

An abstract graphic on a blue background. It features a white line-art brain in the upper right quadrant. From the brain, a complex network of white lines, resembling a circuit board or neural pathways, extends across the entire page. These lines branch out, connect to small white circles, and form various geometric shapes, creating a sense of connectivity and technology.

Responsible use of AI for public policy: Data science toolkit

Felipe González
Teresa Ortiz
Roberto Sánchez Ávalos

Responsible use of AI for public policy:

Data science toolkit

Felipe González, Teresa Ortiz y Roberto Sánchez Ávalos

<https://www.iadb.org/>

Copyright © 2020 Banco Interamericano de Desarrollo. Esta obra se encuentra sujeta a una licencia Creative Commons IGO 3.0 Reconocimiento-NoComercial-SinObrasDerivadas (CC-IGO 3.0 BY-NC-ND) (<https://creativecommons.org/licenses/by-nc-nd/3.0/igo/legalcode>) y puede ser reproducida para cualquier uso no-comercial otorgando el reconocimiento respectivo al BID. No se permiten obras derivadas.

Cualquier disputa relacionada con el uso de las obras del BID que no pueda resolverse amistosamente se someterá a arbitraje de conformidad con las reglas de la CNUDMI (UNCITRAL). El uso del nombre del BID para cualquier fin distinto al reconocimiento respectivo y el uso del logotipo del BID, no están autorizados por esta licencia CC-IGO y requieren de un acuerdo de licencia adicional.

Note que el enlace URL incluye términos y condiciones adicionales de esta licencia.

Las opiniones expresadas en esta publicación son de los autores y no necesariamente reflejan el punto de vista del Banco Interamericano de Desarrollo, de su Directorio Ejecutivo ni de los países que representa.





Inter-American Development Bank – Social Sector

The Social Sector is a multidisciplinary team whose actions are based on the conviction that investing in people can improve their lives and overcome development challenges in Latin America and the Caribbean. Together with the countries of the region, the Social Sector formulates public policy solutions to reduce poverty and improve the provision of education, work, social protection, and health services. The objective is to build a more productive region with equal opportunities for men and women, and with greater inclusion of the most vulnerable groups. For more information, see www.iadb.org/en/about-us/departments/scl.



Inter-American Development Bank – IDB Lab

IDB Lab is the innovation laboratory of the IDB Group where financing, knowledge, and connections are mobilized to catalyze innovation oriented towards inclusion in Latin America and the Caribbean. For IDB Lab, innovation is a powerful tool that can transform the region by creating unprecedented opportunities for populations in vulnerable situations due to their economic, social, and environmental conditions. For more information see <https://bidlab.org/>.



Organisation for Economic Co-operation and Development – OECD

The OECD is an international organisation that works to build better policies for better lives. Our goal is to shape policies that foster prosperity, equality, opportunity and well-being for all.

Together with governments, policy makers and citizens, we work on establishing evidence-based international standards and finding solutions to a range of social, economic and environmental challenges. One example of standard-setting are the OECD Principles on Artificial Intelligence (AI) that are the first such principles adopted by governments. The principles promote AI that is innovative and trustworthy and that respects human rights and democratic values.



OECD.AI Policy Observatory

The [OECD.AI](#) Policy Observatory is an inclusive hub for public policy on AI. It helps countries encourage, nurture and monitor the development and use of trustworthy AI.

Used by policy makers and other stakeholders in over 170 countries, OECD.AI has become a recognised centre for policy-oriented evidence, debate and guidance, supported by strong partnerships with actors from all stakeholder groups and with other international organisations. It provides evidence-based analysis on AI.

OECD.AI is a unique source of [real-time data](#) and visualisations on AI developments. It also contains a [database of AI policies](#) from over 60 countries that allows governments to compare policy responses and develop good practices. OECD.AI measures our collective progress towards trustworthy AI; its [network of experts](#) and the “[AI Wonk](#)” blog facilitate collaborative AI policy discussions.

Other OECD work relevant to the specific application of AI in the public sector can be accessed through the OECD Observatory of Public Sector Innovation (OPSI), which provides an overview of AI measures implemented in the public sector, including on AI governance policy making and public service design and delivery.

fAIr LAC

fAIr LAC Initiative

In collaboration with partners and strategic allies, the Inter-American Development Bank (IDB) leads the fAIr LAC initiative to promote the responsible adoption of Artificial Intelligence (AI) and decision support systems to improve social services delivery and create development opportunities to reduce social inequality. This toolkit is part of a set of documents and tools to guide technical teams and policymakers towards that end (Pombo et al. 2020).



Acknowledgements

The authors would like to express special thanks to Cristina Pombo, coordinator of the IDB fAIr LAC initiative, and to Professor Ricardo Baeza-Yates, Director of Data Science at Northeastern University, Silicon Valley Campus, and a member of the Group of Experts and Experts of fAIr LAC, for their time and valuable contributions. The authors are also grateful for the contributions from Karine Perset, OECD.AI administrator, and Luis Aranda, OECD.AI policy analyst.

The authors also appreciate the support and comments from Luis Tejerina, Elena Arias Ortiz, Natalia González Alarcón, Tetsuro Narita, Constanza Gómez-Mont, Daniel Korn, Ulises Cortés, José Antonio Guridi Bustos, Cesar Rosales, and Sofia Trejo.

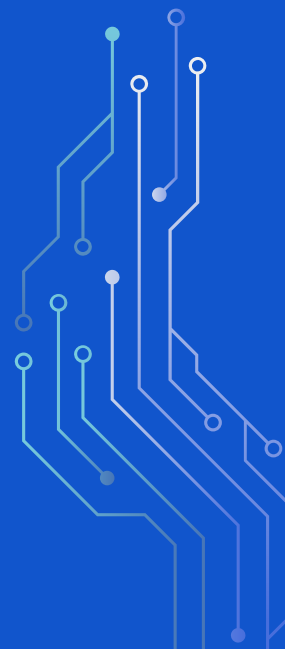


TABLE OF CONTENTS

fAIr LAC Initiative	7
Executive Summary	7
Why This Toolkit?	7
Who Is This Toolkit For?	8
Key Terms	8
Introduction	10
Machine Learning and Decision-Making or Decision-Support Systems	10
Components of an AI System for Public Policies	12
Challenges of the Machine Learning Life Cycle	13
1. Planning and Design	16
1.1. Correct Definition of the Problem and the Public Policy Response	17
1.2. OECD AI Principles	17
2. Data Collection and Processing	20
2.1. Data Quality and Relevance of the Available Data	21
2.2. Data Qualification and Completeness for the Target Population	23
3. Model Building and Validation	27
3.1. Absence or Inappropriate Use of Validation Samples	28
3.2. Data Leakage	29
3.3. Classification Models: Probabilities and Classes	30
3.4. Under and Overfitting	33
3.5. Unquantified Errors and Human Evaluation	34
3.6. Fairness and Differential Performance of Predictors	35
4. Deployment and Monitoring	38
4.1. Performance Degradation	39
4.2. Experiments to Evaluate Model Effectiveness	40
5. Accountability	41
5.1. Interpretability and Explainability of Predictions	42
5.2. Traceability	44
Tools	46
Tool 1: Robust and Responsible AI Checklist	47
Tool 2: Data Profile	52
Tool 3: Model Card	54
Workbooks	56
Data Collection and Processing	56
Model Building and Validation	67
Accountability	89
References	93

EXECUTIVE SUMMARY

From finance and insurance to agriculture and transportation, Artificial intelligence (AI) technologies are diffusing apace in all sectors, creating opportunities but also raising distinctive policy issues.

In the public sector, AI promises to generate productivity gains and improve the quality of public services. By analyzing social network activity in real time, policy makers can for example leverage AI systems to obtain a more accurate, evidence-based assessment of the most pressing societal problems and needs. The outcomes and predictions made by AI systems can inform policy formulation, implementation and evaluation.

Against this backdrop, governments around the world are equipping themselves with the relevant technical skills to leverage the power of AI in support of public policy development. However, given that AI-enabled public policy can significantly impact people's lives and well-being, a systematic approach is needed to ensure that the appropriate safeguards are in place to seize the opportunities from – and address the challenges posed by – the use of AI systems by public policy teams.

Using the AI system lifecycle as the guiding framework for analysis, this toolkit provides technical guidance for public policy teams that wish to use AI technologies to improve their decision-making processes and outcomes. For each phase of the AI system lifecycle – planning and design, data collection and processing, model building and validation, and deployment and monitoring – the toolkit identifies the most common challenges of using AI in public policy contexts and outlines practical mechanisms to detect and mitigate these challenges.

Policy makers and their technical teams should be accountable for the proper functioning of an AI system at each phase of its lifecycle. In this regard, one chapter of the toolkit is dedicated to exploring accountability-related issues in the use of AI for public policy and to outlining practical mechanisms to addressing them.

True to its objective of promoting the responsible use of AI for public policy making each section of the toolkit includes checklists to help guide practical implementation. A “data profile” tool and a “model card” are also provided to help assess data issues and to document an AI system's characteristics, the assumptions made, and the risk mitigation measures implemented throughout the lifecycle. Moreover, the toolkit provides a section with a workbook containing practical examples of some of the challenges and mitigation strategies covered in the report, as well as the relevant code to implement them using R or other programming languages.

Through the fAIr LAC initiative and the OECD.AI Policy Observatory, the IDB and the OECD have partnered to help move the AI policy discussion from high-level principles to practice and implementation. This toolkit is a concrete step in this direction.

Why This Toolkit?

Although there are a significant number of principles in support of ethical AI, they provide high-level guidance on how it should or should not be developed, and there is very little

clarity on the best practices for putting those principles into operation (Vayena 2019). The objective of this toolkit is to identify areas of risk and recommend mitigation measures to avoid outcomes contrary to the aims of decision-makers. Such outcomes include undesirable consequences, wasting of resources due to inadequate targeting, or any other outcome that undercuts what decision-makers are seeking to achieve.¹

Who Is This Toolkit For?

This toolkit is intended for technical teams working on the application of machine learning algorithms for public policy. However, it covers challenges common to other applications of this technology. It is assumed that the reader has basic knowledge of statistics and programming, although when concepts are introduced, brief descriptions and additional references are included. The toolkit includes workbooks with various examples of the challenges and solutions discussed. Different types of models (linear, tree-based, and others) and different implementations (R, Keras, Xgboost) are used to show that these problems arise regardless of the choice of a particular tool or algorithm. Although the codes and examples were developed in R, all the topics and methodologies applied and described in this toolkit can be implemented in any other programming language.²

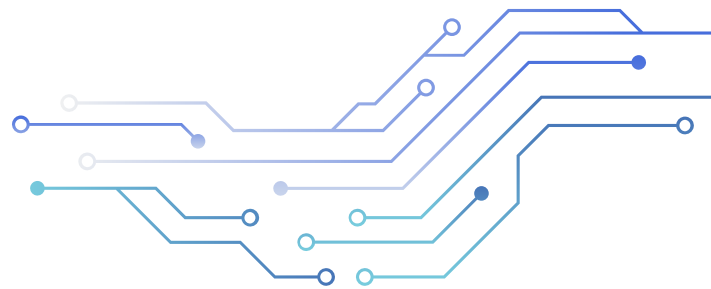
Key Terms

- **Artificial intelligence:** Machine-based system that is capable of influencing the environment by producing an output (predictions, recommendations or decisions) for a given set of objectives. It uses machine and/or human-based data and inputs to (i) perceive real and/or virtual environments; (ii) abstract these perceptions into models through analysis in an automated manner (e.g., with machine learning), or manually; and (iii) use model inference to formulate options for *outcomes*. AI systems are designed to operate with varying levels of autonomy (adapted from OECD 2019c).
- **Algorithmic fairness:** Mathematical representation of a specific definition of fairness that is incorporated into the model selection and fitting process. It is important to take into account that different definitions can be exclusive; that is, satisfying one could imply not satisfying the others (Verma and Rubin, 2018).
- **Algorithmic inequality:** Technical flaws in the models that produce a disparity of results for protected groups, and which must be evaluated in the context of a given definition of algorithmic fairness.
- **Decision-making systems:** Related to the concept of automated and autonomous intelligence. Final decisions and their consequent actions are made without direct human intervention – that is, the system performs tasks previously done by a human. In many contexts, these systems are referred to as automated decision-making systems.
- **Decision support systems:** Related to the concept of assisted or augmented intelligence, decision support systems generate information used as input for decision-making by a human.

¹ This toolkit is not intended to regulate or explain what the aims and objectives of decision-making bodies and actors should be.

² All the material in this toolkit is reproducible according to instructions in the repository (<https://github.com/EL-BID/Manual-IA-Responsable>), which contains a Dockerfile describing the infrastructure dependencies for its replication. The R programming language is used along with the following packages: tidyverse, recipes, themis, rsample, parsnip, yardstick, workflows, tune, knitr, and patchwork.

- **Machine learning:** Set of techniques that allow a system to learn behaviors in an automated manner through patterns and inferences instead of explicit or symbolic instructions entered by a human (OECD, 2019c).
- **Predictive structure:** types of models used to make predictions (linear, random forests, neural networks), the algorithm parameters and hyperparameters and its interactions.
- **Probabilistic guarantees:** Using samples designed with randomization, it is possible under certain assumptions to characterize the behavior of estimators and procedures (with high probability) – for example, a 95% confidence interval for performance metrics that contains the actual value that will be observed.
- **Subpopulations of interest or protected subpopulations:** Subpopulations of the target population for which we want to have concrete performance evaluations of estimates or models.
- **Target population:** Entire group of people, households, geographical areas, etc. targeted by a specific policy.



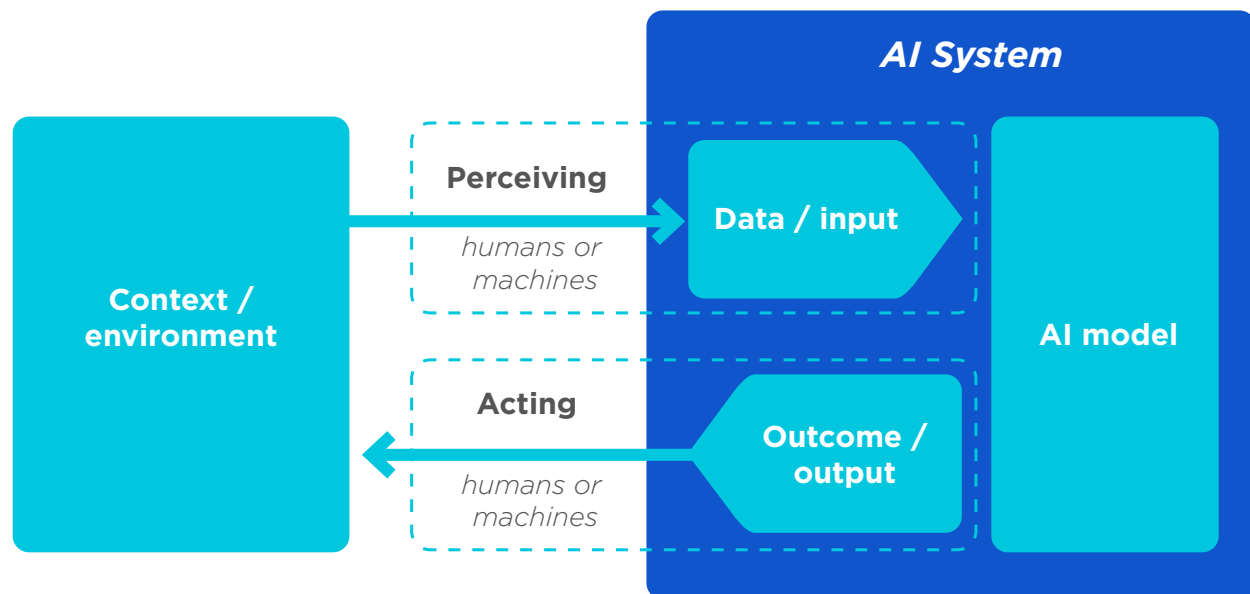
Introduction

Machine learning is a subset of artificial intelligence (AI). Machine learning methods are increasingly used by decision-makers to inform actions or interventions in various contexts, from business to public policy and service delivery. In practice, these methods have been used with varying degrees of success, and there has been growing concern about how to understand the positive or negative performance and influence of these methods on society (Barocas and Selbst 2016; Suresh and Guttag 2019).

Machine Learning and Decision-Making or Decision-Support Systems

The Organisation for Economic Co-operation and Development (OECD) describes an AI system as a machine-based system that is capable of influencing the environment by producing an output (predictions, recommendations or decisions) for a given set of objectives. It uses machine and/or human-based data and inputs to (i) perceive real and/or virtual environments; (ii) abstract these perceptions into models through analysis in an automated manner (e.g., with machine learning), or manually; and (iii) use model inference to formulate options for outcomes. AI systems are designed to operate with varying levels of autonomy (adapted from OECD 2019c).

Figure 1. Stylised conceptual view of an AI system



Although machine learning methods are not the only type of algorithms that AI systems can use, they are the ones that have seen the most growth in recent years. These methods constitute a set of techniques that allows a system to learn behaviors in an automated way through patterns and inferences instead of explicit or symbolic instructions entered by a human (OECD 2019c).

This toolkit discusses some of the most common challenges in the use of machine learning technologies for decision making or decision support. These include detecting and mitigating implementation errors and biases and evaluating the possibility of undesirable results for a company, public sector institution, or society.

Two archetypes of the use of machine learning in the decision-making process are considered:³

1. **Decision-support systems:** Related to the concept of assisted or augmented intelligence, decision support systems generate information used as input for decision-making by a human.
2. **Decision-making systems:** Related to the concept of automated and autonomous intelligence. Final decisions and their consequent actions are made without direct human intervention. That is, the system starts to perform tasks previously done by a human. In many contexts, these systems are referred to as automated decision-making systems.

There is a wide variety of techniques, expert knowledge of the subject, and modelling in general for the development of a successful decision-making/decision support system based on machine learning. This toolkit is not intended to discuss particular machine learning methods or specific hyper-parameter tuning processes (Hastie, Tibshirani, and Friedman 2017; Kuhn and Johnson 2013; Gelman and Hill 2006), but rather to focus on the evaluation of those methods and on the most important challenges shared across systems, regardless of the type of algorithm or technology used.

The evaluation of a machine learning system should be carried out on a case-by-case basis. Questions such as “what is the maximum error rate tolerated?” or “what are unacceptable biases?” can only be considered and answered within the specific context of their application. This includes the purposes and motivations of decision-makers, as well as the risk(s) to end users and stakeholders. In other words, many of the technical criteria should pertain to the specific problem at hand. If the biases and limitations of decision-making/decision support systems are known, even a system with low precision can be useful, if used responsibly. On the other hand, if a system’s limitations are not understood, even high-performing systems can lead to unintended consequences or misuse.

Objectives

- This toolkit focuses on the subset of challenges that are related to technical processes throughout the lifecycle of AI systems used for decision-making or decision support.
- This toolkit describes how different biases and deficiencies can be caused by training data, problems, and decisions in the development of the model, or in the validation or monitoring process, which can produce undesirable or biased results in the decision-making process.

³ These two types of systems are generic, that is, they do not necessarily use machine learning. Also, these systems can be interactive and learn dynamically using reinforcement learning techniques, but in this toolkit we only consider non-interactive systems.

Components of an AI System for Public Policies

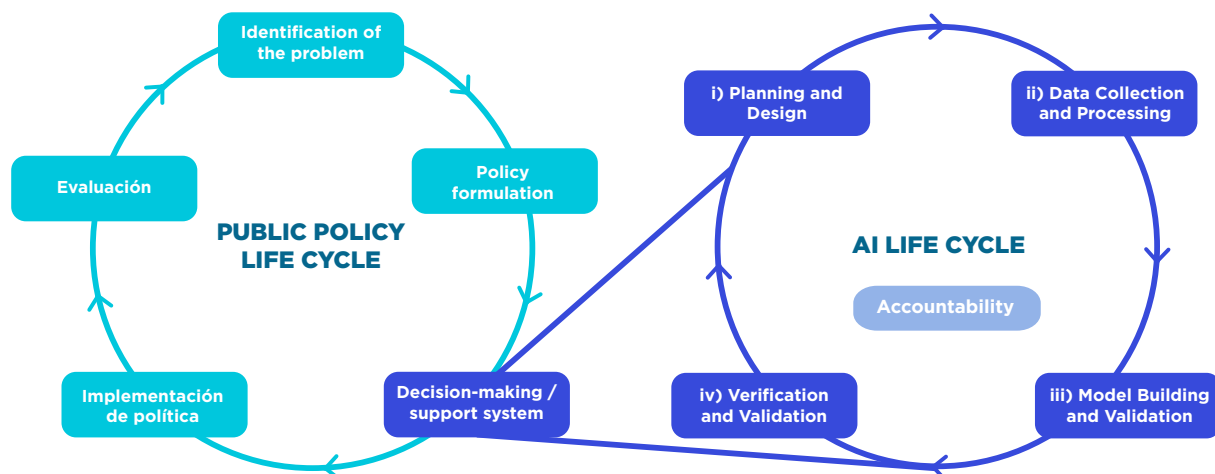
Decision-support AI Systems in the Public Policy Life Cycle

AI does not replace public policy making. Its function is to assist the public policy development cycle by providing information for decision-making. The AI-assisted public policy cycle is made up of the following stages:

- 1. Identification of the problem:** Every AI project should begin with correctly identifying the issue that public policy seeks to address, along with the possible causes and consequences of that issue.
- 2. Policy formulation:** The intervention or policy that is being considered to be applied to certain people, units, or processes is formulated. We will generally assume that there is evidence of the benefit of such a policy when applied to the target population.
- 3. Decision-making/Decision support system:** Once the intervention is defined, the AI cycle begins with the design and development of the decision-making/decision support system, the result of which will be used to focus or guide the intervention chosen in the previous stage (OECD, forthcoming).⁴
- 4. Policy implementation:** Public policy is implemented either as a pilot project or on a larger scale.
- 5. Policy evaluation:** The effectiveness, reliability, cost, expected and unintended consequences, and other relevant characteristics of the policy measure are evaluated. If its results are positive, the intervention is continued or scaled.

In parallel to the public policy making cycle, the development of an AI system has its own lifecycle that includes the following stages (OECD, 2019c): (i) Planning and Design, (ii) Data Collection and Processing, (iii) Model Building and Validation, (iv) Deployment and Monitoring. These phases usually take place iteratively and are not necessarily sequential (Figure 2).

