DeepData: Machine Learning in the Marine Ecosystems

Leonor Pimenta de Oliveira e Silva

Thesis to obtain the Master of Science Degree in

Engenharia Informática e de Computadores

Supervisor(s): Prof. Maria Inês Camarate de Campos Lynce de Faria
Prof. Vasco Miguel Gomes Nunes Manquinho

Examination Committee

Chairperson: Prof. Francisco António Chaves Saraiva de Melo
Supervisor: Prof. Vasco Miguel Gomes Nunes Manquinho
Member of the Committee: Prof. Rui Miguel Carrasqueiro Henriques

November 2019

Acknowledgments

I would like to thank my family for all the support and encouragement throughout all these years.

I would also like to acknowledge my thesis supervisors Prof. Maria Inês Camarate de Campos Lynce de Faria and Prof. Vasco Miguel Gomes Nunes Manquinho for their insight, support and sharing of knowledge that has made this thesis possible.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with references UID/CEC/50021/2019 and CMU/AIR/0022/2017.

Resumo

Nesta tese é apresentada uma ferramenta online que faz uso de aprendizagem automática com o propósito de facilitar a construção de modelos de distribuição das espécies pelos biólogos. Esta ferramenta tem em consideração o modo como os biólogos lidam com os modelos de distribuição das espécies. Actualmente, os biólogos usam maioritariamente algoritmos probabilísticos, tais como o `MaxEnt`, `generalized linear models` e `generalized addictive models`. Esta ferramenta possibilita também o uso de algoritmos de aprendizagem automática, tais como `classification and regression trees`, `random forest` e `support vector machine`. Outros passos envolvidos na construção de modelos de distribuição das espécies, como a preparação da informação e a avaliação do modelo, também são discutidos. Uma explicação do uso desta ferramenta é feita, assim como da sua implementação e avaliação.

Palavras-chave: aprendizagem automática, modelos de distribuição das espécies, ecossistema marinho

Abstract

This thesis presents a web-based machine learning tool to facilitate biologists work of building species distribution models. This web-based tool takes into account the way biologists deal with species distribution models nowadays. Biologists mostly use probabilistic algorithms, such as `maximum entropy`, `generalized linear models` and `generalized addictive models`. We propose the use of machine learning algorithms, such as `classification and regression trees`, `random forest` and `support vector machine`. Other steps involved in the species distribution models, such as data preparation and model evaluation, are also discussed. A concrete explanation of the use of the web-based tool is made, as well as the details of implementation and evaluation.

Keywords: machine learning, species distribution models, marine ecosystem

Contents

Acknowledgments . . . iii
Resumo . . . v
Abstract . . . vii
List of Tables . . . xi
List of Figures . . . xiii

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Topic Overview . . . 2
1.3 Objectives . . . 2
1.4 Contributions . . . 3
1.5 Thesis Outline . . . 4

2 Background **5**
2.1 Species distribution models . . . 5
2.2 Data pre-processing . . . 7
2.2.1 Occurrence data . . . 7
2.2.2 Environmental data . . . 8
2.3 Model selection . . . 11
2.3.1 Presence-only models . . . 11
2.3.2 Presence/absence models . . . 13
2.4 Model evaluation . . . 27
2.4.1 Qualitative response variables . . . 27
2.4.2 Quantitative response variables . . . 29

3 The DeepData tool **33**
3.1 Tool architecture . . . 33
3.1.1 Relational database . . . 33
3.2 Tool implementation . . . 35
3.2.1 Data insertion . . . 35
3.2.2 Input selection . . . 37
3.2.3 Model implementation . . . 42

4 Results **47**
4.1 First case study 47
4.1.1 Problem description 47
4.1.2 Tool testing 48
4.2 Second case study 51
4.2.1 Problem description 51
4.2.2 Tool testing 51

5 Conclusions **55**
5.1 Achievements 55
5.2 Future Work 56

Bibliography **59**

A Tool's flow diagrams
A.1 Tool processes . . . . .

B Case study results
B.1 Jackknife results . . . . .
B.2 Response curves first case . . . . .
B.3 Response curves second case . . . . .

List of Tables

2.1 Commonly used combinations of link function and distribution families 14
2.2 Examples of kernel functions. 21
2.3 Structure of a confusion matrix. 28
2.4 Performance measures. 28
2.5 Formula of different performance metrics. 30

3.1 R packages used for each implemented model. 39
3.2 Relation between the family distribution and the link function. 40
3.3 Kernel functions, where u and v, represent inputs from the feature space. 40

4.1 Variable permutation importance in February, March and April. 50
4.2 Monthly model AUC and standard deviation. 51
4.3 Models AUC and standard deviation. 53

List of Figures

2.1 Representation of the SDM process. 6
2.2 Representation of pseudo-absences generation using Mopa package [8]. 7
2.3 Representation of spatial inference [12]. 9
2.4 `MaxEnt` representation [20]. 13
2.5 Comparison between using mean and using loess for smoothness [24]. 15
2.6 The first image compares the use of a linear and a natural cubic spline. The second image compares the use of the natural cubic spline and regression. 16
2.7 Representation of the difference between `CART` [28], top image, and `Random Forest` [29], bottom image. 17
2.8 Hyperplane through two linearly separable classes [32]. 18
2.9 Representation of feature space manipulation. In the first image the data is linearly separable, while in the second image the data is nonlinear [33]. 20
2.10 Perceptron ilustration [37]. 21
2.11 `Artificial neural network` illustration [37]. 22
2.12 Representation of the effect of weights, left image, and bias, right image, on sigmoid [37]. 22
2.13 One dimensional gradient descent representation [37]. 24
2.14 Backpropagation example [37]. 25
2.15 10-fold cross validation representation [43]. 27
2.16 Representation of discrimination between classifications [40]. 28
2.17 Representation of a ROC curve [40]. 29

3.1 Relational scheme of database. 34
3.2 `DeepData`'s data insertion interface. 36
3.3 `DeepData`'s input interface. 38
3.4 Example of variable aggregation and disaggregation. 42
3.5 Methods of ensemble models. 44

4.1 Probability of presence of crabeater seals for each month. 49
4.2 Spatial distribution of Trindade Petrel for each model. 52

5.1 `DeepData's` year selection for train and test data. 57

A.1 Flowchart symbols and name. . . . .
A.2 Species data insertion process . . . . .
A.3 Environmental data insertion process . . . . .
A.4 Specie selection process . . . . .
A.5 Environmental variable selection process. . . . .
A.6 Model parameters selection process. . . . .
A.7 Pre-processing parameters selection process. . . . .
A.8 Evaluation parameters selection process. . . . .
A.9 Validation process. . . . .
A.10 Model preparation process. . . . .
A.11 Model execution process. . . . .

B.1 Jackknife results for the selected 10 variables. . . . .
B.2 Response curve for February. . . . .
B.3 Response curve for March. . . . .
B.4 Response curve for April. . . . .
B.5 Partial dependence plot for the generalized addictive model. . . . .
B.6 Partial dependence plot for the random forest model. . . . .

Chapter 1

Introduction

The world's oceans face increasing pressure from human influences. Marine ecosystems are utilized by several economic sectors, namely commercial and recreational fishing, tourism and passenger transportation. Species are vulnerable to impacts from all these activities due to competition with fisheries, habitat degradation and disturbance.

Through research and monitoring of species, datasets are created in order to help understanding and managing ecosystems by the characterization of the species habitats. With a reliable dataset consisting of locations where species have been observed, a pattern of the suitable conditions of each species can be inferred. As a result, one can try to infer where each species occurs and does not occur without having to sample the whole ocean. This information can then be used to infer the status of the species.

1.1 Motivation

One concrete example of the importance of this subject is the Archipelago of the Azores.

The Archipelago of the Azores, located between 36º-40º latitude North and 20º-32º longitude West is the most isolated and extensive island group in the north-eastern Atlantic. The Azores is located in the northeast Atlantic on the Mid-Atlantic Ridge over the Azores Triple Junction (ATJ) where the North American, Eurasian and African tectonic plates meet with an average abyssal pane depth of 3000 meters. As a consequence, the archipelago is characterized by high volcanic activity typical of a ridge-hotspot interaction. The seafloor is mostly deep but over 100 seamounts, a faction of the mid-Atlantic ridge and the slopes of the islands compose the shallowest parts. Due to this unique environment, there is a great concentration of extraordinary geological formations with unique biotic adaptions that lead to the Azores recognition as UNESCO territory. Extensive scientific research based in the Azores has opened a window on the functioning of large oceanic, deep-sea and seamount ecosystems and on the impacts of human activities in such ecosystems. With the technological advances of the past few decades, much has been added to our knowledge of deep sea habitats, and people have begun to realize the value and importance of this large and remote habitat to life on Earth. Deep seabed habitats, long perceived to be a biological desert, host a wealth of species. Current estimates for species diversity

in the deep sea range between 500,000 and 10 million species. Recent scientific results highlighted that higher biodiversity can enhance the functioning and efficiency of deep sea ecosystems. Without deep sea life, life on Earth would be compromised because of the fundamental role of the deep sea in global biogeochemical cycles including nutrient regeneration and oxygen itself. As such, the sustainability of our biosphere significantly relies on the goods and services provided by deep sea ecosystems.

The open ocean and deep sea are under increasing threat from various human activities. The most pressing threats come from overfishing, destructive fishing practices and illegal, unreported and unregulated fishing activities. Other emerging problems include ship-based marine pollution, illegal dumping and noise pollution [1, 2].

Given its high levels of biodiversity and wealth of resources, spatial planning is recognized as an essential tool for effective management of all human activities occurring in the deep sea and to ensure a sustainable exploitation of its resources [3]. The success of spatial planning and the design of protected areas rely on a good understanding of the spatial distribution patterns of species. Yet, extensive sampling programs for the deep-sea are costly and technically challenging, in comparison to shallow inshore waters, where spatial planning is a much easier task.

1.2 Topic Overview

Species distribution models (SDMs) explore these relations between environmental and species, to predict the distribution of species across geographic space. However, records of observed species occurrences typically provide information on only a subset of sites occupied by a species. They do not provide information on sites that have not been surveyed, or that may be colonized in the future following climate change or biological invasions. However, this information is important for making robust conservation decisions and can be provided by predictions of species occurrences derived from environmental suitability models that combine biological records with spatial environmental data.

As technology evolves, new methods appear for biologists to model species' distributions. More commonly used methods are based on statistical approaches (e.g. regression) and newer methods are based on machine learning approaches. On this document both approaches will be presented.

1.3 Objectives

`DeepData` started as a project to help researchers of the MAR Institute [1] to study the Azores deep sea in an automatic way. It has now expanded to cover all marine ecosystems. The tool arises to help biologists with these new approaches, and as a way to facilitate their use. Our aim is to develop a web-based machine learning tool. Nowadays, biologists have to program the species distribution models, which can sometimes be hard as it is not their area of expertise. Species distribution models creation is composed of 3 main steps, which are further explained on chapter 2: (i) data pre-processing, (ii) model selection and training and (iii) model evaluation.

[1] http://www.okeanos.uac.pt/en/

The modelling of these steps might be tedious. Therefore, the DeepData tool that we are proposing will provide a simple interface, where the biologist will simply have to select the species, the environmental variables and the model to apply.

1.4 Contributions

The DeepData tool is designed to create a comprehensive modelling and simulation tool. The tool provides a unique place to create species distribution models where various methods can be used, supporting both statistical and machine learning algorithms. DeepData is also user friendly by only requiring the user to select a set of configurations. DeepData also allows the user to load data, model the data to specific characteristics according to the model selected and see the results of such selection.

DeepData's architecture is described in a MSc thesis [4]. This architecture has been revised and some changes to the database were made. The changes include adding a new database, adding primary keys to existing tables and changing the way the data is inserted in the database.

Regarding the pre-processing step, DeepData now allows for pseudo-absences to be generated and spatial autocorrelation and collinearity to be inferred. An extensive number of environmental datasets are available nowadays for fitting SDMs, but only a limited number of them should be included when running SDMs. Although increasing the number of predictors increases the chance of having ecologically relevant ones, it also inflates the risk of overfitting the model and of collinearity issues between variables. Restricting the number of variables and choosing only the most appropriate ones for each species is thus crucial to maximize the performance of SDMs and the accuracy of the predictions. The DeepData tool analyses whether the variables have collinearity issues and alerts the user. The user then decides either to remove the variables or to keep them, considering the domain knowledge concerning the ecological requirements of the species.

This work also delivers a new cross validation method, that allows the training to be performed for a set of years and the testing to be performed on another set of years, provided that each set has different years.

On the model selection and training, the tool now allows for the user to select a set of configurations, so that the user can shape the model to its data.

Finally, on the model evaluation, the user has three new ways to define the binary species distribution model:

- CCR, which is the threshold value or range in values with the maximum number of presence and absence records correctly identified;
- No omission, which is the threshold value or range in values with no omission error, meaning no false positives (predicting absences incorrectly);
- Prevalence, which is the threshold value or range in values with the modeled prevalence closest to the observed prevalence.

The user can also specify how the ensemble models are constructed.

1.5 Thesis Outline

This thesis presents the whole process involved in species distribution models. This includes not only new algorithms (e.g. random forest and support vector machine), but also the statistical methods that are more commonly used (e.g maximum entropy, generalized linear models and generalized addictive models). As well as the importance of treating data before modelling the species distribution, and the analysis of the species distribution model to verify its utility.

This document starts by describing the state of the art of the species distribution model process on chapter 2. It proceeds on describing the `DeepData` tool implementation on chapter 3 and finalizes with the tool evaluation on chapter 4 followed by the conclusion on chapter 5.

Chapter 2

Background

This chapter starts by introducing an overview of the problem and defining the domain knowledge in section 2.1. It then characterizes each piece of the process: (i) data pre-processing in section 2.2, (ii) model selection and training in section 2.3 and (iii) model evaluation in section 2.4..

2.1 Species distribution models

Species distributions models (SDMs) assume that species distributions depend on the physical environment. For example, a depth sea fish cannot be found on near coastal sea because its physiognomy cannot handle it. The concept that species distribution depends on the environment is known as an ecological niche. Therefore, this area of study is also referred to as ecological niche models. An ecological niche describes how an organism or population responds to the distribution of resources and competitors, and how in turn it alters those same factors. According to the ecological niche theory, species are constrained by their tolerance to environmental factors [5].

SDMs try to understand this ecological niche so that it is possible to explain the environment that each species depends on. By projecting this environment into geographic space, it is possible to estimate species' geographic distribution, predicting where the species could survive. Species distribution models are a very useful mechanism to monitor the variations in habitat suitability of species, impacts of climate change and studies of species delimitation [6]. To do this, SDMs use species occurrence data and environmental data. By interpolating both datasets, SDM finds a pattern that describes the ecological niche. Model usefulness and robustness is influenced by the selection of variables and modeling methods and how the relation between environmental and geographic factors is handled [7].

The SDM creation is composed of 3 main steps: (i) data pre-processing, (ii) model selection and training and (iii) model evaluation.
Each one of these steps focuses on some main points:

(i) Data pre-processing:

- gathering relevant data;

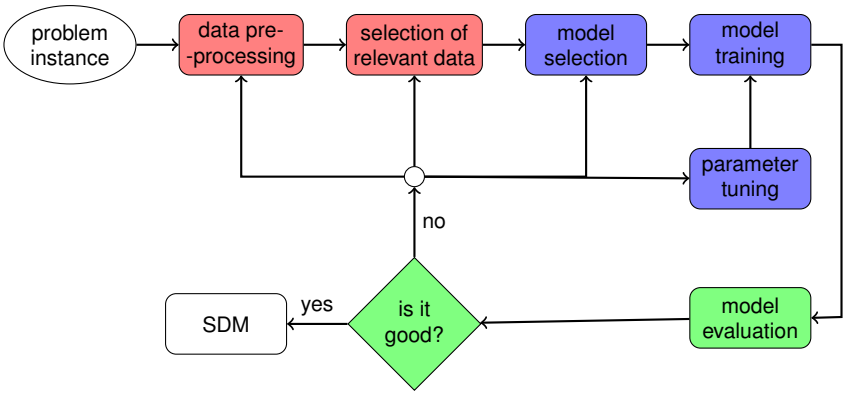


Figure 2.1: Representation of the SDM process.

- assessing its adequacy (the accuracy and comprehensiveness of the species data; the relevance and completeness of the predictors);
- deciding how to deal with correlated predictors;

(ii) Model selection and training:

- selecting an appropriate modeling algorithm;
- fitting the model to the training data [1];

(iii) Model evaluation:

- evaluating the model including the realism of fitted response functions, characteristics of residuals and predictive performance on test data (data used to provide an unbiased evaluation of the model because it has not been used to train);
- mapping predictions to geographic space;
- selecting threshold if continuous variables need reduction to a binary map;

Figure 2.1 represents the SDM process where data pre-processing is represented in red, model selection and training is represented in blue and model evaluation is represented in green. The SDM creation requires that each step is performed multiple times as evaluation is done and knowledge is gained, leading to a better fit of the SDM, as shown in figure 2.1. There is no known right way to create a SDM, only main steps that serve as guidelines. In this document, we focus on species distribution models with the purpose of predicting where might exist a suitable habitat for a species. Predictions are made to new locations within the range of environmental variables sampled by the training data and within the same general time frame as that in which the sampling occurred. We call this procedure model-based interpolation to unsampled locations [7].

[1] data through which the computer learns how to process information to learn and produce results

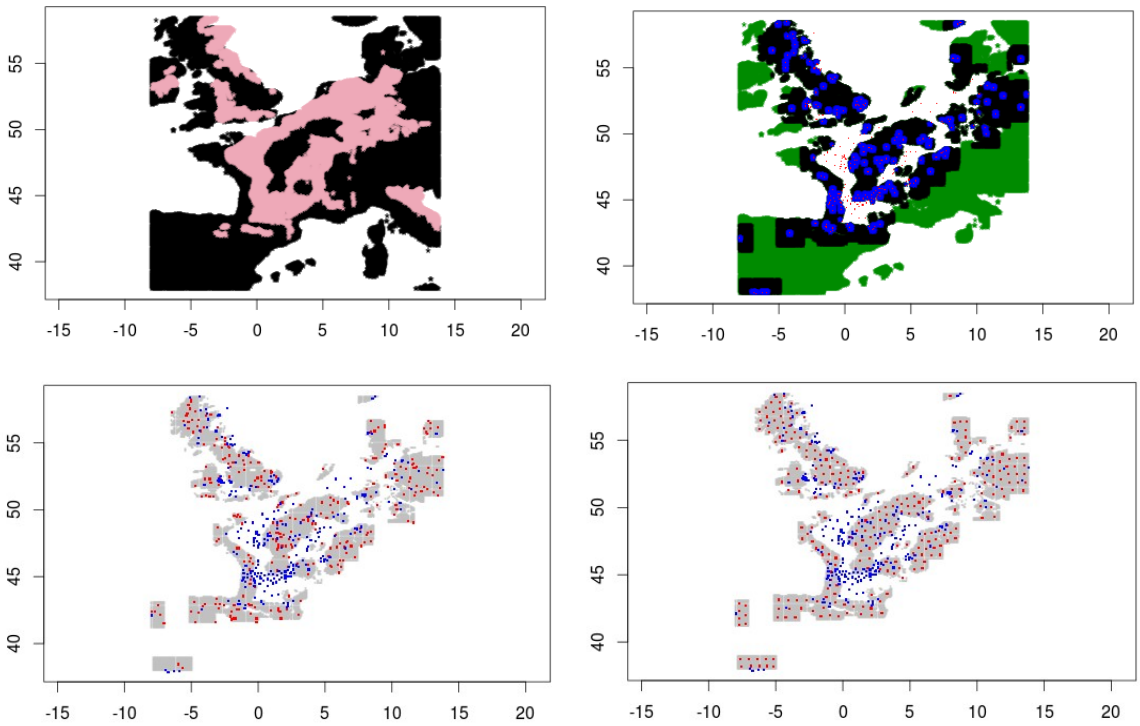


Figure 2.2: Representation of pseudo-absences generation using Mopa package [8].

2.2 Data pre-processing

SDMs relate occurrence's data with environmental data that is thought to determine the species distribution. Therefore, SDMs assume that the occurrence's data covers the species full ecological range. One of the problems of SDMs is having enough occurrence's records, as well as accurate and relevant environmental variables at a sufficiently high spatial resolution.

2.2.1 Occurrence data

Regarding the occurrence's data, the coordinates of the location data need to be accurate so that the species/environment association is reliable. Therefore, some pre-process is needed, such as:

- check for outliers, that may result from typos in coordinates;
- check for duplicate records;
- check date of records and use only the period of time that is of interest in order to reduce time changes;
- check species name uniformity.

Even taking into account all these precautions, occurrence's data might be biased towards the accessibility of sampling locations. Data may be lacking for remote areas. There are two types of occurrences data: presence only and presence/absence. Presence only refers to only having the location of where the species are present, i.e. having no knowledge of where the species is absent. Presence/absence

refers to when we have the location of both where the species are present and are absent. When dealing with absence, we have to be careful because they can mean that an habitat is unsuitable or it is suitable but unoccupied (maybe because it is inaccessible). This type of data is also tricky to get, because the fact that a species is not detected in a location at a moment in time, does not mean it does not exist there. When absence data is not available, it can be inferred based on the presence data, generating pseudo-absence. There are various methods to generate pseudo-absences, depending on the effects desired. Regardless of methods, it is always necessary to classify the background as suitable or unsuitable according to the environmental conditions of the presence localities. As shown in the top left image of figure 2.2 the suitable environment is pink and the unsuitable environment is black. Background data characterizes the environment in the location of interest. Some of the methods are:

- The background can be partitioned into distances to the presences. As shown on the top left image of figure 2.2, it is possible to define different space distances to the presences: blue represents a radius distance of 20km; black of 120km and green of 520km.

- Generation of pseudo-absences can be done at random in a given partition. As shown in the bottom left image of figure 2.2, random pseudo-absences are generated only within the radius of 120km.

- Generation of pseudo-absences can be done with k-means clustering in a given partition. As shown in the bottom right image of figure 2.2, pseudo-absences take into account the distance to each other and are generated only within the radius of 120km.

Adjusting distance and number of pseudo-absences has a great impact on accuracy and differs from problem to problem, as investigated by Barbet-Massin et. al [9] and VanDerWal et. al [10].

2.2.2 Environmental data

Regarding the environmental data, it corresponds to processed raw data. Raw data, extracted daily or hourly, does not make much sense when modelling species distribution as they are highly variable, and species respond to environmental conditions on a larger time period. Therefore, the environmental data used on SDMs corresponds to summary statistics, such as maximum and minimum values for an environment variable. This makes more sense, given that species can tolerate a threshold of an environmental variable. For example, depth sea fishes can only tolerate a given threshold of deepness due to pressure and other factors. Similarly to occurrences data, environmental data is only collected in some locations.

Environmental data need to be in grid type format, where each environmental variable is divided into grid cells representing its value for a location at some resolution (size of one individual cell/grain size). Grid resolution should be relevant to the species. For example, the resolution of the temperature when modelling plant species shouldn't be the same as when modelling fishes species since plants do not move. Also, different environmental variables can have different resolutions due to its variability. For

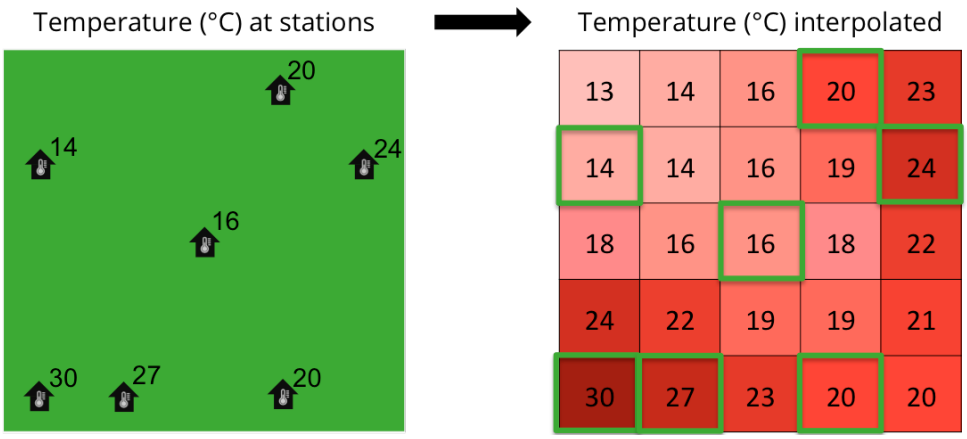


Figure 2.3: Representation of spatial inference [12].

example, temperature and the ocean topology. The resolution should be consistent with the information content of the data. Though in practice this is not always feasible [11].

Environmental data are also characterized by the coordinate reference system (CRS). The CRS provides a standardized way of describing locations. When data with different CRS are combined, it is important to transform them to a common CRS so they align with one another. This is similar to making sure that units are the same when measuring volume or distances. The CRS is composed of:

- Ellipse, which determines the shape of the earth.
- Datum, which defines an origin point of the coordinate axes and defines the direction of the axes.
- Projection, which can be unprojected meaning the locations on Earth's three-dimensional spherical surface are referenced using Latitude and Longitude. Or projected, meaning the elliptical Earth is projected onto a flat surface, and locations are computed from its ellipsoidal latitude and longitude by a standard formula known as a map projection.

To predict the values of the unknown cells, it is used spatial interpolation, which is possible due to spatial autocorrelation, explained on the next section 2.2.2. As shown in figure 2.3, in the left image, we have some places where the temperature is taken and in the right image we have the inference of the temperature to the remaining places.

