European Journal for Philosophy of Science (2021) 11: 102
https://doi.org/10.1007/s13194-021-00405-1

PAPER IN PHILOSOPHY OF SCIENCE IN PRACTICE

# Values and inductive risk in machine learning modelling: the case of binary classification models

**Koray Karaca**[1]

Received: 12 May 2020 / Accepted: 21 July 2021 /Published online: 26 October 2021


**Abstract**
I examine the construction and evaluation of machine learning (ML) binary classification models. These models are increasingly used for societal applications such as classifying patients into two categories according to the presence or absence of a certain disease like cancer and heart disease. I argue that the construction of ML (binary) classification models involves an optimisation process aiming at the minimization of the inductive risk associated with the intended uses of these models. I also argue that the construction of these models is underdetermined by the available data, and that this makes it necessary for ML modellers to make social value judgments in determining the error costs (associated with misclassifications) used in ML optimization. I thus suggest that the assessment of the inductive risk with respect to the social values of the intended users is an integral part of the construction and evaluation of ML classification models. I also discuss the implications of this conclusion for the philosophical debate concerning inductive risk.

**Keywords** Machine learning · Inductive risk · Underdetermination of model construction · Social values

## 1 Introduction

The societal need to extract useful information from large and complex data sets, often referred to as *big data*, has led to the emergence of big data analytics. This is a new field of study encompassing various computational methods that have been offered to cope with the growing complexity of big data analysis. The transformative

This paper is dedicated to the memory of the late Professor Erdinç Sayan, who introduced me to philosophy of science

✉ Koray Karaca
k.karaca@utwente.nl

[1] Department of Philosophy, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

effects of big data analytics on the social aspects of scientific practice have been extensively studied in the literature on critical data studies (for a review, see Iliadis & Russo, 2016). Only in recent years have the epistemological aspects of big data analytics been scrutinized in the philosophy of science literature, focusing mainly on the epistemology of machine learning (ML).[1]

ML is a prominent computational method of data modelling that has an increasing part in big data analytics (see, e.g., Najafabadi et al., 2015). Models based on ML have recently gained prominence in various societal domains owing to their distinctive ability to draw predictions from big data.[2] The accuracy of the predictions of ML models depends on how well these models generalize to new data sets beyond those used to construct and test them. In this regard, the application of ML models to big data is based on inductive generalization and, as a result, their predictions about new data sets are always prone to error. Therefore, there exists a societal risk associated with grounding decision-making processes in social domains—such as healthcare and criminal justice—on the predictions of ML models, in the sense that the errors in these predictions would translate into wrong decisions that could in turn have negative consequences for both society and individuals.[3]

The foregoing considerations indicate that the kind of risk posed by the societal applications of ML models illustrates what has come to be known as inductive risk, which "arises whenever knowledge is inductively based […] and there are clear consequences to getting it wrong" (Douglas, 2017, p. 93). In the philosophy of science literature, inductive risk has been discussed in relation to the context of theory (or hypothesis or model) acceptance, whereas its relevance to the context of theory (or hypothesis or model) construction has been neglected. In this paper, I will address the latter issue in the context of ML models used for binary classification tasks in societal domains, such as the classification of patients into two categories according to the presence or absence of a certain disease like cancer and heart disease, and the classification of credit applicants as low-risk or high-risk customers. I will argue that the construction of these models requires ML modellers to appeal to social (or non-epistemic) values to quantify the inductive risk that will arise from their intended uses. I will further argue that the evaluation of these models is also dependent on social values relevant to their intended uses. I will thus suggest that social values are essential to both the construction and evaluation of ML classification models. I will also discuss the implications of this conclusion for the philosophical debate concerning inductive risk.

[1] See, e.g., Buckner (2019), Zednik (2021), Biddle (2020), Creel (2020), Erasmus et al. (2020), Ratti (2020), and Sullivan (2019).

[2] These domains include healthcare, e.g., prediction and prognosis of chronic diseases—such as cancer (Kourou et al., 2015) and heart diseases (Ghumbre & Ghatol, 2012)—and drug discovery (Lima et al., 2016); financial risk management (van Liebergen, 2017); fraud detection (Phua et al., 2010); manufacturing (Wuest et al., 2016); criminal justice (Dressel & Farid, 2018), and forensics (Mena, 2011).

[3] See, e.g., Cabitza et al. (2017) for a discussion concerning the unintended consequences of using ML in medicine.

## 2 The inductive risk argument and Jeffrey's counterargument

What is referred to as the argument from inductive risk in the literature of contemporary philosophy of science is an elaborated version of an argument put forward by Richard Rudner (1953) against the value-neutrality of science. The concept of inductive risk, which was first introduced by Carl Hempel (1965), was later incorporated into Rudner's argument by Heather Douglas (2000), whose version of the inductive risk argument includes both epistemic and social values, not only ethical values as in Rudner's original argument. The starting point of the inductive risk argument is the inductive underdetermination of theory by empirical evidence, which refers to the following situation:

> [A]t any given stage of a scientific enquiry the available data will in principle be compatible with many different, mutually incompatible theories. This is because theories always outstrip the data on which they are based, if only by universal generalization—the inference from data to theory is always deductively invalid (Okasha, 2002, pp. 303–304).

According to the inductive risk argument, since the decision to accept or reject a hypothesis is (inductively) underdetermined in the aforementioned sense, scientists accept or reject hypotheses with less than the amount of empirical evidence needed for certainty, meaning that "no scientific hypothesis is ever completely verified" (Rudner, 1953, p. 2). As a result, in accepting or rejecting hypotheses, scientists inevitably take an inductive risk that can be characterized as the risk of erring "when making an inductive leap from evidence to hypothesis acceptance or rejection" (Biddle & Kukla, 2017, p. 216). In order for scientists to manage this risk, they need to decide whether the available empirical evidence is sufficiently strong to warrant their decision to accept or reject a given hypothesis, i.e., to take the inductive risk. This decision depends on the severity of the inductive risk that will arise from the consequences of the acceptance or rejection of the hypothesis. In order for scientists to assess the severity of the resulting inductive risk, and thereby to make a hypothesis acceptance or rejection decision, they need to make epistemic and social value judgments that they find relevant to this assessment.

Richard Jeffrey's counterargument against Rudner's inductive risk argument is still regarded as a serious challenge against the value-neutrality of science. Jeffrey rejects one of the underlying premises of Rudner's argument, namely that "the scientist qua scientist accepts or rejects hypotheses" (Rudner, 1953, p. 4). Jeffrey suggests that:

> It is not the business of the scientist as such, least of all of the scientist who works with lawlike hypotheses, to accept or reject hypotheses ... [T]he scientist's proper role is to provide the rational agents in the society which he represents with probabilities for the hypotheses [with respect to available evidence]. (Jeffrey, 1956, p. 245)

Jeffrey's main concern about Rudner's argument is that accepting or rejecting a hypothesis once and for all does not do justice to the variety of decisions to be based on the acceptance or rejection of the hypothesis. Here, Jeffrey's consideration is that inductive risks associated with different decisions can be qualitatively different, as the decision

to accept or reject a scientific hypothesis can have different consequences in different cases. As a result, scientists themselves cannot reliably assess the nature and extent of the inductive risk that will arise from this decision. In Jeffrey's account, this assessment should rather be done by the practitioners who are to make the relevant decision and thus to take the inductive risk.

In order to illustrate the foregoing claim, Jeffrey considers the case of a doctor who is confronted with the decision to inoculate a child with a specific polio vaccine. In this case, there is an inductive risk associated with this decision, as there is no certainty in the hypothesis that bears on the decision to use the vaccine, namely the hypothesis that the vaccine is free from active polio virus. Jeffrey then compares this case with the case of a veterinarian who is also confronted with the decision to inoculate a monkey with the same vaccine. The inductive risks associated with the decisions to be made by the doctor and the veterinarian are different from each other, as the outcomes of these decisions and the negative utilities (or error costs) associated with these outcomes can be different. Therefore, to "accept or reject [the polio vaccine hypothesis] once for all is to introduce an unnecessary conflict between the interests of the physician and the veterinarian" (Jeffrey, 1956, p. 245). In Jeffrey's account, this conflict can be avoided if the scientist who is in charge of making a judgment about the hypothesis in the aforementioned cases only provides the doctor and the veterinarian with a probability in the sense of degree of confirmation on given evidence. Then, based on this probability and the utilities they will attach to the possible outcomes of their decisions, the doctor and the veterinarian can make their own decisions as to whether to use the vaccine.

Jeffrey's account aims to keep the scientists' evaluation of hypotheses free from the influence of social values. To this end, he distinguishes between the epistemic and normative aspects of this evaluation and thereby he delimits the role of social values to the normative aspect that concerns the application of scientific hypotheses. The epistemic aspect consists of assigning probabilities to hypotheses with respect to available empirical evidence, while the normative aspect consists of assigning numerical utilities to the prospective outcomes associated with the applications of scientific hypotheses (in conjunction with probabilities assigned by scientists). In Jeffrey's account, only the epistemic aspect is the scientists' proper task, whereas the normative aspect is the task of those who are to make use of or to be affected by the outcomes of the applications of hypotheses. The normative task can involve social value judgments that have no bearing on the epistemic evaluation of scientific hypotheses. Therefore, Jeffrey does not deny the role of social value judgments in the assessment of inductive risk that arises from the acceptance or rejection of hypotheses. Rather, he denies the role of value judgments in the probabilistic evaluation of scientific hypotheses that is concerned with the determination of their degrees of confirmation with respect to available experimental evidence.