Figure 2. Public Policy Lifecycle Supported by a Decision-making/Decision Support System



Source: Prepared by the authors.

⁴ AI can be used in different ways, including recognition, event detection, forecasting, personalization, interaction support, goal-driven optimization, reasoning with knowledge structures, or any combination of them embedded in a composite system (e.g., a driverless vehicle).

In the interrelation of these two cycles, important challenges are generated that should be evaluated and considered during the development and use of robust and responsible AI systems.

Challenges of the Machine Learning Life Cycle

The construction of robust and responsible decision-making or decision support AI systems requires careful consideration of all possible sources of bias; investigation of deficiencies and documentation of assumptions; clear definition of the algorithmic fairness objectives and criteria that the system must meet; understanding the limitations and tolerable errors in the specific context of the system; and implementation of monitoring measures to avoid undesirable or biased results in decision-making.

To achieve this, this toolkit presents the common challenges and mistakes in building and applying machine learning methods during the AI system lifecycle. The most common problems, the mechanisms to detect them, and suggestions to mitigate them are described according to each stage of the AI system life lifecycle:

1. **Planning and Design:** Refers to the information and criteria from the public policy decision-maker necessary to conceptualize an AI project.
2. **Data Collection and Processing:** Focuses on the data generation process, the selection and control over data sources and inputs⁵, and the identification and mitigation of deficiencies and biases in the data.
3. **Model Building and Validation:** Includes key principles and methods for building robust and correctly validated models.
4. **Deployment and Monitoring:** Evaluation of the model in production and monitoring of key principles to avoid unexpected degradation.

In addition, Accountability is a cross-cutting dimension in the AI system lifecycle that refers to the explanatory and transparency-enhancing measures to promote understanding of the mechanisms through which an AI system produces an output, the output's reproducibility, and the user's capacity to identify and challenge errors or unexpected results. AI actors at each stage of the lifecycle should be accountable for the proper functioning of an AI system based on their roles and the context, and consistent with the state of art.

Three tools are proposed to accompany the development of the AI system:

- **Tool 1: Robust and Responsible AI Checklist:** This tool consolidates the main concerns by stage of the AI life cycle. The checklist must be reviewed continuously by technical teams and decision-makers.
- **Tool 2: Data Profile:** This tool is an initial exploratory analysis during the Data Collection and Processing stage of the AI lifecycle. It provides information to reassess the quality, completeness, temporality, and consistency, as well as possible biases, potential damage, and implications of the use of an AI system.
- **Tool 3: Model Card:** This tool is a final description of an AI system, reporting its main assumptions and most important characteristics, as well as the risk mitigation measures implemented.

⁵ See for instance the OECD Good Practice Principles for Data Ethics in the Public Sector, in specific the principle on AI systems and data. Available at: <https://www.oecd.org/gov/digital-government/good-practice-principles-for-data-ethics-in-the-public-sector.pdf>

Box 1. Sources of Bias in an AI System

Bias poses one of the most pressing challenges throughout the AI lifecycle. Many of the mitigation measures that should be considered during AI model development depend on the correct understanding and treatment of such biases. The problem of bias should be addressed early in the development process through the implementation of review points at each different stage of the lifecycle. At each review point, the experts and end users of the corresponding system should be invited to verify and defend the hypotheses made during each stage and to validate the results of the model. The following are concepts that are important to explain.

System error is the difference between the predicted value resulting from the model and the real value of the variable that is being estimated.⁶ When an error is systematically in one direction or for a specific subset of the data or a specific subpopulation, it is called **bias**. For example, if a variable's value is consistently lower for one subgroup in the data, such as the salary of women with respect to equally qualified men for an equivalent job, the salary variable is biased. Conversely, if the error is random, it is called **noise**.

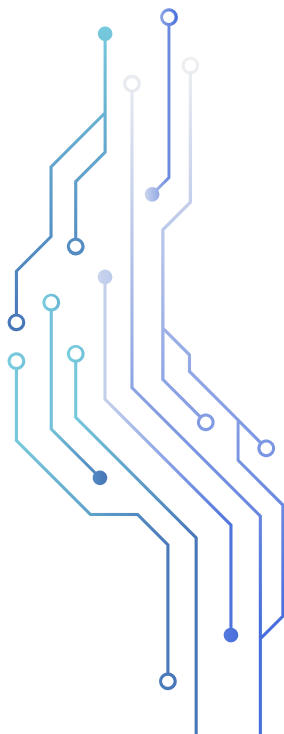
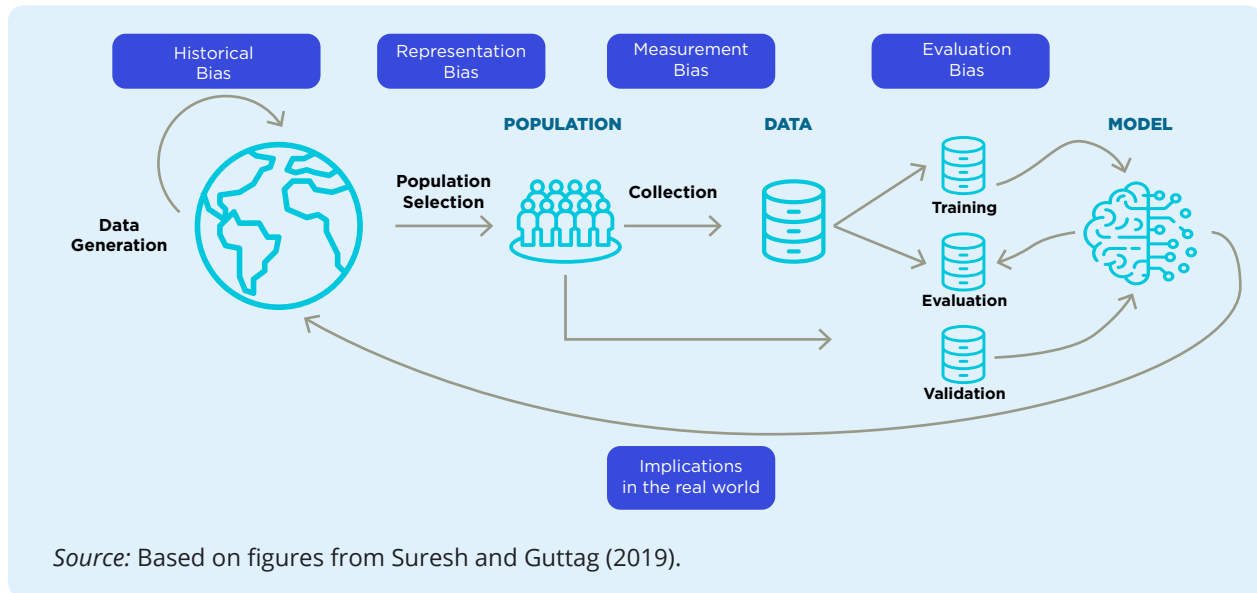
The bias of an AI system can have ethical implications when its results are used to formulate public policies that can be considered unfair or prejudicial to certain subgroups of the population. This assessment of bias is subject to a specific algorithmic fairness definition, to be determined by public policy decision-makers.

An **algorithmic fairness definition** is a mathematical representation of a public policy objective that is incorporated into the model selection and fitting process. Its definition is a task for public policy decision-makers and not for technical teams. The modelling team should only carry out validations to ensure compliance. In Section 3 of this toolkit, different definitions of algorithmic fairness and their implications will be discussed in depth.

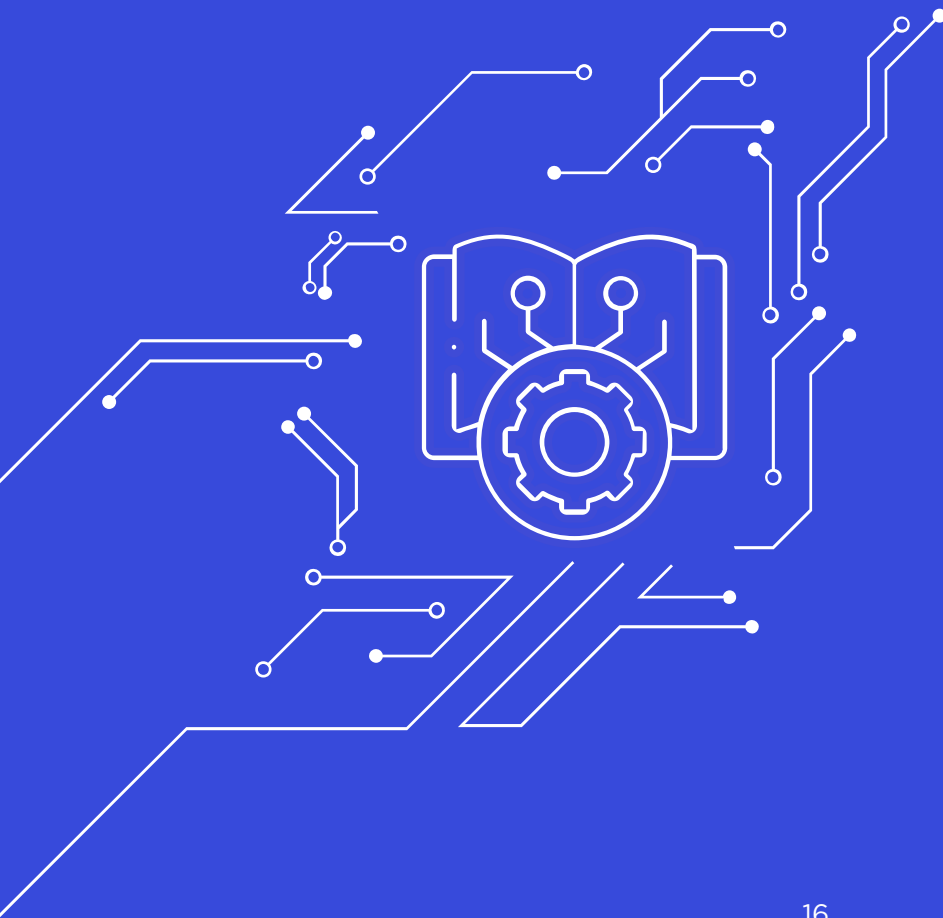
For example, in some cases the objective of a system may be bound by criteria such as demographic parity, equality of possibilities, and having representation by quotas, among many other criteria. On some occasions, compliance with one definition of algorithmic fairness makes it impossible to comply with another, that is, they can be partially or totally exclusive.

There are different **sources of bias**. Some biases are intrinsic to the data, including historical biases or undesirable states; that is, pre-existing patterns that should not be replicated in the model. **Representation bias** occurs when there is incomplete information due either to missing attributes, sample design, or total or partial absence of data from subpopulations. **Measurement biases** arise from the omission (inclusion) of variables that should (not) be included in the model (Suresh and Guttag 2019). Other biases appear due to methodological errors: for example, biases arise during training due to errors in the validation processes, definition of metrics, and evaluation of results (**evaluation bias**), biases arise due to **erroneous assumptions** about the target population that may affect the definition of the model; and biases arise due to the misuse and monitoring of the models, whether because of inappropriate interpretations of their results or temporary changes in the patterns in the real world or in the data-capture methods. Throughout the different sections of this toolkit, the main reasons for these biases will be presented, and different measures to mitigate them proposed.

⁶ In prediction models, there is a trade-off between variance and bias captured by the model and its learning generalization goal. Models with high bias can create systems that under-fit and learn very little from the observed data, but models with high variance can have the opposite effect and over-fit by perfectly learning the training data. The "Model Building and Validation" section of this toolkit describes these phenomena in greater detail and measures to mitigate the corresponding risks.



1. PLANNING AND DESIGN



1. Planning and Design

The implementation of an AI solution cannot be considered separately from the AI public policy lifecycle.⁷ The results of the AI system are only as good as the design of the public policy intervention in which the system is embedded. AI is a tool – and not a substitute – for public policy. This implies that projects that employ robust and responsible AI must start from the definition of the problem and not from the technology itself.

1.1 Correct Definition of the Problem and the Public Policy Response

This toolkit assumes that there are at least two types of actors involved in the development of the AI system: the policy maker and/or decision maker and the technical team that develops and implements the system. The definition of the public policy should always be the responsibility of the decision-maker with knowledge of the social dynamics and issues.

The technical team must be able to understand the problem so as to be able to orient the results of the model towards satisfying the goal of the desired intervention. Likewise, the technical team is responsible for advising and guiding the design of the system, explaining what is feasible, and clearly delineating the system's limitations and risks. Constant communication between the decision-maker(s) and the technical team is required.

For instance, the definition of the population to which the system will be applied, the protected groups and protected attributes, and the algorithmic fairness measures to be applied need to be discussed and understood by both sets of actors.⁸ These definitions have a direct impact on how the quality and coverage of the data – as well as the quality of the model results – can be assessed.

1.2 OECD AI Principles

Although AI has significant potential to streamline some processes and expand state capacity, it should also be noted that it is not a silver bullet. Once the problem and the type of intervention have been defined, it is necessary to contextualize and rethink the use of AI and machine learning in line with the OECD AI Principles (Box 2).

It is important to consider the broader governance that frames the application of an AI system, including the standards and laws in the jurisdiction where the system is to be implemented. It is also important to establish appropriate requirements during the planning and design of the system, as they can define or narrow development options for the technical team. For example, explainability requirements in predictions could limit the use of some algorithms for which it is very difficult to interpret the results.⁹

⁷ See the section entitled “Components of an AI System for Public Policies” in the Introduction of this toolkit.

⁸ Section 3 of this toolkit discusses different definitions of algorithmic justice and their implications in depth.

⁹ The concept of explainability is described in Section 5.1.2 of this toolkit.

Box 2. OECD AI Principles

The Organisation for Economic Co-operation and Development (OECD)'s AI Principles promote the use of Artificial Intelligence (AI) that is innovative and trustworthy and that respects human rights and democratic values. The principles set standards for AI that are practical and flexible enough to stand the test of time. They include five values-based principles for the responsible stewardship of trustworthy AI:

- **Inclusive growth, sustainable development, and wellbeing:** Stakeholders should engage in creating credible AI that can contribute to inducing outcomes that are beneficial for people, as well as for the planet.
- **Human-centered values and fairness:** The values of human rights, democracy, and the rule of law should be incorporated throughout the AI system's lifecycle, while allowing human intervention through safeguard mechanisms.
- **Transparency and explainability:** AI actors that develop or operate AI systems should provide information to foster an overall understanding of the systems among stakeholders that allows for people affected by AI systems to comprehend the outcome and challenge the decision when needed.
- **Robustness, security and safety:** AI systems need to function appropriately throughout their lifecycle. AI actors should ensure traceability and apply systematic risk management approaches to mitigate risks.
- **Accountability:** AI actors developing, deploying or operating AI systems should respect the principles and be accountable for the proper functioning of those systems.

The OECD AI Principles also contain five recommendations for national policies and international cooperation: (1) Investing in AI research and development; (2) Fostering a digital ecosystem for AI; (3) Shaping an enabling policy environment for AI; (4) Building human capacity and preparing for labor market transformation; and (5) Promoting international cooperation for trustworthy AI (OECD 2019a, 2019b). The principles were adopted in May 2019 by OECD member countries and are the first international standard on AI signed up to by governments. Beyond OECD members, other countries including Argentina, Brazil, Costa Rica, Malta, Peru, Romania, Ukraine, Singapore and Egypt have already adhered to the AI Principles, with further adherents welcomed. In June 2019, the G20 adopted human-centred AI Principles that draw from the OECD AI Principles.

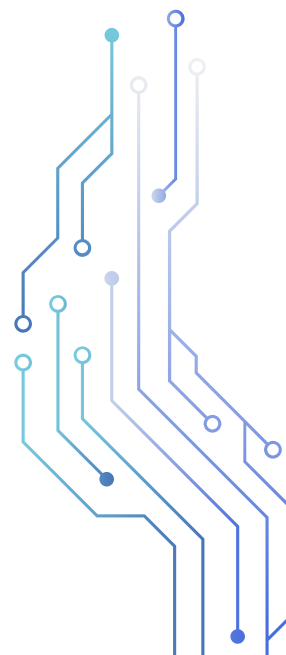
Box 3. Planning and Design Checklist

☒ Correct definition of the problem and the public policy response:

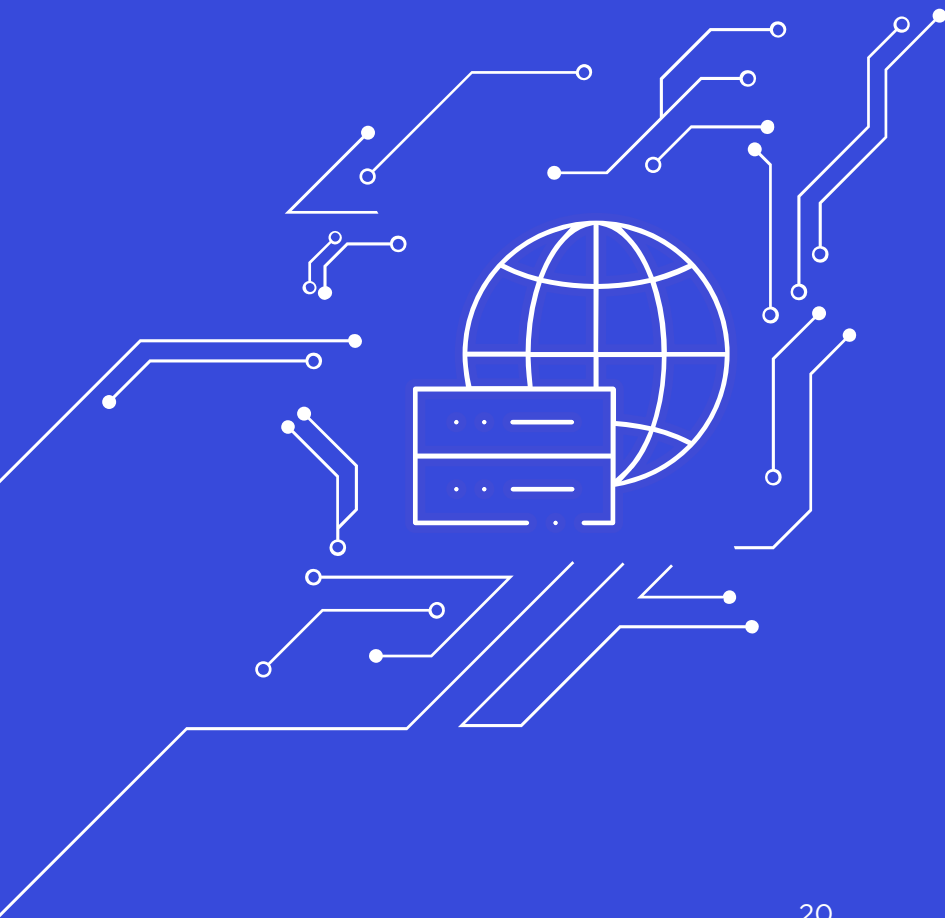
- (Qualitative) Is the public policy problem clearly defined?
- (Qualitative) Describe how this problem is currently being addressed – considering responses by related institutions – and how the use of AI would improve the government response to this problem.
- (Qualitative) Were the protected groups or protected attributes identified within the project (e.g., age, gender, education level, race, level of marginalization, etc.)?
- (Qualitative) Were the actions or interventions to be carried out based on the result of the AI system defined?

☒ AI Principles

- (Quantitative) Has the need for an AI system been justified, considering other possible solutions that do not require the use of personal data and automated decisions?
- (Quantitative) Is there evidence that both public policy action and the recommendation of the AI system will result in a benefit to people and the planet by driving inclusive growth, sustainable development, and well-being?
- (Qualitative) For the implementation of these technologies, have there been similar previous projects and have they been reviewed?
- (Quantitative) Have you considered minimizing the exposure of personally identifiable information (e.g., by anonymizing or not collecting information not relevant to the analysis)?



2. DATA COLLECTION AND PROCESSING



2. Data Collection and Processing

A plethora of data sources can be used to inform public policy decision-making: censuses, surveys, administrative records, web page usage logs, and even satellite images. These data become useful information when they describe the target population or the phenomenon that is being analyzed.

However, the data collected do not always have a frequency, disaggregation, or coverage that make them relevant or afford them the quality required to be used for decision-making. For example, surveys designed using probability sampling specify the type of analysis that can be done with them – by design – but tend to be conducted infrequently and thus may be insufficient to capture patterns in the data. On the other hand, information from administrative records or data from the Internet (interaction on social networks, visits and other measurements on web pages, etc.) and telephony (calls, GPS location, etc.) tend to have a much higher frequency, but only in a few cases represent the whole population, so it is not always possible to use these data to make decisions for the entire population.

Statistical AI systems are based on data. Regardless of whether a supervised or unsupervised model is being implemented, training data are key for any machine learning system. Data quality and qualification can be analyzed using criteria such as volume, completeness, validity, relevance, representativeness, precision, timeliness, accessibility, comparability, and interoperability from different sources. Defining these criteria precisely is difficult, as the context of each problem entails subtle idiosyncrasies. Relevance and precision refer to the quality of measurement and usefulness to inform the decision, while timeliness refers to the fact that the data occur with the timeframe necessary to inform the problem to be analyzed. Accessibility, comparability, and interoperability refer to the fact that the data can be extracted in a timely manner and different data sources have the necessary consistency to be applied jointly in the analysis.¹⁰

This section addresses challenges in data collection and processing related to two common data-related concerns for machine learning systems¹¹:

1. Data quality and relevance of the available data; and
2. Data qualification and completeness for the target population.

Sections 2.1 and 2.2 touch upon some of the issues highlighted in the OECD Good Practice Principles for Data Ethics in the Public Sector with regards to data quality and data qualification. The Good Practice Principles aim to support public officials in the implementation of data ethics in digital government projects, products, and services such that i) trust is placed at the core of their design and delivery and ii) public integrity is upheld through specific actions taken by governments, public organisations and, at a more granular level, public officials (OECD, 2021).

2.1. Data Quality and Relevance of the Available Data

Machine learning algorithms capture observed patterns and relationships from the data they are trained on. Their objective is to identify these same patterns for new cases not observed during the model training. For this reason, the training data determines the way in which the

¹⁰ At this stage it is recommended that the Data Profile ([see Tool 2 in this toolkit](#)) be filled out.

¹¹ While not covered in detail in this section, other data-related concerns – such as data domain and structure – are included in the Data Profile.

algorithm will behave. However, available data are not always ideal for every use case. Two of the main problems are:

1. Undesirable or suboptimal states in collected data.
2. Bad correspondence between ideal and available variables.

2.1.1. Undesirable or Suboptimal States in Collected Data

The first challenge is to identify training data that may have captured “undesirable states” of the real world. These “undesirable states” can include biases and inequities that can lead to harmful outputs for certain subgroups of the population, as well as any other pattern that could be considered suboptimal or undesirable from a social policy point of view.