Spatial autocorrelation

A model that only considers environmental variables ignores geographic proximity even when predictions are mapped into geographic space. The closer together two locations are, the more similar are their measures of species occurrences; this is known as spatial autocorrelation [13, 14]. This similarity is due to biotic processes, such as reproduction, predator-prey interactions, food availability, etc. This similarity phenomenon leads to dependence among samples decaying with distance, which violates the assumption of independence of data. Also leads to underestimation of variance and overestimation of significance of effects. A properly specified model using the adequate variables will display minimal spatial autocorrelation in its residuals, as it will be explained in section 2.4. Spatial autocorrelation can be

assessed through Moran's I measure [15]. Given the environmental data, Moran's I measure evaluates whether the pattern expressed is clustered, dispersed or random. A clustered spatial pattern means most of the values are concentrated on nearby locations or adjacent together. A random spatial pattern means the distribution of the values is homogeneous or independent. A dispersed spatial pattern means that similar values are away from each other and uniformly distributed. This measure is given as:

$$I = \frac{n}{S_0} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} z_i z_j}{\sum_{i=1}^{n} z_i^2} \tag{2.1}$$

$$S_0 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} \tag{2.2}$$

where z_i is the deviation of a variable for cell i from its mean $(x_i - \overline{X})$, $w_{i,j}$ is the spatial weight between cells i and j, n is equal to the number of cells, and S_0 is the aggregate of all spatial weights. There are two ways to deal with spatial autocorrelation [16]:

- Sub-sampling the original data set by eliminating cells in a systematic manner; For example, removing all cells with even coordinates.
- In regression, another variable can be included that accounts for the autocorrelation. This variable is normally referred to as contagion. The measure of contagion shown on equation 2.3, considers a two-order neighbourhood as the weighted average of the number of cells, among a set of k_a neighbours of a central cell y_a:

  $$contagion = \frac{\sum_{b=1}^{k_a} w_{ab} y_b}{\sum_{b=1}^{k_a} w_{ab}} \tag{2.3}$$

  where the weight given to the grid cell y_b is $w_{ab} = \frac{1}{d_{ab}}$ and d_{ab} is the distance between grid cells y_a and y_b. For the first-order neighbourhood, the eight adjacent cells touching cell y_a, d has value one. For the second-order neighbourhood, the sixteen cells concentric to the first-order, d has value two. The distance d_{ab} can be another kind of distance, such as euclidean distance.

Variable selection

Variable ranking is a method of ranking the quality of a variable to the output variable. In this context, it ranks the environmental variables with respect to the occurrences data. Various quality measures might be applied:

- Collinearity is the most used measure in the context of SDMs. It refers to the existence of correlated environmental variables, which can lead to biased models due to inflated variances. Small changes in the data set can strongly affect results and so the SDM tends to be unstable (high variance) and the relative importance of the variables is difficult to assess [17]. Pearson correlation can be used to evaluate the linear relationship between two continuous variables, and Spearman correlation can be used to evaluate if two continuous or ordinal variables change together, but not necessarily at a constant rate. Only checking the collinearity between pairs of variables can be limiting, so

the variance inflation factor (VIF) quantifies the extent of correlation between one variable and the other remaining variables.

- Information gain can be measured by Akaike information criterion (AIC). AIC estimates the relative information lost by a given model, so lower AIC means higher quality. It is used to compare models with different variable selection since its value by its own does not tell us much.

Variable selection can also be performed recursively by adding or removing variables based on some criteria as the ones previously mentioned and by verifying whether improvements were made to the model evaluation, (see on section 2.4). The recursion stops when the improvement on the model evaluation is not statistically significant.

In addiction, standardization is needed when comparing two variables of different scales. If standardization is not performed, then the SDM will favor the variables that appear to have larger variances relatively to other variables as a matter of scale, rather than true contribution. A way of ensuring that every variable has the same scale is by using:

$$x'_k = \frac{x_k - \overline{x}}{\sigma} \tag{2.4}$$

Where x'_k represents the new value of k^{th} variable x, $\overline{x}$ represents the mean value of variable x and σ represents the standard deviation of the variable x. This method makes the values of the variable have zero mean and unit variance.

The success of the prevision depends mostly on the quality of the information used, both of species and environment, because it cannot be biased and it is the base of the learning process. Therefore, pre-processing is the part that takes the longest, and is done several times along the whole process. As figure 2.1 shows, if the model is not good then we have to re-do some processes.

2.3 Model selection

Models for prediction need to balance specific fit to the training data against the generality that enables reliable prediction to new cases. Models can either use presence only data or presence/absence data.

2.3.1 Presence-only models

MaxEnt

Given a set of m occurrence samples $y_1, ..., y_m$ over some space (corresponding to the cells of the grid), and n environmental variables $x_1, ..., x_n$ defined over that space, `MaxEnt` [18, 19] tries to find the distribution of the species. The true distribution of the species is represented as a probability distribution, denoted as π. The distribution π assigns a non-negative probability $\pi(y)$ to each cell, and these probabilities sum to 1. Given our samples $y_1, ..., y_m$ chosen independently from some unknown distribution π, we create a distribution $\hat{\pi}$ which tries to approximate π.

Having the environmental variables $x_1, ..., x_n$ we know that π is characterized by the expectations of the variables under π are given by:

$$\pi[x_n] = \sum_{y \in Y} \pi(y) x_n(y) \tag{2.5}$$

where $x_n(y)$ denotes the real value of the variable x_n in the cell y. This is known as the feature expectation, and it can be approximated using our set of samples $y_1, ..., y_m$. It is natural to expect that the empirical average $\tilde{\pi}[x_n]$, given our samples, is close to its true expectation $\pi[x_n]$.

$$\tilde{\pi}[x_n] = \frac{1}{m} \sum_{i=1}^{m} x_n(y_i) \tag{2.6}$$

Therefore, when approximating $\hat{\pi}$ we have the constraint that:

$$\hat{\pi}[x_n] = \tilde{\pi}[x_n], \text{ for each feature } x_n \tag{2.7}$$

The problem now is that many distributions satisfy these constraint. Our uncertainty is expressed quantitatively by the information which we do not have about some points as:

$$H(\hat{\pi}) = -\sum_{y \in Y} \hat{\pi}(y) \ln \hat{\pi}(y) \tag{2.8}$$

This value is known as entropy, denoted as $H(\hat{\pi})$. Due to being expressed in terms of probabilities, it varies according to our knowledge. The Principle of Maximum Entropy is used to discover the probability distribution which leads to the highest value of uncertainty, ensuring that no information is wrongly assumed. On the foundation of the algorithm is the assumption that the distribution $\hat{\pi}$ coincides with species distribution π, which is not unreasonable, although it does ignore the fact that some locations are more likely to have been more visited than others. Consequently, distribution $\hat{\pi}$ is sampling biased, and will favor areas and environmental conditions that have been better sampled.

Figure 2.4 shows the probability density at predicted presences (light grey) of a species given the probability density at observed presences (dark grey) and the probability density at background (black) of the minimum July temperature. It is also represented the ratio of predicted density with background density (black line), called response curve. It shows that the probability density at predicted presences has a similar mean to the density at observed presences. However, the mode of the predicted presences is shifted towards the mode of the observed presences. This occurs because minimizing the entropy of the predicted distribution makes it as similar as possible to the density of background locations while still satisfying constraints imposed by the density of the presence locations (such as the mean).

`MaxEnt`'s main advantage is that it does not make any assumption about the absent data. By using the constraint that the expected value of each variable x_n must be equal to their empirical average, `MaxEnt` is prone to overfit.

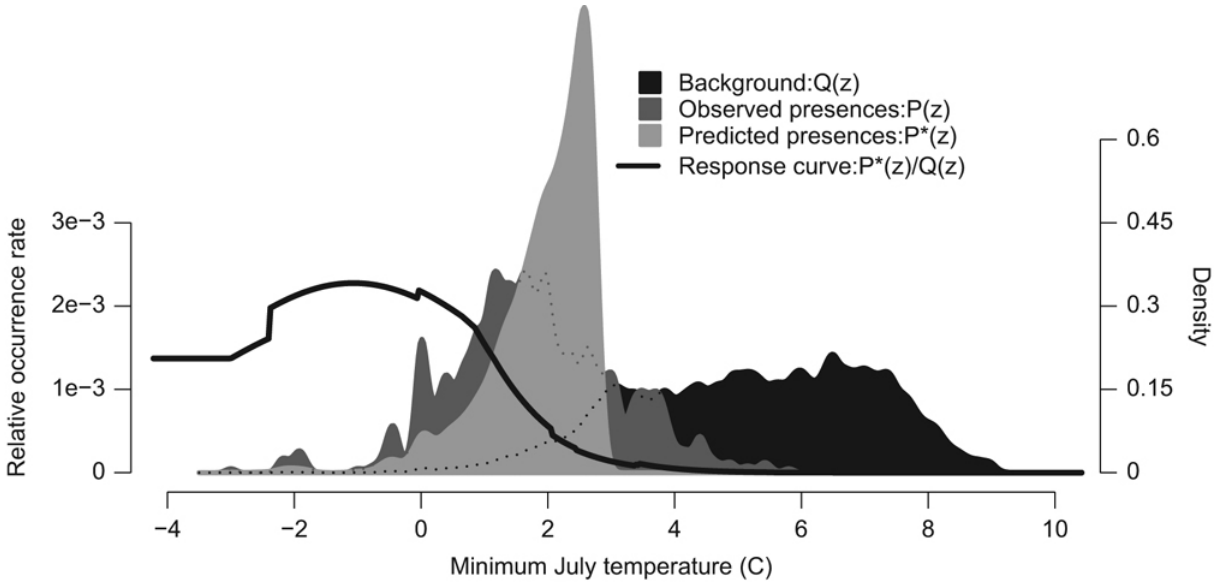


Figure 2.4: `MaxEnt` representation [20].

2.3.2 Presence/absence models

Generalized linear models

`Generalized linear models` (`GLMs`) [21] implement the idea of finding an equation that models our data. The basic idea is that given some environmental variables, and knowing the nature of the problem, we can combine the variables so that they minimize the error between our estimation and the real value.

`GLMs` are composed of three components:

1. Random component: Specifies the conditional distribution of the occurrence variable, Y, given n independent sampled observations. Specifies the family of the distribution.

2. Linear predictor: Equation linking the expected value of Y with a linear combination of the n environmental variables X_n:

$$\eta = \alpha + \beta_1 * X_1 + \beta_2 * X_2 + ...\beta_n * X_n \tag{2.9}$$

   where α is the error that is not modeled by the variables and β_n are the regression coefficients for each variable.

3. Link function: Transforms the expectation of the occurrence variable to a linear predictor:

$$g(E[Y]) = \eta = \alpha + \beta_1 * X_1 + \beta_2 * X_2 + ...\beta_n * X_n \tag{2.10}$$

   Besides mapping the expected occurrences to a linear function of environmental variables, it also removes constraints with the domain of the occurrences.

Therefore, the family of the distribution characterizes the problem, while the link function does the mapping of the domains. Different link functions might be used for each family, although some combinations

Table 2.1: Commonly used combinations of link function and distribution families

Family	Link function	$\eta = g(E[Y])$	range of $E[Y]$
gaussian	identity	$E[Y]$	$(-\infty, +\infty)$
poisson	log	$\log \exp E[Y]$	1,2,...
gamma	inverse	$E[Y]^{-1}$	(0,∞)
inverse gamma	inverse-square	$E[Y]^{-2}$	(0,∞)
binomial	logit	$\log \exp \frac{E[Y}{1-E[Y]}$	$\frac{0,1,...,n_i}{n_i}$

do not make sense, such as using the identity link function with the binomial family. Some examples of commonly used combinations are represented in Table 2.1.

Regarding the nature of our problem, in the context of species prediction one can apply three families:

- Binomial family: used to model a series of binary events, each with only two possible outcomes: 'success' or 'failure'.
- Gaussian family: Used to model continuous data that have symmetric distributions.
- Poisson family: used to model count data, such as the number of occurrences of some event in a defined time period or space, when the probability of an event occurring in a very small time/space is low and the events occur independently.

Since we know the function that describes the relationship between the occurrence and environmental data, this model is a parametric model. Having the basic structure established, this algorithm uses the training data to determine the regression coefficients, also seen as the weight of each variable, that minimize the error. This estimation is done in an iterative way. The algorithm fixes the weights, determines the parameter values that minimize the weighted sum of squared residuals, then updates the weights and repeats the process until the weights are stabilized.

`GLM's` models are limited to a set of parametric shapes, but are relatively easy to interpret and allow a clear understanding of how each of the environmental variables are influencing the occurrences.

Generalized addictive models

`Generalized addictive models` (`GAMs`) [22] do not make the assumption that the regression is linear. So `GAM's` could have the following structure:

$$Y = f(x_1, x_2, ...x_n) + \eta \tag{2.11}$$

where the regression function f can be estimated directly from the data. Unlike the `GLM` we do not estimate parameters since we do not know the structure of our relationship. This structure is fully determined by given data, allowing a more flexible estimation. The problem with this formulation is that it is difficult to estimate f when there are many variables. The sparseness of the data inflates the variance of the estimates. This is a well known problem, called 'the curse of dimensionality' [23], characterized by the rapid increase of variance with the increase of dimensionality. To deal with this, additive models

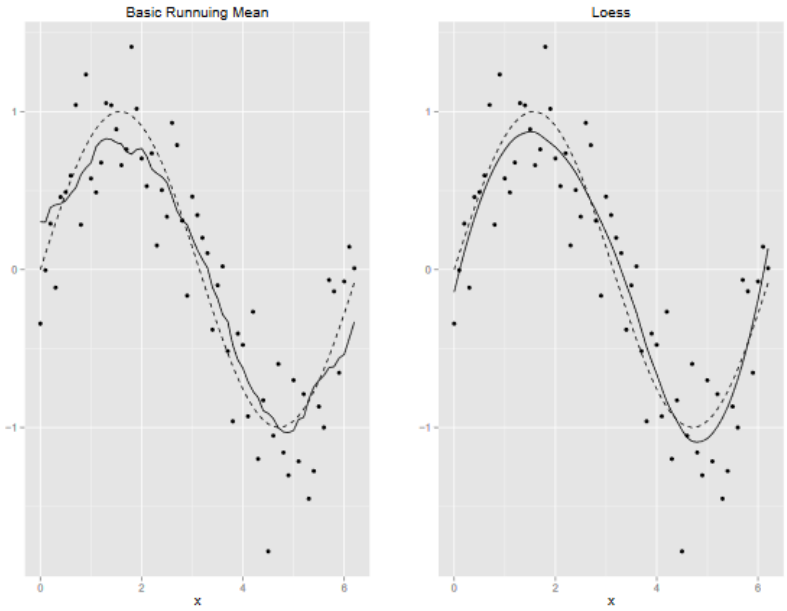


Figure 2.5: Comparison between using mean and using loess for smoothness [24].

were proposed:

$$Y = f_1(x_1) + f_2(x_2) + ... + f_n(x_n) + \eta \tag{2.12}$$

where the n partial-regression functions f_n are assumed to be smooth. To achieve this smoothness three types of classes may be applied:

1. Local regression (loess): smoothness is achieved by sliding a window on the nearest neighbors and computing a weighted average of Y at each step. The weights have into account the distance to the other data points. Level of smoothness is determined by the width of the window, as shown in figure 2.5.

2. Smoothing splines [25]: smoothness is achieved by minimizing:

$$\underbrace{\sum_{i=1}^{m}(y_1 - s(x_1))^2}_{\text{residual sum of squares}} + \underbrace{\gamma \int (s''(x))^2 \mathrm{d}x}_{\text{integrated square of the second derivative}} \tag{2.13}$$

   where the residual sum of squares ensures that we fit the observed data. Smoothness is imposed by calculating the integrated square of the second derivative, which measures the slopes of the slopes. The γ parameter controls the degree of smoothness. The solution to this minimization is to define a function between every two data points, so that every data point is fitted and therefore there is no error. At the intersection of two functions (knot), this is at a given data point represented as a blue dot in figure 2.6, the first and second derivatives of the two functions have to be the same to guarantee smoothness. This is known as natural cubic splines. The main disadvantage of this approach is that it is not practical to define a function between every data points when dealing with large amounts of data. Besides, when prediction is the goal, we generally want to generalize and avoid overfit. A comparison of this spline with the other splines is shown in figure 2.6.

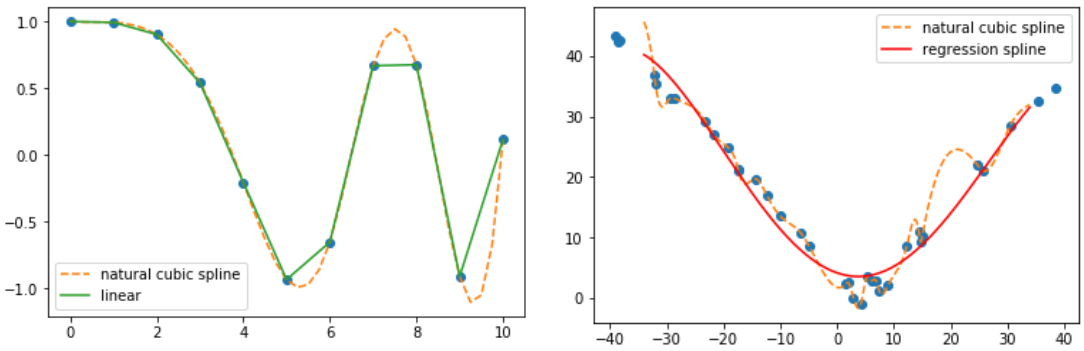


Figure 2.6: The first image compares the use of a linear and a natural cubic spline. The second image compares the use of the natural cubic spline and regression.

3. Regression splines: smoothness is achieved by a linear combination of a finite set of basis functions that do not depend on the variable Y:

$$s(x) = \sum_{j=1}^{m+k+1} \beta_j g_j(x) \tag{2.14}$$

where β_j are the coefficients and g_j are the truncated power basis functions for k^{th} order splines over the knots $t_1, ..., t_m$. Basically, we divide the data in m sections and every section fits a function. As shown in figure 2.6, regression is preferred to natural cubic spline with many data, since it does not overfit.

`GAM's` models are very flexible, being easy to adjust with the use of smoothing functions. One downside is that it does not allow interaction between variables to be modelled. Variables are only added, i.e one cannot model this function: $Y = \frac{1}{f(x1)+f(x2)}$; whereas in `GLMs` one can model this function: $\eta = \frac{1}{x1+x2}$, using the inverse link function.

Classification and regression trees

`Classification and regression trees` (`CARTs`) [26, 27] are composed of:

- Nodes, which are rules for splitting data based on the value of the variable.
- Branches, which correspond to the results of the splitting.
- Leaf nodes, which contain the prediction of a given input.

When our response variable is categorical, we want to distinguish classes and therefore we have a classification problem. When our response variable is numeric or continuous, we want to predict its value and this is a regression problem. Basically, what `CART` does is recursively dividing the variable space into two smaller partitions. This division is done in a greedy way. All variables and all possible splits are analyzed and the best one is chosen at each step.

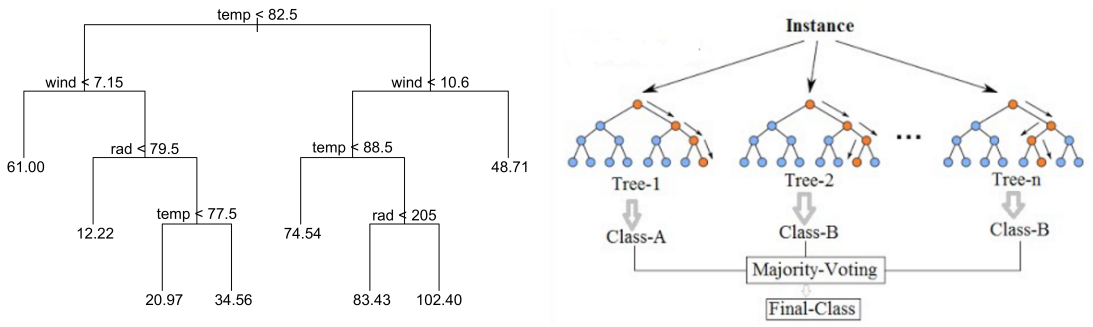


Figure 2.7: Representation of the difference between `CART` [28], top image, and `Random Forest` [29], bottom image.

For regression problems, we try to minimize the sum squared error across all samples Y:

$$\sum_{i=1}^{|Y|}(y_i - prediction_i)^2 \tag{2.15}$$

For classification, the Gini index, which measures the impurity of the leafs, is used. For a leaf node l with C classes and $p(c|l)$ the probability of class c in l, the Gini index $gini(l)$ is:

$$gini(l) = 1 - \sum_{c=1}^{C} p(c|l)^2 \tag{2.16}$$

The $gini_{split}$ index summarizes the gain of using a variable V to make the split into two leafs:

$$gini_{split}(V) = \sum_{i=1}^{2} \frac{N_i}{N} gini(l_i) \tag{2.17}$$

where $\frac{N_i}{N}$ represents the proportion of each branch that we are using to split with respect to the whole possible branches. It attributes weights to each leaf. A common stopping criteria is to set a minimum number of samples assigned to a leaf. It helps to avoid overfit and generalize. Another method to avoid overfitting is pruning. It essentially goes through each leaf node and evaluates the impact in performance of removing the leaf. A leaf is removed if the increase in error is below a given threshold.

`CART` has many advantages such as: (i) being non-parametric and making no distribution assumptions, (ii) not requiring data transformation and being invariant to outliers and (iii) handling high dimensional data. It also has its drawbacks such as: (i) being vulnerable to overfit and (ii) exhibiting high variance and small changes in the data can which result in different splits making interpretation unstable.

Random Forest

`Random Forest` (`RFs`) [30] is based on `CART` and bagging approaches. Essentially it combines various decision trees to achieve more accurate predictions than any individual tree. A method that gathers a set of weak-learners to form one strong learner is called an ensemble method.

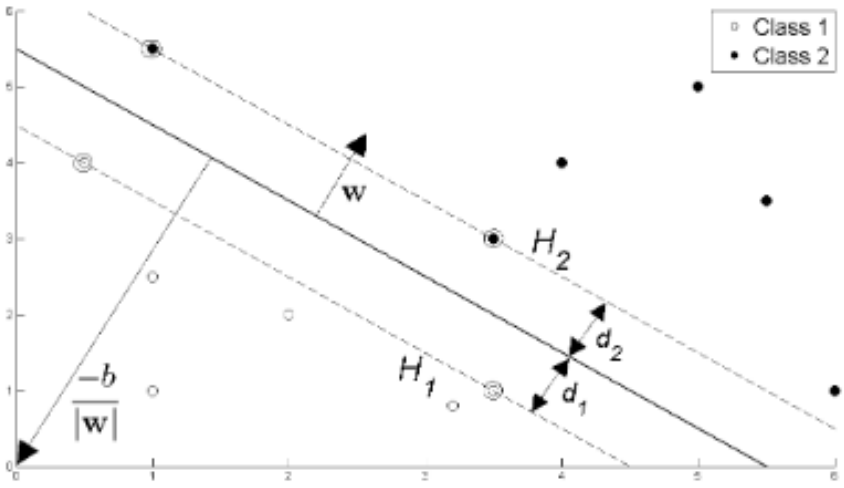


Figure 2.8: Hyperplane through two linearly separable classes [32].

Bagging is the combination of bootstrapping and aggregating. Bootstrapping consists in creating many sub-samples of the training set with replacement meaning that an entry can occur more than once in a sub-sample. Taking a sample of the training set allows the sub-sample to contain different characteristics than it might have contained as a whole. It helps to decrease the variance of the classifier and reduce overfitting. Bagging consists on aggregating the predictions of the models with the different sub-samples to get an overall prediction. `CART` has high variance because it is data sensitive, meaning that the prediction depends on the training data of the algorithm. Hence, `Random Forest` consist on using bagging when the model applied to each sub-sample is `CART`. Because we are using bagging, we are less concerned with overfit on each tree, and no pruning is made. The problem with using `CART` and bagging is that `CART` is greedy and each tree has a lot of structural similarities ending in high correlated predictions. `Random Forest` changes the variable selection of `CART` by only allowing it to look at n random variables.

Support Vector Machines

`Support Vector Machines` (`SVMs`) [31] identifies the two classifiers by separating them through a linear partition on the variable space, as shown in figure 2.8. A new instance is either above or below that partition. If we only have two variables, then the partition is a line. If we have more than two variables, then the partition is an hyperplane.

This hyperplane is described by $w \cdot x + b = 0$ where:

- w is normal to the hyperplane.
- $\frac{b}{\|w\|}$ is the perpendicular distance from the hyperplane to the origin.

There are many hyperplanes that separate the data. `SVM` select the one that is farthest from the closest members of both classes. These data points that are closest to the hyperplane are called support vectors, represented in figure 2.8 as $H1$ and $H2$.

