

Machine Learning for Biodiversity Scientists

An opinionated primer

Timothée Poisot

2024-11-12

Table of contents

Preface	1
1. Introduction	3
1.1. Core concepts in data science	5
1.2. An overview of the content	5
1.3. A note on colors	6
1.4. Some rules about this book	8
References	11
2. Preparing features	15
2.1. The problem: optimal set of BioClim variables for the Corsican nuthatch	15
2.2. What is data leakage?	17
2.3. Variable selection	20
2.4. Multivariate transformations	23
2.5. Application: optimal variables for Corsican nuthatch	24
2.6. The Rashomon effect	28
2.7. Conclusion	29
Appendices	33
References	33
A. Instructor notes	37
References	39
Index	43

List of Figures

2.1. Importance of variables selected as part of the best model	30
2.2. TODO	31
2.3. Updated range map for Sitta whiteheadi after variable selection.	32

Preface

Machine learning is now an established methodology to study biodiversity, and this is a problem.

This may be an opportunity when it comes to advancing our knowledge of biodiversity, and in particular when it comes to translating this knowledge into action (Tuia *et al.* 2022); but make no mistake, this is a problem for us, biodiversity scientists, as we suddenly need to develop competences in an entirely new field in order to remain professionally relevant (Ellwood *et al.* 2019). And as luck would have it, there are easier fields to master than machine learning. The point of this book, therefore, is to provide an introduction to fundamental concepts in data science, from the perspective of a biodiversity scientist, by using examples corresponding to real-world use-cases of these techniques.

But what do we mean by *machine learning* and *data science*? Most science, after all, relies on data in some capacity. What falls under the umbrella of *data science* is, in short, embracing in equal measure quantitative skills (mathematics, machine learning, statistics), programming, and domain expertise, in order to solve well-defined problems. *Machine learning* is a series of techniques (or, more precisely, a high-level approach to these techniques) through which we conduct our data science activities. A core tenet of data science is that, when using it, we seek to “deliver actionable insights”, which is MBA-speak for “figuring out what to do next”. One of the ways in which this occurs is by letting the data speak, after they have been, of course, properly cleaned and transformed and engineered. This entire process is driven by (or, even, subject to) domain knowledge. There is no such thing as data science, at least not in a vacuum: there is data science as a methodology applied to a specific domain.

Think of data science as being its own epistemology (Desai *et al.* 2022), and machine learning as one methodology we can apply to work within this context.

Before we embark into a journey of discovery on the applications of data science to biodiversity, allow me to let you in on a little secret: *data science* is a little bit of a misnomer. In order to understand why, I need (or at least, I really want) to talk about cooking.

Preface

To become a good cook, there are general techniques one *must* master, which we apply to specific steps in recipes; these recipes draw from a common cultural or local repertoire and cultural specifics (but the evolution of recipes is remarkably convergent – most cuisines have a *mirepoix*, bread, and beer). Finally, there is the product, *i.e.* the unique dish that you have cooked. And so it is for data science too: we can abstract a series of processes and guidelines, think about their application within the context of our specific field, study system, or line and research, and all of this will shape the final data product we can serve.

When writing this preface, I turned to my shelf of cookbooks, and picked my two favorites: Robuchon's *The Complete Robuchon* (a no-nonsense list of hundreds of recipes with no place for improvisation), and Bianco's *Pizza, Pasta, and Other Food I Like* (a short volume with very few pizza and pasta, and wonderful discussions about the importance of humility, creativity, and generosity). Data science, if it were cooking, would feel a lot like the second. Deviation from the rules is often justifiable if you feel like it. But this improvisation requires good skills, a clear mental map of the problem, a defined vision of what these deviations will let you achieve, and a library of patterns that you can draw from.

This book will not get you here. But it will speed up the process, by framing the practice of data science as a natural way to conduct research on biodiversity.

1. Introduction

This book started as a collection of notes from several classes I taught in the Department of Biological Sciences at the Université de Montréal, as well as a few workshops I ran for the Québec Centre for Biodiversity Sciences. When teaching data synthesis, data science, and machine learning to biology students, I realized that the field was missing resources that could serve as stepping stones to proficiency.

There are excellent manuals covering the mathematics of data science and machine learning (I will list a few later on). These are important to read, because the field of machine learning is an offshoot of mathematics and computer science, and it is important to become familiar with the core concepts. A little bit of calculus and a whole lot of linear algebra should be more of the same for many ecologists. But these resources are usually less useful as practical guides to the field.

There are many good papers giving overviews of some applications of data science to biological problems (a lot of them are cited in this book). These are important to read, because any attempt to adopt a new methodology (new to us, not new to the field, or new in absolute terms!) must proceed alongside some familiarity of how it has been used by our colleagues. But these articles, although good at showing how these tools are actually used, usually make it difficult to establish more general recommendations.

There are, finally, thousands of tutorials about how to write code to perform any machine learning algorithm you can think of. Some of them are even good. But these tutorials usually suffer (in our case) from being disconnected from the field of biodiversity science, and of course are limited by the language they use, the version of the packages they ran with, and again do not allow for much generalization.

When navigating these resources, one thing that students commonly called for was an attempt to tie concepts together, and to explain when and how human decisions were required in ML approaches (Sulmont *et al.* 2019).

1. Introduction

This is particularly true of students with strong domain knowledge that want to understand how machine learning fits with their ability to do research.

This book is this attempt.

There are, broadly speaking, two situations in which reading this book is useful. The first is when you are done reading some general books about machine learning, and want to see how it can be applied to problems that are more specific to biodiversity research; the second is when you have a working understanding of biodiversity research, and want a stepping stone into the machine learning literature. Note that there is no scenario where you *stop* after reading this book – this is by design. The purpose of this book is to give a practical overview of “how data science for biodiversity happens”, and this needs to be done in parallel to even more fundamental readings.

These are examples of books I like. I found them comprehensive and engaging. They may not work for you.

A wonderful introduction to the mathematics behind machine learning can be found in Deisenroth *et al.* (2020), which provides stunning visualization of mathematical concepts. Yau (2015) is a particularly useful book about the ways to visualize data in a meaningful way. Watt *et al.* (2020) is a solid introduction to the underlying theory of applied machine learning. For ecologists, Dietze (2017) is a comprehensive, and still highly readable, treatise on the problems associated to forecasting. The best way to decide on which book to read is often to look at the books that your colleagues have also read; being able to work through material collectively is useful, and knowing that you can practice the craft of data science within a community will make your learning more effective.

When reading this book, I encourage you to read the chapters in order. They have been designed to be read in order, because each chapter introduces the least possible amount of new concepts, but often requires to build on the previous chapters. This is particularly true of the second half of this book.

1.1. Core concepts in data science

1.1.1. EDA

1.1.2. Clustering and regression

1.1.3. Supervised and unsupervised

1.1.4. Training, testing, and validation

1.1.5. Transformations and feature engineering

1.2. An overview of the content

In [?@sec-clustering](#), we introduce some fundamental questions in data science, by working on the clustering of pixels in Landsat data. The point of this chapter is to question the way we think about data, and to start a discussion about an “optimal” model, hyper-parameters, and what a “good” model is.

In [?@sec-gradientdescent](#), we revisit well-trodden statistical ground, by fitting a linear model to linear data, but using gradient descent. This provides us with an opportunity to think about what a “fitted” model is, whether it is possible to learn too much from data, and why being able to think about predictions in the unit of our problem helps.

In [?@sec-crossvalidation](#), we start introducing one of the most important bit element of data science practice, in the form of cross-validation. We apply this technique to the prediction of plant phenology over a millenia, and think about the central question of “what kind of decision-making can we justify with a model”.

In [?@sec-classification](#), we introduce the task of classification, and spend a lot of time thinking about biases in predictions, which are acceptable, and which are not. We start building a model for the distribution of the Reindeer, which we will improve over a few chapters.

1. Introduction





In Chapter 2, we explore ways to perform variable selection, think of this task as being part of the training process, and introduce ideas related to dimensionality reduction. In Section 2.2, we discuss data leakage, where it comes from, and how to prevent it. This leads us to introducing the concept of data transformations as a model, which will establish some best practices we will keep on using throughout this book.

