \tag{3.36}$$

where

$$g_j(\mathbf{x}) = \frac{P[\,Y = j\,|\,\mathbf{x}\,]}{P[\,Y = 0\,|\,\mathbf{x}\,]} = \beta_{j0} + \beta_{j1}\mathrm{x}_1 + \cdots + \beta_{jp}\mathrm{x}_p, \quad j \in 1, \ldots, J, \tag{3.37}$$

represents the *logit* of category $j$ *versus* the reference category (Fagerland et al., 2008).

Consider a sample of $n$ independent observations, denoted as $(\mathbf{y}_i, \mathbf{x}_i)$, $i = 1, \ldots, n$, where $\mathbf{y}_i$ is a $(J+1) \times 1$ vector of indicators for the observed response category for observation $i$ (*i.e.*, for each $i$, $\mathbf{y}_i$ is a sequence of $J$ 0's and a 1 placed on the observed category), with the corresponding $(J+1) \times 1$ vector of probabilities $\mathbf{P}_i = (\pi_{i0}, \pi_{i1}, \ldots, \pi_{iJ})^T$, $i = 1, \ldots, n$.

#### 3.2.2.1 Estimation of the Model Coefficients

Considering $\mathbf{Z}_D$ as the *Design Matrix* of the model, define $\mathbf{Z}^T = \mathbf{Z}_D^T \otimes \mathbf{I}_J$, where $\mathbf{I}_J$ denotes the $J \times J$ identity matrix, and $\otimes$ indicates the *Kronecker Product*, characterized as, for $\mathbf{A} = \{a_{ij}\}$, $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$ an $m \times n$ matrix and $\mathbf{B}$ an $p \times q$ matrix,

$$\mathbf{A} \otimes \mathbf{B} := \begin{bmatrix} a_{11}\mathbf{B} & \ldots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \ldots & a_{mn}\mathbf{B} \end{bmatrix}. \tag{3.38}$$

Notice that $\mathbf{A} \otimes \mathbf{B}$ is a $pm \times qn$ block matrix.

The authors Bull et al. (2002) show that the $(p+1)J \times (p+1)J$ *Fisher Information Matrix* of a *Multinomial* problem, $I$, can be defined as,

$$I = (\mathbf{Z}^T \mathbf{M} \mathbf{Z}), \tag{3.39}$$

where $\mathbf{M}$ is an $nJ \times nJ$ block diagonal matrix with $n$ $J \times J$ blocks: $\mathbf{M}_i = \{m_{ic1c2}\}$, $m_{ic1c1} = \pi_{ic1}(1 - \pi_{ic1})$, for $c1 = c2$, and $m_{ic1c2} = -\pi_{ic1}\pi_{ic2}$, otherwise, for $i = 1, \ldots, n$, $c1 = 1, \ldots, J$, $c2 = 1, \ldots, J$.

Typically, the *Maximum Likelihood Estimators* of the model parameters, $\hat{\mathbf{B}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)^T$, are calculated using the data from the observations $(\mathbf{y}_i, \mathbf{x}_i)$, $i \in \{1, \ldots, n\}$, by solving the score equations of the *log-likelihood*. Because these equations can not be solved analytically except in ordinary *linear regression* with *Normal* distribution and identity *link function*, we resort to numerical optimization methods.

The *Fisher Scoring Optimization* method updates equation used to obtain the *Maximum Likelihood Estimators*, for the $tth$ iteration as,

$$\mathbf{B}_{(t+1)} = \mathbf{B}_{(t)} + I_{(t)}^{-1} U(\mathbf{B}_{(t)}), \tag{3.40}$$

where $U(\mathbf{B}_{(t)})$ is the *score function* for the *Multinomial* model for iteration $t$ (Bull et al., 2002).

#### 3.2.2.2 High-Dimensionality and Multinomial Logistic Regression

It is clear that, in order to resort to the *Fisher Scoring Optimization* update, at each iteration, the *Fisher Information Matrix* needs to be non-singular. Notice that, in order for $I$ to be non-singular,

$$rank(I) = (p+1)J. \tag{3.41}$$

Let us now assume that the problem is high-dimensional, *i.e.*, $p > n$. Consider the following properties of the matrix rank,

- $rank(A) \leq min\{n, p\}$, for $A$ a $n \times p$ matrix
- $rank(A) = rank(A^T)$, for $A$ matrix
- $rank(AB) \leq min\{rank(A), rank(B)\}$, for $A$ and $B$ matrices

This way, considering (3.39),

$$\begin{aligned} rank(I) &= rank(\mathbf{Z}^T\mathbf{M}\mathbf{Z}) \\ &\leq min\{rank(\mathbf{Z}^T)\,;\, rank(\mathbf{M})\,;\, rank(\mathbf{Z})\} \\ &= min\{rank(\mathbf{Z})\,;\, rank(\mathbf{M})\}, \end{aligned} \tag{3.42}$$

Furthermore, since $n < p$,

$$rank(\mathbf{Z}) \leq min\{(p+1)J\,;\, nJ\} = nJ. \tag{3.43}$$

Since $\mathbf{M}$ is a square $nJ \times nJ$ matrix,

$$rank(\mathbf{M}) \leq nJ. \tag{3.44}$$

Combining the results in (3.42), (3.43) and (3.44), we can now easily conclude that,

$$rank(I) \leq nJ < pJ < (p+1)J. \tag{3.45}$$

Concluding that the *Fisher Information Matrix* is singular when $p > n$. This signifies that, when a problem is high-dimensional, we can not resort to the *Fisher Scoring Optimization* numerical method to attain the *Maximum Likelihood Estimators*. For this reason, the high-dimensionality of a *Multinomial* model calls for the utilization of regularization methods.

### 3.2.3 Predictive Quality

#### 3.2.3.1 Confusion Matrix

A *Confusion Matrix* is a tabular way of analyzing a model's predictive performance. Each entry of this matrix indicates the number of predictions made by the model, and whether the classes were classified correctly or incorrectly. The most conventional form of *Confusion Matrix* results from the binary problem, *i.e.*, *Logistic Regression*, displayed in Table 3.1. Because the classification problem is binary, there are only two classes to classify, a positive and a negative class.

Table 3.1: Confusion Matrix for the classic *Logistic Regression* problem.

| | | **True Class** | | |
|---|---|---|---|---|
| | | **positive** | **negative** | **total** |
| **Predicted Class** | **positive** | True Positive (*TP*) | False Positive (*FP*) | $P'$ |
| | **negative** | False Negative (*FN*) | True Negative (*TN*) | $N'$ |
| | **total** | $P$ | $N$ | |

- **True Positive** (*TP*) refers to the number of predictions where the model correctly predicted the positive class as positive.
- **True Negative** (*TN*) refers to the number of predictions where the model correctly predicted the negative class as negative.
- **False Positive** (*FP*) refers to the number of predictions where the model incorrectly predicted the negative class as positive.
- **False Negative** (*FN*) refers to the number of predictions where the model incorrectly predicted the positive class as negative.

Upon establishing the *Confusion Matrix*, we can can determine a few performance measures. The most common performance measures include:

- ***Accuracy*:** Fraction of samples that were correctly classified by the model,

$$\frac{TP+TN}{TP+TN+FP+FN}. \tag{3.46}$$

- ***Missclassification Rate*:** Fraction of incorrect predictions,

$$\frac{FP+FN}{TP+TN+FP+FN} \quad \text{or} \quad 1-\text{Accuracy}. \tag{3.47}$$

- ***Precision*:** Fraction of positive class predictions that were actually positive,

$$\frac{TP}{TP+FP}. \tag{3.48}$$

  If, for a certain class, $TP+FP=0$, then we consider Precision = 1, since the model did not actually fail to predict the class.

- ***Recall* / *True Positive Rate* / *Sensitivity* / *Probability of Detection*:** Fraction of all positive samples that the model correctly predicted as positive,

$$\frac{TP}{TP+FN}. \tag{3.49}$$

  Once more, if, for a certain class, $TP+FN=0$, then we consider Recall = 1, since the model did not actually fail to predict the class.

- ***Specificity* / *True Negative Rate*:** Fraction of all negative samples are were correctly predicted as negative,

$$\frac{TN}{FP+TN}. \tag{3.50}$$

  In this case, if, for a certain class, $FP+TN=0$, then we consider Specificity = 1, since the model did not actually fail to predict any negative samples.

- ***F1-Score*:** Merging *Precision* and *Recall* into a single measure, it is, mathematically, the harmonic mean of *Precision* and *Recall*,

$$2\times\frac{Precision\times Recall}{Precision+Recall}=\frac{2TP}{2TP+FP+FN}. \tag{3.51}$$

  If $2TP+FP+FN=0$ for a particular class, signifying that no sample from that class was selected to the testing set, we consider F1-Score = 1, since the model did not actually fail to predict the class.

Unlike binary classification, there are no positive/negative classes in problems where the response variable has more than two classes - multi-class outcome - such as in a *Multinomial Regression* problem. The reasoning behind multi-class confusion matrices is the consideration of **TP**, **TN**, **FP** and **FN** for each individual class. Consider the following example: a model aims to predict the outcome of a certain phenomenon that can fall under three different categories: A, B and C. The model's predictive performance is presented through the *Confusion Matrix* displayed in Table 3.2.

Table 3.2: Confusion matrix of a 3-class classification problem

| | | **True Class** | | | |
|---|---|---|---|---|---|
| | | **A** | **B** | **C** | **total** |
| **Predicted Class** | **A** | True A<br>$TP_A$ | False A<br>$F_{A(B)}$ | False A<br>$F_{A(C)}$ | $T_A$' |
| | **B** | False B<br>$F_{B(A)}$ | True B<br>$TP_B$ | False B<br>$F_{B(C)}$ | $T_B$' |
| | **C** | False C<br>$F_{C(A)}$ | False C<br>$F_{C(B)}$ | True C<br>$TP_C$ | $T_C$' |
| | **total** | $T_A$ | $T_B$ | $T_C$ | |

Table 3.3: Confusion matrix metrics of a 3-class classification problem

| | A | B | C |
|---|---|---|---|
| **True Positive** | $TP_A$ | $TP_B$ | $TP_C$ |
| **True Negative** | $TN_A = TP_B + F_{B(C)} + F_{C(B)} + TP_C$ | $TN_B = TP_C + F_{A(C)} + F_{C(A)} + TP_C$ | $TN_C = TP_C + F_{A(B)} + F_{B(A)} + TP_B$ |
| **False Positive** | $FP_A = F_{A(B)} + F_{A(C)}$ | $FP_B = F_{B(A)} + F_{B(C)}$ | $FP_C = F_{C(A)} + F_{C(B)}$ |
| **False Negative** | $FN_A = F_{B(A)} + F_{C(A)}$ | $FN_B = F_{A(B)} + F_{C(B)}$ | $FN_C = F_{A(C)} + F_{B(C)}$ |

Table 3.3 showcases the formulas for the metrics of a 3-class classification problem.

Using these metrics, we can then calculate the performance measures for each class, simply applying the formulas in (3.46), (3.47), (3.48), (3.49), (3.50) and (3.51). There are a few performance measures for studying the overall model instead of considering the performance for each individual class.

Considering that,

- **Total True Positives**: $Total\ TP = TP_A + TP_B + TP_C$
- **Total False Positives**: $Total\ FP\ FP_A + FP_B + FP_C$
- **Total True Negatives**: $Total\ TN\ TN_A + TN_B + TN_C$
- **Total False Negatives**: $Total\ FN\ FN_A + FN_B + FN_C$,

the overall performance measures include,

- ***Total Precision*:**

$$\frac{Total\ TP}{Total\ TP + Total\ FP} \tag{3.52}$$

- ***Total Recall*:**

$$\frac{Total\ TP}{Total\ TP + Total\ FN} \tag{3.53}$$

- ***Micro F1*:** Assesses the quality of multi-classification problems. Simply put, it measures the *F1-score* of the aggregated contributions of all classes.

$$2 \times \frac{Total\ Precision \times Total\ Recall}{Total\ Precision + Total\ Recall} \tag{3.54}$$

- ***Macro F1*:** Calculates the *F1-score* metrics for each class individually and then takes unweighted mean of the measures.

$$\frac{F1\text{-}Score_A + F1\text{-}Score_B + F1\text{-}Score_C}{3}, \tag{3.55}$$

where

$$F1\text{-}Score_k = \frac{2TP_k}{2TP_k + FP_k + FN_k} \tag{3.56}$$

- ***Weighted F1*:** It takes the weighted mean of the measures. The weights for each class are the total number of samples of that class.

$$\frac{T_A \times F1\text{-}Score_A + T_B \times F1\text{-}Score_B + T_C \times F1\text{-}Score_C}{T_A + T_B + T_C} \tag{3.57}$$

#### 3.2.3.2 Cross Entropy

Being one of the major metrics to assess the performance of a classification problem, *Cross Entropy*, or *Log-Loss*, measures the performance of a classification model whose output is probabilities of class affiliation (between 0 and 1). Conceptually, the *Cross Entropy* metric is an indicative of how close the predictive probability of an observation was to predicting the actual class. *Cross Entropy* increases as the predicted probability diverges from predicting the true label. Mathematically, the *Log-Loss* function (*LL*) of a multi-classification model is defined as,

$$LL = -\frac{1}{n}\sum_{i=1}^{n}\sum_{q=0}^{J} a_{iq}\log(p_{iq}), \tag{3.58}$$

where $n$ is the number of samples, $J+1$ is the number of classes that the response variable assumes, $a_{iq}$ is a binary indicator of whether or not label $q$ is the correct classification for instance $i$, and $p_{iq}$ is the model's predicted probability of assigning label $q$ to instance $i$.

#### 3.2.3.3 ROC Curve and AUC Score

A ROC *Curve* (*Receiver Operating Characteristic Curve*) is a probability curve that illustrates the performance of a binary classification model, as its discrimination threshold is varied. It is constructed on the basis of the following two parameters that derive from the model's *Confusion Matrix*,

- ***True Positive Rate*** (**TPR**): ratio of observations that were correctly predicted to be positive out of all the actual positive observations, that is,

$$TPR = \frac{TP}{TP+FN}. \tag{3.59}$$

  Also known as *Sensitivity*, *Recall* or *Probability of Detection*.

- ***False Positive Rate*** (**FPR**): proportion of observations that were incorrectly predicted to be positive out of all the actual negative observations, that is,

$$FPR = \frac{FP}{FP+TN}. \tag{3.60}$$

  Also known as *Probability of False Alarm*.

For class prediction models that give a probability that reflects the degree to which an instance belongs to one class rather than the other, a class selection threshold is established. This way, after estimating the class affiliation probabilities for a certain instance, the model will associate it to the respective class, taking into account that threshold value. Consequently, we can create a curve that reflects *True Positive Rate* against the *False Positive Rate* by varying the discrimination threshold values (Mandrekar, 2010). Assuming *False Positive Rate* on the *x-axis* and *True Positive Rate* on the *y-axis*, the closest the curve is to the top left corner of this orthonormal

axis, the better the classifier's performance, meaning that it produces very low false positives and high true positives.

To summarize the performance of the model into a single measure, one common procedure is to calculate the area under the ROC *Curve*, known as ROC-AUC *Score*. This measures the ability of the classifier to distinguish between classes. As the ROC-AUC *Score* approaches 1, it is more likely that the model is correctly distinguishing between positive and negative classes. In contrast, when the AUC approaches 0, it signifies that the model is predicting negative classes as positive and vice-versa. If AUC $\approx 0.5$ then, presumably, the model does not have class separation capacity.

Given a *Multinomial Regression* model, where the outcome has more than 2 categories, we can study multiple ROC *Curves* considering one class at a time. To accomplish this, for each category of the response variable, we take the class in study as the *positive class* and the remaining classes as the *negative class*, transforming the *Multinomial* problem into multiple binary problems. As a result, instead of a measure of performance of the overall model, we compute measures for each individual class of the outcome.

### 3.2.4 Measures of variable importance

Measures of variable importance are used to assess the weight that each predictor has on the prediction of the response variable. On condition that the manipulated data is standardized before fitting the model, one primary measure of variable importance is the absolute values of the model coefficients. Since *Multinomial Regression* models return a different variable coefficient for each of the response variable's classes, we can measure variable importance ($VI$), for each predictor variable, as the sum of the coefficient's absolute value for each class of the response. Theoretically, for the predictor variable $x_k$, $k \in \{1,\dots,p\}$, we define its importance, $VI_k$, as,

$$VI_k = \sum_{q=0}^{J} |\beta_{qk}| \quad k \in \{1,\dots,p\}. \tag{3.61}$$

### 3.2.5 Residual Analysis

*Residual Analysis* plays an important role in the validation of statistical models. For a given instance, a residual is a measure of the difference between the observed value of the outcome variable and the value predicted by the model.

In *Multinomial Regression*, since the outcome is categorical, we can undertake a residual as a measure of how close the estimated probability is to predicting the actual class. Consider, for example, a 3-class response variable (class A, class B, class C), and a train observation that we know beforehand belongs to class B. We define the residual for that observation as 1 minus the model's fitted probability of belonging to class B. As consequence, the multinomial residuals belong to the interval $[0,1]$. The best fitted observations will have residual values close to 0. In

theory, the multinomial residual for each instance, $r_i$, $i \in \{1,\dots,n_t\}$ where $n_t$ is the number of observations in the *training set*, can be defined as,

$$r_i = 1 - \pi_{iq}, \quad i \in \{1,\dots,n_t\}, \tag{3.62}$$

where $q \in \{0,\dots,J\}$ is the correct class for observation $i$.

## 3.3 Cross-Validation

*Cross-Validation* is a model validation technique for assessing how the results of a statistical analysis will generalize to a new independent set of data. This procedure is mainly used in prediction modeling, where the user wants to estimate how accurately their model will perform in practice. Firstly, the data is partitioned into two groups: a *training set*, that is composed of the data instances used to fit the model, and a *testing set*, used for validation of the fitted model as a means to check its performance. This way, the purpose of *Cross-Validation* is to test the model's ability to predict the studied phenomenon's behavior on a new set of data.

### 3.3.1 K-fold Cross-Validation

The *Cross-Validation goodness of fit* approach becomes a more useful tool when ran multiple times, formulating an iterative method known as *K-fold Cross-Validation*. When $K = 1$, *1-fold Cross-Validation*, consists of simply partitioning the data into the two complementary sets: *training* and *testing*, fitting the model with the *training set* and testing the predictive quality of the model on the *testing set*. For more statistically significant results, multiple rounds ($K > 1$) of *Cross-Validation* can be executed. The dataset, $S$, is partitioned into $K$ complementary subsets, $S_t$, $t \in \{1,\dots,K\}$, such that

$$S = \bigcup_{t=1}^{K} S_t. \tag{3.63}$$

At each iteration $u \in \{1,\dots,K\}$, $S_u$ is used as the *testing set* and the remaining $S_r$, $r \in \{1,\dots,K\}$, $r \neq u$ compose the *training set*, $T_u$, *i.e.*,

$$T_u = \bigcup_{r=1, r\neq u}^{K} S_r. \tag{3.64}$$

Thereafter, the validation results for each iteration can be combined (for example, averaged) to postulate about an estimate of the overall model's predictive performance.

One limitation of this validation method is the fact that it can only be executed within the span of $K$ iterations, making it dependent on the size of the dataset. Hence, if the size of the dataset is small, *K-fold Cross-Validation* might not be an appropriate method of attesting the model's performance. This method is known for being unbiased but conducting high variance results.

### 3.3.2 Monte Carlo Cross-Validation

Also known as *Repeated Random Sub-Sampling Validation*, the premise of this method is very similar to that of *K-fold Cross-Validation*. It too is an iterative method that aims to combine the results of each iteration to postulate about the overall model's performance. The main difference between the two methods is that, instead of dividing the original dataset into $K$ complementary partitions, at each iteration, *Monte Carlo Cross-Validation* creates a random split of the dataset into *training* and *testing sets* (Xu & Liang, 2001). For this reason, unlike *K-fold Cross-Validation*, some observations may never be selected to a *testing set*, whereas others may be selected more than once throughout the different iterations of this process. To overcome *K-fold*'s limitation, since this method is not dependent on the sample size, it can bet repeated a very high number of times. It is remarked as being a biased method, however leading to low variance results.

## 3.4 Regularization Methods

A problem is termed *high-dimensional* when the number of predictor variables, $p$, is larger than the number of data instances/observations, $n$, *i.e.*, $p > n$. Regularization approaches such as *Ridge*, LASSO , and *Elastic Net* have become the methods of choice for analyzing high-dimensional data.

Regularization methods rely on *sparsity assumptions*, guaranteeing that, withing the $p$ features, only a limited portion can be selected as significant to predict the response's behavior. These assumptions often fall under one of the following hypothesis:

- Among the $p$ explanatory variables, only a small number of them are relevant to predict the response variable's behavior;
- Although all of the $p$ explanatory variables are important, one can partition the space so that, in any local region, only a small number are relevant;
- Although all of the $p$ explanatory variables are important, one can find a small number of linear combinations of those variables that explain most of the variability in the response.

Aforesaid, high-dimensional *Linear Modeling* is challenging towards the estimation of the model parameters. Regularization methods that impose a limitation (penalty) on the amount of variables allowed to predict the demeanor of a given phenomenon are, therefore, a popular way to overcome the issues that emerge from these complex problems.

In each of the discussed regularization methods, the intercept coefficient is left out of the penalty term, as it portrays the origin of the response variable, not being linked to any predictor. It is convenient to center the explanatory variables, and estimating the intercept as the mean value of the response when the predictor variables in the model are evaluated at zero (Friedman et al., 2010).

Without loss of generality, assume that the data is standardized, *i.e.*,

$$\sum_{i=1}^{n} x_{ik} = 0 \quad \text{and} \quad \frac{1}{n}\sum_{i=1}^{n} x_{ik}^2 = 1, \quad \text{for} \quad k = 1, \ldots, p. \tag{3.65}$$

### 3.4.1 *Ridge* Method

To address multicollinearity in *Linear Modeling*, *Ridge* was proposed by Hoerl and Kennard (1970), to prevent the poor estimation and high variance of the model's coefficients (Ogutu et al., 2012). *Multicollinearity* is the phenomenon in which some predictors share a linear relationship with each other. This issue often occurs in models with large numbers of variables, as it is more likely for groups of predictors to be highly correlated (Hilt & Seegrist, 1977). This method regulates the dimensionality of a problem by shrinking the coefficients of the less relevant features towards 0, but never actually removing features from the model. In order to address groups of highly correlated variables, *Ridge* approximates their coefficients towards each other, allowing them to exchange strength (Friedman et al., 2010). This method has been shown to achieve good performance measures, however, considering that it does not perform feature selection, its dense solution makes the method inconvenient when solving high-dimensional problems (Aseervatham et al., 2011).

Let $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ and $\tilde{\boldsymbol{\beta}} = (\beta_1, \ldots, \beta_p)^T$ be the vector of model coefficients excluding the intercept component. In the linear case, the *Ridge* method estimates the model coefficients, $\hat{\boldsymbol{\beta}}^R$, resorting to the $\ell_2$ penalization in order to achieve the ideal fitted model,

$$\hat{\boldsymbol{\beta}}^R = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, ||\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}||_2^2 + \lambda ||\tilde{\boldsymbol{\beta}}||_2^2, \tag{3.66}$$

where $||\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}||_2^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \mathbf{x}_i^T \boldsymbol{\beta})^2$ is the $\ell_2$-norm (quadratic) loss function, $\mathbf{x}_i^T$ is the $ith$ row of the model's design matrix $\mathbf{X}$, $||\tilde{\boldsymbol{\beta}}||_2^2 = \sum_{j=1}^{p} \beta_j^2$ is the $\ell_2$-norm penalty on $\tilde{\boldsymbol{\beta}}$, and $\lambda > 0$ is *penalization parameter* that regulates the strength of the penalty. The higher the value of $\lambda$, the greater is the amount of shrinkage.

### 3.4.2 LASSO Method

Proposed by Tibshirani (1996), the LASSO (*Least Absolute Shrinkage and Selection Operator*) method has proven to be a flexible tool that performs both variable selection and coefficient shrinkage, *i.e.*, shrinking some of the model coefficients and setting others to zero. This method focuses on the minimization of the sum of residual squares, for which the sum of the absolute values of the model coefficients is not larger than a certain threshold. Essentially, this results in the method removing irrelevant predictors from the model by setting their corresponding coefficients to zero, thereby making the model simpler and easier to interpret.

LASSO is mainly used in problems that contain a lot of ineffectual variables, which oftentimes happens with high-dimensional datasets. One of its main downfalls is the inefficiency

of addressing groups of highly correlated predictors. In such circumstances, for each group of correlated variables, LASSO tends to arbitrarily select one predictor into the model, discarding the others (Friedman et al., 2010). An additional drawback of the LASSO is not being able to select more variables than the sample size before it saturates, when $p > n$ (Ogutu et al., 2012).

To estimate the model coefficients, $\hat{\beta}^L$, in the linear case, the LASSO method resorts to the $\ell_1$ penalized least squares criterion and obtains the sparse solution to the optimization problem,

$$\hat{\beta}^L = \underset{\beta}{\operatorname{argmin}} \, ||\mathbf{y} - \mathbf{Z}\beta||_2^2 + \lambda ||\tilde{\beta}||_1, \tag{3.67}$$

where $||\tilde{\beta}||_1 = \sum_{j=1}^{p} |\beta_j|$ is the $\ell_1$-norm penalty on $\tilde{\beta}$, which incites sparsity in the solution. $\lambda$ is the *penalization parameter* that controls the degree to which the coefficients are penalized.

The $\ell_1$-norm allows this method to simultaneously regularize the least squares while shrinking some components of $\hat{\beta}^L$ to zero, for some suitably chosen $\lambda$.

### 3.4.3 *Elastic Net* Method

Zou and Hastie (2005) proposed the *Elastic Net* as a *Feature Selection Method* that out-preforms LASSO . Unlike LASSO , this method is robust in the eventuality of highly correlated predictors (Friedman et al., 2010). Similarly to *Ridge* and LASSO, the *Elastic Net* initiates with least squares. Then, it combines the LASSO's penalty with *Ridge*'s, using a mixture of $\ell_1$ (LASSO) and $\ell_2$ (*Ridge*) penalties in order to estimate the model coefficients, $\hat{\beta}^{EN}$, in the linear case, as,

$$\hat{\beta}^{EN} = \left(1 + \frac{\lambda_2}{n}\right)\left\{ \underset{\beta}{\operatorname{argmin}} \, ||\mathbf{y} - \mathbf{Z}\beta||_2^2 + \lambda_2 ||\tilde{\beta}||_2^2 + \lambda_1 ||\tilde{\beta}||_1 \right\} \tag{3.68}$$

Establishing a new parameter, denoted *mixing parameter*, as

$$\alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}, \tag{3.69}$$

then, because of the *Lagrangian* form of optimization problems, the estimation of the model coefficients in (3.68) is equally obtained by solving,

$$\hat{\beta}^{EN} = \underset{\beta}{\operatorname{argmin}} \, ||\mathbf{y} - \mathbf{Z}\beta||_2^2 \quad \text{subject to} \quad P_\alpha(\tilde{\beta}) = \alpha ||\tilde{\beta}||_2^2 + (1-\alpha) ||\tilde{\beta}||_1 \leq s \quad \text{for some } s, \tag{3.70}$$

where $P_\alpha(\tilde{\beta})$ is the *Elastic Net* penalty, which is a convex combination of *Ridge* and LASSO's penalties.

Recall that, while the *Ridge* penalty shrinks the coefficients of correlated predictors towards each other, LASSO tends to pick one predictor from each group of correlated variables and discard the others. The *Elastic Net* penalty mixes these two concepts in a *grouping effect*. In the event of a group of correlated predictors, a mixing parameter of $\alpha = 0.5$ tends to either select or leave out the entire group of features (Hastie et al., 2016). If $\alpha = \delta$, for small $\delta > 0$, then this method performs much like the LASSO, but removes any complications caused by high

correlations between variables (Friedman et al., 2010). In fact, $P_\alpha(\tilde{\beta})$ creates a compromise between *Ridge* and LASSO. When $\alpha = 1$, it simplifies to the *Ridge*, and when $\alpha = 0$ to the LASSO. The $\ell_1$ component does automatic variable selection, while the $\ell_2$ component encourages grouped selection, taking into account the correlation between predictor variables.

Unlike the LASSO, the *Elastic Net* can select more than $n$ explanatory variables when $p > n$ (Ogutu et al., 2012).

### 3.4.4 Penalization and mixing parameters

In a regularized model, the *penalization parameter* is responsible for controlling the model's penalization. The choice of this parameter is vital, since it has a big influence on the amount of features that the method selects. As this non-negative threshold increases, the regularization effect is strengthened, and the purpose of the model develops into keeping the coefficients small instead of minimizing the loss function. The *mixing parameter* controls the proximity of the *Elastic Net* penalty to *Ridge* and LASSO's penalty. *K-fold Cross-Validation* is one of the most common methods for tuning these parameters (see section 3.3.1). It chooses the values with the smallest cross-validated score, *i.e.*, the *penalization* and *mixing parameters* that lead to the model with the smallest error (Jung, 2016). When addressing a classification problem, this method selects the parameters that result in the model with the best prediction performance.

Firstly, as described in section 3.3.1, the data is partitioned into *training* and *testing sets*. Since the *training set* is used for fitting the model, the regularization methods are applied to this portion of the data. The prediction error is evaluated by examining the data under control: the *testing set*. Smaller values of prediction error are expected to indicate better models with higher chance to correctly generalizing to any new set of data. The parameters that lead to the minimum error, if it exists, are considered optimal values of the *penalization* and *mixing parameters* (Obuchi & Kabashima, 2016).

# 4

# APPLICATION

As a result of the growing globalization of seafood markets, traceability has become paramount to safeguard food safety. With consumers being increasingly aware of the potential hazards that contaminated seafood may cause to their health, ensuring that the geographic origin of seafood is not mislabeled is a first step to fight fraudulent practices that aim to cover-up illegal fishing (Astill et al., 2019, Leal et al., 2015, Bennion et al., 2021).

Figure 4.1: *Ruditapes philippinarum*, Gulf of Morbihan, S. Brittany, NW. France, accessed 4 June 2021, <http://www.idscaro.net/sci/04_med/class/fam5/species/ruditapes_phil1.htm>

*Ruditapes philippinarum*, commonly known as Manila clam, is a marine bivalve that is commercially harvested for human consumption, being one of the most important bivalve species grown in aquaculture worldwide (Mamede, Ricardo, Abreu, et al., 2021, Mamede, Ricardo, Gonçalves, et al., 2021, Bennion et al., 2019, Morrison et al., 2019). The place of origin of seafood can be predicted by modeling features such as their biochemical and geochemical fingerprints.