The hyperplanes containing the support vectors for each class are:

$$\begin{aligned} x_i \cdot w + b \geq +1 \quad for \quad y_i = +1 \\ x_i \cdot w + b \leq -1 \quad for \quad y_i = -1 \end{aligned} \tag{2.18}$$

Where i represents the components of class i. These equations combined imply:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \tag{2.19}$$

After having the hyperplanes defined, we want to maximize its width, which corresponds to the difference of the support vectors x_+ and x_-:

$$width = (x_+ - x_-) \cdot \frac{w}{\|w\|} \tag{2.20}$$

where $y_i(x_i \cdot w + b) - 1 = 0$ for the support vector so knowing that for x_-, y_i is -1 and that for x_+, y_i is 1 we have,

$$width = (1 - b + 1 + b) \cdot \frac{w}{\|w\|} = \frac{2}{\|w\|} \tag{2.21}$$

In order to maximize the width we need to minimize $\|w\|$ and for convenience this is the same as:

$$\text{minimize } \tfrac{1}{2}\|w\|^2 \text{ subject to } y_i(x_i \cdot w + b) - 1 \geq 0$$

Using Lagrange multipliers α we get:

$$L = \min_{w,b} \max_{\alpha > 0} \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha[y_i(x_i \cdot w + b) - 1] \tag{2.22}$$

We want to find w and b that minimize L so we have to find the derivatives and set them to 0:

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.23}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.24}$$

The Lagrangian dual problem consists on instead of minimizing w and b subject to constraints involving α, we can maximize over α subject to constraints involving w and b.

$$L = \max_{\alpha > 0} \min_{w,b} \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha[y_i(x_i \cdot w + b) - 1] \tag{2.25}$$

By substituting 2.23 and 2.24 into 2.22 we get a new formulation depending on α that we need to maximize:

$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{2.26}$$

As a result, the solution to α_i only depends on the inner product of $x_i \cdot x_j$. So most α_i will be 0 besides the ones associated with support vectors. To solve b, we know that any data point that is a support

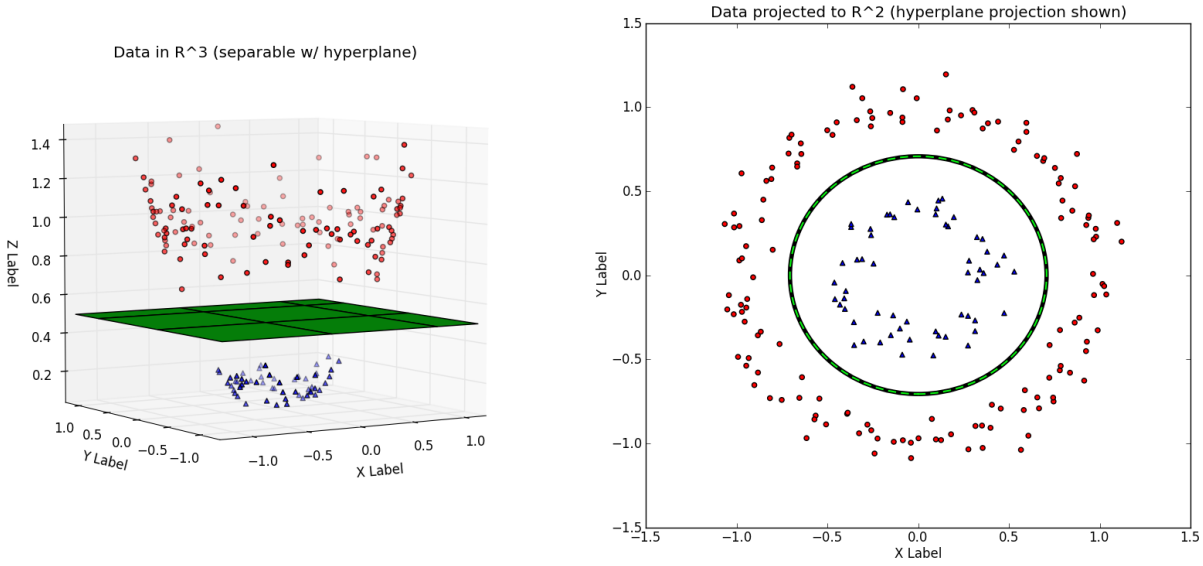


Figure 2.9: Representation of feature space manipulation. In the first image the data is linearly separable, while in the second image the data is nonlinear [33].

vector x_s will respect:

$$y_s(x_s \cdot w + b) = 1 \tag{2.27}$$

Substituting with 2.23:

$$y_s(\sum_{m \in S} \alpha_m y_m x_m \cdot x_s + b) = 1 \tag{2.28}$$

The support vectors S are the ones that have $\alpha_m > 0$. For mathematical convenience we multiply by y_s:

$$\begin{aligned} y_s^2(\sum_{m \in S} \alpha_m y_m x_m \cdot x_s + b) &= y_s \\ b &= y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s \end{aligned} \tag{2.29}$$

Instead on using an arbitrary support vector x_s, it is better to take an average over all of the support vectors:

$$b = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s) \tag{2.30}$$

When the data is not linearly separable in the variable space we can apply a transformation ϕ to make the data linearly separable in another variable space, as shown in figure 2.9. What we would have to maximize is then:

$$L = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j \phi(x_i) \cdot \phi(x_j) \tag{2.31}$$

Since the maximization only depends on the inner product, then we only need $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ known as the kernel function. Table 2.2 presents some examples of kernel functions.

`SVM's` do not require many knowledge about the data and with an appropriate kernel function it can solve any complex problem, but the choice of an adequate kernel function is not obvious.

Table 2.2: Examples of kernel functions.

Name	expression
gaussian	$\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
polynomial	$(x_i \cdot x_j + a)^b$
sigmoidal kernel	$\tanh ax_i \cdot x_j - b$

Artificial neural networks

`Artificial neural networks` (`ANNs`) [34, 35] are inspired on the capacity of the human brain to learn. Some problems depend on many factors and therefore are difficult to formulate as an algorithm. The human brain is composed of nerve cells or neurons, that communicate between them with electrical signals. These electrical signals are transmitted through branches called dendrites, assuring that all neurons are connected. Neurons work like switches. According to the electrical input, they either emit or hold an electrical response.

On `ANNs` the activation function is associated with the neuron that makes the neuron switch. This activation function can output 0 meaning no electrical impulse or 1 meaning an electrical impulse. This equivalent to the biological neuron is called a perceptron [36].

A perceptron can have many inputs, as shown in figure 2.10, which can be outputs from other perceptrons, as they are connected. The $w1$, $w2$ and $w3$ represent the different weights of the different inputs $x1$, $x2$ and $x3$, respectively. These weights distinguish between the importance of each input. So the output can be formulated as:

$$output = \begin{cases} 0, & \text{if } \sum_j w_j x_j \leq \text{threshold.} \\ 1, & \text{if } \sum_j w_j x_j > \text{threshold.} \end{cases} \tag{2.32}$$

Which can be written as:

$$output = \begin{cases} 0, & \text{if } \sum_j w_j x_j + b \leq 0. \\ 1, & \text{if } \sum_j w_j x_j + b > 0. \end{cases} \tag{2.33}$$

where b represents the threshold and is called the preceptron's bias. As a way to make the transition be-

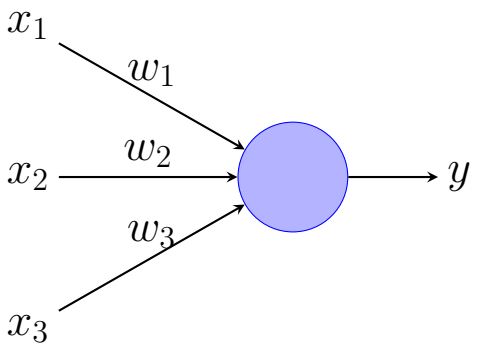


Figure 2.10: Perceptron ilustration [37].

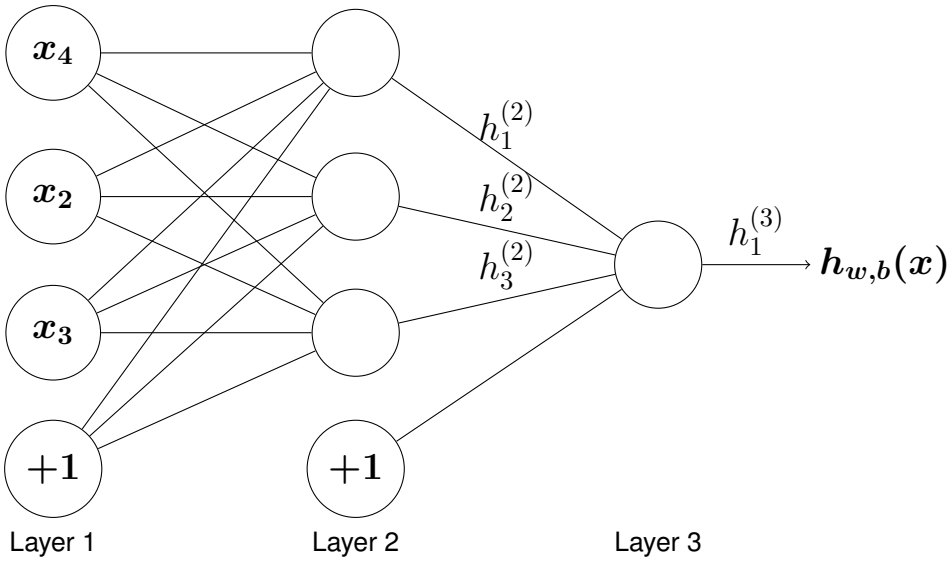


Figure 2.11: `Artificial neural network` illustration [37].

tween electrical and no electrical impulse more subtle, the sigmoid function can be used as an activation function:

$$f(z) = \frac{1}{1 + \exp -z} \tag{2.34}$$

where z represents the input:

$$z = \sum_j w_j x_j + b \tag{2.35}$$

As figure 2.12 shows, the use of sigmoid makes changes smooth. While having a threshold defining the output makes that little changes on the input produce a completely different output, sigmoids make these little changes more subtle. Varying the weight w of the branches, varies the steepness of the sigmoid (see figure 2.12). While the bias b varies the shift of the sigmoid on figure 2.12. Having defined the single component of the `ANN`, we can know define the `ANN` as a composition of perceptrons connected with each other. A common simple structure for `ANN` can be seen in figure 2.11, where the layer 1 represents the input layer, the layer 2 represents the hidden layer and layer 3 represents the output layer. `ANN` can have more than one hidden layer, and each layer can have more perceptrons, but all

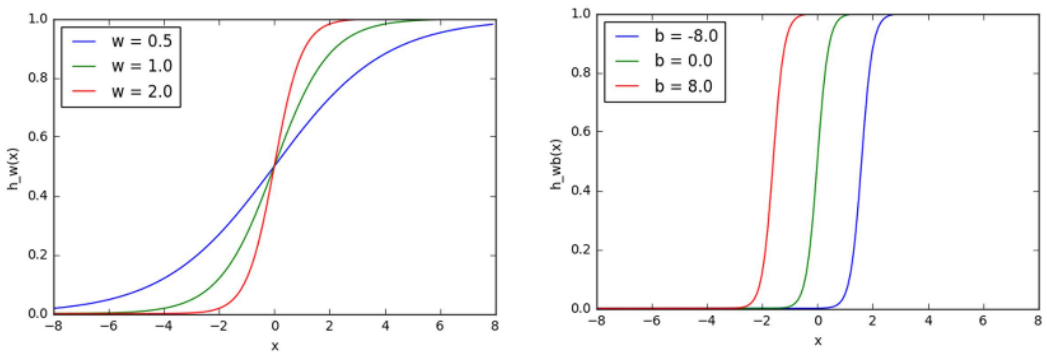


Figure 2.12: Representation of the effect of weights, left image, and bias, right image, on sigmoid [37].

of the 3 layers have to exist, being this representation the simplest one. On figure 2.11, we can also observe that each node of layer 1 connects to every perceptron on layer 2, and each perceptron on layer 2 connects to the perceptron on layer 3.

Having defined the structure of the ANN, we can now define the learning algorithm. Take the figure 2.11, given an input x composed of neuron $x1$, neuron $x2$ and neuron $x3$, the output $h_{w,b}(x)$ of the ANN is iteratively calculated by first calculating the output of $x1$ in the second layer, the output of $x2$ in the second layer and finally the output of $x3$ in the second layer, respectively represented by $h_1^{(2)}$, $h_2^{(2)}$ and $h_3^{(2)}$. This process is called feed-forward process:

$$\begin{aligned} h_1^{(2)} &= f(w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + w_{13}^{(1)}x_3 + b_1^{(1)}) \\ h_2^{(2)} &= f(w_{21}^{(1)}x_1 + w_{22}^{(1)}x_2 + w_{23}^{(1)}x_3 + b_2^{(1)}) \\ h_3^{(2)} &= f(w_{31}^{(1)}x_1 + w_{32}^{(1)}x_2 + w_{33}^{(1)}x_3 + b_3^{(1)}) \\ h_{w,b}(x) = h_1^{(3)} &= f(w_{11}^{(2)}h_1^{(2)} + w_{12}^{(2)}h_2^{(2)} + w_{13}^{(2)}h_3^{(2)} + b_1^{(2)}) \end{aligned} \tag{2.36}$$

where $w_{ij}^{(l)}$ represents the weight of layer l on position j to layer $l+1$ on position i and $b_i^{(l)}$ represents the bias no layer l associated with the perceptron of layer $l+1$ on position i. In equations 2.36, $f()$ refers to the activation function. It is clear that the output of the ANN varies with the weights and bias, therefore by varying these factors we change the output. We want our ANN to predict as best as possible the output, so we want to minimize the error, which can be defined as a cost function, $J(w,b)$, defined by the weights w, bias b for a given input x and output y:

$$\begin{aligned} J(w,b) &= \frac{1}{m}\sum_{z=0}^{m}\frac{1}{2}\left\|y^z - h_{w,b}(x^z)\right\|^2 \\ J(w,b,x,y) &= \frac{1}{m}\sum_{z=0}^{m}\frac{1}{2}\left\|y^z - y_{pred}(x^z)\right\|^2 \end{aligned} \tag{2.37}$$

The cost function represents the mean squared error (MSE) over all training set m. The $\frac{1}{2}$ in the expression 2.37 is to facilitate the derivative of the cost function. So now we want to find the weights and bias that minimize this function, which can be done with gradient descent.

Take the figure 2.13 as an example of the variation of error, given by the cost function, with one weight. The x cross represents the minimum that we want to achieve and we start at a random position, represented by 1. If we calculate the gradient of the error with respect to the weight w, represented as the slope at that point, we know not only how fast the error function varies at that point but also the direction of the minimum of the function. In the represented case, the gradient is negative with respect to an increase in the weight, so a step in that direction will decrease the error. Therefore, we can update the weight according to:

$$w_{new} = w_{old} - \alpha * \nabla error \tag{2.38}$$

where w_{new} denotes the new weight, w_{old} denotes the previous weight, $\nabla error$ the gradient of the error and α the step size. The step size determines how quickly the solution converges, however if too big the solution can bounce around the minimum without converging. As the weight approaches the minimum,

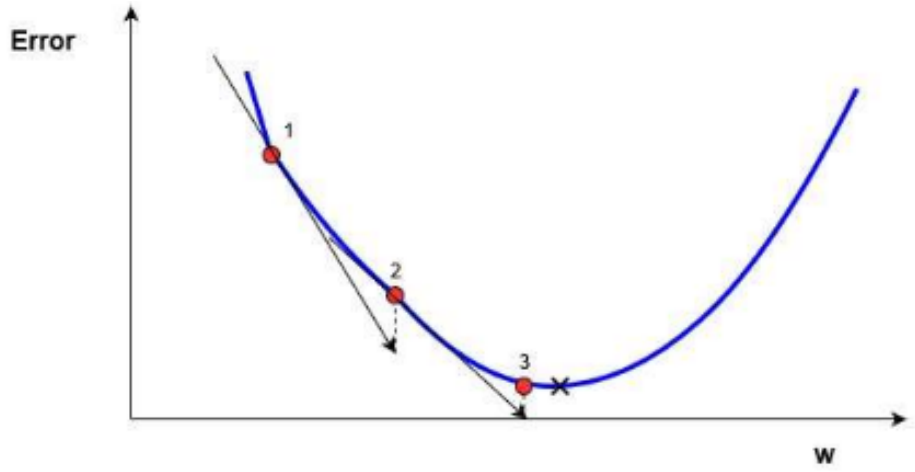


Figure 2.13: One dimensional gradient descent representation [37].

the gradient will reduce and the slope will 'flatten out. As this happens we want to stop the algorithm. Our error function is $J(w, b)$, so by substituting in equation 2.38 we get:

$$w_{new} = w_{old} - \alpha \nabla J(w, b) \tag{2.39}$$

since $\nabla J(w, b)$ is the collection of all partial derivatives with respect to each weight and bias, if we consider them individually we get:

$$\begin{aligned} w_{ij}^{l} &= w_{ij}^{l} - \alpha \frac{\partial J(w, b)}{\partial w_{ij}^{l}} \\ b_{i}^{l} &= b_{i}^{l} - \alpha \frac{\partial J(w, b)}{\partial b_{i}^{l}} \end{aligned} \tag{2.40}$$

To determine the partial derivative $\frac{\partial J(w,b)}{\partial w_{ij}^{l}}$, recall the feed-forward process illustrated on equations 2.36. We know that:

$$h_{w,b}(x) = h_{1}^{(3)} = f(w_{11}^{(2)} h_{1}^{(2)} + w_{12}^{(2)} h_{2}^{(2)} + w_{13}^{(2)} h_{3}^{(2)} + b_{1}^{(2)}) \tag{2.41}$$

For simplicity lets consider:

$$\begin{aligned} h_{1}^{(3)} &= f(z_{1}^{(2)}) \\ z_{1}^{2} &= w_{11}^{(2)} h_{1}^{(2)} + w_{12}^{(2)} h_{2}^{(2)} + w_{13}^{(2)} h_{3}^{(2)} + b_{1}^{(2)} \end{aligned} \tag{2.42}$$

Take the case that we want to update the weight $w_{12}^{(2)}$. With use of the chain rule we conclude that:

$$\frac{\partial J}{w_{12}^{(2)}} = \frac{\partial J}{\partial h_{1}^{(3)}} \frac{\partial h_{1}^{(3)}}{\partial z_{1}^{(2)}} \frac{\partial z_{1}^{(2)}}{\partial w_{12}^{(2)}} \tag{2.43}$$

Which can be further decomposed on:

$$\frac{\partial J}{\partial h_{1}^{(3)}} = -(y - h_{1}^{(3)}) \tag{2.44}$$

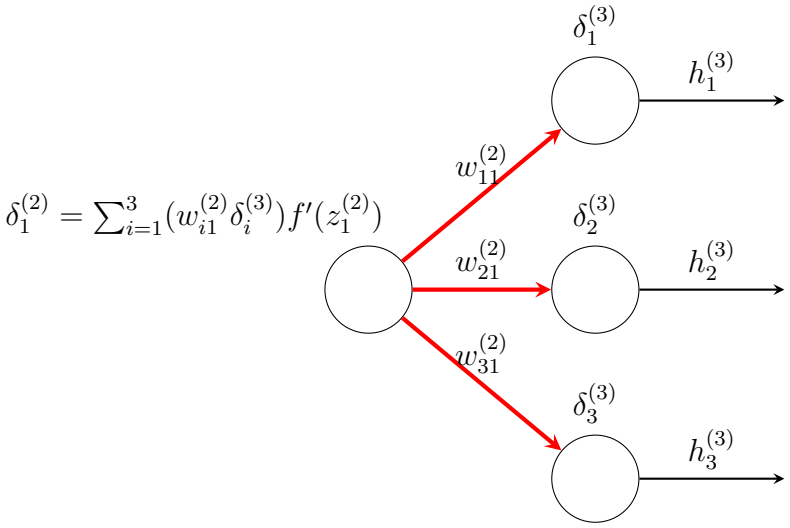


Figure 2.14: Backpropagation example [37].

Considering the case that the activation function is the sigmoid function:

$$\frac{\partial h_1^{(3)}}{\partial z_1^{(2)}} = f'(z_1^{(2)}) = f(z_1^{(2)})(1 - f(z_1^{(2)})) \tag{2.45}$$

And finally:

$$\frac{\partial z_1^{(2)}}{\partial w_{12}^{(2)}} = \frac{\partial}{\partial w_{12}^{(2)}}(w_{11}^{(1)}h_1^{(2)} + w_{12}^{(1)}h_2^{(2)} + w_{13}^{(1)}h_3^{(2)} + b_1^{(1)}) = h_2^{(2)} \tag{2.46}$$

Now take the case where the perceptron is not on the output layer, represented as layer 2 on figure 2.11. Since the hidden layer is not directly connected with the cost function, we use the back-propagation method [38]. Hence, if we want to adjust the weight $w_{11}^{(1)}$ we need take into account all the dependencies until reached the output layer, so:

$$\frac{\partial J}{\partial w_{11}^{(1)}} = \frac{\partial J}{\partial h_1^{(3)}}\frac{\partial h_1^{(3)}}{\partial z_1^{(2)}}\frac{\partial z_1^{(2)}}{\partial h_1^{(2)}}\frac{\partial h_1^{(2)}}{\partial z_1^{(1)}}\frac{\partial z_1^{(1)}}{\partial w_{11}^{(1)}} \tag{2.47}$$

For simplification, lets consider that:

$$\delta_i^l = \frac{\partial J}{\partial h_i^{(l)}}\frac{\partial h_i^{(l)}}{\partial z_i^{(l-1)}} = -(y - h_i^z)f'(z_i^{(l-1)}) \tag{2.48}$$

We can rewrite equation 2.47 with 2.48 as:

$$\frac{\partial J}{\partial w_{11}^{(1)}} = \delta_1^{(3)}w_{11}^{(2)}f'(z_1^{(2)}) \tag{2.49}$$

Consider a second example, illustrated on figure 2.14, where there are 3 outputs that contribute to the cost function. These contributions are taken into account according to the weight of the branch, so:

$$\delta_j^{(l)} = (\sum_{i=1}^{N} w_{ij}^{(l)}\delta_i^{(l+1)})f'(z_j^{(l)}) \tag{2.50}$$

Where j is the perceptron number in layer l, i is the perceptron number in layer $l+1$ and N is the number of perceptrons on layer $l+1$. Finally, we conclude that the general formula do readjust the weights is:

$$\frac{\partial}{\partial w_{ij}^{(l)}} J(w,b) = h_j^{(l)} \delta_i^{(l+1)} \tag{2.51}$$

ANN allow to model with nonlinear data with large number of inputs. Although, since ANN are black boxes we cannot know how much each input variable is influencing the output variables. Besides there is no specific rule for determining its structure.

Ensemble models

Ensemble models [39] use multiple learning models to obtain better predictive performance than the performance of a single model. A single model can have biases and inaccuracies that affect the reliability. By combining the decisions of different models, these effects can be reduced, improving the overall performance. This is due to the fact that correct answers are reinforced while incorrect ones then to be blended. The combination of models can be done in various ways:

- Maximum voting,
- Averaging,
- Weighted averaging,
- Bagging.

In maximum voting, each model does a prediction to each data point. Each of these predictions is considered a vote. Then the final prediction, meaning the prediction of the ensemble, corresponds to the majority.

Averaging method is similar to maximum voting, but instead of the final prediction being the majority, it is the average of all the single predictions.

In weighted average, instead of being a simple average of the predictions, it gives a weight to each prediction. This weight defines the importance of each model, which can be accessed through some evaluation metric, such as accuracy.

Bagging is quite different from the other approaches as it only uses one model but trains it various times. Bagging means bootstrap aggregation, and bootstrap is the process of selecting random samples with replacement from the dataset and using these samples to train the model. Then, the model is trained on each of the bootstrap samples and the final model is an aggregation of the sample models.

For regression problems, the outcome is an average of all the models while for classification problems, the outcome is based on voting.

2.4 Model evaluation

Assessing the model performance [40–42] helps determining its suitability and the aspects that need improvement. Without a relevant accuracy assessment, the model has no value. It also allows comparing different models. When assessing the performance, the type of response of the model needs to be taken into account. The response can be either quantitative (continuous responses such as the ones of `GLM`) or qualitative (categorical responses such as the ones of `SVM`). A way of checking whether there is overfit is using the k-fold classification. With k-fold the entire dataset is partitioned into k folds and the model is estimated k times with each run of the model using $k-1$ folds to train the model and the remaining fold to evaluate it, such that each fold is used only once for model evaluation. An example using 10 folds is presented in figure 2.15.

A good value of k ensures that the best possible estimate of the model is done. The k value needs to be large enough to minimize bias, as the training data needs to be as close as possible as we were using the entire dataset. However, it also needs to be small enough so that the testing set is significant, minimizing the variance. With the error measurements we can assess the model's bias by checking if the mean of the errors is low, indirectly ensuring that the model is accurate. To assess the model's variance, we can check whether the standard deviation is high, meaning the model's performance varies a lot with the dataset, not being able to generalize.

2.4.1 Qualitative response variables

Performance can be assessed by constructing a confusion matrix, whose structure is presented in Table 2.3, and from which we can derive various measures of performance, some of which are shown in Table 2.4.