Example

In 2015, Amazon experimented with a human resources recommendation system based on supervised learning techniques. The model was trained using a database from the company’s candidate selection processes stored over the previous 10 years. That database identified whether a candidate had been accepted or rejected for the job by the department. The system was based on the assumption that the algorithm could capture good candidates and reduce the work of the human resources department when making a first selection of candidates. What the team had not taken into account is that the technology industry has been characterized as predominantly male. So the system tended to recommend a higher proportion of men, since more men had been accepted into those positions historically, creating a bias that seemed to show that men were more successful when in fact it was capturing bias.

Box 4. Undesirable or Suboptimal States in Collected Data Checklist

- (Qualitative) Discuss possible historical social inequalities in the use case with specialists in the field.
- (Quantitative) Perform an exploratory analysis of the available data with which the model will be trained to identify possible historical biases or undesirable states.

2.1.2. Poor Correspondence between Ideal and Available Variables

When public policy decisions are made, they are based on the definition of one or more “ideal” target variables that the decision-maker has in mind. However, the ideal variables may or may not be available in the accessible data. In many cases, it is necessary to use substitute or proxy variables to get closer to the ideal variable. When we introduce these types of variables into machine learning models, we may be learning implicit biases that may not be desirable. For example, a scholarship that seeks to benefit the smartest students (ideal variable) will run into the problem of defining what is meant by “smart” and finding a variable that can describe this concept. An IQ test assigns a value using a standardized test that is described as a proxy

variable for intelligence. However, the test measures only some dimensions of intelligence, so it will underestimate the intelligence of some people (Wilson 2014).

The ideal target variables should be clearly stated. The available variables must be analyzed to understand how suitable they are to be used as a proxy for the ideal variable. Systematic biases must be identified within the context of their use.



Examples

The U.S. health care system implemented an algorithm to predict the medical care needs of different patients. In this case, the public policy decision-maker wanted a tool that would preventively indicate which patients were at high risk of requiring more medical care using historical information from hospitals. Given that the ideal complication risk variable was not available, the algorithm used the expenditure incurred by patients during their illness as a proxy variable, under the hypothesis that sicker people would end up spending more on medical treatment to overcome the disease. Obermeyer et al. (2019) showed that this system was racially biased because it underestimated the number of black patients in need of health care. The racial bias was caused by this subpopulation spending, on average, less money than white patients. Using expenditure as a proxy for risk of complications showed that healthier white patients appeared to require more healthcare than sicker black patients. In this case, using health spending as a proxy for the need for medical care was inappropriate because it was biased by an omitted variable of economic inequality.

Box 5. Poor Correspondence between Ideal and Available Variables

- (Qualitative) The ideal target variables should be clearly stated. The collected/available variables must be analyzed to understand how suitable they are to substitute for the target variable. Systematic biases or validity of the proxy metric should be identified.
- (Qualitative) Has the use of the selected response variable been clearly justified for the purposes of the intervention?

2.2. Data Qualification and Completeness for the Target Population

Machine learning models used in the public sector are intended to generate information that inform actions or policies for a target population. Most of the time, data sources do not include the entire population (as would be the case of a census), so only a subset or sample is available (e.g., a survey, administrative database, etc.), from which one should seek to develop extrapolations, predictions, or estimates that help in decision-making.

2.2.1. Probabilistic and Natural Samples

In statistics, a sample is a subset of cases or individuals from a population. There are two sampling possibilities:

1. **Probability sampling:** This is the name given to a sample in which the cases are selected from a probabilistic design that infer several possible models to explain data and for which deciding which model to use is uncertain (e.g., simple, stratified, cluster random

sample, etc.). In this case, all the predictions and estimates that are to be applied to the target population can be evaluated for their precision with probabilistic guarantees. That is, error ranges can be provided for estimates of quantities associated with the entire target population.

For example, a national household survey with a probabilistic design generally consists of a definition of stratification, with units of random selection at different levels (primary, secondary units, etc.). Each household is selected with a known probability. Even if the sample is designed in a non-representative way (e.g., more households in rural or low-income areas), it is possible to make inference for the entire population with certain guarantees about the size of the estimation error.

- 2. Natural samples (non-probabilistic):** A natural or non-probabilistic sample occurs when the cases are not selected randomly but by a flawed process or a partially understood natural process. In this case, it is not possible to know what will happen when we apply a policy resulting from a model in the general population; nor is it possible to construct error ranges for predictions and estimates using statistical methods that have probabilistic guarantees. That is, the estimated quantities and predictions have unknown error, the models and characteristics useful in the sample may not apply in the target population, and the situation may be aggravated for underrepresented – or protected – groups (see Williams 1981, who shows that predictive values of anemia may be different for different racial groups, and that predictions developed for one group may perform poorly for another.)

A usual case of this type of sample occurs when particular subgroups of the population are excluded by a flawed data capture mechanism (selection bias). This is the case for social networks or phone call record data where the population without access to the Internet or a smartphone is excluded.

Natural samples of data can result in:

- Estimation and/or prediction errors or biases
- Predictive structures different from those we would observe in the target population (invalid models)
- Extrapolations that are not supported by the data
- Under- or over-representation of subsets of the population.

Probability sampling would be the preferred method for most machine learning projects. In this case, it is possible to understand exactly which sub-populations were sampled, at what rates, and how these rates are related to population rates. However, having a probability sample is not always possible.

This does not mean that natural samples are not useful, since in many cases they are the only source of data available for decision-making. However, it is important to understand where the data come from in order to take into account its limitations and identify the risks involved when making decisions for the entire population.

A typical case is the data samples that come from social networks where the user's demographic composition differs substantially from the general population. A study for the United Kingdom found that, on average, Twitter and Facebook users are considerably younger than

the general population and more likely to have higher levels of education than non-users (Prosser & Mellon, 2016). Any study with these data should explain how these particularities can affect the results.

An important element to take into account is that having balanced samples in terms of population characteristics is neither a necessary nor a sufficient condition to qualify the database as appropriate for the construction of machine learning models. For example, in the case of information collected from social networks, having a sample that contains 50 percent men and 50 percent women does not tell you anything about the type of conclusions that can be drawn from those data. This is because the selection of these observations, not occurring through a probabilistic process, could present a bias in some other dimension and will not necessarily generalize to the total population.

Box 6. Probabilistic and Natural Samples Checklist

- (Qualitative) Have the possible differences between the database and the population for which the AI system is being developed been analyzed? (Use literature related to the topic and information from experts. Study in particular unmeasured selection biases.)
- (Quantitative) Although models can be built with various data sources, designed or natural, validation should ideally be carried out with a sample that allows statistical inference to the target population. The validation sample must appropriately cover the target population and sub-populations of interest.

2.2.2. Missing or Incomplete Attributes

Many machine learning projects are compromised because of poor data qualification. When collecting data from the real world through non-probability samples, it is very common for some observations to have missing data, that is, observations for which not all the attributes are available.

Missing or incomplete attributes are a phenomenon that can have a significant effect on the conclusions drawn from the data. On the one hand, when crucial information about the units is unknown, this can result in models with poor performance and little utility for decision-making. The absent information might also be associated with relevant characteristics of the units for which you want to predict.

When there are missing observations, different imputation methods can be implemented, but it is important to explore the reasons or the “censorship mechanism” behind the missing values. In the literature there are three main assumptions (Rubin 2002):

- **Missing Completely at Random (MCAR):** Occurs when the probability of missing is the same for all observations – that is, the censorship or fault occurs totally randomly.
- **Missing at Random (MAR):** Occurs when the missing values do not depend on the values that this variable takes, but there is a relationship between the missing values and other observed data of the individual.
- **Missing Not at Random (MNAR):** Occurs when the missing values depend on the values that this variable takes or on unobserved data. For example, people with higher income tend not to disclose their income on self-reported income surveys.

Box 7. Missing or Incomplete Attributes Checklist

- (Qualitative) Has an analysis of missing values and missing variables been performed?
- (Qualitative) Have important omitted variables – for which there are no associated measurements – been identified? (If any)
- (Qualitative) Have the reasons for the missing observations been identified? (If any)
- (Quantitative) The imputation processes have to be evaluated in terms of their sensitivity to assumptions and data. Preferably, multiple imputation methods should be used to assess imputation uncertainty (Little and Rubin 2002; Buuren and Groothuis-Oudshoorn 2011).

2.3. Causal Comparison

When humans rationalize the world, they try to understand it in terms of cause and effect. If we understand why something happened, we can alter our behavior to change future outcomes.

A machine learning model can give us results that seem to describe causal relationships that do not necessarily exist. This can lead to inadequate policies and wrong decision-making.

Econometric techniques such as randomized controlled trials, natural experiments, difference-in-differences methods, and instrumental variables are used to assess causality. They control for phenomena such as selection bias or endogeneity due to omitted variables, among others. In recent years, through works such as Athey (2018), machine learning algorithms have begun to introduce these experimental techniques. Processes like A/B testing have started to be used broadly in digital contexts because they facilitate the creation of large experiments on the Internet. However, in most cases machine learning algorithms do not seek to describe causal relationships, so it is necessary to be careful with this type of use of algorithms (Stuart 2008).

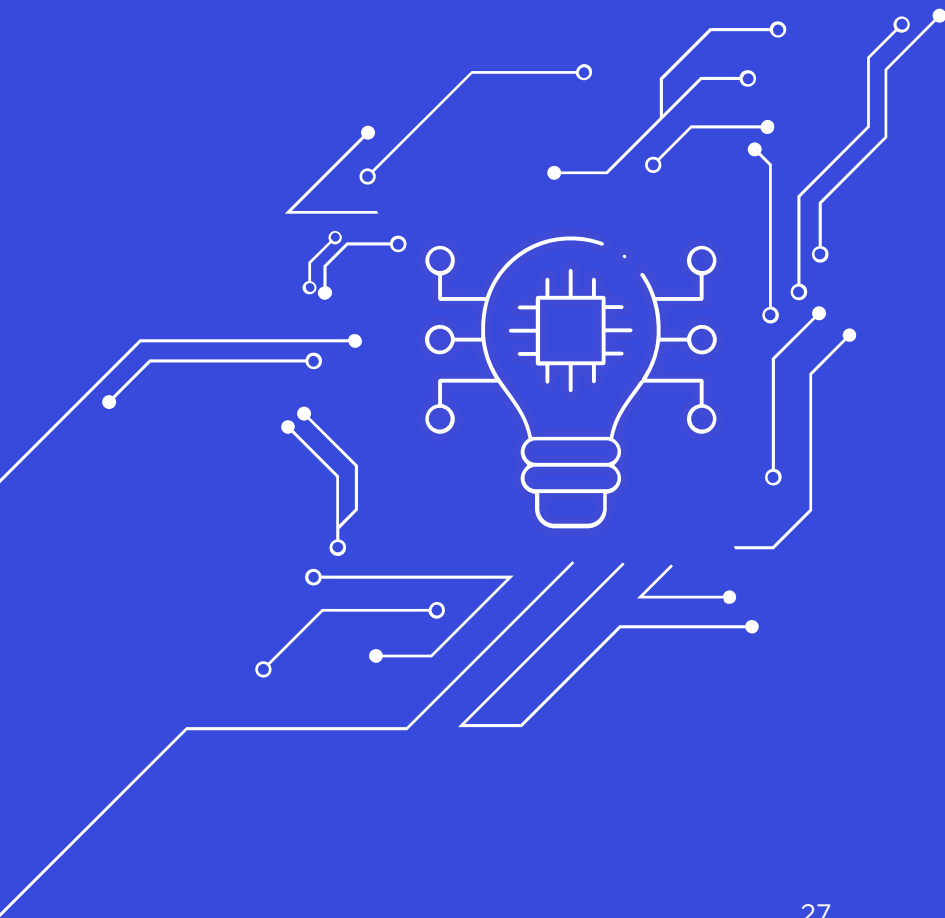
Box 8. Causal Comparison Checklist

- (Qualitative) Understand and describe the reasons why the response variable is correlated with known and unknown variables. Describe possible biases based on expert knowledge and analysis.
- (Qualitative) In case no work has been done to ensure causality in the results, were the limitations of the results explicitly communicated to the public policy decision-maker?
- (Quantitative) In the case of attempting causal inference with models, the hypotheses, considerations, or methods used to support a causal interpretation must be described. Robustness checks should be conducted and documented.

Activity: Filling out the [Data Profile](#) (see Tool 2) is recommended during the Data Collection and Processing stage of the AI lifecycle. At the end of this stage, it is recommended that the Source and Data Management Section of the [Model Card](#) (see Tool 3) be filled out and that a discussion be held with the public policy decision-maker.



3. MODEL BUILDING AND VALIDATION



3. Model Building and Validation

The process of developing a model involves making many decisions with implications for the model's results. Several types of methodological decisions, if flawed, may lead to errors that generate biases or that prevent the system from generalizing results adequately.

Another group of decisions is conceptual in nature and can substantially change the way the system behaves. How do we choose between two models? What type of errors do we report? What definition of algorithmic justice should we choose? As discussed at the beginning of the toolkit, none of these questions make sense outside the context of the AI system's specific application. However, it is possible to create a framework for understanding these errors so that they can be discussed by technical teams and public policy decision-makers.

3.1. Absence or Inappropriate Use of Validation Samples

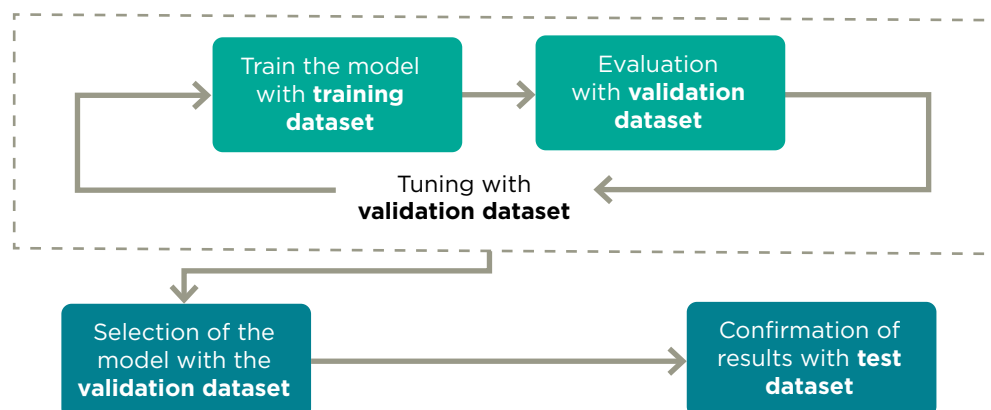
Machine learning models are primarily trained to create predictions in unobserved cases. It is useless to evaluate a system in terms of its prediction performance regarding the observations with which it was trained, since the system could only memorize each answer.¹² The system's usefulness lies in the extent to which it can make correct predictions using data outside the training set (*out-of-sample*).

Validation generally involves at least two data samples (training and validation), and preferably three (Figure 3):

1. **Training data:** Subset of the data used to train the model.
2. **Validation data:** Subset of the data with which the training is evaluated iteratively.
3. **Test data:** Subset of the data that should be kept hidden until after the model is selected and used to confirm the results.

To avoid a random partition in training and validation data that favors or hinders the evaluation, a cross-validation is generally carried out. This consists of dividing the data into k pieces, calculating the average of k evaluations, where the validation data are each of the pieces and the remaining $k-1$ s are the training data. This is called *k-fold evaluation* and usually $k = 5$ or $k = 10$ is chosen.

Figure 3. Evaluation Stages



Source: Prepared by the authors.

¹² This phenomenon is related to overfitting, which will be discussed later.

The first challenge is not having an appropriate validation process. In this case, the model results would only represent the training dataset. The performance metrics of this set should not be used as an indicator of the potential behavior of the model for new cases, as it could be overestimating its performance.

Successful validation is also related to quality criteria such as the completeness and representativeness of the information ([see Section 2](#)). This is because, if the target population is different from that represented by the data used during training, that population might have completely different behavior, even if the evaluation process was carried out correctly,

Box 9. Absence or Inappropriate Use of Validation Sample Checklist

- (Quantitative) Were the validation and test samples constructed properly? Did they consider an appropriate size, covering subgroups of interest and protected and avoiding information leaks during their construction?
 - The construction of the validation sample must be produced under a sampling design that allows inference to the target population (Lohr 2009).
 - The validation sample should cover subgroups of interest and protected, so that it is possible to make inferences as to their subpopulations. That includes appropriate sample sizes according to the sampling methodology (Lohr 2009).
 - If such a sample is not available, it is essential that there be an analysis of risks and limitations of the natural sample conducted by experts and professionals who know the process that generated these sample data.

3.2. Data Leakage

Data leakage occurs when information from outside the designed training dataset is used in the creation of the model, contaminating the training dataset (Kaufman, Rosset, and Perlich 2011). This additional information modifies the learning process and casts doubt on model validation as a way of estimating the production performance of system.

This occurs in two ways:

- **Training-test contamination:** When the training sample receives data leaks from the test or validation set.
- **Target leakage:** Inclusion of a feature that is not going to be available when the model is used in production.

3.2.1. Training-Test Contamination

Training-test contamination occurs when all or part of the validation or test samples are used for the construction of the models during training. This error often results in unrealistic performance levels in the validation set because the model is making predictions based on observations that it has seen before.

This error often occurs when applying pre-processing methodologies that aggregate and share information from the database composition to individual observations – for example, scaling a variable, creating co-variables with averages or counts, over- or under-sampling, etc. These processes should be performed after splitting the training and validation dataset.

Box 10. Training-Validation Data Leak Checklist

- (Quantitative) Any processing and preparation of training data should avoid using the validation or test data in any way. A solid barrier must be maintained between training versus validation and testing. This includes data recoding, normalizations, selection of variables, identification of outliers, and any other type of preparation of any variable to be included in the models. This also includes sample weights or balances based on over- or under-sampling.

3.2.2. Target Leakage

This error occurs when a model is trained with information that will not be available in the same way or with the same quality when the model is put into production. This generally has to do with the temporality of the data or groupings. In more subtle cases this error can be difficult to detect, since the variable is present, but the information is updated retroactively. An example of this can be seen in crime and mortality statistics. Reports of a theft may take time to be reported in authorities' databases due to bureaucratic or administrative processes, and the observed incidence of a period could systematically increase as time passes. In this example, the target variable is available in production but may not be complete given certain lags inherent to the reporting. If this phenomenon is not considered during the training and data are used that are already complete, the evaluation of the model may appear accurate, but in production the accuracy of the data will be significantly degraded.

Box 11. Target Leakage

The validation scheme should **replicate as closely as possible** the scheme under which the predictions will be applied. This includes replicating:

- Temporary windows for observation and registration of variables and prediction windows.
- If there are groups in the data, considering whether there will be information available for each group when the prediction is made, or whether it would be necessary to predict for new groups.

3.3. Classification Models: Probabilities and Classes

In machine learning, supervised classification algorithms are systems whose objective is to assign a category or class label to new observations. It is called binary classification when the target variable has two classes (e.g., classifying an email as spam or not spam), and multi-class classification when there are more than two classes (e.g., plant species identification algorithm).

3.3.1. Imbalanced Data

In a classification problem, an imbalanced dataset occurs when the distribution of observations across the known classes is not equally distributed. These types of datasets will have one or more classes with many examples referred to as the majority class, and one or more classes with fewer observations referred to as the minority classes (e.g., groups with less than 1 percent of total observations). These later groups present considerable difficulties for predictive models, as there may be little information about them.

In highly imbalanced data, class predictors can perform poorly (e.g., they never predict the minority class) even if the performance measures are good. Notably, if we always predict the majority class, the accuracy will be equal to the percentage of elements in this class.



Examples

- Consider that you have 1 million data points, with 999,000 in the majority class and 1,000 in the minority class. It may be a good idea to subsample the negatives by a given fraction (say 10 percent) by re-weighting each negative data point by 10.
- Consider that you have 1 million data points, with 999,950 in the majority class and 50 in the minority class. It may be impossible to properly discriminate the 50 observations. Building validation sets makes the situation worse: you cannot validate predictive performance or build a well-performing model. In these cases, it may be best not to build the model until more information is gathered.

Box 12. Class Imbalance Checklist

- (Quantitative) Make **probability predictions** instead of class predictions. These probabilities can be incorporated into the subsequent decision process as such.
- (Quantitative) When the absolute number of minority cases is very small, it can be very difficult to find appropriate information to discriminate that class. More data need to be collected from the minority class.
- (Quantitative) Sub-sampling the dominant class (weighting the cases up to avoid losing calibration) can be a successful strategy to reduce data size and training time without affecting predictive performance.
- (Quantitative) Replicate the minority class to better balance the classes (over-sampling).
- (Quantitative) Some machine learning techniques allow you to weight each class by a different weight so that the total weight of each class is balanced.

3.3.2. Arbitrary Cut-off Point

In classification problems for decision-making, it is recommended that be used instead of classifying the observation only with its most probable class. The output of a probabilistic classification algorithm is a probability distribution over the set of classes. These methods can provide information to the policymaker about the uncertainty regarding the classification.

To make the decision on whether the observation should be classified as positive or negative, the technical team must choose the threshold at which the observation is classified as belonging to each class. A cut-off point of 0.5 is often mistakenly accepted for binary classifications, as this is the default value for many machine learning models. This decision can

have important implications if it is made outside the context of the problem at hand, so it is important that it be discussed and selected taking into consideration the types of errors and their implications.

Box 13. Arbitrary Cut-off Point Checklist

- (Quantitative) Using **probabilistic classification algorithms** is more suitable for decision-making to incorporate uncertainty regarding the classification.
- (Quantitative) Avoid standard probability cut-off points such as 0.5. Choose an optimal interpretation of the predicted probabilities using the receiving operating characteristic curve and other measures to analyze errors.

3.3.3. Adequateness of Assessment Metrics

In classification problems, the cut-off points are taken with criteria related to the context of the decision. Most are constructed by analyzing the classification confusion matrix, as shown in Table 1.