In this chapter, we exploit a dataset retrieved from 30 Manila clam samples, detailing information on 44 composition features, with the purpose of identifying the feature's distinguishing capabilities between three geographic origins: Ria de Vigo (Galiza, Spain), Ria de Aveiro (Aveiro, Portugal), Estuário do Tejo (Lisbon, Portugal), i.e, a classical *Multinomial Logistic Regression* problem. However, given the high-dimensionality of the dataset (number of variables higher than number of observations), the estimation of the model coefficients posses difficulties. To overcome this problem, we applied *Ridge*, LASSO and *Elastic Net* methods to model the origin of the clams. Additionally, since datasets of only 30 samples challenge the process of model validation, the re-sampling method of *Monte Carlo Cross-Validation* was also implemented to establish different *training* and *testing sets* for model implementation and validation.

All of the employed methods were implemented using the software `R` (R Core Team, 2021). *Ridge*, LASSO and *Elastic Net* were employed, through the `glmnet` package (Simon et al., 2011), and with the assistance of `ensr` (DeWitt, 2019) and `glmnetUtils` (Ooi, 2021) packages for displaying the estimated values produced by each regularized model. Most of the displayed graphics were produced using the package `ggplot2` (Wickham, 2016).

## 4.1 Data Description

The dataset includes an outcome variable portraying the geographic origin of the 30 samples of clams, *i.e.*, a 3 class categorical variable with the levels: `G` (Ria de Vigo located in Galiza, Spain; 8° 43' 9.59"W, 42° 15' 38.44"N), `Rav` (Ria de Aveiro in Aveiro, Portugal; 8° 41' 18.93"W, 40° 46' 6.95"N), and `T` (Estuário do Tejo in Lisbon, Portugal; 9° 0' 58.66"W, 38° 45' 16.55"N). The predictors were 44 continuous variables: 26 concerning the quantification of the clams' adductor mussel fatty acid composition (biochemical fingerprint), and the remaining 18 regard the ratios of chemical elements to Calcium concentrations of their shell (geochemical fingerprint). In this dataset, the biochemical and geochemical fingerprints of the Manila clam regard the fatty acids and chemical components displayed in Tables 4.1 and 4.2, respectively.

The exploited data accommodated 10 Manila clams from Ria de Vigo, 10 from Ria de Aveiro and 10 from Estuário do Tejo. Prior to implementing the regularization methods, we first performed an exploratory data analysis, where we were able to acquire useful information about the data, such as the variability of the quantification of each feature considering the different locations, and the presence of groups of correlated variables.

Figures 4.2 and 4.3 display an ensemble of *boxplots* that demonstrate how the three different locations, although possessing the exact same species of clam, lead to clear distinct biochemical and geochemical fingerprints. With this analysis we aim to infer that, in fact, we can differentiate the three geographic origins in terms of their clams' biochemical and geochemical fingerprints. If that is the case, fitting a *Generalized Linear Model* to this dataset to predict the location of origin of Manila clams seems appropriate.

Considering Figure 4.2, it is evident that the quantification of the majority of features differs significantly in each of the considered locations of harvest. For example, fatty acids `FA14:0`, `FA16:1n_7`, `FA18:1n_7`, `FA20:1n_7`, `FA20:4n_3`, `FA20:5n_3`, `FA22:3n_6`, `FA22:4n_6` and

Table 4.1: Fatty acid profile (FA) of the adductor muscle of *Ruditapes philippinarum* used to model the location of origin

| Fatty Acid Denomination | Fatty Acid | Fatty Acid Denomination | Fatty Acid |
|---|---|---|---|
| FA14:0 | $14:0$ | FA20:1n_7 | $20:1n-7$ |
| FA15:0 | $15:0$ | FA20:2n_6 | $20:2n-6$ |
| FA16:0 | $16:0$ | FA20:3n_6 | $20:3n-6$ |
| FA16:1n_9 | $16:1n-9$ | FA20:4n_6 | $20:4n-6$ |
| FA16:1n_7 | $16:1n-7$ | FA20:4n_3 | $20:4n-3$ |
| FA17:0 | $17:0$ | FA20:5n_3 | $20:5n-3$ |
| FA18:0 | $18:0$ | FA22:2n_9 | $22:2n-9$ |
| FA18:1n_9 | $18:1n-9$ | FA22:2n_6 | $22:2n-6$ |
| FA18:1n_7 | $18:1n-7$ | FA22:3n_6 | $22:3n-6$ |
| FA18:2n_6 | $18:2n-6$ | FA22:4n_6 | $22:4n-6$ |
| FA18:3n_3 | $18:3n-3$ | FA22:5n_6 | $22:5n-6$ |
| FA18:4n_3 | $18:4n-3$ | FA22:5n_3 | $22:5n-3$ |
| FA20:1n_9_11 | $20:1n-9-11$ | FA22:6n_3 | $22:6n-3$ |

Table 4.2: Chemical features used to model the location of origin of *Ruditapes philippinarum*

| Element Symbol | Element Name | Element Symbol | Element Name |
|---|---|---|---|
| Na | Sodium | Zn | Zinc |
| Mg | Magnesium | Sr | Strontium |
| Al | Aluminum | Y | Yttrium |
| P | Phosphorus | Ba | Barium |
| Mn | Manganese | La | Lanthanum |
| Fe | Iron | Ce | Cerium |
| Co | Cobalt | Nd | Neodymium |
| Ni | Nickel | Gd | Gadolinium |
| Cu | Copper | U | Uranium |

FA22:5n_3 exhibit clear higher levels when harvested from Ria de Vigo (G), while fatty acids FA15:0, FA17:0, FA18:1n_9, FA18:2n_6, FA18:4n_3, FA20:1n_9_11, FA20:4n_6, FA22:2n_9, FA22:5n_6 and FA22:6n_3 have the lowest levels when found in this same location. Levels of FA18:0, FA22:2n_9 and FA22:6n_3 are clearly higher when the clams are harvested from Ria de Aveiro (Rav), although fatty acids FA16:1n_7, FA18:1n_7, FA18:3n_3, FA20:1n_7, FA20:2n_6, FA20:3n_6, FA20:4n_3, FA20:5n_3, FA22:3n_6 and FA22:5n_3 are lowest on this same region. Estuário do Tejo (T) endures the clams with higher levels of fatty acids FA18:2n_6, FA18:3n_3, FA18:4n_3 and FA22:5n_6, and the lowest levels of FA14:0. Additionally, notice how clams harvested in Ria de Vigo have a much lower range of FA16:0 values compared to the clams harvested in the other two locations. The same happens with the range of levels of FA20:4n_6, FA22:2n_9 and FA22:5n_6.

Considering Figure 4.3, once more note how different locations of harvest conduct clams with significantly different geochemical fingerprints. The shells of clams harvested from Ria de Vigo (G) clearly reveal lower levels of sodium, magnesium and aluminum, along with higher quantities of strontium, yttrium and uranium. Additionally, it is also where we can find shells

with the lowest levels of iron and barium, and the highest levels of phosphorus, lanthanum and gadolinium. Estuário do Tejo (T) generates shells more enriched in magnesium, manganese and iron, and less in strontium and uranium. The highest amount of aluminum, along with lowest amount of cobalt were found in Ria de Aveiro (Rav). It is worth noting that the shells of clams gathered from Ria de Vigo, have a higher range of values of sodium, phosphorus, barium and gadolinium, and there is no chemical component where the range of values for this location is much smaller than for the other two. Shells from Ria de Aveiro embrace a much lower range of values of manganese and iron, and there is no noticeable component where the range of values is much higher in Ria de Vigo compared to the other locations. To finalize, in Estuário do Tejo, we find that the range of iron is clearly higher than for the other regions. Furthermore, there is no evident component where the range of values is much lower in Estuário do Tejo, than the other locations.

Due to the high-dimensionality of the training dataset (44 predictors, 24 train observations), regularization methods were implemented to assess which variables could be linked back to the specie's origin. While the *Ridge* method did not discard any variables, both LASSO and *Elastic Net* selected only the variables that the algorithm saw fit to accurately predict the specie's location of origin. To address the low statistically significant values of predictive performance that emerge when manipulating datasets with reduced number of observations, we resorted to employing a 1,000 iteration of *Monte Carlo Cross-Validation*. Therefore, we constructed 1,000 *testing* and *training sets* at random, hence fitting 1,000 distinct models for each regularization method. We ran and tested the models separately, and ultimately, for each regularization method, gathered useful conclusions by averaging the results of the 1,000 fitted models. At each iteration, the dataset was partitioned into *training* and *testing sets* with a respective ratio of 80%/20%, leading to *training sets* of 24 observations and *testing sets* of 6 observations.

The applied regularization techniques rely heavily on the presence of groups of highly correlated variables, and, therefore, we employed a correlation analysis on the manipulated dataset. Figure 4.4 displays the *Pearson correlation* between predictors (produced with the assistance of the `corrplot` package, (Wei & Simko, 2017)). In fact, there are a few groups of correlated variables, for example fatty acids FA14:0, FA16:1n_7, FA22:4n_6, FA20:1n_7, FA22:3n_6, FA22:5n_3, FA18:1n_7 and FA20:5n_3, and chemical elements U, Sr, Y and P, being, for the most part, very (negatively) correlated with fatty acids FA22:2n_9, FA22:6n_3, FA17:0, FA20:4n_6, FA18:1n_9, FA18:2n_6, FA18:3n_3, FA18:4n_3, FA15:0, FA20:1n_9_11, FA22:5n_6 and chemical element Na.

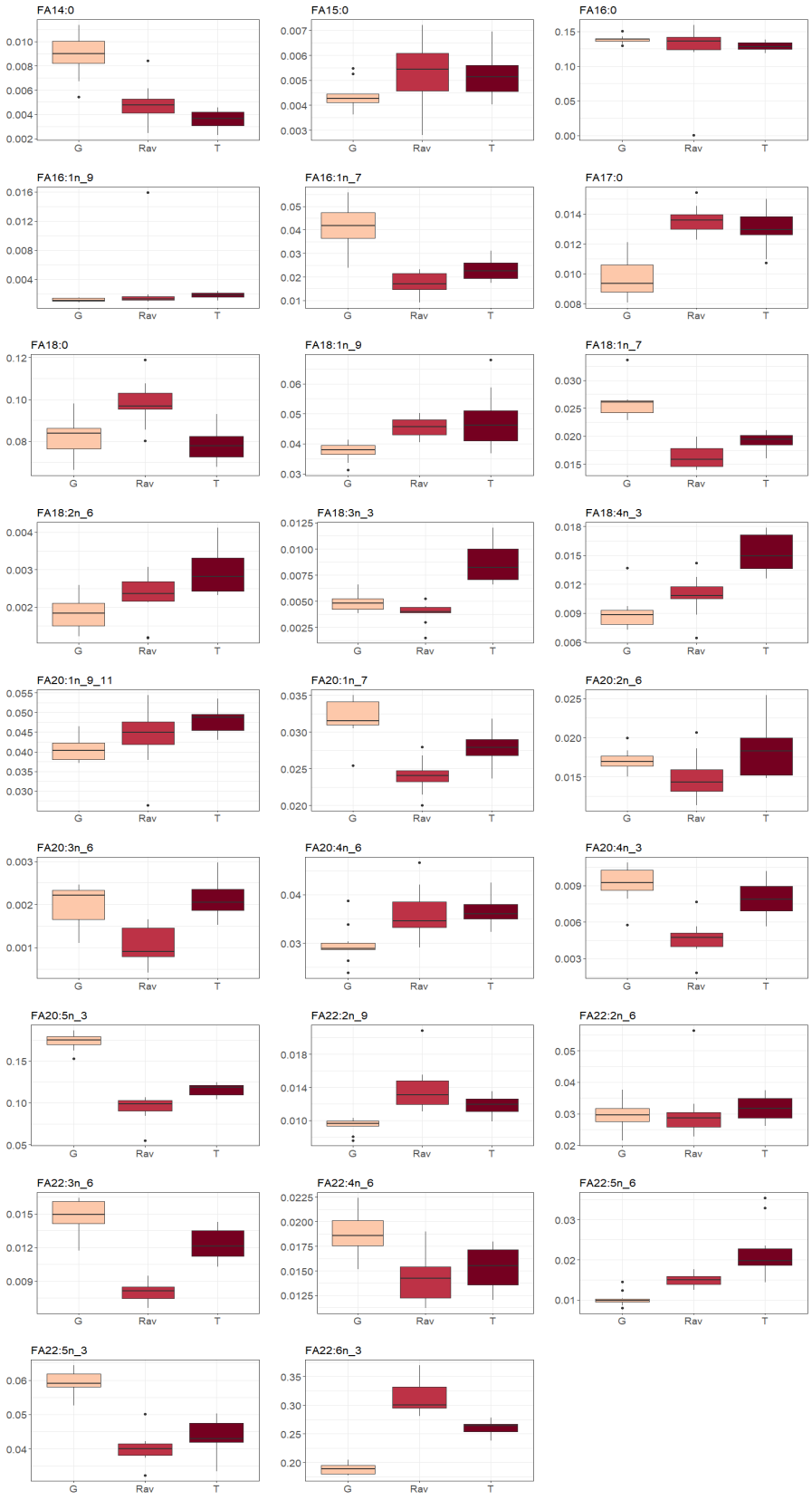



Figure 4.2: *Boxplots* representative of the distribution of the quantification of the different fatty acids in the different locations. Each plot corresponds to a different feature, and includes one *boxplot* for each location: Ria de Vigo (G), Ria de Aveiro (Rav) and Estuário do Tejo (T)

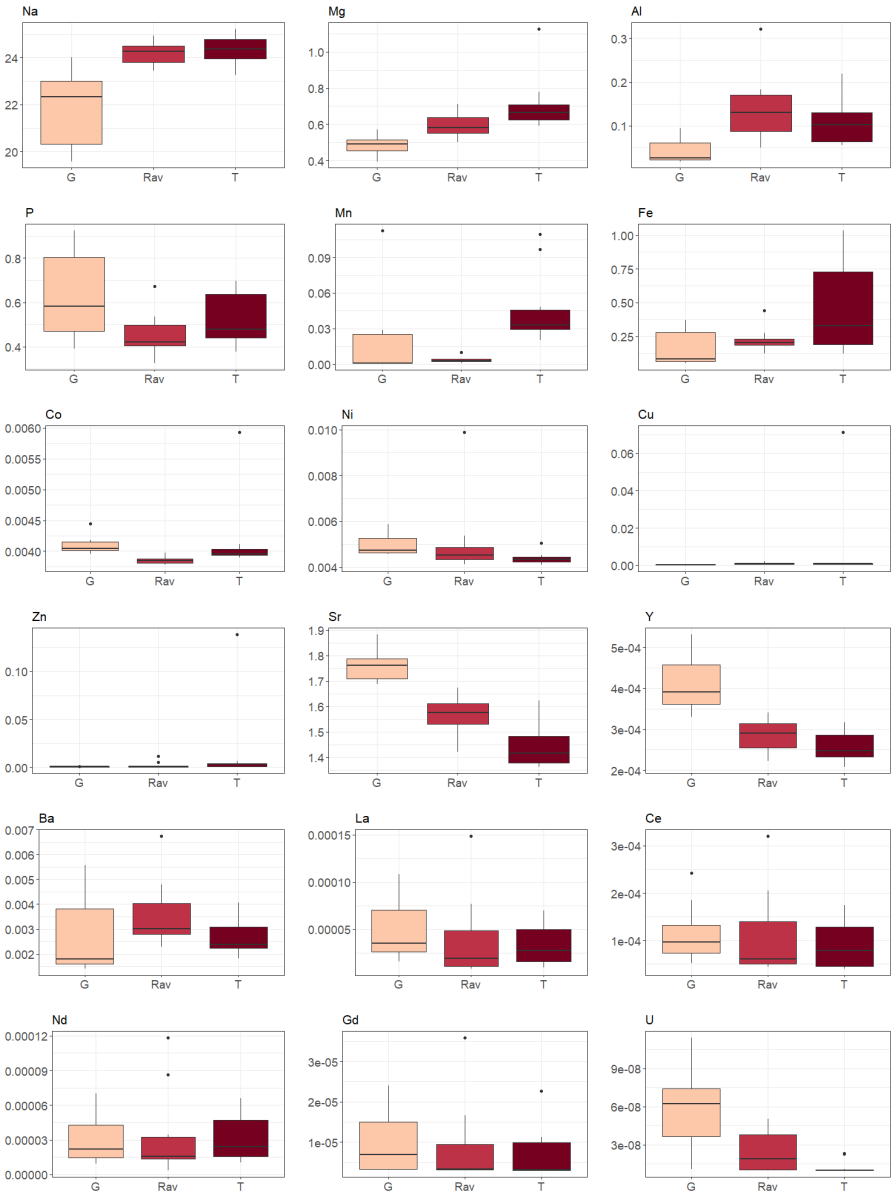



Figure 4.3: *Boxplots* representative of the distribution of the quantification of the different chemical elements of the clams' shell in the different locations. Each plot corresponds to a different feature, and includes one *boxplot* for each location: Ria de Vigo (G), Ria de Aveiro (Rav) and Estuário do Tejo (T)

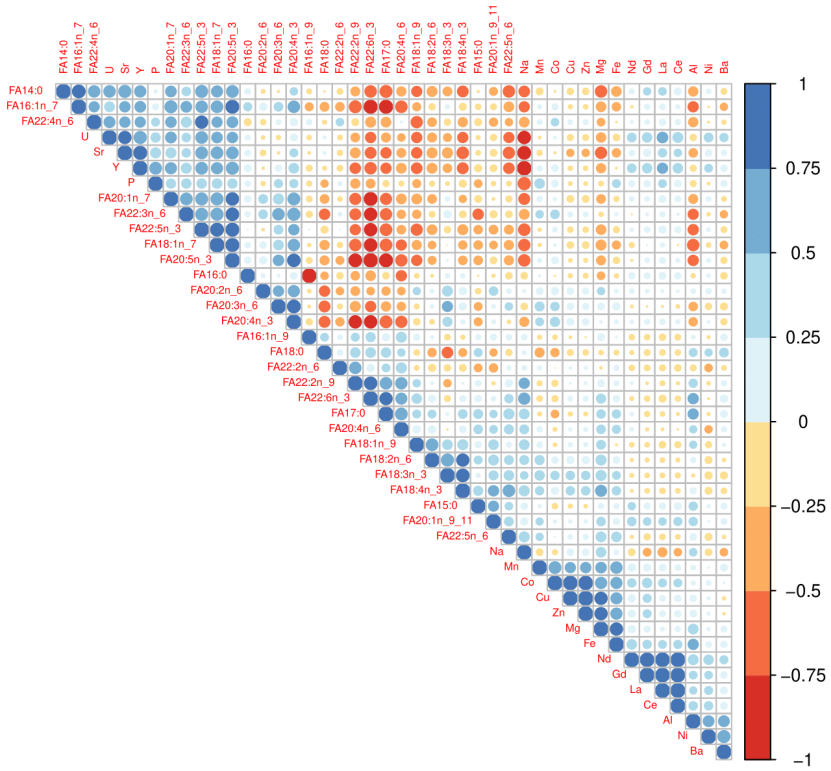



Figure 4.4: *Pearson correlation* between predictors by way of a color matrix. Each line and column corresponds to a different predictor, and their respective entry indicates the *Pearson correlation* between the two predictors

## 4.2 Results and Discussion

The `glmnet` package performs *Ridge*, LASSO and *Elastic Net* considering the penalty term (Hastie et al., 2016),

$$\lambda\Big[\frac{1-\alpha}{2}||\beta||_2^2+\alpha||\beta||_1\Big]. \tag{4.1}$$

This way, contrary to what was mentioned in section 3.4, to compute *Ridge* we set $\alpha = 0$, and $\alpha = 1$ for LASSO.

### 4.2.1 Penalization and mixing parameters

Finding the best *penalization parameter*, $\lambda$, is crucial for fitting regularized models. Choosing this parameter at random would most likely lead to faulty variable selection and inadequate predictive models. For this reason, *K-fold Cross-Validation* was implemented to find the ideal penalty parameters, *i.e.*, the penalty that conducted the model with the least predictive error

(misclassification error), while simultaneously shrinking the maximum number of model coefficients. Once more, assuming that a good ratio for *training*/*testing sets* is 80%/20%, we opted for a *5-fold Cross-Validation*. This procedure was done, for each iteration of the *Monte Carlo Cross-Validation*, and each regularization method, using the `cv.glmnet` function from the `glmnet` package.

The plots in Figures 4.5 and 4.6 show the graphical representations behind the process of picking the ideal *penalization parameter* in *Ridge* and LASSO, respectively, for the first 6 iterations of the *Monte Carlo Cross-Validation*. These plots showcase the percentage of misclassification error as function of $\log(\lambda)$, and consequently of $\lambda$, as well as the number of variables selected by the model as the penalty parameter increases. Because *Ridge* does not perform variable selection, the amount of variables selected is constant at 44. In contrast, we observe that, for LASSO, as the penalty term increases, so does the number of variables discarded by the model. Remark the vertical dotted lines and their respective penalization value. The left value of $\log(\lambda)$ corresponds to the model that had the minimum error while dropping the most number of variables. The right value of $\log(\lambda)$ corresponds to the model that dropped the highest number of variables while still being within a standard error of 1 from the model depicted on the left line. If only one dotted line is shown it implies that both these values for $\log(\lambda)$ are identical, *i.e.*, there was no model within 1 standard error of the model with minimum error that dropped more variables than the latter. Both of these values for $\log(\lambda)$ are viable for ideal *penalization parameter*. We chose the ideal estimate as the one that culminated in the model with the smallest error, while selecting the least amount of predictors into the model (left dotted line). For the first 6 iterations of *Monte Carlo Cross-Validation*, the chosen *penalization parameter* for *Ridge* and LASSO are displayed in Table 4.3.

Table 4.3: Tuned penalization parameter ($\lambda$) for *Ridge* and LASSO, using *5-fold Cross-Validation*, for the first 6 iterations ($i = 1,\ldots,6$) of *Monte Carlo Cross-Validation*

| **iteration** $i$ | **penalization parameter** $\lambda$ | |
|---|---|---|
| | *Ridge* | LASSO |
| 1 | 5.6405 | 0.2680 |
| 2 | 6.6988 | 0.2642 |
| 3 | 4.5303 | 0.2054 |
| 4 | 18.1641 | 0.2346 |
| 5 | 6.4636 | 0.1757 |
| 6 | 7.9556 | 0.1636 |

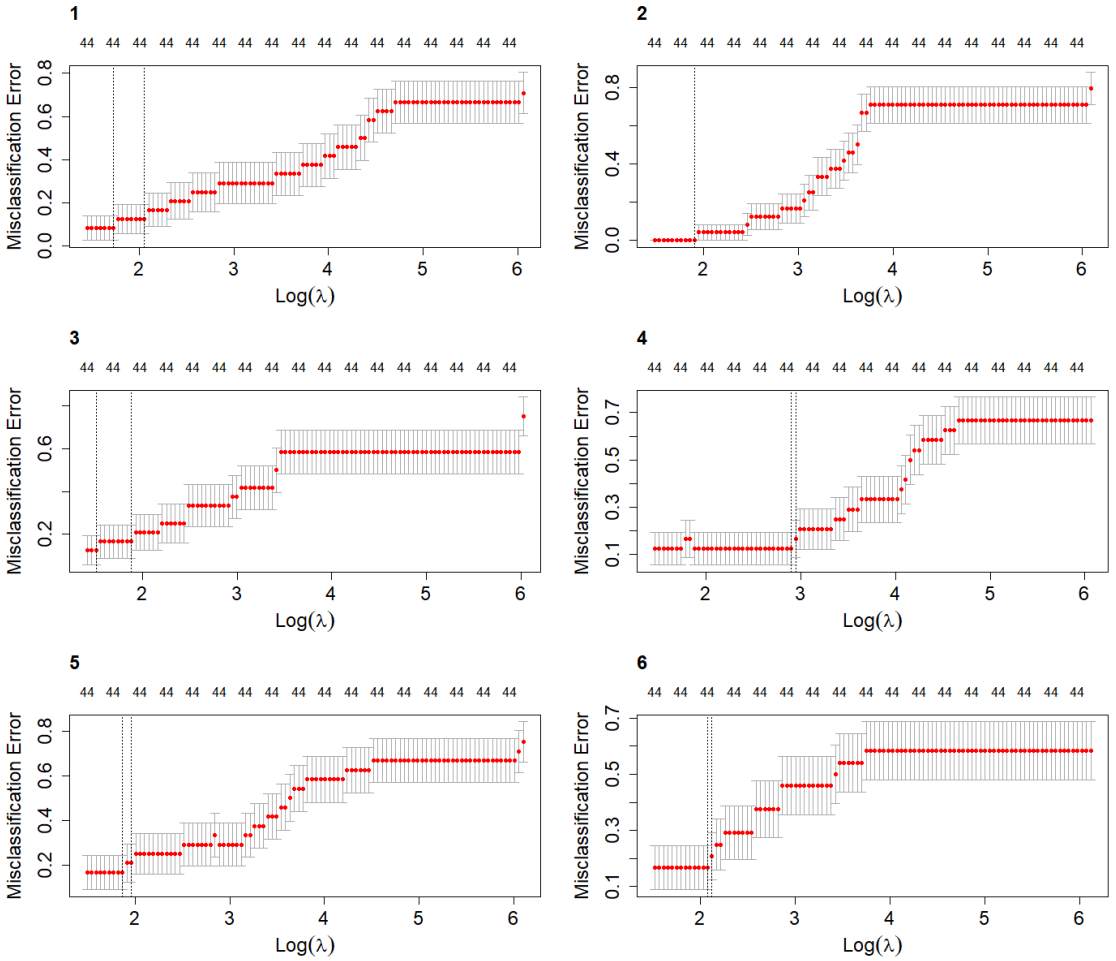



Figure 4.5: Graphical representation of the tuning process of the *penalization parameter* by means of *5-fold Cross-Validation*. These plots display the misclassification error as a function of $\log(\lambda)$ for the first 6 iterations of *Monte Carlo Cross-Validation*, considering the *Ridge* method of regularization. At each tested value of $\log(\lambda)$, the average error and standard deviation over the folds is computed and displayed in interval form. Two vertical dotted lines are shown: left line depicts the value of $\log(\lambda)$ linked to the model that performed the highest penalization, while maintaining the minimum misclassification error; right line outlines the value of $\log(\lambda)$ for the model that performed the highest penalization while still being within a standard error of 1 from the model depicted on the left line. If only one dotted line is shown it implies that both these values for $\log(\lambda)$ coincide. Numbers at the top of the plots represent the amount of variables selected by the model as the penalty increases

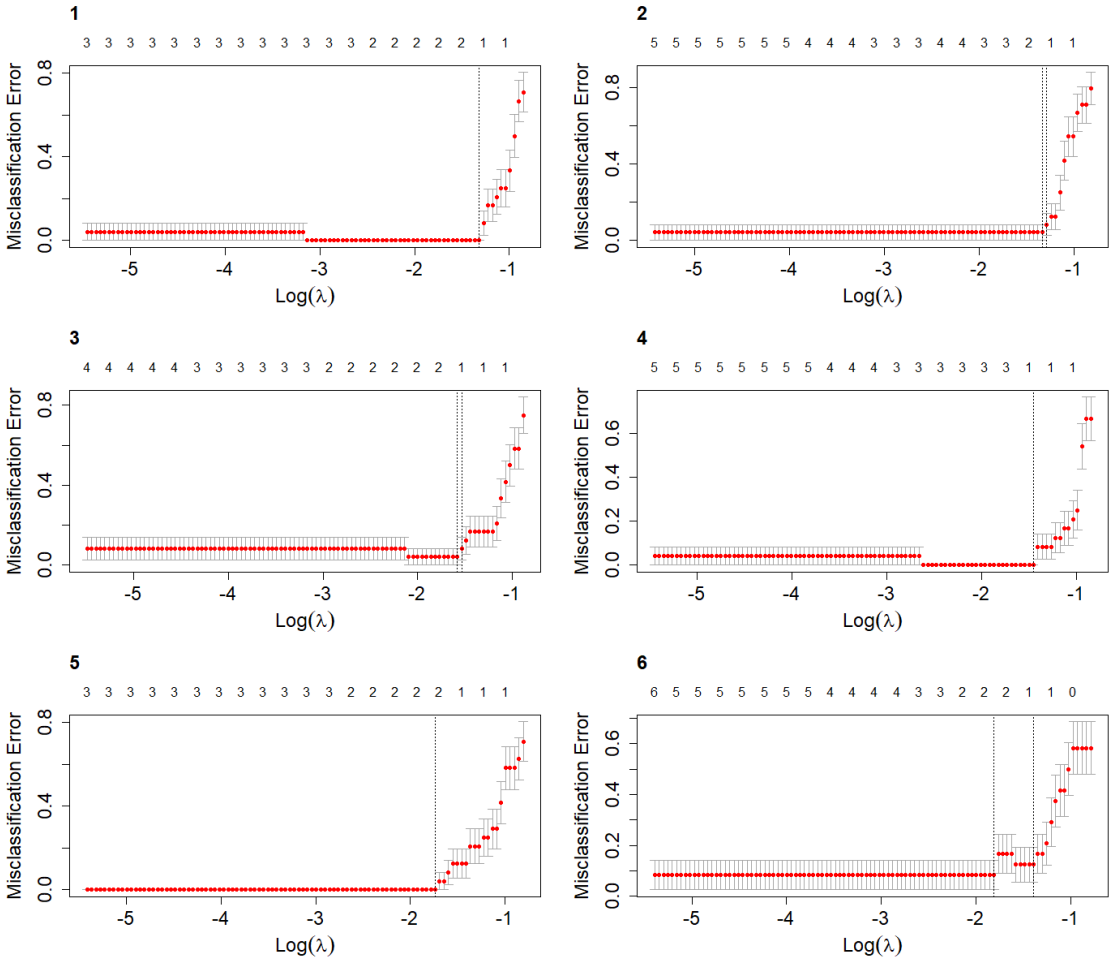



Figure 4.6: Graphical representation of the tuning process of the *penalization parameter* by means of *5-fold Cross-Validation*. These plots display the misclassification error as a function of $\log(\lambda)$ for the first 6 iterations of *Monte Carlo Cross-Validation*, considering the LASSO method of regularization. At each tested value of $\log(\lambda)$, the average error and standard deviation over the folds is computed and displayed in interval form. Two vertical dotted lines are shown: left line depicts the value of $\log(\lambda)$ linked to the model that dropped the most amount of variables, while maintaining the minimum misclassification error; right line outlines the value of $\log(\lambda)$ for the model that dropped the higher amount of variables while still being within a standard error of 1 from the model depicted on the left line. If only one dotted line is shown it implies that both these values for $\log(\lambda)$ coincide. Numbers at the top of the plots represent the amount of variables selected by the model as the penalty increases

*Elastic Net* is more complex than the other two methods, since it runs *K-fold Cross-Validation* to estimate the ideal *penalization parameter*, $\lambda$, and *mixing parameter*, $\alpha$, simultaneously. The *Elastic Net* penalty is regulated by the *mixing parameter*, that essentially connects *Ridge* and LASSO's penalties. The *penalization parameter* controls the overall penalty of the model (check the penalty expression (4.1)). Once more, we estimated the ideal *mixing* and *penalization parameters* using *5-fold Cross-Validation*, selecting the parameter that resulted in the model with the lowest error. This was done using the `train` function from the `caret` package (Kuhn, 2020).