The accuracy measures the proportion of correctly predicted instances. However, when the proportion of classes, known as prevalence, is not the same, the measure is overly optimistic. Accuracy is prevalence sensitive. Take the case of rare species: if the model learns to always classifies as absent,

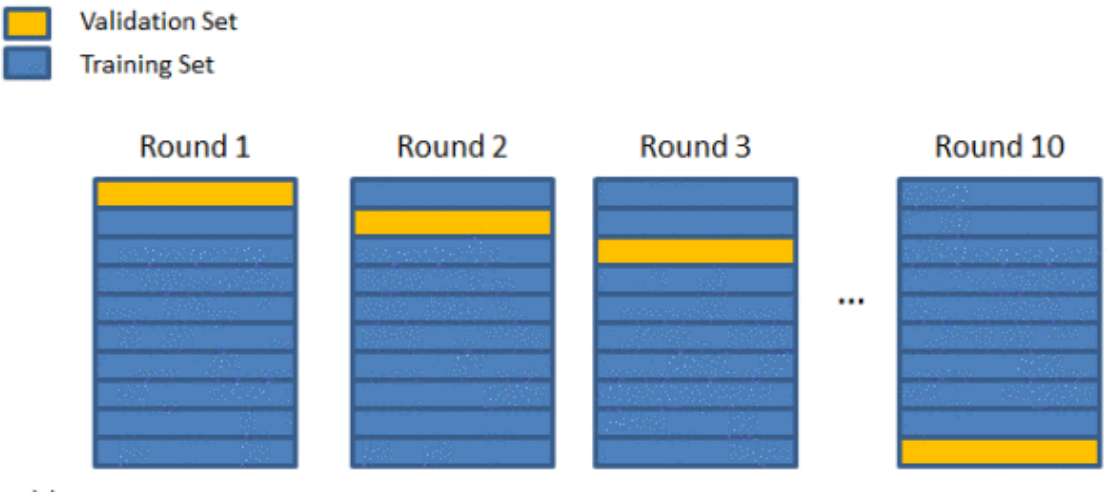


Figure 2.15: 10-fold cross validation representation [43].

Table 2.3: Structure of a confusion matrix.

		Actual positive	Actual negative	
Predicted	positive	TP	FP	PP
	negative	FN	TN	PN
		AP	AN	N

Table 2.4: Performance measures.

measure	formula
accuracy	$\frac{TP+TN}{N}$
sensitivity	$\frac{TP}{TP+FN}$
specificity	$\frac{TN}{TN+FP}$
kappa statistic	$\frac{\frac{TP+TN}{N} - \frac{PP*AP+PN*AN}{N^2}}{1-\frac{PP*AP+PN*AN}{N^2}}$

then the accuracy will be high although the model does not predict anything. Other measures, such as sensitivity, which measures the proportion of observed presences that are predicted as such, and specificity, which measures the proportion of observed absences that are predicted as such, are independent to the proportion of presence/absence. Take the same case of rare species: if the model always classify as absent, then sensitivity will be 0, because none of the presences were identified.

The Kappa statistic corresponds to $\frac{(observed_agreement)-(expected_agreement)}{1-(expected_agreement)}$. It assesses the extent to which models predict occurrence at a rate higher than expected by chance. The numerator represents the discrepancy between the observed probability of success and the probability of success under the assumption of independence. Independence implies predictions are random.

The discriminative capacity of a model can be evaluated graphically by plotting $p(\pi|y = 1)$ and $p(\pi|y = 0)$ and examining the degree of overlap. To classify an instance as one class or another, a threshold must be defined (vertical line in figure 2.16). For a given threshold, the proportion of presence sites falling to the right of this threshold defines the sensitivity, while the proportion of absence sites falling to the left of the threshold defines the specificity. The proportion of absence sites to the right of the threshold is the false positive fraction, while the proportion of the presence sites to the left of the threshold is the false negative fraction. The choice of the decision threshold in SDM is usually based on domain knowledge. For example, if a species is endangered and the model is intended to identify potential re-introduction sites, then it is important that the habitat is indeed suitable. Therefore, a high threshold would result in the identification of only the sites with a high predicted probability of presence. The choice of an actual threshold value is therefore almost arbitrary and has strong effects. Knowing that the number of correct predictions and the number of incorrect predictions must add to the number

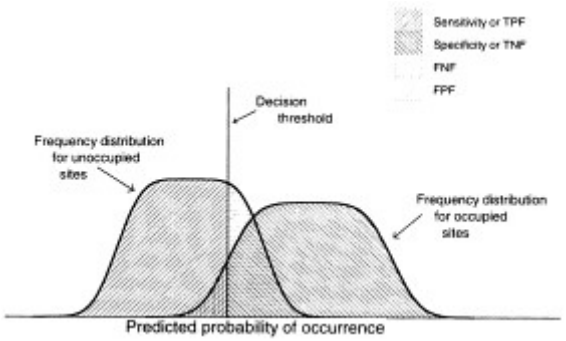


Figure 2.16: Representation of discrimination between classifications [40].

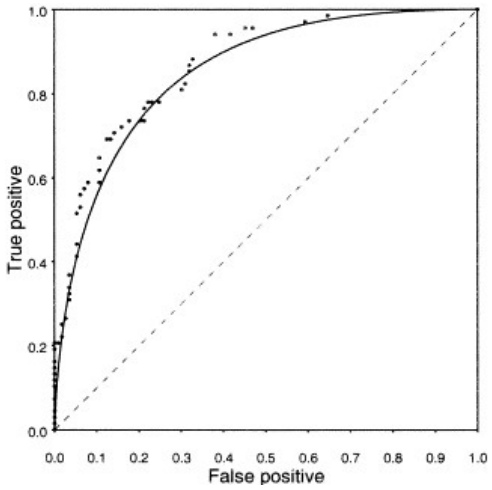


Figure 2.17: Representation of a ROC curve [40].

of observations, we conclude that:

$$\begin{aligned} p\left(\frac{present}{x=1}\right) + p\left(\frac{absent}{x=1}\right) &= 1 \\ p\left(\frac{absent}{x=0}\right) + p\left(\frac{present}{x=0}\right) &= 1 \end{aligned} \tag{2.52}$$

Therefore if we only specify $p(\frac{present}{x=1})$, which corresponds to the sensitivity, and $p(\frac{present}{x=0})$, which corresponds to the false positive fraction, we describe the whole model. We can now vary the decision threshold across this new area, and plot the pairs of sensitivity and false positives. These points define a smooth curve, called the relative operating characteristic (ROC) curve, shown in figure 2.17.

The ROC curve describes the compromise made between sensitivity and false positive as the decision threshold varies. Also, it is not sensitive to prevalence as sensitivity and false positives are measured as a proportion of all sites within a given observed state. Therefore, ROC analysis is independent of species prevalence and the decision threshold. The Area Under the ROC Curve (AUC) is used as a global metric predicting the overall discriminatory ability of the model. The AUC is the probability that a randomly chosen presence site will be ranked above a randomly chosen absence site. When no absence is available, such as with `MaxEnt`, then the AUC is the probability that a randomly chosen presence site will be ranked above a randomly chosen background site. An AUC of 1 indicates a perfectly predicting model and an AUC of 0.5 indicates that a model is as good as if it were randomly generated.

2.4.2 Quantitative response variables

There are many ways to assess the error of the model, some of which are presented in Table 2.5.

Mean absolute error (MAE) measures the average magnitude of the error, without considering their distance. Mean square error (MSE) and root mean square error (RMSE) give more weight to large error, because of the square. The fact that RMSE uses the root makes it easier to interpret its value. Error assessment can also be done by visualizing the residual vs fitted values. A residual is the difference between the observed and the predicted. This plot tests the assumptions of whether the relationship

Table 2.5: Formula of different performance metrics.

measure	formula
Mean absolute error	$\frac{1}{n}\sum_{i=1}^{n}\|(y_i - \hat{y_i})\|$
Mean square error	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$
Root mean square error	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2}$

between your variables is linear and whether there is equal variance along the regression line. Recalling the regression equation, both in GLM and in GAM, we had a noise factor, denoted η. The remaining part of the equation is the part fitted by the model. Therefore, if the regression is well fitted, then only the noise remains to explain and so the residual will be random. If the residual is not random, then it means that something is missing in the model. Also, if the residuals have a mean different from zero, then the regression is biased. It might also help to see the relationship between each environmental variable and the occurrences, after modelling every environmental variable, which is known as partial residuals. Partial residuals represent the residual of a variable in regards to the occurrences after subtracting the contribution of the other variables.

Another visual tool is the quantile-quantile plots (QQplots), which allows us to assess if the data follows some theoretical distribution. Some models, such as GLM, assume that the residuals are normally distributed. To visually assess spatial correlation, scale location plots are used. These plots show if residuals are spread equally along the ranges of predictors. Strong residual geographic patterning usually indicates that key variables are missing, the model is not adequate or geographic factors are influential, such as predators. Residual leverage plots are used to assess influential data points, i.e. points whose inclusion or exclusion produce different results on the model. Since the model´s purpose it to generalize the data, this situation is not desirable. Finally, partial dependence plots allow the visualization of the relationship between the occurrences and a environmental variable x_i, while accounting for the effect of the other variables $x_j \in X\backslash\{x_i\}$. The effect of the variables x_j can be accounted in various ways: each variable x_j can be considered as their mean, mode or other relevant statistic or we can combine each value of x_i with all the values of X.

For this last method if we have 10 observations and want the partial dependence plot for the variable x_1, we need to:

- For $n \in 1, 2, ..., 10$:
  - Copy the n observations and replace the n values of x_1 with the constant $x_1(n)$.
  - Compute the predicted values for the new data.
  - Compute the average prediction to obtain $f_n(x_1)$.

where $x_1(n)$ is the value of x_1 in observation n. The partial dependence plot for the variable x_1 corresponds to the plot of the pairs $x_1, f_n(x_1)$.

A way to measure how much unexplained variance there is in our model is by using deviance. It is similar at looking at the total of the residuals. Since its value is hard to interpret because it depends on sample size and on the number of variables used by the model, a common use of this measure is to compare models. We can consider a model with all the environmental variables (saturated model) and that therefore explains the data on its whole, and compare it with a model with less environmental variables (simpler model) that generalizes better. It measures if the reduction on deviance by adding or removing variables is significant. The difference between the deviance of the more complex and the simpler model has a chi-square distribution so its significance can be assessed by analyzing the p-value of the chi-square test with degrees of freedom equal to the difference of the number of variables of the two models.

Chapter 3

The DeepData tool

This chapter starts by explaining DeepData's architecture in section 3.1, which covers the database structure. The chapter continues explaining DeepData's implementation in section 3.2, which covers the interface interaction with the database and the user interaction with the interface.

3.1 Tool architecture

DeepData's architecture is describe in Magda Resende's thesis [4], with some changes which are explained in this section.

3.1.1 Relational database

An extensive number of environmental datasets are available nowadays for fitting SDMs, such as:

- World Ocean Atlas 2013 (WOA13 [44]), which provides a set of objectively analyzed climatological fields (temperature, salinity, dissolved oxygen, apparent oxygen utilization, percent oxygen saturation, phosphate, silicate and nitrate) at standard depth levels for annual periods for the world ocean.
- European Marine Observation and Data Network (EMODnet [45]), which provides data across seven discipline-based themes: bathymetry, geology, seabed habitats, chemistry, biology, physics and human activities.
- Ocean Biogeographic Information System (OBIS [46]), which provides data about the species occurrences.
- World Register of Marine Species (WoRMS [47]), which provides information about the taxonomy of the species.

Data from these datasets, concerning the geographic space of the Azores EZZ, is already included in DeepData as the environmental variables are commonly used to model species distribution models and as a way to allow the user to experiment DeepData without having to insert data. Concerning WoRMS

Figure 3.1: Relational scheme of database.

dataset, it is not used to model the species distribution model but to give information about the taxonomy of the specie that is being modelled.

The existing database is composed of the following tables:

- `Condicoes_ambientais_quarto_grau`, which stores the environmental variables (temperature, salinity, silicate, nitrate, phosphate, density, conductivity, dissolved oxygen, apparent oxygen saturation and apparent oxygen utilization) associated to its `latitude`, `longitude`, `depth`, `source` and `decade`.
- `Ocorre_quarto_grau`, which stores information about the species' occurrences. It associates the `species` from table `Taxonomia_Espécie` with its `latitude`, `longitude`, `depth`, `source`, `decade`, `year` and `month`.
- `Taxonomia_Reino`, which stores the species' `kingdom`.
- `Taxonomia_Filo`, which associated the species `kingdom` with its `phylum`.
- `Taxonomia_Classe`, which associated the species `phylum` with its `class`.
- `Taxonomia_Ordem`, which associated the species `class` with its `order`.
- `Taxonomia_Famímilia`, which associated the species `order` with its `family`.
- `Taxonomia_Género`, which associated the species `family` with its `genus`.
- `Taxonomia_Espécie`, which associated the species `genus` with its `species`.

To the table `Condicoes_ambientais_quarto_grau`, shown in figure 3.1, were added the primary keys `resx` and `resy`, meaning the resolution of the latitude and longitude, respectively. This change was made

to prevent the situation when data of a variable is uploaded with different resolutions and have the same coordinates. For example, if a variable is uploaded with a resolution of $1°$ and then with a resolution of $0.5°$, both starting at the same coordinates, e.g. $(-34°, 32°)$, then all the coordinates of the first upload belong to the second upload as well. If there were no `resx` and `resy` keys, the values of the first upload would all be replaced with the values of the second upload leading to information loss.

Changes to the table `Ocorre_quarto_grau`, shown in figure 3.1, were also made. The primary keys `latitude`, `longitude`, `depth` and `source` no longer refer to the primary keys `latitude`, `longitude`, `depth` and `source` from the table `Condicoes_ambientais_quarto_grau`. This change allows for the existence of species data that do not necessarily match the environmental data. For example, if we have environmental data with a resolution of $1°$ and starting at latitude $32°$ and longitude $-34°$, then there is one cell with latitude between $32°$ and $33°$ and longitude between $-34°$ and $-33°$. Therefore, with the previous implementation the species data would have to have latitude $32°$ and longitude $-34°$ for that specific cell. This new implementation allows for the species' latitude between the range of $32°$ and $33°$ and species' longitude between the range of $-34°$ and $-33°$ to correspond to that one cell.

A new table `Indice` was created, to make `DeepData` load faster. Instead of having to search the whole `Condicoes_ambientais_quarto_grau` table to be aware of the existing variables and the resolutions and decades available for each variable, now all the information is stored in this much smaller table, making the search much faster.

Changes to the insertion of new data were also implemented, which are explained later in subsection 3.2.1.

3.2 Tool implementation

For the implementation of the species distribution models, the software used was `R`. Although `python` is also commonly used for machine learning, most of the published work on SDMs used the `R` software [48–50]. Ecologists tend to use `R` while computer engineers tend to use `python`, because `R` has been established for a long time and includes a broader range of methods employed in ecological analysis as well as numerous routines for data exploration [51]. Although `python` has also been improving in these fields, `R` fulfills better the Ecologists needs. `Python` has the advantage that it is better for deployment, and therefore it is used to implement other parts of the application, including fetching the data from the database needed for computing the SDM.

3.2.1 Data insertion

`DeepData` allows both the insertion of species data and environmental data.

Species data insertion

`DeepData` allows the insertion of species data threw a csv file, with the following structure: `species' name, latitude, longitude, month collected, year collected, depth, data source name`. Each

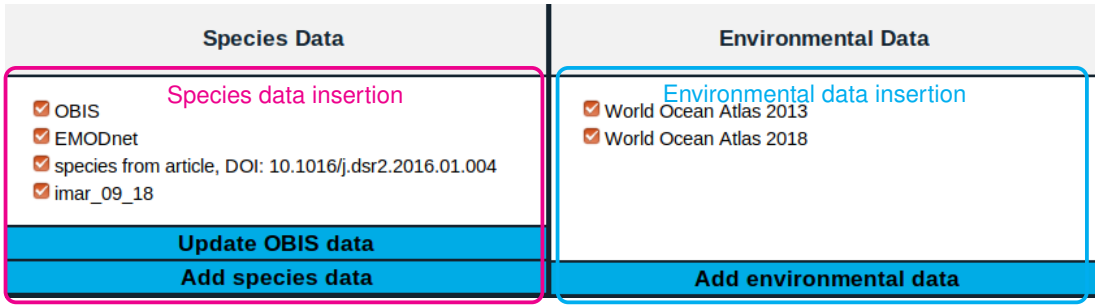


Figure 3.2: DeepData's data insertion interface.

entry in the csv file is then associated with a decade, so that the information in the table Ocorre_quarto_grau is associated with the information in the table Condicoes_ambientais_quarto_grau. Also, every new species inserted in table Ocorre_quarto_grau is also inserted in table Taxonomia_Especie so that the information in both tables is also associated. In the past, the insertion was done once at a time with INSERT or UPDATE statements. Now, it is used LOAD DATA LOCAL INFILE which is a highly optimized mysql statement that directly inserts data into the Ocorre_quarto_grau table from a csv file. Figure A.2 of appendix A represents a flowchart of this process.

As shown in figure 3.2, highlighted in magenta, DeepData allows the user to decide whether the user wants to use the default species data.

Environmental data insertion

The user can upload csv files, but also ascii files. Ascii files are composed of a header that defines the properties of the file, such as:

- Number of cell columns, represented by ncols.
- Number of cell rows, represented by nrows.
- Longitude coordinate of the origin, by center or lower left corner of the cell. Represented by xllcenter or xllcorner, respectively.
- Latitude coordinate of the origin, by center or lower left corner of the cell. Represented by yllcenter or yllcorner, respectively.
- Cell size, which might have two values, if the cell sizes for the longitude and latitude are different. Represented by cellsize.
- The values that mean no data, represented by nodata_value.

The header is followed by cell value information, separated by a space character.

Csv files requires that the file has the following structure: latitude, longitude, depth, decade, environmental variable value, data source name. Ascii files require the user to insert the depth and the data source name associated to the file and select the decade. With this information, the ascii file can be converted to a csv file.

For both files, the user has to specify whether the variable already exists. In which case, the user wishes to upload more data. Otherwise, if it is a new variable, in which case the name that will be used is the name of the file without the extension. Without the changes to the database explained in subsection 3.1.1, standardization of the latitude and longitude was needed as a way of imposing the same structure to all variables, meaning that latitude had the interval of 32.125 to 43.875 with a resolution of $0.25°$, while longitude had the interval of -34.125 to -21.875 with a resolution of $0.25°$. Suppose this standardization was not done, and that for some variable there was data corresponding to a resolution of $0.25°$. Now suppose the user inserts new data with a resolution of $0.35°$ starting in the same coordinates. If no standardization was done, what would happen is that the database would have data corresponding to a latitude of: 32.125 (corresponding to the starting point), 32.375 (corresponding to the $0.25°$ resolution) and 32.475 (corresponding to the $0.35°$ resolution). This would generate an error in the pre-processing phase, when trying to create a grid of data, since the cell sizes for the latitude would be different. The first cell would have a resolution of $0.25°$ and the second a resolution of $0.10°$.

With the implemented changes, we can differentiate the two different structures, joining them in one structure in the pre-processing phase (see section 3.2.3). The insertion was previously done once at a time with `INSERT` or `UPDATE` statements while now it is used `LOAD DATA LOCAL INFILE` which is a highly optimized mysql statement that directly inserts data into the `Condicoes_ambientais_quarto_grau` table from a csv file. If the variable is new, or the decade or resolution is new for an existing variable, then this information is inserted in to `Indice` table. Figure A.3 of appendix A represents a flowchart of this process.

As shown in the figure 3.2, highlighted in cyan, `DeepData` allows the user to decide if the user wants to use the default environmental data.

3.2.2 Input selection

`DeepData` allows the user to select the inputs of the main categories:

- Species,
- Environmental variables,
- Model parameters,
- Pre-processing parameters,
- Evaluation parameters.

Species selection

The species can be selected through the taxonomy hierarchy, or by directly selecting the species name. The user can also choose either to generate pseudo-absences or not. Figure 3.3 shows this part of the interface highlighted in red. Figure A.4 of appendix A represents a flowchart of this process.

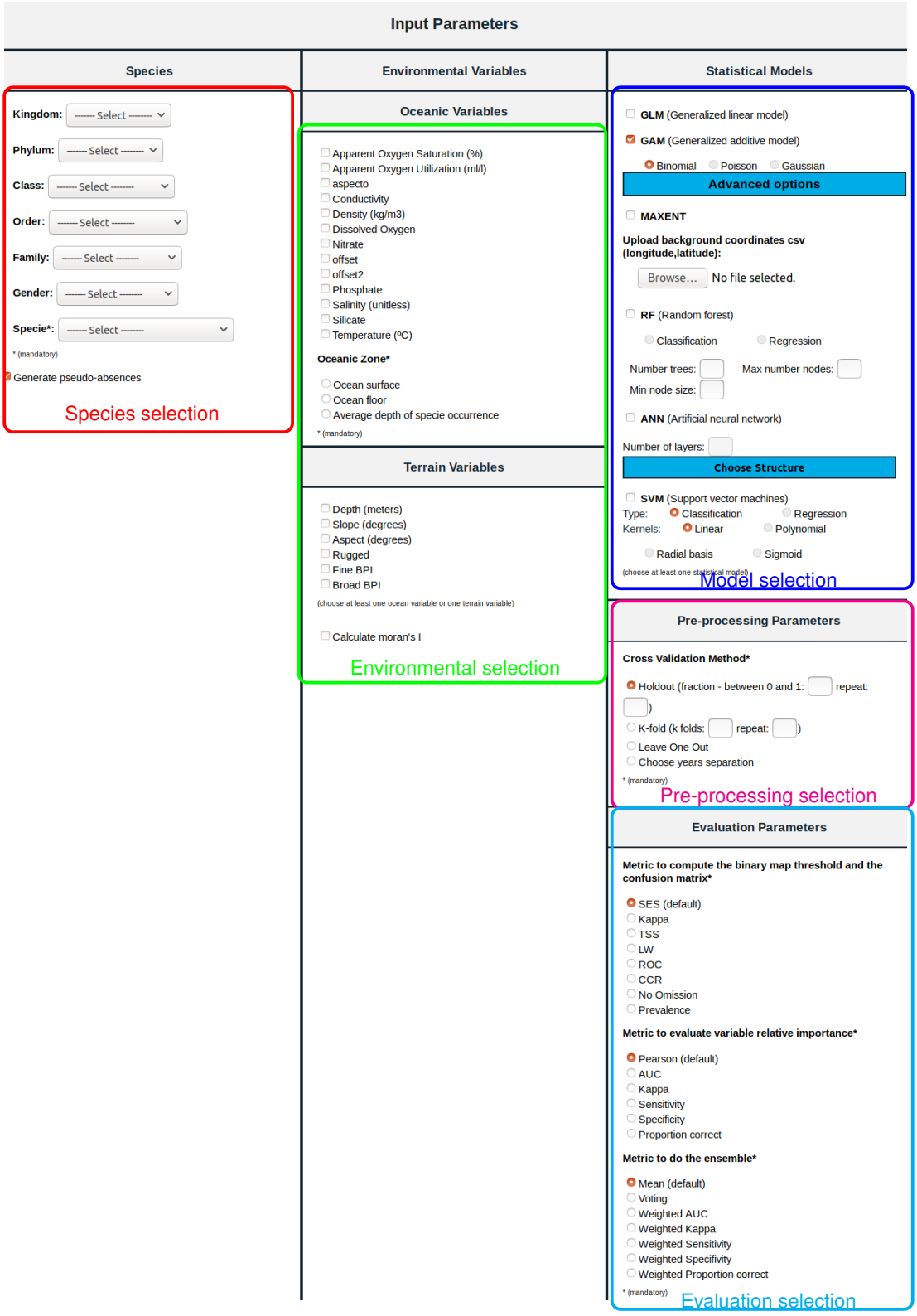


Figure 3.3: `DeepData`'s input interface.

Table 3.1: R packages used for each implemented model.

Model	R package
Generalized addictive model	mgcv
Generalized linear model	SSDM
MaxEnt	dismo
Random Forest	randomForest
Artificial neural network	neuralnet
Support vector machine	e1071

Environmental variables selection

For the environmental variables, the user can select oceanic variables, which can have different resolutions, and terrain variables. DeepData also allows the selection of various oceanic and terrain variables, and the definition of the oceanic zone. The oceanic zone can be:

- Ocean surface, meaning that only 0 to 5 meters of depth is considered.
- Ocean floor, meaning that only 5500 to 5400 meters of depth is considered.
- Average depth of the species occurrence, meaning depth values are prioritized by number of occurrences of the species. This feature is further explained in sub-section 3.2.3.

The interval of the ocean floor is much larger than the interval of the ocean surface, since spatial variation in environmental variables decreases with depth [52]. There is also the possibility to calculate the Morans'I correlation coefficient. Figure 3.3 shows this part of the interface highlighted in green. Figure A.5 of appendix A represents a flowchart of this process.

Model parameters selection

DeepData allows the specification of the model parameters. Figure 3.3 highlights the model parameters selection in blue. Each model is implemented with a specific R package, shown in table 3.1.

In order to compute generalized addictive models, DeepData allows the specification of the family of the distribution, which can be:

- Binomial, in the case of presence-absence data, which is used as default.
- Poisson, in the case of count data.
- Gaussian, in the case of count data with a normal distribution.

It also allows the specification of the link function, in accordance with the family selected as shown in table 3.2. Smoothness of fit of each variable can also be controlled, differing on the basis used to represent the smooth function. Possible splines are:

- Thin plate spline, which is used as default.
- Duchon spline.
- Cubic spline.

Table 3.2: Relation between the family distribution and the link function.

family distribution	link function
binomial	logit probit cauchit log cloglog
possion	log identity sqrt
gaussian	log identity inverse

Table 3.3: Kernel functions, where u and v, represent inputs from the feature space.

Kernel	Function
Linear	$u'v$
Polynomial	$(\gamma u'v + coef)^{degree}$
Radial basis	$\exp(-\gamma\|u - v\|^2)$
Sigmoid	$\tanh(\gamma u'v + coef)$

- Spline on the sphere.
- P-splines.
- Random effects.
- No smoothing.
- Offset.