Table 1. Confusion Matrix

		Real	
		Positive	Negative
Predicted	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

Errors in a classification model can be divided into false positives and false negatives. A false positive is an observation for which the model incorrectly predicts the positive class. And a false negative is an observation for which the model incorrectly predicts the negative class. These performance measures can be combined in different ways depending on the use case and the social policy objective. The most commonly used metrics are:

1. **Accuracy:** One of the most used metrics to evaluate classification models is the fraction of predictions correctly made by the model:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

2. **Precision:** Fraction of those observations classified as positive by the model that were actually positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

3. **Sensitivity (Recall):** Fraction of positives observations that the model classified correctly.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4. Specificity: Fraction of negatives observations that the model classified correctly.

$$\text{Specificity} = \frac{VN}{VN + FP}$$

The context should be considered when defining the criteria to assess classification models. For example, if the model is ranking the prevalence of a fatal disease, the cost of not diagnosing a sick person's disease (false negative) is much greater than the cost of sending a healthy person for more tests (false positive). In other words, depending on the application, the cost of false negatives can be very different from the cost of false positives. For this reason, the use of cost-benefit analysis is recommended, since it compares the result of the model in the decision-making context.

These criteria can also be misleading depending on the composition of the training and evaluation database. For instance, where imbalanced data are used, an accuracy of 95 percent can actually mean significant model underperformance. Partial solutions to this issue include using measures that combine precision and sensitivity such as the F1 score or the Precision-Recall curve that can help analyze the trade-off between true positives and false positives in the context of the application.

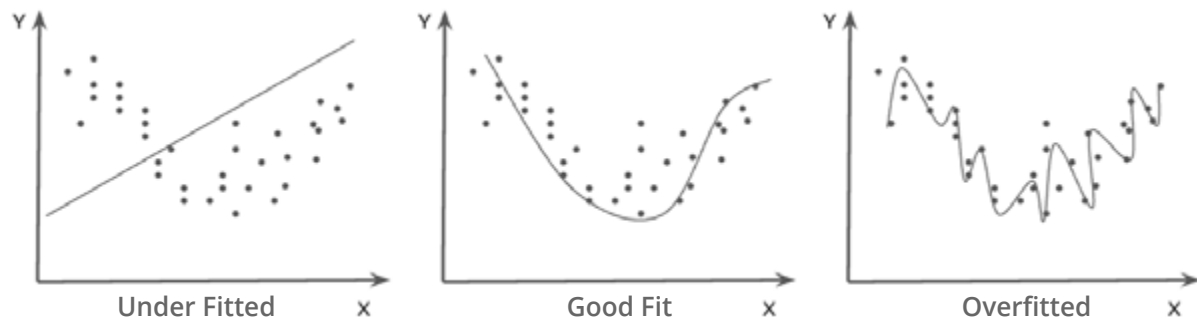
Box 14. Adequacy of the Assessment Metrics Checklist

- (Qualitative) Were the implications of the different types of errors for the specific use case and the correct way to evaluate them questioned?
- (Qualitative) Were the limitations of the model clearly explained? This implies identifying both false positives and false negatives and the implications that a system decision would have on the life of the target population.
- (Quantitative) Was a cost-benefit analysis of the system conducted and compared with the status quo or with the use of other decision-making or decision support strategies? (When possible)

3.4. Under and Overfitting

Generalization refers to the ability of a model to perform accurately on unobserved data during the training process. Generalization is important because the data collected are only a sample, and as such they may be incomplete and noisy. When a model fails to generalize or performs poorly, it is usually due to one of the following related phenomena (Figure 4):

- **Overfitting** occurs when the model memorizes the particularities of the training data but is unable to generalize to unseen examples. A model that is too complex for the available data tends to capture non-informative characteristics as part of the predictive structure. This is often reflected in a model that performs very well on the training data but has a poor performance on the validation dataset.
- **Underfitting** occurs when the model is unable to perform well with the training data or generalize to new data. This happens when individual characteristics of observations are over-grouped and given little weight. An underfitted model tends to ignore patterns in the predictive structure. This is reflected in systematic and identifiable errors – for example, systematic under or overprediction for certain groups or values of the input variables.

Figure 4. Under and Overfitting

Source: Prepared by the author

Box 15. Underfit and Overfit Checklist

- (Quantitative) Overfitting: If necessary, methods should be refined to moderate the overfitting, including such methods as regularization, restricting the functional space of possible models, using more training data, or disturbing the training data (Hastie, Tibshirani, and Friedman 2017).
- (Quantitative) Underfit: Data on protected groups or other sensitive variables should be reviewed to verify that there are no undesirable systematic errors.

3.5. Unquantified Errors and Human Evaluation

In many cases, some biases in the model are not captured by the chosen performance metrics.

For example, a document search system that performs well in performance metrics may systematically return short documents, producing biased results and over-selecting promotional or briefing-type documents. Reasons for this type of bias can range from pre-processing errors – including miscalculated data attributes – to selecting attributes that only consider part of the problem.

3.5.1. Failures Not Measured by the Model

Some algorithms produce low-quality results that escape the lens of the validation metrics. These models may have poor performance when put into production. Reasons for this include:

- Pre-processing errors when calculating predictions.
- Treatment of data that excludes important metrics to make quality or fair predictions.
- Absence of metrics that measure certain types of errors.

This can be a difficult problem to solve, as these errors may be non-visible or not directly measurable. It is necessary to discover these biases or errors outside the technical evaluation context, and if possible include additional evaluation metrics that would capture these problems.

Box 16. Unmeasured Errors and Human Review Checklist

- (Qualitative) Was a human assessment conducted with use case experts to look for known biases or errors? Establishing monitoring schemes that allow for the identification of unmeasured errors or biases is recommended. For example, panels of reviewers can be used to examine predictions and consider whether they are reasonable. These panels must be balanced in terms of user type and expertise and include decision-makers if necessary.

3.6. Fairness and Differential Performance of Predictors

Machine-learning-based methods can produce unfair or discriminatory results for subgroups of the population (Buolamwini and Gebru 2018; Barocas and Selbst 2016; Bolukbasi et al. 2016). This may be caused by all the aforementioned challenges, including poorly designed sourcing and handling of the data and errors in the design of the model.

Examples of differential performance and bias include different acceptance rates for receiving benefits in different groups or detection errors in human faces that are different depending on race.

The evaluation of the results of a decision-making or decision support system is carried out taking into account the objectives of the decision-maker, which may be different from and even contradictory to the objectives from the point of view of the machine learning problem. For example, a decision-maker might sacrifice the overall performance of a model to improve the performance of the model in a subgroup, even though this subgroup is small compared to the population as a whole (e.g., affirmative action to correct existing social gaps).

Although the analysis of the ethical implications in machine learning models and their relationship with a definition of justice is still an open field of study, there is an important strand of literature that seeks to implement mathematical definitions of fairness in the models to describe their impartiality towards subgroups and their ability to make decisions that mitigate unwanted outcomes.

Terms

Protected attribute: A protected characteristic or variable is one for which model predictions should meet a certain fairness criterion. More than one protected variable can exist in a dataset (e.g., age, gender, race, etc.).

3.6.1. Algorithmic Fairness and Inequality

What is meant by “justice” can change according to a culture or tradition and can also be specific to a public policy project or problem. For example, in certain cases policies seek social inclusion through affirmative action – such as diversity quotas and reparation policies – while in other cases these policies are simply based on regional or territorial arguments. These criteria should be integrated in the design process, in the analysis of the training data, during the error evaluation process, and in the output of the system. This process can be separated into two important stages:

- **Algorithmic fairness definition:** Mathematical representation of a specific definition of fairness that is incorporated into the model selection and fitting process. It is important to take into account that different definitions can be exclusive; that is, satisfying one may imply not satisfying the others (Verma and Rubin 2018).
- **Algorithmic inequality:** Technical flaws in the models that produce disparity of results for protected groups, and that must be evaluated under the definition of algorithmic fairness previously determined.

The objective of the model developer is to establish clear guidelines to avoid deficiencies in the model from producing undesirable disparities for subgroups of a protected variable (e.g., gender, race, or level of marginalization). For this, it is necessary to select a definition of algorithmic fairness in advance. The following three definitions of algorithmic justice are among the most widely used (although others can be defined depending on the particular problem and objectives of the decision-maker):

1) *Omission of protected variables and demographic parity*

Two widely contested disparity-prevention strategies between groups of a protected variable are to *ignore* the variable and to aim to achieve *demographic parity* in predictions.

The first strategy is intended to eliminate the possibility of disparity by **not** including variable in the model. This approach usually fails to solve the problem because:

- Typically, there are other attributes associated with that can produce similar results, even if is not considered (e.g., geographic area or postal code and socioeconomic level).
- There may be important reasons to include in predictive models. For example, in the case of blood pressure, there are variations in racial groups () in terms of predisposition to high blood pressure (Lackland 2014), so a model that evaluates the risk of heart attack would be more accurate and appropriate if it includes variable .

In the second strategy, *demographic parity* establishes that each segment of a protected class (e.g., gender or certain age ranges) must obtain a positive result in the same proportion (such as the allocation of school scholarships). This is undesirable in and of itself: for example, if we wanted to construct a classifier for a certain disease, we would need to consider that it is possible that women and men are affected differently. However, demographic parity can be an objective of decision-makers in and of itself, which must be taken into account when building the model.

2) *Equality of opportunity*

The concept of *equality of opportunity* (Hardt, Price, and Srebro 2016) is one less dependent on decision-makers' objectives. It refers to the predictive performance across different groups defined by a protected variable (Verma and Rubin 2018). If is the variable we want to predict and is our prediction, we say that our prediction satisfies equality of opportunity when and are independent given the true value

This means that should not influence the prediction when we know the true value , or, in other words, belonging or not belonging to the protected group A should not influence the result of the classification.

Predictors that significantly deviate from this criterion are likely to produce disparities associated with the protected variable A. Under an equality of opportunity assumption, the predic-

tive error rates for each subgroup of A should be similar. For instance, for binary classification models, the false positive and false negative rates should be approximately equal.

For example, suppose you want to create a system to select the recipients of a prestigious scholarship. The institution defines membership in an indigenous community as a protected variable that, for simplicity, we will assume takes two values: indigenous or not indigenous. The predictor satisfies *equality of opportunity* when both the false positive and false negative rates are the same for indigenous people and for non-indigenous people.

3) Counterfactual justice

This measure considers that a predictor is “fair” if its result remains the same when the value of the protected attribute is modified (such as introducing a change in race, gender, or other condition).

In practice, there is no single algorithmic justice measure that works for all problems and, in most cases, seeking compliance with one definition implies not fully complying with the others. The choice should be made considering the context, and the reasons behind it should be justified and documented.

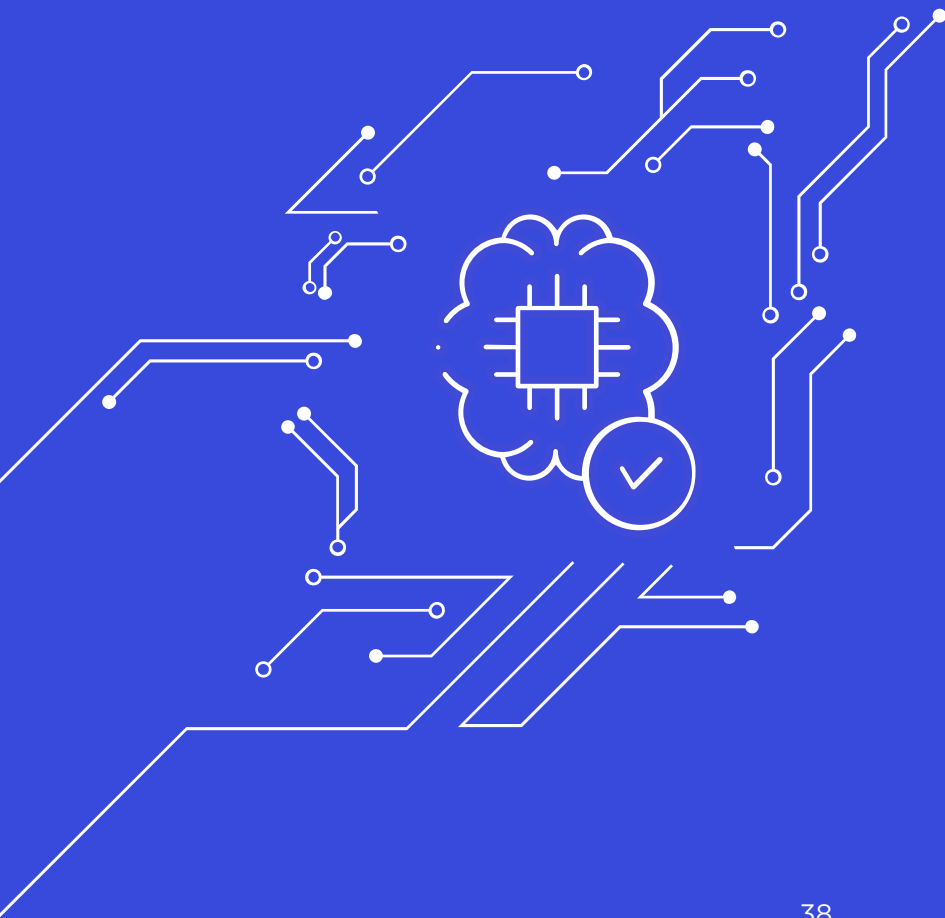
Box 17. Algorithmic Fairness and Inequality Checklist

- (Qualitative) Identify protected groups or attributes (e.g., age, gender, race, poverty level, etc.)
- (Qualitative) Was the algorithmic fairness criterion to be used in the model defined with experts and decision-makers?
- (Quantitative) When protected attributes exist, an assessment must be made of how far predictions deviate from the chosen algorithmic fairness definition.
- (Quantitative) There must be proper post-processing of predictions if necessary to achieve the chosen algorithmic fairness criterion.
- (Quantitative) In the case of classification models, cut-off points for different subgroups can be adjusted to achieve the chosen algorithmic fairness criterion.
- (Quantitative) Collect more relevant information from protected subgroups (both cases and characteristics) to improve predictive performance for minority groups.

Activity: At the end of this phase, it is recommended that the Model Development sections of the [Model Card](#) be filled out and that a discussion be held with the public policy decision-maker.



4. DEPLOYMENT AND MONITORING



4. Deployment and Monitoring

When machine learning methods are used to make decisions, it is necessary to:

- Monitor model performance and variables used over time.
- Monitor, in particular, undesirable results that may result from user interaction with the systems.
- Evaluate the data collection and processing process to improve performance or evaluate results.

4.1. Performance Degradation


The performance of a model can degrade over time for multiple reasons, including:

- Machine learning models that assume a static relationship between the input and output variables can degrade the quality of their predictions due to changes in the underlying relationships between the variables.
- Changes in data collection methods and methodological adjustments can harm model performance. For example, in the case of administrative data, a ministry could change the data collection processes, digitize the systems, or systematize data cleaning or processing in a way that makes a given model obsolete.
- Model degradation also occurs in interactive systems – where the system and its users form a closed feedback loop – as users can only interact with elements that are decided by the system, and vice versa.

To mitigate these possible errors, it is necessary to monitor input variables and their relation with model behavior, and to update assumptions accordingly, jointly with decision-makers and domain experts.

The behavior of error metrics over time should also be monitored, including total positive and negative rates disaggregated by protected group or variable of interest and the distribution of predictions over time.

Box 18. Performance Degradation Checklist

-  Performance degradation:
- (Qualitative) Is there a plan to monitor the performance of the model and the collection of information over time?
 - (Quantitative) Monitor various metrics associated with predictions in predefined subgroups (including protected variables).
 - (Quantitative) Monitor drift in variable distributions with respect to the training set.
 - (Quantitative) Monitor changes in the data collection and processing methodology that may reduce the quality of predictions.
 - (Qualitative) When feasible, a fraction of the predictions should be examined by humans and scored according to some pre-defined rubric or scale for each variable of interest.

4.2. Experiments to Evaluate Model Effectiveness

The data collection mechanisms to maintain the model should be designed in a way that keeps the model up to date and reaching optimal performance.

Improvements to the data collection process and to the overall performance of the model could be difficult to assess without strong counterfactuals. In this sense, experimental tests – of type A/B, for example – should be used when possible to understand the desirable or undesirable consequences of using the model (Vaver and Koehler 2011).

Box 19. Experiments and Data Collection Checklist

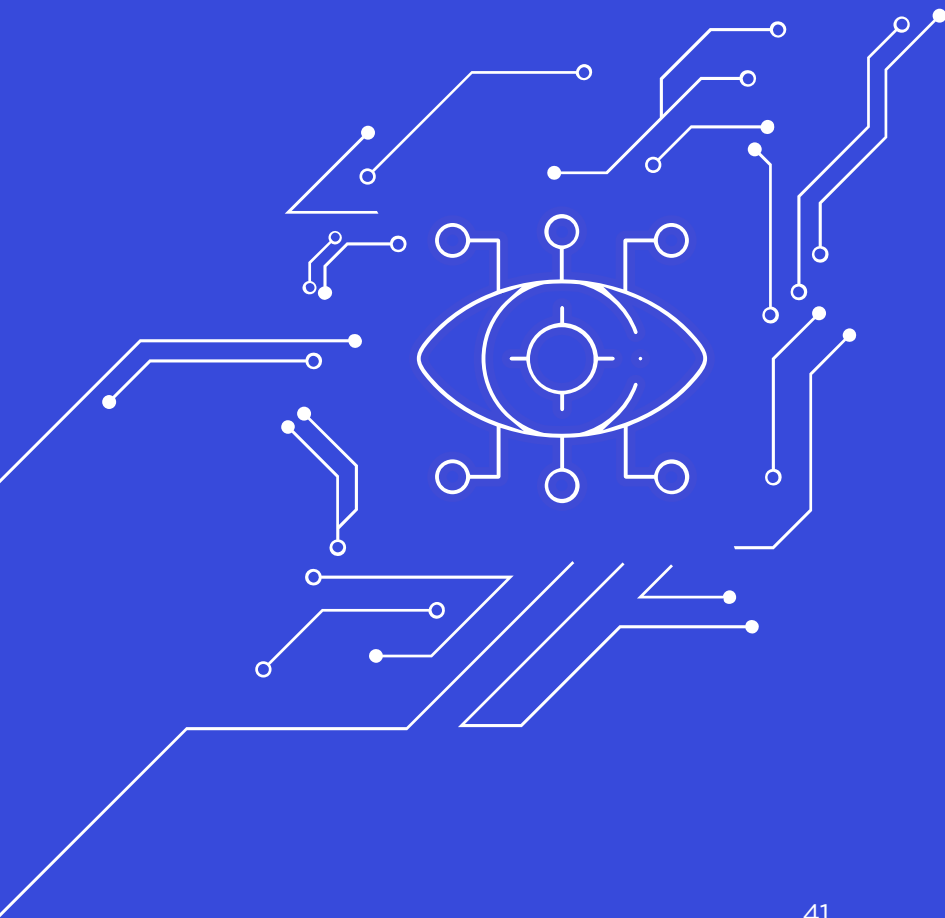
- (Quantitative) When possible, plan to assign randomized (or status quo) treatments to some units under experimental designs. Make performance and behavior comparisons between this sample and the results under the algorithmic regime.
- (Quantitative) Identify unobserved variables and seek ways to measure them. If possible, re-fit the model and evaluate model performance using this information.



Activity:

At the end of this phase, it is recommended that the Use and Monitoring Section of the [Model Card](#) be filled out and that a discussion be held with the public policy decision-maker.

5. ACCOUNTABILITY



5. Accountability

Regulations such as the European Union's General Data Protection Regulation (GDPR) define accountability as the requirement for organizations to put in place appropriate technical and organizational measures and to be able to demonstrate what they did and its effectiveness when requested to do so.

Although the development of technical standards and norms for AI systems is still a pending task for the AI community, this toolkit has described the main technical aspects and measures to avoid and mitigate bias during the AI lifecycle. However, several challenges remain that are related to the social and legal requirements that the use of these systems entails in real-world applications.

This section reviews the concepts of interpretability, explainability, and traceability of AI systems.

5.1. Interpretability and Explainability of Predictions

5.1.1. Interpretability

There is no concrete mathematical definition of **interpretability** (Molnar 2019). It generally refers to the degree to which a human can consistently predict a model's results (Kim, 2016). The more interpretable a model is, the easier it is for an individual to understand the process that led to a certain decision (Miller 2019). A model with high interpretability is desirable in a high-risk social policy application where the criterion of accountability becomes fundamental.

There are several reasons why having some degree of interpretability in the models used to make decisions is important from a technical perspective (Molnar 2019):

1. To learn more about the problem, including causal relationships.
2. To achieve social acceptability of the use of the model.
3. To detect potential biases in the algorithm.
4. To debug and improve models.

Complex algorithms such as deep neural networks can have millions of relationships between their parameters, so obtaining model interpretability in these algorithms is still an open field in machine learning. When high interpretability is necessary, the use of intrinsically interpretable methods such as linear regression, logistic regression, and decision trees is recommended.

5.1.2. Explainability of Individual Predictions

In many cases, it may be legally or ethically necessary to provide **explanations** of how a model reached certain conclusions (e.g., why a person was not granted a loan, or why someone does not qualify for a social program).¹³

In research areas such as computer vision and natural language processing, the most successful implementations are usually developed with highly complex models, such as deep neural networks, that are not very transparent as to the underlying assumptions used to reach a given prediction (Carrillo, Cantú, and Noriega 2020).

¹³ In the European Union, for example, Article 22 of the GDPR describes the right of a person to challenge the decision of a system, especially when it is automatic.

While this is an area of ongoing research, several methods already exist to increase the explainability of predictions (Molnar 2019). Methods such as counterfactual explanations (Wachter, Mittelstadt, and Russell 2017), Shapley values (Lundberg and Lee 2017), and integrated gradients for deep networks (Sundararajan, Taly, and Yan 2017) can be used.

Box 20. Explainability of Individual Predictions Checklist

- (Qualitative) Were the legal and ethical explainability requirements in the project's context analyzed?
- (Qualitative) Is there a process in place to provide explanations to particular individuals about why a decision was made?
- (Qualitative) Were the pros and cons of the algorithms discussed according to their level of interpretability and explainability in order to choose the most appropriate one?
- (Quantitative) For simpler models (e.g., linear or decision trees), ad-hoc explanations can be constructed.
- (Quantitative) For deep neural networks, use available methods such as counterfactual explanations, Shapley values, or integrated gradients.

5.1.3. Parsimonious Models

It is widely held that an ML model is always better when more covariates are used; this is partially correct as the model can find patterns among the interrelationship of variables. However, when interpretability is taken into account, more parsimonious methods that use fewer but relevant features are preferable to models that use many but perhaps less relevant features.