Table 4.4 summarizes the estimated optimum values for the *penalization* and *mixing parameters* for the first 12 iterations of the *Monte Carlo Cross-Validation*. For the model fit in the $6th$ iteration, the regularization method with the best prediction quality was LASSO ($\alpha = 1$), with a penalization parameter value of $\lambda \approx 0.2593$. On the $9th$ iteration we observed a mixing parameter of $\alpha = 0.9$, which meant that the model performed much like the LASSO, but removing possible complications caused by highly correlated variables. Notice that the iterations with higher penalization values were mostly linked to lower mixing values, *i.e.*, the models that were closer to *Ridge* than to LASSO. This may have been due to the fact of *Ridge* not performing variable selection, but rather reducing the coefficient values of the least important variables, thus having higher penalties that promote a higher coefficient shrinkage.

Table 4.4: Optimum values for *mixing* ($\alpha$) and *penalization* ($\lambda$) *parameters* for *Elastic Net*, using *5-fold Cross-Validation*, considering the first 12 iterations ($i = 1, \ldots, 12$) of *Monte Carlo Cross-Validation*

| **Iteration** $i$ | **Mixing parameter** $\alpha$ | **Penalization parameter** $\lambda$ |
|---|---|---|
| 1 | 0.3 | 0.5615 |
| 2 | 0.1 | 0.5800 |
| 3 | 0.3 | 0.5432 |
| 4 | 0.1 | 0.5651 |
| 5 | 0.2 | 0.5862 |
| 6 | 1.0 | 0.2593 |
| 7 | 0.1 | 0.5844 |
| 8 | 0.1 | 0.5687 |
| 9 | 0.9 | 0.1663 |
| 10 | 0.6 | 0.3625 |
| 11 | 0.1 | 0.5763 |
| 12 | 0.3 | 0.5344 |

### 4.2.2 Variable Selection and Coefficient Shrinkage

The `glmnet` package fits *Multinomial Regression* models by rendering each class as an individual *Logistic Regression* model, *i.e.*, considering one class against the other two. This way, we were

able to analyze the variable selection done for each class, which was pertinent considering that different regions of origin of the species could very well be determined by different predictors.

Figures 4.7, 4.8 and 4.9 depict the graphical representations of the coefficient shrinkage for each class of the response variable, along the growth of the penalty parameter, for the first three iterations of *Monte Carlo Cross-Validation*, and for *Ridge*, LASSO and *Elastic Net* methods, respectively. On top of the plots the number of variables selected by the model is displayed. For the *Ridge* method, since there is no variable selection, the number of variables remains constant at 44. LASSO selected a very small number of variables to predict each class, even for the lowest considered penalty values. This could have been due to one of the main characteristics of the LASSO regarding how it handles groups of correlated variables. Naturally, the higher the penalty value, the lesser predictors were chosen. *Elastic Net* behaved somewhat as a midway between *Ridge* and LASSO, selecting more variables than LASSO, but not keeping them all inside the model, presumably given its *grouping effect*. With very low penalty terms we observed that, in *Elastic Net*, each class needed around 20 to 30 predictors.

Studying the velocity of convergence of the variable's coefficients to 0 also helped to define the importance of the different predictors in predicting the response's behavior. The most important predictors have the slowest convergence, while the less important variables exhibit fast convergence. This signifies that, at very low penalties, the least important variables will already have small coefficients, contrary to the important variables. Analyzing the model derived from the first iteration of *Monte Carlo Cross-Validation*, we defined three levels of importance: high, medium and low. The thresholds were established through the quantile measures of the absolute values of the model's coefficients at the lowest considered penalty, conditioned to the regularization method applied. A variable was termed important, if its absolute coefficient was higher than the $3rd$ quantile of the model's absolute coefficients; variables of medium importance had coefficients whose absolute values fell between the $2nd$ and $3rd$ quantiles; the remaining variables were considered of low importance. Tables C.1, C.2 and C.3 (see Appendix C) display the different variables' importance qualification, conditioned to the regularization techniques, for the first iteration model of *Monte Carlo Cross-Validation*.

Subsequently to this analysis, we adjusted the models for each of the regularization methods, considering the optimal *penalization* and *mixing parameters*

Since the nature of the response variable was categorical, fitted models estimated one coefficient per selected variable and per class of the response variable. Not dropping any predictors, which was the case for *Ridge*, instead of $44 + 1 = 45$ coefficients (taking into account the intercept coefficient, $\beta_0$), the fitted models estimated $44 \times 3 + 3 = 135$ coefficients. For LASSO and *Elastic Net*, the amount of coefficients to estimate depended on the choice of the *penalization* and *mixing parameters*, being capable of varying from 3 (intercept coefficient for each class of the response variable) and 135.

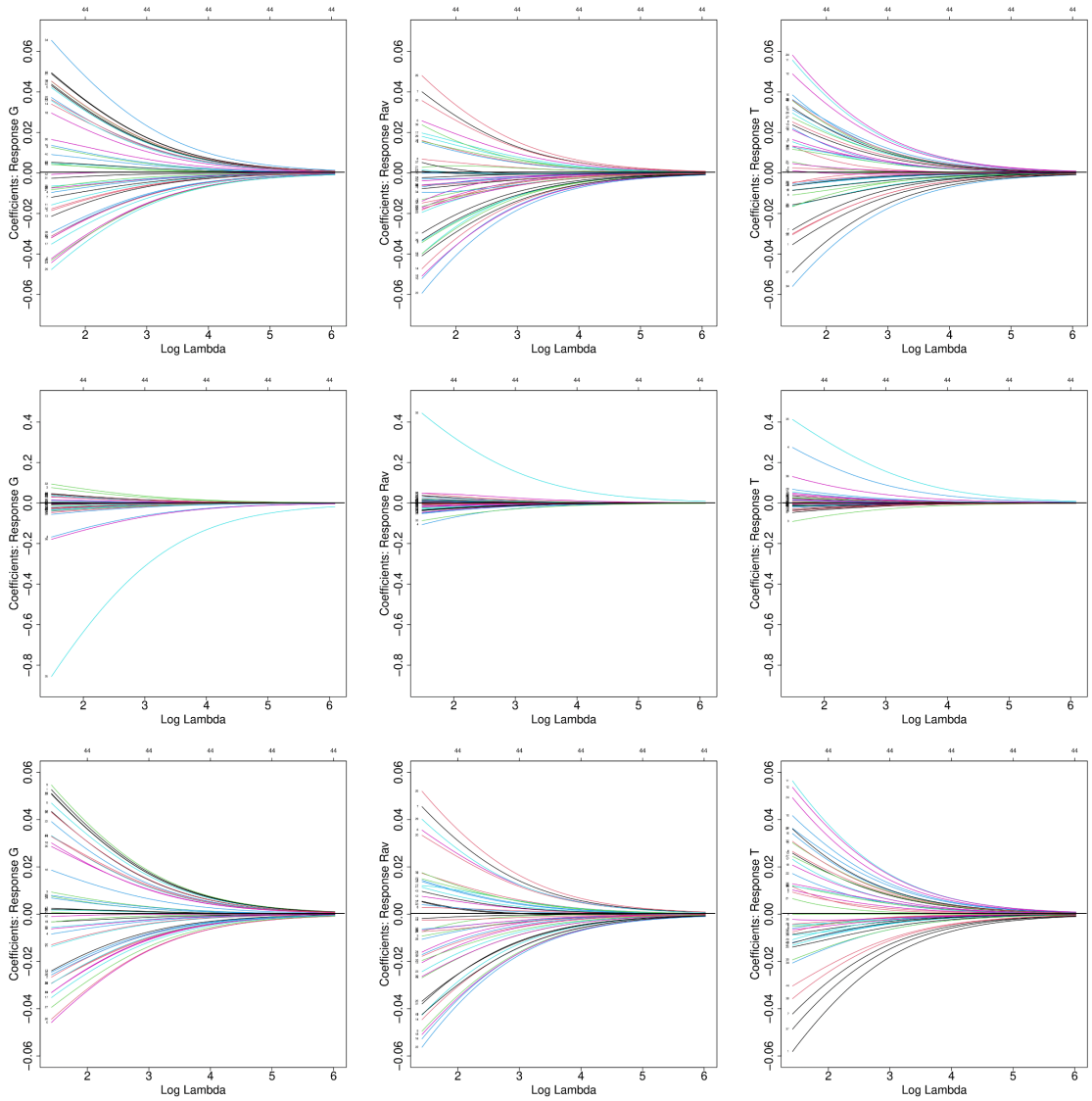



Figure 4.7: *Ridge*'s coefficient shrinkage for each class of the response variable, for the first 3 iterations of *Monte Carlo Cross-Validation*. The different classes are presented as columns: left: Ria de Vigo; middle: Ria de Aveiro; right: Estuário do Tejo. The different iteration models are presented as rows: top: 1*st* iteration model; middle: 2*nd* iteration model; bottom: 3*rd* iteration model. Different lines represent different parameter's coefficient shrinkage. Along the increase of the penalty parameter the coefficients converge to 0. The coefficients never actually achieve the null value, as this method does not perform variable selection. Numbers at the top of the plots represent the amount of predictors selected by the model as the penalty increases

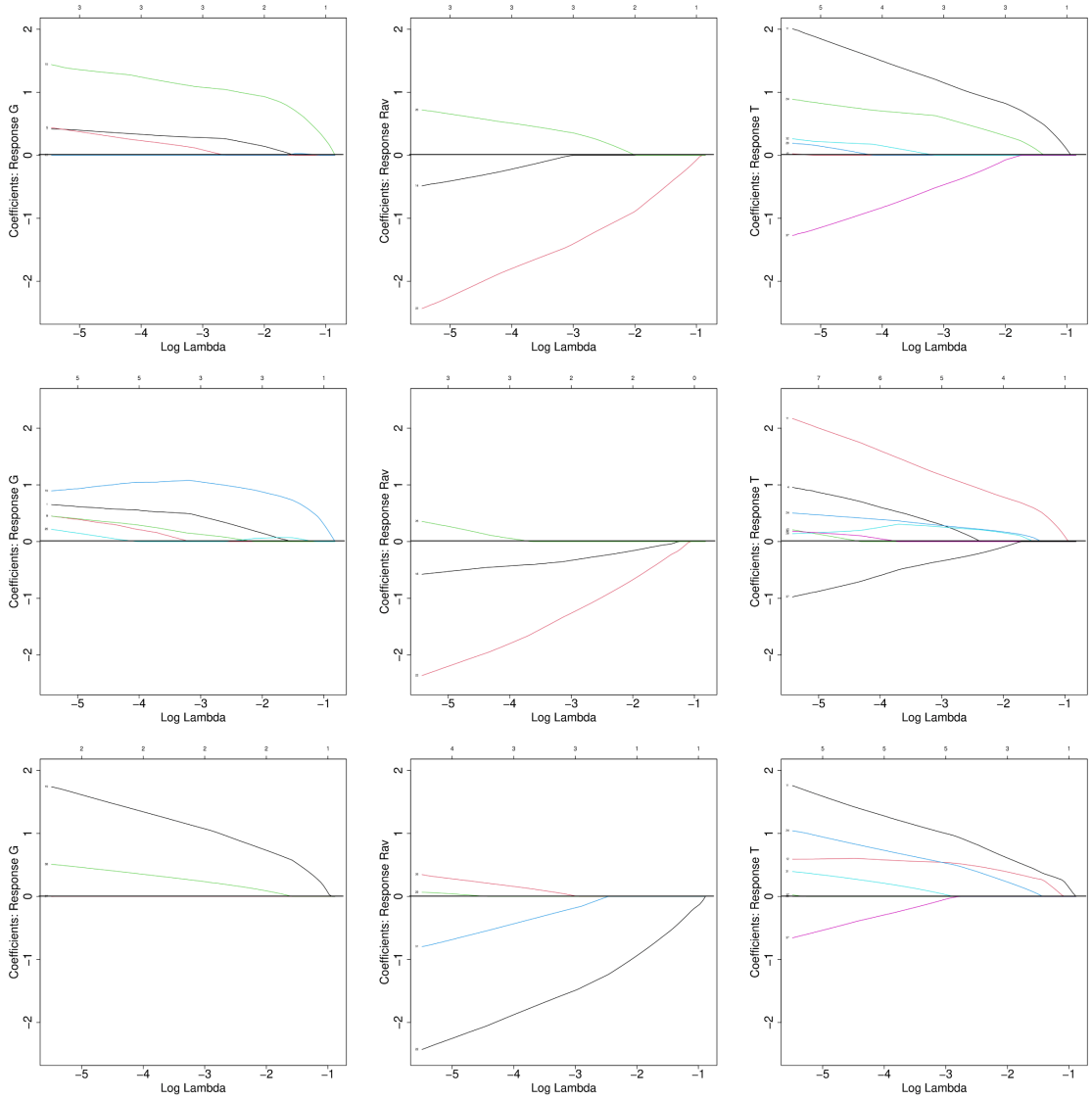



Figure 4.8: LASSO's coefficient shrinkage for each class of the response variable, for the first 3 iterations of *Monte Carlo Cross-Validation*. The different classes are presented as columns: left: Ria de Vigo; middle: Ria de Aveiro; right: Estuário do Tejo. The different iteration models are presented as rows: top: 1*st* iteration model; middle: 2*nd* iteration model; bottom: 3*rd* iteration model. Different lines represent different parameter's coefficient shrinkage. Along the increase of the penalty parameter the coefficients converge to 0. Numbers at the top of the plots represent the amount of predictors selected by the model as the penalty increases

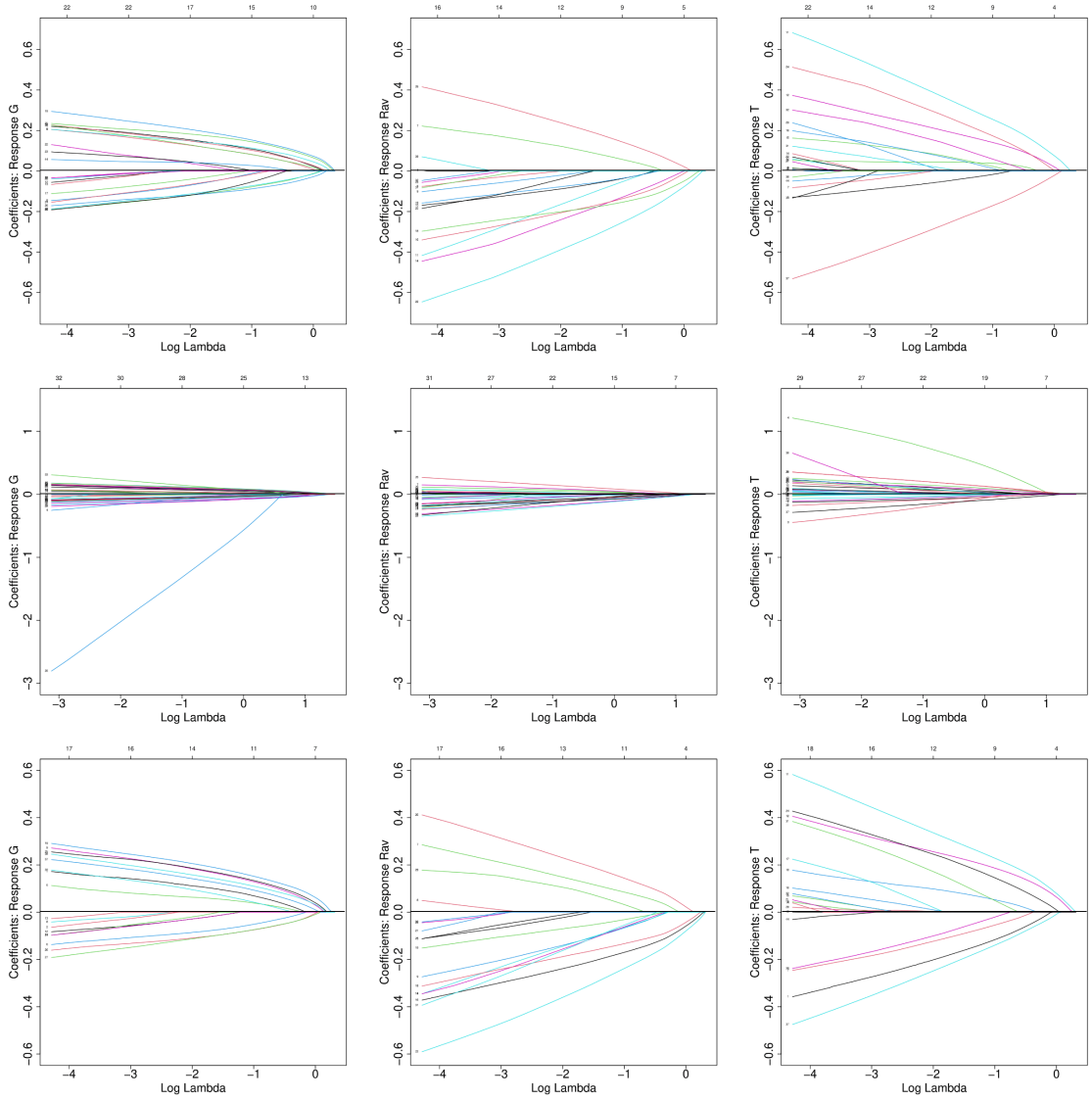
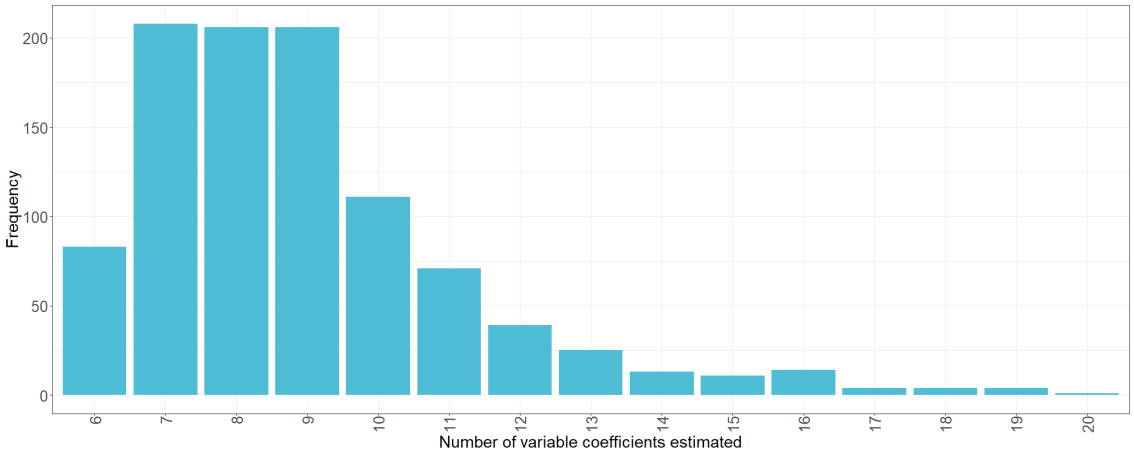
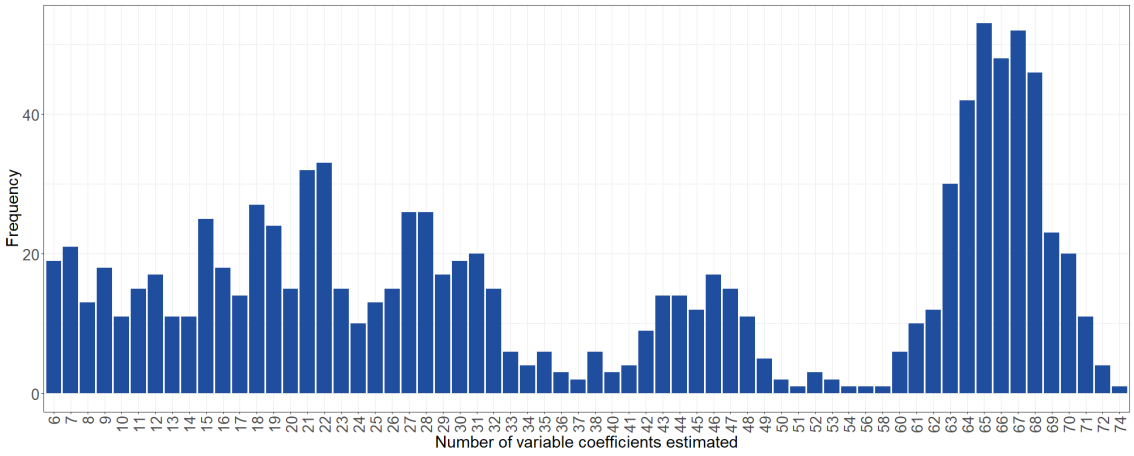



Figure 4.9: *Elastic Net*'s coefficient shrinkage for each class of the response variable, for the first 3 iterations of *Monte Carlo Cross-Validation*. The different classes are presented as columns: left: Ria de Vigo; middle: Ria de Aveiro; right: Estuário do Tejo. The different iteration models are presented as rows: top: 1*st* iteration model; middle: 2*nd* iteration model; bottom: 3*rd* iteration model. Different lines represent different parameter's coefficient shrinkage. Along the increase of the penalty parameter the coefficients converge to 0. Numbers at the top of the plots represent the amount of predictors selected by the model as the penalty increases

Figures 4.10(a) and 4.10(b) display the number of variable coefficients estimated throughout the 1,000 models according to LASSO and *Elastic Net*, respectively. The number of coefficients estimated by *Elastic Net* was much higher than LASSO's. For the 1,000 adjusted models, LASSO estimated between 6 to 20 variable coefficients in order to explain the behavior of the response, having a greater frequency of estimating between 7 to 9 variable coefficients. The *Elastic Net* estimated between 6 to 74 variable coefficients, remarking a clear higher frequency of estimating 60 to 70. Remark how, in terms of this measure, *Elastic Net* was roughly a midway through LASSO and *Ridge*. These results reflected the expected behavior of LASSO and *Elastic Net*'s penalties when faced with groups of highly correlated variables. Since LASSO arbitrarily selects one variable from each group and drops the remaining (and *Elastic Net* applies a grouping effect) it is expected that the amount of variable coefficients estimated in a LASSO model be far less than the amount estimated in an *Elastic Net* model.




(a) LASSO




(b) *Elastic Net*

Figure 4.10: Frequency of the number of variable coefficients (a) LASSO and (b) *Elastic Net* had to estimate throughout the 1,000 models. The $x$-axis represents the number of variable coefficients estimated by the model, *i.e.*, the total amount of variables used to predict each class of the response variable. Each bar represents the frequency of models that selected the corresponding amount of predictors

Table 4.5 displays the selected predictors and respective estimated coefficients given by LASSO and *Elastic Net* methods, for the first iteration of *Monte Carlo Cross-Validation*. Notice how, in this iteration, LASSO selected 2 features as sufficient to predict Ria de Vigo, 1 for Ria de Aveiro and 1 for Estuário do Tejo. In total, the algorithm finds that 4 predictors is enough to distinguish between the three locations. *Elastic Net* selects 12, 9 and 6 predictors as sufficient to predict Ria de Vigo, Ria de Aveiro and Estuário do Tejo, respectively.

Table 4.5: Model coefficients of the models fit by means of LASSO and *Elastic Net* regularization methods, for the first iteration of *Monte Carlo Cross-Validation*. For each regularized model, the variables selected to explain each class of the response variable are displayed along with their respective coefficients

| **LASSO** | | | | | |
|---|---|---|---|---|---|
| Ria de Vigo | | Ria de Aveiro | | Estuário do Tejo | |
| Predictor | Coefficient | Predictor | Coefficient | Predictor | Coefficient |
| Intercept | -0.0200 | Intercept | -0.0027 | Intercept | 0.0227 |
| FA20:5n_3 | 0.5675 | FA22:3n_6 | -0.3483 | FA18:3n_3 | 0.4346 |
| FA22:5n_3 | 0.0213 | - | - | - | - |
| ***Elastic Net*** | | | | | |
| Ria de Vigo | | Ria de Aveiro | | Estuário do Tejo | |
| Predictor | Coefficient | Predictor | Coefficient | Predictor | Coefficient |
| Intercept | -0.0301 | Intercept | -0.0125 | Intercept | 0.0435 |
| FA14:0 | 0.0792 | FA18:0 | 0.0259 | FA18:3n_3 | 0.1924 |
| FA16:1n_7 | 0.0569 | FA18:3n_3 | -0.0067 | FA18:4n_3 | 0.1031 |
| FA17:0 | -0.0554 | FA20:1n_7 | -0.0784 | FA22:5n_6 | 0.1137 |
| FA18:1n_7 | 0.0912 | FA20:2n_6 | -0.0089 | Mg | 0.0255 |
| FA20:5n_3 | 0.1241 | FA20:3n_6 | -0.0974 | Fe | 0.0167 |
| FA22:5n_3 | 0.1170 | FA20:4n_3 | -0.1242 | Sr | -0.1155 |
| FA22:6n_3 | -0.0686 | FA20:5n_3 | -0.0110 | - | - |
| Na | -0.0514 | FA22:3n_6 | -0.1891 | - | - |
| Al | -0.0142 | FA22:6n_3 | 0.0979 | - | - |
| Sr | 0.0753 | - | - | - | - |
| Y | 0.0564 | - | - | - | - |
| U | 0.0124 | - | - | - | - |

### 4.2.3 Model Validation

Model validation is the process of confirming that the outputs of the fitted models are approved as performing correctly according to the *testing sets*. In such circumstances, it would imply that these models would probably also have an accurate prediction to any other new sets of data where the response is unknown.

Note how the output of a *Multinomial Logistic Regression* model, for each test observation, is a set of three probabilities (that summed equal 1) of that observation belonging to each class of the response variable. The distribution of the class affiliation probabilities throughout the

1,000 fit models, for each regularization method, are displayed in Figure 4.11. To interpret these plots, notice how we can only assure that a certain class was chosen for a test observation if the probability of class affiliation is ≥ 0.5 (failing this, the observation might or might not belong to that class). In effect, we are able to see how, for the Ria de Vigo class of the response variable (G), there is a higher frequency of models that select this class with high probabilities, compared to the other two (more consistently seen in *Ridge*). It is worth noticing the basic statistical measures for the probability of class affiliation, per class of the response, and per regularization method, displayed in Table 4.6.

Table 4.6: Basic statistical measures (minimum, maximum, mean and variance) for the range of values of probability of class affiliation, per class, and per regularization method

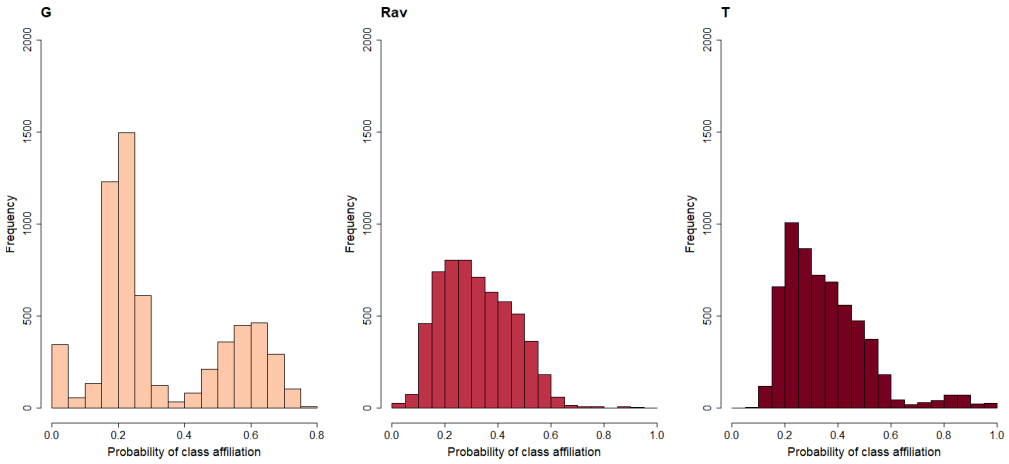
| | **Ridge** | | | **LASSO** | | | **Elastic Net** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Measure** | G | Rav | T | G | Rav | T | G | Rav | T |
| Min | 0.0000 | 0.0001 | 0.0387 | 0.0000 | 0.0000 | 0.0009 | 0.0000 | 0.0000 | 0.0134 |
| Max | 0.7611 | 0.9612 | 0.9999 | 0.9990 | 0.9919 | 1.0000 | 0.9803 | 0.9838 | 1.0000 |
| Mean | 0.3249 | 0.3212 | 0.3540 | 0.3238 | 0.3217 | 0.3544 | 0.3182 | 0.3103 | 0.3715 |
| Variance | 0.0383 | 0.0187 | 0.0249 | 0.0694 | 0.0552 | 0.0594 | 0.0711 | 0.0457 | 0.0581 |

Legend: Min - minimum, Max - maximum; G - Ria de Vigo, Rav - Ria de Aveiro, T - Estuário do Tejo

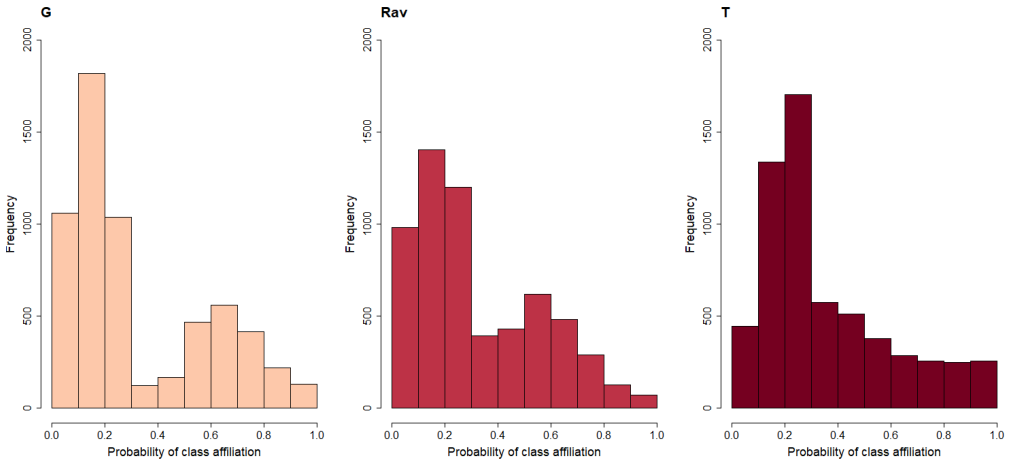
We also studied, for the *testing sets* of the 1,000 models and considering each regularization method, how many times a certain class was predicted in comparison to the actual number of times that class appeared in the testing set (see Figure 4.12). For all the regularization methods, we see how Ria de Vigo (G) is the best predicted class due to fact that it is the class where the correspondent frequency of prediction is closest to the actual class frequency. For the Ria de Aveiro class (Rav), note that, for the 3 regularization methods, the frequency of prediction is lower than the true class frequency, contrary to the Estuário do Tejo class (T), where the opposite occurs. These differences are more noticeable in *Elastic Net* and *Ridge*, and less in LASSO.