For computing `MaxEnt`, `DeepData` allows uploading a background file, which must be composed of the latitude and longitude. If no file is given, `DeepData` randomly selects 10000 points of the coordinate space to be used as background.

For computing random forests, `DeepData` allows tuning the following parameters:

- Number of trees to grow. This should not be set to a number too small, to ensure that every input row gets predicted at least a few times. Default is set to 500.
- Minimum size of the terminal nodes. Setting this parameter to a large number causes smaller trees to be grown (and thus take less time). The default values are 1 for classification and 5 for regression.
- Maximum number of terminal nodes that the trees can have. If not given, trees are grown as much as possible (subject to limits by node size).

For computing neural networks, the structure can be defined by first indicating the number of layers, which corresponds to the sum of the input layer, hidden layers and output layers. Afterwards, the number of perceptrons for each layer is defined. Note that neural networks do not have any default structure.

For computing support vector machines, the kernel can be defined as:

- Linear, which is the default value,
- Polynomial,
- Radial basis,

- Sigmoid.

The corresponding function is defined in table 3.3.

Figure A.6 of appendix A represents a flowchart of this process.

Pre-processing parameters selection

For the modelling phase, the application allows cross-validation to be performed. `DeepData` allows the user to select one of the following methods:

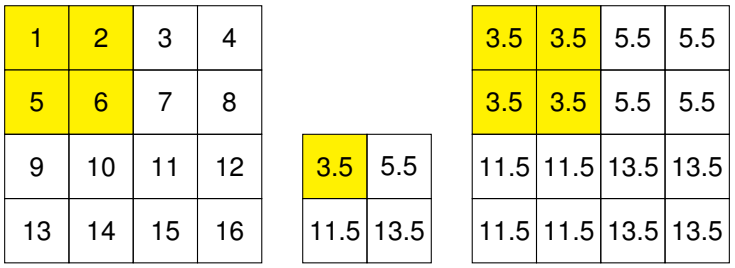
- Holdout, which separates the dataset into train set and test set according to the fraction, being the train set larger than the test set. Training is performed as many times as there are test partitions.
- Leave one out, which separates the data in three sections and at each repetition uses two for training and one for testing.
- K-fold, which separates the data in k folds and trains the model over the k number of combinations.
- Years separation, which separates the data according to the years selected for train and test, with the constraint that each year can only be either train or test.

Figure 3.3 shows these parameters highlighted in magenta. Figure A.7 of appendix A represents a flowchart of this process.

Evaluation parameters selection

Regarding the model evaluation, the user has to select the metric to compute the binary map threshold and the confusion matrix. `DeepData` allows this threshold to be defined by:

- SES, which is the threshold value or range in values that maximizes sensitivity equal to specificity.
- Kappa, which is the threshold value or range in values with the maximum Kappa statistic.
- TSS, which is the threshold value or range in values that maximizes sensitivity plus specificity.
- LW, which is the minimum prediction probability for the occurrence (presence) records.
- ROC, which is the threshold value or range in values where the ROC curve is closest to point (0,1).
- CCR, which is the threshold value or range in values with the maximum number of presence and absence records correctly identified.
- No omission, which is the threshold value or range in values with no omission error, meaning no false positives (predicting absences incorrectly).
- Prevalence, which is the threshold value or range in values with the modeled prevalence closest to the observed prevalence.



(a) original (b) aggregated by factor 2 (c) disaggregated by factor 2

Figure 3.4: Example of variable aggregation and disaggregation.

While the first three metrics (SES, Kappa and TSS) can be applied to all models, the last metrics (No Omission and Prevalence) can only be applied to `MaxEnt`. The remaining metrics (LW, ROC and CRR) can be applied to all models except `MaxEnt`, i.e. `GAM`, `GLM`, `RF`, `ANN` and `SVM`.

If more than one model is selected, then an ensemble model is computed additionally to the models selected. This ensemble can be done by:

- Averaging,
- Voting,
- Weighted AUC,
- Weighted Kappa,
- Weighted Sensitivity,
- Weighted Specificity,
- Weighted Proportion correct.

Figure 3.3 shows these parameters highlighted in cyan. Figure A.8 of appendix A represents a flowchart of this process.

3.2.3 Model implementation

Before starting with the model construction, `DeepData` verifies whether all inputs required were selected or inserted by the user. Regarding the model parameters, all models have default options except the `ANN` which requires the user to define the structure of the `ANN`. Figure A.9 of appendix A represents a flowchart of this process.

Model preparation

The first step is to load the species and the environmental variables. Regarding the environmental variables, when `average depth of the species occurence` is selected, `DeepData` used to consider a fixed depth to load the variables, corresponding to the mean depth of the species occurrences. This is

a fair assumption if we consider a place where bathymetry does not change much. By analyzing the species' occurrence data, we conclude that species might occur at every depth, making mean not the best metric. So a ranking of depth by the frequency of the species is used. If the species appears 40 times at depth 200, 30 times at depth 350 and 10 times ate depth 20, we first load all variable values at depth 200. For the cells where data at this depth is not available, we load the values for a depth of 350 and finally for the remaining cells the depth of 20.

The next step is to unify the different variables. When the variables have different resolutions, `DeepData` uses the lowest resolution as the standard resolution, i.e. the finer resolution. As shown in figure 3.4, the aggregation of the original variable by a factor 2, meaning each cell covers $1/4$ of the area, is equal to the mean of the four original cell values. On the left of figure 3.4 are represented, in yellow, four cells, whose aggregation will correspond to its mean, represented, with yellow, in the middle figure. The figure 3.4 also shows the disaggregation of the aggregated variable by a factor of 2, as an attempt to reproduce the original variable. So in the right figure are represented, in yellow, four cells, corresponding to the disaggregation of the yellow cell in the middle figure. As shown, the disaggregation corresponds to the expansion of the cell to the wanted extent. Now each cell covers 4 times the area. Therefore `DeepData` uses the lower resolution, because aggregating leads to information loss while disaggregating has no effect on the amount of information.

When the variables have different extents, the overlaying extent is used. All variables are cropped to this new extent. It is assumed that all variables are defined on the CRS called EPSG:4326, which corresponds to:

$$+proj = longlat \quad +datum = WSG84 \quad +ellps = WGS84 \tag{3.1}$$

Before the modeling phase, some feedback on the data used is given to the user. If the user selects the option `Calculate moran's I`, moran's I is calculated for each variable. While usually moran's I considers as weight the inverse of the distance between two cells, meaning the further they are the less weight to the measurement they have, in this implementation only the neighbors of a cell are considered. This simplification was introduced as a way of overcoming problems of memory and time with variables with small resolutions. Memory problems are solved because there is no need to have a matrix with all weights, contrary to usual implementations. Time problems are solved because the computation considers less cells, and therefore makes less calculations.

Collinearity between the variable is verified through the `usdm` package. For variance inflation factors larger than 3, which means that the standard error is 1.7 times larger than if the variables were not correlated, the script stops and a pop-up appears asking the user whether he/she wants to remove any variable or not. If the user chooses to maintain all the variables then the collinearity is not verified again, while if the user chooses to remove some, then the collinearity is verified for the remaining.

If the user selects the option `Generate pseudo-absences`, shown in figure 3.3, the `mopa` package is used to generate the pseudo-absences. The first step is to classify the background, using the environmental variables of the presence localities, as suitable or unsuitable. Then the pseudo-absences are generated at random, having the same proportion of presences, meaning prevalence of 0.5, and a

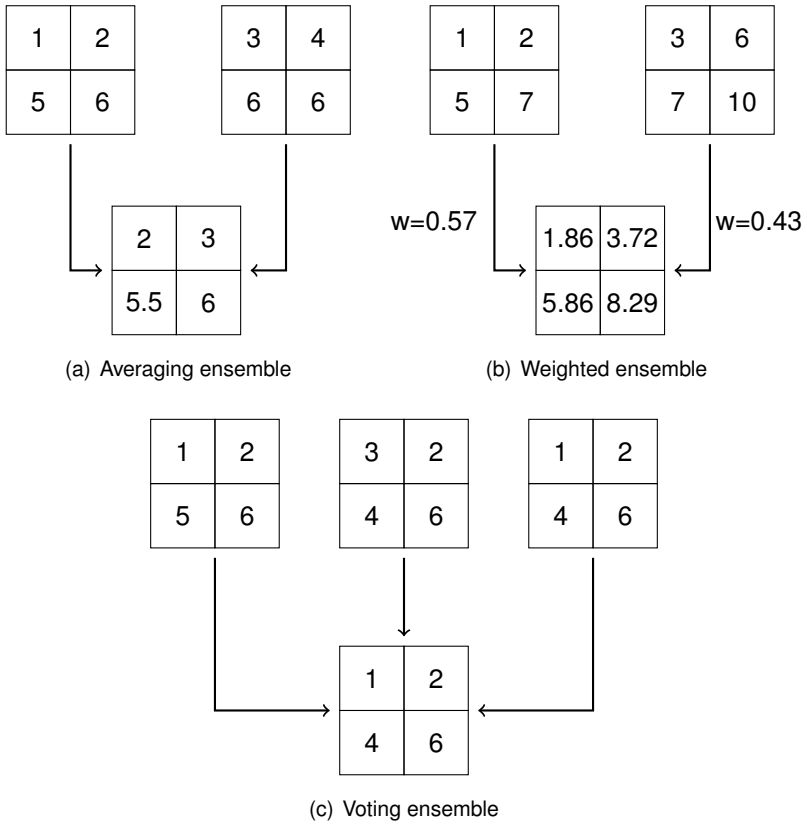


Figure 3.5: Methods of ensemble models.

minimum distance to the presences of 30km. Figure A.10 of appendix A represents a flowchart of this process.

Model execution

The first step of model training consists of dividing the data, composing sets of data, according to the cross-validation method. Afterwards, each selected model is trained and tested on each set. If more than one model is selected, then an ensemble model is constructed according to the parameters selected. With averaging ensemble, an average for each cell is computed over all models, as shown in figure 3.5(a). With voting ensemble, the majority of agreement for each cells over all models is considered, as shown on figure 3.5(c). With weighted ensemble, we differentiate the contribution of each model according to AUC, kappa, sensitivity, specificity and proportion correct measurements. These measurements are taken from the testing of each model. The weight of each model is calculated by:

$$w_i = \frac{metric_i}{\sum_i^N metric_i} \tag{3.2}$$

Where w_i represents the weight of model i, $metric_i$ represents the value of the metric of the model i and N represents the total of models. Take the case where we want to ensemble two models according to their AUC. Consider that the first model has 0.8 AUC and the second one has 0.6 AUC. Then the first model has weight 0.57 and the second has weight 0.43, as shown in figure 3.5(b).

For each model, DeepData estimates the mean AUC and its standard deviation as well as the mean threshold and its standard deviation, from all the modelling repetitions, defined by the cross validation method. DeepData presents the threshold, accuracy, omission rate, sensitivity, specificity, proportion of correctly predicted occurrences and kappa statistic of the best model. It also return the calculated VIF of each variable and the number of presences used.

DeepData also returns a zip file, with model evaluation plots, specific to each model. For the GAM model, DeepData returns a zip file containing:

- Residuals.png, which is composed of: normal QQplot, residuals vs linear predictors, histogram of residuals and response vs fitted values.
- Patial_dependence_plots.png, which plots the component smooth functions of the model, in the scale of the linear predictor.
- Gam.png, which plots the predicted occurrence values over the environmental space.
- Gam_uncertainty.png, which plots the standard error estimates returned for each prediction over the environmental space.
- Akaike's Information Criterion, which is not on the zip file, but on the evaluation results.txt

For the RF model, DeepData returns a zip file containing:

- Variable_importance.png, when more than one variable is used to do the modelling. Plots each variable importance according to the mean decrease in accuracy and the mean decrease in node purity.
- Effect_variable.png, which plots the marginal effect of a variable on the predicted occurrence.
- RF.png, which plots the predicted occurrence values over the environmental space.

For the SVM model, DeepData returns a zip file containing:

- SVM.png, which plots the predicted occurrence values over the environmental space.

For the ANN model, DeepData returns a zip file containing:

- ANN.png, which plots the predicted occurrence values over the environmental space.

For the MaxEnt model, DeepData returns a zip file containing:

- Species_omission.png, which shows how testing and training omission and predicted area vary with the choice of cumulative threshold.
- Species_roc.png, which plots the ROC curve.
- Species_(number of repetition)_(name of the variable).png, which plots the response curves of a variable for each repetition.

- `Species_(number of repetition)_(name of the variable)_only.png`, which plots the response curve corresponding to a model that only uses the variable, disregarding other variables.
- `Maxent.html`, which opens an html page with all the above plots and explanation. Also, information about the statistical significance of the prediction and analysis of variable contribution is provided.
- `MAXENT.png`, which plots the predicted occurrence values over the environmental space.

Regarding the ensemble model, `DeepData` creates the `Rplots.png` which plots the predicted occurrence values of the ensemble method over the environmental space. Finally, a file with a plot with the used presences, called the `Pplots.png`, is also generated.

Figure A.11 of appendix A represents a flowchart of this process.

Chapter 4

Results

The purpose of `DeepData` is to facilitate the work of biologists, who have the domain knowledge but that might not be used to programming. Therefore, to show the usefulness of the developed tool, we selected two papers, whose data we have access to, and tried to obtain the same results. Two papers were examined, the first (see section 4.1.1) regarding the use of `MaxEnt`, and the second (see section 4.2) regarding the use of `Random Forest` and `Generalized addictive model`. We start by giving an overview of each paper and continue describing the steps needed to produce the results with `DeepData`. Each section starts first by characterizing the problem and then by giving a description of the steps used to solve the problem.

4.1 First case study

4.1.1 Problem description

The first paper selected is entitled ”Habitat modelling of crabeater seals (Lobodon carcinophaga) in the Weddell Sea using the multivariate approach Maxent” [53], which uses `MaxEnt` to identify suitable habitat conditions to the crabeater seal.

Although crabeater seals are the most abundant Antarctic species, they inhabit the hardly accessible Antarctic pack ice zone, specially the Weddell Sea. Consequently, abundance estimates are hard to obtain.

During this study period, the sea ice cover of the Weddell Sea was exceptionally low. The impact of this adversity on crabeater seals is not clear, but it could mean a reduction of their habitat. This is becoming more important since the sea ice cover in the Southern Ocean is predicted to decrease.

This study uses MaxEnt to model the influence of certain environmental variables on the distribution of crabeater seals in the Weddell Sea. The purpose is to identify the suitable habitat to the crabeater seals, within the Weddell Sea.

Regarding species occurrence data, fifteen crabeater seals of both sexes and different age classes were equipped with satellite-linked dive recorders (SDRs) between January 28 and February 6, 1998. Regarding enviromental variables, a set of 13 environmental variables was used to analyze the habitat

preferences of crabeater seals: sea ice concentration, sea ice thickness, sea ice freezing rate, water surface and bottom temperature, surface and bottom salinity, surface and bottom zonal current velocity, surface and bottom meridional current velocity, slope, and distance to shelf break. Sea ice thickness, sea ice freezing rate, temperature, salinity and current velocity data were derived from the Finite Element Sea ice-Ocean Model (FESOM) with a resolution of 5 km × 5 km. Sea ice concentration was recorded by the Special Sensor Microwave/Imager (SSM/I) of the Defence Meteorological Satellite Program (DMSP) at the National Snow and Ice Data Center (NSIDC) with a resolution of 25 km × 25 km. These values ranged from 0 % (open water) to 100 % (closed ice cover). A map on slope was derived from the International Bathymetric Chart of the Southern Ocean with a resolution of 0.5 km × 0.5 km. Distance to shelf break (1000 m isobath) was calculated using the `Near` tool in `ArcGIS`. All variables, except slope and distance to shelf break, were considered monthly, meaning that each month has the mean values of the variable on that period.

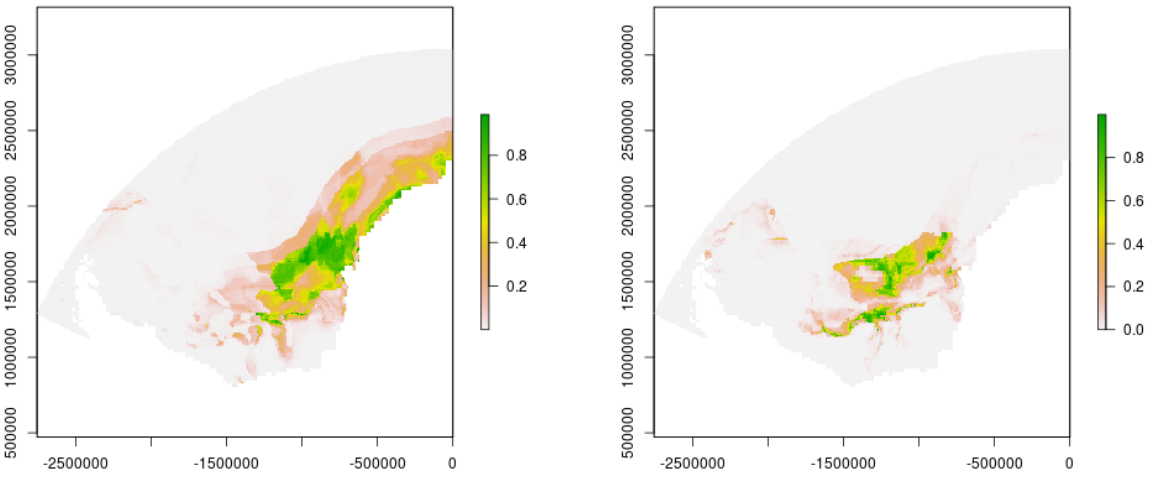
All seal location were assigned to the environmental variables according to the sampling months. Then, the values from all environmental raster files were resampled by using the `Fishnet` tool in `ArcGIS` with a resolution of 5 km × 5 km, corresponding to the grid size of the FESOM raster, which contributed most of the variables (10 of 13). Additionally, a 5 km × 5 km resolution is a suitable determination for a seal's position. Thus, this new resolution allowed not only for a better reconciliation between seal locations and environmental parameters but also accounted for spatial autocorrelation of the tracking data, which mostly disappeared at a distance of greater than 5 km. Prior to model building, the seal location data were subsampled to diminish potential biases. Therefore, only location data from February, March and April 1998 were used for modelling. All data can be found in pangaea [54].

4.1.2 Tool testing

Both species occurrence and environmental data for each month were loaded into `DeepData`. To recreate the paper, the tool configurations were:

- Use `MaxEnt`, with the background file loaded.
- Do not generate pseudo absences, since `MaxEnt` only uses presences.
- Use average depth of species occurrence.
- Do not calculate moran's I.
- Use holdout with fraction of 80% and repeat 20.

This model was used with all variables loaded to verify the influence of each environmental variable contributing to the model by a measure called permutation importance and to identify the variables that mattered most concerning the seal distribution. Jackknife test was also used to analyze the relative importance of each variable. Detailed results are presented in appendix B.1. As shown in table 4.1, from the 13 variables, slope, bottom zonal current velocity and bottom meridional current velocity did not contribute more than 5 % to neither monthly model and therefore they were removed from the final



(a) February

(b) March

(c) April

Figure 4.1: Probability of presence of crabeater seals for each month.

model. Besides, we can also see that distance to shelf break and sea ice concentration are important variables for determining crabeater seal distribution throughout all three months. Additional variables with moderate overall importance to the models were velocity meridional surface (February) and velocity zonal surface (March) as well as water temperature surface (April).

Slight differences in values are due to the fact that the original paper used the `samples with data` method. This method uses a `csv` file that has both the coordinates of the species and the values of the environmental variables for that coordinates. This makes the step of getting the environmental values from each `ascii` file unnecessary. Therefore, the `samples with data` method allows one to have different environmental values for coordinates that fit in the same grid of the `ascii`. The problem is that the `dismo` package does not allow this interaction, and so coordinates that fit in the same grid have the same environmental values.

Regarding the models evaluation, AUC values are high, showing that the predictions are far from random. On the other hand, standard deviations are low, meaning there is a high degree of uniformity between the repetitions, as shown in table 4.2. The models predictions are shown in figure 4.1, and the response curves are presented in appendix B.2.

Through the response curves and considering distance to shelf break, the model predictions of February and March both revealed that crabeater seals generally prefer a range of 400 km off the continental shelf break due to high probabilities of presence. In April, the probability of presence increases with increasing distance. Considering the sea ice concentrations, the model predictions of February are high for sea ice concentrations of 0 % and slightly less for concentrations between 0 and 50 %. While for March, the probability is high for values of sea ice concentration between 20% and 80 %. Finally, for April species tend to prefer sea ice concentrations around 90%.

Table 4.1: Variable permutation importance in February, March and April.

Variable	February	March	April
Distance to shelf break	41.6	20.1	42.8
Sea ice freezing rate	0.8	0.3	4.2
Sea ice thickness	0.9	0.3	4.6
Sea ice concentration	11.7	35.3	7.7
Salinity bottom	1	6.2	1
Salinity surface	1.4	7.7	3.2
Water temperature bottom	8	3.3	1
Water temperature surface	8.3	10	17.4
Velocity meridional bottom	1.3	0.4	0.5
Velocity meridional surface	22.8	3	11.6
Velocity zonal bottom	0.1	0.7	1.4
Velocity zonal surface	1.4	11.8	4.4
Slope	0.6	1.1	0.2

Table 4.2: Monthly model AUC and standard deviation.

Month	AUC	training/test points
February	0.93 ± 0.005	631/157
March	0.96 ± 0.006	385/96
April	0.94 ± 0.005	201/50

4.2 Second case study

4.2.1 Problem description

The second paper selected is entitled ”Population Estimates of Trindade Petrel (Pterodroma arminjoniana) by Ensemble Nesting Habitat Modelling” [50], which uses the ensemble model to identify suitable habitat conditions to the Trindade Petrel, which is an endangered gadfly petrel breeding off a Brazil oceanic island.

SDMs applied to seabird studies have been mostly used to predict at-sea distribution, while colony prediction has not been fully explored. Other methods have been helpful in estimating occurrence in inaccessible seabird species nesting. This study focuses on the use of SDMs to model nest distribution data as a way to identify the habitat suitability for seabird colonies nesting in non reachable areas.

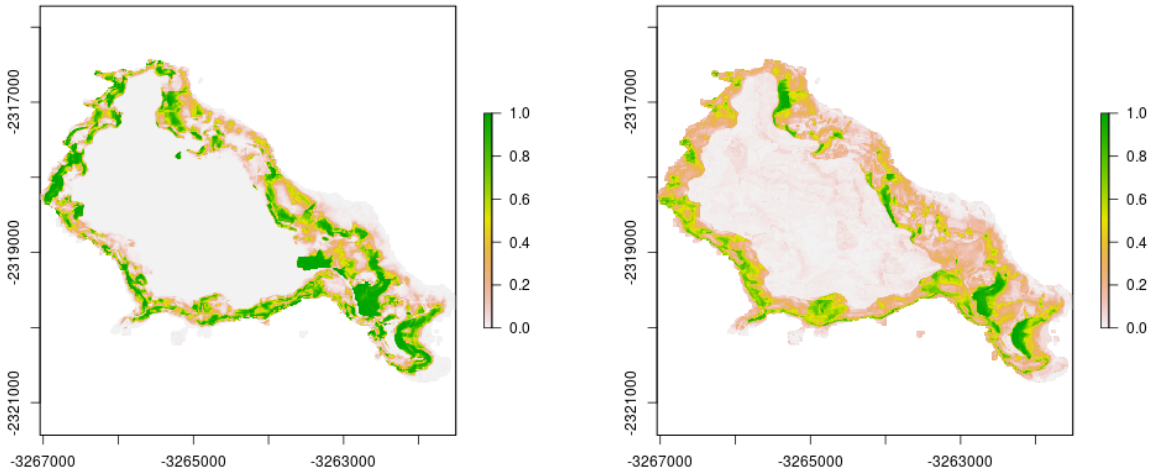
Contrary to the few studies that use this method, in this study an ensemble method is used taking into account that different models can produce different distributions.

Regarding occurrence data, 411 nests were identified between 2000 and 2007 and, afterwards, between September and November of 2014. Regarding environmental variables, a set of 5 environmental variables was used: elevation, slope, flow length, aspect and insulation. The variables were generated from an elevation shapefile of Trindade Island, with a resolution of 17 m × 17 m. Flow length measures the downstream distance along the flow path for each cell. Insulation measures the amount of sun light estimated to reach the surface, which results from the topographical features.

This paper uses the `biomod2` package to create the distribution model. To create the ensemble model, it starts by testing which models best fit the data, so that the best three models are used by the ensemble model. The tested models are: `GAM`, `GLM`, `Multiple adaptive regression splines`, `RF`, `Generalized boosted model`, `ANN`, `MaxEnt Phillips` and `MaxEnt Tsuruoka`. The difference between the two variants of `MaxEnt` is the package that implements each of them. While `MaxEnt Tsuruoka` only uses an `R` package, `MaxEnt Phillips` uses a java software which is called within an `R` package.