## 3 Inductive risk in the context of model construction

Unlike Rudner and Hempel who associated the concept of inductive risk merely with theory or hypothesis acceptance or rejection, Douglas suggested that "just as there is inductive risk for accepting theories, there is inductive risk for accepting methodologies, data, and interpretations" (2000, p. 565). Contrary to Jeffrey's account, she thus

suggested that "non-epistemic values are a required part of the internal aspects of scientific reasoning for cases where inductive risk includes risk of non-epistemic consequences" (Ibid, p. 559). In a similar vein, Daniel Steel argued that "probabilistic assessments of evidence or degrees of confirmation themselves depend on accepting data, background knowledge, and probability models, and hence are also subject to the argument from inductive risk" (2015, p. 87). Another prominent paper that aims to extend the inductive risk argument to the aspects of scientific inquiry beyond theory or hypothesis choice is a joint paper by Justin Biddle and Eric Winsberg where they visit Jeffrey's account in relation to climate modelling (Biddle & Winsberg, 2009). The argument of this paper was elaborated by Winsberg (2012) in a later paper that is a notable exception to the neglect of inductive risk in the context of model construction.

While Winsberg's discussion in this paper is not sufficiently detailed and explicit about the relevance of inductive risk to the construction of climate models, he suggests that the choice of methodologies used for constructing climate models bears upon the inductive risk that will arise from the acceptance of these models:

> When a climate modeller is confronted with a choice between two ways of solving a modeling problem, she may be aware that each choice strikes a different balance of inductive risks with respect to a problem that concerns her at the time. Choosing which way to go, in such a circumstance, will inevitably reflect a value judgment. (Ibid., p. 124)

In the same paper, Winsberg also points out that the assessment of inductive risk associated with the acceptance of climate models is made through what is called uncertainty quantification, which consists in "giving quantitative estimates of the degree of uncertainty associated with the predictions of climate models" (Ibid., p. 111). Winsberg argues that performing uncertainty quantification requires a division of labor between the epistemic and social value-laden considerations, and thus that the latter considerations are essential to the decision concerning model acceptance in climate science. Therefore, Winsberg's argument goes against Jeffrey's claim that probability assignments (associated with theoretical predictions) are based on purely epistemic considerations.[4]

Winsberg's discussion of inductive risk in the context of climate modeling suggests that different methodological choices can lead to the construction of climate models involving different inductive risks, thus illustrating Douglas' claim that inductive risk is also associated with the choice of methodologies. It should however be noted that Douglas does not specifically consider methodologies used for theory or model construction, nor does she draw a link between inductive risk and the context of theory or model construction. I shall argue that such a link exists as a result of one important aspect of model construction in science, namely that it is subject to inductive underdetermination in the sense that no finite set of empirical evidence can uniquely determine a model. I shall call this aspect of model construction its underdetermination by available empirical evidence.

[4] Winsberg's argument was criticized by Morrison (2014) and Parker (2014). Morrison argues that uncertainty quantification "involves subjective elements, [but that] in no way does that detract from its status as an epistemic exercise" (Morrison 2014, p. 939). Parker states that Winsberg's "argument exaggerates the influence of social values on estimates of uncertainty in climate prediction" (Parker, 2014, p. 27).

The underdetermination of model construction signifies the non-uniqueness of the set of methodological choices that can be used to construct a model that is compatible with the available empirical evidence. This in turn means that different (or rival) models that are compatible with the same set of empirical evidence but that yield different or conflicting empirical results can be constructed by adopting different methodological choices. Adopting a particular set of methodological choices for the construction of a model is always accompanied by a particular set of background assumptions (such as simplifications, idealizations and approximations) that is necessary to construct the model. These modelling assumptions can in turn translate into uncertainties in the empirical consequences of the model. Since accepting a model entails accepting its empirical consequences, the inductive risk taken in accepting a particular model amounts to the risk of accepting its empirical consequences that can be erroneous, to varying extents, due to the fact they involve uncertainties that are caused by the assumptions underlying the construction of the model. Winsberg illustrates this aspect of model construction in the context of climate science as follows:

> While the construction of climate models is guided by basic science—science in which we have a great deal of confidence—these models also incorporate a barrage of auxiliary assumptions, approximations, and parameterizations, all of which contribute to a degree of uncertainty about the predictions of these models. (Ibid., p. 116)

The above discussion suggests that the inductive risk associated with accepting a model is built into the model through its underlying background assumptions. The decision of how much inductive risk is to be built into the model is different from the decision to accept or reject it. It is therefore important to distinguish between the assessment of inductive risk in the context of model construction and its assessment in the context of model evaluation concerning model acceptance or rejection. The former assessment is made by modellers who are confronted with making methodological choices concerning model construction, while the latter assessment is made by those who are confronted with the decision to accept or reject a given model. Given that these contexts and the associated decision makers are different and also that in both contexts the decisions are underdetermined by the available empirical evidence, these assessments are based on different value judgments. For example, in the context of climate science, the value judgments concerning the construction of climate models are different from the value judgments of the policy makers who are to make use of the results of these models. In the remainder of this paper, I shall examine the role of inductive risk in both the construction and evaluation of ML binary classification models.

## 4 Construction and evaluation of ML binary classification models

### 4.1 Essential elements and aspects of ML

The kind of learning problem for which ML is used is how to mathematically model the mapping relationship between a set of inputs $X = (x_1, \ldots, x_n)$ and its corresponding set of outputs $Y = (y_1, \ldots, y_n)$ in a given set of data for a specified task, such as

classification tasks. In the case of binary (two-class) classification which is widely used in social applications, the inputs are assigned to the set of outputs consisting of two distinct classes denoted by a binary tuple $(y_1, y_2)$.[5] For example, in the context of healthcare, classifying patients into two groups (such as cancer or non-cancer) based on their observed characteristics is a binary classification task for which ML is used (see, e.g., Kourou et al., 2015). ML is based on the assumption that there exists a target function $f : X \rightarrow Y$ that correctly maps all the given inputs to the corresponding outputs for a given task. This means that the function $f$ is the true mathematical representation of the foregoing mapping relationship. In ML applications, given a set of inputs for which corresponding outputs need to be predicted for a specified task, the function $f$ is unknown, and ML is used to find the function $g$, called final hypothesis, that optimally represents the sought mapping relationship and thus provides an optimal approximation to the unknown function $f$. The function $g$ is considered to be the ML model of the foregoing mapping relationship.

The essential elements of ML include the training data set, the learning algorithm, the hypothesis set ($H$), and the loss (or error) function together with an optimisation procedure. $H$ contains the candidate functions that can be used to represent the mapping relationship under consideration. The learning algorithm is a set of instructions usually executed through a program run on a computer or a computer-like machine. The learning algorithm uses the training data to find (or compute) which function—i.e., the function $g$—in $H$ is the optimal approximation to the function $f$. There are two main types of ML, namely supervised and unsupervised ML.[6] In the case of supervised learning, the training data contains examples of inputs together with the corresponding correct outputs, while in the case of unsupervised learning the training data does not contain the information regarding corresponding outputs for the given inputs. In this paper, I shall discuss only supervised learning, as ML classification models are typically based on this type of ML.

In the case of supervised learning, the training data set ($D$) consists of $N$ instances or examples: $D = \{x^t, y^t\}_{t=1}^N$, where $x^t$ stands for the arbitrary dimensional input, and $y^t$ stands for the associated desired (correct) output, which is "0/1" in binary-class learning problems.[7] ML is used to find an approximation to $y^t$ given $x^t$ for each instance $t$ in the training data set. This approximation is denoted by a parametric function $g(x^t|\theta)$, where $\theta$ stands for the set of parameters. One particular set of values for $\theta$ instantiates one particular hypothesis in the hypothesis set, i.e., $g \in H$. The difference between the desired output $y^t$ and $g(x^t|\theta)$ is called the approximation error, i.e., the error due to using $g$ in the solution of the learning problem. Approximation error is measured through a loss (or error) function: $L(y^t, g(x^t|\theta))$, which is used to quantify the loss incurred due to incorrect predictions (such as misclassifications) made during training.[8] The term training error is used to denote

[5] See Alpaydin (2010, Chapter 2), for a discussion on ML for classification with multiple classes.

[6] There are also two other types of ML, namely semi-supervised ML (see Zhu & Goldberg, 2009) and reinforcement ML (see Alpaydin, 2010, Chapter 18).

[7] In this part, I follow the treatment offered in the widely used textbook by Alpaydin (2010).