Potential biases can occur when using data characteristics or variables that, although valid for a given time and dataset, are easily susceptible to change when the data-generating process evolves. Algorithms or predictive methods that use many irrelevant attributes are at higher risk of failing both explicitly and silently when data sources or data-generating processes change.

Examples may be the use of variables that are being actively influenced by some policy that will not continue in the future or learning characteristics from a non-exhaustive training set (e.g., in image recognition, recognizing animal species by the context in which information was collected, such as a zoo, camera trap, landscape, etc.).

This type of bias harms a system's explainability and may be difficult to detect, but parsimonious methods and expert knowledge can mitigate the risk.

Box 21. Parsimonious Models Checklist

- (Qualitative) Including all available features to build and train a model may increase the risk of disproportionately affecting users. The variables to be included in the learning process must have some theoretical support or explanation of why they can help in the prediction task.
- (Quantitative) More parsimonious methods that use fewer, but relevant, features are preferable to models that use many, but less relevant, features.
- (Quantitative) Methods such as partial dependence plots (Friedman 2001) or permutations-based importance (Breiman 2001; Molnar 2019) can point to problematic variables that are heavily weighted in prediction against past observations or expert knowledge.

5.2. Traceability

A data-to-decision process that is not **traceable** is one whose execution steps are poorly documented: they include poorly specified processes or operator decisions, extract data from undocumented or inaccessible sources, omit necessary code or materials, or do not give the necessary information to ensure the reproducibility of results.

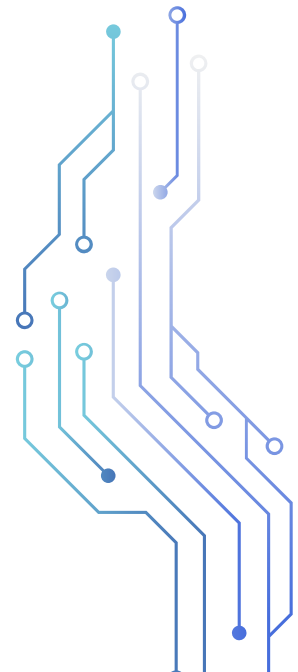
Traceability allows users to understand the processes followed by an AI system to arrive at an outcome, including the system's shortcomings and limitations. When there is little traceability in a model, the risks outlined throughout this document can be difficult to identify and may even be exacerbated. In contrast, all steps from data collection to decision-making are clearly documented and unambiguously specified in a traceable project.

Box 22. Traceability Checklist

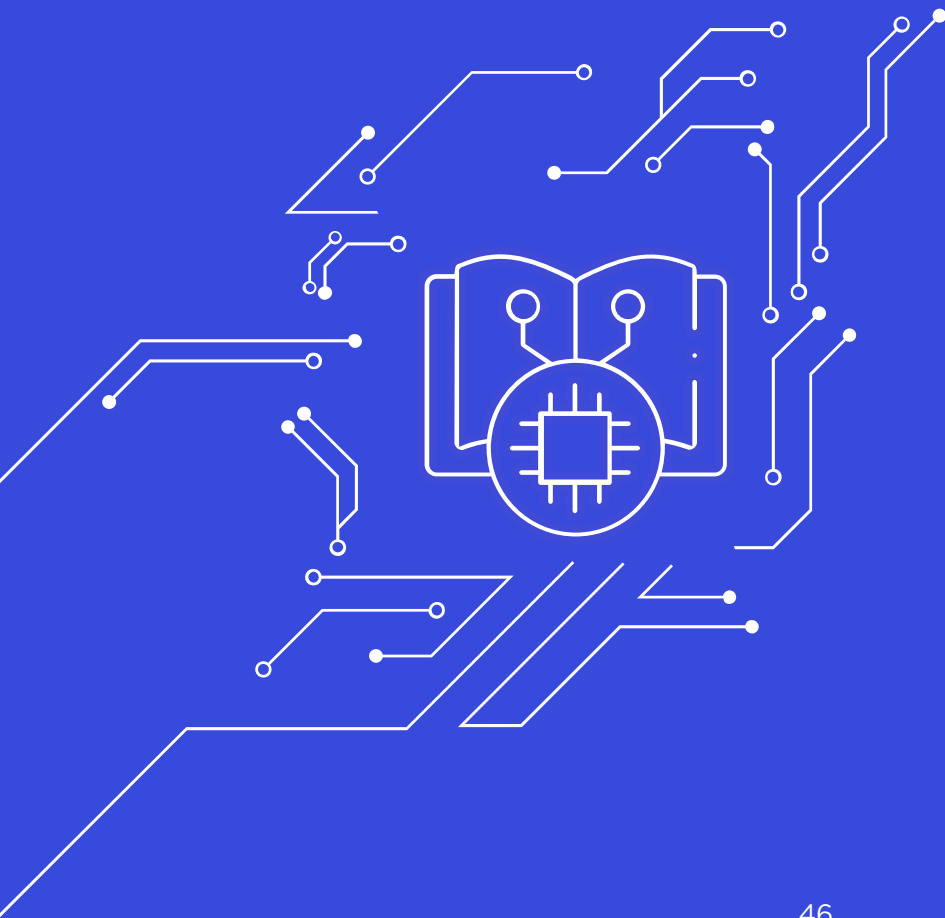
- (Quantitative) Is the AI lifecycle well documented (including data provenance and collection mechanisms, infrastructure used, model dependencies and code, metrics, and interpretation of results)? The documentation should include:
 1. **Data sources**, including dataset metadata, data collection processes and data processing information. ([see Tool 2](#))
 2. **Complete and appropriately documented** code, defining necessary libraries and their appropriate versions, to allow any third party to understand the purpose of each part of the code.
 3. **Information** on how the code should be executed, including detailed documentation of the parameters and computing requirements. This information must guarantee **reproducibility of the original results** by a third party.
 4. **Information on how the results of the computational process were used and included** in the decision-making process.
 5. **Information about the monitoring strategy**, including details about performance metrics and thresholds as well as expected model behavior and mitigation actions.

Ideally, the aforementioned steps should be replicable by a third party with minimal or no intervention from the original system creators and operators.

- (Qualitative) Have the deficiencies, limitations, and biases of the model been communicated to stakeholders so that they are considered in decision-making and decision support?
- (Qualitative) Has the technical team completed the Data Profile ([see Tool 2](#)) and the Model Card ([see Tool 3](#)), and has a process for continuous updating of these tools been defined?



TOOLS



Tool 1: Robust and Responsible AI Checklist

This tool consolidates the main concerns by risk dimension of the AI lifecycle. The checklist must be reviewed continuously by the technical team accompanied by the decision-maker (Fritzler 2015; drivendata 2019).

Planning and Design

☒ Correct definition of the problem and the public policy response

- (Qualitative) Is the public policy problem clearly defined?
- (Qualitative) Describe how this problem is currently being addressed – considering responses by related institutions – and how the use of AI would improve the government response to this problem.
- (Qualitative) Were the protected groups or protected attributes identified within the project (e.g., age, gender, education level, race, level of marginalization, etc.)?
- (Qualitative) Were the actions or interventions to be carried out based on the result of the AI system defined?

☒ AI Principles

- (Quantitative) Has the need for an AI system been justified, considering other possible solutions that do not require the use of personal data and automated decisions?
- (Quantitative) Is there evidence that both public policy action and the recommendation of the AI system will result in a benefit to people and the planet by driving inclusive growth, sustainable development, and well-being?
- (Qualitative) For the implementation of these technologies, have there been similar previous projects, and have they been reviewed?
- (Quantitative) Have you considered minimizing the exposure of personally identifiable information (e.g., by anonymizing or not collecting information not relevant to the analysis)?

Lifecycle

☒ Data Collection and Processing

Data quality and relevance of the available data

- (Qualitative) Discuss possible historical social inequalities in the use case with specialists in the field.
- (Quantitative) Perform an exploratory analysis of the available data with which the model will be trained to identify possible historical biases or undesirable states.

Poor Correspondence between Ideal and Available Variables

- (Qualitative) The ideal target variables should be clearly stated. The collected/available variables must be analyzed to understand how suitable they are to substitute for the target variable. Systematic biases or validity of the proxy metric should be identified.
- (Qualitative) Has the use of the selected response variable been clearly justified for the purposes of the intervention?

✓ Data qualification and completeness for the target population***Probabilistic and Natural Samples***

- (Qualitative) Have the possible differences between the database and the population for which the AI system is being developed been analyzed? (Use literature related to the topic and information from experts. Study in particular unmeasured selection biases.)
- (Quantitative) Although models can be built with various data sources, designed or natural, validation should ideally be carried out with a sample that allows statistical inference to the target population. The validation sample must appropriately cover the target population and sub-populations of interest.

Missing or incomplete attributes

- (Qualitative) Has an analysis of missing values and variables been performed?
- (Qualitative) Has it been determined whether there are important omitted variables for which there are no associated measurements? (If any)
- (Qualitative) Have the reasons for the missing observations been identified? (If any)

✓ Causal comparison

- (Qualitative) Understand and describe the reasons why the response variable is correlated with known and unknown variables. Describe possible biases based on expert knowledge and analysis.
- (Qualitative) In the event that no work was done to ensure causality in the results, were the limitations of the results explicitly communicated to the public policy decision-maker?

Model Building and Validation**✓ Absence or inappropriate use of validation samples**

- (Quantitative) Were the validation and test samples constructed properly, considering an appropriate size, covering subgroups of interest and protected subgroups, and avoiding information leaks during its implementation?

✓ Data leakage

Training-Validation Data Leak

- (Quantitative) Any processing and preparation of training data should avoid using the validation or test data in any way. A solid barrier must be maintained between training versus validation and testing. This includes data recoding, normalizations, selection of variables, identification of outliers, and any other type of preparation of any variable to be included in the models. This also includes sample weights or balances based on oversampling/undersampling.

Target Leakage

- The validation scheme should replicate as closely as possible the scheme under which the predictions will be applied.

✓ Probabilities and classes

Imbalanced Data

- (Quantitative) Make probability predictions instead of class predictions. These probabilities can be incorporated into the subsequent decision process as such.
- (Quantitative) When the absolute number of minority cases is very small, it can be very difficult to find appropriate information to discriminate that class. More data need to be collected from the minority class.
- (Quantitative) Sub-sampling the dominant class (weighting the cases up to avoid losing calibration) can be a successful strategy to reduce data size and training time without affecting predictive performance.
- (Quantitative) Replicate the minority class to better balance the classes (over-sampling).

Arbitrary Cut-off Point

- (Quantitative) Using probabilistic classification algorithms is more suitable for decision-making to incorporate uncertainty regarding the classification.
- (Quantitative) Avoid standard probability cut-off points such as 0.5. Choose an optimal interpretation of the predicted probabilities using the receiving operating characteristic curve and other measures to analyze errors.

Adequateness of assessment metrics

- (Qualitative) Were the implications of the different types of errors for the specific use case, as well as the correct way to evaluate them, questioned?
- (Qualitative) Were the limitations of the model clearly explained? This implies identifying both false positives and false negatives and the implications that a system decision would have on the life of the target population.
- (Quantitative) Was a cost-benefit analysis of the system conducted and compared with the status quo or with the use of other decision-making or decision support strategies? (When possible)

Underfit and Overfit Checklist

- (Quantitative) Overfitting: If necessary, methods should be refined to moderate the overfitting, including such methods as regularization, restricting the functional space of possible models, using more training data, or disturbing the training data (Hastie, Tibshirani, and Friedman 2017).
- (Quantitative) Underfit: Data on protected groups or other sensitive variables should be reviewed to verify that there are no undesirable systematic errors.

✓ Unquantified errors and human evaluation***Failures Not Measured by the Model***

- (Qualitative) Was a human assessment conducted with use-case experts to look for known biases or errors? Establishing monitoring schemes that allow for the identification of unmeasured errors or biases is recommended. For example, panels of reviewers can be used to examine particular predictions and consider whether they are reasonable. These panels must be balanced in terms of user type and expertise, including decision-makers if necessary.

✓ Fairness and differential performance***Algorithmic Fairness and Inequality***

- (Qualitative) Was the algorithmic fairness criterion to be used in the model defined with experts and decision-makers?
- (Quantitative) When protected attributes exist, an assessment must be made of how far predictions deviate from the chosen algorithmic fairness definition. (e.g., tested for disparate error rates)?
- (Quantitative) In the case of classification models, cut-off points for different subgroups can be adjusted to achieve the chosen algorithmic fairness criterion.

Deployment and Monitoring**✓ Performance degradation:**

- (Qualitative) Is there a plan to monitor the performance of the model and the collection of information over time?
- (Quantitative) Monitor various metrics associated with predictions in predefined subgroups (including protected variables).
- (Quantitative) Monitor drift in variable distributions with respect to the training set.
- (Quantitative) Monitor changes in the data collection and processing methodology that may reduce the quality of predictions.
- (Quantitative) When possible, plan to assign randomized (or status quo) treatments to some units under experimental designs. Make performance and behavior comparisons between this sample and the results under the algorithmic regime.
- (Quantitative) Identify unobserved variables and seek ways to measure them. If possible, re-fit the model and evaluate model performance using this information.

Accountability

☒ Interpretability and explanation of predictions

Explainability of Individual Predictions

- (Qualitative) Were the legal and ethical explainability requirements in the project's context analyzed?
- (Qualitative) Is there a process in place to provide explanations to particular individuals about why a decision was made?
- (Qualitative) Were the pros and cons of the algorithms discussed according to their level of interpretability and explainability to choose the most appropriate one?

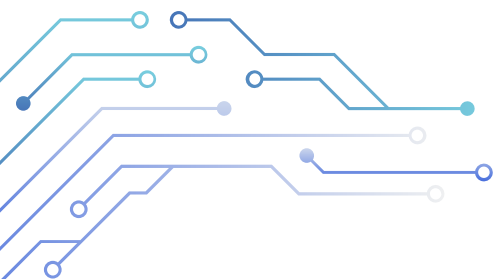
Parsimonious Models

- (Qualitative) Including all available features to build and train a model may increase the risk of disproportionately affecting users. The variables to be included in the learning process must have some theoretical support or explanation of why they can help in the prediction task.
- (Quantitative) More parsimonious methods that use fewer, but relevant, features are preferable to models that use many, but less relevant, features.
- (Quantitative) Methods such as partial dependence plots (Friedman 2001) or permutations-based importance (Breiman 2001; Molnar 2019) can point to problematic variables that are heavily weighted in prediction against past observations or expert knowledge.

☒ Traceability

- (Quantitative) Is the AI lifecycle well documented (including data collection, infrastructure used, dependencies, code, metrics, and interpretation of results)?
- (Qualitative) Have the deficiencies, limitations, and biases of the model been communicated to stakeholders so that they are considered in decision-making/decision support?

(Qualitative) Has the technical team completed the Data Profile ([see Tool 2](#)) and the Model Card ([see Tool 3](#)), and has a process for continuous updating of these tools been defined?



Tool 2: Data Profile

The Data Profile is an exploratory analysis that provides information to evaluate the quality, integrity, temporality, consistency, and possible biases of a dataset that will be used to train a machine learning model (Gebru et al. 2018).

Data and Input Collection and Origin

- Name of dataset used.
- What institution created the dataset?
- For what purpose did the institution create the dataset used?
- What mechanisms or procedures were used to collect the data (e.g., household survey, sensor, software, API)? Does they comply with existing data protection regulations?
- What is the scale of the dataset?
- Obtain documentation for each variable within the dataset. Provide a short description, including its name and type, what it represents, how it is measured, etc.

Data and Input Domains

- What is the data domain (e.g., proprietary, public, personal)?
- If personal, are the data identified, pseudonymized, unlinked pseudonymized, anonymized, or aggregated?
- If proprietary, are any intellectual property rights considerations?

Data and Input Structure

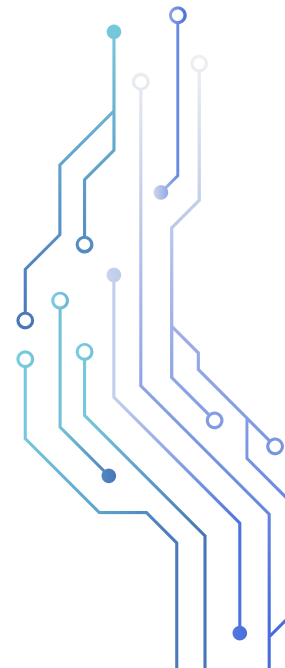
- Are the data static or dynamic? If dynamic, how often will they be updated?

Data Quality & Qualification

- How were the data obtained (observed, derived, synthetic, or provided by individuals or organizations)?
- Is the data representative of the population of interest?
- Analyze spatial and temporal coverage of the data.
- Analyze coverage of protected groups (sex, race, age, etc.).
- Describe the type of sampling used to obtain the data.
- Describe the important dimensions in which the data sample may differ from the population, in particular unmeasured selection biases. Use literature related to the subject and information from experts.
- Identify possible “undesirable states” in the data that could lead to prejudicial biases and inequities for a given subgroup, or any other pattern that is considered suboptimal or undesirable from a social policy point of view.
- Are there any missing values? If so, explain the reasons why this information is not available (this includes information intentionally removed). Identify reasons

for missing data and think about whether the missing data are associated with the variable to predict. Document any imputation processes used to substitute for missing data.

- Capture frequency (weekly, monthly, daily) or average number of observations per individual. What version of the dataset is being used?
- Is this the most appropriate dataset available given the problem at hand?



Tool 3: Model Card

The rubric presented here is a follow-up card that summarizes the main characteristics of a machine-learning-based decision-making or decision support system and highlights the main assumptions, the most important characteristics of the system, and the mitigation measures implemented (Mitchell et al. 2019).

Planning and Design

1. Basic information
 - People who developed the model, date, version, type
2. Use cases
 - Background
 - Target population and forecast horizon
 - Actors and components that will interact with the results
 - Use cases considered during development
 - Uses not considered and related warnings
 - Definition of protected groups

Data Collection and Processing

3. Training data
 - Dataset used and its labeling
 - Preprocessing or data preparation steps
 - Potential biases and shortcomings depending on the use case (2)

Model Building and Validation

4. Modeling
 - Algorithms that were used for training, assumed parameters or constraints
 - Input or assumptions made using expert knowledge
 - Data interaction (no interaction, active interaction, passive interaction)
5. Performance metrics
 - Technical metrics used to select and evaluate models
 - Cost-benefit analysis of the model for its use case according to (2)
 - Definition of protected groups and selected fairness measures
6. Validation data
 - Datasets used and their labeling
 - Pre-processing steps
 - Evaluation of adaptation of validation data according to the use case (2)

- Potential biases and shortcomings depending on the use case (2)
7. Quantitative analysis summary
 - Validation error reported
 - Summary of cost-benefit analysis
 - Report of fairness measures for protected groups

Deployment and Monitoring

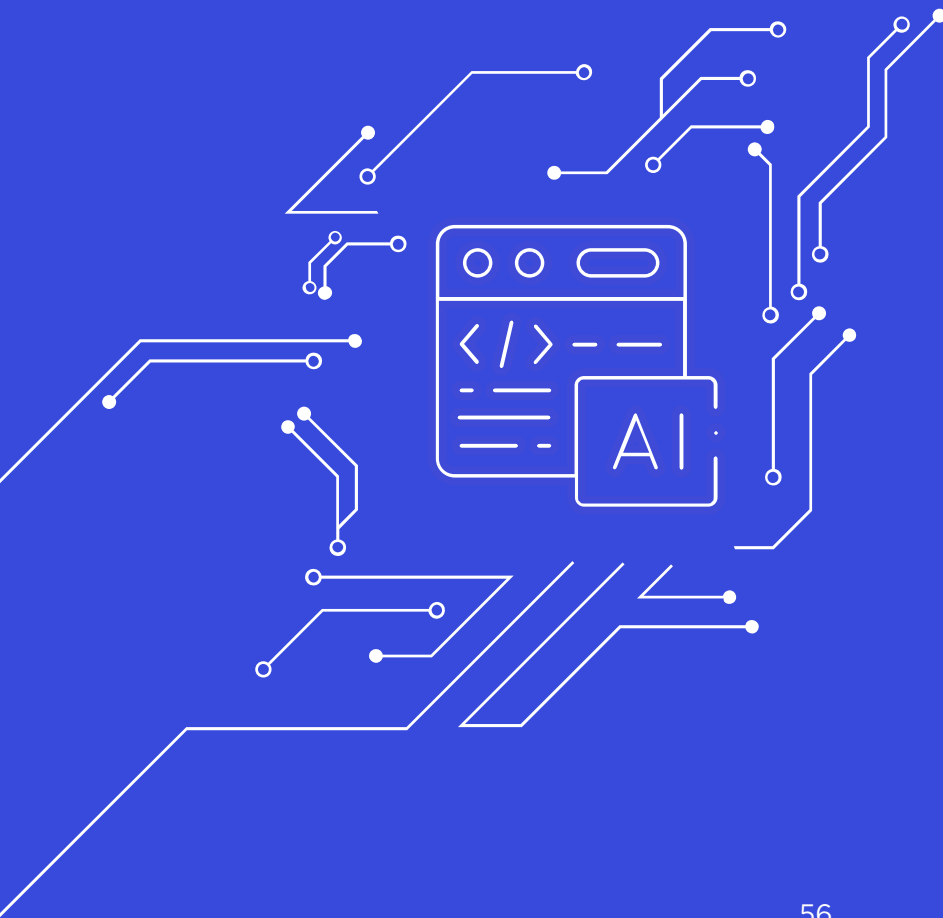
8. Monitoring recommendations
 - Monitoring and improvement strategy in production
 - Human monitoring strategies (if applicable)

Accountability

9. Explainable predictions
 - Strategy to explain particular predictions
 - Strategy to understand the importance of different attributes
10. Other ethical considerations, recommendations, and warnings



WORKBOOKS



Workbooks

This section shows several examples of the challenges and solutions explained in the main document. Different types of models (linear, tree-based, and others) and different implementations (R, keras, xgboost) are used to show that these problems arise regardless of the choice of particular tools.

Booklets use decimal point notation to maintain consistency with packages that use it. The R programming language is used along with the following packages: tidyverse, recipes, themeis, rsample, parsnip, yardstick, workflows, tune, knitr, and patchwork.

All material is reproducible according to instructions in this [repository](#)¹⁴, which contains a Dockerfile that describes the infrastructure dependencies for its replication.

Data Collection and Processing

Data Quality and Relevance of the Available Data

Using models that predict the wrong metric can lead to wrong decisions. Sometimes the problem is clear, as when the proxy metric has obvious shortcomings, and other times it can be more subtle.