#### 4.2.3.1 Cross Entropy

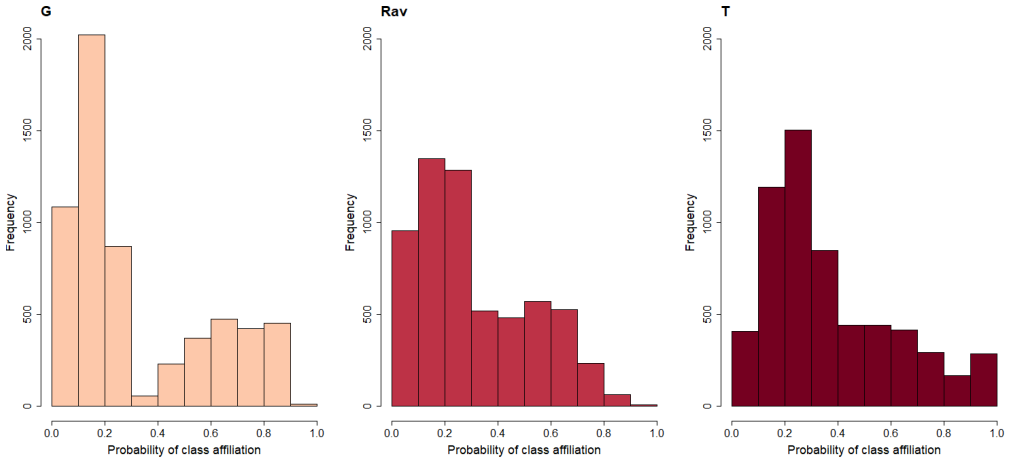
Subsequent to applying the three regularization methods to the 1,000 models, we validated the regularized models in the corresponding *testing sets* with the intention of examining predictive quality. The output of these models, for each testing sample, is a vector of probabilities of affiliation to each class of the response variable. For instance, regarding the first iteration of *Monte Carlo Cross-Validation*, the predictive probabilities given by the models regularized by the three methods are displayed in Tables 4.7, 4.8 and 4.9. Naturally, although for this particular iteration, all the models predicted the correct class for the 6 test samples, the probabilities in which these classes were predicted differ.




(a) *Ridge*

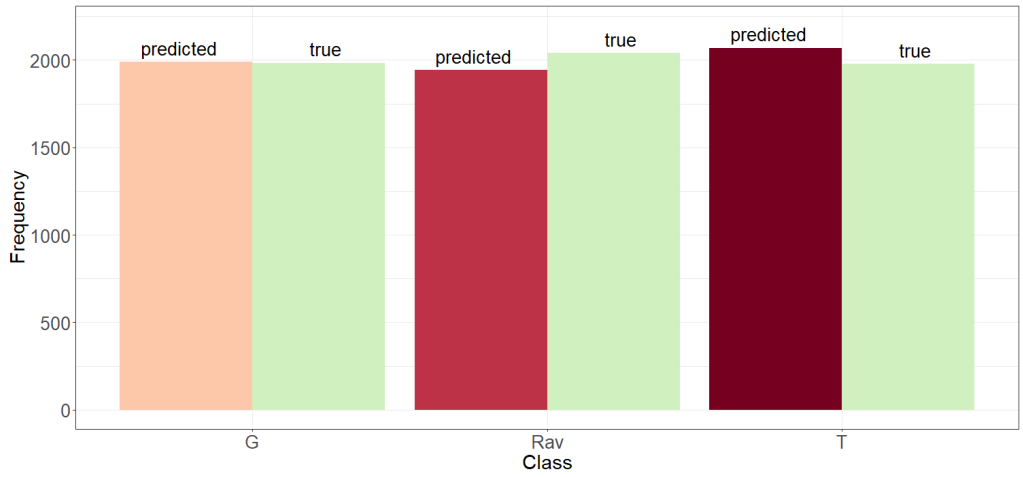



(b) LASSO

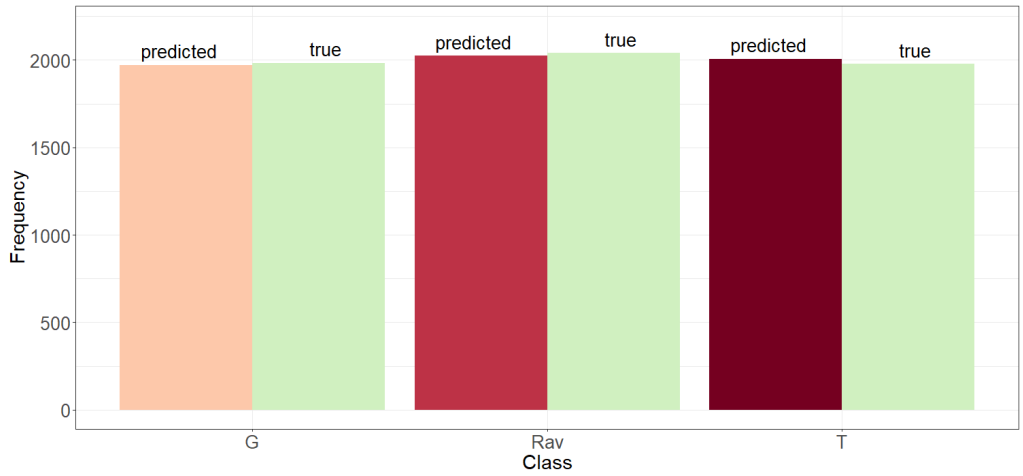



(c) *Elastic Net*

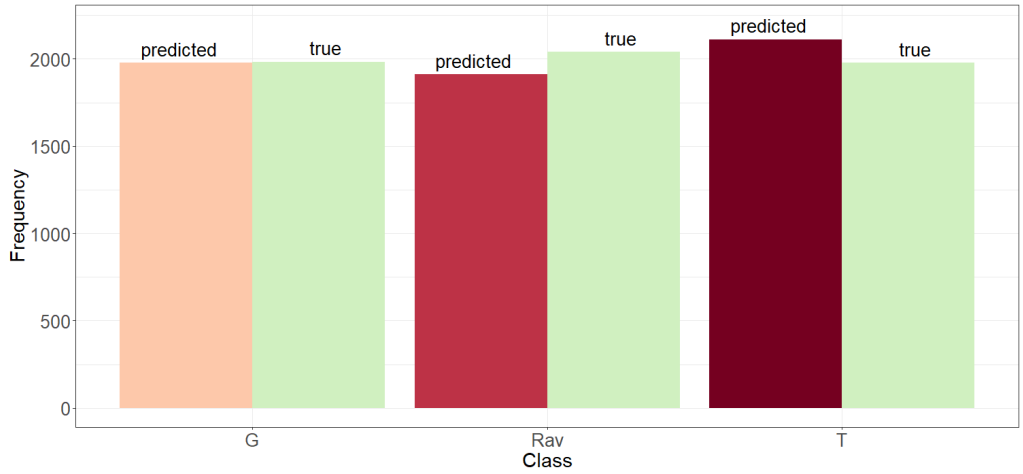
Figure 4.11: Distribution of the probability of class membership for the testing observations throughout the 1,000 testing sets, and considering the models adjusted by means of (a) *Ridge*, (b) LASSO, (c) *Elastic Net* methods of regularization. Left: Ria de Vigo, middle: Ria de Aveiro, right: Estuário do Tejo




(a) *Ridge*




(b) LASSO




(c) *Elastic Net*

Figure 4.12: Frequency of the number of times the different classes were predicted compared to the number of times they actually appeared in the 1,000 testing sets, considering (a) *Ridge*, (b) LASSO and (c) *Elastic Net* methods of regularization

Table 4.7: Class affiliation predicted probabilities for the testing observations relative to the first iteration of the *Monte Carlo Cross-Validation*, considering the *Ridge* method of regularization

| ***Ridge*** | | | | | |
|---|---|---|---|---|---|
| **Test observation index** | **Ria de Vigo** G | **Ria de Aveiro** Rav | **Estuário do Tejo** T | **Predicted class** | **Actual class** |
| 5 | 0.2203 | 0.2422 | 0.5376 | T | T |
| 9 | 0.2060 | 0.3616 | 0.4323 | T | T |
| 15 | 0.2325 | 0.4497 | 0.3178 | Rav | Rav |
| 20 | 0.2587 | 0.4871 | 0.2543 | Rav | Rav |
| 26 | 0.7063 | 0.1294 | 0.1643 | G | G |
| 30 | 0.6078 | 0.1832 | 0.2090 | G | G |

Table 4.8: Class affiliation predicted probabilities for the testing observations relative to the first iteration of the *Monte Carlo Cross-Validation*, considering the LASSO method of regularization

| ***Ridge*** | | | | | |
|---|---|---|---|---|---|
| **Test observation index** | **Ria de Vigo** G | **Ria de Aveiro** Rav | **Estuário do Tejo** T | **Predicted class** | **Actual class** |
| 5 | 0.2370 | 0.2825 | 0.4805 | T | T |
| 9 | 0.2216 | 0.3750 | 0.4034 | T | T |
| 15 | 0.1972 | 0.5559 | 0.2469 | Rav | Rav |
| 20 | 0.2351 | 0.5051 | 0.2597 | Rav | Rav |
| 26 | 0.5275 | 0.1953 | 0.2772 | G | G |
| 30 | 0.6310 | 0.1445 | 0.2246 | G | G |

Table 4.9: Class affiliation predicted probabilities for the testing observations relative to the first iteration of the *Monte Carlo Cross-Validation*, considering the *Elastic Net* method of regularization

| ***Elastic Net*** | | | | | |
|---|---|---|---|---|---|
| **Test observation index** | **Ria de Vigo** G | **Ria de Aveiro** Rav | **Estuário do Tejo** T | **Predicted class** | **Actual class** |
| 5 | 0.1952 | 0.1814 | 0.6233 | T | T |
| 9 | 0.2257 | 0.3785 | 0.3957 | T | T |
| 15 | 0.1975 | 0.5084 | 0.2941 | Rav | Rav |
| 20 | 0.1795 | 0.5328 | 0.2877 | Rav | Rav |
| 26 | 0.7014 | 0.1215 | 0.1771 | G | G |
| 30 | 0.6699 | 0.1442 | 0.1860 | G | G |

Evidently, the predictive quality of the models should take the affiliation probabilities into consideration, *i.e.*, if, between two models, one of them predicts the correct classes with higher probabilities, then it should be considered a better model in terms of predictive quality. That

being the case, *Cross Entropy* (or *Log-Loss*) measures how close a model is to predicting the correct class with 100% certainty (this is, with probability 1). For example, still reflecting on the first iteration of *Monte Carlo Cross-Validation*, the *Cross Entropy* valued 0.6372 for *Ridge*, 0.6685 for LASSO, and 0.5767 for *Elastic Net*. This implies that, for this iteration, the method that yielded the model with the best predictive quality, in terms of this measure, was *Elastic Net*, followed by *Ridge* and then LASSO (since better predictive models entail lower *Cross Entropy* values). Nonetheless, these conclusions are not very valuable when considering a single model and 6 testing observations. Averaging the *Cross Entropy* results from the 1,000 models, we get an average of around 0.7468 for *Ridge*, 0.9017 for LASSO, and 0.6203 for *Elastic Net*, once more resulting in *Elastic Net* being the method that, on average, constructs models with the lowest *Cross Entropy* values, followed by *Ridge* and then LASSO.

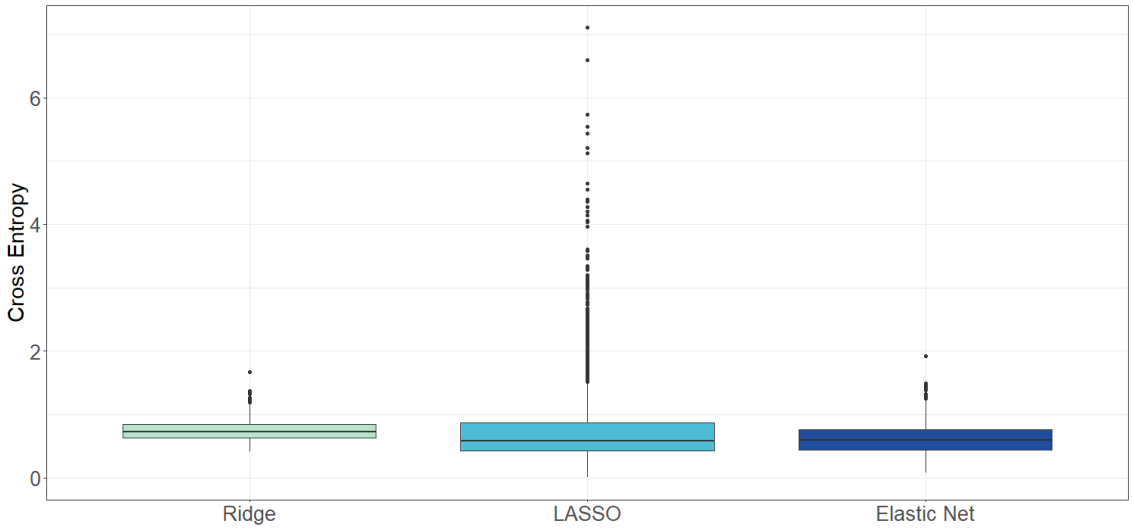
Besides computing the *Cross Entropy* for the *testing sets*, one can also take into consideration the values of *Cross Entropy* for the *training sets*, in order to gain a better understanding of each regularization method's behavior. Simply looking at the average *training Cross Entropy* we retained very distinct conclusions from the ones made on the *testing Cross Entropy*:

- *Ridge training Cross Entropy* $\approx 0.5969$
- LASSO *training Cross Entropy* $\approx 0.4151$
- *Elastic Net training Cross Entropy* $\approx 0.4205$

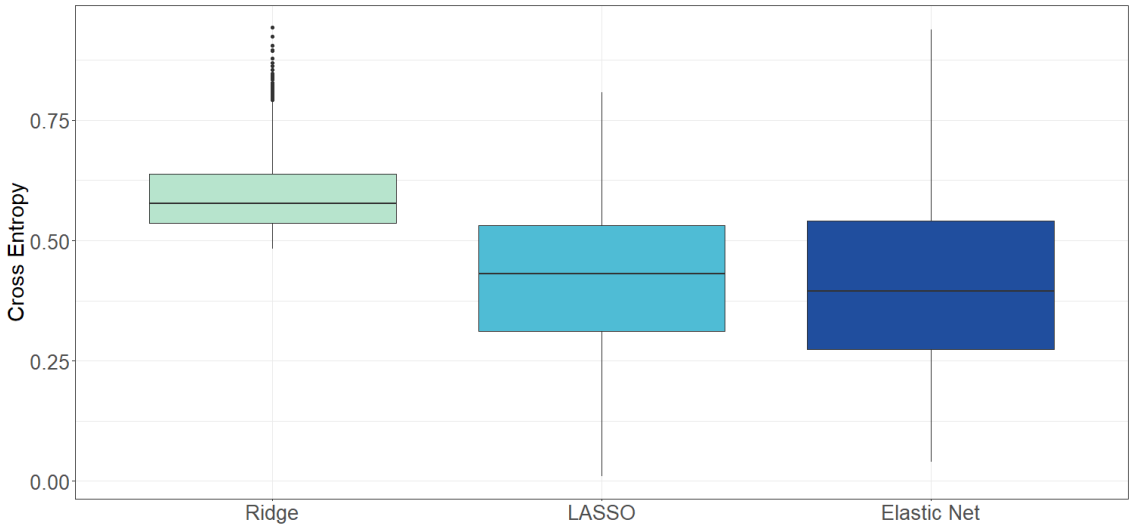
Evidently, this signifies that, on average, the LASSO method produced the models with the best *training Cross Entropy* values, followed by *Elastic Net* and then *Ridge*. In addition, notice that LASSO and *Elastic Net*'s average values are remarkably closer conversely to the average value for the *Ridge* method.

Because the conclusions made for the *training* and *testing Cross Entropy* average values were so distinct, we constructed the *boxplots* that illustrate the 1,000 models' *Cross Entropy* values, for each regularization method. This way, for each iteration of *Monte Carlo Cross-Validation* and each regularization method, we computed the *Cross Entropy* values for both the *testing* and the *training sets*. Figures 4.13(a) and 4.13(b) show the *boxplots* of the *Cross Entropy* values throughout the 1,000 models for the *training* and *testing sets*, respectively. With respect to Figure 4.13(a), comparatively to *Ridge* and *Elastic Net*, the LASSO method generated the models with the highest, and lowest, *Cross Entropy* values, several of them being *outliers*. We concluded that, unlike the other two methods which had a more condensed range of *Cross Entropy* values, LASSO could be considered slightly unstable, as this method displayed a higher variability in its models' *testing Cross Entropy* values, *i.e.*, different models led to big differences in *Cross Entropy*. It is worth noting that around 75% of the values from *Ridge* were higher than the median of the *Cross Entropy* values from both LASSO and *Elastic Net*. Regarding *training Cross Entropy*, in Figure 4.13(b), notice how, conversely to *testing Cross Entropy*, the values were, in general, much lower, varying between 0 and 1. Here, we no longer observed a pattern of LASSO yielding the models with the highest values, but instead observed *Ridge* being the method with the worst

behavior in terms of *training Cross Entropy* values. Notice how almost 100% of the *Ridge training Cross Entropy* values were higher than 0.5, while for the other two methods, around 75% of the values were lower than this same threshold. In this case, LASSO and *Elastic Net* appeared to perform identically.




(a) *Testing Cross Entropy*




(b) *Training Cross Entropy*

Figure 4.13: (a) *Testing* and (b) *Training Cross Entropy boxplots* for comparing the different values between the three regularization methods

#### 4.2.3.2 Confusion Matrices

An alternative way to analyze the predictive quality of the models is formulating, for each regularization method, a *Confusion Matrix* and computing the indicators and metrics discussed in Subsection 3.2.3. To perform a *Confusion Matrix* analysis, we merged the testing samples of the 1,000 models, into a larger *testing set* composed of 6,000 observations for testing. These

*Confusion Matrices* for *Ridge*, LASSO and *Elastic Net*, are displayed in Table 4.10. Right away, we observed that the Ria de Vigo (G) class was correctly predicted more often, when compared to the other two classes.

Table 4.10: *Confusion Matrices* of the three regularization methods, considering the 6,000 testing observations (merge of the 1,000 testing sets, with 6 observations each)

| | **True Class** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | *Ridge* | | | LASSO | | | *Elastic Net* | | |
| **Predicted Class** | G | Rav | T | G | Rav | T | G | Rav | T |
| G | 1,982 | 2 | 5 | 1,968 | 0 | 3 | 1,974 | 0 | 4 |
| Rav | 0 | 1,628 | 314 | 0 | 1,952 | 71 | 0 | 1,804 | 107 |
| T | 0 | 410 | 1,659 | 14 | 88 | 1,904 | 8 | 236 | 1,867 |

Legend: G - Ria de Vigo, Rav - Ria de Aveiro, T - Estuário do Tejo

To complete this analysis we computed the *Confusion Matrix* indicators and measures mentioned in Subsection 3.2.3 (see Tables 4.11, 4.12 and 4.13). Analyzing Table 4.11, we can clearly see that Ria de Vigo (G) was correctly predicted more often than the other two classes (higher number of True Positives and True Negatives; lower number of False Positives and False Negatives). The *Confusion Matrix* indicators for classes Ria de Aveiro (Rav) and Estuário do Tejo (T) appear similar.

Table 4.11: *Confusion Matrix* indicators for the three regularization methods

| | *Ridge* | | | LASSO | | | *Elastic Net* | | |
|---|---|---|---|---|---|---|---|---|---|
| **Indicator** | G | Rav | T | G | Rav | T | G | Rav | T |
| TP | 1,982 | 1,628 | 1,659 | 1,968 | 1,952 | 1,904 | 1,974 | 1,804 | 1,867 |
| FP | 7 | 314 | 410 | 3 | 71 | 102 | 4 | 107 | 244 |
| TN | 4,011 | 3,646 | 3,612 | 4,015 | 3,889 | 3,920 | 4,014 | 3,853 | 3,778 |
| FN | 0 | 412 | 319 | 14 | 88 | 74 | 8 | 236 | 111 |

Legend: TP - True Positives, FP - False Positives, TN - True Negatives, FN - False Negatives; G - Ria de Vigo, Rav - Ria de Aveiro, T - Estuário do Tejo

Reaching significant conclusions based solely on the *Confusion Matrix* indicators can be a complex task, which is why we resort to the performance measures displayed in Tables 4.12 and 4.13. Studying Table 4.12, note how, for the three methods, the *Accuracy*, *Precision*, *Recall*, *Specificity* and *F1-Score* for predicting Ria de Vigo (G) class was higher than for the other two classes (while the *Misclassification Rate* was lower), which evidently implies that this class

was the best predicted out of the three. Essentially this means that we have higher assurance that, when the model predicts the origin as Ria de Vigo, it is a correct prediction. Additionally, we observed that the performance measures for Ria de Vigo were identical among the three regularization methods. *Ridge* method seemed to perform the worst with regards to predicting Ria de Aveiro (Rav) and Estuário do Tejo (T), while LASSO achieved the best results for these two classes.

Table 4.12: Individual class performance measures considering the *Confusion Matrix* indicators for the three regularization methods

| | *Ridge* | | | LASSO | | | *Elastic Net* | | |
|---|---|---|---|---|---|---|---|---|---|
| **Measure** | G | Rav | T | G | Rav | T | G | Rav | T |
| acc | 0.9988 | 0.8790 | 0.8785 | 0.9972 | 0.9735 | 0.9707 | 0.9980 | 0.9428 | 0.9408 |
| mcr | 0.0012 | 0.1210 | 0.1215 | 0.0028 | 0.0265 | 0.0293 | 0.0020 | 0.0572 | 0.0592 |
| prec | 0.9965 | 0.8383 | 0.8018 | 0.9985 | 0.9649 | 0.9492 | 0.9980 | 0.9440 | 0.8844 |
| rec | 1.0000 | 0.7980 | 0.8387 | 0.9929 | 0.9569 | 0.9626 | 0.9960 | 0.8843 | 0.9439 |
| spec | 0.9983 | 0.9207 | 0.8981 | 0.9993 | 0.9821 | 0.9746 | 0.9990 | 0.9730 | 0.9393 |
| F1 | 0.9982 | 0.8177 | 0.8199 | 0.9957 | 0.9609 | 0.9558 | 0.9970 | 0.9132 | 0.9132 |

Legend: acc - *Accuracy*, mcr - *Misclassification Rate*, prec - *Precision*, rec - *Recall*, spec - *Specificity* and F1 - *F1-Score*; G - Ria de Vigo, Rav - Ria de Aveiro, T - Estuário do Tejo

Regarding Table 4.13, and taking into consideration that the ideal model would have the *F1-scores* valuing 1, while the worst model would have null *F1-scores*, we concluded that, generally, the LASSO method yielded the models with the best overall performance measures, followed by *Elastic Net* and then *Ridge* methods.

Table 4.13: Overall model performance measures (*Micro F1*, *Macro F1* and *Weighted F1*) considering the *Confusion Matrix* indicators for the three regularization methods

| **Measure** | *Ridge* | LASSO | *Elastic Net* |
|---|---|---|---|
| *Micro F1* | 0.8782 | 0.9707 | 0.9408 |
| *Macro F1* | 0.8785 | 0.9708 | 0.9411 |
| *Weighted F1* | 0.8780 | 0.9707 | 0.9409 |

#### 4.2.3.3 ROC curve and AUC

As mentioned in Section 3.2, analyzing ROC *curves* is an alternative way to study and compare the predictive quality of different models. Once more, in order to carry out this analysis, we resorted to merging the 1,000 different testing samples into one *testing set* composed of 6,000 observations.

A ROC *curve* is typically established in binary problems, where the response variable has two classes: positive and negative. In our *Multinomial* problem the response variable has three classes. For this reason, we opted to construct, for each class, a ROC *curve* of that class against the other two. This means that, for each of the regularization methods, we have three ROC *curves*, representative of each class of the response variable. These curves were constructed resorting to packages ROCR (Sing et al., 2005) and PRROC (Grau et al., 2015). Analyzing Figure 4.14, (and with assistance of Figures B.1, B.2 and B.3), we can see that, for the *Ridge* method, although the predictive quality was remarkably good for all classes, it was best when predicting Ria de Vigo. Since the perfect ROC *curve* is the one closest to the top left corner of the orthonormal axis, we can conclude that the discrimination of Ria de Vigo performed almost perfectly. By examining the AUC-*score* for each class (*Area Under the Curve*) we can verify that, after Ria de Vigo (AUC $\approx$ 1), Estuário do Tejo (AUC $\approx$ 0.9028) had a better score than Ria de Aveiro (AUC $\approx$ 0.8627). With respect to Figure 4.15(a) (along with B.4, B.5 and B.6) remark how, overall, the LASSO method appeared to perform better than *Ridge*. Again, regarding the curves' respective AUC, we can conclude that the LASSO method performed a better discrimination over Ria de Vigo class (AUC $\approx$ 0.9990), followed by Estuário do Tejo (AUC $\approx$ 0.9785), and then Ria de Aveiro (AUC $\approx$ 0.9630). Regarding Figure 4.15(b), along with B.7, B.8 and B.9, we see how, overall, LASSO still performed better than *Elastic Net*. Still, alike the other two regularization methods, Ria de Vigo was the best predicted class (AUC $\approx$ 1), followed by Estuário do Tejo (AUC $\approx$ 0.9397), and then Ria de Aveiro (AUC $\approx$ 0.8942).

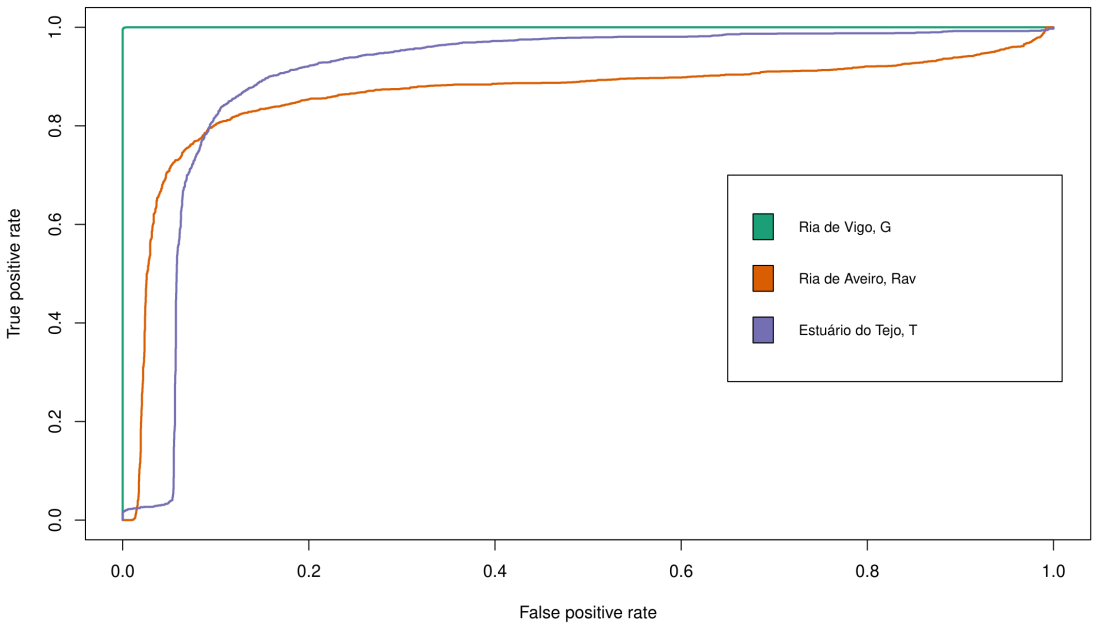
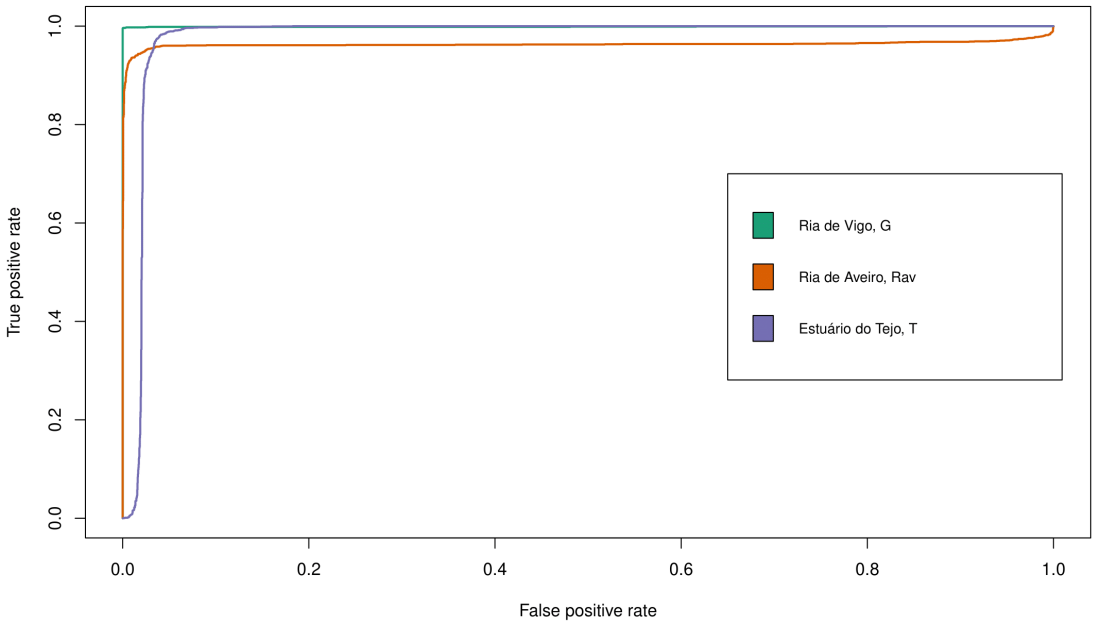
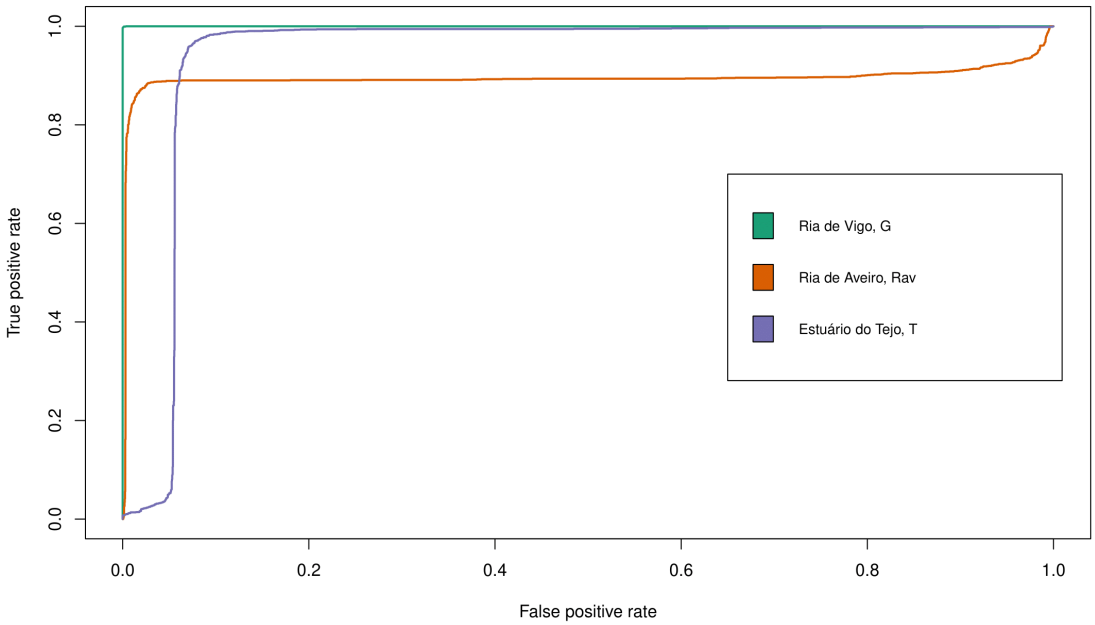