In ?@sec-tuning, we conclude story arcs that had been initiated in a few previous chapters, and explore training curves, the tuning of hyper-parameters, and moving-threshold classification. We provide the final refinements to our model of the Reindeer distribution.

In ?@sec-explanations, we will shift our attention from prediction to understanding, and explore techniques to quantify the importance of variables, as well as ways to visualize their contribution to the predictions. In doing so, we will introduce concepts of model interpretation and explainability.

In ?@sec-bagging, ...

1.3. A note on colors

Type	Meaning	Color
All	generic	
	no data	
Cross-validation	training	
	validation	

Type	Meaning	Color
Species range	testing	
	presence	
	absence	
Range change	loss	
	no change	
	gain	

In addition, there are three important color *palettes*. Information that is *sequential* is nature, which is to say it increases on a continuous scale without a logical midpoint, is rendered with these colors (from low to the left, to high values to the right):



The diverging palette is used for values that have a clear midpoint (usually values centered on 0). The midpoint will always correspond to the central color, and this palette is symmetrical:

1. Introduction



Finally, the categorical data are represented using the following palette:



1.4. Some rules about this book

When I started aggregating these notes, I decided on a series of four rules. No code, no simulated data, no long list of model, and above all, no *iris* dataset. In this section, I will go through *why* I decided to adopt these rules, and how it should change the way you interact with the book.

1.4.1. No code

This is, maybe, the most surprising rule, because data science *is* programming (in a sense). But sometimes there is so much focus on programming that we lose track of the other, important aspects of the practice of data science: abstractions, relationship with data, and domain knowledge.

This book *did* involve a lot of code. Specifically, this book was written using *Julia* (Bezanson *et al.* 2017), and every figure is generated by a notebook, and they are part of the material I use when teaching from this content in the classroom. But code is *not* a universal language, and unless you are really familiar with the language, code can obfuscate. I had no intention to write a *Julia* book (or an *R* book, or a *Python* book). The point is to

think about data science applied to ecological research, and I felt like it would be more inclusive to do this in a language agnostic way.

And finally, code rots. Code with more dependencies rots faster. It take a single change in the API of a package to break the examples, and then you are left with a very expensive monitor stand. With a few exceptions, the examples in this book do not use complicated packages either.

1.4.2. No simulated data

I have nothing against simulated data. I have, in fact, generated simulated data in many different contexts, for training or for research. But the limit of simulated is that we almost inevitably fail to include what makes real data challenging: noise, incomplete or uneven sampling, data representation artifacts. And so when it is time to work on real data, everything seems suddenly more difficult.

Simulated data have *immense* training value; but it is also important to engage with the imperfect actual data, as we will overwhelmingly apply the concepts from this book to them. For this reason, there are no simulated data in this book. Everything that is presented correspond to an actual use case that proceeds from a question we could reasonably ask in the context, paired with a dataset that could be used to answer this question.

1.4.3. No model zoo

My favorite machine learning package is *MLJ* (Blaom *et al.* 2020). When given a table of labels and a table of features, it will give back a series of models that match with these data. It speeds up the discovery of models considerably, and is generally a lot more informative than trying to read from a list of possible techniques. If I have questions about an algorithm from this list, I can start reading more documentation about how it works.

Reading a long enumeration of things is boring; unless it's sung by Yakko Warner, I'm not interested, and I refuse to inflict it on people. But more importantly, these enumerations of models often distract from thinking about the problem we want to solve in more abstract terms. I rarely wake up in the morning and think "oh boy I can't wait to train a SVM today"; chances are, my thought process will be closer to "I need to tell the mushroom people where I think the next good foraging locations will be". The rest, is implementation details.

1. Introduction

In fact, 90% of this book uses only two models: linear regression, and the Naïve Bayes Classifier. Some other models are involved in a few chapters, but these two models are breathtakingly simple, work surprisingly well, run fast, and can be tweaked to allow us to build deep intuitions about how machines learn. They are perfect for the classroom, and give us the freedom to spent most of our time thinking about how we interact with models, and why, and how we make methodological decisions.

1.4.4. No *iris* dataset

From a teaching point of view, the *iris* dataset is like hearing Smash Mouth in a movie trailer, in that it tells you two things with absolute certainty. First, that you are indeed watching a movie trailer. Second, that you could be watching Shrek instead. There are datasets out there that are *infinitely more* exciting to use than *iris*.

But there is a far more important reason not to use *iris*: eugenics.

Listen, we made it several hundred words in a text about quantitative techniques in life sciences without encountering a sad little man with racist ideas that academia decided to ignore because “he just contributed so much to the field, and these were different times, maybe we shouldn’t be so quick to judge?”. Ronald Aylmer Fisher, statistics’ most racist nerd, was such a man; and there are, of course, those who want to consider the possibility that you can be outrageously racist as long as you are an outstanding scientist (Bodmer *et al.* 2021).

The *iris* dataset was first published by Fisher (1936) in the *Annals of Eugenics* (so, there’s a bit of a red flag there already), and draws from several publications by Edgar Anderson, starting with Anderson (1928); Unwin & Kleinman (2021) have an interesting historiographic deep-dive into the correspondence between the two. Judging by the dates, you may think that Fisher was a product of his time. But this could not be further from the truth. Fisher was dissatisfied with his time, to the point where his contributions to statistics were done in service of his views, in order to provide the appearance of scientific rigor to his bigotry.

Fisher advocated for forced sterilization for the “defectives” (which he estimated at, oh, roughly 10% of the population), argued that not all races had equal capacity for intellectual and emotional development, and held a host of related opinions. There is no amount of contribution to science that pardon these views. Coming up with the idea of the null hypothesis does not even out lending “scientific” credibility to ideas whose logical

(and historical) conclusion is genocide. That Ronald Fisher is still described as a polymath and a genius is infuriating, and we should use every alternative to his work that we have.

Thankfully, there are alternatives!

The most broadly known alternative to the *iris* dataset is *penguins*, which was collected by ecologists (Gorman *et al.* 2014), and published as a standard dataset (Horst *et al.* 2020) so that we can train students without engaging with the “legacy” of eugenicists. The *penguins* dataset is also genuinely good! The classes are not so obviously separable, there are some missing data that reflect the reality of field work, and the data about sex and spatial location have been preserved, which increases the diversity of questions we can ask. We won’t use *penguins* either. It’s a fine dataset, but at this point there is little that we can write around it that would be new, or exciting. But if you want to apply some of the techniques in this book? Go *penguins*.

References

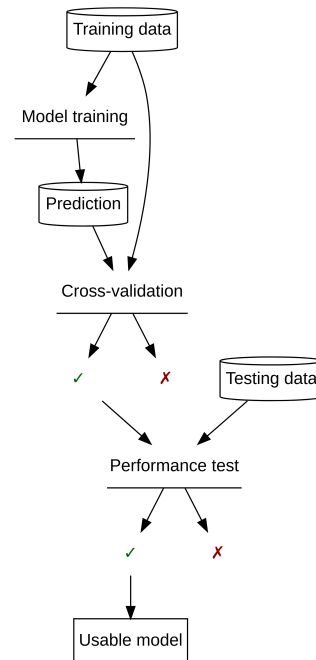
- Anderson, E. (1928). [The problem of species in the northern blue flags, *iris versicolor* L. And *iris virginica* L.](#) *Annals of the Missouri Botanical Garden*, 15, 241.
- Bezanson, J., Edelman, A., Karpinski, S. & Shah, V.B. (2017). [Julia: A Fresh Approach to Numerical Computing.](#) *SIAM Review*, 59, 65–98.
- Blaom, A., Kiraly, F., Lienart, T., Simillides, Y., Arenas, D. & Vollmer, S. (2020). [MLJ: A julia package for composable machine learning.](#) *Journal of Open Source Software*, 5, 2704.
- Bodmer, W., Bailey, R.A., Charlesworth, B., Eyre-Walker, A., Farewell, V., Mead, A., *et al.* (2021). [The outstanding scientist, R.A. Fisher: his views on eugenics and race.](#) *Heredity*, 126, 565–576.
- Deisenroth, M.P., Faisal, A.A. & Ong, C.S. (2020). [Mathematics for machine learning.](#)
- Dietze, M. (2017). [Ecological forecasting.](#)
- Fisher, R.A. (1936). [The Use Of Multiple Measurements In Taxonomic Problems.](#) *Annals of Eugenics*, 7, 179–188.
- Gorman, K.B., Williams, T.D. & Fraser, W.R. (2014). [Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins \(Genus *Pygoscelis*\).](#) *PLoS ONE*, 9, e90081.
- Horst, A.M., Hill, A.P. & Gorman, K.B. (2020). [Allisonhorst/palmerpenguins: v0.1.0.](#) Zenodo.
- Sulmont, E., Patitsas, E. & Cooperstock, J.R. (2019). [Can you teach me to machine learn?](#) *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*.