[8] Different loss functions are used in the context of ML. For a discussion, see Alpaydin (2010, Section 13).

the total loss incurred due to the approximation error, which is defined to be the sum of losses over the individual input–output instances classified during the training: $E(\theta|D) = \frac{1}{N}\sum_{t=1}^{N} L(r^t, g(x^t|\theta))$. The chosen learning algorithm is implemented, according to a pre-determined procedure, in such a way that instances drawn from the training data, such as $(x^1, y^1), (x^2, y^2), \ldots (x^n, y^n)$, are introduced to the learning algorithm until the latter finds the set of optimal parameters that minimizes the total loss: $\theta^* = Arg\ min_\theta E(\theta|D)$. As a result of this optimisation process,[9] the learning algorithm singles out the function $g(x^t|\theta^*)$ as the optimal solution of the given learning problem, which is the ML model of the sought mapping relationship.[10]

While training error is concerned with the construction of a ML model, what is called test error is concerned with the evaluation (or testing) of the model and denotes the error made by the model when it is applied to a new data set, called test data, which has not been introduced to it before.[11] What is called generalization error (Jakubovitz et al., 2019) is defined to be the difference between training error and test error, and it is taken to be a measure of how well the model generalizes from the training data to new data sets and thereby makes accurate predictions.[12] There are two main problems associated with ML optimisation that are indicative of insufficient generalization. The first is called overfitting and occurs when a ML model fits the training data too closely, meaning that the model has learned the peculiarities of the training data, such as mislabeled instances, that are not present in new data sets and thus irrelevant to the representation of the sought mapping relationship. Overfitting is thus indicated by a significantly large generalization error. The other problem is called underfitting and occurs when training error is high, meaning that the model has not learned enough from the training data. Both overfitting and underfitting should be avoided in order for an ML model to optimally generalize from the

[9] In this paper, ML optimisation refers to the optimisation of model parameters (such as weights), rather than the optimisation of hyperparameters, which control the training process, such as number of training iterations.

[10] ML optimisation is based on the principle of empirical risk minimization in statistical learning theory (Vapnik, 1999). According to this principle, the goal of ML learning is to find the function $g(x, \alpha_0)$ that minimizes the expected value of the total loss given by the risk functional: $R(\alpha) = \int L(y, g(x, \alpha))\, dP(x, y)$, also called empirical risk, where $P(x, y)$ is a joint probability distribution over the training data; $L(y, g(x, \alpha))$ represents the loss or discrepancy between the response $y$ of the supervisor to a given input $x$ and the response $g(x, \alpha)$ provided by the learning machine. Since the joint probability distribution is unknown to the learning algorithm in practical applications, empirical risk is computed by averaging the loss function $L$ over the training set: $R(\alpha) = \frac{1}{N}\sum_{i=1}^{N} L\big(y_i, g\big(x_i, \alpha\big)\big)$.

[11] What is called cross validation is an intermediate phase between training and testing phases (see Alpaydin, 2010, Chapter 2). Cross validation serves to choose the best hypothesis in the hypothesis set, namely the one that fits the validation data (which is different from both training and test data sets) the best. For example, cross validation can be used to determine the right polynomial order if the hypothesis set contains polynomial hypotheses of different orders.

[12] Both training and generalization errors concern the accuracy of the predictions of ML models and arise from the inaccuracies in the modelling assumptions. There is a different kind of error, called software error, that arises from the malfunctioning of computer software. Since ML algorithms are software, software error can arise in the context of ML model construction and contribute to both training and generalization errors. See Symons and Alvarado (2016) for a discussion of the social implications of software error in the context of big data software applications, such as Google's flu tracker.

training data to new data sets and thus to yield accurate predictions about new data sets (Dietterich, 1995). Underfitting can be avoided by reducing the total loss by means of training. However, further training increases generalization error and thus causes overfitting. What is called *early stopping* is a criterion used to stop training before it starts incurring overfitting. Finding the right criterion for early stopping is a challenging issue for which various methods have been proposed in the ML literature (see, e.g., Prechelt, 2012).

### 4.2 Supervised ML: an illustrative example

In order to illustrate supervised ML, I shall make use of the following often-used textbook example where ML is used to obtain a credit decision model that a bank can use to make a classification of its credit applicants as low-risk and high-risk customers (Abu-Mostafa et al., 2012; Alpaydin, 2010). Based on this classification, the bank will accept applications from low-risk customers and reject applications from high-risk customers. This is a binary-classification problem that can be solved by using supervised ML. In this example, supervised ML is used to model the mapping relationship between the set of inputs—i.e., $(x_1, \ldots, x_n)$—consisting of the records of the credit applicants and the set of outputs—i.e., $(y_1, y_2)$—consisting of two distinct classes labeled as low-risk and high-risk customers, corresponding to the approval or disapproval of their credit applications, respectively. The training data used in this example is taken to be the credit scoring data from the past years that contains the personal records of the former credit applicants with respect to various financial factors, including their income, savings, profession, age, past financial history, and the information as to whether previous credits were paid back or not. Since we suppose that the kind of ML used in this example is supervised ML, the training data contains instances of correct pairs of inputs (records of former applicants) and their corresponding outputs (credit decisions). Figure 1 illustrates the basic structure of ML as it is used in the banking example.

Since the foregoing financial factors are not equally important for the final credit decision, the credit score of an applicant is determined by multiplying each financial factor by its corresponding weight parameter according to its relative importance in the final decision. The credit score of an applicant is taken to be the weighted sum of the values of its financial factors, and it can therefore be defined through a linear polynomial function: $\sum_{i=1}^{n} w_i x_i$, where $x_1, \ldots, x_n$ stand for the values of the financial factors considered relevant to the credit decision, and $w_i, \ldots, w_n$ stand for the weight parameters associated with these financial factors. A credit applicant will be classified by the credit approval formula as a low-risk customer, and thus her credit application will be accepted, if her credit score is greater than a threshold. Otherwise, the applicant will be classified as a high-risk customer and thus her credit application will be rejected. Therefore, mathematically speaking, a given credit application will be accepted if $g(x) = 1$, i.e., if $\sum_{i=1}^{n} w_i x_i > d$; and it will be rejected if $g(x) = 0$, i.e., if $\sum_{i=1}^{n} w_i x_i \leq d$, where $d$ is a threshold parameter. The hypothesis set for the solution of this classification problem can thus be expressed in the form of a unit step function as: $g(x|w, d) = \begin{cases} 1 \ if(\sum_{i=1}^{n} w_i x_i) > d) \\ 0 \ if(\sum_{i=1}^{n} w_i x_i) \leq d) \end{cases}$. The set of parameters

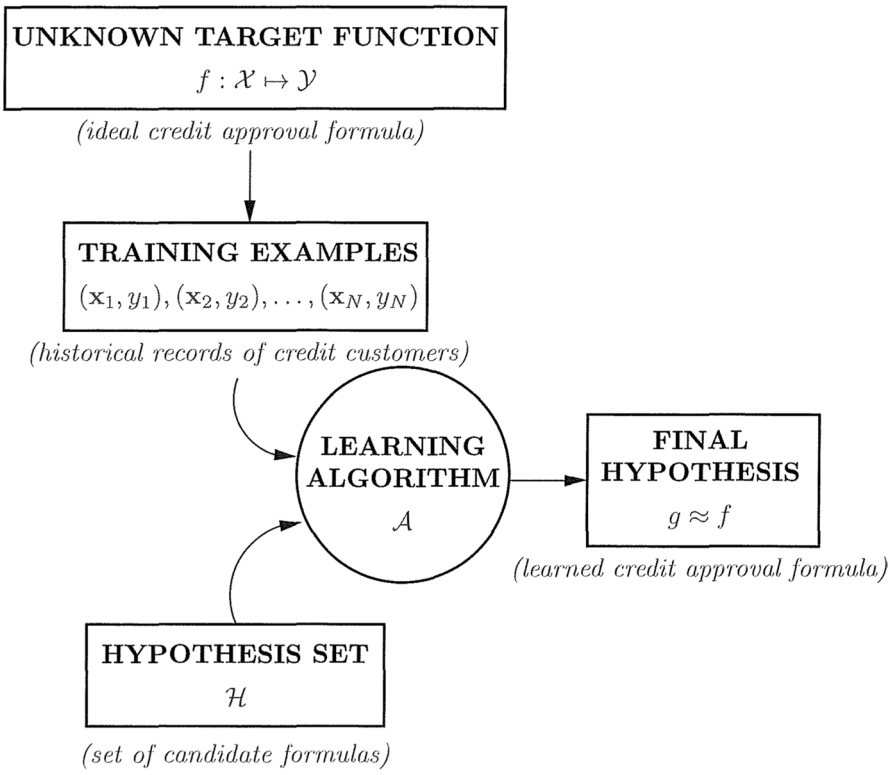


**Fig. 1** The basic set-up of ML in the banking example ( Source: Abu-Mostafa et al., 2012, Fig. 1.2)

$\theta = \{w, d\}$ for this hypothesis set consists of the weight parameters and the threshold parameter, which is also treated as a weight parameter.[13]

Through an optimisation process, the learning algorithm determines the optimal values for the set of parameters that will in turn single out one particular hypothesis as the credit decision model. In supervised ML, the (weight and threshold) parameters are typically set to zero or small random numbers at the beginning of the optimisation process (or training). In the above example, the learning algorithm uses these initial values to calculate $g(x|w, d)$ and thereby predicts the output for each input introduced from the training data. Since the examples of correct input–output pairs are introduced to the learning algorithm in the case of supervised ML, each time an incorrect prediction (i.e., a misclassification) is made, the learning algorithm updates (or adjusts) the parameters so as to reduce the total error (or loss) in the classification of customers.[14] In this way, the algorithm *learns* how to correctly

[13] The threshold parameter is the bias term that controls the predisposition of the model to yield one of the final outcomes.