The example shown below seeks to predict the demand for a certain product (let's think about vaccines or some medicine) in order to make supply decisions.

There is historical data on inventory (80 weeks) and sales, and a *predictor* variable associated with sales (in the case of vaccines it could be temperature) and another on inventory depletion. We separate the data in training and testing, fitting the model with the subset of training data. In this case, a linear model is used with the dependent variable sales and week covariates and the *predictor* covariate.

```
train <- sales%>% filter (week < 60)

test <- sales%>% filter (week> = 60, week <= 80)

train%>% select (-demand) %>% head() %>% kable()
```

Week	Inventory	Sales	Forecast	Depletion
1	153	110	-27.7014124	0
2	170	148	0.7664636	0
3	158	130	-15.2606032	0
4	162	142	4.2461227	0
5	159	159	28.5107593	1
6	162	162	14.8895964	1

¹⁴ <https://github.com/EL-BID/Manual-IA-Responsable>

```
linear_mod <- lm (sales ~ week + predictor, data = sales)

mod_linear

##

## Call:

## lm (formula = sales ~ week + predictor, data = sales)

##

## Coefficients:

## (Intercept) week predictor

##      140.9935      0.8166      0.5535
```

We evaluate the prediction error.

```
preds <- predict (mod_linear, newdata = test)

round (mean (abs (preds - test$sales)) / mean (test$sales), 3)

## [1] 0.04
```

The percentage error is low. The fitted data and predictions look as follows:

```
preds <- predict (mod_linear, newdata = sales)

sales_long <- sales%>% mutate (pred = preds)%>%

  pivot_longer (cols = all_of (c ("sales", "pred")), names_to = "type", val-
ues_to = "units")

ggplot (sales_long%>% mutate (units = ifelse (type == "sales" & week > 80, NA,
units)),

aes(x = week, y = units, group = type, color = type)) +

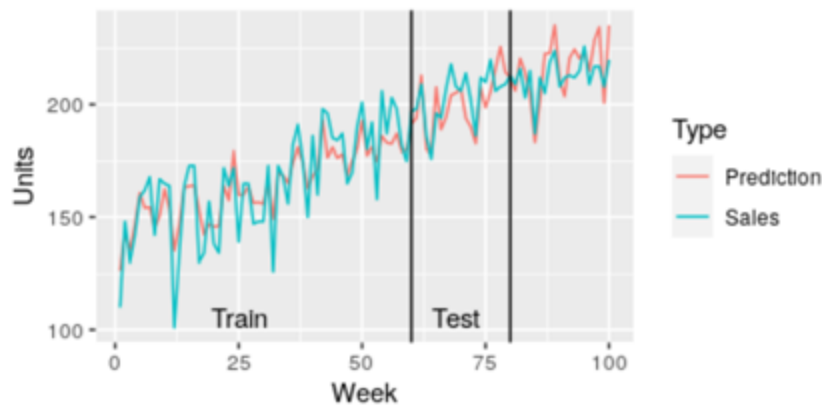
geom_line() +

  geom_vline (xintercept = 80) +

  geom_vline (xintercept = 60) +

  annotate("text", x = 25, y = 105, label = "train") +

  annotate("text", x = 69, y = 105, label = "test")
```



But making demand or inventory decisions with this type of model is wrong. The reason is that there is a difference between the ideal variable (real demand for medicines) and the observed variable (sale of medicines). The difference is that there are inventory depletions, that is, periods when, although there was demand, there was not enough inventory for all buyers. This is marked in red in the following graph.

```

preds <- predict (mod_linear, newdata = sales)

sales_long <- sales%>% mutate (pred = preds)%>%

  pivot_longer (cols = all_of (c ("sales","pred")), names_to = "type",
values_to = "units")

ggplot (sales_long%>% mutate (units = ifelse (type == "sales" & week > 80, NA,
units)), aes (x = week)) + >

geom_line (aes (group = type, color = type, y = units)) +

  geom_point (data = filter (sales, depletion == 1, week < 80), aes (y = sales),
color = "red") +

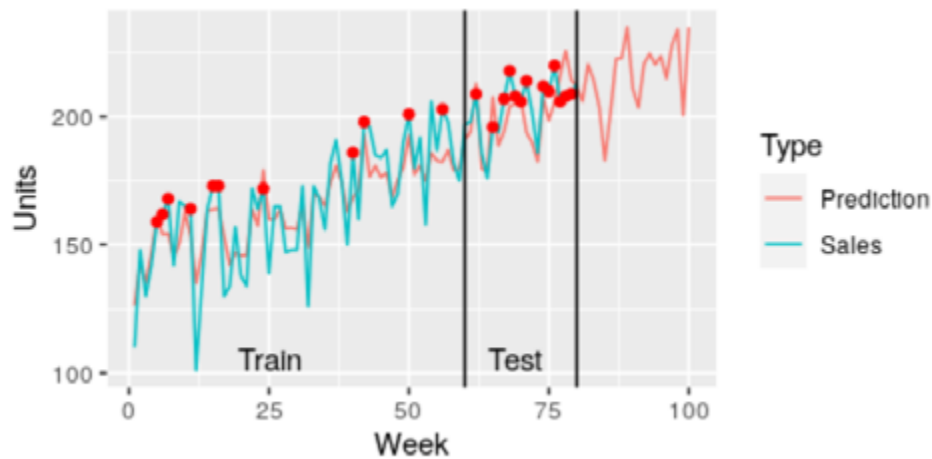
geom_vline (xintercept = 80) +

  geom_vline (xintercept = 60) +

annotate("text", x = 25, y = 105, label = "train") +

annotate("text", x = 69, y = 105, label = "test")

```



If we were to use the policy suggested by the predictions (e.g., 5 percent more), we would see the sales in the first graph below. However, if we used an inventory policy with 280 units, we would observe the following:

```

preds <- predict (mod_linear, newdata = sales)

sales_obs <- sales%>% mutate (pred = preds)%>%
  mutate (inventory = 1.05 * pred)%>%
  mutate (sales = ifelse (week>; 80 , pmin (inventory, demand), sales))

sales_long <- sales_obs%>%
  pivot_longer (cols = all_of (c ("sales", "pred")), names_to = "type", va-
lues_to = "units")

g1 <- ggplot (sales_long, aes (x = week)) +

geom_line (aes (group = type, color = type, y = units)) +
  geom_point (data = filter (sales_obs, sales == inventory, week> 80),
aes (y = sales), color = "red") +
  geom_vline (xintercept = 80) + labs (subtitle = "Inventory: Predictions +
5%")

preds <- predict (mod_linear, newdata = sales)

sales_obs <- sales%>% mutate (pred = preds)%>%
  mutate (inventory = 280)%>%
  mutate (sales = ifelse (week>; 80 , pmin (inventory, demand), sales))

sales_long <- sales_obs%>%
  pivot_longer (cols = all_of (c ("sales", "pred")), names_to = "type", va-
lues_to = "units")

```

```

g1 <- ggplot (sales_long, aes (x = week)) +

geom_line (aes (group = type, color = type, y = units)) +

  geom_point (data = filter (sales_obs, sales == inventory, week> 80),
aes (y = sales), color = "red") +
  geom_vline (xintercept = 80) + labs (subtitle = "Inventory: 300 cte
units")

g1/g2

```

Therefore:

- Prediction policy exacerbates the burnout problem.
- An unintended use of data without considering its generating process can lead to large errors in decisions.
- In this case, the confusion comes from not separating the concepts of demand and sales. Other more suitable demand indicators or models would help solve the problem.
- Simplistic solutions such as only taking data where stockouts do not occur can make the situation even worse, as they increase bias (we select weeks where sales tend to be low) and reduce accuracy.

Natural Samples and Bias

When the training samples are different from the populations to which the models are to be applied, there are difficulties in correctly validating the predictions.

Natural Samples: Poor Representativeness

For this example, data from the national household income and expenditure survey in Mexico will be used (INEGI 2014) to simulate a scenario that we want to exemplify.

```

set.seed (128)

survey_entry <- read_csv("data / enigh-example.csv")

income_data <- income_survey%>%

mutate (num_focos = FOCOS)%>%

  mutate (income_miles = (INGCOR / 1000))%>%

  mutate (cell_tel = ifelse (SERV_2 == 1, "Yes", "No"))%>%

mutate (piso_firme = ifelse (PISOS!= 1 | is.na (PISOS), "Yes", "No"))%>%

mutate (washer = ifelse (WASH!= 1 | is.na (LAVAD), "Yes", "No"))%>%

  mutate (car = VEHI1_N>; 0)%>%

mutate (marginalization = fct_reorder (marginalization, income_miles,

```

```

median))%>%>

rename (occupied = PEROCU)%>%>

rename (educacion_jef = LEVELAPROB)%>%>

select (income_miles, num_focos, tel_cellular,
marginalization, occupied, firm_floor, washing machine, car, education_jef)

income_split <- initial_split (data_entry, prop = 0.7)

train <- training (income_split)

test <- testing (income_split)

```

Suppose you are interested in estimating household income and you use a cell phone survey to conduct it. Furthermore, suppose you only access areas that do not have very high marginalization.

```

sample_biased <- filter (train,

                        tel_cellular == "Yes",

                        marginalization == "Very low")

biased_split <- initial_split (bias_sample)

bias_train <- training (bias_split)

bias_validation <- testing (biased_split)

```

A linear model is built for the logarithm of income with the available data.

```

library(splines)

formula <- as.formula ("log (income_miles) ~ ns (num_focuses, 3) +

ns (occupied, 3) + washing machine + car + piso_firme +

                        ns (education_jef, 3) «)

bias_mod <- lm (formula, data = bias_train)

# we take a representative sample to compare, the same size as the biased

mod_representative <- lm (formula, data = sample_n (train, nrow (train_bias)))

```

And the error is evaluated in a test sample constructed with data with the same biased characteristics as the training data (households with a cell phone and a very low degree of marginalization).

```

preds_val <- predict (bias_mod, newdata = bias_validation)

mean (abs (preds_val - log (1 + bias_validation $ thousand_income)))%>% round
(2)

## [1] 0.37

```

The error in a sample more similar to the population to which the algorithm is intended to apply is greater.

```

bias_test_preds <- predict (bias_mod, newdata = test)

preds_test <- predict (mod_representative, newdata = test)

test $ pred_bias <- preds_test_bias

test $ pred_rep <- test_preds

mean (abs (bias_test_preds - log (1 + test $ thousands_income)))%>% round (2)

## [1] 0.42

```

However, the main problem is reflected in the graphs below, where logarithmic scales are used to make multiplicative comparisons, which are interesting due to the nature of income. Each point represents a household, the sample is more similar to the population to which the methodology will be applied, and the prediction of households using the model is plotted on the horizontal axis, while the vertical axis corresponds to the income of each household. As a reference, the line $y = x$ and a smoother are added. The focus is on performance for relatively low-income households (less than 10,000 pesos per month):

```

breaks_y <- c (3, 5, 10, 20, 40, 80)

g_bias <- ggplot (test%>% filter (pred_bias < log (30)),

aes (x = exp (pred_biased), y = income_miles)) +

  geom_point (alpha = 0.5) +

  geom_abline () + geom_smooth (method = "loess", span = 1) +

  scale_x_log10 (limits = c (5, 30)) + scale_y_log10 (breaks = breaks_y) +

  xlab("Forecast (thousands per quarter)") +

  ylab("Current income (thousands per quarter)") +

  labs (subtitle = "Test performance \ nwith training bias")

```

```

g_bias <- ggplot (test %>% filter(pred_sesgada < log(30)),
aes (x = exp (pred_rep), y = income_miles)) +

  geom_point (alpha = 0.5) +

  geom_abline () + geom_smooth (method = "loess", span = 1) +

  scale_x_log10 (limits = c (5, 30)) + scale_y_log10 (breaks = breaks_y) +

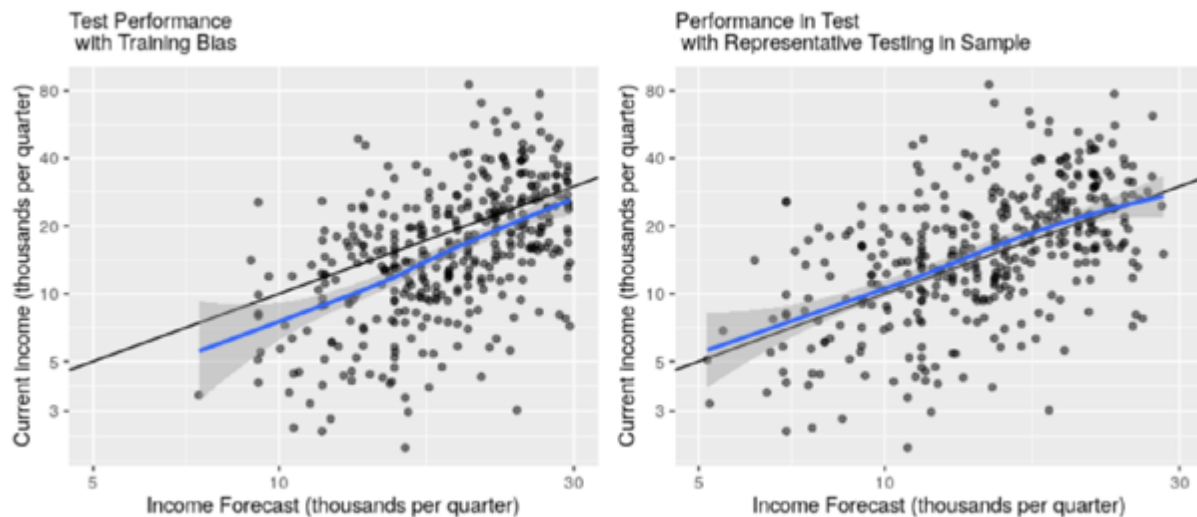
  xlab("Forecast (thousands per quarter)") +

  ylab("Current income (thousands per quarter)") +

  labs (subtitle = "Performance in test \ nwith representative sample in training")

g_bias + g_representative

```



Although it is commonly expected to over-predict relatively low observed values, and to do the opposite for relatively high values, for households with incomes of less than 10,000 pesos per month the biased model overpredicts true income by around 40 percent:

```

test_low <- test%>% filter (income_miles < 3*10)

bias <- mean (exp (low_test $ pred_biased)) / mean (low_test $ thousands_income)
- 1

round (bias,3)

## [1] 0.412

```


When compared with the same trained model with a representative sample, the effect is considerably less.

```
test_low <- test%>% filter (income_miles < 3*10)

bias <- mean (exp (low_test $ pred_biased)) / mean (low_test $ thousands_income)
- 1

round (bias,3)

## [1] 0.152
```

There are then two problems:

1. Bias produces a considerably larger error in implementation than in validation.
2. Even worse, the bias is greater for lower-income households (predictions are high), which can produce poor targeting if we seek to identify lower-income households.

Natural Samples: Causal Comparisons

This example is taken from Hastie, Tibshirani, and Friedman (2017) and Rossouw (1983). The following data are considered, with the goal being to predict heart disease (chd):¹⁵

```
sa_heart <- read_csv("data / sa-heart.csv")
sa_heart <- sa_heart%>%
  rename (arterial_pressure = sbp, tobacco = tobacco, cholesterol_ldl = ldl,
  adiposity = adiposity, fam_history = famhist, type_a = typea, obesity = obesity,
  age = age, coronary_en = chd)
sa_heart
```

```
## # A tibble: 462 x 10
## arterial_pressure tobacco cholesterol_ldl adiposity history_fam type_a
##           <dbl>   <dbl>           <dbl>     <dbl> <chr>           <dbl>
## 1           160    12             5.73      23.1 Present         49
## 2           144    0.01           4.41      28.6 Absent         55
## 3           118    0.08           3.48      32.3 Present         52
## 4           170    7.5            6.41      38.0 Present         51
## 5           134   13.6           3.5       27.8 Present         60
```

¹⁵ The data can be accessed at <http://archive.ics.uci.edu/ml/datasets/heart+Disease>.

```
## 6          132    6.2          6.47          36.2 Present          62
## 7          142    4.05          3.38          16.2 Absent          59
## 8          114    4.08          4.59          14.6 Present          62
## 9          114    0          3.83          19.4 Present          49
## 10         132    0          5.8          31.0 Present          69

## #... with 452 more rows, and 4 more variables: obesity<dbl>, alcohol<dbl>,
## # age <dbl>, enf_coronary <dbl>
```

```
library(recipes)

set.seed(125)

sa_split <- rsample :: initial_split (sa_heart, prop = 0.75)

sa_split
```

```
## <Training/Validation/Total>
```

```
## <347/115/462>
```

```
receta_sa <- training(sa_split) %>%
  recipe(enf_coronary ~ .) %>%
  step_dummy(history_fam) %>%
  step_mutate (enf_coronary= factor (enf_coronary))%>%
  prep()

sa_entrena <- recipe_sa%>% juice

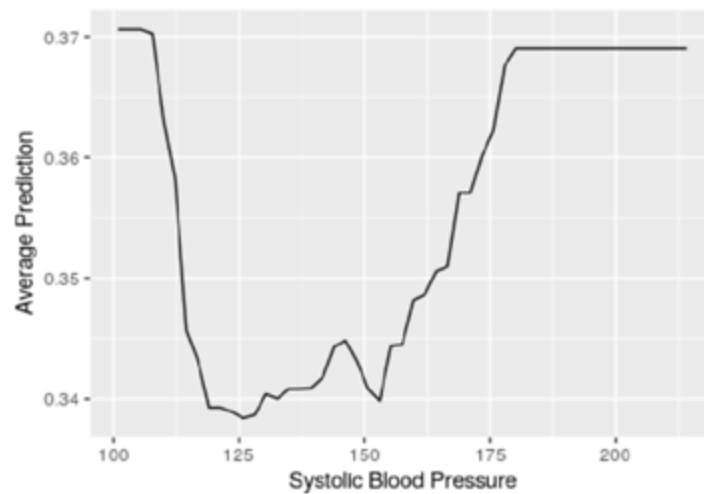
sa_boosted <- boost_tree (trees = 3000, mode = "classification",
                          learn_rate = 0.001, tree_depth = 2,
                          sample_size = 0.5)%>%
  set_engine("xgboost")%>%
  fit (enf_coronary ~., data = sa_entrena)
```

You can evaluate this model and fine-tune parameters as well. Here we are interested in interpreting the effect of the variables in this model. For this, the partial dependence graph of the prevalence of heart disease and the obesity variable is considered.

```
library(pdp)

pdp_ob <- pdp :: partial (sa_boosted $ fit, pred.var = "arterial_pressure",
  plot = TRUE, plot.engine = "ggplot2", prob = TRUE,
  train = sa_entrena%>% dplyr :: select (-enf_coronary))>

pdp_ob + xlab ("Systolic Blood Pressure") + ylab("Average Prediction")
```



The correct interpretation of this partial dependence graph (Hastie, Tibshirani, and Friedman 2017) depends on the fact that this is a retrospective study, where some patients at risk of heart disease underwent interventions to reduce their risk, including taking medicines to reduce blood pressure. A causal interpretation of blood pressure reductions as a promoter of heart disease is incorrect and potentially dangerous.

Model Building and Validation

Leak Training Validation

Several examples of how leakages from training to validation produce biased estimates of predictor performance will be presented here.

Selecting Variables before Dividing the Data

Any pre-processing step must be done without using validation data. This includes when methods such as cross-validation are used.

This example is originally from Hastie, Tibshirani, and Friedman (2017) and will use synthetic data generated through the following process:

1. Simulating response variables and with binomial distribution.
2. Simulating 1,000 independent covariates, each with a standard normal distribution.

```
simulate <- function(n = 100, p = 500, prob = 0.5) {
  data <- map (1:p, ~ rnorm (n)) %>%
bind_cols ()
  data $ y <- rbinom (n, 1, prob)
  data
}

set.seed (8234)

training_data <- simulate (n = 200, p = 1000)
test_data <- simulate (n = 2000, p = 1000)

dim (data_train)

## [1] 200 1001
```

```
data_train %>% group_by(y) %>% tally() %>% kable()
```

y	n
0	113
1	87

The selection of variables is given by the function below. This function selects the variables most correlated with the target variable.

```

select <- function(data, num_var = 10) {
  correlations <- data%>%
    pivot_longer (cols = matches ("V"), names_to = "variable", values_to = "x")%>%
  group_by (variable)%>%>
  summarize (corr = abs (cor (y, x)))%>%>
  arrange (desc (corr))

  # select

  selected <- correlations%>%
  top_n (num_var, wt = corr)%>%>
  pull (variable)

  data%>% select (one_of (c ("y", selected)))
}

```

Wrong Method

Here are the 10 variables that were selected. By itself, this method is not wrong, but when run on the data to be used in validation (cross-validation), then the performance estimate is optimistic:

```

filtered_data <- select (data_train)

filtered_data%>% head%>%>>

mutate_if (is.numeric, round, 3)%>% kable ()

```

y	V337	V464	V984	V461	V525	V732	V39	V774	V491	V682
0	1.592	-0.587	1.763	-0.847	0.452	-0.604	-0.400	-1.146	-0.938	0.136
0	1.782	0.604	0.739	-0.533	1.752	0.945	1.142	-0.638	-0.342	1.308
0	1.528	0.635	-0.326	0.734	-0.207	0.974	1.574	2,401	0.428	0.176
0	0.799	-1.436	0.724	0.366	1.680	0.476	0.376	-1.673	-0.683	0.161
0	0.759	-0.208	-0.373	0.208	-1.009	-0.028	-1.209	0.759	2,038	1.402
1	-0.377	-1.044	1.358	-0.223	0.469	1,221	0.582	0.378	-0.116	0.173

For whatever validation cut-off is made (whether separating a dataset or cross-validating), the percentage of hits appears to be greater than 0.5:

```
cut_validation <- filtered_data%>% sample_frac (0.7)

validate <- anti_join (filtered_data, validation_cut)

model_1 <- glm (y ~., validation_cut, family = "binomial")

mean (as.numeric (predict (model_1, valid)> 0) == valid $ y)%>% round (2)>

## [1] 0.73
```

However, the actual performance of the model will be:

```
mean (as.numeric (predict (model_1, test_data)> 0) == test_data $ y)%>% round
(2)>

## [1] 0.49
```

Correct Method

The selection of variables must be done in each round of cross-validation.

```
cut_validation <- filtered_data%>% sample_frac (0.7)

cut_filtered_data <- select (validation_cut)

validate <- anti_join (data_train, cut_validation)

model_1 <- glm (and ~., cut_filtered_data, family = "binomial")

mean (as.numeric (predict (model_1, valid)> 0) == valid $ y)%>% round (2)>

## [1] 0.52
```

Oversample before Partitioning

One of the ways to solve class imbalance problems is to use oversampling techniques. However, you have to be very careful to avoid information leakage errors when applying these techniques.