Figure 4.14: Graphical comparison of the ROC *curves* between the three classes of the response variable, conditioned to the *Ridge* regularization method and considering the merged 6,000 observation *testing set*




(a) LASSO




(b) *Elastic Net*

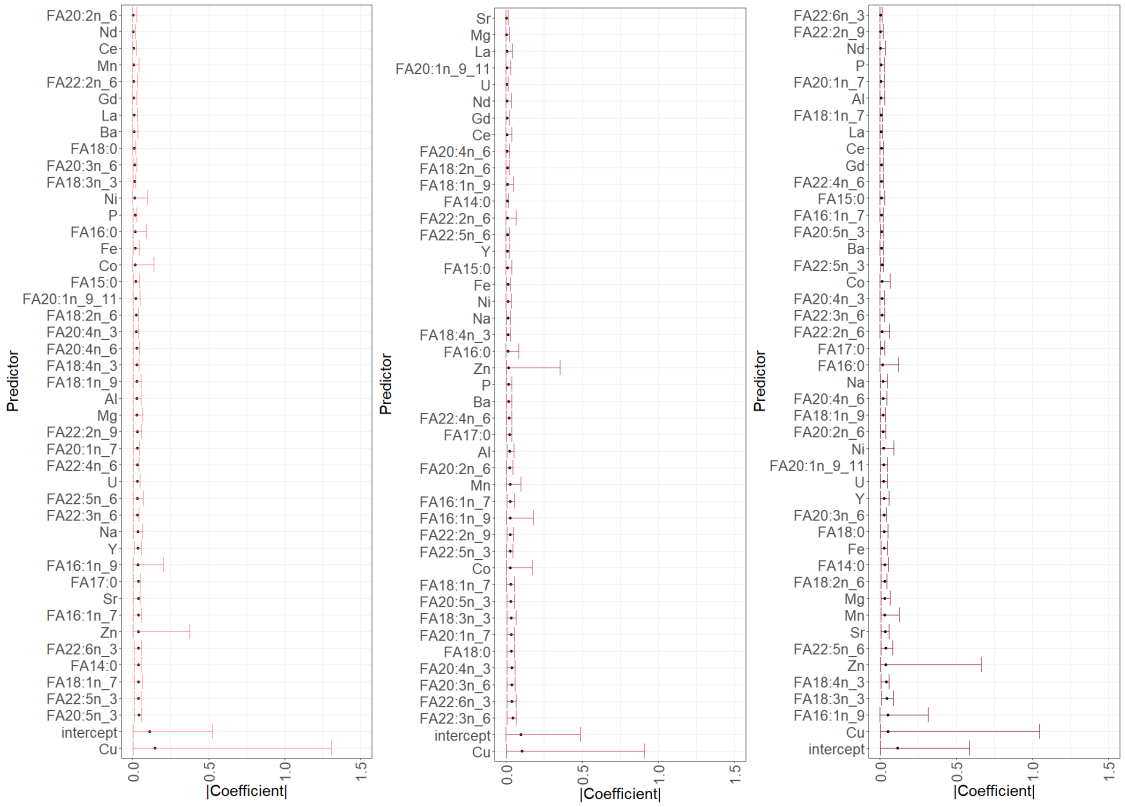
Figure 4.15: Graphical comparison of the ROC *curves* between the three classes of the response variable, conditioned to the regularization method ((a) LASSO, (b) *Elastic Net*) and considering the merged 6,000 observation *testing set*

### 4.2.4 Variable Importance

Due to the data standardization, to study the importance (or weight) of each variable on the prediction of the clams' location of origin, we considered the absolute value of their corresponding model coefficient. By doing so, we conclude that the most important variables are the ones that have the highest absolute coefficient values. Additionally, because regularized *Multinomial Logistic Regression* returns, for each selected variable, a coefficient per class, we have the ability to choose to either study the variable effect on predicting each of the individual locations of origin, or the collective impact that the variable has on predicting the classes indiscriminately. For the sake of concluding about the individual class weight of the selected variables, we study the absolute values of the coefficients in each class. In order to conclude about the overall effect that a predictor has on the model, we can simply sum the absolute values of that predictor's coefficient of each class. In this subsection, we approach both of these methods of analyzing variable importance.

To perform a variable importance analysis on a single model, studying the velocity of convergence of the variables' coefficients to 0 also helped to define the importance of the different predictors in predicting the response's behavior. Once more, regard Figures 4.7, 4.8 and 4.9. The most important predictors have the slowest convergence, while the less important variables exhibit fast convergence. This signifies that, at very low penalties, the least important variables will already have small coefficients, contrary to the important variables. Analyzing the model derived from the first iteration of *Monte Carlo Cross-Validation*, we defined three levels of importance: high, medium and low. The thresholds were established through the quantile measures of the absolute values of the model's coefficients at the lowest considered penalty, conditioned to the three different regularization methods. A variable was termed important, if its absolute coefficient was higher that the $3rd$ quantile of the model's absolute coefficients; variables of medium importance had coefficients whose absolute values fell between the $2nd$ and $3rd$ quantiles; the remaining variables were considered of low importance. Tables C.1, C.2 and C.3 display the different variables' importance qualification, conditioned to the regularization techniques.

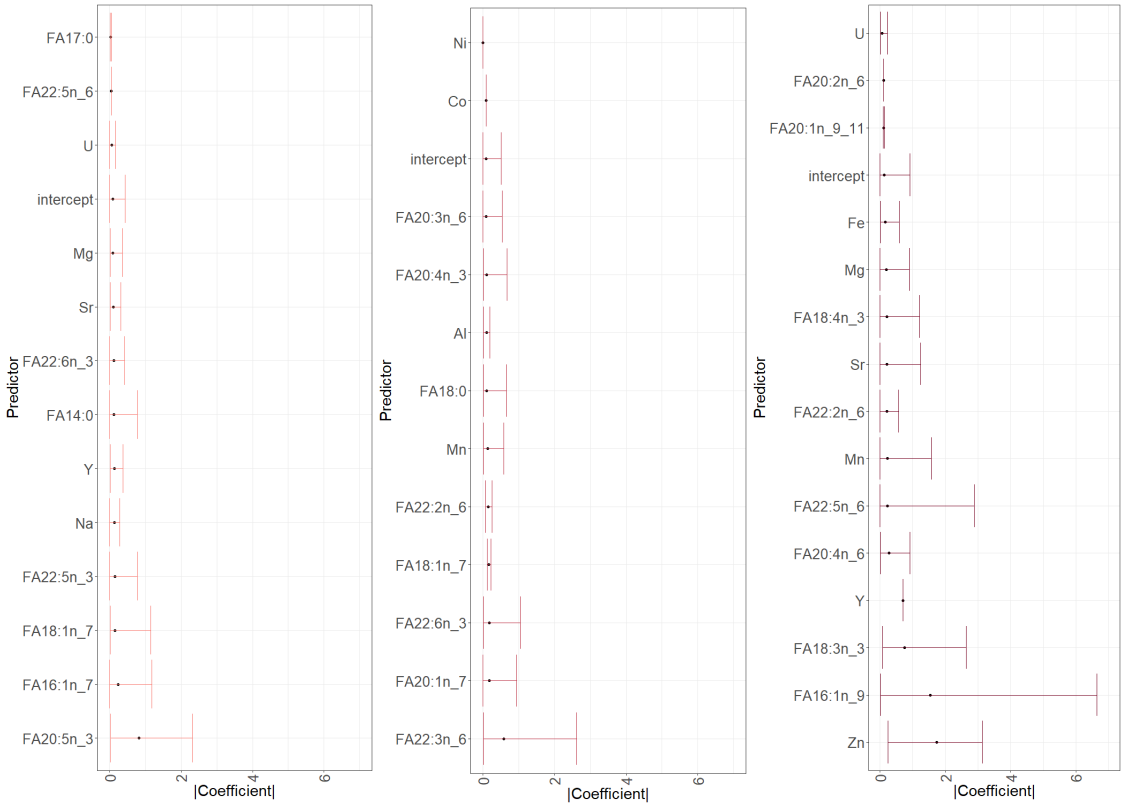
To perform a variable importance analysis on the 1,000 models adjusted considering the three regularization methods, we studied the range in which the absolute values of the coefficients varied. This also helped better understand the precision in which the coefficients were estimated - high precision leads to a small range of value variation, while low precision conducts the opposite. To perform this analysis, we constructed graphic representations of the range of values for each selected predictor's coefficient along the 1,000 models, for each regularization method and class of the response variable. Regarding the *Ridge* method (Figure 4.16), it is important to notice that, because this regularization method does not perform feature selection, we obtained intervals for the 44 variables. Features like `Cu`, `Zn` and `FA16:1n_9` displayed low precision for all of the classes of the response variable, given how the range of values they varied in was significantly superior compared to the other variables. Nevertheless, most predictors' coefficients appeared to have been estimated with high precision.




(a) Ria de Vigo (b) Ria de Aveiro (c) Estuário do Tejo

Figure 4.16: Range of the absolute values of *Ridge*'s selected predictors' coefficients for predicting (a) Ria de Vigo, (b) Ria de Aveiro, (c) Estuário do Tejo, along the 1,000 models. For each interval, the black colored dot represents the average absolute value of the corresponding variable coefficient, throughout the 1,000 models. The predictors are displayed in ascending order of their average absolute coefficient value

With respect to the LASSO method (Figure 4.17), notice how, because it performs robust feature selection, the amount of variables selected to predict each of the clams' location of origin throughout the 1,000 models was very diminished compared to the *Ridge* method. We observed how, in comparison to *Ridge*, the coefficient estimations appeared to be much less precise. By analyzing which predictors were chosen to explain each class of the response variable across the 1,000 models, we monitored how the biochemical fingerprints of the clams seemed to have a higher relevancy in predicting the clams' location, comparatively to the geochemical fingerprints. To predict Ria de Vigo, features like FA20:5n_3, FA16:1n_7, FA18:1n_7, FA22:5n_3 and FA14:0 had associated low precision. As for Ria de Aveiro, that was the issue for FA22:3n_6, FA20:1n_7 and FA22:6n_3. Lastly, for Estuário do Tejo, the same issue lied with most of the selected features such as Zn, FA16:1n_9, FA18:3n_3, FA20:4n_6, FA22:5n_6, Sr, FA18:4n_3 and Mg. Note how, for predicting Estuário do Tejo, LASSO estimated the model coefficients for the selected features with less precision than when predicting the other two classes.

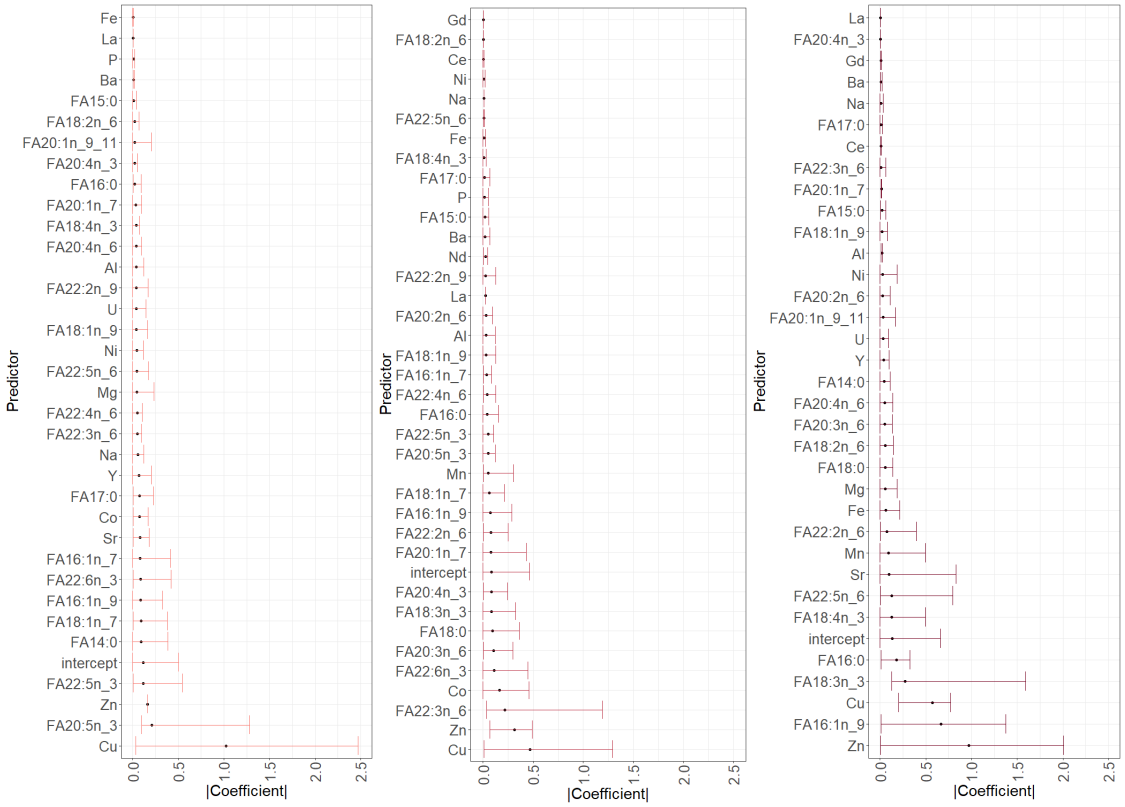



(a) Ria de Vigo (b) Ria de Aveiro (c) Estuário do Tejo

Figure 4.17: Range of the absolute values of LASSO's selected predictors' coefficients for predicting (a) Ria de Vigo, (b) Ria de Aveiro, (c) Estuário do Tejo, along the 1,000 models. For each interval, the black colored dot represents the average absolute value of the corresponding variable coefficient, throughout the 1,000 models. The predictors are displayed in ascending order of their average absolute coefficient value

Finally, considering the models generated by the *Elastic Net* method (Figure 4.18), we observed how this method selects less variables than *Ridge*, but more than LASSO, being somewhat partway from both of these methods, in terms of variable selection. In comparison to *Ridge*, the coefficient estimations done through *Elastic Net* appeared to be less precise, although, when comparing to LASSO, they were, in general, more precise. Again, we monitored the higher relevancy of the biochemical fingerprints of the clams to predict their location of origin. To predict Ria de Vigo, features like `Cu`, `FA20:5n_3`, `FA22:5n_3`, `FA14:0`, `FA18:1n_7`, `FA16:1n_9`, `FA22:6n_3` and `FA16:1n_7` had associated low precision. As for Ria de Aveiro, that was seen for `Cu`, `Zn`, `FA22:3n_6`, `Co`, `FA22:6n_3`, `FA20:3n_6`, `FA18:0`, `FA18:3n_3`, `FA20:4n_3`, `FA20:1n_7`, `FA22:2n_6`, `FA16:1n_9` and `Mn`. Lastly, for Estuário do Tejo, the same issue lied with features such as `Zn`, `FA16:1n_9`, `Cu`, `FA18:3n_3`, `FA16:0`, `FA18:4n_3`, `FA22:5n_6` `Sr`, `Mn` and `FA22:2n_6`.

With regards to coefficient estimation precision, we can conclude how, in general, *Ridge* method produces the estimates with the best precision, followed by *Elastic Net* and then LASSO.




(a) Ria de Vigo (b) Ria de Aveiro (c) Estuário do Tejo

Figure 4.18: Range of the absolute values of *Elastic Net*'s selected predictors' coefficients for predicting (a) Ria de Vigo, (b) Ria de Aveiro, (c) Estuário do Tejo, along the 1,000 models. For each interval, the black colored dot represents the average absolute value of the corresponding variable coefficient, throughout the 1,000 models. The predictors are displayed in ascending order of their average absolute coefficient value

To complete this analysis, regard Tables D.1 through D.9. These tables display, considering the distribution of values of the model coefficients throughout the 1,000 models, the probability of variable selection (prob), and the *average* ($\overline{|\beta|}$), *standard error of the mean* - SEM (SEM($|\beta|$)), *minimum* (min($|\beta|$)) and *maximum* (max($|\beta|$)) of the absolute values of each selected predictor's coefficient. We observe how the coefficient estimation through the LASSO method led to higher *standard error of the mean* values. Because this measure assesses how far the sample mean of the data is likely to be from the true population mean, this signifies that there is a higher uncertainty around the LASSO's estimate of the mean absolute coefficient values. Additionally, considering the tables that regard the LASSO method (D.4, D.5 and D.6), we see how, for each class of the response variable, only 2-3 predictors have a high probability of being selected. Meanwhile, *Elastic Net* (D.7, D.8 and D.9) selects 5-10 predictors with high probability. Despite this fact, the variables that LASSO selects with high probability have also a high chance of being selected by *Elastic Net*.

To finalize the individual class coefficient analysis, note how, regardless of the regularization method applied, the variable importance of each predictor changes significantly when

considering the prediction of different locations of origin of the clams.

Subsequent to studying the range of values of the model coefficients, throughout the 1,000 models, for each individual class of the response variable, we performed an analysis on the variable importance without discriminating between the different locations of origin of the clams. This way, for each of the 1,000 models, we measured the predictors' importance as defined in equation 3.61. We aimed to understand if there were variables that verified having a high variable importance more often than others, throughout the 1,000 models of each regularization method. In order to perform this analysis, for each model, we selected the 10 most important predictors. Afterwards, we combined these results to check for patterns in the top 10 most important variables across the 1,000 models of each method. This study was useful to conclude if there were any predictors, and if so, which, that were most frequently selected as having a heavy influence for the prediction of the clams' geographic origin. In Figure 4.19, we observe how, for the three regularization methods, the probability of fatty acids FA18:3n_3, FA20:5n_3 and FA22:3n_6 belonging to the top 10 most relevant variables was very high. It is also important to remark that, again, the geochemical fingerprints of the clams appear to not be as relevant for predicting their location of origin, when compared to plenty of fatty acids.

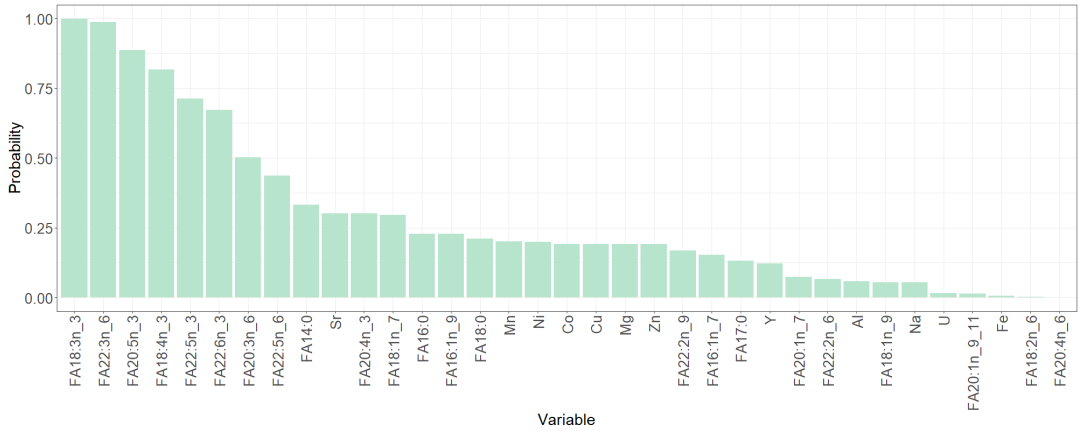
### 4.2.5 Residual Analysis

As mentioned in Section 3.2, in *Multinomial Logistic Regression* models, a residual measures how close the fitted probabilities of class affiliation of the train observations are to predicting the the actual class they belong in. Consider the following hypothetical example: the sets of probabilities resulting from the output of *Ridge*, LASSO and *Elastic Net* for the same train observation, that we know beforehand belongs to the Ria de Aveiro class, are:

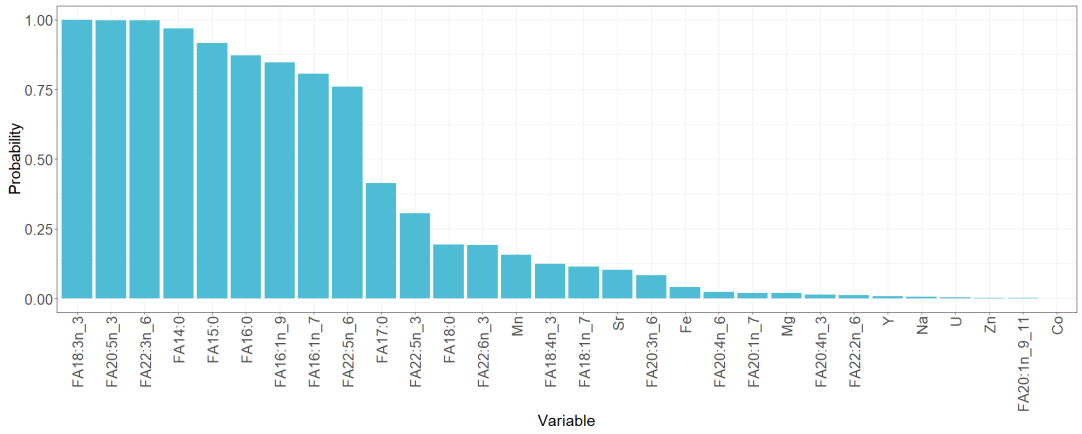
- **Ridge:** ( G = 0.3, Rav = 0.4, T = 0.3)
- **LASSO:** ( G = 0.2, Rav = 0.7, T = 0.1)
- **Elastic Net:** ( G = 0.3, Rav = 0.5, T = 0.2)

It is clear that, because prior to fitting any model we knew that the observation belonged to the Ria de Aveiro class, in this case, LASSO is the method that produces the finest set of probabilities (because comparatively to the other two methods, LASSO's probability estimate of the observation belonging to the correct class is the highest), followed by *Elastic Net* and then *Ridge*. For each of these methods, we define the residual for this observation as 1 minus the output probability of that observation belonging to the Ria de Aveiro class, that is:

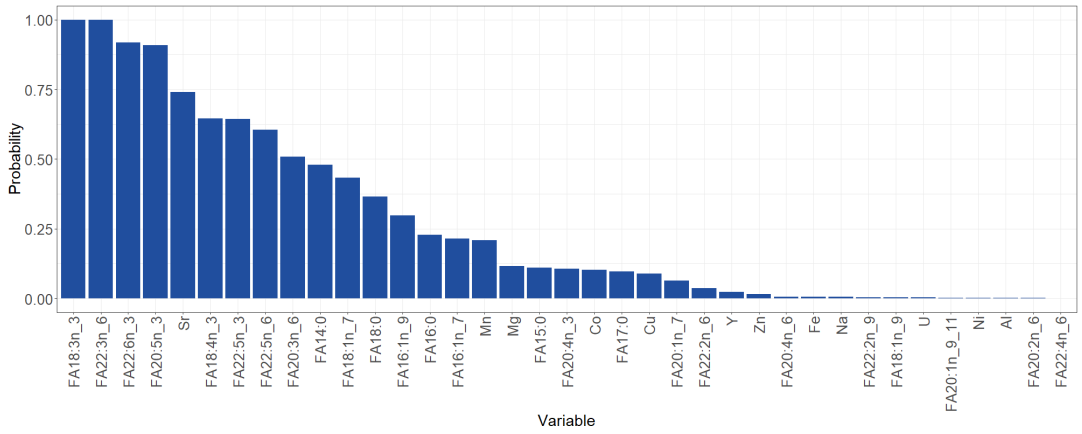
- **Ridge:** $r_{Ridge} = 1 - 0.4 = 0.6$
- **LASSO:** $r_{LASSO} = 1 - 0.7 = 0.3$
- **Elastic Net:** $r_{ElasticNet} = 1 - 0.5 = 0.5$,




(a) *Ridge*




(b) LASSO




(c) *Elastic Net*

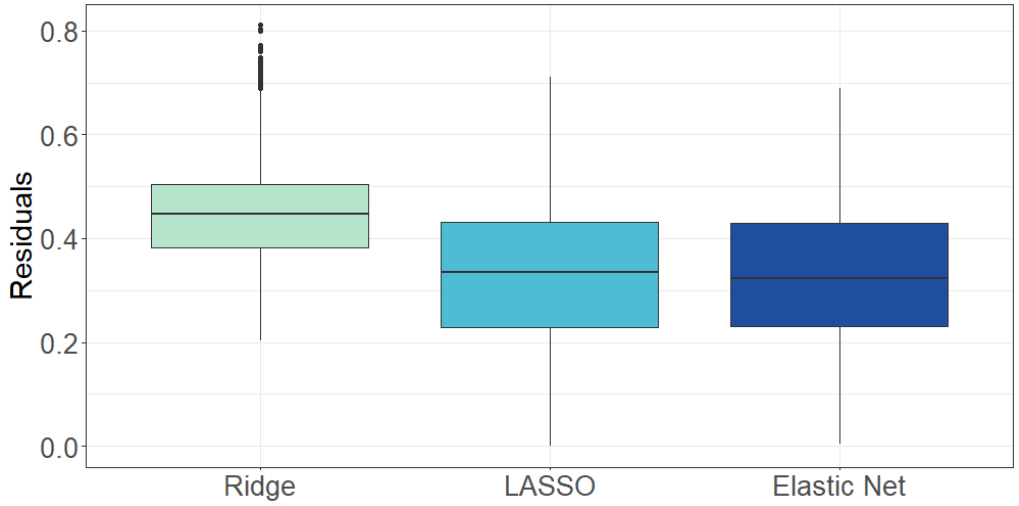
Figure 4.19: Probability of a variable being selected for the top 10 features with highest variable importance throughout the 1,000 models, conditioned to the regularization method: (a) *Ridge*, (b) LASSO, (c) *Elastic Net*. The variables are displayed in descending order of probability of selection

reiterating the conclusion that, since LASSO has the lowest residual, it is the method that best fits this observation, followed by *Elastic Net* and then *Ridge*.

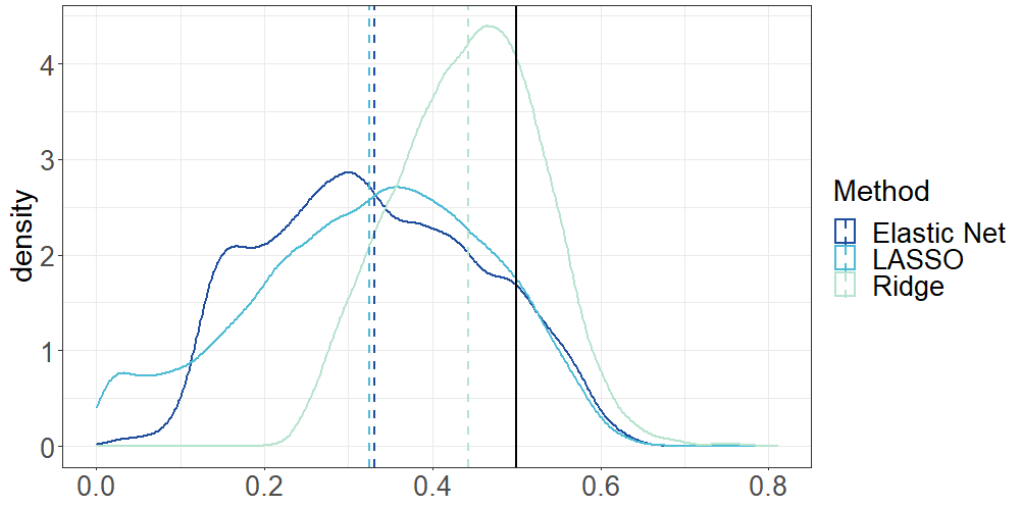
It is important to remark that, because a method performs better for a single observation, it does not mean that it fits the best models, since we need to accommodate the entirety of the training set. Furthermore, we also need to take into consideration the 1,000 different fit models. In the present case, for each of the 1,000 iterations, we divided the dataset into *training*/*testing sets* with a ratio of 80%/20%, leading to *training sets* of 24 observations and *testing sets* of 6 observations. By this reason, for each of the 1,000 models of each regularization method, we had 24 residual values. Using the same reasoning as in previous analysis, we combined the 1,000 *training sets* into one large *training set* composed of $24 \times 1,000 = 24,000$ observations, in order to better understand the differences in residuals between the three methods.

Regard Figure 4.20(a). The median residual produced when fitting a *Ridge* model was higher than the median LASSO or *Elastic Net* residual. Both in LASSO and *Elastic Net* more than 25% of the residuals had values below 0.2, while *Ridge* could not achieve residual values below this same threshold. Additionally, notice how in *Ridge*, around 25% of the residuals had values above 0.5, meaning that, considering the 1,000 *Ridge* models, the correct class failed to be predicted at least 25% of the times. With respect to Figure 4.20(b), notice how the residual distributions of LASSO and *Elastic Net* methods appeared to be very similar, contrary to *Ridge*'s, which had a completely different shape and location: more narrow and located around higher residual values. We see how the density curves for LASSO and *Elastic Net* displayed a more flattened property, which in theory signifies these methods' residuals were more scattered, while *Ridge*'s residuals were more condensed within a smaller range of values. To complete the residual analysis, and as a measure of predictive performance of the models, a vertical continuous line is shown on the density curves plot, representative of the 0.5 residual threshold value. Theoretically, the models that have a higher concentration of residual values below the 0.5 threshold, have a better predictive performance compared to the models that have a higher concentration of residuals above this same threshold. Therefore, again, we verify that, out of the three regularization methods, *Ridge* displayed the poorest performance.