4.2.2 Tool testing

`DeepData` allows the creation of these models: `GAM`, `GLM`, `RF`, `ANN` and `MaxEnt Phillips`. Regarding the `ANN` model, `DeepData` uses `neuralnet` package, while `biomod2` implements the `nnet` package. The main difference between the two packages is that `nnet` only allows single-hidden-layer neural network, while `neuralnet` allows for any possible combination. Besides, `nnet` allows the hidden layer not to be defined, in which case the use of a hidden layer with 2, 4, 6 or 8 nodes are tested, and the combination with the best AUC is selected. In the present problem, with the package implemented on `DeepData` the best



(a) GAM (b) RF

Figure 4.2: Spatial distribution of Trindade Petrel for each model.

combination was a hidden layer with 8 nodes, and therefore is the one considered.

Both species occurrence and environmental data were loaded into `DeepData`. To recreate the experiments in the paper, the tool configurations were:

Do this 20 times:

- Use `GLM`.
- Use `GAM`, with binomial family.
- Use `MaxEnt`, with default values.
- Use `RF`, with classification and min node size of 5.
- Use `ANN`, with 1 layer with 8 nodes.
- Generate pseudo absences, since only `MaxEnt` uses presences.
- Use average depth of species occurrence.
- Do not to calculate moran's I.
- Use holdout with fraction of 80% and repeat 3.

`DeepData` only considers AUC from the evaluation metrics used on the paper. On table 4.3 are presented the results obtained. As we can see, in contrast to the original paper, `MaxEnt` has a high AUC value. This is due to the fact that in our tool `MaxeEnt` uses the default 10000 background points, while in the paper are used 411 background points. Besides this, all results are as expected, with both `RF` and `GAM` with the best results.

Since the ensemble model considers one more model that `DeepData` cannot produce, meaning that the final distribution cannot be achieved. Instead, in figure 4.2, are presented the individual distributions

Table 4.3: Models AUC and standard deviation.

Model	AUC
`RF`	0.95 ± 0.021
`GAM`	0.95 ± 0.031
`MaxEnt`	0.94 ± 0.023
`GLM`	0.77 ± 0.032
`ANN`	0.79 ± 0.110

for each model. `DeepData` also returns the partial dependence plots for each model, which are presented in appendix B.3. As noted on section 2.4, there is not only one way of calculating these plots and therefore caution is needed when comparing different plots.

Similarly, the response curves presented on the paper use the evaluation strip [55]. The method consists in generating a dataset for each variable by varying the values of that variable over its range, while keeping the other variables constant. The response curve corresponds to plotting the predicted occurrence for each dataset, showing how the model responds to the variation of each variable.

All things considered, we can say that the overall tendency for the models are the same. We see that Trindade Petrel prefer elevations lower than 250 m, in areas with slope above 40° of inclination and a flow length smaller than 10 m. They prefer terrains usually facing northerly direction with intermediate sun incidence.

Chapter 5

Conclusions

This chapter starts by first giving an overview of the accomplished work in section 5.1. 5.2 then proposes work that could be done in the future.

5.1 Achievements

This thesis presents a `DeepData` tool that allows a simple and efficient way of creating species distribution models, conserving the user domain knowledge and allowing it to experiment different variable combinations and different models, while turning it more efficient as the user does not have to think about programming. It has options for all parts of the modelling process: (i) data pre-processing, (ii) model selection and (iii) model evaluation. Furthermore, it allows to load data of species and environmental data concerning the Azores Exclusive Economic Zone. Furthermore, allows the user to insert its own data of both species and environmental variables.

The developed tool provides a comprehensive interface to perform the entire modelling process using different state-of-the-art approaches. Nowadays, the two most used software tools for SDM modelling are `MaxEnt` and `R` [56]. The approaches used by these tools are quite different. While `MaxEnt` uses a click approach, `R` uses a syntax driven approach. `DeepData` is the bridge between click and syntax driven approaches. By displaying the available options, the user clicks on the wanted options and `DeepData` generates the syntax for the `R` software. One concern with the click approach is that it works like a 'black-box' software, meaning that the details are hidden from the user. This thesis provides full specification of all default options and options available for all its processes, so that the user is fully conscious of the model.

Although the interface is mostly composed of click options, flexibility is not compromised. Each modelling phase has it own options allowing great tuning, while making it easier for inexperienced users. It also allows multiple SDMs to be fitted and compared simultaneously. This makes comparison between different models is possible because both pre-processing and evaluation methods applied are the same.

`DeepData` usefulness was tested by comparison between published results in scientific forums and the `DeepData`'s results. Two publications were used and for both of them `DeepData` could replicate the

results of the paper, meaning that the tool provides, at least, sufficient flexibility for practical cases.

5.2 Future Work

The major limitation of the DeepData tool is that it does not take into account that some users might want to use data that is private. Most of the studies use data that belongs to the government and therefore data that is not for public use.

One way to resolve this problem is to create information access control. Information access control is composed of authentication and authorization. Authentication is concerned with confirming that the user who says he is, while authorization is concerned with the level of access each user is granted. Given the complexity of the problem, we can simply create a database table for the users information, with attributes such as user_id, email, password and name. All variables added could be stored in a second table, which would have the same primary keys as the Condicoes_ambientais_quarto_grau table and also user_id as a primary key, allowing to associate a user with a variable. We could have a separate table just for the user_id and variable_id but this would be a problem when two users give the same name to a variable. DeepData would load both variables, because joining the tables would not be able to distinguish them.

With this approach, when the user is authenticated, DeepData has not only to load all variables from Condicoes_ambientais_quarto_grau and Terrain_variables_quarto_grau but also the variables associated with the user which are stored in the second hypothetical table.

Another very important and increasing problem is climate change. Successful conservation strategies will require an understanding of climate change and the ability to predict the future. There are two ways of dealing with climate change [57]:

- Mechanistic SDM, which uses physiological information about species to determine the range of environmental conditions that species can tolerate. Then, these tolerances are mapped into geographical space corresponding to the predicted species distribution.

- Climate envelope models, also known as correlative SDM, which rely on statistical correlations between occurrence data and environmental variables to outline a range (envelope) of environmental conditions within which species can exist. Data used for training and testing have a time period different from the data used to project the species distribution.

Since mechanistic SDMs parameters are not derived from the current distribution of the species, the results are independent of the current climate. Therefore, these models have a more accurate understanding of the relationship between climate and the species life cycle. For example, a study made to the reptile Heteronotia binoei [58] concluded where the Australian landscape soil temperature would be suitable for the development of the reptile eggs. Given that the eggs require 600 days of above $20°C$ to achieve complete development. The problem with mechanistic SDMs is that the type of data it uses is hard to get and that it does not account for non-climatic influences such as biotic interactions.

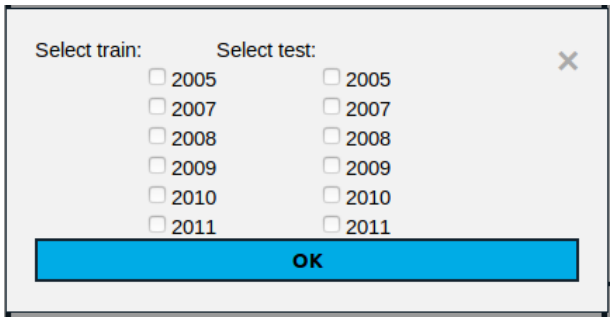


Figure 5.1: `DeepData's` year selection for train and test data.

With climate envelope models, even if biotic interactions are not directly modeled, by considering empirical data of the species distribution, which is constrained by non-climatic variables, these interactions are indirectly considered [59]. For example, take the case where species tolerate a temperature between $10°C$ and $20°C$. But for temperatures between $7°C$ and $12°C$ there is a predator and so for the empirical data species only occur between temperatures of $12°C$ and $20°C$. So, non-climatic variables are indirectly taken into account. Studies were made to evaluate the accuracy of climate envelope models compared to mechanistic SDMs [60, 61].

To implement climate envelope models, we have to allow the selection of the time period for the projection of species distributions. Similarly to the year selection for train and test data, shown in figure 5.1, we would have the constraint that the projection years have to be greater than the years used for training and testing.

Bibliography

[1] IOC-UNESCO and UNEP. *Large Marine Ecosystems: Status and Trends*. United Nations Environment Programme(UNEP), 2016.

[2] IOC-UNESCO and UNEP. *The Open Ocean: Status and Trends*. United Nations Environment Programme(UNEP), 2016.

[3] A. Guisan, R. Tingley, J. B. Baumgartner, I. Naujokaitis-Lewis, P. R. Sutcliffe, A. I. T. Tulloch, T. J. Regan, L. Brotons, E. McDonald-Madden, C. Mantyka-Pringle, T. G. Martin, J. R. Rhodes, R. Maggini, S. A. Setterfield, J. Elith, M. W. Schwartz, B. A. Wintle, O. Broennimann, M. Austin, S. Ferrier, M. R. Kearney, H. P. Possingham, and Y. M. Buckley. Predicting species distributions for conservation decisions. *Ecology Letters*, 16(12):1424–1435, 2013. doi: 10.1111/ele.12189. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/ele.12189`.

[4] M. C. A. Resende. Uma aplicação web para o mar profundo dos Açores. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2018.

[5] A. H. Hirzel and G. Le Lay. Habitat suitability modelling and niche theory. *Journal of Applied Ecology*, 45(5):1372–1381, 2008. doi: 10.1111/j.1365-2664.2008.01524.x. URL `https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2664.2008.01524.x`.

[6] L. L. Porfirio, R. M. B. Harris, E. C. Lefroy, S. Hugh, S. F. Gould, G. Lee, N. L. Bindoff, and B. Mackey. Improving the use of species distribution models in conservation planning and management under climate change. *PLOS ONE*, 9(11):1–21, 2014. doi: 10.1371/journal.pone.0113749. URL `https://doi.org/10.1371/journal.pone.0113749`.

[7] J. Elith and J. Leathwick. Species distribution models: Ecological explanation and prediction across space and time. *Annual Review of Ecology, Evolution and Systematics*, 40:677–697, 2009. doi: 10.1146/annurev.ecolsys.110308.120159.

[8] SantanderMetGroup. Pseudoabsence data generation, 2017. URL `https://github.com/SantanderMetGroup/mopa/wiki/Pseudoabsence-data-generation`.

[9] M. Barbet-Massin, F. Jiguet, C. H. Albert, and W. Thuiller. Selecting pseudo-absences for species distribution models: how, where and how many? *Methods in Ecology and Evolution*, 3(2):327–338, 2008. doi: 10.1111/j.2041-210X.2011.00172.x. URL `https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.2041-210X.2011.00172.x`.

[10] J. VanDerWal, L. P. Shoo, C. Graham, and S. E. Williams. Selecting pseudo-absence data for presence-only distribution modeling: How far should you stray from what you know? *Ecological Modelling*, 220(4):589 – 594, 2009. ISSN 0304-3800. doi: https://doi.org/10.1016/j.ecolmodel.2008.11.010. URL http://www.sciencedirect.com/science/article/pii/S0304380008005486.

[11] C. Seo, J. H. Thorne, L. Hannah, and W. Thuiller. Scale effects in species distribution models: implications for conservation planning under climate change. *Biology Letters*, 2008. doi: 10.1098/rsbl.2008.0476. URL http://rsbl.royalsocietypublishing.org/content/5/1/39.article-info.

[12] Data for species distribution models. https://support.bccvl.org.au/support/solutions/articles/6000161294-data-for-species-distribution-models, 2016.

[13] M. B. A. Carsten F. Dormann, Jana M. McPherson. Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. *Ecography*, 2007. doi: 10.1111/j.2007.0906-7590.05171.x. URL https://doi.org/10.1111/j.2007.0906-7590.05171.x.

[14] B. Crase, A. C. Liedloff, and B. A. Wintle. A new method for dealing with residual spatial autocorrelation in species distribution models. *Ecography*, 35(10):879–888, 2012. doi: 10.1111/j.1600-0587.2011.07138.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0587.2011.07138.x.

[15] P. A. P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950. ISSN 00063444. URL http://www.jstor.org/stable/2332142.

[16] M. B. A. P. Segurado and W. E. kunin. Consequences of spatial autocorrelation for niche-based models. *Journal of Applied Ecology*, 43(3):433–444, 2006. doi: 10.1111/j.1365-2664.2006.01162.x. URL https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2664.2006.01162.x.

[17] C. F. Dormann, J. Elith, S. Bacher, C. Buchmann, G. Carl, G. Carré, J. R. G. Marquéz, B. Gruber, B. Lafourcade, P. J. Leitão, T. Münkemüller, C. McClean, P. E. Osborne, B. Reineking, B. Schröder, A. K. Skidmore, D. Zurell, and S. Lautenbach. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, 36(1):27–46, 2013. doi: 10.1111/j.1600-0587.2012.07348.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0587.2012.07348.x.

[18] R. E. S. Steven J. Phillips, Robert P. Anderson. Maximum entropy modeling of species geographic distributions. *ecological modelling*, 2005. doi: 10.1016/j.ecolmodel.2005.03.026. URL https://doi.org/10.1016/j.ecolmodel.2005.03.026.

[19] S. J. Phillips, M. Dudík, and R. E. Schapire. A maximum entropy approach to species distribution modeling. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015412. URL http://doi.acm.org/10.1145/1015330.1015412.

[20] A practical guide to maxent for modeling species' distributions: what it does, and why inputs and settings matter. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1600-0587.2013.07872.x, 2013.

[21] A. J. Dobson. *An Introduction to Generalized Linear Models, Second Edition*. Taylor & Francis, 2010. ISBN 1420057685.

[22] T. Hastie and R. Tibshirani. *Generalized Additive Model*. American Cancer Society, 2005. ISBN 9780470011812. doi: 10.1002/0470011815.b2a09018. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/0470011815.b2a09018.

[23] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957. URL http://books.google.com/books?id=fyVtp3EMxasC&pg=PR5&dq=dynamic+programming+richard+e+bellman&client=firefox-a#v=onepage&q=dynamic%20programming%20richard%20e%20bellman&f=false.

[24] K. Larsen. Gam: The predictive modeling silver bullet. https://multithreaded.stitchfix.com/assets/files/gam.pdf, 2015.

[25] P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31, 10 January 1978. URL https://doi.org/10.1007/BF01404567.

[26] C. Drew, Y. Wiersma, and F. Huettmann. *Predictive Species and Habitat Modeling in Landscape Ecology: Concepts and Applications*. Springer New York, 2010. ISBN 9781441973900. URL https://books.google.pt/books?id=1V5gupaI5_IC.

[27] L. Breiman. *Classification and regression trees*. Wadsworth statistics/probability series. Wadsworth International Group, 1984. ISBN 9780534980535. URL https://books.google.pt/books?id=uxPvAAAAMAAJ.

[28] Glms, gams, and carts. http://geog.uoregon.edu/bartlein/courses/geog495/lec15.html, 2018.

[29] Random forest explained. https://www.kdnuggets.com/2017/10/random-forests-explained.html, 2017.

[30] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[31] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995. ISSN 1573-0565. doi: 10.1007/BF00994018. URL https://doi.org/10.1007/BF00994018.

[32] T. Fletcher. Support vector machines explained. https://cling.csd.uwo.ca/cs860/papers/SVM_Explained.pdf, 2008.

[33] Everything you wanted to know about the kernel trick. http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html, 2017.

[34] G. Palm. Warren mcculloch and walter pitts: A logical calculus of the ideas immanent in nervous activity. In G. Palm and A. Aertsen, editors, *Brain Theory*, pages 229–230, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg. ISBN 978-3-642-70911-1.

[35] R. Morris. D.o. hebb: The organization of behavior, wiley: New york; 1949. *Brain Research Bulletin*, 50(5):437, 1999. ISSN 0361-9230. doi: https://doi.org/10.1016/S0361-9230(99)00182-3. URL http://www.sciencedirect.com/science/article/pii/S0361923099001823.

[36] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain [j]. *Psychol. Review*, 65:386 – 408, 1958. doi: 10.1037/h0042519.

[37] An introduction to neural networks for beginners. https://adventuresinmachinelearning.com/wp-content/uploads/2017/07/An-introduction-to-neural-networks-for-beginners.pdf, 2017.

[38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back propagating errors. *Nature*, 323:533–536, 1986. doi: 10.1038/323533a0.

[39] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 1st edition, 2012. ISBN 1439830037, 9781439830031.

[40] J. Pearce and S. Ferrier. Evaluating the predictive performance of habitat models developed using logistic regression. *Ecological Modelling*, 133(3):225 – 245, 2000. ISSN 0304-3800. doi: https://doi.org/10.1016/S0304-3800(00)00322-7. URL http://www.sciencedirect.com/science/article/pii/S0304380000003227.

[41] A. Guisan and N. E. Zimmermann. Predictive habitat distribution models in ecology. *Ecological Modelling*, 135(2):147 – 186, 2000. ISSN 0304-3800. doi: https://doi.org/10.1016/S0304-3800(00)00354-9. URL http://www.sciencedirect.com/science/article/pii/S0304380000003549.

[42] O. Allouche, A. Tsoar, and R. Kadmon. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (tss). *Journal of Applied Ecology*, 43(6):1223–1232, 2006. doi: 10.1111/j.1365-2664.2006.01214.x. URL https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2664.2006.01214.x.

[43] Cross validation and the bias-variance tradeoff. https://codesachin.wordpress.com/2015/08/30/cross-validation-and-the-bias-variance-tradeoff-for-dummies/, 2015.

[44] W. ocean atlas 13. https://www.nodc.noaa.gov/OC5/woa13/woa13data.html.

[45] Emodnet. http://www.emodnet.eu/.

[46] Obis. https://obis.org/.

[47] Worms. http://www.marinespecies.org/.

[48] H. E. Parra, C. K. Pham, G. M. Menezes, A. Rosa, F. Tempera, and T. Morato. Predictive modeling of deep-sea fish distribution in the azores. *Deep Sea Research Part II: Topical Studies in Oceanography*, 145:49 – 60, 2017. doi: https://doi.org/10.1016/j.dsr2.2016.01.004.

[49] S. Fukuda, B. D. Baets, W. Waegeman, J. Verwaeren, and A. M. Mouton. Habitat prediction and knowledge extraction for spawning european grayling (thymallus thymallus l.) using a broad range of species distribution models. *Environmental Modelling & Software*, 47:1 – 6, 2013. ISSN 1364-8152. doi: https://doi.org/10.1016/j.envsoft.2013.04.005. URL http://www.sciencedirect.com/science/article/pii/S1364815213001047.

[50] L. Krüger. Population estimates of trindade petrel (pterodroma arminjoniana) by ensemble nesting habitat modelling. 2018. doi: 10.19080/IJESNR.2018.10.555793.

[51] J. Lai, C. J. Lortie, R. A. Muenchen, J. Yang, and K. Ma. Evaluating the popularity of r in ecology. *Ecosphere*, 10(1):e02567, 2019. doi: 10.1002/ecs2.2567. URL https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1002/ecs2.2567.

[52] M. Costello, Z. Basher, R. Sayre, S. Breyer, and D. Wright. Stratifying ocean sampling globally and with depth to account for environmental variability. *Scientific Reports*, 8, 2018. doi: 10.1038/s41598-018-29419-1.

[53] D. A. Nachtsheim, K. Jerosch, W. Hagen, J. Plötz, and H. Bornemann. Habitat modelling of crabeater seals (lobodon carcinophaga) in the weddell sea using the multivariate approach maxent. *Polar Biology*, 40(5):961–976, May 2017. ISSN 1432-2056. doi: 10.1007/s00300-016-2020-0. URL https://doi.org/10.1007/s00300-016-2020-0.

[54] D. A. Nachtsheim, K. Jerosch, W. Hagen, J. Plötz, and H. Bornemann. Crabeater seals (Lobodon carcinophaga) in the Weddell Sea during DRE1998 campaign, with link to files of gridded seal locations and environmental parameters for Maxent analyses. PANGAEA, 2015. doi: 10.1594/PANGAEA.855006. URL https://doi.org/10.1594/PANGAEA.855006. In supplement to: Nachtsheim, DA et al. (2016): Habitat modelling of crabeater seals (Lobodon carcinophaga) in the Weddell Sea using the multivariate approach Maxent. Polar Biology, 40(5), 961-976, https://doi.org/10.1007/s00300-016-2020-0.

[55] J. Elith, S. Ferrier, F. Huettmann, and J. Leathwick. The evaluation strip: A new and robust method for plotting predicted responses from species distribution models. *Ecological Modelling*, 186:280–289, 2005. doi: 10.1016/j.ecolmodel.2004.12.007.

[56] E. Meineri, A.-S. Deville, D. Grémillet, M. Gauthier-Clerc, and A. Béchet. Combining correlative and mechanistic habitat suitability models to improve ecological compensation. *Biological Reviews*, 90 (1):314–329, 2015. doi: 10.1111/brv.12111. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/brv.12111.

[57] S. E. Ahmed, G. McInerny, K. O'Hara, R. Harper, L. Salido, S. Emmott, and L. N. Joppa. Scientists and software – surveying the species distribution modelling community. *Diversity and Distributions*,

21(3):258–267, 2015. doi: 10.1111/ddi.12305. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/ddi.12305.

[58] M. Kearney and W. P. Porter. Mapping the fundamental niche: Physiology, climate, and the distribution of a nocturnal lizard. *Ecology*, 85(11):3119–3131, 2004. doi: 10.1890/03-0820. URL https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/03-0820.

[59] R. G. Pearson and T. P. Dawson. Predicting the impacts of climate change on the distribution of species: are bioclimate envelope models useful? *Global Ecology and Biogeography*, 12(5):361–371, 2003. doi: 10.1046/j.1466-822X.2003.00042.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1466-822X.2003.00042.x.

[60] R. J. Hijmans and C. H. Graham. The ability of climate envelope models to predict the effect of climate change on species distributions. *Global Change Biology*, 12(12):2272–2281, 2006. doi: 10.1111/j.1365-2486.2006.01256.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2486.2006.01256.x.

[61] M. R. Kearney, B. A. Wintle, and W. P. Porter. Correlative and mechanistic models of species distribution provide congruent forecasts under climate change. *Conservation Letters*, 3(3):203–213, 2010. doi: 10.1111/j.1755-263X.2010.00097.x. URL https://conbio.onlinelibrary.wiley.com/doi/abs/10.1111/j.1755-263X.2010.00097.x.

Appendix A

Tool's flow diagrams

In this section it is presented the flow diagrams of the tool's processes. The purpose of these diagrams, is to provide a simple and fast visualization of all the processes provided by the tool.

A.1 Tool processes

On figure A.1, are presented the flowchart symbols and names, from which meaning can be deduced, used on the diagrams.

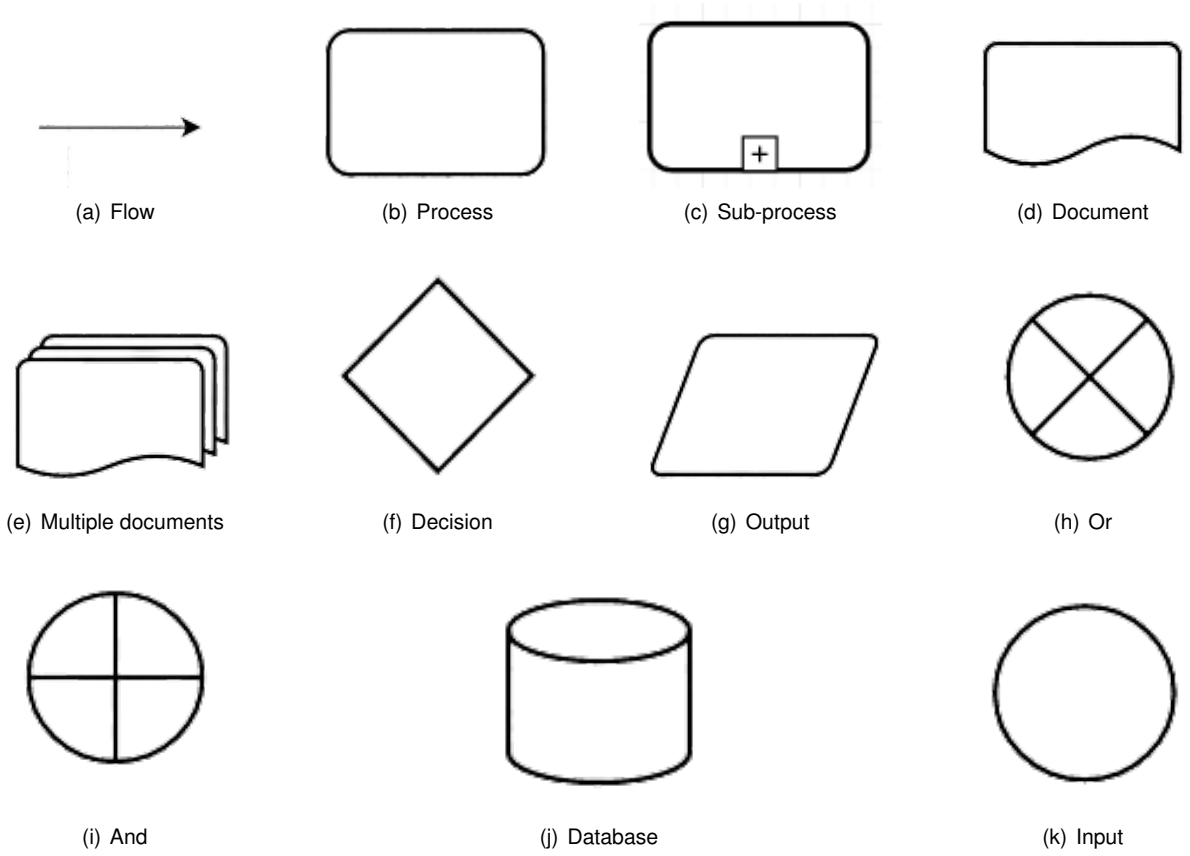


Figure A.1: Flowchart symbols and name.