1. Introduction

Unwin, A. & Kleinman, K. (2021). [The Iris Data Set: In Search of the Source of *Virginica*](#). *Significance*, 18, 26–29.

Watt, J., Borhani, R. & Katsaggelos, A. (2020). [Machine learning refined](#).

Yau, N. (2015). [Visualize this](#).



Flowchart 1.1: An overview of the process of coming up with a usable model. The process of creating a model starts with a training dataset made of predictors and responses, which is used to train a model. This model is cross-validated on its training data, to estimate whether it can be fully retrained. The fully trained model is then applied to an independent testing dataset, and the evaluation of the performance determines whether it will be used.

2. Preparing features

In [?@sec-classification](#), we introduced a simple classifier trained on a dataset of presence and pseudo-absences of a species (*Sitta whiteheadi*), which we predicted using the mean annual temperature as well as the annual total precipitation. This choice of variables was motivated by our knowledge of the fact that most species tend to have some temperature and precipitation they are best suited to. But we can approach the exercise of selecting predictive variables in a far more formal way, and this will form the core of this chapter. Specifically, we will examine two related techniques: variable selection, and feature engineering.

There are two reasons to think about variable selection and feature engineering – first, the variables we have may not all be predictive for the specific problem we are trying to solve; second, the variables may not be expressed in the correct “way” to solve our problem. This calls for a joint approach of selecting and transforming features. Before we do anything to our features (transformation or selection), we will discuss the important problem of data leakage, and use this discussion to establish a framework to pick hyper-parameters of the model.

2.1. The problem: optimal set of BioClim variables for the Corsican nuthatch

The BioClim suite of environmental variables are 19 measurements derived from monthly recordings of temperature and precipitation. They are widely used in species distribution modeling, despite some spatial discontinuities due to the methodology of their reconstruction (Booth 2022); this is particularly true when working from the WorldClim version (Fick & Hijmans 2017), and not as problematic when using other data products like CHELSA (Karger *et al.* 2017).

2. Preparing features

The definitions of the 19 BioClim variables are given in Table 2.1. As we can see from this table, a number of variables are either derived from the same months, or calculated through direct (sometimes linear) combinations of one another. For this reason, and because there are 19 variables, this is a good dataset to evaluate the use of variable selection and transformation.

Table 2.1.: List of the 19 BioClim variables, including indications of their calculation. The model we used in ?@sec-classification used BIO1 and BIO12.

Layer	Description	Details
BIO1	Annual Mean Temperature	
BIO2	Mean Diurnal Range	Mean of monthly (max temp - min temp)
BIO3	Isothermality	BIO2/BIO7) ($\times 100$
BIO4	Temperature Seasonality	standard deviation $\times 100$
BIO5	Max Temperature of Warmest Month	
BIO6	Min Temperature of Coldest Month	
BIO7	Temperature Annual Range	BIO5-BIO6
BIO8	Mean Temperature of Wettest Quarter	
BIO9	Mean Temperature of Driest Quarter	
BIO10	Mean Temperature of Warmest Quarter	
BIO11	Mean Temperature of Coldest Quarter	
BIO12	Annual Precipitation	
BIO13	Precipitation of Wettest Month	
BIO14	Precipitation of Driest Month	
BIO15	Precipitation Seasonality	Coefficient of Variation
BIO16	Precipitation of Wettest Quarter	
BIO17	Precipitation of Driest Quarter	
BIO18	Precipitation of Warmest Quarter	
BIO19	Precipitation of Coldest Quarter	

This is true even when we are not transforming the variables! The identity function $f(x) = x$ is a transformation step that we can “train” (pretty easily, it turns out, as it has no parameters).

In this chapter, we will try to improve the model introduced in ?@sec-classification, by evaluating different methods to prepare our predictor variables. At this point, it is important to shift the way we think about the

model: it is not *just* the classifier that turns the features into the prediction, but it is in fact **the entire process of transforming raw data into predictions**. This includes the transformation of these raw data, and as we will illustrate throughout this chapter, this must come with changes in the way we think about what “training” means.

2.2. What is data leakage?

Data leakage is a concept that is, if you can believe it, grosser than it sounds.

The purpose of this section is to put the fear of data leakage in you, because it can, and most assuredly *will*, lead to bad models. This is to say, as we discussed in **sec-gradientdescent-trainedmodel**, models that do not adequately represent the underlying data or the relationships that exists within it, in part because we have built-in some biases into them. In turn, this can eventually lead to decreased explainability of the models, which erodes trust in their predictions (Amarasinghe *et al.* 2023). As illustrated by Stock *et al.* (2023), a large number of ecological applications of machine learning are particularly susceptible to data leakage, meaning that this should be a core point of concern for us.

2.2.1. Consequences of data leakage

We take data leakage so seriously because it is one of the top ten mistakes in applied machine learning (Nisbet *et al.* 2018). Data leakage happens information “leaks” from the training conditions to the evaluation conditions; in other words, when the model is evaluated after mistakenly being fed information that would not be available in real-life situations. Note that this definition of leakage is different from another notion, namely the loss of data availability over time (Peterson *et al.* 2018).

It is worth stopping for a moment to consider what these “real-life situations” are, and how they differ from the training of the model. Most of this difference can be summarized by the fact that when we are *applying* a model, we can start from the model *only*. Which is to say, the data that have been used for the training and validation of the model may have been lost, without changing the applicability of the model: it works on entirely new data, and on new data only. The legacy of the training and validation data is only found in the parameters of

2. Preparing features

the trained model. We have discussed this situation in [?@sec-crossvalidation-testing](#): the test of a model is conducted on data that have never been used for training, because we want to evaluate its performance in the conditions where it will be applied.

Because this is the behavior we want to simulate with a validation dataset, it is very important to fully disconnect the testing data from the rest of the data. We can illustrate this with an example. Let's say we want to work on a time series of population size, such as provided by the *BioTIME* project (Dornelas *et al.* 2018). One naïve approach would be to split this the time series at random into three datasets. We can use one to train the models, one to validate these models, and a last one for testing.

Congratulations! We have created data leakage! Because we are splitting our time series at random, the model will likely have been trained using data that date from *after* the start of the validation dataset. In other words: our model can peek into the future. This is highly unlikely to happen in practice, due to the ways the laws of physics work. A strategy that would prevent leakage would have been to pick a cut-off date to define the validation dataset, and then to decide how to deal with the training and testing sets.

2.2.2. Avoiding data leakage

The most common advice given in order to prevent data leakage is the “learn/predict separation” (Kaufman *et al.* 2011). Simply put, this means that whatever happens to the data used for training cannot be *simultaneously* applied to the data used for testing (or validation).

A counter-example where performing the transformation *before* the analysis is when the transformation is explicitly sought out as an embedding, where we want to predict the position of instances in the embedded space, as in *e.g.* Runghen *et al.* (2022).

Assume that we want to transform our data using a Principal Component Analysis (PCA; Pearson (1901)). Ecologists often think of PCA as a technique to explore data (Legendre & Legendre 2012), but it is so much more than that! PCA is a model, because we can learn, from the data, a series of weights (in the transformation matrix), which we can then apply to other datasets in order to project them in the space of the projection learned from the training data.