Besides comparing the residual behavior between methods, we also studied how they varied between the three different classes of the response variable, conditioned to the regularization method. Regard Figure 4.21. On average, for the three regularization methods, we observe that the Ria de Vigo class (G) produced models with the lowest residuals, compared to the other two classes of the response variable. Around 75% of the *Ridge* residuals from Ria de Vigo were lower than 0.4, while, for the other two classes, around 75% were higher than this same value. Similarly, for the *Elastic Net* regularization method, 50% of the residuals produced for Ria de Vigo were below 0.3, whereas 75% of the residuals for Ria de Aveiro (Rav) and Estuário do Tejo (T) classes were above this same value. Notice how the residual values in Ria de Aveiro and Estuário do Tejo had fairly similar distributions, especially for *Ridge* and *Elastic Net*, while Ria de Vigo's distribution curve was generally located around smaller residual values. Again, to complete the residual analysis, and as a measure of predictive performance of the models, a vertical continuous line is shown on the density curves plot, representative of the 0.5 residual



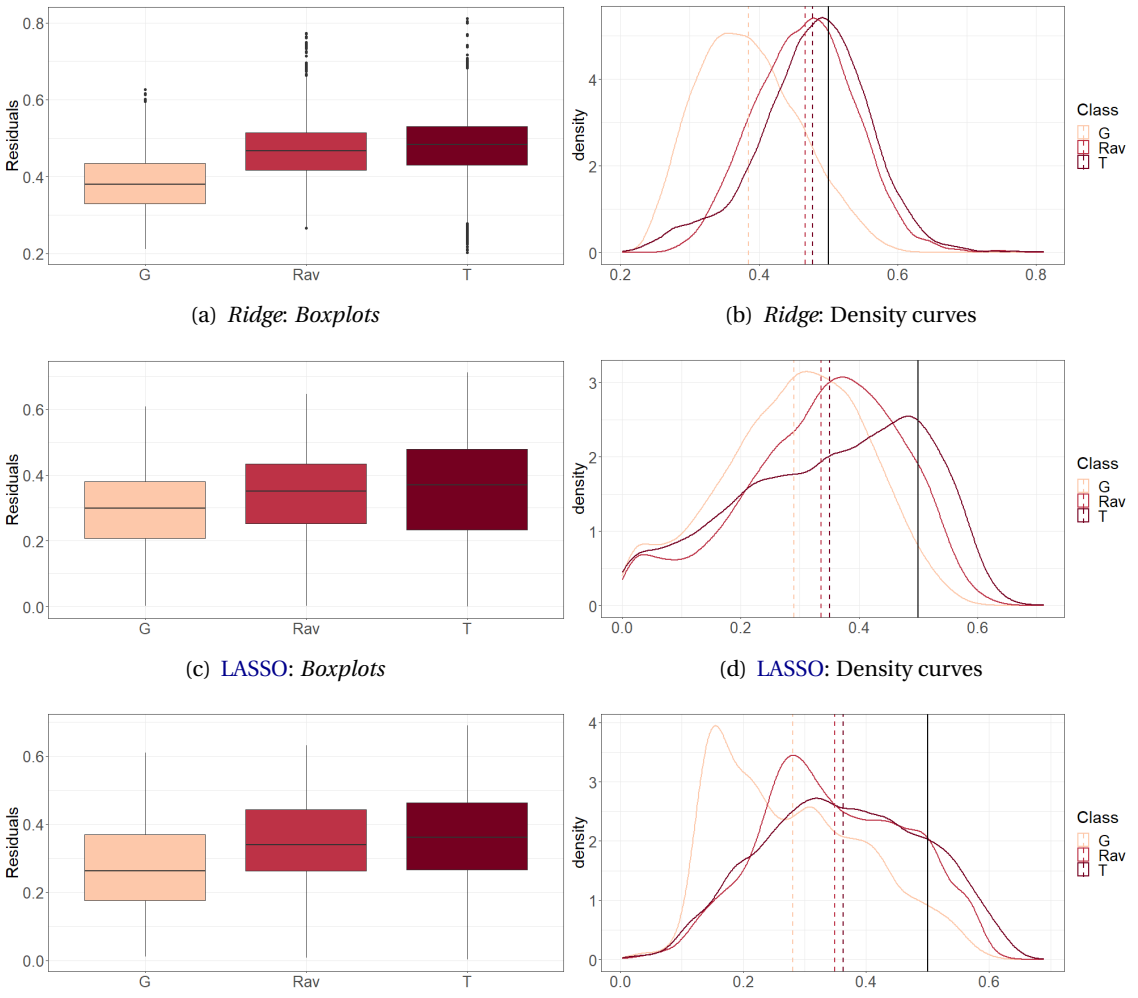

(a) *Boxplots*




(b) Density curves

Figure 4.20: (a) *Boxplots* and (b) Density curves of the model residuals, considering the three regularization methods, and the merged 24,000 observations training set. On the density curves plots, there are displayed 4 vertical lines: 3 dotted lines representative of the mean residual values in each regularization method, 1 continuous line representative of the 0.5 residual threshold value

threshold value. We verify, once more, that, out of the three classes of the response variable, *Ria de Vigo*, G, showed to be the class with the best predictive quality, regardless of the regularization method.




(a) *Ridge*: *Boxplots*

(b) *Ridge*: Density curves

(c) LASSO: *Boxplots*

(d) LASSO: Density curves

(e) *Elastic Net*: *Boxplots*

(f) *Elastic Net*: Density curves

Figure 4.21: *Boxplots* ((a), (c), (e)) and density curves ((b), (d), (f)) of the residuals of *Ridge* ((a), (b)), LASSO ((c), (d)) and *Elastic Net* ((e), (f)), distinguishing between the three classes of the response variable. On the density curves plots, there are displayed 4 vertical lines: 3 dotted lines representative of the mean residual values in each class of the response variable, 1 continuous line representative of the 0.5 residual threshold value

# 5

# CONCLUSION

The main interest of this thesis was the application regularization methods to bypass the complications linked to high-dimensional data and problem-solving. With the intention of showing how the development of such methods has been a breakthrough in plenty scientific fields, we proved, mathematically, the reason why high-dimensional problems lead to complexities within the domain of *Linear Modeling* and *Generalized Linear Modeling*. In short, it concerns the non-invertibility of the *Fisher Information Matrix* when $p > n$, $p$ being the number of predictors and $n$ being the number of observations of the contemplated data. The trouble arises when estimating the model's coefficients, as it entails calculating the inverse of the *Fisher Information Matrix*, which does not exist when a dataset has more variables than data points.

To study the performance of three regularization methods, *Ridge*, LASSO and *Elastic Net*, we applied them to a set of data containing detailed information on the biochemical and geochemical fingerprints of *Ruditapes philippinarum*, with the purpose of predicting the location of origin of this species: Ria de Vigo in Galiza, Spain, Ria de Aveiro in Aveiro, Portugal, or Estuário do Tejo in Lisbon, Portugal. Finding which features are most relevant for predicting a certain event is essential, as attaining information on each feature can be time and cost demanding. Besides identifying which features have a higher weight on predicting the specie's location of origin, we confirmed that *Ridge* produces very dense models, since it does not perform feature selection, LASSO can sometimes be a slightly unstable method, perhaps due to how it addresses groups of highly correlated variables, and that *Elastic Net* is, to some extent, a midway between *Ridge* and LASSO. Given the small sample size, obtaining these results was enabled through the application of *Monte Carlo Cross-Validation*.

The amount of features selected by a LASSO model was very small compared to an *Elastic Net* Elastic Net model, referencing the fact that, because the dataset is composed of various groups of highly correlated variables, LASSO addresses this issue by selecting one predictor from each group and dropping the remaining.

Regardless of the regularization method, we reached that the location of Ria de Vigo was correctly predicted most frequently. While the number of times that Ria de Aveiro class was predicted was less than the actual Ria de Aveiro occurrences, for the Estuário do Tejo class the exact opposite occurred.

To analyze the predictive performance of the different regularization methods applied, we resorted to studying *Cross Entropy*, *Confusion Matrices* and ROC *curves*. Concerning *Cross Entropy*, we arrived at the conclusion that, on average, *Elastic Net* was the method with the best *testing Cross Entropy*, followed by *Ridge*, and then LASSO. Strangely enough, when we computed the *training Cross Entropy*, we found that LASSO was the method that, on average, had the best *training Cross Entropy*, followed by *Elastic Net* and then *Ridge*. After computing graphical representations for both *testing* and *training Cross Entropy* values, we observed the instability of LASSO in terms of *testing Cross Entropy* but not *training Cross Entropy*. Additionally, concerning *training Cross Entropy*, *Ridge* produced the models with the highest values, while LASSO and *Elastic Net* behaved in a similar way.

*Confusion Matrices* facilitate the computation of performance measures such as *Accuracy*, *Misclassification Rate*, *Precision*, *Recall*, *Specificity* and a series of *F1-Score* measures. This performance analysis technique allows us to both study the overall models, not differentiating between the classes of the response variable, but also conduct a single class analysis. When studying the predictive performance of the individual classes, although all of them showed good predictive quality, we found that the prediction of Ria de Vigo was oftentimes superior compared to the other two classes, which behaved more or less identically. The computation of measures like *Micro F1*, *Macro F1* and *Weighted F1* assess predictive performance for the overall models, enabling a comparative study between the three regularization methods. We observed that, in terms of these *F1-Scores*, LASSO accomplished the best predictive models, followed by *Elastic Net* and then *Ridge*.

ROC *curves* are typically employed in problems with a binary response variable, where the curve of False Positive Rate against True Positive Rate is drawn and used to calculate the area under the curve (AUC). For this reason, in our study, for each class of the response variable, we computed a ROC *curve* of that class against the other two, essentially transforming a classic *Multinomial* model into three individual *Logistic* models. Through analyzing the AUC values, we again found that Ria de Vigo had the best predictive quality out of the three classes, followed by Estuário do Tejo and then Ria de Aveiro, regardless of the regularization method. Additionally, LASSO produced the ROC *curves* with the highest AUC values for all of the classes of the response variable.

We found that *Ridge* produced the models with the highest residuals, while LASSO and *Elastic Net* behaved in a similar way. Considering the distinction between the three classes of the response variable, we found that, for all of the regularization methods, Ria de Vigo led to the smallest residuals, while the other two classes displayed an identical behavior.

We showed that the *Ridge* is, overall, the poorest performing method when addressing this high-dimensional classification problem. Between LASSO and *Elastic Net*, none of them is emphasized as the best method, as they both outperform each other in different aspects. Even so, the number of features selected by LASSO was much smaller compared to *Elastic Net*, perhaps due to the existence of groups of highly correlated variables in the dataset.

The location of Ria de Vigo was the one correctly predicted most frequently, regardless of the regularization method employed. The prediction of Ria de Aveiro and Estuário do Tejo behaved

somehow equally.

It was clear that fatty acids FA18:3n_3, FA20:5n_3 and FA22:3n_6 played a significant role in predicting the geographic origin of Manila clams. We also found that the biochemical fingerprints of the adductor muscle of this bivalve (fatty acids) had a higher impact in predicting its origin when compared to the geochemical fingerprints of their shell (chemical elements).

Overall, our analysis relied heavily on the fact that the information provided on the location assigned to each of the 30 samples of clams is truthful, but that might not always be the case. Future studies should address scenarios where none of the locations known appears to be a viable option to generate acceptable results. Additionally, it will also be relevant to verify if the predictive perform between the three methods, much like the quality of prediction of the different locations, varies if confronted with a higher amount of data instances. By fine-tuning these approaches, one will make a valuable contribution towards the fight against illegal, unreported, and unregulated (IUU) fishing, one of the major threats to achieve United Nations Sustainable Development Goal (SDG) 14 – Life Below Water (FAO, 2021).

## References

Algamal, Z. Y., & Lee, M. H. (2015). Regularized logistic regression with adjusted adaptive elastic net for gene selection in high dimensional cancer classification. *Computers in biology and medicine*, *67*, 136–145.

Aseervatham, S., Antoniadis, A., Gaussier, É., Burlet, M., & Denneulin, Y. (2011). A sparse version of the ridge logistic regression for large-scale text categorization. *Pattern Recognition Letters*, *32*(2), 101–106.

Asenso, T. Q., Zhang, H., & Liang, Y. (2020). Pliable lasso for the multinomial logistic regression. *Communications in Statistics-Theory and Methods*, 1–16.

Astill, J., Dara, R. A., Campbell, M., Farber, J. M., Fraser, E. D., Sharif, S., & Yada, R. Y. (2019). Transparency in food supply chains: A review of enabling technology solutions. *Trends in Food Science & Technology*, *91*, 240–247.

Bennion, M., Morrison, L., Brophy, D., Carlsson, J., Abrahantes, J. C., & Graham, C. T. (2019). Trace element fingerprinting of blue mussel (mytilus edulis) shells and soft tissues successfully reveals harvesting locations. *Science of The Total Environment*, *685*, 50–58.

Bennion, M., Morrison, L., Shelley, R., & Graham, C. (2021). Trace elemental fingerprinting of shells and soft tissues can identify the time of blue mussel (mytilus edulis) harvesting. *Food Control*, *121*, 107515.

Bolón-Canedo, V. (2014). Novel feature selection methods for high dimensional data.

Bull, S. B., Mak, C., & Greenwood, C. M. (2002). A modified score function estimator for multinomial logistic regression in small samples. *Computational Statistics & Data Analysis*, *39*(1), 57–74.

DeWitt, P. (2019). *Ensr: Elastic net searcher* [R package version 0.1.0].

Fagerland, M. W., Hosmer, D. W., & Bofin, A. M. (2008). Multinomial goodness-of-fit tests for logistic regression models. *Statistics in medicine*, *27*(21), 4238–4253.

Fan, J., & Fan, Y. (2008). High dimensional classification using features annealed independence rules. *Annals of statistics*, *36*(6), 2605.

Fan, J., & Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(5), 849–911.

Fan, J., Samworth, R., & Wu, Y. (2008). Ultrahigh dimensional variable selection: Beyond the linear model. *arXiv preprint arXiv:0812.3201*.

FAO. (2021). 14.4.1 Fish stocks sustainability | Sustainable Development Goals | Food and Agriculture Organization of the United Nations [Online; accessed 6 October 2021].

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, *33*(1), 1.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *science*, *286*(5439), 531–537.

Grau, J., Grosse, I., & Keilwagen, J. (2015). Prroc: Computing and visualizing precision-recall and receiver operating characteristic curves in r. *Bioinformatics*, *31*(15), 2595–2597.

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Hastie, T., Qian, J., & Tay, K. (2016). An introduction to glmnet.

Hilt, D. E., & Seegrist, D. W. (1977). *Ridge, a computer program for calculating ridge regression estimates* (Vol. 236). Department of Agriculture, Forest Service, Northeastern Forest Experiment ...

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

Jung, Y. (2016). Efficient tuning parameter selection by cross-validated score in high dimensional models.

Kuhn, M. (2020). *Caret: Classification and regression training* [R package version 6.0-86].

Leal, M. C., Pimentel, T., Ricardo, F., Rosa, R., & Calado, R. (2015). Seafood traceability: Current needs, available tools, and biotechnological challenges for origin certification. *Trends in biotechnology*, *33*(6), 331–336.

Mai, Q., Zou, H., & Yuan, M. (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika*, *99*(1), 29–42.

Mamede, R., Ricardo, F., Abreu, M. H., da Silva, E. F., Patinha, C., & Calado, R. (2021). Spatial variability of elemental fingerprints of sea lettuce (ulva spp.) and its potential use to trace geographic origin. *Algal Research*, *59*, 102451.

Mamede, R., Ricardo, F., Gonçalves, D., da Silva, E. F., Patinha, C., & Calado, R. (2021). Assessing the use of surrogate species for a more cost-effective traceability of geographic origin using elemental fingerprints of bivalve shells. *Ecological Indicators*, *130*, 108065.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, *5*(9), 1315–1316.

Morrison, L., Bennion, M., Gill, S., & Graham, C. T. (2019). Spatio-temporal trace element fingerprinting of king scallops (pecten maximus) reveals harvesting period and location. *Science of The Total Environment*, *697*, 134121.

Obuchi, T., & Kabashima, Y. (2016). Cross validation in lasso and its acceleration. *Journal of Statistical Mechanics: Theory and Experiment*, *2016*(5), 053304.

Ogutu, J. O., Schulz-Streeck, T., & Piepho, H.-P. (2012). Genomic selection using regularized linear regression models: Ridge regression, lasso, elastic net and their extensions. *BMC proceedings*, *6*(2), 1–6.

Ooi, H. (2021). *Glmnetutils: Utilities for 'glmnet'* [R package version 1.1.8].

Pereira, J. M., Basto, M., & da Silva, A. F. (2016). The logistic lasso and ridge regression in predicting corporate failure. *Procedia Economics and Finance*, *39*, 634–641.

Pino, A., & Morell, C. (2013). Analytical and experimental study of filter feature selection algorithms for high-dimensional datasets. *Fourth International Workshop on Knowledge Discovery, Knowledge Management and Decision Support*, 339–349.

R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., et al. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, *98*(26), 15149–15154.

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2011). Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, *39*(5), 1–13.

Sing, T., Sander, O., Beerenwinkel, N., & Lengauer, T. (2005). Rocr: Visualizing classifier performance in r. *Bioinformatics*, *21*(20), 7881.

Starkweather, J., & Moske, A. K. (2011). Multinomial logistic regression. *Consulted page at September 10th: http://www. unt. edu/rss/class/Jon/Benchmarks/MLR_JDS_Aug2011. pdf*, *29*, 2825–2830.

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

Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2003). Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, 104–117.

Turkman, M. A. A., & Silva, G. L. (2000). Modelos lineares generalizados–da teoria à prática. *VIII Congresso Anual da Sociedade Portuguesa de Estatıstica, Lisboa*.

Wang, H. (2009). Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*, *104*(488), 1512–1524.

Wang, Y. (2005). A multinomial logistic regression modeling approach for anomaly intrusion detection. *Computers & Security*, *24*(8), 662–674.

Wei, T., & Simko, V. (2017). *R package "corrplot": Visualization of a correlation matrix* [(Version 0.84)].

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York.

Xu, Q.-S., & Liang, Y.-Z. (2001). Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, *56*(1), 1–11.

Yang, S., Wen, J., Zhan, X., & Kifer, D. (2019). Et-lasso: A new efficient tuning of lasso-type regularization for high-dimensional data. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 607–616.

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. *Proceedings of the 20th international conference on machine learning (ICML-03)*, 856–863.

Zeng, Z., Zhang, H., Zhang, R., & Zhang, Y. (2015). A mixed feature selection method considering interaction. *Mathematical Problems in Engineering, 2015*.

Zhao, Z., & Liu, H. (2007). Searching for interacting features. *IJCAI International Joint Conference on Artificial Intelligence*.

Zhu, J., & Hastie, T. (2004). Classification of gene microarrays by penalized logistic regression. *Biostatistics, 5*(3), 427–443.

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.

# A

# LITERATURE REVIEW

Table A.1: Description of the studies mentioned in the Literature Review (see chapter 2) - part 1

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **20-NewsGroups (large-scale text categorization)** | | | | | |
| A compilation of emails written in the context of 20 different news groups, each assigned to a unique category according to its topic. | - 20 classes<br>- 12,492 training<br>- 6,246 testing | 51,666 ctg. | 1. LASSO<br>2. *Ridge*<br>3. *Selected Ridge* | 1. Outperformed by *Ridge*. The authors state that "the variable selection of the LASSO is too aggressive and eliminates interesting features."<br>2. 51,666 ctg.<br>3. $0.10 \times 51,666 \approx 5,167$ ctg. (same performance as *Ridge*) | Aseervatham et al., 2011 |
| **Bankruptcy** | | | | | |
| Dataset of 2,032 non-bankrupt firms and 401 bankrupt firms belonging to the hospitality industry, over the period 2010-2012. | - 2 classes<br>- 2,433 samples | "(...) values of certain indicators of the company." | 1. *Ridge*<br>2. LASSO | LASSO and *Ridge* perform in a similar way, though tend to favor the category of the dependent variable that appears with heavier weight in the training set. | Pereira et al., 2016 |

Table A.2: Description of the studies mentioned in the Literature Review - part 2

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **Brain Cancer** | | | | | |
| The goal is to select a few number of genes that are significantly sufficient to explain brain cancer. | - 2 classes<br>- 14 training<br>- 7 testing | 12,625 genes | 1. Correlation Feature Selection (CFS)<br>adequacy: Cross-Validation (CV)/ Distribution Optimally Balanced Stratified Cross Validation (DOB-SCV)<br>2. Fast Correlation-Based Feature (FCBF)<br>adequacy: CV/ DOB-SCV<br>3. INTERACT (INT)<br>adequacy: CV/ DOB-SCV | 1. 36/49 genes<br>2. 1/1 genes<br>3. 36/49 genes | Bolón-Canedo, 2014 |
| **Breast Cancer Prognosis** | | | | | |
| The goal is to select a few number of genes that are significantly sufficient to explain breast cancer prognosis: patients who remained disease-free for at least 5 years, and patients who developed distant metastases within 5 years. | - 2 classes<br>- 78 training<br>- 19 testing | 24,481 genes | 1. CFS<br>2. FCBF<br>3. INT | 1. 130 genes<br>2. 99 genes<br>3. 102 genes | Bolón-Canedo, 2014 |

Table A.3: Description of the studies mentioned in the Literature Review - part 3

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **Breast Carcinoma** | | | | | |
| Experimental samples gathered from cDNA microarrays to identify breast carcinoma based on variations from gene expression levels. | - 5 classes<br>- 60 training<br>- 25 testing | 456 genes | 1. *Pliable* LASSO<br>2. LASSO<br>3. *Elastic Net* | 1. 65 genes<br>2. 11 genes<br>3. 39 genes | Asenso et al., 2020 |
| **Central Nervous System Tumor** | | | | | |
| The goal is to select a few number of genes that are significantly sufficient to explain central nervous system tumors. | - 2 classes<br>- 40 training<br>- 20 testing | 7,129 genes | 1. CFS<br>adequacy: CV/ DOB-SCV<br>2. FCBF<br>adequacy: CV/ DOB-SCV<br>3. INT<br>adequacy: CV/ DOB-SCV | 1. 44/44 genes<br>2. 33/35 genes<br>3. 33/34 genes | Bolón-Canedo, 2014 |
| **Chinese Supermarket** | | | | | |
| Number of costumers and sale volumes of the variety of products in the shop from 2004 to 2005, to study the set of products that best influences the shop sales. | - Nr. of costumers<br>- Total of 464 days<br>- 60% training<br>- 40% testing | 6,398 prod. | 1. Efficient Tuning (ET) - LASSO<br>2. CV - LASSO<br>3. Bayesian Information Criterion (BIC) - LASSO<br>4. Estimation Stability Cross-Validation (ESCV) - LASSO | 1. 68 prod.<br>2. 111 prod.<br>3. 100 prod.<br>4. 72 prod. | H. Wang, 2009, Yang et al., 2019 |

Table A.4: Description of the studies mentioned in the Literature Review - part 4

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **Colon Cancer** | | | | | |
| The goal is to select a few number of genes that are significantly sufficient to explain colon cancer. | - 2 classes<br>- 42 training<br>- 20 testing | 2,000 genes | 1. CFS adequacy: CV/ DOB-SCV<br>2. FCBF adequacy: CV/ DOB-SCV<br>3. INT adequacy: CV/ DOB-SCV<br>4. *Elastic Net*<br>5. *Adaptive Elastic Net*<br>6. *AERidge*<br>7. *Adjusted Adaptive Elastic Net* | 1. 24/25 genes<br>2. 14/15 genes<br>3. 14/16 genes<br>4. 24 genes Standard error (se) ≈1.17<br>5. 24 genes se ≈1.08<br>6. 23 genes se ≈1.09<br>7. 28 genes se ≈0.94 | Bolón-Canedo, 2014, Algamal and Lee, 2015 |

Table A.5: Description of the studies mentioned in the Literature Review - part 5

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **DMOZ (large-scale text categorization)** | | | | | |
| A collection of html documents from the DMOZ website, which is an open directory project that aims to classify the whole web into categories. | - 3,503 classes<br>- 20,249 training<br>- 7,257 testing | 133,348 ctg. | 1. LASSO<br>2. *Ridge*<br>3. *Selected Ridge* | 1. "As expected in the case where the number of features is largely greater than the number of documents, the *Ridge* method clearly outperforms LASSO (...)"<br>2. 133,348 ctg.<br>3. "*Selected Ridge* performs better than the LASSO in terms of micro-F1, but however has a macro-F1 slightly lower than the value obtained by LASSO." | Aseervatham et al., 2011 |
| **Diffuse Infiltrative Glioma** | | | | | |
| The goal is to select a few number of genes that are significantly sufficient to explain diffuse infiltrative gliomas. | - 2 classes<br>- 57 train samples<br>- 28 test samples | 22,283 genes | 1. CFS<br>adequacy: CV/ DOB-SCV<br>2. FCBF<br>adequacy: CV/ DOB-SCV<br>3. INT<br>adequacy: CV/ DOB-SCV | 1. 141/156 genes<br>2. 116/118 genes<br>3. 117/123 genes | Bolón-Canedo, 2014 |

Table A.6: Description of the studies mentioned in the Literature Review - part 6

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **DLBCL - Diffuse Large B-Cell Lymphoma** | | | | | |
| The goal is to select a few number of genes that are significantly sufficient to explain DLBCL. | - 2 classes<br>- 32 training<br>- 15 testing | 4,026 genes | 1. CFS<br>adequacy: CV/ DOB-SCV<br>2. FCBF<br>adequacy: CV/ DOB-SCV<br>3. INT<br>adequacy: CV/ DOB-SCV | 1. 61/65 genes<br>2. 35/37 genes<br>3. 45/51 genes | Bolón-Canedo, 2014 |
| **DLBCL - Diffuse large B-cell lymphoma II** | | | | | |
| The study aims to select a small number of genes that are significantly sufficient to differentiate between diffuse large B-cell lymphoma and follicular lymphoma. | - 2 classes<br>- 77 samples | 7,129 genes | 1. *Elastic Net*<br>2. *Adaptive Elastic Net*<br>3. *AERidge*<br>4. *Adjusted Adaptive Elastic Net* | 1. 54 genes<br>se ≈ 1.25<br>2. 55 genes<br>se ≈ 1.12<br>3. 49 genes<br>se ≈ 1.11<br>4. 61 genes<br>se ≈ 1.04 | Algamal and Lee, 2015 |

Table A.7: Description of the studies mentioned in the Literature Review - part 7

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **Leukemia** | | | | | |
| Consists of a microarray problem where the goal is to construct a diagnostic rule based on the expression level of genes to predict the type of leukemia. | - 2 classes<br>- 38 training<br>- 34 testing | 7,129 genes | 1. *Golub*<br>2. Support Vector Machine (SVM) - Recursive Feature Elimination (RFE)<br>3. SVM - Univariate Ranking (UR)<br>4. Penalized Logistic Regression (PLR) - RFE<br>5. PLR -UR<br>6. Nearest Shrunken Centroids (NSC)<br>7. *Elastic Net*<br>8. Feature Annealed Feature Selection (FAIR) | 1. 50 genes<br>Misclassification Error (mce) = 4/34<br>2. 31 genes<br>mce = 1/34<br>3. 22 genes<br>mce = 3/34<br>4. 26 genes<br>mce = 1/34<br>5. 16 genes<br>mce = 3/34<br>6. 21 genes<br>mce = 2/34<br>7. 45 genes<br>mce = 0/34<br>8. 11 genes<br>mce = 1/34 | Zou and Hastie, 2005, J. Fan and Fan, 2008, Zhu and Hastie, 2004, Golub et al., 1999 |
| **Lung Cancer** | | | | | |
| Experimental samples to identify between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung based on variations from gene expression levels. | - 2 classes<br>- 32 training<br>- 149 testing | 12,533 genes | 1. NSC<br>2. FAIR | 1. 26 genes<br>mce = 11/149<br>2. 31 genes<br>mce = 7/149 | J. Fan and Fan, 2008 |

Table A.8: Description of the studies mentioned in the Literature Review - part 8

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **Neuroblastoma** | | | | | |
| The study contains information on patients of the German Neuroblastoma Trials, diagnosed between 1989 and 2004. The goal is to understand whether each patient survived 3 years after the diagnosis based on the gene expression levels. | - 2 classes<br>- 125 training<br>- 114 testing | 10,707 genes | 1. Sure Independence Rules (SIS)<br>2. Iterated Sure Independence Screening (FCBF)<br>3. var2-SIS<br>4. var2-FCBF<br>5. LASSO<br>6. NSC | 1. 5 genes<br>mce = 19/114<br>2. 23 genes<br>mce = 22/114<br>3. 10 genes<br>mce = 22/114<br>4. 12 genes<br>mce = 21/114<br>5. 57 genes<br>mce = 22/114<br>6. 9,413 genes<br>mce = 24/114 | J. Fan et al., 2008 |
| **Ohsumed (large-scale text categorization)** | | | | | |
| A collection of medical abstracts originally designed for content-based information retrieval. | - 23 classes<br>- 6,286 training<br>- 7,643 testing | 20,520 ctg. | 1. LASSO<br>2. *Ridge*<br>3. *Selected Ridge* | 1. "not only it has the best performance in terms of micro and macro-F1, but it also gives a very sparse solution."<br>2. 20,520 ctg.<br>3. $0.12 \times 20,520 \approx 2,462$ ctg. (same performance as *Ridge*) | Aseervatham et al., 2011 |

Table A.9: Description of the studies mentioned in the Literature Review - part 9

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **Ovarian Cancer** | | | | | |
| The goal is to select a few number of genes that are significantly sufficient to explain ovarian cancer. | - 2 classes<br>- 169 training<br>- 84 testing | 15,154 genes | 1. CFS<br>adequacy: CV/ DOB-SCV<br>2. FCBF<br>adequacy: CV/ DOB-SCV<br>3. INT<br>adequacy: CV/ DOB-SCV | 1. 35/33 genes<br>2. 27/26 genes<br>3. 32/31 genes | Bolón-Canedo, 2014 |
| **Prostate Cancer** | | | | | |
| The goal is to identify between prostate tumor samples and normal prostate samples based on variations from gene expression levels. | - 2 classes<br>- 102 training<br>- 34 testing | 12,600 genes | 1. NSC<br>2. FAIR<br>3. CFS<br>4. FCBF<br>5. INT | 1. 6 genes ( mce = 9/34)<br>2. 2 genes ( mce = 9/34)<br>3. 89 genes<br>4. 77 genes<br>5. 73 genes | J. Fan and Fan, 2008, Bolón-Canedo, 2014 |
| **Prostate Cancer II** | | | | | |
| The goal is to select a few number of genes that are significantly sufficient to differentiate between prostate tumor/non-tumor. | - 2 classes<br>- 102 samples | 5,966 genes | 1. *Elastic Net*<br>2. *Adaptive Elastic Net*<br>3. *AERidge*<br>4. *Adjusted Adaptive Elastic Net* | 1. 44 genes se ≈ 1.13<br>2. 44 genes se ≈ 1.07<br>3. 42 genes se ≈ 1.03<br>4. 48 genes se ≈ 0.87 | Algamal and Lee, 2015 |

Table A.10: Description of the studies mentioned in the Literature Review - part 10