In this example, you will see that oversampling a small class before separating validation data or cross-validating can produce overly optimistic estimates of prediction error.

Suppose we have a severe imbalance between our two classes:

```
set.seed (99134)

data_imbalance <- simulate (n = 500, p = 20, prob = 0.1)%>%

  mutate (y = factor (y, levels = c (1, 0)))

data_desbalance %>% group_by(y) %>% tally() %>% kable()
```

y	n
1	41
0	459

Wrong Method

Suppose the Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla 2002) is applied first to try to balance the data.

```
recipe_balance <- recipe (and ~., unbalance_data)%>%
  step_smote (y)%>%>
  prep()
data_smote <- juice (recipe_balance)
```

Obtaining this:

```
data_smote%>% group_by(y) %>% tally() %>% kable()
```

y	n
1	459
0	459

Now training and validation will be separated.

```
sep_data_smote <- initial_split (data_smote)
train_smote <- training (sep_data_smote)
test_smote <- testing (sep_data_smote)
```

And a classification method is generated using a random forest of decision trees.

```
metrics <- metric_set (accuracy, recall, precision)
forest <- rand_forest (trees = 500, mtry = 20, mode = "classification")%>%
  set_engine("ranger")%>%
```

```

fit (y ~., data = train_smote)

forest%>%>

predict (test_smote)%>%>

bind_cols (test_smote)%>%>

metrics (truth = y, estimate = .pred_class)%>%>

  mutate_if(is.numeric, round, 3) %>% kable

```

.metric	.estimator	.estimate
Accuracy	Binary	0.926
Recall	Binary	0.973
Precision	Binary	0.887

At first glance it seems that the performance is excellent. However, since there is no relationship between 'y' and the rest of the covariates we know that this is an error.

Correct Method

Before doing the rebalancing of classes, training and validation are separated. If you like, this part can be done using stratified sampling, for example, but here it is built with simple random sampling.

```

sep_data <- initial_split (data_imbalance, prop = 0.5)

train <- training (sep_data)

test <- testing (sep_data)

recipe_balance <- recipe (y ~., data = train)%>%

  step_smote (y)%>%>

  prep()

train_balanced <- juice (recipe_balance)

forest_1 <- rand_forest (trees = 500, mtry = 20, mode = "classification")%>%

  set_engine("ranger")%>%

  fit (y ~., data = balanced_train)

forest_1%>%>

predict (test)%>%>

```



```
bind_cols (test)%>%>

metrics (truth = y, estimate = .pred_class)%>%>

  mutate_if(is.numeric, round, 3) %>% kable

kable ()
```

.metric	.estimator	.estimate
Accuracy	Binary	0.828
Recall	Binary	0.000
Precision	Binary	0.000

Although the accuracy seems high, the precision and sensitivity are zero. A trivial classifier that always predicts the ruling class may have better accuracy than the one we have constructed.

Leaks in Implementation

Variables Not Available at the Time of Prediction

In this case, we show an example where a variable is used erroneously that will not be available at the time of making the predictions (data from Greene 2003).

```
credit <- read_csv("data / AER_credit_card_data.csv")%>%

rename (expense = expenditure, dependents = dependents, income = income,

age = age, owner = owner)%>%>

  mutate (owner = fct_recode (owner, c ( si ="yes"))

credit %>% head %>%

mutate_if (is.numeric, round, 3)%>% kable ()
```

Card	Reports	Age	Income	Share	Expense	Owner	Self-employed	Dependents	Months	Major Cards	Active
Yes	0	37.7	4.5	0.0	125.0	Yes	No	3	54	1	12
Yes	0	33.2	2.4	0.0	9.9	No	No	3	34	1	13
Yes	0	33.7	4.5	0.0	15.0	Yes	No	4	58	1	5
Yes	0	30.5	2.5	0.1	137.9	No	No	0	25	1	7
Yes	0	32.2	9.8	0.1	546.5	Yes	No	2	64	1	5
Yes	0	23.2	2.5	0.0	92.0	No	No	0	54	1	1

You want to build a model to predict which applications were accepted and automate the selection process. A logistic regression with Keras and L2 penalty is used.

```

set.seed(823)

credit_split <- initial_split (credit)

train <- training (credit_split)

test <- testing (credit_split)

#data preparation

recipe_credit <- recipe (card ~., credit)%>%

step_normalize (all_numeric ())%>%>

step_dummy (all_nominal (), -card)

# model

regular_model <-

  logistic_reg (penalty = 1)%>%

  set_engine("keras", epochs = 500, verbose = FALSE)%>%

  set_mode("classification")

# adjust preprocessing parameters

recipe_prep <- credit_recipe%>% prep (train)

# preprocess data

train_prep <- bake (recipe_prep, train)

test_prep <- bake (recipe_prep, test)

# adjust model

fit <- regular_model%>%

fit (card ~ expense + dependents + income + age + owner_yes, data = train_prep)

# evaluate

metrics <- metric_set (accuracy, recall, precision)

fit%>% predict (test_prep)%>%>>

bind_cols (test)%>%>

metrics (truth = factor (card), estimate = .pred_class)%>%>

  mutate_if(is.numeric, round, 3) %>% kable

kable ()

```

.metric	.estimator	.estimate
Accuracy	Binary	0.833
Recall	Binary	0.393
Precision	Binary	0.892

And it seems to be performing reasonably well. If we remove the variable expenditure, the performance of the model is totally degraded:

```
adjustment_2 <- regular_model %>%
  fit (card ~ expense + dependents + income + age + owner_yes, data = train_prep)
fit%>% predict (test_prep)%>%>>
bind_cols (test)%>%>
metrics (truth = factor (card), estimate = .pred_class)%>%>
  mutate_if(is.numeric, round, 3) %>% kable
kable ()
```

.metric	.estimator	.estimate
Accuracy	Binary	0.745
Recall	Binary	0.000
Precision	Binary	n.a.

Sensitivity is very poor and precision cannot be calculated because the model does not make positive predictions for the test set.

The reason for this performance degradation is that spending refers to the use of credit cards. This includes the card for which you want to make an acceptance prediction.

```
train%>%>
  mutate (some_expenditure = expense>; 0)%>%
group_by (some_cost, card)%>%>
tally ()%>%>
kable ()
```

some_expense	Card	No.
False	No	212
False	Yes	19
True	Yes	759

This indicates that some expense probably includes expense on the current card, and the expense variable is measured after the delivery of the card:

The performance of this model for new applications will be poor, since the expense variable, at the time of application, obviously does not count how much each client will spend in the future.

Cut-off Point Evaluation

The best cut-off decisions can be made with cost-benefit analysis using lift curves, like those in the previous example, based on gains and losses for each decision. Although this information is often not available, it is ideal to evaluate how the model helps and how much the actions we intend to take are worth. It is possible to do this analysis with uncertain cost-benefit values.

Suppose we are thinking of a treatment to retain students in some training or improvement program.

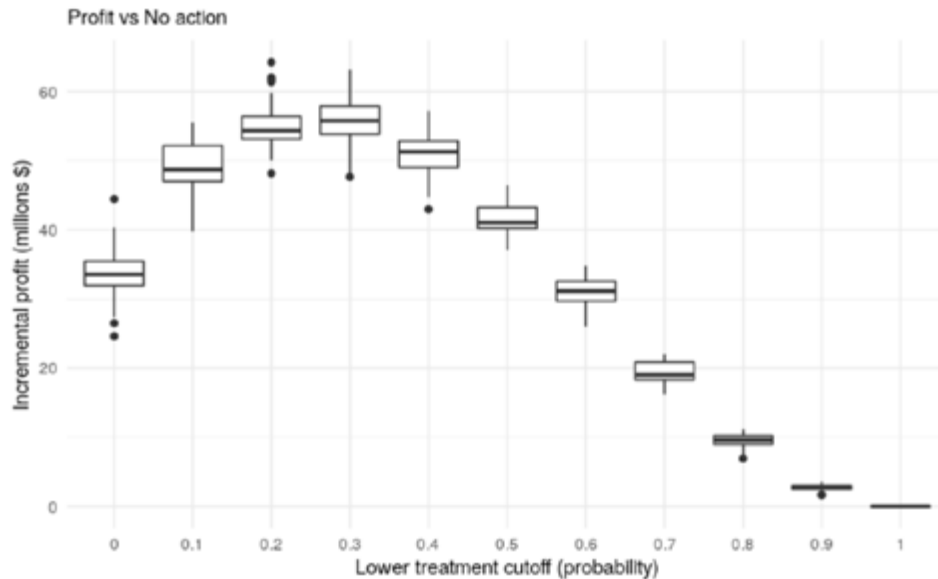
- The retention treatment costs 5,000 pesos per student.
- It is estimated through experiments or some external analysis that the treatment reduces the probability of dropping out by 60 percent.
- There is some kind of assessment of the social value of a student persisting in the program.

You can evaluate the model in the context of the problem as follows:

- Assuming that a percentage of the students most likely to rotate will be treated.
- The expected cost is calculated if a percentage of the students is treated: we simulate reducing their probability of dropping out due to the treatment and we add the costs of treating them.
- The model is compared against the scenario of not applying any treatment

It is not necessary to use highly technical measures to give a summary of how the treatment and model can help to maintain the value of the portfolio.

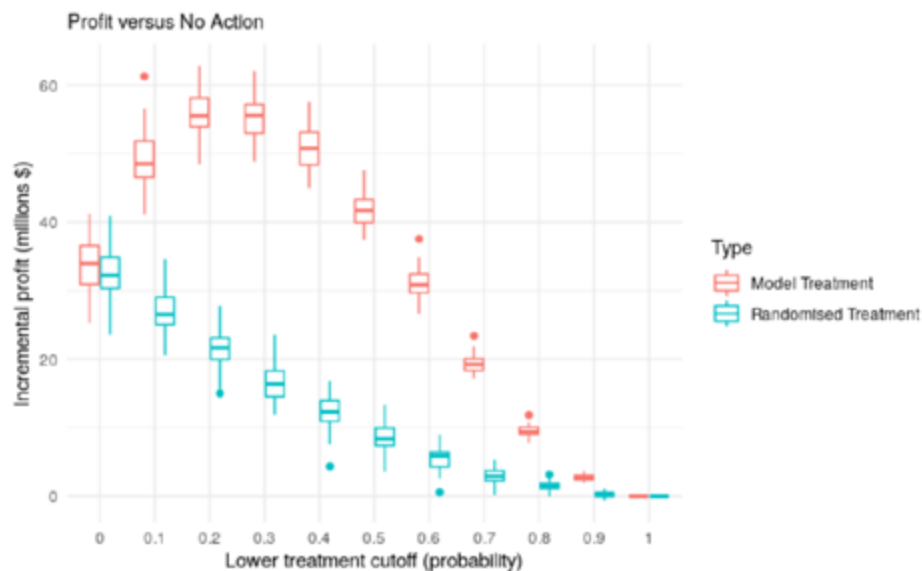
```
ggplot (filter (loss_sim, type == "Model treatment"),
        aes (x = factor (cut), y = - loss / 1e6)) +
  geom_boxplot () + ylab("Incremental profit (millions)") +
  xlab("Lower treatment cutoff (probability)") +
  labs (subtitle = "Profit vs no action") + theme_minimal ()
```



You can choose a cut-off point between 0.2 and 0.3, for example, or do more simulations to refine your choice.

If you want to separate the effect of the treatment with the effect of the treatment applied according to the model, it can be compared with the action that consists of treating the students at random:

```
ggplot (loss_sim, aes (x = factor (cut), y = - loss / 1e6,
  group = interaction (type, cut), color = type)) +
  geom_boxplot () + ylab("Incremental profit (millions)") +
  xlab("Lower treatment cutoff (probability)") +
  labs (subtitle = "Profit vs no action") + theme_minimal ()
```



The bottom line is that the model **helps considerably in the targeting of the program** (the area between the two curves shown above).

Class Imbalance

When you have a severe class imbalance, you can face two problems: there are in absolute terms too few elements of a class to be able to discriminate it effectively (even when you have the correct attributes or feature), or the usual predictive evaluation methods are deficient to evaluate the performance of predictions.

Consider the following scenario put forth in James et al. (2017):

“The data contains 5,822 real customer records. Each record consists of 86 variables, which contain sociodemographic data (variables 1-43) and product ownership (variables 44-86). Sociodemographic data is derived from postal codes. All customers living in areas with the same ZIP code have the same sociodemographic attributes. Variable 86 (Purchase) indicates whether the client purchased a caravan insurance policy.”¹⁶

You want to predict the variable Purchase:

```
caravan <- read_csv("data / caravan.csv") %>%
mutate (MOSTYPE = factor (MOSTYPE),
MOSHOOFD = factor (MOSHOOFD)) %>%
  mutate (Purchase = fct_recode (Purchase, yes = "Yes", no = "No")) %>%
mutate (Purchase = fct_rev (Purchase)) %>%
select (-Purchase)
nrow (caravan)

## [1] 5822
```

¹⁶ Data and more information are available at <http://www.liacs.nl/-putten/library/cc2000/data.html>

```
caravan%>% count (Purchase)%>%>>

mutate (pct = 100 * n / sum (n))%>%

mutate (pct = round (pct, 2))
```

```
## # A tibble: 2 x 3

## Buy n pct

##   <fct>   <int> <dbl>

## 1 yes  348  5.98

## 2 no  5474 94.0
```

This is the natural distribution of response seen in the data, and you have relatively little data in the “Yes” category.

Stratified sampling will be used to obtain similar proportions in training and test sets.

```
set.seed(823)

caravan_split = initial_split (caravan, strata = Buy, prop = 0.9)

caravan_split

## <Training/Validation/Total>

## <5240/582/5822>
```

```
train <- training (caravan_split)

test <- testing (caravan_split)
```

Logistic regression will be used (the same applies for other methods that produce class probabilities, such as boosting, random trees, or neural networks).

```
library(tune)

#data preparation

caravan_recipe <- recipe (Buy ~., train)%>%

step_dummy (all_nominal (), -Purchase)

caravan_receta_prep <- caravan_receta%>% prep

# model

model_log <-
```

```
logistic_reg ()%>%
  set_engine("glm")%>%
  set_mode("classification")%>%
fit (Buy ~., data = caravan_receta_prep%>% juice)>
```

Incorrect Analysis

The confusion matrix for the training data is:

```
predictions_ent_glm <- model_log%>%
predict (new_data = juice (caravan_receta_prep))%>%>
bind_cols (juice (caravan_receta_prep)%>% select (Buy))>
predictions_ent_glm%>%>
conf_mat (Purchase, .pred_class)

##              Truth
## Prediction if not
## yes 6 9
## no 299 4926
```

And the test ones are:

```
test_processed <- bake (caravan_recipe_prep, test)
predictions_ent_glm <- model_log%>%
predict (new_data = test_processed)%>%>
bind_cols (test_processed%>% select (Buy))>
predictions_glm%>%>
conf_mat (Purchase, .pred_class)

##              Truth
## Prediction if not
## yes 0 4
## no 43 535
```

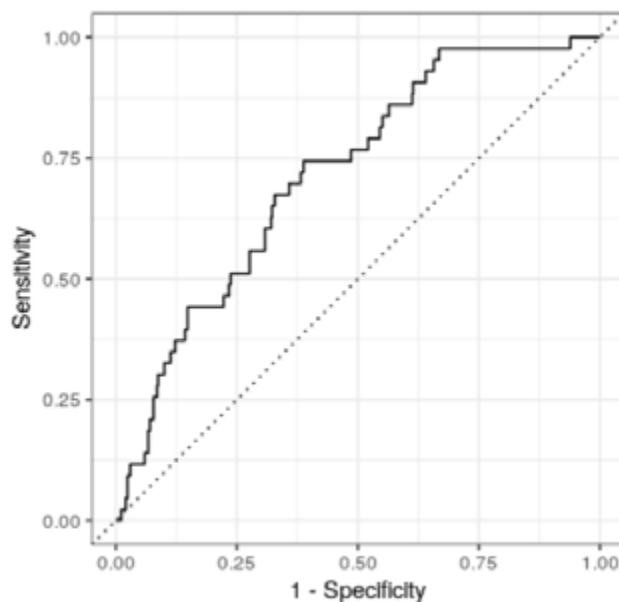
You get poor performance according to this confusion matrix (test and training). The sensitivity is very low, although the specificity (rate of correct negatives) is high. A typical conclusion is that *the model has no predictive value, or that it is necessary to oversample the low occurrence class.*

Correct Analysis

Instead of starting with oversampling/undersampling, which modifies the natural proportions of categories in the data, we can work with probabilities instead of class predictions with a cut-off of 0.5.

For example, this can be visualized with a receiving operating characteristic curve (or lift curve, precision-recall, or some other curve that takes probabilities into account).

```
predictions_prob <- model_log%>%
  predict (new_data = test_processed, type = "prob")%>%
  bind_cols (test_processed%>% select (Buy))>
  select (.pred_si, Buy)
roc_data <- roc_curve (predictions_prob, Buy, .pred_si)
autoplot (roc_data) +
  xlab("1 - specificity") + ylab("sensitivity")
```



It can be seen that it is possible to achieve good levels of sensitivity if some degradation in specificity is accepted, which is originally very high. For example, cutting at 0.05 can obtain specificity and sensitivity that are possibly adequate for the problem.

```
datos_roc %>% filter(abs(.threshold - 0.04) < 1e-4) %>% round(4)

## # A tibble: 2 x 3
## .threshold specificity sensitivity
```

##	<dbl>	<dbl>	<dbl>
## 1	0.0399	0.553	0.744
## 2	0.0399	0.555	0.744

What happens if we oversample?

Oversample:

```
caravan_recipe <- recipe (Buy ~., train)%>%
  step_dummy (MOSTYPE, MOSHOOFD)%>%>
  step_smote (Purchase)
smote_prep <- prep (caravan_receta_smote)
# model
train_1 <- juice (smote_prep)
train_1%>% count (Purchase)>
```

```
## # A tibble: 2 x 2
## Buy n pct
##   <fct>   <int>
## 1 yes  4935
## 2 no   4935
```

```
model_log_smote <-
  logistic_reg ()%>%
    set_engine("glm")%>%
    set_mode("classification")%>%
  fit (Buy ~., data = train_1)
```

In training the confusion matrix is *apparently* better.

```
predictions_ent_glm <- model_log_smote%>%
  predict (new_data = train_1)%>%>
  bind_cols (train_1%>% select (Buy))>
  predictions_ent_glm%>%>
  conf_mat (Purchase, .pred_class)
```

```
##           Truth

## Prediction if not

## yes 3854 1271

## no 1081 3664
```

But while testing, the results are very similar. The model built by sub-sampling the ruling class is also added.

```
train_sub <- caravan_recipe%>% step_downsample (Purchase)%>% prep ()%>% juice>>

model_log_sub <-

logistic_reg ()%>%>

  set_engine("glm")%>%

  set_mode("classification")%>%

fit (Buy ~., data = train_sub)

predictions_prob <- model_log_smote%>%

  predict (new_data = test_processed, type = "prob")%>%

bind_cols (test_processed%>% select (Buy)) %>%

select (.pred_si, Buy)

predictions_prob_sub <- model_log_sub%>%

  predict (new_data = test_processed, type = "prob")%>%

bind_cols (test_processed%>% select (Buy)) %>%

select (.pred_si, Buy)

roc_smote_data <- roc_curve (predictions_prob, Buy, .pred_si)

roc_sub_data <- roc_curve (predictions_prob_sub, Buy, .pred_si)

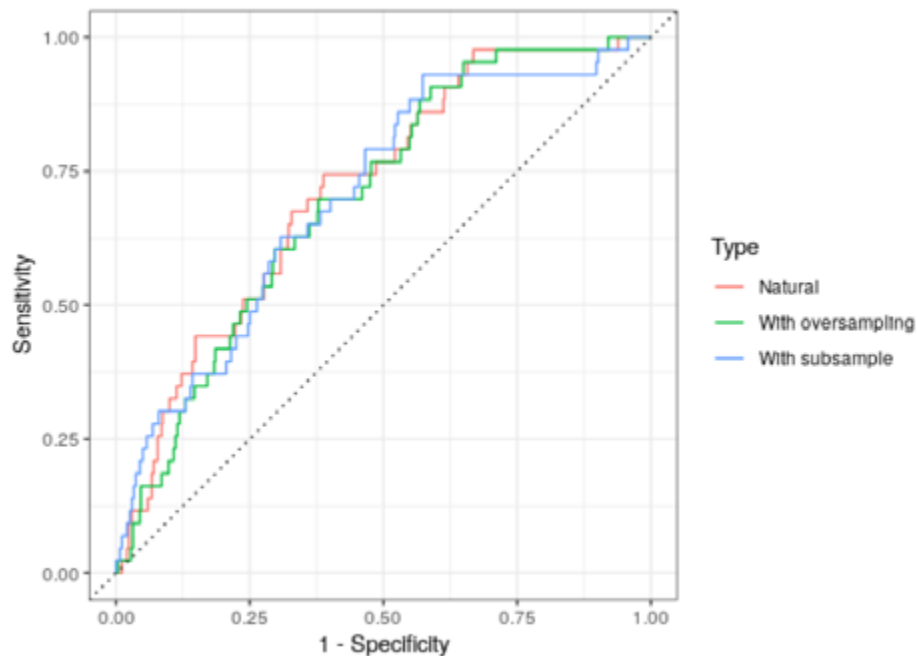
roc_comp_data <- bind_rows (roc_data%>% mutate (type = "natural"),

                           data_roc_smote%>% mutate (type = "with oversampling"),

                           data_roc_sub%>% mutate (type = "with subsample")

)
```

```
ggplot (comp_roc_data,
      aes (x = 1 - specificity, y = sensitivity, color = type)) +
  geom_path () +
  geom_abline (lty = 3) +
  coord_equal () +
  theme_bw ()
```



The original problem was not that the fit was not working, but that the wrong cut-off point was evaluated. A cut-off point of 0.5 with SMOTE is equivalent to a much smaller one without SMOTE.

Worse still, the **probabilities of the oversampled model do not reflect the occurrence rates of the response of interest**, which can produce misleading summaries of the response rates that are expected to be observed in production.