2.2. What is data leakage?

If we have a dataset \mathbf{X} , which we split into two components \mathbf{X}_0 for training, and \mathbf{X}_1 for validation, there are two ways to use a PCA to transform these data. The first is $\mathbf{T} = \mathbf{X}\mathbf{W}$, which uses the full dataset. When we predict the position of the validation data, we could use the transformation $\mathbf{T}_1 = \mathbf{X}_1\mathbf{W}$, but this would introduce data leakage: we have trained the transformation we apply to \mathbf{X}_1 using data that is already in \mathbf{X}_1 , and therefore we have not respected the learn/predict separation.

There is a biological argument against using all the data to learn transformations anyways. Assume that we are working on a specific area in space, but want to project our bioclimatic variables using a PCA before we do this. If we work in the high arctic, is the relationship between temperature and precipitation in the Amazonian rain forest, the Serengeti desert, and the Voses mountain relevant to our problem? Surely not! For this reason, it makes sense to limit the data we use to train the transformation to the data we would have used to train the model.

The second (correct) way to handle this situation is to perform our PCA using $\mathbf{T}_0 = \mathbf{X}_0\mathbf{W}_0$, which is to say, the weights of our PCA are derived *only* from the training data. In this situation, whenever we project the data in the validation set using $\mathbf{T}_1 = \mathbf{X}_1\mathbf{W}_0$, we respect the learn/predict separation: the transformation of \mathbf{X}_1 is entirely independent from the data contained in \mathbf{X}_1 .

There are a *lot* of peer-reviewed articles that introduce data leakage by applying PCA on bioclimatic data layers. Remember, common practices and good practices are not the same thing!

2.2.3. How to work in practice?

Although avoiding data leakage is a tricky problem, there is a very specific mindset we can adopt that goes a long way towards not introducing it in our analyses, and it is as follows: *every data transformation step is a modeling step that is part of the learning process*. We do not, for example, apply a PCA and train the model on the projected variables – we feed raw data into a model, the first step of which is to perform this PCA for us.

This approach works because **everything that can be represented as numbers is a model that can be trained**.

If you want to transform a variable using the z-score, this is a model! It has two parameters that you can learn from the data, μ (the average of the variable) and σ (its standard deviation). You can apply it to a data point y with $\hat{y} = (y - \mu)\sigma^{-1}$. Because this is a model, we need a dataset to learn these parameters from, and because we want to maintain the learn/predict separation, we will use the training dataset to get the values of μ_0 and σ_0 . This way, when we want to get the z-score of a new observation, for example from the testing dataset, we

2. *Preparing features*

can get it using $\hat{y}_1 = (y_1 - \mu_0)\sigma_0^{-1}$. The data transformation is entirely coming from information that was part of the training set.

One way to get the learn/predict transformation stupendously wrong is to transform our validation, testing, or prediction data using $\hat{y}_1 = (y_1 - \mu_1)\sigma_1^{-1}$. This can be easily understood with an example. Assume that the variable y_0 is the temperature in our training dataset. We are interested in making a prediction in a world that is 2 degrees hotter, uniformly, which is to say that for whatever value of y_0 , the corresponding data point we use for prediction is $y_1 = y_0 + 2$. If we take the z-score of this new value based on its own average and standard deviation, a temperature two degrees warmer in the prediction data will have the same z-score as its original value, or in other words, we have hidden the fact that there is a change in our predictors! Because we learn the correct prediction from the training data, we can only apply this prediction with the parameters derived from the training data.

Treating the data preparation step as a part of the learning process, which is to say that we learn every transformation on the training set, and retain this transformation as part of the prediction process, we are protecting ourselves against both data leakage *and* the hiding of relevant changes in our predictors.

2.3. Variable selection

2.3.1. The curse of dimensionality

The number of variables we use for prediction is the number of dimensions of a problem. It would be tempting to say that adding dimensions should improve our chances to find a feature alongside which the classes become linearly separable.

Alas.

The “curse of dimensionality” is the common term of everything breaking down when the dimensions of a problem increase. In our perspective, where we rely on the resemblance between features to make a prediction, increasing the dimensions of a problem means adding features, and it has important consequences on the distance between observations. Picture two points positioned at random on the unit interval: the average distance between them is $1/3$. If we add one dimension, keeping two points but turning this line into a cube,

the average distance would be about $1/2$. For a cube, about $2/3$. For n dimensions, we can figure out that the average distance grows like $\sqrt{n/6 + c}$, which is to say that when we add more dimensions, we make the average distance between two points go to infinity. This effect is also affecting ecological studies (e.g. Smith *et al.* 2017).

Therefore, we need to approach the problem of “which variables to use” with a specific mindset: we want a lot of information for our model, but not so much that the space in which the predictors exist turns immense. There are techniques for this.

2.3.2. Step-wise approaches to variable selection

In order to try and decrease the dimensionality of a problem, we can attempt to come up with a method to decide which variables to include, or to remove, from a model. This practice is usually called “stepwise” selection, and is the topic of *intense* debate in ecology, although several studies point to the fact that there is rarely a best technique to select variables (Murtaugh 2009), that the same data can usually be adequately described by competing models (WHITTINGHAM *et al.* 2006), and that classifiers can show high robustness to the inclusion of non-informative variables (Fox *et al.* 2017). Situations in which variable selection has been shown to be useful is the case of model transfer (Petitpierre *et al.* 2016), or (when informed by ecological knowledge), the demonstration that classes of variables had no measurable impact on model performance (Thuiller *et al.* 2004).

Why, so, should we select the variables we put in our models?

The answer is simple: we seek to solve a specific problem in an optimal way, where “optimal” refers to the maximization of a performance measure we decided upon *a priori*. In our case, this is the MCC. Therefore, an ideal set of predictors is the one that, given our cross-validation strategy, maximizes our measure of performance.

2.3.2.1. Forward selection

In forward selection, assuming that we have f features, we start by building f models, each using one feature. For example, using the BioClim variables, m_1 would be attempting to predict presences and absences based

2. Preparing features

only on temperature. Out of these models, we retain the variable given by $\text{argmax}_f \text{MCC}(m_f)$, where $\text{MCC}(m_f)$ is the average value of MCC for the f -th model on the validation datasets. This is the first variable we add to our set of selected variables. We then train $f - 1$ models, and then again add the variable that leads to the best possible *increase* in the average value of the MCC. When we cannot find a remaining variable that would increase the performance of the model, we stop the process, and return the optimal set of variables. Forward selection can be constrained by, instead of starting from variables one by one, starting from a pre-selected set of variables that will always be included in the model.

There are two important things to consider here. First, the set of variables is only optimal under the assumptions of the stepwise selection process: the first variable is the one that boosts the predictive value of the model the most *on its own*, and the next variables *in the context of already selected variables*. Second, the variables are evaluated on the basis of their ability to *improve the performance of the model*; this does not imply that they are relevant to the ecological processes happening in the dataset. Inferring mechanisms on the basis of variable selection is foolish (Tredennick *et al.* 2021).

2.3.2.2. Backward selection

The opposite of forward selection is backward selection, in which we start from a complete set of variables, then remove the one with the *worst* impact on model performance, and keep proceeding until we cannot remove a variable without making the model worse. The set of variables that remain will be the optimal set of variables. In almost no cases will forward and backward selection agree on which set of variables is the best – we have to settle this debate by either picking the model with the least parameters (the most parsimonious), or the one with the best performance.

Why not evaluate all the combination of variables?

Keep in mind that we do not know the number of variables we should use; therefore, for the 19 BioClim variables, we would have to evaluate $\sum_f \binom{19}{f}$, which turns out to be an *immense* quantity (for example, $\binom{19}{9} = 92378$). For this reason, a complete enumeration of all variable combinations would be extremely wasteful.

2.3.3. Removal of colinear variables

Co-linearity of variables is challenging for all types of ecological models (Graham 2003). In the case of species distribution models (De Marco & Nóbrega 2018), the variables are expected to be strongly auto-correlated, both because they have innate spatial auto-correlation, and because they are derived from a smaller set of raw data (Dormann *et al.* 2012). For this reason, it is a good idea to limit the number of colinear variables.