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **Ramaswamy data** | | | | | |
| Consists of a set of tumor samples, spanning 14 tumor classes that account for 80% of new cancer diagnosis in the US. | - 4 classes<br>- 144 training<br>- 54 testing | 16,063 genes | 1. SVM -UR<br>2. PLR -UR<br>3. SVM - RFE<br>PLR - RFE | 1. 617 genes<br>mce = 12/54<br>2. 617 genes<br>mce = 12/54<br>3. 315 genes<br>mce = 9/54<br>4. 294 genes<br>mce = 9/54 | Zhu and Hastie, 2004 |
| **Reuters (large-scale text categorization)** | | | | | |
| A collection of news-wire articles covering different domains, where each document was manually assigned to one or more categories, according to its subject. | - 90 classes<br>- 7,770 training<br>- 3,019 testing | 6,760 ctg. | 1. LASSO<br>2. *Ridge*<br>3. *Selected Ridge* | 1. $0.0043 \times 6,760 \approx$ 29 ctg. (same performance as *Ridge*)<br>2. 6,760 ctg.<br>3. $0.05 \times 6,760 =$ 338 ctg. (worst performance) | Aseervatham et al., 2011 |
| **SMK** | | | | | |
| Gene expression data from smokers with and without lung cancer. This is diagnostic gene expression profile that could be used to distinguish between the two classes. | - 2 classes<br>- 125 training<br>- 62 testing | 19,993 genes | 1. CFS<br>adequacy: CV/ DOB-SCV<br>2. FCBF<br>adequacy: CV/ DOB-SCV<br>3. INT<br>adequacy: CV/ DOB-SCV | 1. 107/103 genes<br>2. 50/55 genes<br>3. 51/51 genes | Bolón-Canedo, 2014 |

Table A.11: Description of the studies mentioned in the Literature Review - part 11

| Summary | Response | Features | Methods | Findings | Ref. |
|---|---|---|---|---|---|
| **SRBCT - small, round blue cell tumors of childhood** | | | | | |
| These cancers belong to four distinct diagnostic categories and often present diagnostic dilemmas in clinical practice. The goal is to identify genes most relevant to the classification. | - 4 classes<br>- 63 training<br>- 20 testing | 2,308 genes | 1. *Pliable* LASSO<br>2. LASSO<br>3. *Elastic Net*<br>4. PLR -UR<br>5. PLR - RFE<br>6. SVM -UR<br>7. SVM - RFE | 1. 25 genes mce = 1/20<br>2. 12 genes mce = 1/20<br>3. 31 genes mce = 1/20<br>4. 15 genes mce = 0/20<br>5. 8 genes mce = 0/20<br>6. 21 genes mce = 0/20<br>7. 32 genes mce = 0/20 | Asenso et al., 2020, Zhu and Hastie, 2004 |

# B

# ROC CURVES

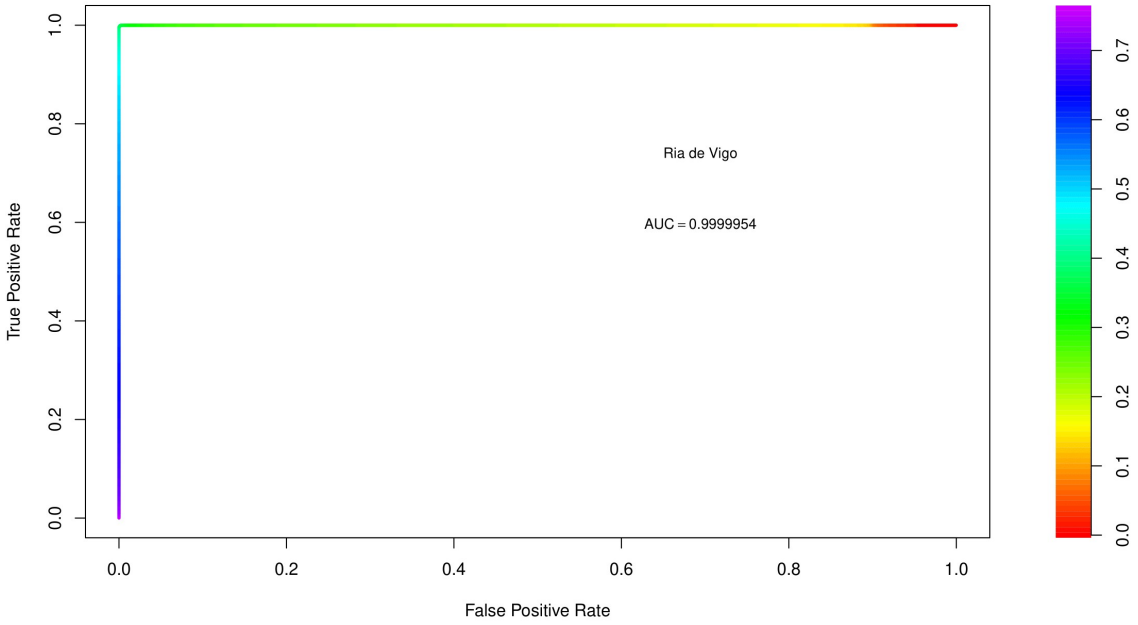



Figure B.1: Ria de Vigo ROC *curve* and AUC, considering the *Ridge* method of regularization and the merged testing set of 6,000 observations

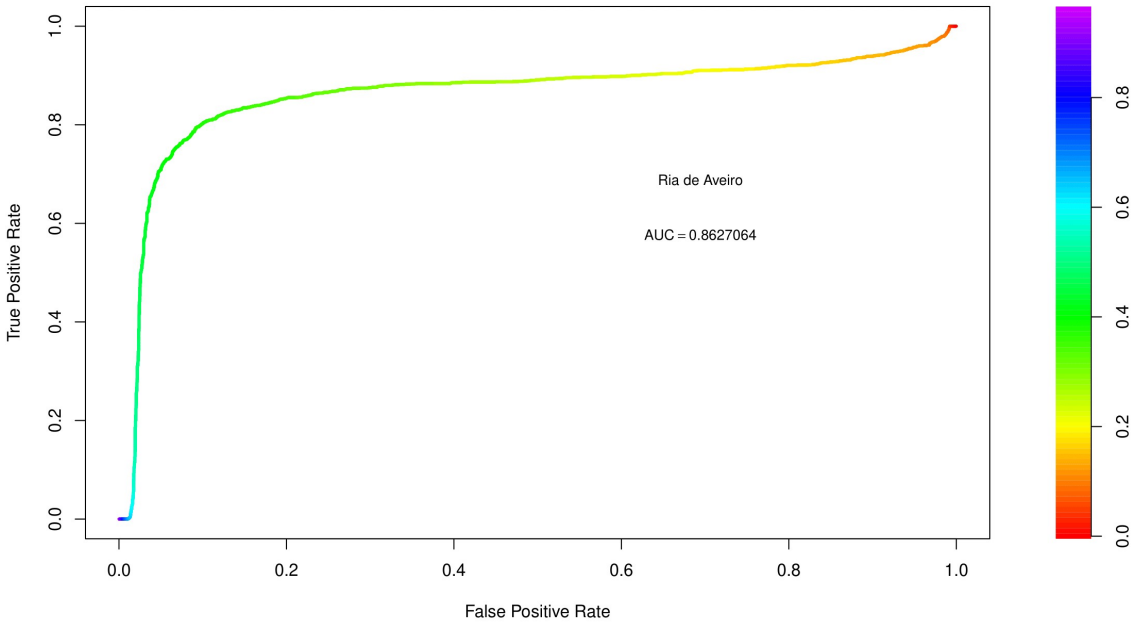



Figure B.2: Ria de Aveiro ROC *curve* and AUC, considering the *Ridge* method of regularization and the merged testing set of 6,000 observations

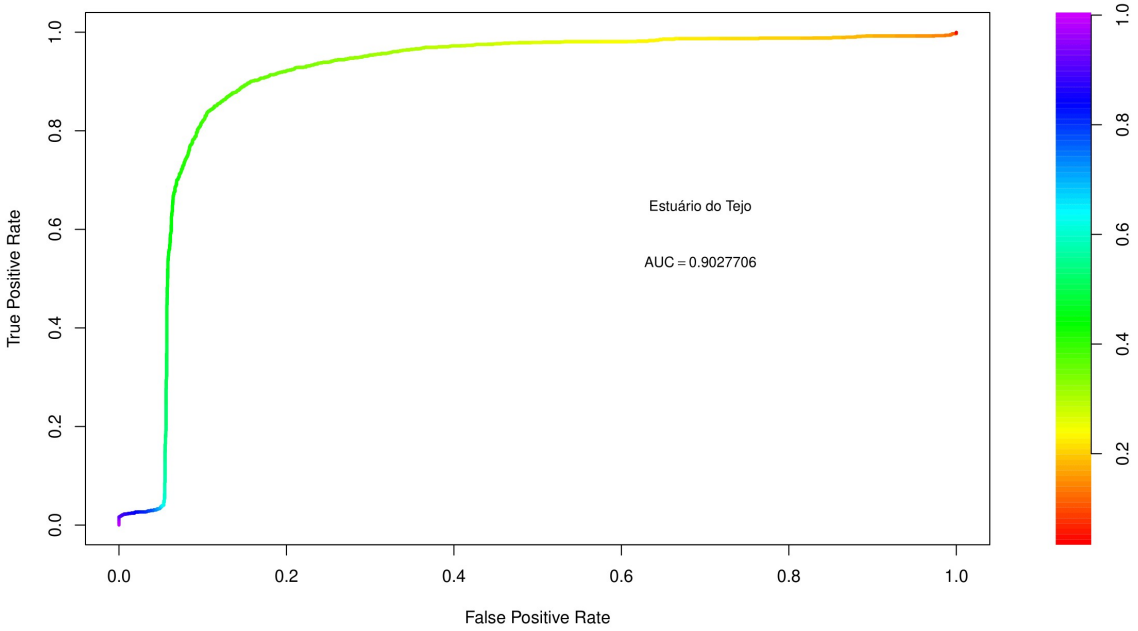



Figure B.3: Estuário do Tejo ROC *curve* and AUC, considering the *Ridge* method of regularization and the merged testing set of 6,000 observations

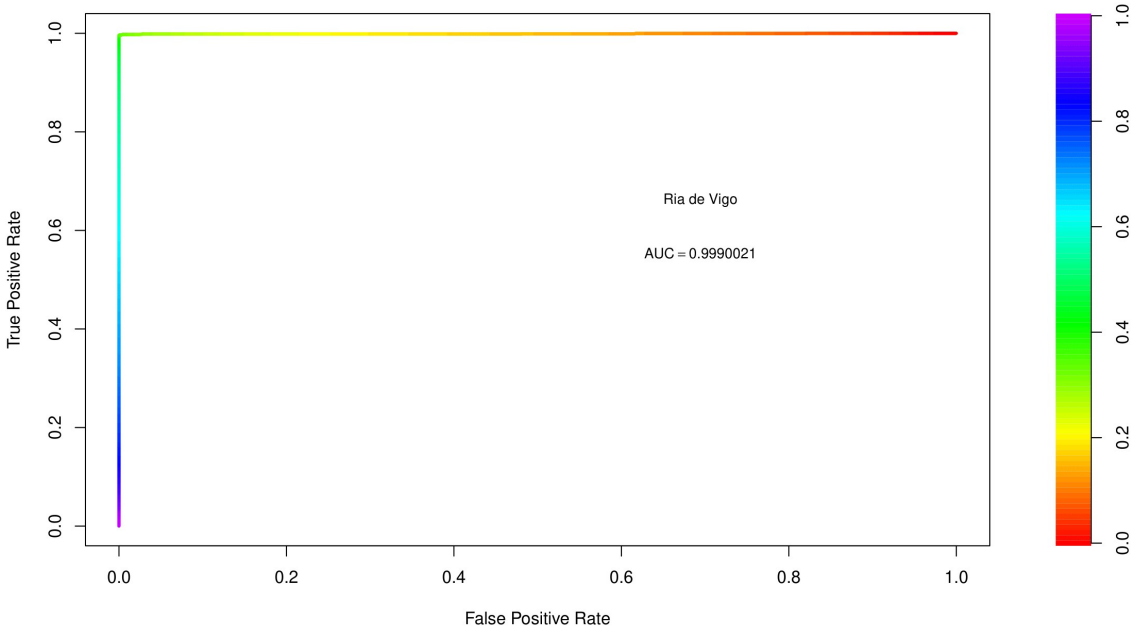



Figure B.4: Ria de Vigo ROC *curve* and AUC, considering the LASSO method of regularization and the merged testing set of 6,000 observations

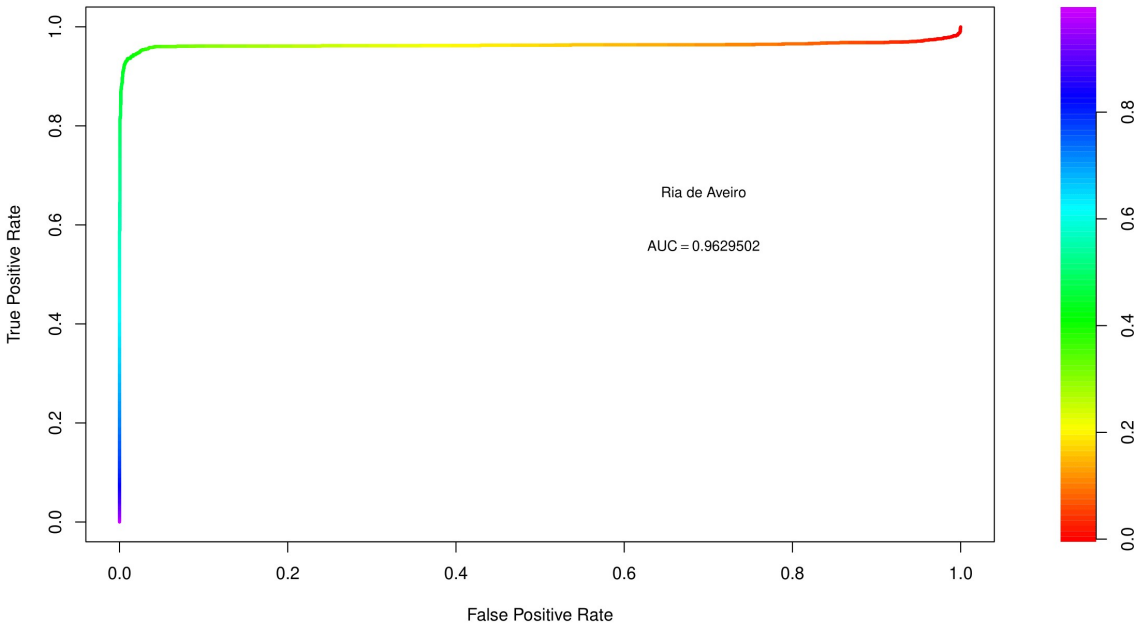



Figure B.5: Ria de Aveiro ROC *curve* and AUC, considering the LASSO method of regularization and the merged testing set of 6,000 observations

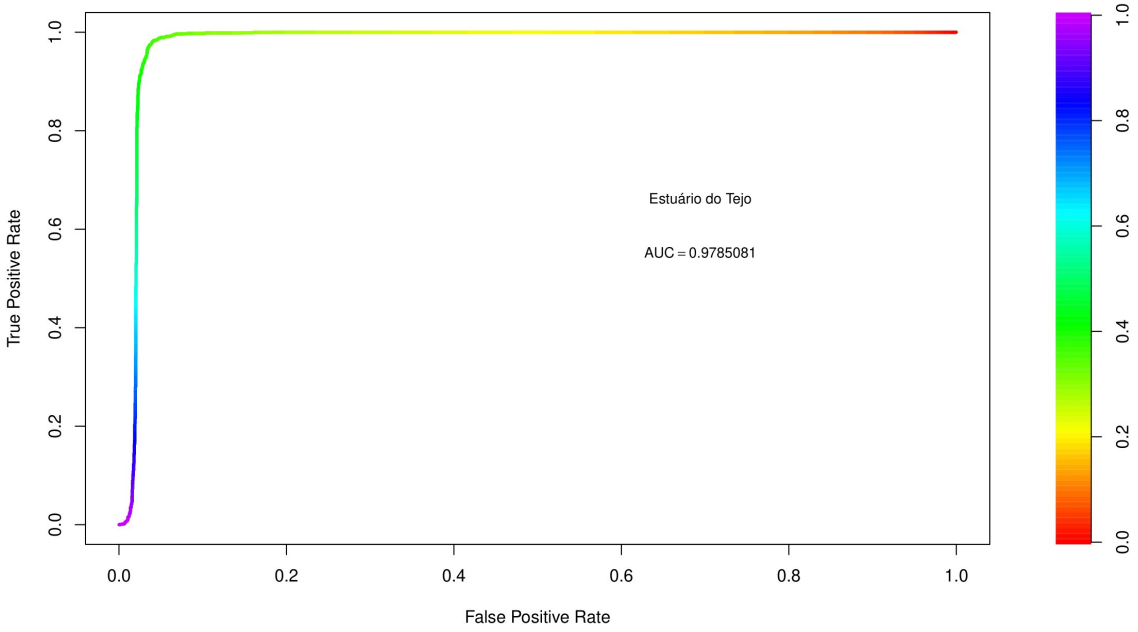



Figure B.6: Estuário do Tejo ROC *curve* and AUC, considering the LASSO method of regularization and the merged testing set of 6,000 observations

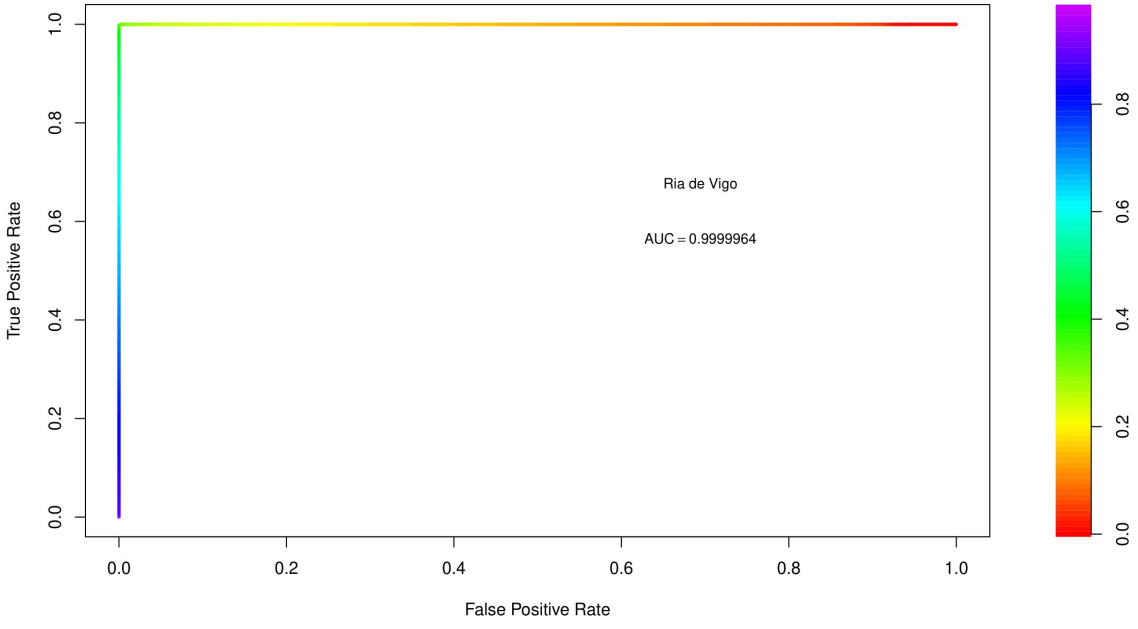



Figure B.7: Ria de Vigo ROC *curve* and AUC, considering the *Elastic Net* method of regularization and the merged testing set of 6,000 observations

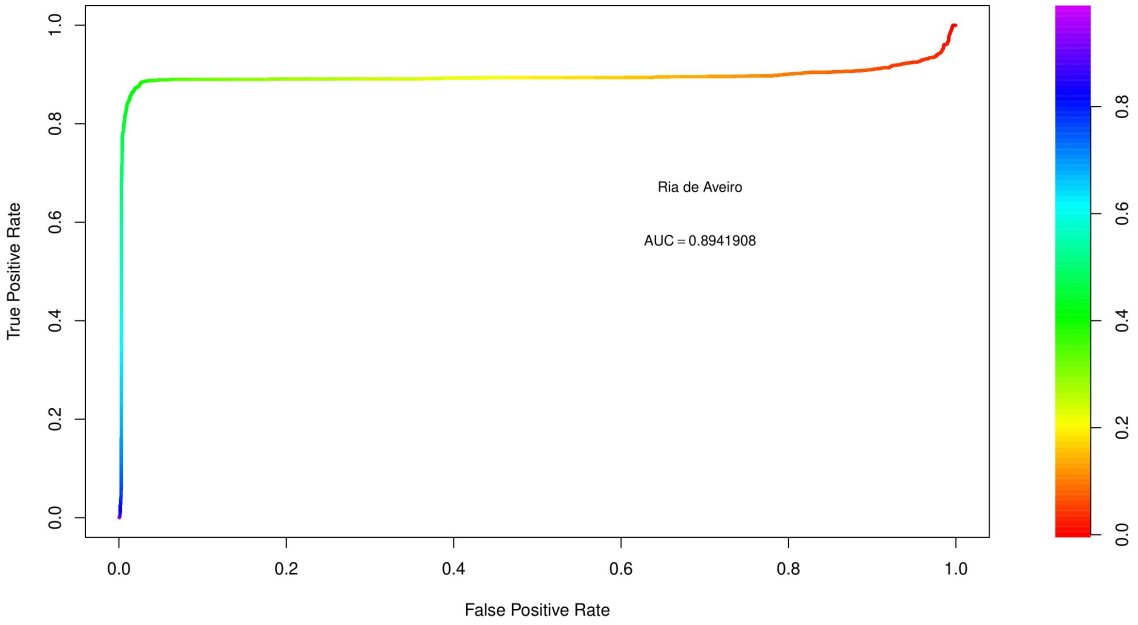



Figure B.8: Ria de Aveiro ROC *curve* and AUC, considering the *Elastic Net* method of regularization and the merged testing set of 6,000 observations

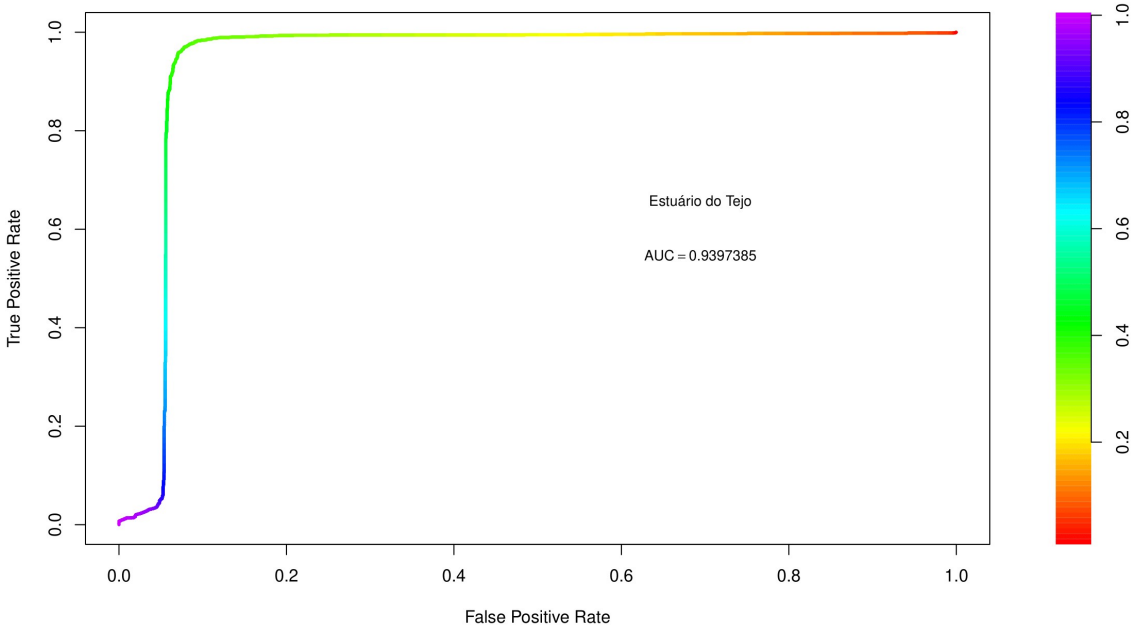



Figure B.9: Estuário do Tejo ROC *curve* and AUC, considering the *Elastic Net* method of regularization and the merged testing set of 6,000 observations

# C

# VARIABLE IMPORTANCE

Table C.1: High importance predictors, per class of the response and per regularization method, considering the first iteration model of *Monte Carlo Cross-Validation*. Predictors are displayed in decreasing order of importance

| **High Importance** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *Ridge* | | | LASSO | | | *Elastic Net* | | |
| G | Rav | T | G | Rav | T | G | Rav | T |
| Ni | FA22:6n_3 | FA22:5n_6 | FA20:5n_3 | FA22:3n_6 | FA18:3n_3 | FA20:5n_3 | FA22:6n_3 | FA18:3n_3 |
| FA20:5n_3 | FA20:5n_3 | FA18:3n_3 | - | - | Sr | FA22:5n_3 | FA20:4n_3 | FA22:5n_6 |
| FA22:5n_3 | FA20:1n_7 | FA18:4n_3 | - | - | - | - | FA20:3n_6 | FA18:4n_3 |
| Y | FA20:4n_3 | Sr | - | - | - | - | FA18:3n_3 | Fe |
| FA18:1n_7 | FA20:3n_6 | Ni | - | - | - | - | FA20:1n_7 | Al |
| Sr | FA22:3n_6 | - | - | - | - | - | FA22:3n_6 | Sr |
| FA14:0 | - | - | - | - | - | - | - | - |
| FA16:1n_7 | - | - | - | - | - | - | - | - |
| FA17:0 | - | - | - | - | - | - | - | - |
| Na | - | - | - | - | - | - | - | - |
| FA22:6n_3 | - | - | - | - | - | - | - | - |
| FA22:5n_6 | - | - | - | - | - | - | - | - |
| Al | - | - | - | - | - | - | - | - |

Table C.2: Medium importance predictors, per class of the response and per regularization method, considering the first iteration model of *Monte Carlo Cross-Validation*. Predictors are displayed in decreasing order of importance

| **Medium Importance** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *Ridge* | | | LASSO | | | *Elastic Net* | | |
| G | Rav | T | G | Rav | T | G | Rav | T |
| FA22:3n_6 | FA22:2n_9 | FA20:3n_6 | - | FA22:6n_3 | FA22:5n_6 | Sr | FA18:0 | FA20:3n_6 |
| U | FA17:0 | Fe | - | FA20:1n_7 | - | Y | FA22:5n_3 | - |
| FA22:4n_6 | Ba | Mg | - | - | - | FA14:0 | - | - |
| FA20:1n_7 | Mn | FA20:2n_6 | - | - | - | FA18:1n_7 | - | - |
| FA20:4n_3 | FA22:5n_3 | Mn | - | - | - | FA16:1n_7 | - | - |
| FA20:1n_9_11 | FA18:1n_7 | FA18:2n_6 | - | - | - | Na | - | - |
| Mg | FA16:1n_7 | Al | - | - | - | FA17:0 | - | - |
| FA22:2n_9 | - | FA18:1n_9 | - | - | - | Al | - | - |
| FA18:2n_6 | - | FA20:1n_9_11 | - | - | - | - | - | - |
| FA18:4n_3 | - | FA22:3n_6 | - | - | - | - | - | - |
| FA18:1n_9 | - | FA20:4n_3 | - | - | - | - | - | - |
| FA20:4n_6 | - | FA18:0 | - | - | - | - | - | - |
| - | - | U | - | - | - | - | - | - |
| - | - | Y | - | - | - | - | - | - |
| - | - | FA14:0 | - | - | - | - | - | - |

Table C.3: Low importance predictors, per class of the response and per regularization method, considering the first iteration model of *Monte Carlo Cross-Validation*. Predictors are displayed in decreasing order of importance

| **Low Importance** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *Ridge* | | | LASSO | | | *Elastic Net* | | |
| G | Rav | T | G | Rav | T | G | Rav | T |
| P | Al | FA17:0 | FA16:1n_7 | - | Fe | FA22:3n_6 | Ba | FA20:2n_6 |
| FA20:3n_6 | Na | FA20:4n_6 | FA14:0 | - | Al | FA22:4n_6 | FA16:1n_9 | Mn |
| FA16:0 | FA15:0 | Zn | - | - | FA20:2n_6 | U | Fe | FA20:1n_7 |
| La | FA16:1n_9 | Cu | - | - | - | Mg | FA22:4n_6 | FA22:3n_6 |
| Gd | FA18:1n_9 | FA20:1n_7 | - | - | - | FA18:2n_6 | FA16:1n_7 | FA20:4n_6 |
| Ce | Sr | Co | - | - | - | FA22:2n_9 | Mn | Mg |
| Co | FA22:2n_6 | Nd | - | - | - | FA18:4n_3 | FA18:1n_7 | FA22:2n_6 |
| FA20:2n_6 | Ce | FA22:2n_6 | - | - | - | FA20:1n_9_11 | FA20:5n_3 | FA18:1n_9 |
| Nd | Gd | FA15:0 | - | - | - | FA20:4n_6 | FA20:2n_6 | FA20:1n_9_11 |
| Mn | FA20:1n_9_11 | P | - | - | - | FA18:1n_9 | - | FA18:2n_6 |
| Cu | La | FA22:6n_3 | - | - | - | FA22:5n_6 | - | Y |
| Ba | Nd | FA22:2n_9 | - | - | - | FA22:6n_3 | - | U |
| Zn | U | Gd | - | - | - | - | - | FA18:0 |
| FA22:2n_6 | Zn | Ce | - | - | - | - | - | FA14:0 |
| FA16:1n_9 | Cu | La | - | - | - | - | - | Ni |
| FA18:0 | Mg | FA16:1n_9 | - | - | - | - | - | - |
| FA18:3n_3 | FA14:0 | FA16:1n_7 | - | - | - | - | - | - |
| Fe | Ni | FA20:5n_3 | - | - | - | - | - | - |
| FA15:0 | FA16:0 | FA18:1n_7 | - | - | - | - | - | - |
| - | FA22:5n_6 | FA22:5n_3 | - | - | - | - | - | - |
| - | Y | FA22:4n_6 | - | - | - | - | - | - |
| - | Co | Ba | - | - | - | - | - | - |
| - | FA18:4n_3 | - | - | - | - | - | - | - |
| - | Fe | - | - | - | - | - | - | - |
| - | P | - | - | - | - | - | - | - |
| - | FA22:4n_6 | - | - | - | - | - | - | - |

# D

# Coefficients of the Selected Featuress

**Notes:**

- If a coefficient does not have an associated *Standard Error of the Mean* value, it signifies that, out of the 1,000 fit models, only one selected that coefficient's respected variable as significant, which inhibits computing the SEM value.
- Tables D.1 through D.9 are arranged by ascending order of probability of variable selection throughout the 1,000 models.