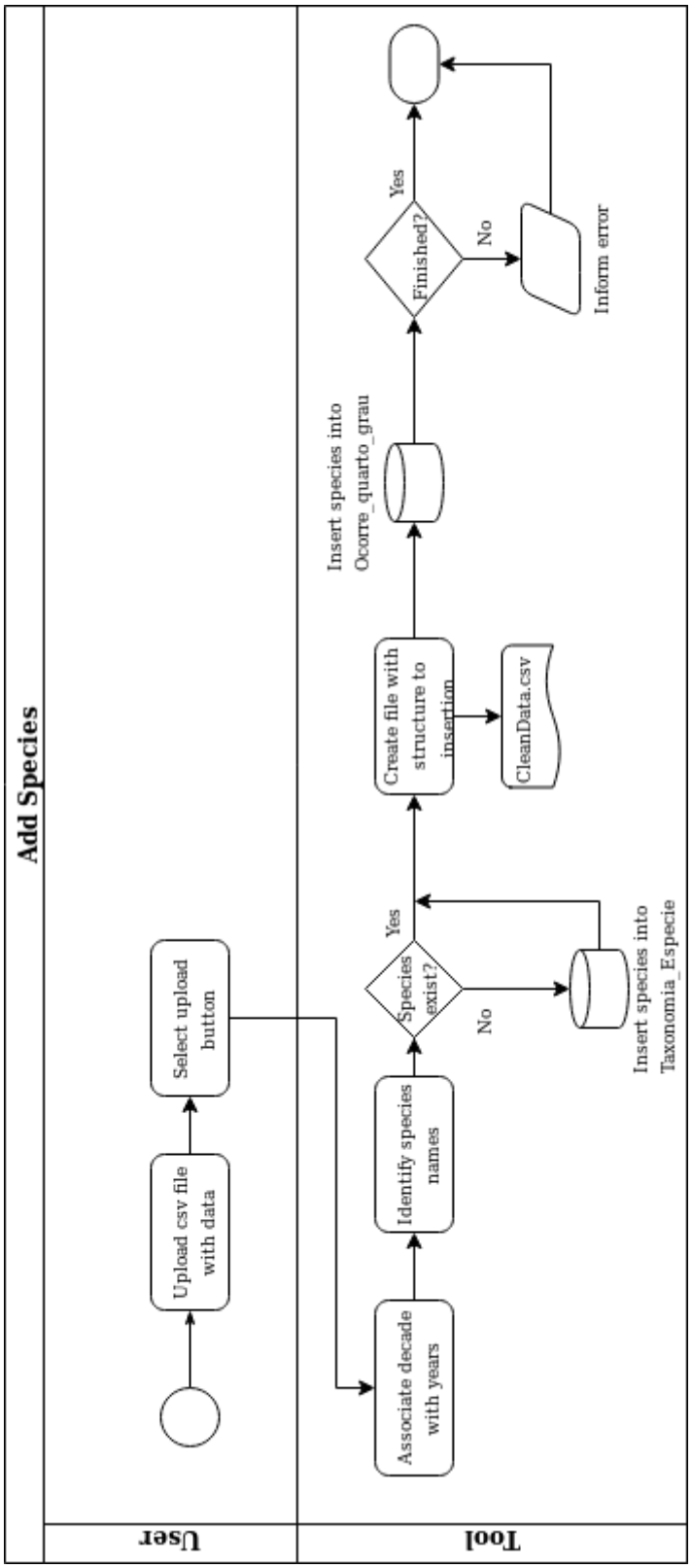


Figure A.2: Species data insertion process

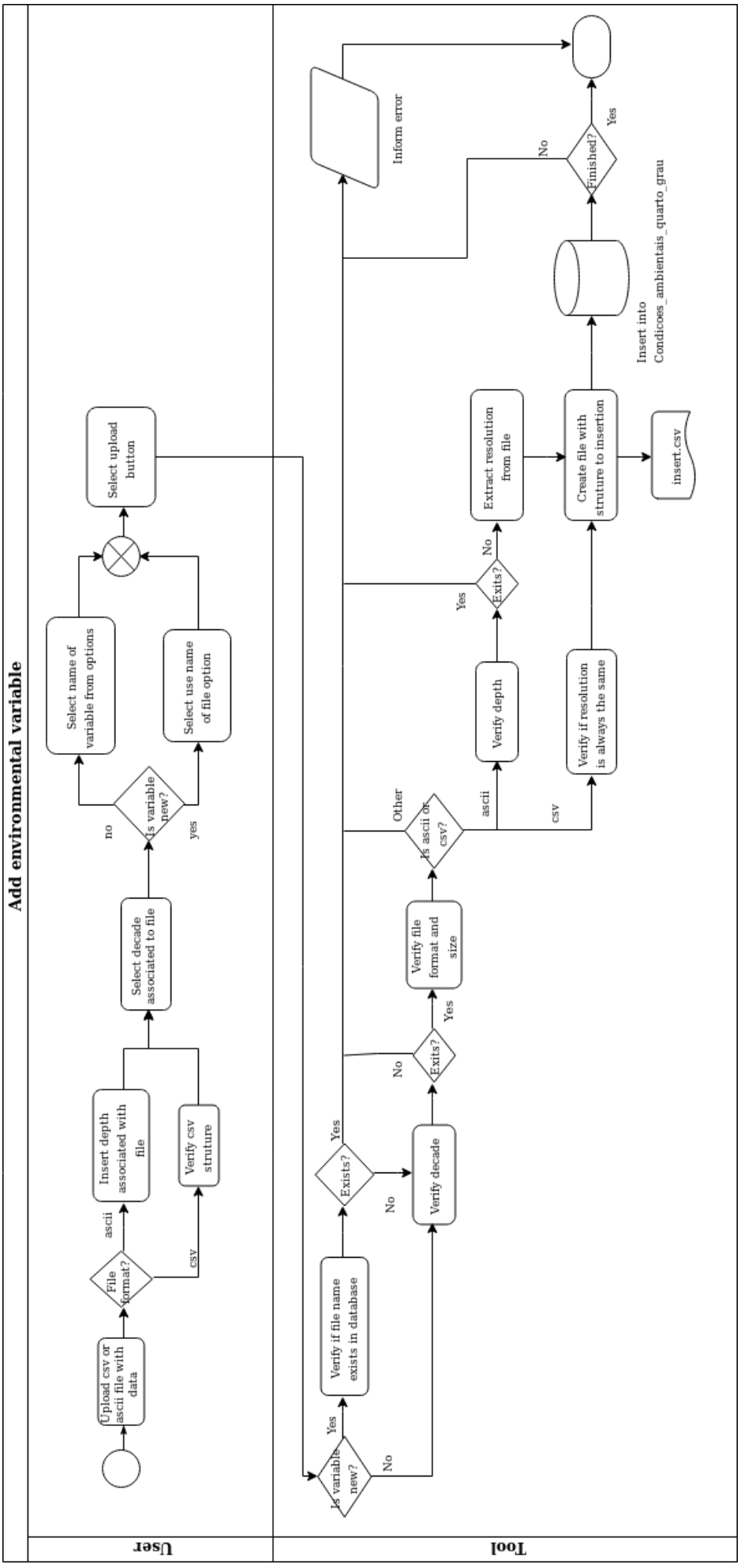


Figure A.3: Environmental data insertion process

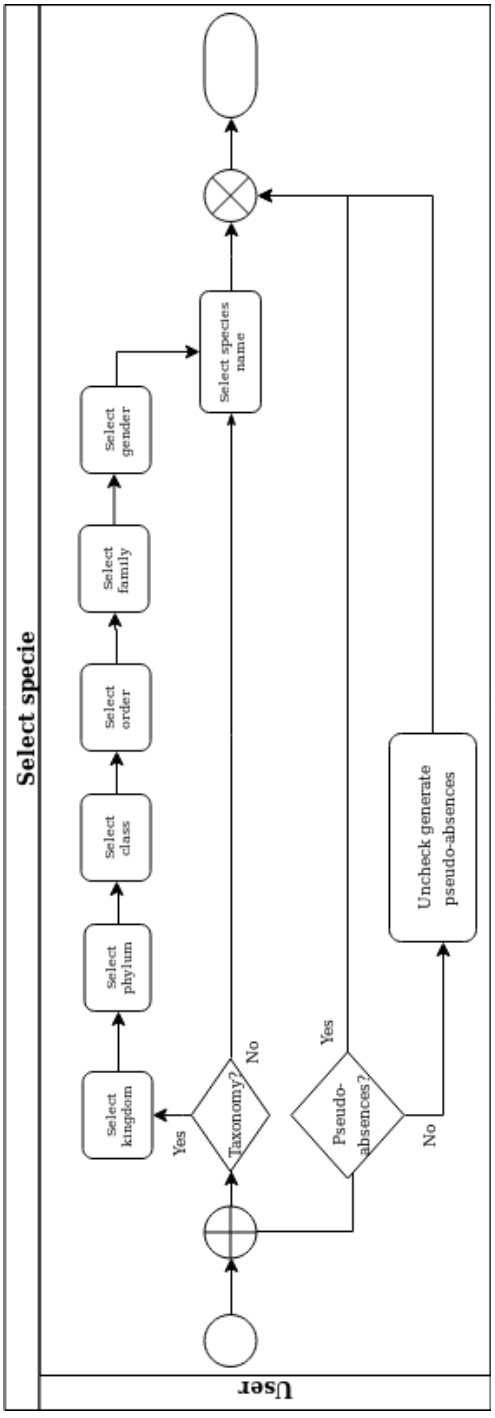


Figure A.4: Specie selection process

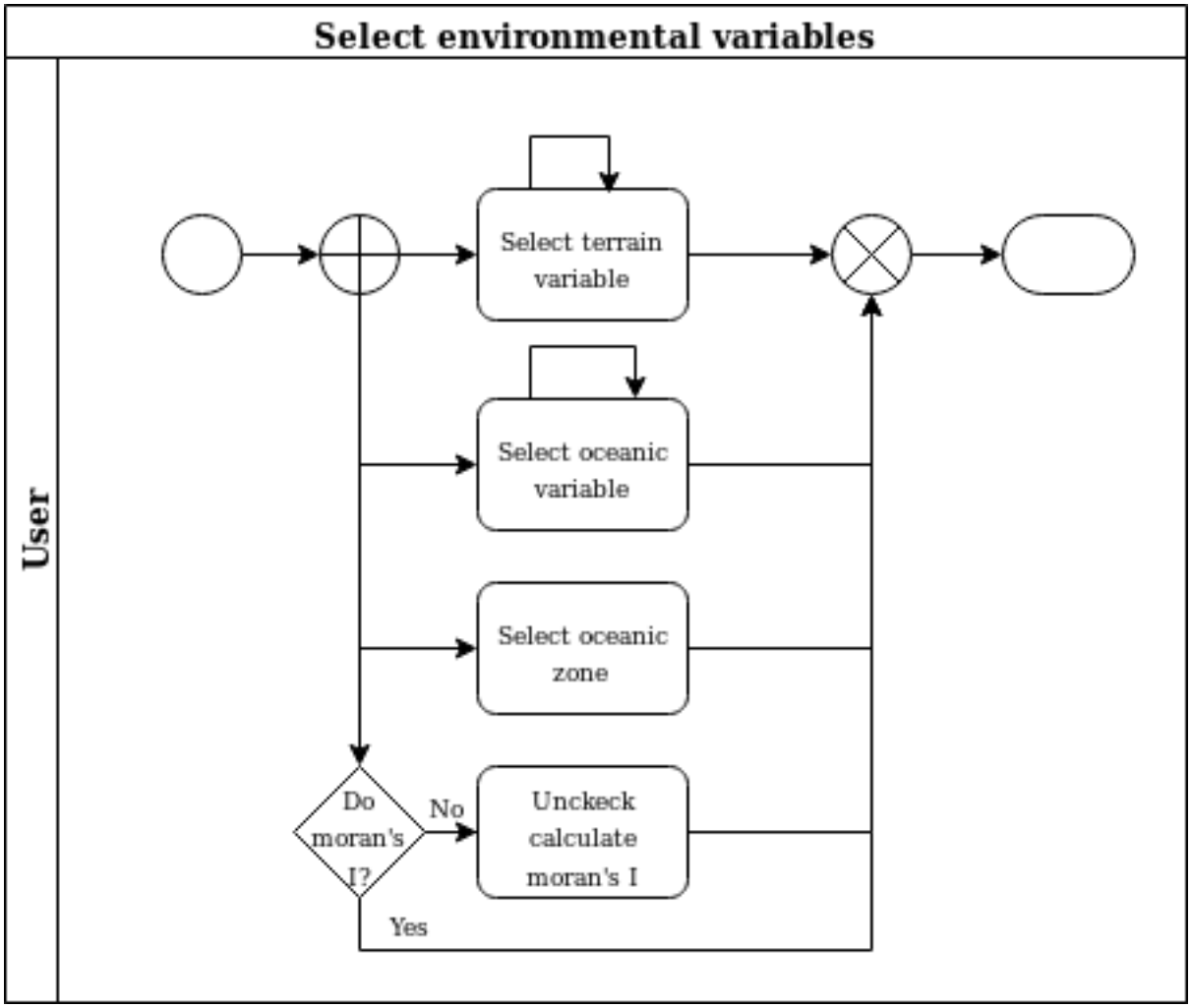


Figure A.5: Environmental variable selection process.

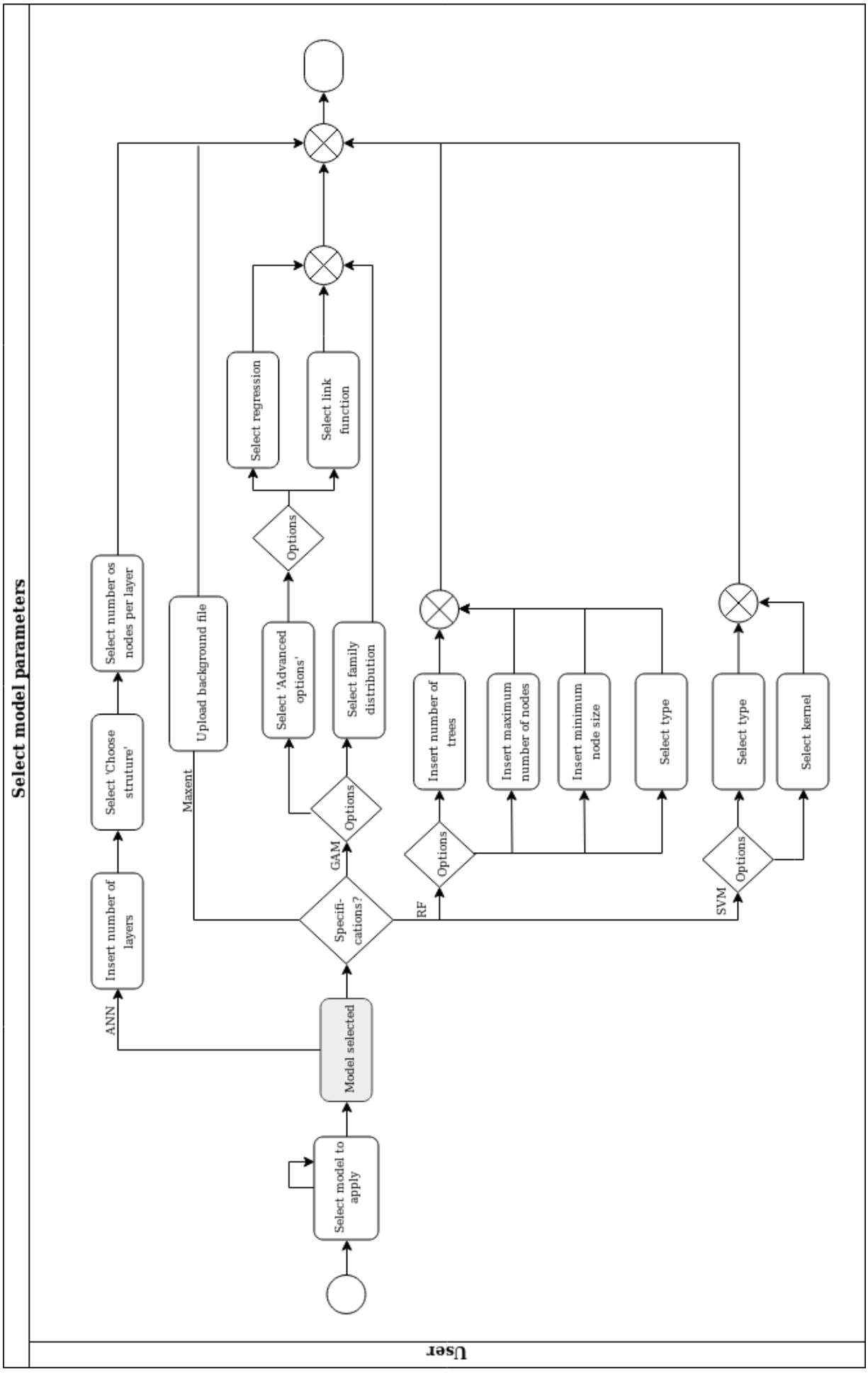


Figure A.6: Model parameters selection process.

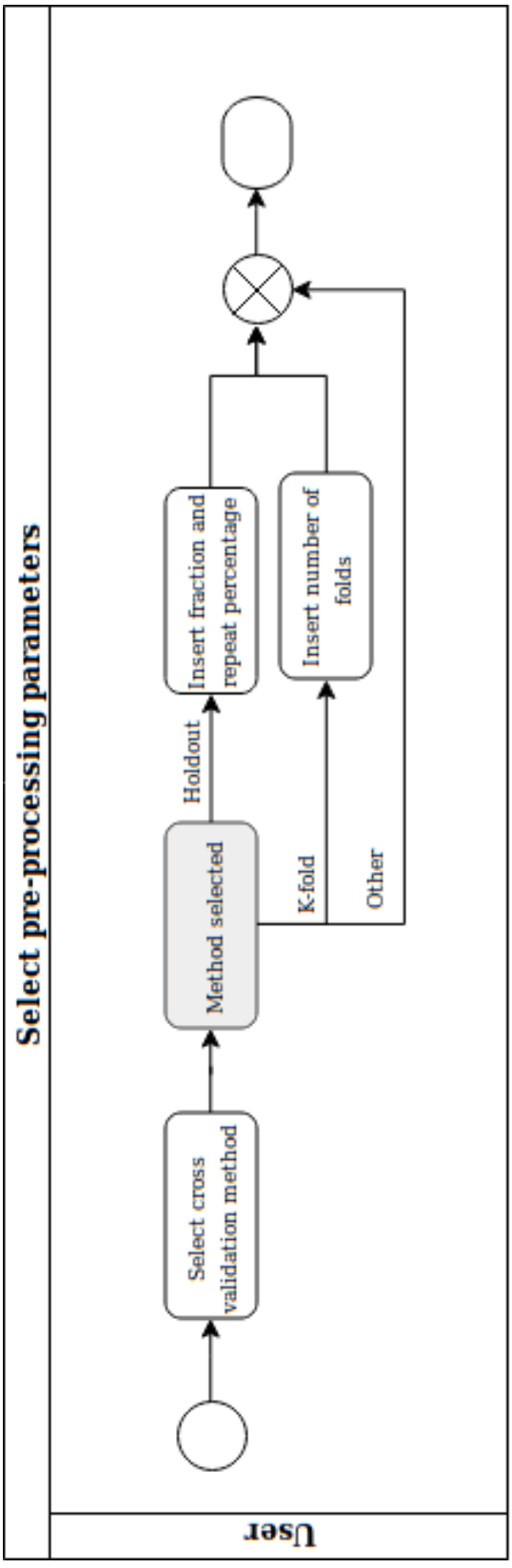


Figure A.7: Pre-processing parameters selection process.

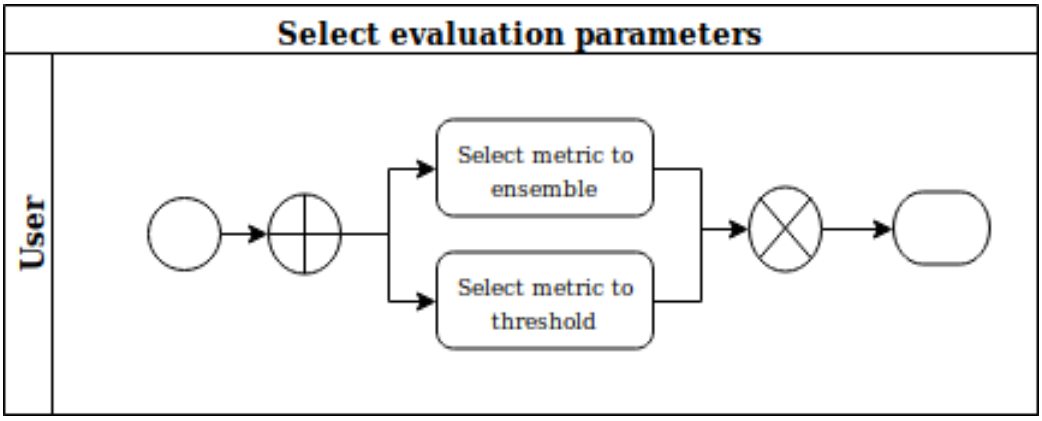


Figure A.8: Evaluation parameters selection process.

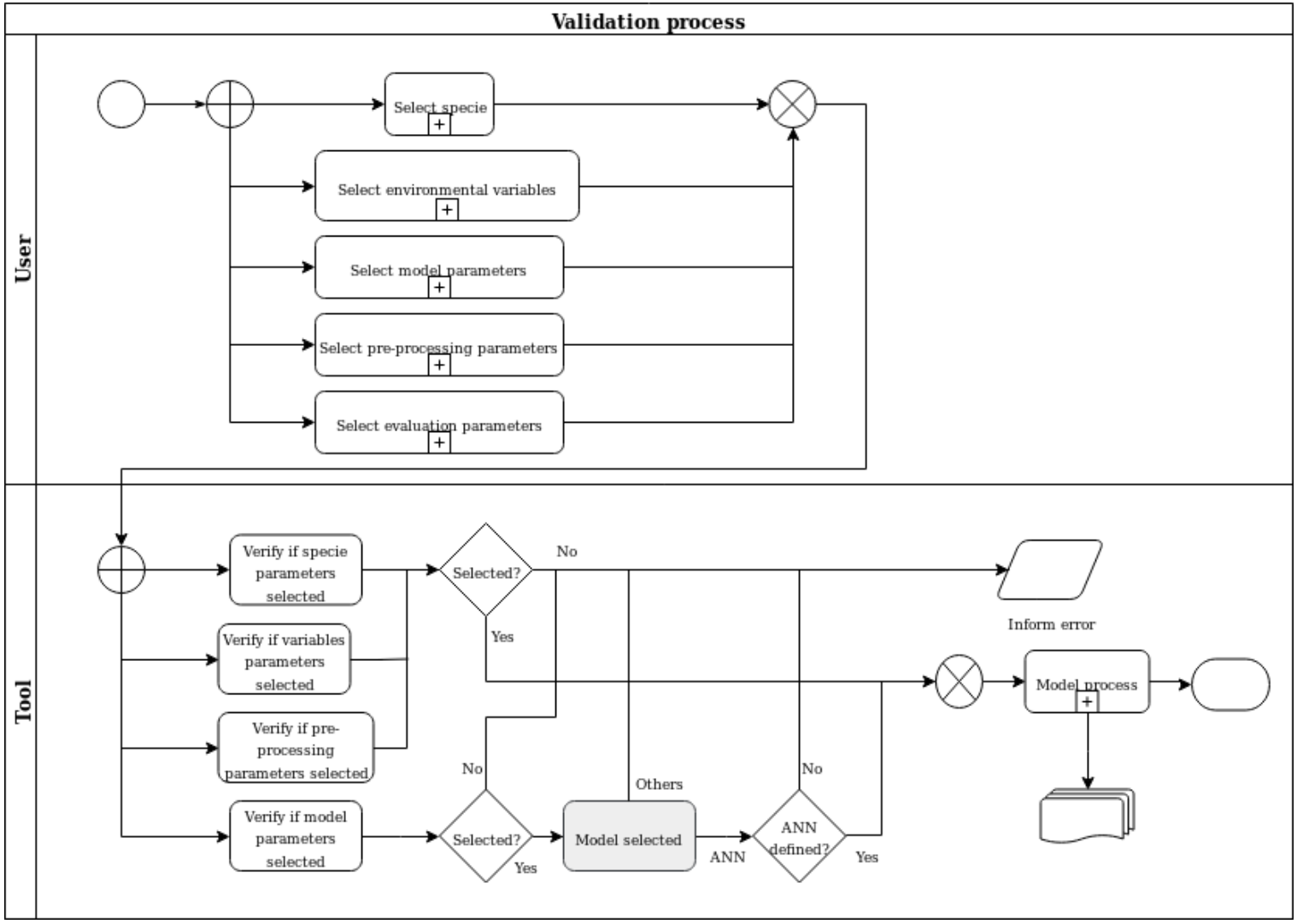


Figure A.9: Validation process.