THIS PARAGRAPH IS NOT FINISHED

2.4. Multivariate transformations

2.4.1. PCA-based transforms

Principal Component Analysis (PCA) is one of the most widely used multi-variate techniques in ecology, and is a very common technique to prepare variables in applied machine learning. One advantage of PCA is that it serves both as a way to remove colinearity, in that the principal components are orthogonal, and as a way to reduce the dimensionality of the problem as long as we decide on a threshold on the proportion of variance explained, and only retain the number of principal components needed to reach this threshold. For applications where the features are high-dimensional, PCA is a well established method to reduce dimensionality *and* extract more information in the selected principal components (Howley *et al.* 2005). In PCA, the projection matrix \mathbf{P} is applied to the data using $\mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}})$, where \mathbf{x} is the feature matrix with means $\bar{\mathbf{x}}$. Typically, the dimensions of \mathbf{P} are *lower* than the dimensions of \mathbf{x} , resulting in fewer dimensions to the problem. Cutoffs on the dimensions of \mathbf{P} are typically expressed as a proportion of the overall variance maintained after the projection. Variants of PCA include kernel PCA (Schölkopf *et al.* 1998), using a higher-dimensional space to improve the separability of classes, and probabilistic PCA (Tipping & Bishop 1999), which relies on modeling the data within a latent space with lower dimensionality.

2. Preparing features

2.4.2. Whitening transforms

Another class of potentially very useful data transformations is whitening transforms, which belongs to the larger category of decorrelation methods. These methods do not perform any dimensionality reduction, but instead remove the covariance in the datasets. Whitening has proven to be particularly effective at improving the predictive ability of models applied to data with strong covariance structure (Koivunen & Kostinski 1999). In essence, given a matrix of features \mathbf{x} , with averages μ and covariance \mathbf{C} , a whitening transform \mathbf{W} is *one of the matrices* that satisfies $\mathbf{W}^T \mathbf{C} \mathbf{W} = \mathbf{I}$.

In other words, the whitening transform results in a new set of features with unit variance and no covariance: the dimensionality of the problem remains the same but the new random variables are independent. Given any dataset with covariance matrix \mathbf{C} , if any \mathbf{W} is a whitening transform, then so to are any matrices $\mathbf{W}\mathbf{R}$ where \mathbf{R} performs a rotation with $\mathbf{R}^T \mathbf{R} = \mathbf{I}$. The optimal whitening transform can be derived through a variety of ways (see e.g. Kessy *et al.* 2018). The whitening transform is applied to the input vector using $\mathbf{W}^T (\mathbf{x} - \mu)$: this results in new random variables that have a mean of 0, and unit variance. The new input vector after the transformation is therefore an instance of “white noise” (Vasseur & Yodzis 2004).

When performing Whitening transformation, the first variable in the new space will usually have pretty high correlation to the first raw variable. But because the purpose of Whitening is to remove the covariance (and ensure unit variance), this gets less and less true as we increase the rank of the variables. By the time the last selected variable is reached, we should expect it to be almost purely noise.

2.5. Application: optimal variables for Corsican nuthatch

Before we start, we can re-establish the baseline performance of the model from **?@sec-classification**. In this (and the next) chapters, we will perform k-folds cross-validation (see **?@sec-crossvalidation-kfolds** for a refresher), using $k = 15$. This strategy gives an average MCC of 0.727, which represents our “target”: any model with a higher MCC will be “better” according to our criteria.

In a sense, this initial model was *already* coming from a variable selection process, only we did not use a quantitative criteria to include variables. And so, it is a good idea to evaluate how our model performed, relative to

a model including *all* the variables. Running the NBC again using all 19 BioClim variables from Table 2.1, we get an average MCC on the validation data of 0.748. This is a small increase, but an increase nevertheless – our dataset had information that was not captured by temperature and precipitation. But this model with all the variables most likely includes extraneous information that does not help, or even hinders, the predictive ability of our model. Therefore, there is probably a better version of the model somewhere, that uses the optimal set of variables, potentially with the best possible transformation applied to them.

In this section, we will start by evaluating the efficiency of different approaches to variable selection, then merge selection and transformation together to provide a model that is optimal with regards to the training data we have (the workflow is outlined in [?@fig-predictors-workflow](#)). In order to evaluate the model, we will maintain the use of the MCC; in addition, we will report the PPV and NPV (like in [?@sec-classification](#)), as well as the accuracy and True-Skill Statistic (TSS). The TSS is defined as the sum of true positive and true negative rates, minus one, and is an alternative measure to the MCC (although it is more sensitive to some biases). Although several authors have advocated for the use of TSS (ALLOUCHE *et al.* 2006), Leroy *et al.* (2018) have an interesting discussion of how the TSS is particularly sensitive to issues in the quality of (pseudo) absence data. For this reason, and based on the literature we covered in [?@sec-classification](#), there is no strong argument against using MCC as our selection measure.

To prevent the risk of interpreting the list of variables that have been retained by the model, we will *not* make a list of which they are (yet). This is because, in order to discuss the relative importance of variables, we need to introduce a few more concepts and techniques, which will not happen until [?@sec-explanations](#); at this point, we will revisit the list of variables identified during this chapter, and compare their impact on model performance to their actual importance in explaining predictions.

In [?@sec-tuning](#), we will revisit the question of how the MCC is “better”, and spend more time evaluating alternatives. For now, we can safely *assume* that MCC is the best.

2.5.1. Variable selection

We will perform four different versions of stepwise variable selection. Forward, forward from a pre-selected set of two variables (temperature and precipitation), backward, and based on the Variance Inflation Factor (with a cutoff of 10). The results are presented in Table 2.2.

2. Preparing features

Table 2.2.: Consequences of different variable selection approaches on the performance of the model, as evaluated by the MCC averaged over the validation datasets. The highest MCC value, corresponding to the best model, is in bold.

Model	Variables	MCC
?@sec-classification baseline	1,12	0.727
All variables		0.748
Fwd. sel.	8,4,7,6,15	0.777
Constr. sel.	1,12,10,3,8	0.773
Backw. sel.	1,2,3,5,6,7,8,9,10,11,14,15,17	0.775
Var. infl. fac.	2,6,18	0.754

The best model is given by forward selection, although backwards selection also gives a very close performance. At this point, we may decide to keep these two strategies, and evaluate the effect of different transformations of the data.

2.5.2. Variable transformation

Based on the results from Table 2.2, we retain forward and backwards selection as our two stepwise selection methods, and now apply an additional transformation (as in ?@fig-predictors-workflow) to the subset of the variables. The results are presented in Table 2.3. Based on these results, and using the MCC as the criteria for the “best” model, we see that combining forward selection with a whitening transform gives the best predictive performance. Note that the application of a transformation *does* change the result of variable selection, as evidences by the fact that the number of retained variables changes when we apply a transformation.

Table 2.3.: Model performance when coupling variable selection with variable transformation.

Variable selection	Transformation variables	Nb.	Latent var.	Var. expl.	MCC (val.)	MCC (train)
?@sec-classification baseline		2			0.727	0.729
	PCA	2	1	1.00	0.476	0.478
Fwd. sel.		5			0.777	0.780
	PCA	4	4	1.00	0.796	0.797
Backw. sel.		13			0.775	0.777
	PCA	15	3	0.99	0.773	0.773
Constr. sel.		5			0.773	0.774
	PCA	6	2	1.00	0.484	0.485

2.5.3. Variable importance

Before moving on, it is worth taking a moment to think about what we have done so far. We have optimized two components of our model: the transformation of the raw data before they are fed to the classifier (we use a PCA for this step), and then (or simultaneously) the list of variables which we put into the model. In **?@sec-classification**, we had assumed that temperature and precipitation were important variables based on decades of ecological folklore (or, as we say, “expert knowledge”). But what about the list of variables currently retained in this model? Although it may be tempting to come up with *ad hoc* explanations for why they matter, we can rather ask the question of “how much do these variables actually matter?”.

The process we will adopt is simple: we train a model on a training set \mathbf{X}_t , and measure its performance on a validation set \mathbf{X}_v . The performance of the model on the validation we note p_v . Because we want to avoid re-training our model (because we may not have access to the training data, in practice), the only thing we can act on is the validation set. The process we will use is to identify a feature j , and then across the entire validation set, shuffle its j -th column. At this point, we can re-apply our model to this shuffled validation set

2. Preparing features

(let's note it $\mathbf{X}_v^{(j)}$), and measure the performance of the model on this perturbed dataset as $p_v^{(j)}$. The impact of shuffling the variable j is given as $m_j = \|p_v - p_v^{(j)}\|$.