Table D.1: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Ria de Vigo as the location of origin, throughout the 1,000 *Ridge* models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.1116 | 0.0028 | 0.0006 | 0.5215 |
| FA14:0 | 1.0000 | 0.0370 | 0.0002 | 0.0076 | 0.0538 |
| FA15:0 | 1.0000 | 0.0181 | 0.0002 | 0.0040 | 0.0428 |
| FA16:0 | 1.0000 | 0.0159 | 0.0005 | 0.0022 | 0.0904 |
| FA16:1n_9 | 1.0000 | 0.0352 | 0.0017 | 0.0019 | 0.2032 |
| FA16:1n_7 | 1.0000 | 0.0359 | 0.0002 | 0.0074 | 0.0592 |
| FA17:0 | 1.0000 | 0.0358 | 0.0002 | 0.0074 | 0.0518 |
| FA18:0 | 1.0000 | 0.0083 | 0.0001 | 0.0001 | 0.0203 |
| FA18:1n_9 | 1.0000 | 0.0271 | 0.0002 | 0.0050 | 0.0547 |
| FA18:1n_7 | 1.0000 | 0.0374 | 0.0002 | 0.0079 | 0.0612 |
| FA18:2n_6 | 1.0000 | 0.0224 | 0.0002 | 0.0057 | 0.0386 |
| FA18:3n_3 | 1.0000 | 0.0123 | 0.0001 | 0.0017 | 0.0211 |
| FA18:4n_3 | 1.0000 | 0.0265 | 0.0002 | 0.0061 | 0.0397 |
| FA20:1n_9_11 | 1.0000 | 0.0211 | 0.0002 | 0.0039 | 0.0468 |
| FA20:1n_7 | 1.0000 | 0.0288 | 0.0002 | 0.0066 | 0.0427 |
| FA20:2n_6 | 1.0000 | 0.0028 | 0.0001 | 0.0000 | 0.0251 |
| FA20:3n_6 | 1.0000 | 0.0115 | 0.0001 | 0.0014 | 0.0268 |
| FA20:4n_6 | 1.0000 | 0.0263 | 0.0002 | 0.0054 | 0.0461 |
| FA20:4n_3 | 1.0000 | 0.0246 | 0.0002 | 0.0055 | 0.0375 |
| FA20:5n_3 | 1.0000 | 0.0404 | 0.0002 | 0.0088 | 0.0569 |
| FA22:2n_9 | 1.0000 | 0.0288 | 0.0003 | 0.0066 | 0.0599 |
| FA22:2n_6 | 1.0000 | 0.0053 | 0.0001 | 0.0000 | 0.0295 |
| FA22:3n_6 | 1.0000 | 0.0313 | 0.0002 | 0.0067 | 0.0427 |
| FA22:4n_6 | 1.0000 | 0.0290 | 0.0002 | 0.0056 | 0.0404 |
| FA22:5n_6 | 1.0000 | 0.0296 | 0.0003 | 0.0054 | 0.0684 |
| FA22:5n_3 | 1.0000 | 0.0391 | 0.0002 | 0.0081 | 0.0549 |
| FA22:6n_3 | 1.0000 | 0.0367 | 0.0002 | 0.0081 | 0.0576 |
| Na | 1.0000 | 0.0324 | 0.0002 | 0.0070 | 0.0645 |
| Mg | 1.0000 | 0.0283 | 0.0003 | 0.0055 | 0.0658 |
| Al | 1.0000 | 0.0271 | 0.0002 | 0.0054 | 0.0556 |
| P | 1.0000 | 0.0148 | 0.0001 | 0.0026 | 0.0280 |
| Mn | 1.0000 | 0.0053 | 0.0002 | 0.0000 | 0.0397 |
| Fe | 1.0000 | 0.0165 | 0.0001 | 0.0036 | 0.0451 |
| Co | 1.0000 | 0.0168 | 0.0009 | 0.0000 | 0.1411 |
| Ni | 1.0000 | 0.0130 | 0.0007 | 0.0000 | 0.0964 |
| Cu | 1.0000 | 0.1467 | 0.0095 | 0.0014 | 1.3083 |
| Zn | 1.0000 | 0.0364 | 0.0022 | 0.0016 | 0.3738 |
| Sr | 1.0000 | 0.0358 | 0.0002 | 0.0074 | 0.0489 |
| Y | 1.0000 | 0.0347 | 0.0002 | 0.0076 | 0.0579 |
| Ba | 1.0000 | 0.0082 | 0.0002 | 0.0000 | 0.0329 |
| La | 1.0000 | 0.0080 | 0.0002 | 0.0001 | 0.0289 |
| Ce | 1.0000 | 0.0043 | 0.0001 | 0.0000 | 0.0245 |
| Nd | 1.0000 | 0.0031 | 0.0001 | 0.0000 | 0.0157 |
| Gd | 1.0000 | 0.0054 | 0.0001 | 0.0000 | 0.0274 |
| U | 1.0000 | 0.0292 | 0.0002 | 0.0058 | 0.0485 |

Table D.2: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Ria de Aveiro as the location of origin, throughout the 1,000 *Ridge* models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.0987 | 0.0025 | 0.0000 | 0.4873 |
| FA14:0 | 1.0000 | 0.0081 | 0.0001 | 0.0004 | 0.0175 |
| FA15:0 | 1.0000 | 0.0105 | 0.0002 | 0.0000 | 0.0373 |
| FA16:0 | 1.0000 | 0.0144 | 0.0004 | 0.0001 | 0.0836 |
| FA16:1n_9 | 1.0000 | 0.0272 | 0.0010 | 0.0024 | 0.1771 |
| FA16:1n_7 | 1.0000 | 0.0271 | 0.0002 | 0.0042 | 0.0559 |
| FA17:0 | 1.0000 | 0.0218 | 0.0002 | 0.0032 | 0.0381 |
| FA18:0 | 1.0000 | 0.0357 | 0.0003 | 0.0051 | 0.0543 |
| FA18:1n_9 | 1.0000 | 0.0077 | 0.0002 | 0.0000 | 0.0495 |
| FA18:1n_7 | 1.0000 | 0.0304 | 0.0002 | 0.0040 | 0.0541 |
| FA18:2n_6 | 1.0000 | 0.0077 | 0.0001 | 0.0000 | 0.0206 |
| FA18:3n_3 | 1.0000 | 0.0336 | 0.0003 | 0.0041 | 0.0651 |
| FA18:4n_3 | 1.0000 | 0.0136 | 0.0001 | 0.0005 | 0.0284 |
| FA20:1n_9_11 | 1.0000 | 0.0051 | 0.0001 | 0.0000 | 0.0288 |
| FA20:1n_7 | 1.0000 | 0.0341 | 0.0002 | 0.0054 | 0.0566 |
| FA20:2n_6 | 1.0000 | 0.0225 | 0.0002 | 0.0027 | 0.0446 |
| FA20:3n_6 | 1.0000 | 0.0383 | 0.0003 | 0.0059 | 0.0576 |
| FA20:4n_6 | 1.0000 | 0.0074 | 0.0001 | 0.0000 | 0.0223 |
| FA20:4n_3 | 1.0000 | 0.0369 | 0.0002 | 0.0064 | 0.0580 |
| FA20:5n_3 | 1.0000 | 0.0314 | 0.0002 | 0.0048 | 0.0537 |
| FA22:2n_9 | 1.0000 | 0.0273 | 0.0002 | 0.0056 | 0.0494 |
| FA22:2n_6 | 1.0000 | 0.0087 | 0.0004 | 0.0000 | 0.0652 |
| FA22:3n_6 | 1.0000 | 0.0435 | 0.0003 | 0.0068 | 0.0652 |
| FA22:4n_6 | 1.0000 | 0.0213 | 0.0002 | 0.0029 | 0.0384 |
| FA22:5n_6 | 1.0000 | 0.0094 | 0.0001 | 0.0002 | 0.0207 |
| FA22:5n_3 | 1.0000 | 0.0278 | 0.0002 | 0.0035 | 0.0455 |
| FA22:6n_3 | 1.0000 | 0.0384 | 0.0003 | 0.0062 | 0.0645 |
| Na | 1.0000 | 0.0135 | 0.0001 | 0.0018 | 0.0271 |
| Mg | 1.0000 | 0.0035 | 0.0001 | 0.0000 | 0.0234 |
| Al | 1.0000 | 0.0221 | 0.0002 | 0.0040 | 0.0501 |
| P | 1.0000 | 0.0174 | 0.0002 | 0.0026 | 0.0366 |
| Mn | 1.0000 | 0.0255 | 0.0002 | 0.0032 | 0.0965 |
| Fe | 1.0000 | 0.0113 | 0.0001 | 0.0009 | 0.0259 |
| Co | 1.0000 | 0.0283 | 0.0010 | 0.0019 | 0.1704 |
| Ni | 1.0000 | 0.0113 | 0.0001 | 0.0000 | 0.0341 |
| Cu | 1.0000 | 0.1032 | 0.0067 | 0.0007 | 0.9113 |
| Zn | 1.0000 | 0.0162 | 0.0011 | 0.0005 | 0.3527 |
| Sr | 1.0000 | 0.0029 | 0.0001 | 0.0000 | 0.0142 |
| Y | 1.0000 | 0.0095 | 0.0001 | 0.0004 | 0.0211 |
| Ba | 1.0000 | 0.0176 | 0.0002 | 0.0014 | 0.0367 |
| La | 1.0000 | 0.0049 | 0.0002 | 0.0000 | 0.0399 |
| Ce | 1.0000 | 0.0060 | 0.0001 | 0.0000 | 0.0359 |
| Nd | 1.0000 | 0.0056 | 0.0001 | 0.0000 | 0.0346 |
| Gd | 1.0000 | 0.0056 | 0.0001 | 0.0000 | 0.0208 |
| U | 1.0000 | 0.0055 | 0.0001 | 0.0000 | 0.0142 |

Table D.3: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Estuário do Tejo as the location of origin, throughout the 1,000 *Ridge* models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.1142 | 0.0025 | 0.0009 | 0.5847 |
| FA14:0 | 1.0000 | 0.0289 | 0.0002 | 0.0044 | 0.0550 |
| FA15:0 | 1.0000 | 0.0082 | 0.0002 | 0.0000 | 0.0288 |
| FA16:0 | 1.0000 | 0.0171 | 0.0010 | 0.0000 | 0.1206 |
| FA16:1n_9 | 1.0000 | 0.0507 | 0.0028 | 0.0003 | 0.3140 |
| FA16:1n_7 | 1.0000 | 0.0089 | 0.0001 | 0.0000 | 0.0203 |
| FA17:0 | 1.0000 | 0.0139 | 0.0001 | 0.0021 | 0.0297 |
| FA18:0 | 1.0000 | 0.0274 | 0.0002 | 0.0028 | 0.0506 |
| FA18:1n_9 | 1.0000 | 0.0194 | 0.0002 | 0.0012 | 0.0347 |
| FA18:1n_7 | 1.0000 | 0.0070 | 0.0001 | 0.0000 | 0.0179 |
| FA18:2n_6 | 1.0000 | 0.0299 | 0.0002 | 0.0051 | 0.0451 |
| FA18:3n_3 | 1.0000 | 0.0459 | 0.0003 | 0.0078 | 0.0851 |
| FA18:4n_3 | 1.0000 | 0.0400 | 0.0003 | 0.0068 | 0.0586 |
| FA20:1n_9_11 | 1.0000 | 0.0226 | 0.0002 | 0.0036 | 0.0472 |
| FA20:1n_7 | 1.0000 | 0.0057 | 0.0001 | 0.0000 | 0.0282 |
| FA20:2n_6 | 1.0000 | 0.0213 | 0.0002 | 0.0022 | 0.0385 |
| FA20:3n_6 | 1.0000 | 0.0268 | 0.0002 | 0.0026 | 0.0418 |
| FA20:4n_6 | 1.0000 | 0.0193 | 0.0002 | 0.0021 | 0.0440 |
| FA20:4n_3 | 1.0000 | 0.0122 | 0.0001 | 0.0004 | 0.0268 |
| FA20:5n_3 | 1.0000 | 0.0089 | 0.0001 | 0.0011 | 0.0202 |
| FA22:2n_9 | 1.0000 | 0.0030 | 0.0001 | 0.0000 | 0.0214 |
| FA22:2n_6 | 1.0000 | 0.0125 | 0.0004 | 0.0001 | 0.0606 |
| FA22:3n_6 | 1.0000 | 0.0123 | 0.0002 | 0.0001 | 0.0299 |
| FA22:4n_6 | 1.0000 | 0.0078 | 0.0001 | 0.0000 | 0.0249 |
| FA22:5n_6 | 1.0000 | 0.0390 | 0.0003 | 0.0063 | 0.0848 |
| FA22:5n_3 | 1.0000 | 0.0113 | 0.0001 | 0.0019 | 0.0231 |
| FA22:6n_3 | 1.0000 | 0.0029 | 0.0001 | 0.0000 | 0.0138 |
| Na | 1.0000 | 0.0189 | 0.0001 | 0.0029 | 0.0489 |
| Mg | 1.0000 | 0.0303 | 0.0003 | 0.0050 | 0.0666 |
| Al | 1.0000 | 0.0058 | 0.0002 | 0.0000 | 0.0302 |
| P | 1.0000 | 0.0040 | 0.0001 | 0.0000 | 0.0283 |
| Mn | 1.0000 | 0.0305 | 0.0003 | 0.0048 | 0.1249 |
| Fe | 1.0000 | 0.0278 | 0.0002 | 0.0045 | 0.0493 |
| Co | 1.0000 | 0.0115 | 0.0002 | 0.0001 | 0.0646 |
| Ni | 1.0000 | 0.0224 | 0.0005 | 0.0020 | 0.0890 |
| Cu | 1.0000 | 0.0518 | 0.0037 | 0.0021 | 1.0472 |
| Zn | 1.0000 | 0.0391 | 0.0026 | 0.0014 | 0.6620 |
| Sr | 1.0000 | 0.0351 | 0.0003 | 0.0056 | 0.0577 |
| Y | 1.0000 | 0.0252 | 0.0002 | 0.0034 | 0.0571 |
| Ba | 1.0000 | 0.0095 | 0.0001 | 0.0001 | 0.0214 |
| La | 1.0000 | 0.0072 | 0.0001 | 0.0000 | 0.0176 |
| Ce | 1.0000 | 0.0077 | 0.0001 | 0.0000 | 0.0221 |
| Nd | 1.0000 | 0.0039 | 0.0001 | 0.0000 | 0.0349 |
| Gd | 1.0000 | 0.0077 | 0.0001 | 0.0000 | 0.0205 |
| U | 1.0000 | 0.0239 | 0.0002 | 0.0034 | 0.0468 |

Table D.4: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Ria de Vigo as the location of origin, throughout the 1,000 LASSO models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.0895 | 0.0024 | 0.0000 | 0.4259 |
| FA20:5n_3 | 0.9990 | 0.8213 | 0.0082 | 0.0204 | 2.3161 |
| FA14:0 | 0.4180 | 0.1209 | 0.0053 | 0.0002 | 0.7655 |
| FA22:5n_3 | 0.3080 | 0.1434 | 0.0072 | 0.0000 | 0.7781 |
| FA16:1n_7 | 0.2250 | 0.2400 | 0.0158 | 0.0003 | 1.1711 |
| FA18:1n_7 | 0.1240 | 0.1465 | 0.0167 | 0.0019 | 1.1441 |
| FA22:6n_3 | 0.0410 | 0.1187 | 0.0181 | 0.0005 | 0.4207 |
| Sr | 0.0290 | 0.0951 | 0.0158 | 0.0117 | 0.3130 |
| Mg | 0.0140 | 0.0905 | 0.0240 | 0.0035 | 0.3480 |
| Na | 0.0090 | 0.1311 | 0.0292 | 0.0014 | 0.2741 |
| Y | 0.0070 | 0.1220 | 0.0484 | 0.0018 | 0.3673 |
| U | 0.0060 | 0.0497 | 0.0245 | 0.0002 | 0.1632 |
| FA17:0 | 0.0030 | 0.0286 | 0.0124 | 0.0037 | 0.0413 |
| FA22:5n_6 | 0.0010 | 0.0348 | - | 0.0348 | 0.0348 |

Table D.5: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Ria de Aveiro as the location of origin, throughout the 1,000 LASSO models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.0814 | 0.0025 | 0.0000 | 0.5104 |
| FA22:3n_6 | 0.9980 | 0.5841 | 0.0115 | 0.0064 | 2.6134 |
| FA22:6n_3 | 0.1760 | 0.1689 | 0.0145 | 0.0034 | 1.0424 |
| FA18:0 | 0.0960 | 0.0989 | 0.0094 | 0.0075 | 0.6472 |
| FA20:3n_6 | 0.0910 | 0.0878 | 0.0098 | 0.0006 | 0.5271 |
| FA20:1n_7 | 0.0250 | 0.1694 | 0.0462 | 0.0001 | 0.9399 |
| Mn | 0.0210 | 0.1308 | 0.0282 | 0.0118 | 0.5711 |
| FA20:4n_3 | 0.0140 | 0.0972 | 0.0452 | 0.0091 | 0.6689 |
| FA22:2n_6 | 0.0030 | 0.1425 | 0.0528 | 0.0614 | 0.2417 |
| Al | 0.0020 | 0.0989 | 0.0938 | 0.0052 | 0.1927 |
| FA18:1n_7 | 0.0020 | 0.1649 | 0.0565 | 0.1084 | 0.2213 |
| Co | 0.0010 | 0.0805 | - | 0.0805 | 0.0805 |
| Ni | 0.0010 | 0.0003 | - | 0.0003 | 0.0003 |

Table D.6: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Estuário do Tejo as the location of origin, throughout the 1,000 LASSO models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.1254 | 0.0037 | 0.0001 | 0.9169 |
| FA18:3n_3 | 1.0000 | 0.7571 | 0.0107 | 0.0838 | 2.6361 |
| FA22:5n_6 | 0.7600 | 0.2290 | 0.0081 | 0.0001 | 2.8846 |
| Mn | 0.1580 | 0.2264 | 0.0194 | 0.0004 | 1.5649 |
| FA18:4n_3 | 0.1270 | 0.2005 | 0.0173 | 0.0002 | 1.2017 |
| FA16:1n_9 | 0.0930 | 1.5298 | 0.1351 | 0.0084 | 6.6432 |
| Sr | 0.0840 | 0.2080 | 0.0251 | 0.0002 | 1.2440 |
| Fe | 0.0530 | 0.1581 | 0.0187 | 0.0030 | 0.5875 |
| FA20:4n_6 | 0.0280 | 0.2755 | 0.0417 | 0.0026 | 0.9087 |
| FA22:2n_6 | 0.0110 | 0.2086 | 0.0496 | 0.0010 | 0.5636 |
| Mg | 0.0080 | 0.1928 | 0.1041 | 0.0010 | 0.8956 |
| U | 0.0040 | 0.0624 | 0.0524 | 0.0042 | 0.2193 |
| Zn | 0.0030 | 1.7357 | 0.8343 | 0.2373 | 3.1208 |
| FA20:1n_9_11 | 0.0020 | 0.1060 | 0.0185 | 0.0876 | 0.1245 |
| FA20:2n_6 | 0.0010 | 0.1037 | - | 0.1037 | 0.1037 |
| Y | 0.0010 | 0.6927 | - | 0.6927 | 0.6927 |

Table D.7: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Ria de Vigo as the location of origin, throughout the 1,000 *Elastic Net* models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.1148 | 0.0029 | 0.0002 | 0.5012 |
| FA20:5n_3 | 1.0000 | 0.2113 | 0.0062 | 0.0965 | 1.2822 |
| FA22:5n_3 | 0.9420 | 0.1173 | 0.0013 | 0.0015 | 0.5440 |
| FA18:1n_7 | 0.9120 | 0.0922 | 0.0010 | 0.0013 | 0.3816 |
| FA14:0 | 0.9000 | 0.0933 | 0.0012 | 0.0000 | 0.3857 |
| FA16:1n_7 | 0.8860 | 0.0826 | 0.0012 | 0.0003 | 0.4147 |
| FA22:6n_3 | 0.8810 | 0.0836 | 0.0010 | 0.0009 | 0.4200 |
| Sr | 0.8280 | 0.0797 | 0.0009 | 0.0011 | 0.1818 |
| Y | 0.8110 | 0.0656 | 0.0009 | 0.0000 | 0.2015 |
| FA17:0 | 0.8050 | 0.0740 | 0.0009 | 0.0009 | 0.2273 |
| Na | 0.6770 | 0.0539 | 0.0008 | 0.0001 | 0.1205 |
| FA22:3n_6 | 0.5560 | 0.0518 | 0.0010 | 0.0004 | 0.1003 |
| U | 0.5240 | 0.0409 | 0.0009 | 0.0000 | 0.1439 |
| FA22:5n_6 | 0.4980 | 0.0472 | 0.0012 | 0.0004 | 0.1762 |
| FA22:4n_6 | 0.4950 | 0.0478 | 0.0008 | 0.0001 | 0.1041 |
| Mg | 0.4790 | 0.0474 | 0.0018 | 0.0007 | 0.2352 |
| FA22:2n_9 | 0.4750 | 0.0408 | 0.0012 | 0.0000 | 0.1707 |
| FA18:1n_9 | 0.4680 | 0.0412 | 0.0011 | 0.0002 | 0.1623 |
| FA20:1n_7 | 0.4670 | 0.0337 | 0.0008 | 0.0001 | 0.0959 |
| Al | 0.4560 | 0.0391 | 0.0010 | 0.0002 | 0.1198 |
| FA18:4n_3 | 0.4240 | 0.0366 | 0.0007 | 0.0001 | 0.0765 |
| FA20:4n_6 | 0.4230 | 0.0384 | 0.0009 | 0.0004 | 0.0953 |
| FA20:4n_3 | 0.3570 | 0.0227 | 0.0006 | 0.0000 | 0.0527 |
| FA20:1n_9_11 | 0.3370 | 0.0221 | 0.0012 | 0.0002 | 0.2013 |
| FA18:2n_6 | 0.3320 | 0.0189 | 0.0007 | 0.0003 | 0.0689 |
| FA16:1n_9 | 0.1540 | 0.0857 | 0.0049 | 0.0004 | 0.3235 |
| FA15:0 | 0.1490 | 0.0114 | 0.0007 | 0.0000 | 0.0388 |
| Cu | 0.0910 | 1.0228 | 0.0514 | 0.0337 | 2.4687 |
| Co | 0.0870 | 0.0751 | 0.0038 | 0.0029 | 0.1654 |
| Ni | 0.0700 | 0.0442 | 0.0035 | 0.0008 | 0.1135 |
| P | 0.0250 | 0.0069 | 0.0011 | 0.0006 | 0.0228 |
| Fe | 0.0200 | 0.0040 | 0.0007 | 0.0000 | 0.0119 |
| FA16:0 | 0.0170 | 0.0236 | 0.0053 | 0.0012 | 0.0918 |
| Ba | 0.0040 | 0.0071 | 0.0024 | 0.0036 | 0.0143 |
| La | 0.0010 | 0.0051 | - | 0.0051 | 0.0051 |
| Zn | 0.0010 | 0.1633 | - | 0.1633 | 0.1633 |

Table D.8: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Ria de Aveiro as the location of origin, throughout the 1,000 *Elastic Net* models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.0827 | 0.0021 | 0.0000 | 0.4635 |
| FA22:3n_6 | 1.0000 | 0.2176 | 0.0039 | 0.0330 | 1.1912 |
| FA22:6n_3 | 0.8990 | 0.1079 | 0.0013 | 0.0003 | 0.4451 |
| FA20:3n_6 | 0.8340 | 0.1051 | 0.0013 | 0.0012 | 0.2976 |
| FA20:4n_3 | 0.7870 | 0.0847 | 0.0011 | 0.0010 | 0.2420 |
| FA18:0 | 0.7770 | 0.0936 | 0.0014 | 0.0003 | 0.3615 |
| FA20:1n_7 | 0.6850 | 0.0800 | 0.0015 | 0.0002 | 0.4317 |
| FA18:3n_3 | 0.6340 | 0.0855 | 0.0016 | 0.0002 | 0.3257 |
| FA20:5n_3 | 0.5720 | 0.0531 | 0.0009 | 0.0001 | 0.1197 |
| FA18:1n_7 | 0.5270 | 0.0604 | 0.0013 | 0.0001 | 0.2103 |
| FA22:5n_3 | 0.4670 | 0.0495 | 0.0011 | 0.0005 | 0.1075 |
| Mn | 0.4410 | 0.0534 | 0.0014 | 0.0013 | 0.3009 |
| FA16:1n_7 | 0.4220 | 0.0353 | 0.0008 | 0.0005 | 0.0850 |
| FA22:2n_9 | 0.3980 | 0.0239 | 0.0007 | 0.0005 | 0.1242 |
| FA22:4n_6 | 0.3540 | 0.0393 | 0.0012 | 0.0006 | 0.1251 |
| Al | 0.3400 | 0.0297 | 0.0013 | 0.0001 | 0.1205 |
| FA20:2n_6 | 0.3270 | 0.0287 | 0.0011 | 0.0001 | 0.0945 |
| Ba | 0.2050 | 0.0195 | 0.0010 | 0.0001 | 0.0655 |
| FA17:0 | 0.1730 | 0.0147 | 0.0011 | 0.0001 | 0.0678 |
| P | 0.1380 | 0.0157 | 0.0010 | 0.0001 | 0.0510 |
| Co | 0.1330 | 0.1632 | 0.0083 | 0.0003 | 0.4558 |
| FA16:0 | 0.0820 | 0.0411 | 0.0037 | 0.0002 | 0.1537 |
| Ni | 0.0740 | 0.0062 | 0.0006 | 0.0001 | 0.0214 |
| FA16:1n_9 | 0.0720 | 0.0719 | 0.0077 | 0.0002 | 0.2851 |
| FA22:2n_6 | 0.0600 | 0.0790 | 0.0057 | 0.0013 | 0.2507 |
| FA18:4n_3 | 0.0520 | 0.0097 | 0.0011 | 0.0000 | 0.0299 |
| Fe | 0.0510 | 0.0083 | 0.0008 | 0.0003 | 0.0250 |
| Cu | 0.0460 | 0.4703 | 0.0535 | 0.0077 | 1.2907 |
| FA15:0 | 0.0310 | 0.0168 | 0.0026 | 0.0003 | 0.0580 |
| FA18:1n_9 | 0.0100 | 0.0308 | 0.0126 | 0.0001 | 0.1249 |
| Ce | 0.0050 | 0.0035 | 0.0013 | 0.0005 | 0.0069 |
| Zn | 0.0050 | 0.3157 | 0.0719 | 0.0684 | 0.4881 |
| FA22:5n_6 | 0.0020 | 0.0068 | 0.0062 | 0.0006 | 0.0130 |
| Nd | 0.0020 | 0.0230 | 0.0216 | 0.0015 | 0.0446 |
| FA18:2n_6 | 0.0010 | 0.0012 | - | 0.0012 | 0.0012 |
| Gd | 0.0010 | 0.0010 | - | 0.0010 | 0.0010 |
| La | 0.0010 | 0.0239 | - | 0.0239 | 0.0239 |
| Na | 0.0010 | 0.0068 | - | 0.0068 | 0.0068 |

Table D.9: Probability of variable selection (prob), average ($\overline{|\beta|}$), standard error of the mean (sem($|\beta|$)), minimum (min($|\beta|$)) and maximum (max($|\beta|$)) of the absolute coefficient values of the selected features to predict Estuário do Tejo as the location of origin, throughout the 1,000 *Elastic Net* models

| **Predictor** | **prob** | $\overline{\lvert\beta\rvert}$ | **sem**($\lvert\beta\rvert$) | **min**($\lvert\beta\rvert$) | **max**($\lvert\beta\rvert$) |
|---|---|---|---|---|---|
| intercept | 1.0000 | 0.1316 | 0.0032 | 0.0000 | 0.6602 |
| FA18:3n_3 | 1.0000 | 0.2708 | 0.0056 | 0.1256 | 1.5944 |
| FA22:5n_6 | 0.9220 | 0.1264 | 0.0024 | 0.0014 | 0.7923 |
| FA18:4n_3 | 0.8980 | 0.1282 | 0.0015 | 0.0001 | 0.4956 |
| Sr | 0.7410 | 0.0957 | 0.0018 | 0.0002 | 0.8281 |
| Mn | 0.6260 | 0.0914 | 0.0026 | 0.0002 | 0.4976 |
| FA18:2n_6 | 0.5370 | 0.0547 | 0.0010 | 0.0000 | 0.1434 |
| Fe | 0.5280 | 0.0617 | 0.0014 | 0.0001 | 0.2120 |
| FA14:0 | 0.5270 | 0.0474 | 0.0010 | 0.0002 | 0.1105 |
| Mg | 0.5080 | 0.0570 | 0.0015 | 0.0000 | 0.1878 |
| FA18:0 | 0.4860 | 0.0568 | 0.0012 | 0.0002 | 0.1385 |
| FA20:3n_6 | 0.4600 | 0.0533 | 0.0011 | 0.0000 | 0.1309 |
| Y | 0.3990 | 0.0392 | 0.0008 | 0.0005 | 0.0973 |
| U | 0.3900 | 0.0317 | 0.0010 | 0.0001 | 0.0933 |
| FA20:4n_6 | 0.3490 | 0.0509 | 0.0015 | 0.0003 | 0.1379 |
| FA20:2n_6 | 0.3470 | 0.0297 | 0.0010 | 0.0000 | 0.1077 |
| FA20:1n_9_11 | 0.3320 | 0.0305 | 0.0013 | 0.0001 | 0.1668 |
| Ni | 0.2700 | 0.0259 | 0.0020 | 0.0003 | 0.1864 |
| FA18:1n_9 | 0.2650 | 0.0215 | 0.0010 | 0.0002 | 0.0805 |
| FA16:1n_9 | 0.2250 | 0.6683 | 0.0115 | 0.0092 | 1.3730 |
| FA16:0 | 0.2110 | 0.1801 | 0.0039 | 0.0074 | 0.3253 |
| Na | 0.1300 | 0.0092 | 0.0006 | 0.0001 | 0.0302 |
| FA22:2n_6 | 0.0960 | 0.0725 | 0.0063 | 0.0008 | 0.3992 |
| FA22:3n_6 | 0.0790 | 0.0122 | 0.0013 | 0.0002 | 0.0613 |
| FA17:0 | 0.0380 | 0.0099 | 0.0012 | 0.0004 | 0.0297 |
| FA15:0 | 0.0210 | 0.0191 | 0.0033 | 0.0017 | 0.0600 |
| Zn | 0.0200 | 0.9694 | 0.1586 | 0.0055 | 2.0088 |
| Ba | 0.0080 | 0.0088 | 0.0021 | 0.0008 | 0.0193 |
| Ce | 0.0050 | 0.0106 | 0.0025 | 0.0017 | 0.0146 |
| FA20:4n_3 | 0.0050 | 0.0027 | 0.0006 | 0.0008 | 0.0041 |
| Al | 0.0040 | 0.0219 | 0.0039 | 0.0116 | 0.0300 |
| Cu | 0.0040 | 0.5696 | 0.1322 | 0.2016 | 0.7700 |
| Gd | 0.0040 | 0.0077 | 0.0024 | 0.0032 | 0.0133 |
| FA20:1n_7 | 0.0020 | 0.0135 | 0.0041 | 0.0094 | 0.0175 |
| La | 0.0010 | 0.0014 | - | 0.0014 | 0.0014 |