Equality with Protected Attributes

The following example is derived from Hardt, Price, and Srebro (2016). Suppose you have a protected attribute *A* that has two values: blue and orange. Orange is the disadvantaged minority group. Simulated data will be used as follows, with the score attribute associated with the protected attribute:

```

inv_logit <- function(x) {
  1 / (1 + exp (-x))
}

simulate_data <- function(n = c (10000, 2000)) {
  score_azul <- pmax (rnorm (n [1], 50, 10), 0)
  score_orange <- pmax (rnorm (n [2], 40, 10), 0)

  blue <- tibble (type = "blue", score = score_blue)

  orange <- tibble (type = "orange", score = score_orange)

  data <- bind_rows (blue, orange)%>%

    mutate (coef_0 = ifelse (type == "blue", 0.0, 0),

      prob_real_pos = inv_logit(-1 + coef_0 + 0.1 * (score-40))) %>%

    mutate(atr_1 = rpois(nrow(.), 3))
  data%>% select (-coef_0)%>%>>

    mutate (pay = map_dbl (prob_real_pos, ~ rbinom (1, 1, .x)))%>%
  select (-prob_real_pos)
}

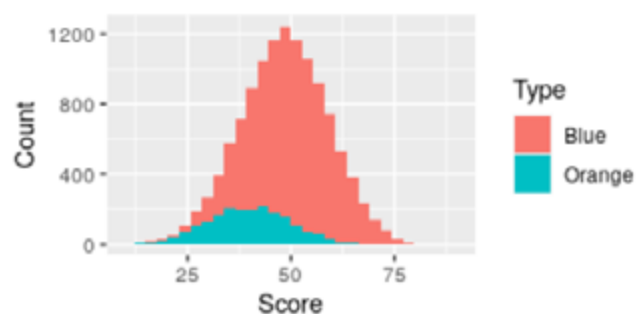
set.seed (1221)

tbl_data <- simulate_data ()

```

Using a histogram for the score, a minority group is obtained with values of the lowest score variable.

```
ggplot (tbl_datos, aes (x = score, fill = type)) + geom_histogram ()
```



A simple logistic regression model is fitted.

```
reg_log <- glm (pay ~ score + atr_1 + type, tbl_datos, family = "binomial")

tbl_datos <- tbl_datos%>% mutate (prob_pos = predict (reg_log, type = "re-
sponse"))
```

The actual compliance rates are the same for the two groups. First, a strategy is considered where the same cut-off point is applied for all groups.

```
cut_result <- function(data_tbl, cuts) {
  result <- tbl_datos%>%
    mutate(recibe = ifelse(type == "Blue", prob_pos > cortes[1], prob_pos >
cortes[2]),
    decision = ifelse (receive, "Accepted", "Rejected"))
  result%>% group_by (type, decision, pay)%>% count ()%>%>>>
  ungroup ()

}

count_results <- cut_result (tbl_data, c (0.6, 0.6))
results_counting

## # A tibble: 8 x 4
## decision type pays n
##   <chr>    <chr>    <dbl> <int>
## 1 blue Accepted 0 905
## 2 blue Accepted 1 2400
## 3 blue Rejected 0 4 149
## 4 blue Rejected 1 2546
## 5 orange Accepted 0 47
## 6 orange Accepted 1 101
## 7 orange Rejected 0 1353
## 8 orange Rejected 1 499

results_counting%>%>
group_by (type, decision)%>%>
```

```

summarize (n = sum (n))%>%>

mutate (total = sum (n))%>%>

mutate (prop = n / total)%>%>

  filter (decision == "Accepted")

## # A tibble: 2 x 5

## # Groups: type [2]

## decision type n total prop

##   <chr>   <chr>   <int> <int> <dbl>

## 1 blue Accepted 3305 10000 0.330

## 2 orange Accepted 148 2000 0.074

```

Note that the orange group has received considerably less acceptance than the blue group, both in total and in proportion. Furthermore, with the precision or rate of true positives we can evaluate what proportion of those who would comply if they were accepted were accepted according to the cut-off point.

```

results_counting%>%>

  filter (pay == 1)%>%>

group_by (type)%>%>

mutate (tvp = n / sum (n))%>%>

  filter (decision == "Accepted")

## # A tibble: 2 x 5

## # Groups: type [2]

## type decision pays n tvp

##   <chr>   <chr>   <dbl> <int> <dbl>

## 1 blue Accepted 1 2400 0.485

## 2 orange Accepted 1 101 0.168

```

It can be seen that the orange group is also at a disadvantage, since among those who comply there are fewer acceptance decisions.

The next step is to consider **demographic parity**. In this case, it was decided to give the same number of loans to each group, depending on their size.

```

calculate_parity_points <- function(tbl_data, prop) {
  tbl_datos%>% group_by (type)%>%>>
    summarize (cut = quantile (prob_pos, 1 - prop))
}

tbl_parity_cuts <- calculate_parity_points (tbl_data, 0.45)
tbl_parity_cuts

## # A tibble: 2 x 2
##   cut type
##   <chr>  <dbl>
## 1 blue  0.521
## 2 orange 0.297

```

The cut-off point for blue is more demanding than for orange. In itself that is not a problem, but it is observed that:

```

cuts_parity <- cuts_parity_tbl%>% pull (cut)
count_results <- cut_result (tbl_data, parity_cuts)
results_counting%>%>
  filter (pay == 1)%>%
  group_by (type)%>%>
  mutate (tvp = n / sum (n))%>%>
  filter (decision == "Accepted")

## # A tibble: 2 x 5
## # Groups: type [2]
##   type decision pays n tvp
##   <chr>  <chr>    <dbl> <int> <dbl>
## 1 blue Accepted 1 3094 0.626
## 2 orange Accepted 1 410 0.683

```

And so in addition to being more demanding with the blue group, those who comply with the blue group are also given fewer acceptance decisions. Additionally, considerably fewer people are accepted from the population.

The **equal opportunity solution is to do** a cut-off so that the acceptance rate within the group of those who pay is similar for both populations, which occurs at approximately 0.35:

```
calculate_opportunity_cuts <- function(tbl_data, prop) {
  tbl_datos%>%
    filter (pay == 1)%>%
    group_by (type)%>%
    mutate (rank_p = rank (prob_pos) / length (prob_pos))%>%
    filter (rank_p <prop)%>%
    top_n (1, rank_p)%>%
    select (type, cut = prob_pos)
}

cuts_op <- calculate_cuts_opportunity (tbl_datos, 0.35)
count_results <- cut_result (tbl_datos, cut_op%>% pull (cut))
results_counting%>%
  filter (pay == 1)%>%
  group_by (type)%>%
  mutate (tvp = n / sum (n))%>%
  filter (decision == "Accepted")

## # A tibble: 2 x 5
## # Groups: type [2]
## type decision pays n tvp
##   <chr>    <chr>    <dbl> <int> <dbl>
## 1 blue Accepted 1 3215 0.650
## 2 orange Accepted 1 391 0.652
```

Note: If the positive outcome variable is unfairly assigned, then this method does not solve the problem. In this case, it is relevant to understand what the criteria are with which a successful result is considered depending on the group of the protected attribute (e.g., if a particular segment is allowed greater arrears in payments and another group allowed less arrears, or if a group is considered a repeat offender for a much lesser offense than other groups).

Accountability

Interpretability

Measures such as importance of permutations can be used to examine models. In this example, we return to the credit application acceptance prediction exercise, and we consider importance based on permutations (Molnar 2019):

```
set.seed(823)

credit_split <- initial_split (credit)
train <- training (credit_split)
test <- testing (credit_split)

#data preparation
recipe_credit <- recipe (card ~., credit)%>%
  step_normalize (all_numeric ())%>%
  step_dummy (all_nominal (), -card)

# model
regular_model <-
  logistic_reg (penalty = 1)%>%
  set_engine("keras", epochs = 500, verbose = FALSE)%>%
  set_mode("classification")

# adjust preprocessing parameters
recipe_prep <- credit_recipe%>% prep (train)

# preprocess data
train_prep <- bake (recipe_prep, train)
test_prep <- bake (recipe_prep, test)

# adjust model
fit <- regular_model%>%
  fit (card ~ expense + dependents + income + age + owner_yes, data = train_prep)

library(iml)

model <- fit $ fit

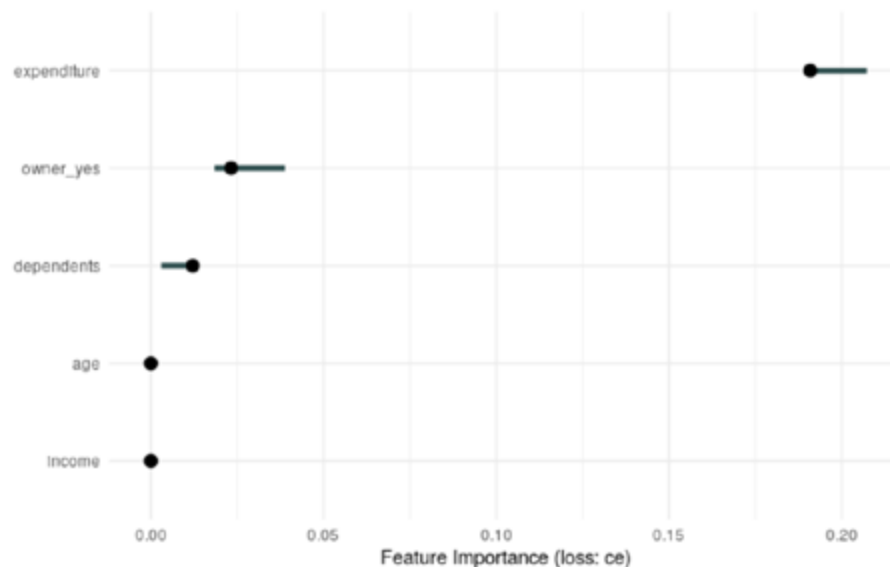
train_x <- train_prep%>% dplyr :: select (expense, dependents, income, age, owner_yes)

predictor <- Predictor $ new (model, data = train_x, y = ifelse (train_prep $ card
=="yes",2,1),

                             type = "prob")

imp <- FeatureImp $ new (predictor, loss = "ce", compare = "difference")

plot (imp) + theme_minimal ()
```



It is seen that, for this network without hidden layers, the importance is concentrated in a single predictor, *expenditure*, which as seen represents an information leak. This diagnosis is useful in general, and although not as dramatic as this example, it can point out which variables are important to consider carefully.

It is important to also consider the effect of variables associated with protected groups, and if necessary, carefully examine how they affect predictions.

- Parsimonious models, which use fewer attributes, facilitate analysis, maintain data flow, and reduce exposure to leakage problems or undesirable effects.

Explanation of Predictions

To explain individual predictions, the Shapley values are used (Molnar 2019; Lundberg and Lee 2017). These graphs below indicate the assigned contribution of each attribute to an individual prediction, under the idea of considering marginal effects on the prediction depending on the presence or absence of other attributes. The contributions obtained add up to the difference between the particular prediction and the average prediction.

Averages across interest groups can also be examined.

Consider the example of factors for detecting heart disease (Rossouw 1983):

```
model_sa <- sa_boosted $ fit

sa_entrena_x <- sa_entrena%>% dplyr :: select (-enf_coronaria)

predict_fun <- function(object, newdata) {

  new_data_x = xgb.DMatrix (data.matrix (newdata), missing = NA)

  results <-predict (model_sa, new_data_x)

  return(results)
```

```

}

predictor <- Predictor $ new (model_sa, data = sa_entrena_x, y = sa_entrena $
chd,

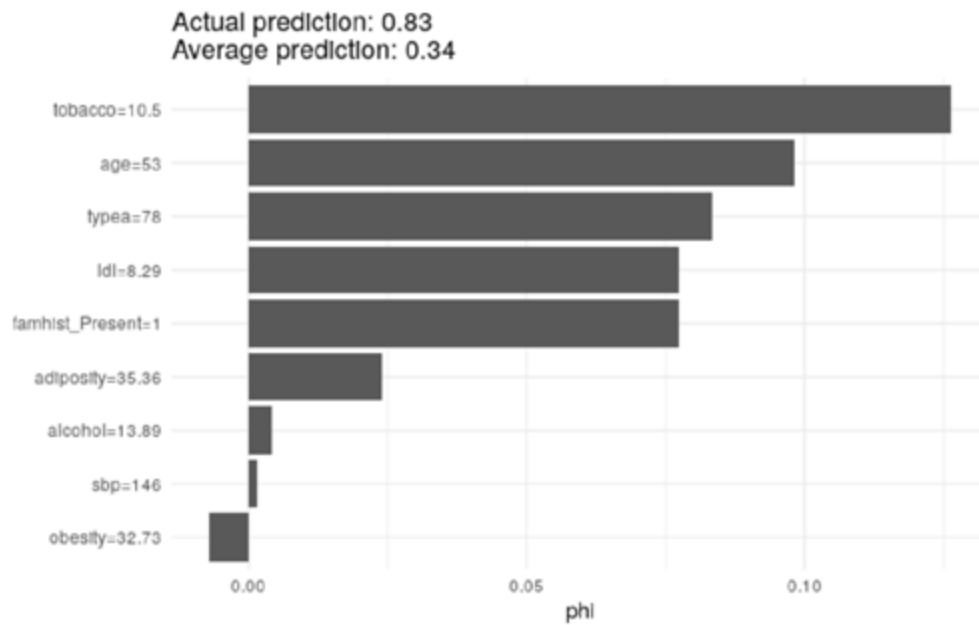
      type = "prob", predict.function = predict_fun)

# the case of interest is case 15

shapley_values <- Shapley $ new (predictor, x.interest = (sa_entrena_x [15,]))

shapley_values $ plot () + theme_minimal ()

```



In this case, several measures contribute positively to the likelihood of heart disease, such as tobacco use, age, and cholesterol measurements. These contributions explain the very high probability of this particular individual.

In contrast, the following person is close to average, with age and cholesterol levels increasing positively, but non-tobacco use and no family history of diabetes:

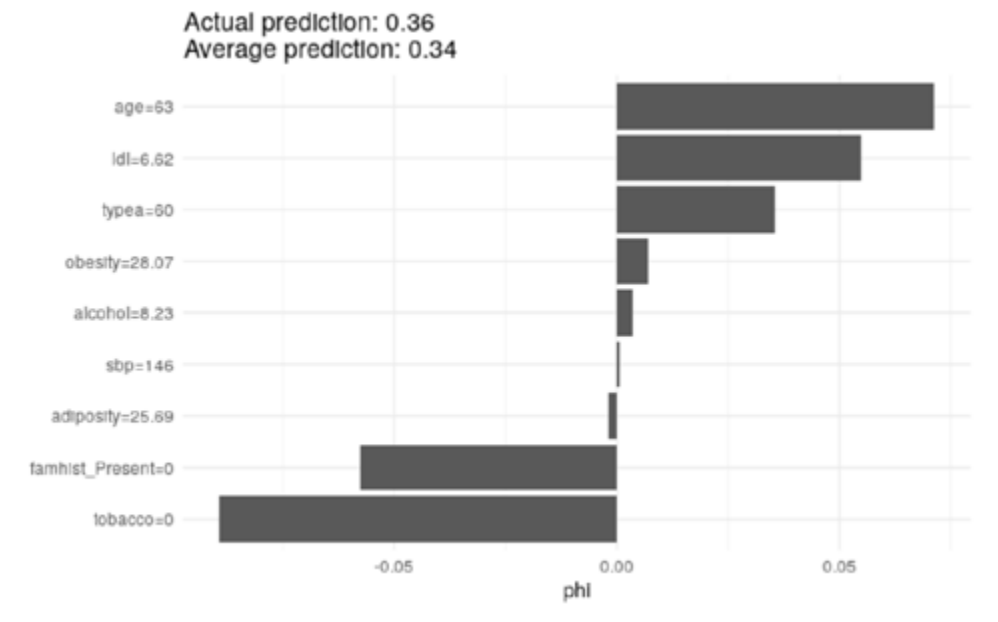
```

# the case of interest is case 24

shapley_values <- Shapley $ new (predictor, x.interest = (sa_entrena_x [24,]))

shapley_values $ plot () + theme_minimal ()

```



Remark: As in the model and partial dependence graphs discussed above, these coefficients should **not** be interpreted causally (e.g., cholesterol needs to be lowered for these two individuals). This is the information that the model uses to build the prediction from the average prediction over the population.

Shapley values can be calculated for two age groups, for example.

References

- Athey, S. W. (2018). Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. *Journal of the American Statistical Association*.
- Barocas, S., & Selbst, A. D. (2014). Big Data's Disparate Impact. SSRN eLibrary.
- Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *ArXiv*, abs/1607.06520.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), págs. 5-32.
- Buolamwini, J., & Gebru, T. (23–24 Feb de 2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. En S. A. Friedler, & C. Wilson (Ed.), *Conference Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. 81, págs. 77-91. New York, NY, USA: PMLR.
- Buuren, S. v., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3), págs. 1-67.
- Carrillo, A., Cantú, L., & Noriega, A. (2020). Individual Explanations in Machine. IADB.
- Chawla, N. V. (2002). SMOTE: Synthetic Minority over-Sampling Technique. *Journal of Artificial Intelligence Research* 16: 321–57.
- drivendata. (2019). An ethics checklist for data scientists. Obtenido de <https://deon.driven-data.org/>
- Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5), págs. 1189-1232.
- Fritzler, A. (2015). An ethical checklist for data science. Obtenido de <http://www.dssgfellowship.org/2015/09/18/an-ethical-checklist-for-data-science/>
- Gebru, T., Morgenstern, J., Vecchione, B., Wortman, J., Wallach, H., Daumé, H., & Crawford, K. (2018). Datasheets for Datasets. Retrieved from <https://arxiv.org/pdf/1803.09010.pdf>
- Gelman, A., & Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models* (1 ed.). Cambridge University Press.
- Greene, W. (2003). *Econometric Analysis*. Pearson Education. Obtenido de <https://books.google.com.mx/books?id=njAcXDIR5U8C>
- Hardt, M. a. (2016). Equality of Opportunity in Supervised Learning. *CoRR*, abs/1610.02413.
- Harini Suresh, J. V. (2019). A Framework for Understanding Unintended Consequences of Machine Learning. MIT. Obtenido de <https://arxiv.org/pdf/1901.10002.pdf>

- Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The Elements of Statistical Learning*. Springer New York Inc.
- INEGI. (2014). Encuesta Nacional de Ingresos Y Gastos de Los Hogares (Enigh-2014). Diseño Muestral. Obtenido de http://internet.contenidos.inegi.org.mx/contenidos/Productos/prod_serv/contenidos/espanol/bvinegi/productos/nueva_estruc/702825070359.pdf
- James, G. D. (2017). *Data for an Introduction to Statistical Learning with Applications in R*. Obtenido de <https://CRAN.R-project.org/package=ISLR>
- Kaufman, S., Rosset, S., & Perlich, C. (01 de 2011). Leakage in Data Mining: Formulation, Detection, and Avoidance., 6, págs. 556-563.
- Kim, B. R. (2016). Examples are not enough, learn to criticize! Criticism for interpretability. *Advances in Neural Information Processing Systems*.
- Kuhn, M. F. (2020). *Rsample: General Resampling Infrastructure*. Obtenido de <https://CRAN.R-project.org/package=rsample>
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York.
- Lackland, D. (06 de 2014). Racial Differences in Hypertension: Implications for High Blood Pressure Management. *The American journal of the medical sciences*, 348.
- Little, R. J., & Rubin, D. B. (2002). *Statistical analysis with missing data*. Wiley.
- Lohr, S. L. (2009). *Sampling: Design and Analysis*. Cengage Learning.
- Lundberg, S., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *ArXiv*, abs/1705.07874.
- Miller, T. (2019). Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artif. Intell.*, 267, págs. 1-38.
- Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., . . . Gebru, T. (2019). Model Cards for Model Reporting. Retrieved from <https://arxiv.org/abs/1810.03993>
- Molnar, C. (2019). *Interpretable Machine Learning*.
- Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464), págs. 447-453.
- OECD (Forthcoming). (s.f.). *Framework for the Classification of AI Systems*. Paris.: OECD Publishing.
- OECD. (2019c). *Artificial Intelligence in Society*. Paris: OECD Publishing.
- OECD. (2021). *Good Practice Principles for Data Ethics in the Public Sector*.

- Pombo, C., Cabrol, M., Alarcón, N. G., & Ávalos, R. S. (2020). fAIr LAC: Adopción ética y responsable de la inteligencia artificial en América Latina y el Caribe. doi:<http://dx.doi.org/10.18235/0002169>
- Prosser, C., & Mellon, J. (2016). Twitter and Facebook are Not Representative of the General Population: Political Attitudes and Demographics of Social Media Users. Available at SSRN: <https://ssrn.com/abstract=2791625> or <http://dx.doi.org/10.2139/ssrn.2791625>.
- Rossouw, J. E. (1983). Coronary Risk Factor Screening in Three Rural Communities. The Coris Baseline Study. South African Medical Journal = Suid-Afrikaanse Tydskrif Vir Geneeskunde.
- Rubin, R. J. (2002). Statistical Analysis with Missing Data, Second Edition . John Wiley & Sons, Inc.
- Stuart, K. I. (2008). Misunderstandings Among Experimentalists and Observationalists about Causal Inference. Journal of the Royal Statistical Society, Series A, 171, part 2, págs. 481-502.
- Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. ArXiv, abs/1703.01365.
- Suresh, H., & Guttag, J. V. (2019). A Framework for Understanding Unintended Consequences of Machine Learning. ArXiv, abs/1901.10002.
- Vaver, J., & Koehler, J. (2011). Measuring Ad Effectiveness Using Geo Experiments. Google Inc.
- Vayena, A. J. (2019). The global landscape of AI ethics guidelines. Springer Science and Business Media LLC.
- Verma, S., & Rubin, J. (2018). Fairness Definitions Explained. Conference Proceedings of the International Workshop on Software Fairness (págs. 1-7). New York, NY, USA: Association for Computing Machinery.
- Wachter, S., Mittelstadt, B. D., & Russell, C. (2017). Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. ArXiv, abs/1711.00399.
- Washingtonpost. (04 de 2019). 21 more studies showing racial disparities in the criminal justice system. Obtenido de <https://www.washingtonpost.com/opinions/2019/04/09/more-studies-showing-racial-disparities-criminal-justice-system/>
- Williams, D. M. (09 de 1981). Racial differences of hemoglobin concentration: measurements of iron, copper, and zinc. The American Journal of Clinical Nutrition, 34(9), págs. 1694-1700.
- Wilson, J. (2014). What your IQ score doesn't tell you. CNN.