We can repeat this process a number of times (because we are relying on a shuffle, so it is better to perform this process many times over), and report the importance of variable j as \bar{m}_j . This importance is expressed on the scale of the measure of performance we use. When it is large, it means that shuffling the variable j has a very large impact on the performance of the model: it is important to know the correct value of this feature to make the “right” prediction. Because the absolute score \bar{m}_j is not very informative, we can instead report a *relative* importance of a variable, which is simply $\bar{m}_j / \sum_{i \in \text{var.}} \bar{m}_i$. These will sum to one, and can be interpreted as the proportional importance of the selected variables.

But what does the process of shuffling simulate? Low quality data, sampling, wrong information, loss of information etc

2.5.4. Model selection and output

In Table 2.2 and Table 2.3, we have evaluated a series of several modeling strategies, defined by a variable selection and transformation technique. Using the MCC as our reference for what constitutes the best model, we can now apply the model to the relevant set of predictors, in order to see how these refinements result in a new predicted range for the species.

In Figure 2.2, we see how the use of a PCA turns co-linear variables into a much more manageable set of points. Although we can still think of our model in terms of the untransformed predictors we feed it, the classifier will only attempt to learn parameters from the projected data. These results are presented in Figure 2.3.

2.6. The Rashomon effect

The Rashomon effect is defined as the fact that, given a single problem, we can find many models that will achieve, on average, the same performance (or the same loss). In other words, the same event (the prediction we want to make) can often admit several seemingly reasonable yet incompatible interpretations (just like in

the movie usually invoked as an example of the Rashomon effect: *Hoodwinked!*). So far, we have assumed that we were able to pick the *best* model, but the *best* is often a relatively minor improvement over the second-best model.

This chapter is a good illustration of the Rashomon effect. Most models in Table 2.3 have a very similar performance on the same problem. We know this because we evaluated the performance of these models on the *exact* same folds, training, and validation points. By picking the best model, we had to discard a large number of models that were, all things considered, similarly good.

This is not really a problem if all we care about is “making a good prediction”. Even then, a similar loss or MCC score can be achieved through different predictions once mapped out. But where the Rashomon effect is particularly problematic is when we start attempting to provide *explanations* for the predictions. All the best models (under different variable selection and data transformation schemes) had selected different features. What this means is that, if we attempt to explain a specific prediction with regards to, for example, annual precipitation, this task is only meaningful in a model that uses annual precipitation as a feature.

The Rashomon effect manifests at all scales of the machine learning process, and we will see instances of it again in the following chapters.

2.7. Conclusion

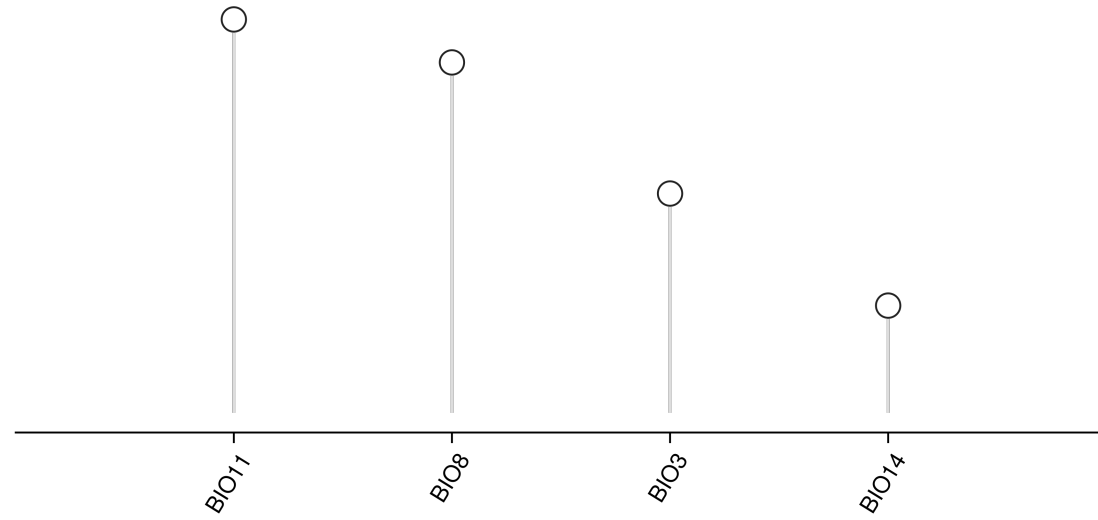
In this chapter, we have discussed the issues with dimensionality and data leakage, and established a methodology to reduce the number of dimensions (and possibly re-project the variables) while maintaining the train/predict separation. This resulted in a model whose performance (as evaluated using the MCC) increased quite significantly, which resulted in the predicted range of *Sitta whiteheadi* changing in space.

In **?@sec-tuning**, we will finish to refine this model, by considering that the NBC is a probabilistic classifier, and tuning various hyper-parameters of the model using learning curves and thresholding. This will result in the final trained model, the behavior of which we will explore in **?@sec-explanations**, to understand *how* the model makes predictions.

We will talk about the Rashomon effect again in **?@sec-tuning**, in **?@sec-bagging**, and further in **?@sec-counterfactuals**.

2. *Preparing features*

Figure 2.1.: Relative importance of the variables selected as part of the best model. The importance of the variables has been measured by comparing their performance on validation sets before and after shuffling the column of the feature matrix corresponding to the variable to test.



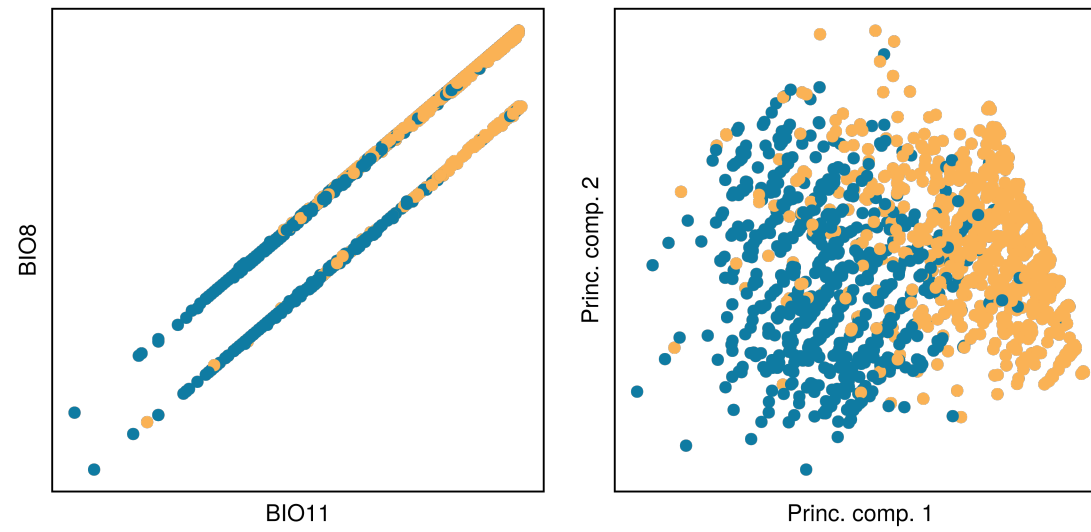
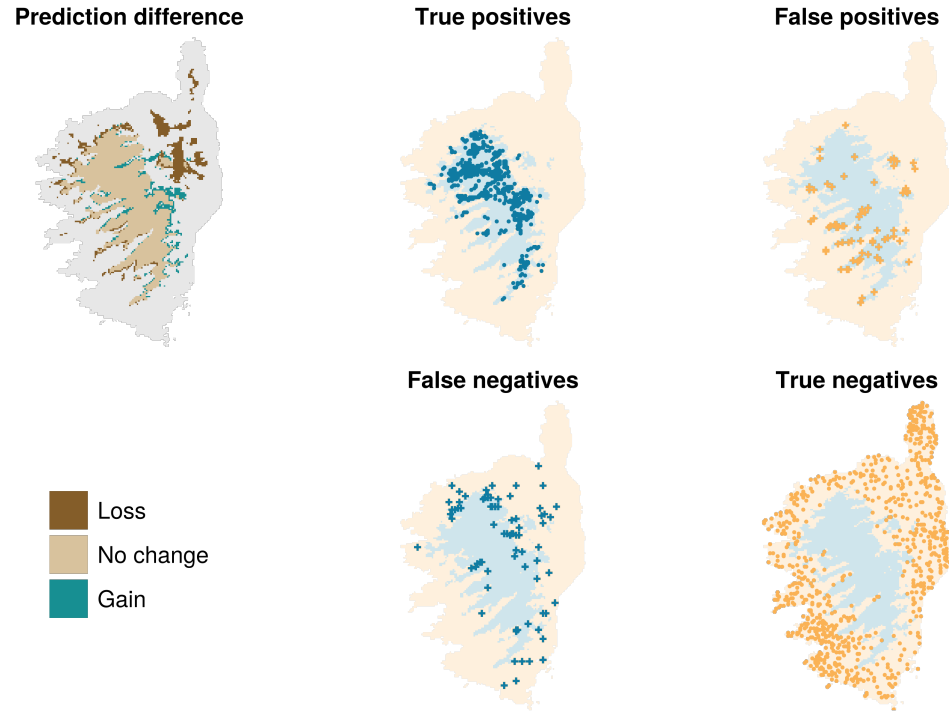