[14] This is typically done by using the gradient descent algorithm that computes the gradient of the loss function in order to find the parameters (weights and bias) of the model that minimize the total loss (see Alpaydin, 2010, Section 10).

classify the given inputs. The optimisation process continues until the error attains its minimum value, without causing overfitting. This signifies that the optimal values for the parameters have been computed by the learning algorithm. The optimal values for the parameter set $\theta_{opt} = \{w_{opt}, d_{opt}\}$ determined as a result of the optimisation process single out $g(x|w_{opt}, d_{opt})$ as the credit decision model that will be used by the bank to decide if a future credit application should be accepted or rejected. This is a linear and parametric binary classification model. It is also deterministic as it produces either acceptance or rejection decision, without any assignment of probabilities to these outputs.[15]

### 4.3 Underdetermination of ML model construction and inductive risk

Training data is the sole empirical element in ML model construction, in the sense that it provides the necessary empirical evidence for modelling, without singling out the methodological choices concerning the other essential modelling elements. These methodological choices are therefore underdetermined by the training data, meaning that the solution of a given learning problem, namely the estimation of the unknown target function, can be given by different ML models based on different methodological choices determined by the preferences of the relevant ML modellers. This indicates the non-uniqueness of the solution of a given learning problem by ML (Alpaydin, 2010; Domingos, 2012), thus illustrating what I have previously called (inductive) underdetermination of model construction.

A particular set of modelling assumptions chosen by the ML modeller reflects her preference for a particular (mathematical) representation of the target function over its possible representations. In this sense, the ML modeller's choice of a particular set of modelling assumptions indicates a kind of preference bias that introduces subjectivity into ML model construction. In the ML literature, this preference bias is called inductive bias (Alpaydin, 2010; Mitchell, 1997) due to the inductive character of ML model construction, namely that it consists in generalizing from the input–output relations found by the learning algorithm in the training data to a model (of the target function) that (mathematically) represents the mapping relationships in new data sets having the same characteristics as the training data. Inductive bias is essential to ML model construction in the sense that it enables drawing the foregoing inductive inference necessary to construct a ML model.

Since ML model construction is basically an estimation process based on inductive inference, the results of ML models are always prone to error, as indicated by the presence of training error and test error. Test error can be regarded as a measure of the inductive risk associated with the use or application of a model, as it signifies the extent of error involved in its results when it is applied to data sets different

[15] ML classification models can also be probabilistic (see, Bishop, 2006, Chapter 2).

from the training data. Unlike test error, training error can be directly controlled by the modeller as it can be minimized during model construction in such a way that the problems of overfitting and underfitting are avoided. Training error can thus be regarded as an indication of the error-proneness of the model based on its performance on the training data. In this regard, training error indicates the extent of inductive risk built into the model during its construction. This means that the minimization of training error during ML optimisation serves to minimize the inductive risk built into the ML model.[16]

It should be noted that inductive bias does not necessarily give rise to discriminatory bias such as racial or gender bias (see, e.g., Mehrabi et al., 2019). Inductive bias can involve discriminatory bias in cases where some modelling elements, including training data, are discriminatory against certain groups of people. In such cases, the predictive performance of the constructed model will be undermined because ML optimisation process will be carried out in a discriminatory way. For example, if the training data involves discriminatory bias,[17] it is not fairly representative of the data sets to which the model is to be applied. The lack of representativeness of training data in this sense would result in errors in the predictions of ML models. Therefore, discriminatory bias can give rise to inductive risk. However, this kind of inductive risk cannot be managed during training because discriminatory bias is an essential feature of the training data with respect to which ML optimisation is carried out. Since discriminatory bias is not a legitimate form of bias, it should be properly addressed during ML model construction. However, identifying and handling discriminatory bias has its own challenges that are currently discussed in the ML literature (see, e.g., Bordia & Bowman, 2019).

### 4.4 Cost-sensitive ML optimisation

ML optimisation consists in minimizing the total loss due to training errors. ML is called cost-sensitive if the optimisation is carried out by using different costs for different types of training errors, such as false positives and false negatives in binary classification tasks. ML is called cost insensitive if the optimisation is carried out by using equal costs for different types of training errors (Elkan, 2001; Ling & Sheng, 2011). In the context of binary classification tasks, the total loss is minimized with respect to what is called a cost matrix whose entries consist of numerical utilities attached to different misclassification costs (Elkan, 2001). Typically, in practical applications of ML, while zero utility is assigned to correct classifications, finite utilities are assigned to misclassifications in order to weigh their costs differently. In the credit application example, the costs for different classification errors can be

[16] This is mathematically indicated by the principle of empirical risk minimization on which ML optimisation is based (see Footnote 10 in this paper). It is also worth noting that in the context of ML this principle should be understood in the sense of reducing the expected risk (or error) to the least possible value without causing overfitting.

[17] This can occur intentionally such as due to discriminatory labelling of training data, or unintentionally such as using biased historical data. For a discussion, see, e.g., Barocas and Selbst (2016), who also suggest that value judgments can also affect the assignment of class labels to instances in training data sets.

noted as follows: Cost-1 is the cost of misclassifying (rejecting an application from) a low-risk customer (false negative or reject); and Cost-2 is the cost of misclassifying (accepting an application from) a high-risk customer (false positive or accept). These costs yield the following cost matrix:

| | Actual accept | Actual reject |
|---|---|---|
| Classify accept | No error, zero cost | False accept, Cost-2 |
| Classify reject | False reject, Cost-1 | No error, zero cost |

In the context of binary classification tasks, cost-insensitive ML optimisation boils down to minimizing the rate of misclassifications, which is ratio of the total number of misclassifications to the total number of classifications. This means that the aim of cost-insensitive ML optimisation is to maximize the predictive accuracy of classification models. Whereas in the case of cost-sensitive ML, due to differing error costs, ML optimisation does not boil down to minimizing the rate of misclassifications, and thus it does not aim at maximizing the predictive accuracy of classification models.

In the credit application example, the inductive risk arises from the risk of error to be caused by the use of the ML credit-decision model in making decisions on credit applicants. This inductive risk should be expected to have different negative financial consequences or costs for the bank. This means that Cost-1 and Cost-2 should be different from each other, meaning that ML optimisation needs to be carried out in a cost-sensitive way. The ratio of these costs affects the extent of ML optimisation, i.e., the extent of training, needed to determine the parameters of the credit-decision model.

The foregoing cost-ratio is determined by the trade-off made by the bank between two financial values, namely financial risk-taking and financial security. Depending on this trade-off, the following two cases will arise. If the bank values financial risk-taking more than financial security, it will be more willing to accept applications than to reject them. This means that the cost of misclassifying (rejecting) a low-risk customer is more than the cost of misclassifying (accepting) a high-risk customer, i.e., Cost-1 > Cost-2. On the other hand, if the bank values financial security more than financial risk-taking, it will be more willing to reject applications than to accept them. This means that the cost of misclassifying (accepting) a high-risk customer is more than the cost of misclassifying (rejecting) a low-risk customer, i.e., Cost-2 > Cost-1. Since the ratios of misclassification costs are different in the foregoing cases, the associated ML optimisation processes will result in different (optimal) values of model parameters, thus yielding different credit-decision models. In both cases, the cost ratio to be used in the ML optimisation depends on the bank's inductive risk profile, which determines how the bank makes the trade-off between financial risk-raking and financial security.

Cost-sensitive ML has found many societal applications in contexts where classification tasks involve unequal error costs (see, e.g., Turney, 2000; Johnson & Khoshgoftaar, 2019). It is also used in classification tasks with imbalanced classes (He & Garcia, 2009) where "one or several classes (the majority classes) vastly

outnumber the other classes (the minority classes), which are usually the most important classes and often with the highest misclassification costs" (Garcia et al., 2012, p. 347). In these cases, the minority class, called the positive class, and the majority class, called the negative class, typically have significantly different error costs. This means that classification tasks involving imbalanced classes are a special case of classifications tasks involving unequal error costs. For instance, in medical applications, patients having cancer constitute the minority class in a given population, while those not having cancer constitutes the majority class. The cost of a false negative, i.e., misclassifying a cancer case as non-cancer, has a much higher cost than a false positive, i.e., misclassifying a non-cancer case as cancer. This is because the former case might result in the delay of the treatment of the case, which is a life-threatening situation, while the former case requires carrying out some additional medical checks on patients.[18] The above example illustrates that in cases where classification tasks involve imbalanced classes, inductive risk stems mainly from false negative errors that are due to incorrect predictions of ML classification models about minority classes, such as classification of a cancer case as non-cancer. In these cases, cost-sensitive ML optimisation is useful to handle inductive risk as it enables determining the parameters of ML classification models in a way that takes account of unequal error costs associated with false negative and false positive errors.