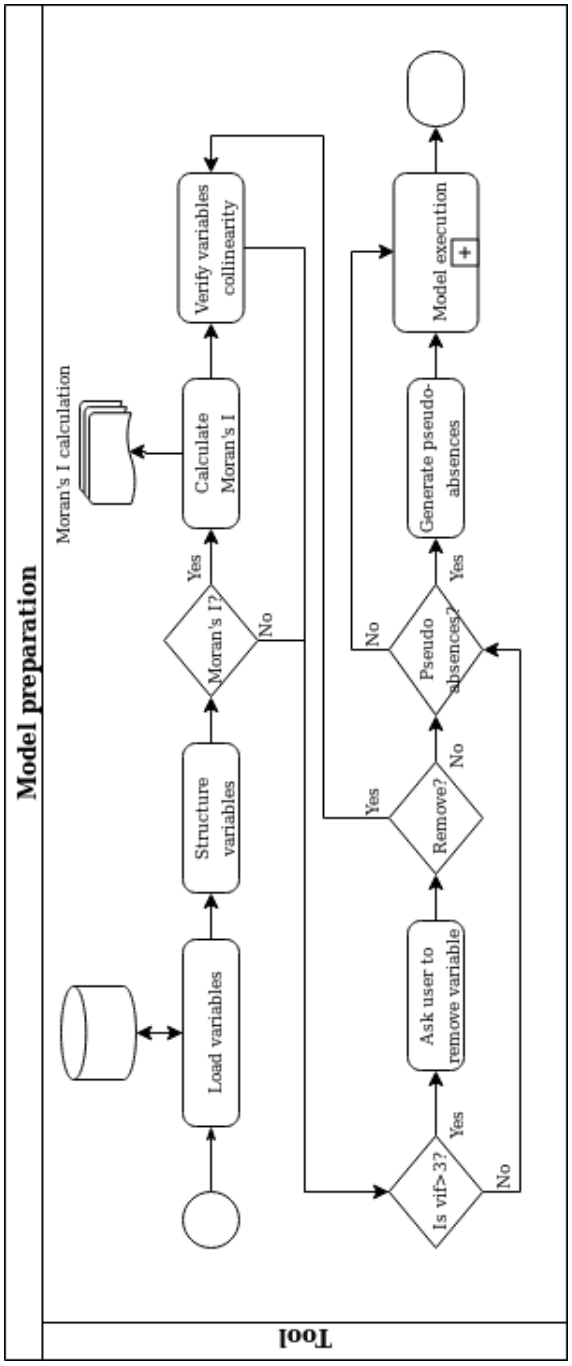


Figure A.10: Model preparation process.

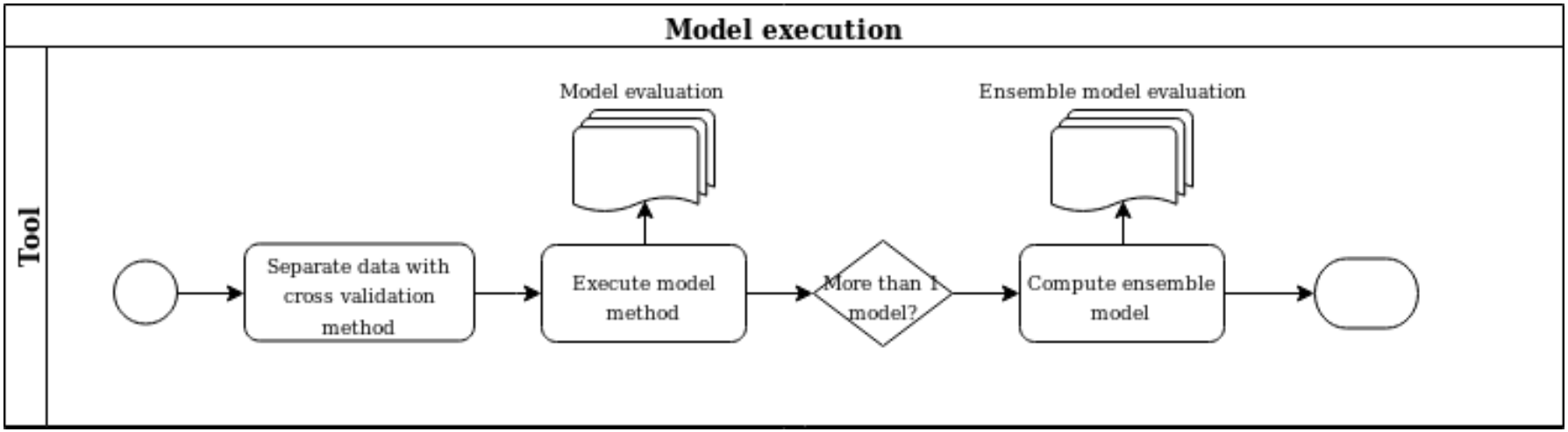
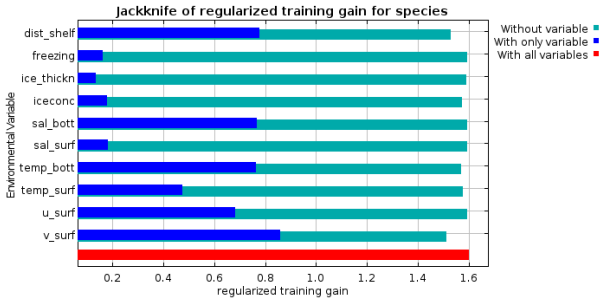


Figure A.11: Model execution process.

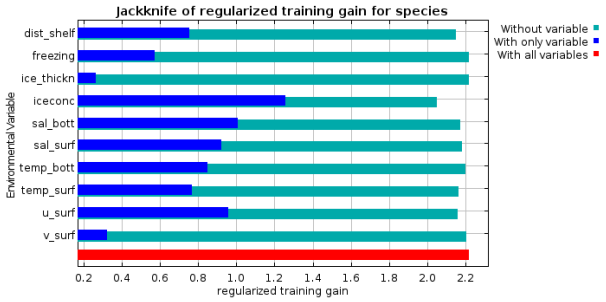
Appendix B

Case study results

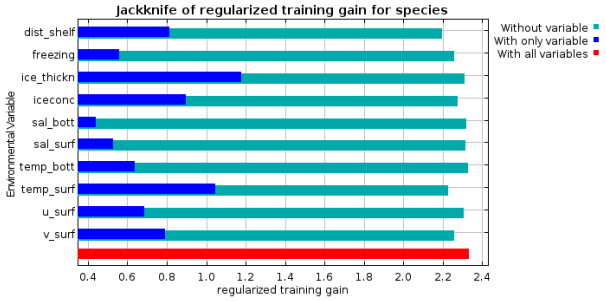
B.1 Jackknife results



(a) February



(b) March



(c) April

Figure B.1: Jackknife results for the selected 10 variables.

B.2 Response curves first case

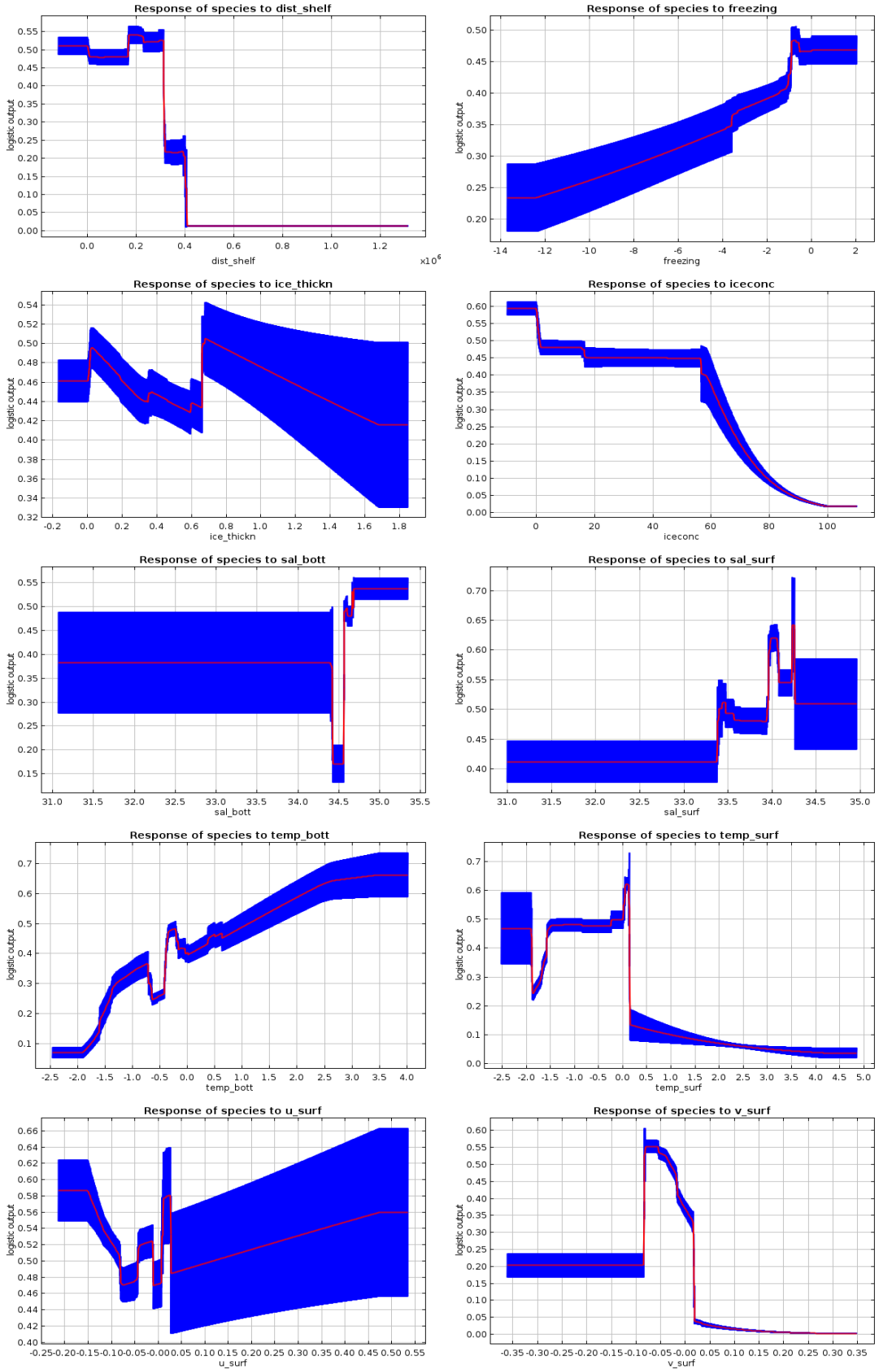


Figure B.2: Response curve for February.

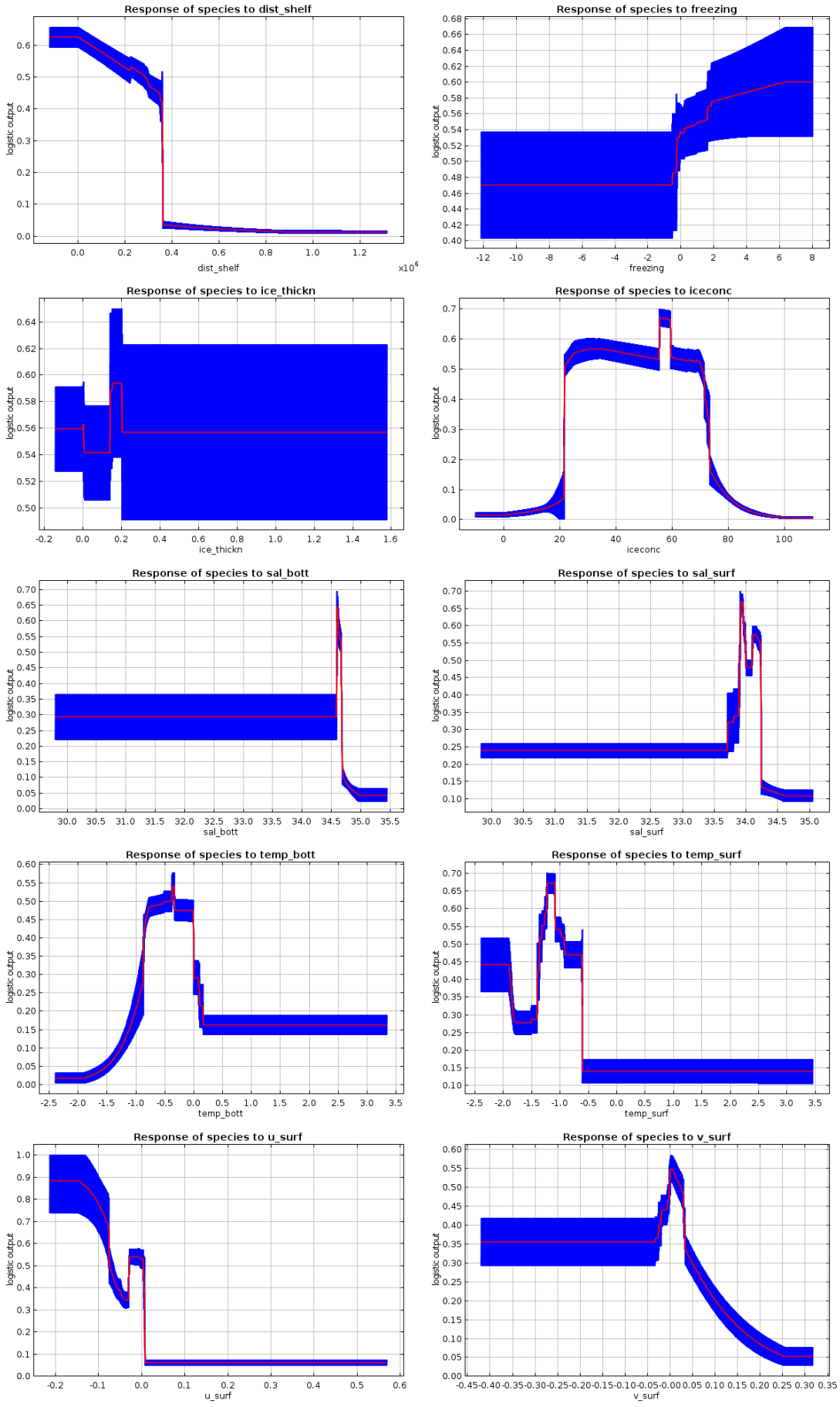


Figure B.3: Response curve for March.

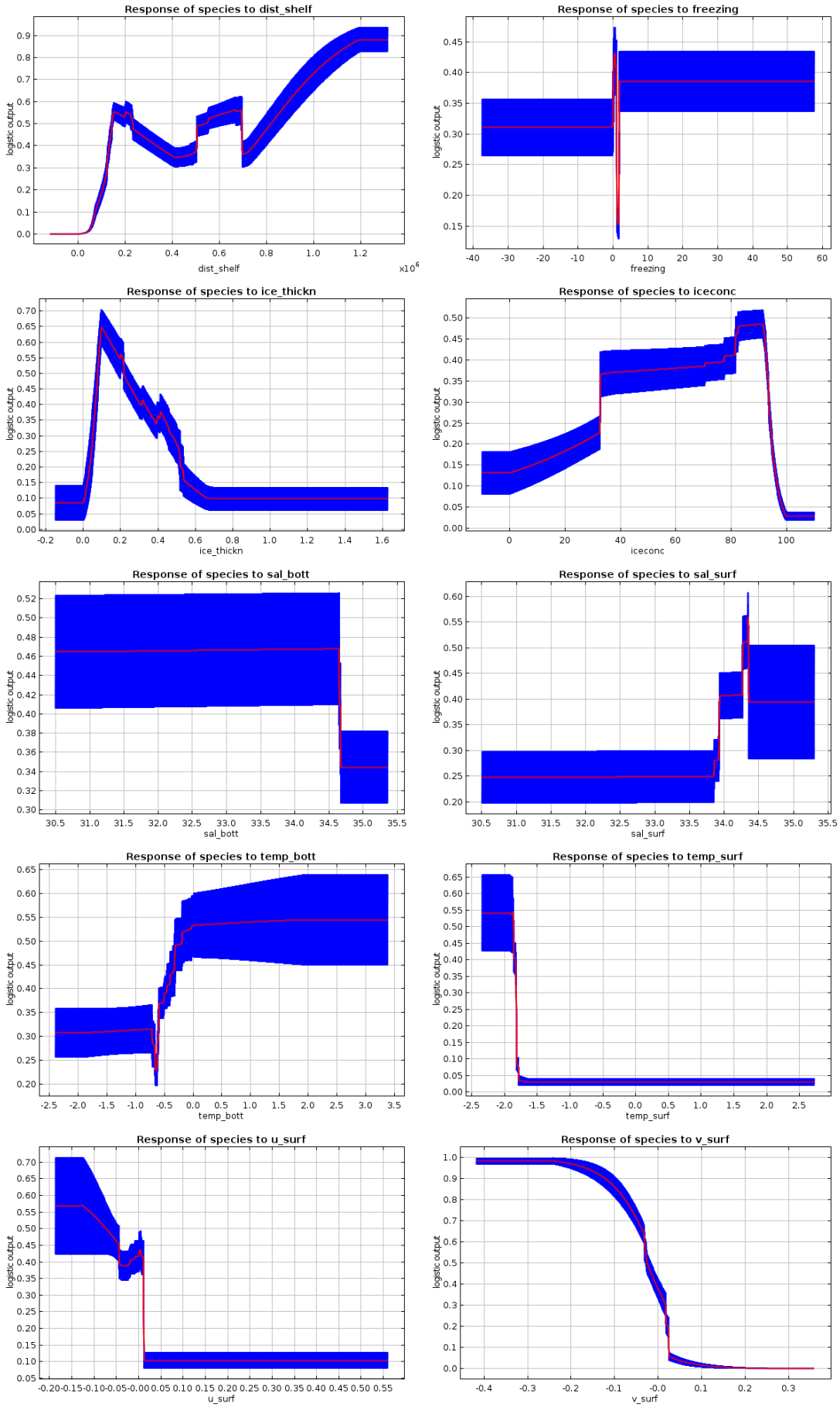


Figure B.4: Response curve for April.

B.3 Response curves second case

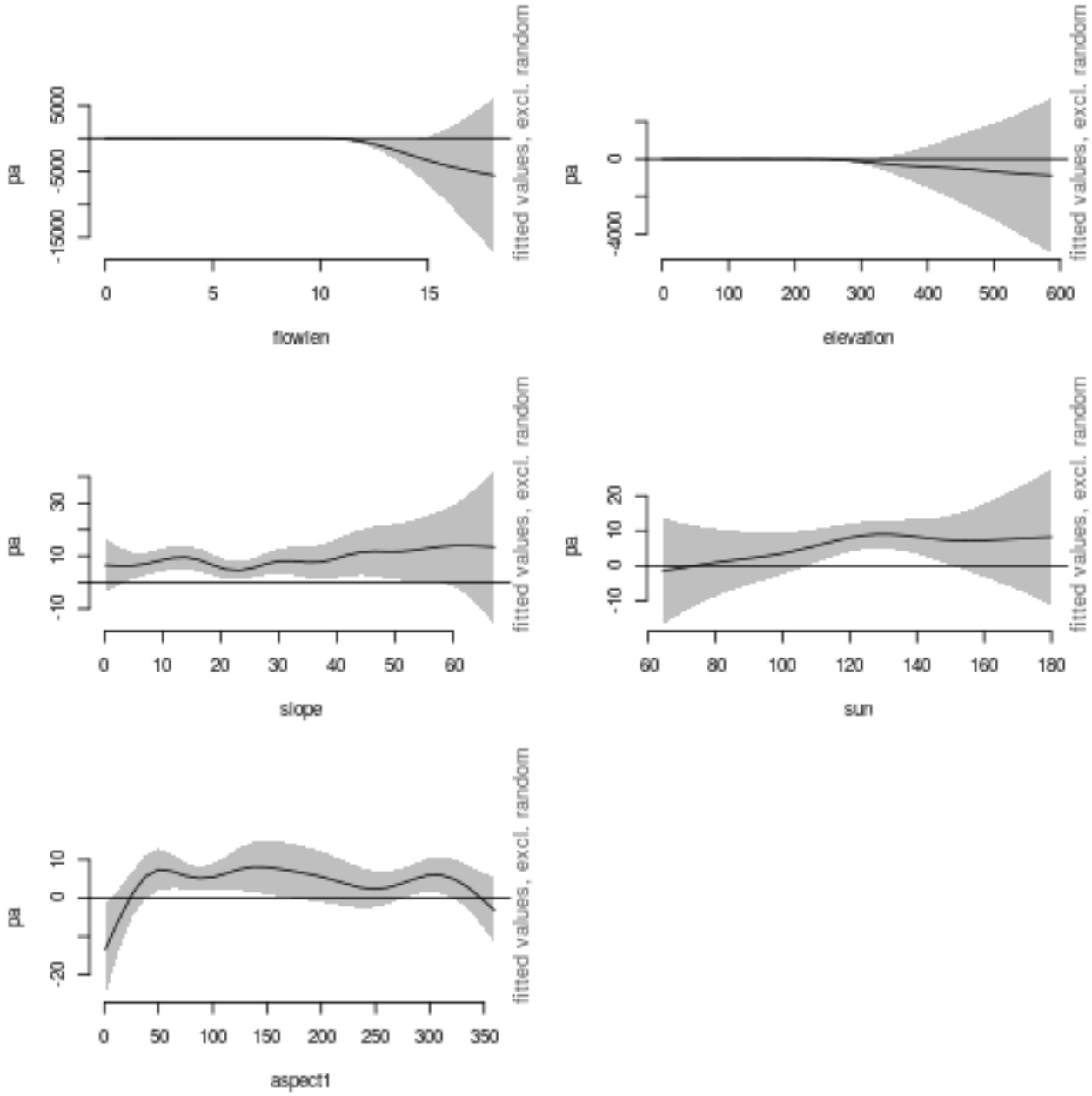


Figure B.5: Partial dependence plot for the generalized addictive model.

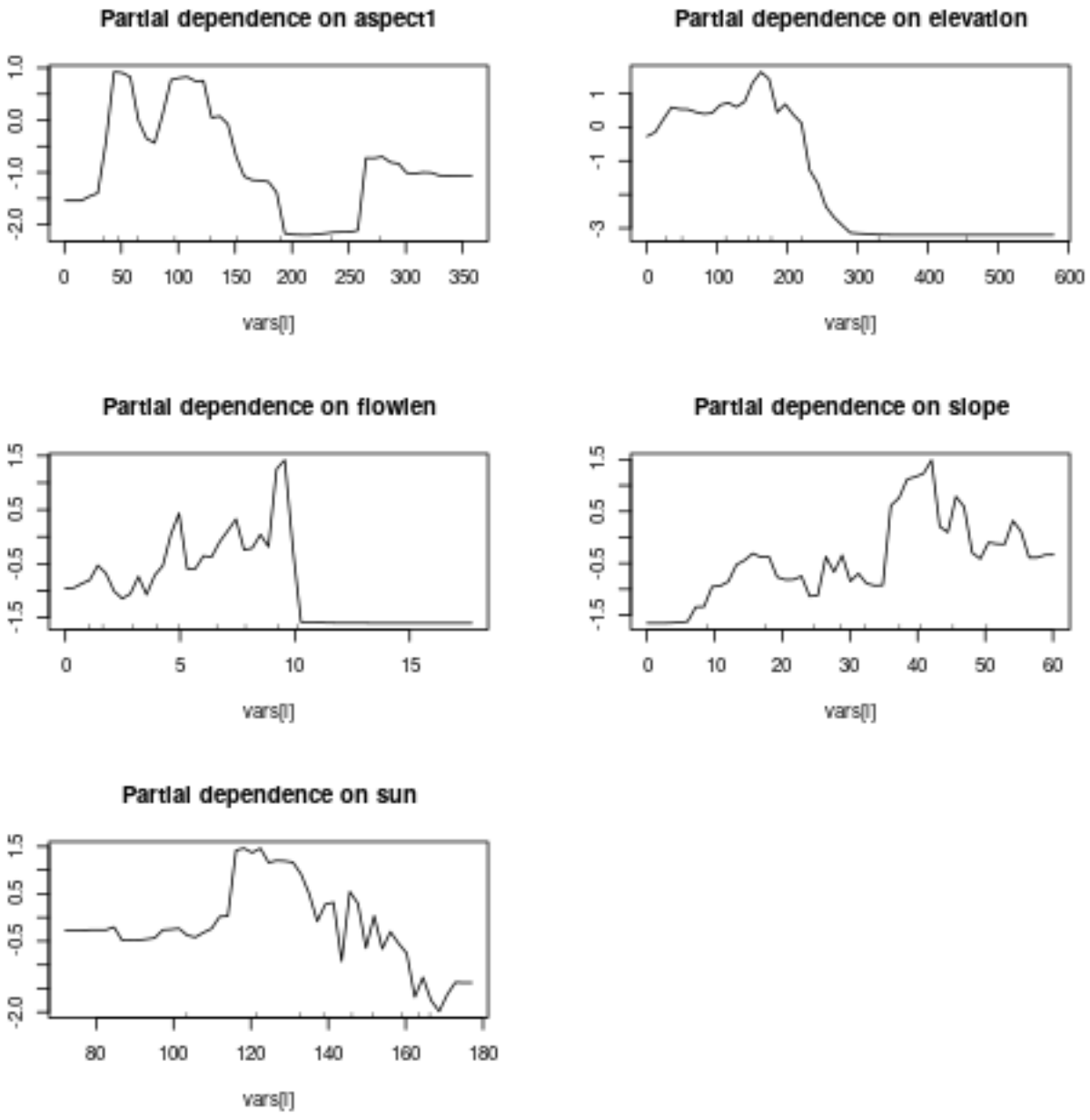


Figure B.6: Partial dependence plot for the random forest model.