Figure 2.2.: orig var vs. their transformation

2. *Preparing features*

Figure 2.3.: Consequences of different variable transformations on the predicted range of *Sitta whiteheadi*. Note that the small area of predicted presence in the Cap Corse (the Northern tip) has disappeared with the new set of variables and their optimal transformation.



References

- ALLOUCHE, O., TSOAR, A. & KADMON, R. (2006). [Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic \(TSS\)](#). *Journal of Applied Ecology*, 43, 1223–1232.
- Amarasinghe, K., Rodolfa, K.T., Lamba, H. & Ghani, R. (2023). [Explainable machine learning for public policy: Use cases, gaps, and research directions](#). *Data & Policy*, 5.
- Booth, T.H. (2022). [Checking bioclimatic variables that combine temperature and precipitation data before their use in species distribution models](#). *Austral Ecology*, 47, 1506–1514.
- De Marco, P. & Nóbrega, C.C. (2018). [Evaluating collinearity effects on species distribution models: An approach based on virtual species simulation](#). *PLOS ONE*, 13, e0202403.
- Dormann, C.F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., *et al.* (2012). [Collinearity: a review of methods to deal with it and a simulation study evaluating their performance](#). *Ecography*, 36, 27–46.
- Dornelas, M., Antão, L.H., Moyes, F., Bates, A.E., Magurran, A.E., Adam, D., *et al.* (2018). [BioTIME: A database of biodiversity time series for the Anthropocene](#). *Global Ecology and Biogeography*, 27, 760–786.
- Fick, S.E. & Hijmans, R.J. (2017). [WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas](#). *International Journal of Climatology*, 37, 4302–4315.
- Fox, E.W., Hill, R.A., Leibowitz, S.G., Olsen, A.R., Thornbrugh, D.J. & Weber, M.H. (2017). [Assessing the accuracy and stability of variable selection methods for random forest modeling in ecology](#). *Environmental Monitoring and Assessment*, 189.
- Graham, M.H. (2003). [CONFRONTING MULTICOLLINEARITY IN ECOLOGICAL MULTIPLE REGRESSION](#). *Ecology*, 84, 2809–2815.
- Howley, T., Madden, M.G., O’Connell, M.-L. & Ryder, A.G. (2005). [The effect of principal component analysis on machine learning accuracy with high dimensional spectral data](#). Springer London, pp. 209–222.
- Karger, D.N., Conrad, O., Böhner, J., Kawohl, T., Kreft, H., Soria-Auza, R.W., *et al.* (2017). [Climatologies at high resolution for the earth’s land surface areas](#). *Scientific Data*, 4.
- Kaufman, S., Rosset, S. & Perlich, C. (2011). [Leakage in data mining](#). *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*.
- Kessy, A., Lewin, A. & Strimmer, K. (2018). [Optimal Whitening and Decorrelation](#). *The American Statistician*, 72, 309–314.
- Koivunen, A.C. & Kostinski, A.B. (1999). [The Feasibility of Data Whitening to Improve Performance of Weather Radar](#). *Journal of Applied Meteorology*, 38, 741–749.

2. Preparing features

- Legendre, P. & Legendre, L. (2012). *Numerical ecology*. Developments in environmental modelling. Third English edition. Elsevier, Oxford, UK.
- Leroy, B., Delsol, R., Hugueny, B., Meynard, C.N., Barhoumi, C., Barbet-Massin, M., *et al.* (2018). [Without quality presence-absence data, discrimination metrics such as TSS can be misleading measures of model performance](#). *Journal of Biogeography*, 45, 1994–2002.
- Murtaugh, P.A. (2009). [Performance of several variable-selection methods applied to real ecological data](#). *Ecology Letters*, 12, 1061–1068.
- Nisbet, R., Miner, G., Yale, K., Elder, J.F. & Peterson, A.F. (2018). *Handbook of statistical analysis and data mining applications*. Second edition. Academic Press, London.
- Pearson, K. (1901). [LIII. On lines and planes of closest fit to systems of points in space](#). *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2, 559–572.
- Peterson, A.T., Asase, A., Canhos, D., Souza, S. de & Wieczorek, J. (2018). [Data leakage and loss in biodiversity informatics](#). *Biodiversity Data Journal*, 6.
- Petitpierre, B., Broennimann, O., Kueffer, C., Daehler, C. & Guisan, A. (2016). [Selecting predictors to maximize the transferability of species distribution models: lessons from cross-continental plant invasions](#). *Global Ecology and Biogeography*, 26, 275–287.
- Runghen, R., Stouffer, D.B. & Dalla Riva, G.V. (2022). [Exploiting node metadata to predict interactions in bipartite networks using graph embedding and neural networks](#). *Royal Society Open Science*, 9.
- Schölkopf, B., Smola, A. & Müller, K.-R. (1998). [Nonlinear Component Analysis as a Kernel Eigenvalue Problem](#). *Neural Computation*, 10, 1299–1319.
- Smith, M.L., Ruffley, M., Espíndola, A., Tank, D.C., Sullivan, J. & Carstens, B.C. (2017). [Demographic model selection using random forests and the site frequency spectrum](#). *Molecular Ecology*, 26, 4562–4573.
- Stock, A., Gegr, E.J. & Chan, K.M.A. (2023). [Data leakage jeopardizes ecological applications of machine learning](#). *Nature Ecology & Evolution*.
- Thuiller, W., Araújo, M.B. & Lavorel, S. (2004). [Do we need land-cover data to model species distributions in Europe?](#) *Journal of Biogeography*, 31, 353–361.
- Tipping, M.E. & Bishop, C.M. (1999). [Probabilistic Principal Component Analysis](#). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61, 611–622.
- Tredennick, A.T., Hooker, G., Ellner, S.P. & Adler, P.B. (2021). [A practical guide to selecting models for exploration, inference, and prediction in ecology](#). *Ecology*, 102.
- Vasseur, D.A. & Yodzis, P. (2004). [THE COLOR OF ENVIRONMENTAL NOISE](#). *Ecology*, 85, 1146–1152.

References

WHITTINGHAM, M.J., STEPHENS, P.A., BRADBURY, R.B. & FRECKLETON, R.P. (2006). [Why do we still use stepwise modelling in ecology and behaviour?](#) *Journal of Animal Ecology*, 75, 1182–1189.