### 4.5 Evaluation of ML binary classification models

In binary classification tasks, true positives (*TP*) and true negatives (*TN*) denote respectively the positive and negative instances (in a given data set) correctly classified (as positive and negative respectively) by a binary classification model, while false positives (*FP*) and false negatives (*FN*) denote respectively the positive and negative instances incorrectly classified (as negative and positive respectively) by the model. The fit of a ML classification model to the test data is measured through what is called a performance metric. One such metric is classification accuracy that is defined as the ratio of the correctly classified instances to the total number of instances classified in a given data set: $\frac{TP+TN}{TP+TN+FP+FN}$. However, classification accuracy is not an appropriate metric to evaluate the predictive performance of ML classification models with unequal error costs, as it does not take account of unequal costs assigned to *FP* and *FN* (Provost et al., 1998). For the same reason, classification accuracy is also not an appropriate metric to evaluate the performance of ML classification models with imbalanced classes. In this context, since classification accuracy does not take into account differing error costs, it can give misleading results due to the fact that the instances in the negative (majority) class outnumber those in the positive (minority) class. In the extreme case where all the instances in the positive class are misclassified and those in the negative class are correctly classified, i.e., $TP + TN = TN$, $TP + TN + FP + FN \approx TN$, classification accuracy approximates to

[18] Imbalanced data sets can arise in many real-world applications of ML classification models concerning the detection of rare events such as frauds and natural disasters (for a review, see, e.g., Haixiang et al., 2017).

1. This is grossly misleading given that the main aim of the classification problem is to correctly classify the instances of the positive (minority) class. A model classifying all instances as belonging to the majority class, e.g., all patients as non-cancer, has a very high predictive accuracy due to the fact that the instances of the majority class greatly outnumber those of the minority class, but such a model is useless as it fails to correctly classify any of the instances belonging to the minority class.

What is called receiver operating characteristic (ROC) plot is a prominent metric used to evaluate the predictive performance of ML classification models with unequal error costs (Provost & Fawcett, 1997).The ROC metric consists of a plot of true positive rate: $TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$ (on the y-axis) versus false positive rate: $FPR = \frac{FP}{N} = \frac{FP}{FP+TN}$ (on the x-axis). Here, *TPR* is the ratio of the positive (relevant) instances correctly classified to all the positive instances in a given data set, while *FPR* is the ratio of the negative instances incorrectly classified to the ratio of all the negative instances. *TPR* is also called sensitivity (or recall), and it is a measure of how well a classification model can correctly classify the relevant instances, i.e., those in the positive class. For the sake of completeness, one can also define false negative rate: $FNR = \frac{FN}{P} = \frac{FN}{FN} + TP = 1 - TPR$, and true negative rate: $TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = 1 - FPR$. *TNR* is also called specificity and it is a measure of how well a classification model can correctly classify the instances in the negative class.

In the ROC plot, each point corresponds to a $TPR - FPR$ pair associated with a classification model as shown in Fig. 2. The location of a classification model in the ROC plot shows how the model makes the balance between *TPR* and *FPR*, i.e., between the benefits and errors associated with the positive class, respectively. The ROC plot has a number of interesting aspects that enable comparing the predictive performance of different classification models (Ibid.). The point (0,1) (exemplified by the classification model *D* in Fig. 2) represents perfect classification performance which amounts to correctly predicting all positive and negative instances, i.e., both sensitivity and specificity are equal to one. The diagonal line $y = x$ represents what is called random guessing, for which *TPR* and *FPR* are expected to be equal to each other. For example, the performance of the classification model *C* (in Fig. 2) is close to random guessing. The lower right triangle in the ROC plot represents those classification models, such as *E* in Fig. 2, whose performance is worse than that of random guessing, because in this region *FPR* is greater than *TPR*. The upper left triangle in the ROC plot represents those classification models, such as *A*, *D* and *B* in Fig. 2, that perform better than random guessing. It is important to note that "one point in ROC space is better than another if it is to the northwest (*TPR* is higher, *FPR* is lower, or both) of the first" (Fawcett, 2006, p. 862). This is because as one moves towards the northwest direction in the ROC plot, *TPR* increases while *FPR* decreases. This provides a way to compare the predictive performances of different classification models. For example, in Fig. 2, the predictive performance of the model *B* is better than that of the model *C*, as *B* is to the northwest of *C* in the ROC plot. However, when one compares the models *A* and *B* in Fig. 2, since neither of them is located northwest of the other, one needs to strike a balance between *TPR* and *FPR* in making a choice between these models.

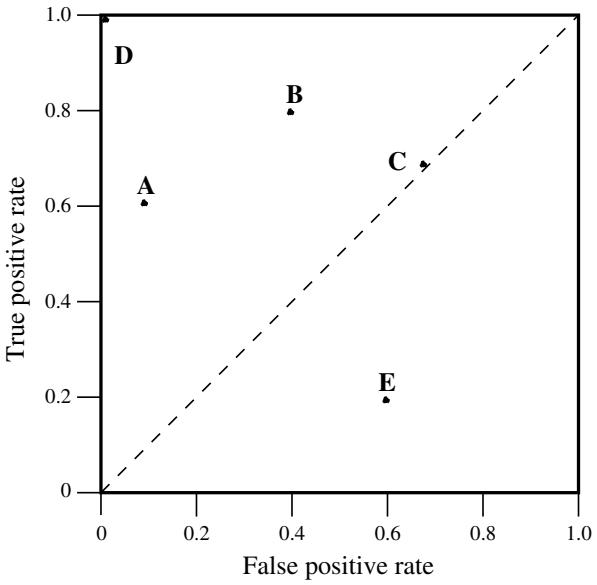


**Fig. 2** A ROC plot showing five classification models (Fawcett, 2006, p. 862)

An important drawback of the ROC metric is that it provides an overly optimistic evaluation of the performance of ML classification models with highly imbalanced classes (Davis & Goadrich, 2006). In these cases, since "the number of negative examples greatly exceeds the number of positives examples […] a large change in the number of false positives can lead to a small change in the false positive rate used in ROC analysis" (Ibid., p. 233). This means that even if a classification model has a high sensitivity, it still produces a high number of *FP* as compared to *TP*, if its precision is low. Precision is defined as the ratio of the positive instances correctly classified to all the instances classified as positive, i.e., $\frac{TP}{TP+FP}$. What is called $F_1$ score (or measure) is a metric widely used to evaluate ML classification models with imbalanced classes (Forman & Scholz, 2010). $F_1$ score combines sensitivity and precision into one single metric that is defined as their harmonic mean: $F_1$ score $= 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity}$, which ranges between 1 and 0.[19] The idea underlying $F_1$ score is that the harmonic mean of two numbers gets smaller than their arithmetic mean, as the difference between them increases. This means that in order for a classification model to have a high $F_1$ score, both its precision and sensitivity should be high. For classification tasks with imbalanced classes, such as classification of patients, a high $F_1$ score is an important evaluation criterion for ML classification models.[20] Both ROC and $F_1$ score are measures of how well a model makes the balance between *TPR* and *FPR*. Unlike ROC, $F_1$ score involves precision and thereby "captures the effect of the large number of negative examples on the [model's]

[19] The harmonic mean ($H$) of $n$ positive real numbers $x_i, \ldots, x_n$ is given by the following formula: $H = \frac{1}{\sum_{i=1}^{n} \frac{1}{x_i}}$.

[20] For a review, see Johnson and Khoshgoftaar (2019).

performance" (Davis & Goadrich, 2006, p. 233). As a result, it provides a more accurate evaluation of ML classification models with imbalanced classes (Saito & Rehmsmeier, 2015).[21]

## 5 Value judgments in the construction and evaluation of ML classification models

The cost matrix is used to quantify and thereby to minimize the total loss and thus the inductive risk associated with the intended use of a ML classification model. The error costs specified in a cost matrix are determined by the ML modeller by weighing the relative importance of *FP* and *FN* with respect to the inductive risk profile of the intended users of the model. Here, inductive risk profile can be defined as the extent of the inductive risk that the intended users of a ML classification model can afford to take. Since ML optimisation is carried out by means of a learning algorithm with respect to the error costs specified in a cost matrix, the inductive risk profile of the intended users of a ML classification model directly bears upon the optimisation process underlying the construction of the model. As I have previously argued, the underdetermination of ML model construction by the training data makes it necessary for ML modellers to make methodological choices to determine the elements of ML model construction. As part of this modelling process, ML modellers also need to make methodological choices to determine the error costs that are essential to the cost-sensitive ML optimisation. To this end, they need to make social value judgments by taking into consideration the inductive risk profile of the intended users of ML classification models.

In the credit application example, the values of financial risk-taking and financial security are the particular social values that bear on the inductive risk profile of the bank, which is the user of the ML credit-decision model. Since these financial values conflict with each other, the ML modeller needs to make a trade-off between them in assigning different costs to *FP* and *FN* in the cost matrix used for optimisation. As discussed earlier in this paper, conflicting financial value judgments regarding the relative importance of financial risk-taking and financial security in credit decisions lead to different ratios between the costs of *FP* and *FN*, which indicate different balances of the inductive risk associated with the bank's use of the ML credit-decision model. In the case of ML models used for the classification of cancer instances, patient safety is the particular social value that bears on the inductive risk profile of the users of these models. In this case, ML modellers assign significantly higher costs to *FN* than *FP*. This case illustrates that in the case of ML classification problems with unequal error costs (including imbalanced classes), in order for ML modellers to assess the relative importance of *FP* and *FN* and thereby to determine error costs to be used in ML optimisation, they need to make judgments based on social values that they find relevant to the inductive risk profiles of the users

[21] It is worth noting that several other metrics are also used for the evaluation of ML classification models with imbalanced classes (for a review, see Branco et al., 2015).

of these models. Therefore, value judgments based on social values are involved in the construction of ML classification models mainly through cost-sensitive ML optimisation. Depending on different choices of error costs, different ML optimisations can be performed on the same training data, resulting in different optimal values of the model parameters and thus in different ML classification models. This indicates that methodological choices concerning cost-sensitive ML model construction are epistemically unforced, meaning that there are no "decisive, purely epistemic grounds for considering one model-building option to be better than all other available options" (Parker, 2014, p. 26).

The discussion in the previous section shows that the choice of a ML classification model (from among a set of alternative models) depends on the evaluation of the model according to the chosen performance metric. ROC and $F_1$ score are two prominent metrics used to evaluate the performance of ML classification models with unequal error costs. The choice of a ML classification model depends on whether it is the optimal model in the sense that it strikes the best balance (among the available models) between *TPR* and *FPR* with respect to the users' inductive risk profile.[22] In the credit application example, ROC is an appropriate metric to evaluate the performance of different ML credit-decision models. The bank's trade-off between financial risk-taking and financial security, which is based on its inductive risk profile, will determine which ML credit-decision model is optimal in the foregoing sense. Therefore, like model construction, model choice is also epistemically unforced in the context of ML classification models, meaning that the evaluation of these models requires taking into account the relevant users' inductive risk profiles that are dependent on social values.

The above discussion also suggests that the appropriateness of the choice of ROC and $F_1$ score metrics for the evaluation of ML classification models with unequal error costs is based on the consideration that unlike classification accuracy, these metrics make it possible to weigh *TPR* and *FPR* with respect to the inductive risk profiles of the users of these models. $F_1$ score is preferred for the evaluation of ML classification models with imbalanced classes, because, unlike ROC, it is sensitive to class imbalances. These considerations underlying the appropriateness of the metric choice are purely epistemic, in the sense that they are independent of social value judgments. Similarly, the measurement of a model's fit to test data is purely epistemic. This measurement is based solely on whether the predictions of the model about class labels are true, which can be ascertained by comparing these predictions with the actual class labels present in the test data. This comparison must be performed in a way that is entirely free from social value judgments in order for the chosen performance metric to provide an objective measurement of the model's fit to the test data. Unlike the appropriateness of the metric choice, the appropriateness of the test-data choice depends on considerations based on social values in societal applications of ML. For instance, test data is required to be non-discriminatory in this context. This requirement is the result of a value judgment typically based on

[22] This balance can also be seen as being between *FPR* and *FNR*, given that $TPR = 1 - FNR$.

fairness, which is a social value (see, e.g., Mehrabi et al., 2019; Biddle, 2020).[23] However, as I have pointed out above, this type of considerations based on social value judgments are irrelevant to the measurement of the model's fit to the chosen test data.

Therefore, the evaluation of ML classification models with unequal error costs consists of two distinct and related parts. The first part is purely epistemic and concerned with the measurement of the extent of the model's fit to the test data through an appropriate performance metric. The second part is concerned with the adequacy of this fit for the intended applications or uses of the model.[24] The latter part of the evaluation is normative in the sense that it is based on the users' social value judgments bearing on the assessment of the inductive risk associated with the application of ML classification models. The distinction between the epistemic and normative parts of model evaluation in the context of ML binary classification models indicates that social value judgments are not as pervasive as Douglas' account suggests, namely that they do not penetrate into the measurement of model's fit to test data. It is also worth noting that the above distinction accords with Jeffrey's suggestion that social value judgments should only influence the decisions concerning the applications of scientific hypotheses.

## 6 Algorithmic and epistemic opacity in deep ML

The binary ML classification model used in the credit application example illustrates what is a called a *perceptron*, which consists of one single *artificial neuron* (or *node*).[25] As shown in Fig. 3, an artificial neuron is a structure where the weighted sum of the inputs plus the bias term are transformed into an output by means of an activation function, which limits the range of possible outputs to a specified range.[26] An underlying assumption of the perceptron model is linearity, as a result of which the inputs

[23] What is called "fairness-aware ML" is a new approach that is based on the idea that the requirement of fairness imposes certain constraints on ML model construction (see, e.g., Zafar et al., 2017). This approach is consistent with the dependence of ML model construction on social values, given that fairness is a social value. In the same literature, the role of fairness in model evaluation is also discussed, especially regarding the relationship between model fit and fairness in ML classification models (see, e.g., Zliobaite, 2015; Menon & Williamson, 2018). Different metrics have been proposed to measure the fairness of ML classification models (see, e.g., Verma & Rubin, 2018). This literature suggests that the measurement of model fairness is separate from the measurement of model fit. Therefore, the former measurement can be seen as part of the normative evaluation of ML classification models. This means that the measurement of model fit to test data is not affected by value judgments based on fairness.

[24] Here, I draw upon a distinction made by Morrison (2014) between model accuracy and adequacy of model accuracy for purpose or application. In Morrison's account, the evaluation of model accuracy is purely epistemic, whereas the evaluation of the adequacy of model accuracy is normative in the sense that it involves non-epistemic considerations. Instead of accuracy, I prefer using the term model fit to data, given that accuracy of classification is not an appropriate aspect to measure for the evaluation of ML classification models with unequal error costs.

[25] The concept of artificial neuron is defined in analogy to neurons in the human brain. For a discussion, see Haykin (2009, Chapter 2).

[26] In the case of the perceptron model, the activation function is the unit step function. In neural network models, sigmoid is the widely used activation function that limits the outputs to [0, 1].

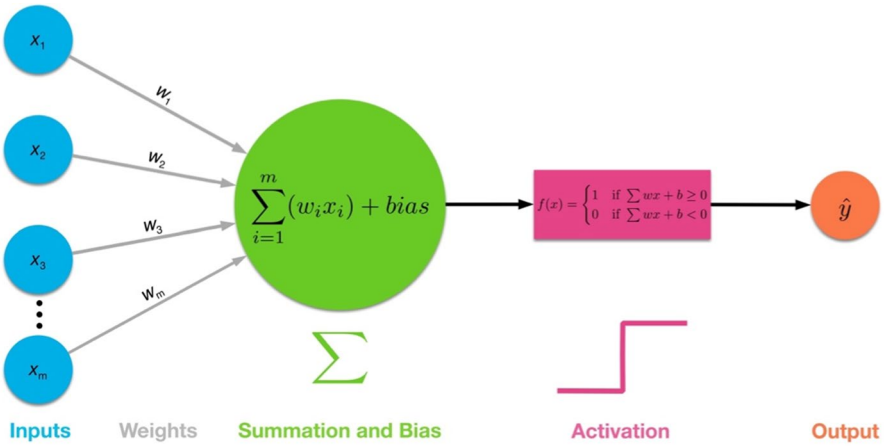


**Fig. 3** The structure of the perceptron model ( Source: Kang, 2017)

are directly mapped to the corresponding outputs. However, linearity does not capture the relationships that can exist among the inputs and thus the intricate relationships between the inputs and the corresponding outputs. Neural networks are a prominent class of non-linear ML models that are widely used in many societal applications of big data.[27] The kind of ML underlying the construction of these models is called *deep* ML, as it takes places through multiple intermediate layers of interconnected neurons that are incorporated between the input and output layers as in the case of multilayer perceptrons as shown in Fig. 4. These intermediate layers are often called *hidden layers* in the sense that their inputs and corresponding outputs are not directly read off from the network, unlike those of the input and output layers. The following passage succinctly summarizes the role of hidden layers in the construction of deep neural networks:

> In real-world problems, input variables tend to be highly interdependent and they affect the output in combinatorially intricate ways. The hidden layer neurons allow us to capture subtle interactions among our inputs which affect the final output downstream. Another way to interpret this is that the hidden layers represent higher-level "features" or attributes of our data. Each of the neurons in the hidden layer weigh the inputs differently, learning some different intermediary characteristics of the data, and our output neuron is then a function of these instead of the raw inputs. By including more than one hidden layer, we give the network an opportunity to learn multiple levels of abstraction of the original input data before arriving at a final output (Kogan, 2021).

Multilayer perceptrons[28] are the most prominent class of deep neural networks where each node in a given layer (except the output layer) connects to every node in the next

[27] For a recent review, see, e.g., Emmert-Streib et al. (2020). For a review aimed at philosophers, see Buckner (2019).

[28] The other types of neural networks include what are called convolutional neural networks and recurrent neural networks (see Haykin, 2009).