A. Instructor notes

References

- ALLOUCHE, O., TSOAR, A. & KADMON, R. (2006). [Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic \(TSS\)](#). *Journal of Applied Ecology*, 43, 1223–1232.
- Amarasinghe, K., Rodolfa, K.T., Lamba, H. & Ghani, R. (2023). [Explainable machine learning for public policy: Use cases, gaps, and research directions](#). *Data & Policy*, 5.
- Anderson, E. (1928). [The problem of species in the northern blue flags, iris versicolor l. And iris virginica l.](#) *Annals of the Missouri Botanical Garden*, 15, 241.
- Bezanson, J., Edelman, A., Karpinski, S. & Shah, V.B. (2017). [Julia: A Fresh Approach to Numerical Computing](#). *SIAM Review*, 59, 65–98.
- Blaom, A., Kiraly, F., Lienart, T., Simillides, Y., Arenas, D. & Vollmer, S. (2020). [MLJ: A julia package for composable machine learning](#). *Journal of Open Source Software*, 5, 2704.
- Bodmer, W., Bailey, R.A., Charlesworth, B., Eyre-Walker, A., Farewell, V., Mead, A., *et al.* (2021). [The outstanding scientist, R.A. Fisher: his views on eugenics and race](#). *Heredity*, 126, 565–576.
- Booth, T.H. (2022). [Checking bioclimatic variables that combine temperature and precipitation data before their use in species distribution models](#). *Austral Ecology*, 47, 1506–1514.
- De Marco, P. & Nóbrega, C.C. (2018). [Evaluating collinearity effects on species distribution models: An approach based on virtual species simulation](#). *PLOS ONE*, 13, e0202403.
- Deisenroth, M.P., Faisal, A.A. & Ong, C.S. (2020). [Mathematics for machine learning](#).
- Desai, J., Watson, D., Wang, V., Taddeo, M. & Floridi, L. (2022). [The epistemological foundations of data science: a critical review](#). *Synthese*, 200.
- Dietze, M. (2017). [Ecological forecasting](#).
- Dormann, C.F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., *et al.* (2012). [Collinearity: a review of methods to deal with it and a simulation study evaluating their performance](#). *Ecography*, 36, 27–46.

- Dornelas, M., Antão, L.H., Moyes, F., Bates, A.E., Magurran, A.E., Adam, D., *et al.* (2018). [BioTIME: A database of biodiversity time series for the Anthropocene](#). *Global Ecology and Biogeography*, 27, 760–786.
- Ellwood, E.R., Sessa, J.A., Abraham, J.K., Budden, A.E., Douglas, N., Guralnick, R., *et al.* (2019). [Biodiversity Science and the Twenty-First Century Workforce](#). *BioScience*, 70, 119–121.
- Fick, S.E. & Hijmans, R.J. (2017). [WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas](#). *International Journal of Climatology*, 37, 4302–4315.
- Fisher, R.A. (1936). [The Use Of Multiple Measurements In Taxonomic Problems](#). *Annals of Eugenics*, 7, 179–188.
- Fox, E.W., Hill, R.A., Leibowitz, S.G., Olsen, A.R., Thornbrugh, D.J. & Weber, M.H. (2017). [Assessing the accuracy and stability of variable selection methods for random forest modeling in ecology](#). *Environmental Monitoring and Assessment*, 189.
- Gorman, K.B., Williams, T.D. & Fraser, W.R. (2014). [Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins \(Genus *Pygoscelis*\)](#). *PLoS ONE*, 9, e90081.
- Graham, M.H. (2003). [CONFRONTING MULTICOLLINEARITY IN ECOLOGICAL MULTIPLE REGRESSION](#). *Ecology*, 84, 2809–2815.
- Horst, A.M., Hill, A.P. & Gorman, K.B. (2020). [Allisonhorst/palmerpenguins: v0.1.0](#). Zenodo.
- Howley, T., Madden, M.G., O’Connell, M.-L. & Ryder, A.G. (2005). [The effect of principal component analysis on machine learning accuracy with high dimensional spectral data](#). Springer London, pp. 209–222.
- Karger, D.N., Conrad, O., Böhrner, J., Kawohl, T., Kreft, H., Soria-Auza, R.W., *et al.* (2017). [Climatologies at high resolution for the earth’s land surface areas](#). *Scientific Data*, 4.
- Kaufman, S., Rosset, S. & Perlich, C. (2011). [Leakage in data mining](#). *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*.
- Kessy, A., Lewin, A. & Strimmer, K. (2018). [Optimal Whitening and Decorrelation](#). *The American Statistician*, 72, 309–314.
- Koivunen, A.C. & Kostinski, A.B. (1999). [The Feasibility of Data Whitening to Improve Performance of Weather Radar](#). *Journal of Applied Meteorology*, 38, 741–749.
- Legendre, P. & Legendre, L. (2012). *Numerical ecology*. Developments in environmental modelling. Third English edition. Elsevier, Oxford, UK.
- Leroy, B., Delsol, R., Hugueny, B., Meynard, C.N., Barhoumi, C., Barbet-Massin, M., *et al.* (2018). [Without quality presence–absence data, discrimination metrics such as TSS can be misleading measures of model performance](#). *Journal of Biogeography*, 45, 1994–2002.
- Murtaugh, P.A. (2009). [Performance of several variable-selection methods applied to real ecological data](#). *Ecol-*

- ogy Letters, 12, 1061–1068.
- Nisbet, R., Miner, G., Yale, K., Elder, J.F. & Peterson, A.F. (2018). *Handbook of statistical analysis and data mining applications*. Second edition. Academic Press, London.
- Pearson, K. (1901). [LIII. On lines and planes of closest fit to systems of points in space](#). *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2, 559–572.
- Peterson, A.T., Asase, A., Canhos, D., Souza, S. de & Wieczorek, J. (2018). [Data leakage and loss in biodiversity informatics](#). *Biodiversity Data Journal*, 6.
- Petitpierre, B., Broennimann, O., Kueffer, C., Daehler, C. & Guisan, A. (2016). [Selecting predictors to maximize the transferability of species distribution models: lessons from cross-continental plant invasions](#). *Global Ecology and Biogeography*, 26, 275–287.
- Runghen, R., Stouffer, D.B. & Dalla Riva, G.V. (2022). [Exploiting node metadata to predict interactions in bipartite networks using graph embedding and neural networks](#). *Royal Society Open Science*, 9.
- Schölkopf, B., Smola, A. & Müller, K.-R. (1998). [Nonlinear Component Analysis as a Kernel Eigenvalue Problem](#). *Neural Computation*, 10, 1299–1319.
- Smith, M.L., Ruffley, M., Espíndola, A., Tank, D.C., Sullivan, J. & Carstens, B.C. (2017). [Demographic model selection using random forests and the site frequency spectrum](#). *Molecular Ecology*, 26, 4562–4573.
- Stock, A., Gregr, E.J. & Chan, K.M.A. (2023). [Data leakage jeopardizes ecological applications of machine learning](#). *Nature Ecology & Evolution*.
- Sulmont, E., Patitsas, E. & Cooperstock, J.R. (2019). [Can you teach me to machine learn?](#) *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*.
- Thuiller, W., Araújo, M.B. & Lavorel, S. (2004). [Do we need land-cover data to model species distributions in Europe?](#) *Journal of Biogeography*, 31, 353–361.
- Tipping, M.E. & Bishop, C.M. (1999). [Probabilistic Principal Component Analysis](#). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61, 611–622.
- Tredennick, A.T., Hooker, G., Ellner, S.P. & Adler, P.B. (2021). [A practical guide to selecting models for exploration, inference, and prediction in ecology](#). *Ecology*, 102.
- Tuia, D., Kellenberger, B., Beery, S., Costelloe, B.R., Zuffi, S., Risse, B., et al. (2022). [Perspectives in machine learning for wildlife conservation](#). *Nature Communications*, 13, 792.
- Unwin, A. & Kleinman, K. (2021). [The Iris Data Set: In Search of the Source of *Virginica*](#). *Significance*, 18, 26–29.
- Vasseur, D.A. & Yodzis, P. (2004). [THE COLOR OF ENVIRONMENTAL NOISE](#). *Ecology*, 85, 1146–1152.
- Watt, J., Borhani, R. & Katsaggelos, A. (2020). [Machine learning refined](#).

A. Instructor notes

WHITTINGHAM, M.J., STEPHENS, P.A., BRADBURY, R.B. & FRECKLETON, R.P. (2006). [Why do we still use stepwise modelling in ecology and behaviour?](#) *Journal of Animal Ecology*, 75, 1182–1189.

Yau, N. (2015). [Visualize this](#).

Index

Interpretability, [29](#)

Rashomon effect, [29](#)