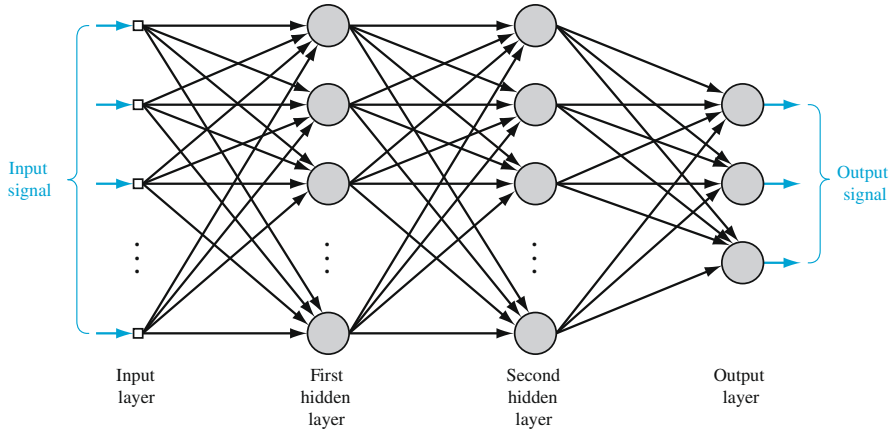


**Fig. 4** The structure of a multilayer perceptron with two hidden layers ( Source: Haykin, 2009, Fig. 4.1)

layer through the weight parameters that scale the outputs of the nodes in the previous layer. Each node in the neural network can be seen as a perceptron whereby the mapping relationship between its inputs (from the nodes in the previous layer) and outputs is modeled in the way described in Section 4 of this paper. Each node receives, as an input, a weighted sum of the outputs of all the nodes in the previous layer. By means of an appropriate activation function (e.g., sigmoid function), this weighted sum is transformed into an output that will be an input for the next layer. This process repeats in the forward direction until the values of the nodes in the output layer are computed as the final output(s) of the neural network.[29] A multilayer perceptron network can thus be regarded as a ML model that consists of a non-linear function representing the mapping relationship between the set of initial inputs and corresponding final output(s).[30]

In the construction of deep ML models, the initially assigned weights evolve into their optimal values through an optimisation process that takes place through hidden layers each of which acts as an intermediary step in this process. Since deep ML models are constructed by using big data, the total amount of computation required for the optimisation of weights in these models is so vast that it is impracticable for human agents to study and understand the entire optimisation process. This in turn indicates that the implementation of the learning algorithm in deep ML is opaque to human agents (Burrell, 2016; Creel, 2020) and thus that optimisation looks like a *black box* process, in the sense that human agents can access the initial inputs and final outputs, but not the inputs and outputs of the nodes in the hidden layers. As a result of this algorithmic opacity, while human agents can possess an understanding of the working logic of the learning algorithm, they

[29] The training of a neural network is similar to the training of a perceptron. The weights (including biases) of a network are computed by the backpropagation algorithm based on gradient descent. Since a neural network has hidden layers, the parameters are updated in a backward direction, i.e., from the output layers towards the input layer (see, Alpaydin, 2010, Section 11.).

[30] This non-linear function can be seen as a functional composition of weights and activation functions associated with hidden layers (see, e.g., Bauckhage et al., 2018).

lack an understanding of how exactly the initial weights are updated by the learning algorithm as instances from the training data are introduced to it.

Does algorithmic opacity entail epistemic opacity in the context of deep ML models? This question is relevant to understanding the effect of algorithmic opacity on the nature of the knowledge produced by deep ML models. One prominent definition of epistemic opacity is offered by Paul Humphreys as follows: "[a] process is essentially epistemically opaque to X if and only if it is impossible, given the nature of X, for X to know all of the epistemically relevant elements of the process" (Humphreys, 2009, p. 618). Based on this definition, Humphreys goes on to suggest that:

> distinguishing between the weaker and stronger senses [of epistemic opacity] is useful. It is obviously possible to construct definitions of 'partially epistemically opaque' and 'fully epistemically opaque' which the reader can do himself or herself if so inclined. What constitutes an epistemically relevant element will depend upon the kind of process involved. (Ibid.)

To use Humphreys' terminology, the values of the weight parameters are epistemically relevant elements of the process of ML model construction, in that the knowledge of them would enable human agents to trace the outputs of the model back to their inputs, thereby accounting for how the model actually arrives at its predictions. Algorithmic opacity makes it practically impossible for human subjects to examine how the weights linking inputs to outputs through hidden layers are optimised by the learning algorithm so as to minimize the total error. This means that human agents do not have the knowledge of the evolution of the weights, which are usually initially assigned to small random values, into their optimal values. Therefore, the process of optimisation in deep ML is essentially epistemically opaque—in the sense of Humphreys' account. An important consequence of this epistemic opacity concerns the predictions of deep ML models. In order to account for these predictions, it is necessary to understand how the optimal values of their weight parameters are computed by the learning algorithms, because these predictions are obtained by virtue of the optimal values of the weight parameters. Since this information is not available due to the problem of algorithmic opacity, it is often stated in the ML literature that deep ML models yield predictions without explanations.[31]

The epistemically relevant elements of ML model construction also include the essential elements of this process—namely, training data, learning algorithm, hypothesis set, and loss function. Similarly, the value-judgments underlying the methodological choices of ML modellers are also epistemically relevant elements, in that they enable ML modellers to determine the foregoing essential elements of ML model construction. It should also be noted that one can know the essential elements of a ML model construction process without knowing their underlying value judgments. Since algorithmic opacity concerns the optimisation (of weight parameters)

[31] Yet, this is currently an open issue that is actively researched within the field of what is called explainable ML. For a review, see, e.g., Samek et al. (2017). The epistemological aspects of ML predictions have been recently examined by philosophers in terms of explanation, understanding, interpretability, and novelty; see, e.g., Zednik (2021), Erasmus et al. (2020), Ratti (2020), and Sullivan (2019).

part of the process of ML model construction, the knowledge concerning the essential modeling elements and their underlying value judgments are not affected by this kind of opacity. This means that human agents can investigate how these epistemically relevant elements have been determined prior to the process of ML optimisation. In the above discussion, I have argued that such an investigation should look into the value-judgments underlying the methodological choices concerning these elements. This suggests that the part of the ML model construction concerning the determination of the essential modelling elements is epistemically transparent to human agents to the extent that their underlying value judgments are known by them. ML modellers are obviously in a better position than other human agents in knowing the modelling elements and their underlying value judgments. However, there is no fundamental difference between them with respect to epistemic opacity, as the modelling elements and their underlying value judgments can be learned by other human agents through the investigation of the modelling process. Therefore, to use Humphrey's terminology, while the part of the (deep) ML model construction concerning the optimisation of the weight parameters is fully epistemically opaque due to the fact that this process is algorithmically opaque, the part of the modelling process concerning the choice of modelling elements is partially epistemically opaque, depending on the extent of the lack of knowledge regarding the underlying value judgments.

## 7 Conclusions

In this paper, I have argued that the construction of ML classification models illustrates inductive underdetermination of model construction, in the sense that the methodological choices underlying the construction of these models are underdetermined by the training data, which constitutes the sole empirical evidence for ML model construction. Since ML classification models are typically used for societally relevant purposes, their construction requires ML modellers to assess and thereby minimize the inductive risk that will arise from the intended uses of these models. This assessment is made through the quantification of error costs. The methodological choices concerning this quantification are underdetermined by the training data. Therefore, given that the inductive risk associated with the application of ML classification models has social implications, ML modellers need to make social value judgments in order to quantify the error costs to be used in the minimization of inductive risk. The above conclusions concerning the role of inductive risk and social values in the construction of ML classification models do not quite fit into the inductive risk argument, as has been discussed thus far in the philosophical literature. According to this argument, inductive risk bears solely on the decision to accept or reject a theory (or a model), and as a result the role of social value judgments in scientific methodology is limited to the assessment of inductive risk that pertains to this decision.

The above conclusions also conflict with Jeffrey's normative suggestion that the assessment of inductive risk with respect to social values should not be a part of scientific methodology. Social value judgments bear on the construction

of ML classification models because these are optimisation models, meaning that their parameters are optimised so as to serve their intended purposes. The extent of optimisation required to construct a ML classification model is determined by the inductive risk profile of its users. Therefore, optimisation is the main reason why the assessment of inductive risk by ML modellers is an essential part of the construction of ML models. This is generally true for optimisation models, namely the kind of models optimised from the beginning to serve their intended purposes. I thus suggest that the construction of optimisation models requires the minimization of inductive risk that will arise from their intended uses. Whereas inductive risk does not typically figure in model construction in physical and biological sciences where the aim of modeling is mainly to account for natural phenomena, such as the modelling of sub-atomic phenomena in high-energy physics and the modelling of gene structures in molecular biology. Since the occurrence of a natural phenomenon is independent of the intended use of the model to be constructed about this phenomenon, the modelling of the phenomenon is independent of the inductive risk that will arise from the use of the model.

The foregoing considerations indicate that minimization of inductive risk is key to the construction of ML classification models used for societal purposes. This is also true for the construction of other types of ML models whose intended uses give rise to inductive risk. Therefore, ML modellers need to consider the inductive risk profiles of the intended users of the models that they aim to construct as well as the social values that they find relevant to their inductive risk profiles. In this regard, the increasing reliance on ML within big data analytics has the potential to lead to a big data modelling practice where the epistemic task needed to construct models of big data also involves modellers' engagement with the intended users of these models.

**Acknowledgements** I would like to thank Owen King, Christin Seifert and the two anonymous referees of this journal for helpful comments and discussions. Earlier versions of this paper have been presented at conferences and colloquia at Concordia University, the University of Stuttgart and the University of Twente. I thank the audiences at these events for their feedback.

## Declarations

**Ethical approval** The submitted paper does not report any results for which ethicalapproval is required.

**Informed consent** The submitted paper does not report any results for which informed of individuals is needed.

**Conflict of Interest** No conflict of interest with any third party.

## References


Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H.-T. (2012). *Learning from data*. AMLbook.com.

Alpaydin, E. (2010). *Introduction to machine learning*. The MIT Press.

Barocas, S., & Selbst, A. D. (2016). Big data's disparate impact. *California Law Review, 104*, 671–732.

Bauckhage, C., Ojeda, C., Schücker, J., Sifa, R., & Wrobel, S. (2018). Informed machine learning through functional composition. In *Proceedings of LWDA* (pp. 33–37).

Biddle, J. B. (2020). On predicting recidivism: Epistemic risk, tradeoffs, and values in machine learning. *Canadian Journal of Philosophy*. https://doi.org/10.1017/can.2020.27

Biddle, J. B., & Winsberg, E. (2009). Value judgments and the estimation of uncertainty in climate modeling. In P. D. Magnus & J. Busch (Eds.), *New waves in the philosophy of science* (pp. 172–197). Palgrave MacMillan.

Biddle, J. B., & Kukla, R. (2017). The geography of epistemic risk. In K. Elliott & T. Richards (Eds.), *Exploring inductive risk: Case studies of values in science* (pp. 215–237). Oxford University Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bordia, S., & Bowman, S. R. (2019). Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop* (pp. 7–15), Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Branco, P., Torgo, L., & Ribeiro, R. P. (2015). A survey of predictive modelling under imbalanced distributions. ACM Computing Surveys, August 2016 Article No: 31.

Burrell, J. (2016). How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society, 3,* 1–12.

Buckner, C. (2019). Deep learning: A philosophical introduction. *Philosophy Compass, 14*, e12625.

Cabitza, F., Rasoini, R., & Gensini, G. F. (2017). Unintended consequences of machine learning in medicine. *JAMA, 318*, 517–518.

Creel, K. A. (2020). Transparency in complex computational systems. *Philosophy of Science*. https://doi.org/10.1086/709729

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 233–240).

Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM Computing Surveys, 27*, 326–327.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*, 78–87.

Douglas, H. (2000). Inductive risk and values in science. *Philosophy of Science, 67*, 559–579.

Douglas, H. (2017). Science, values, and citizens. In M. P. Adams, Z. Biener, U. Feest, & J. A. Sullivan (Eds.), *Eppur si muove: Doing history and philosophy of science with Peter Machamer.* Springer.

Dressel, J., & Farid, H. (2018). The accuracy, fairness, and limits of predicting recidivism. *Science Advances, 4*(1), eaao5580.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of International Joint Conference on Artificial Intelligence* (pp. 973–978).

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., & Dehmer, M. (2020). An introductory review of deep learning for prediction models with big data. *Frontiers in Artificial Intelligence, 3*, 4.

Erasmus, A., Brunet, T. D. P., & Fisher, E. (2020). What is interpretability? *Philosophy of Technology*. https://doi.org/10.1007/s13347-020-00435-2

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874.

Forman, G., & Scholz, M. (2010). Apples-to-Apples in cross-validation studies: Pitfalls in classifier performance measurement. *ACM SIGKDD Explorations, 12*, 49–57.

García, V., Sánchez, J. S., Martín-Félez, R., & Mollineda, R. A. (2012). Surrounding neighborhood-based SMOTE for learning from imbalanced data sets. *Progress in Artificial Intelligence, 1*, 347–362.

Ghumbre, S. U., & Ghatol, A. A. (2012). Heart disease diagnosis using machine learning algorithm. In S. C. Satapathy, P. S. Avadhani, & A. Abraham (Eds.), *Proceedings of the International Conference*

*on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam*, India, January 2012. Advances in Intelligent and Soft Computing (Vol. 132). Springer, Berlin, Heidelberg.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications, 73*, 220–239.

Haykin, S. (2009). *Neural networks and learning machines*. Pearson Education Inc.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*, 1263–1284.

Hempel, C. G. (1965). Science and human values. In *Aspects of scientific explanation and other essays in the philosophy of science* (pp. 81–96). The Free Press.

Humphreys, P. (2009). The philosophical novelty of computer simulation methods. *Synthese, 169*, 615–626.

Iliadis, A., & Russo, F. (2016). Critical data studies: An introduction. *Big Data & Society, 3*, 1–7.

Jakubovitz, D., Giryes, R., & Rodrigues, M. R. D. (2019). Generalization error in deep learning. In H. Boche, G. Caire, R. Calderbank, G. Kutyniok, R. Mathar, & P. Petersen (Eds.), *Compressed sensing and its applications* (pp. 153–193). Springer.

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data, 6*, 27.

Jeffrey, R. C. (1956). Valuation and acceptance of scientific hypotheses. *Philosophy of Science, 22*, 237–246.

Kang, N. (2017). Multi-layer neural networks with sigmoid function— Deep learning for rookies (2). https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f. Accessed 6 Sept 2021.

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal, 13*, 8–17.

Kogan, G. (2021). Neural networks. Retrieved May 26, 2021, from https://ml4a.github.io/ml4a/neural_networks/

Lima, A. N., Philot, E. A., Trossini, G. H. G., Scott, L. P. B., Maltarollo, V. G., & Honorio, K. M. (2016). Use of machine learning approaches for novel drug discovery. *Expert Opinion on Drug Discovery, 11*, 225–239.

Ling, C. X., & Sheng, V. S. (2011). Cost-sensitive learning. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of machine learning*. Springer.

Mena, J. (2011). *Machine learning forensics for law enforcement, security, and intelligence*. CRC Press.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. ACM Computing Surveys, July 2021 Article No: 115.

Menon, A. K., & Williamson, R. C. (2018). The cost of fairness in binary classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency* (Vol. 81, pp. 107–118), PMLR.

Mitchell, T. (1997). *Machine learning*. McGraw Hill.

Morrison, M. (2014). Values and uncertainty in simulation models. *Erkenntnis, 79*, 939–959.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data, 2*, 1.

Okasha, S. (2002). Underdetermination, holism and the theory/data distinction. *The Philosophical Quarterly, 52*, 303–319.

Parker, W. S. (2014). Values and uncertainties in climate prediction, revisited. *Studies in History and Philosophy of Science, 46*, 24–30.

Phua, C., Lee, V., Smith-Miles, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. Retrieved May 26, 2021, from https://arxiv.org/abs/1009.6119

Prechelt, L. (2012). Early stopping—But when? In G. Montavon, G. B. Orr, K. R. Müller (Eds.), *Neural networks: Tricks of the trade*. Lecture notes in computer science (Vol. 7700). Springer

Provost, F., & Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of Third Internat. Conf. on Knowledge Discovery and Data Mining (KDD-97)* (pp. 43–48). AAAI Press, Menlo Park, CA.

Provost, P., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In J. Shavlik, (Ed.), *Proceedings of ICML-98* (pp. 445–453). Morgan Kaufmann, San Francisco, CA.

Ratti, E. (2020). What kind of novelties can machine learning possibly generate? The case of genomics. *Studies in the History and Philosophy of Science (Part A), 83*, 86–96.

Rudner, R. (1953). The scientist qua scientist makes value judgments. *Philosophy of Science, 20*, 1–6.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One, 10*(3), e0118432.

Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learningmodels. *ITU Journal: ICT Discoveries, Special Issue, 1*, 39–48.

Symons, J., & Alvarado, R. (2016). Can we trust big data? Applying philosophy of science to software. *Big Data & Society, 3*(2), 2053951716664747.

Steel, D. (2015). Acceptance, values, and probability. *Studies in History and Philosophy of Science, 53*, 81–88.

Sullivan, E. (2019). Understanding from machine learning models. *British Journal for Philosophy of Science*. https://doi.org/10.1093/bjps/axz035

Turney, P. (2000). Types of cost in inductive concept learning. In *Proceedings of the Cost-Sensitive Learning Work-shop at the 17th ICML-2000 Conference* (pp. 15–21). Stanford University, California: NRC.

van Liebergen, B. (2017). Machine learning: A revolution in risk management and compliance? *Journal of Financial Transformation, 45*, 60–67.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks, 10*, 988–999.

Verma, S., & Rubin, J. (2018). Fairness definitions explained. In *IEEE/ACM International Workshop on Software Fairness* (pp. 1–7).

Winsberg, E. (2012). Values and uncertainties in the predictions of global climate models. *Kennedy Institute of Ethics Journal, 22*, 111–137.

Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production & Manufacturing Research, 4*, 23–45.

Zafar, M. B., Valera, I., Rodriguez, M. G., & Gummadi, K. P. (2017). Fairness constraints: Mechanisms for fair classification. In *Proceedings of 20th AISTATS* (pp. 962–970).

Zednik, C. (2021). Solving the black box problem: A normative framework for explainable artificial intelligence. *Philosophy & Technology, 34,* 265–288.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 3*, 1–130.

Zliobaite, I. (2015). On the relation between accuracy and fairness in binary classification. In *ICML Workshop on Fairness, Accountability, and Transparency in Machine Learning*. Retrieved May 26, 2021, from https://arxiv.org/abs/1505.05723