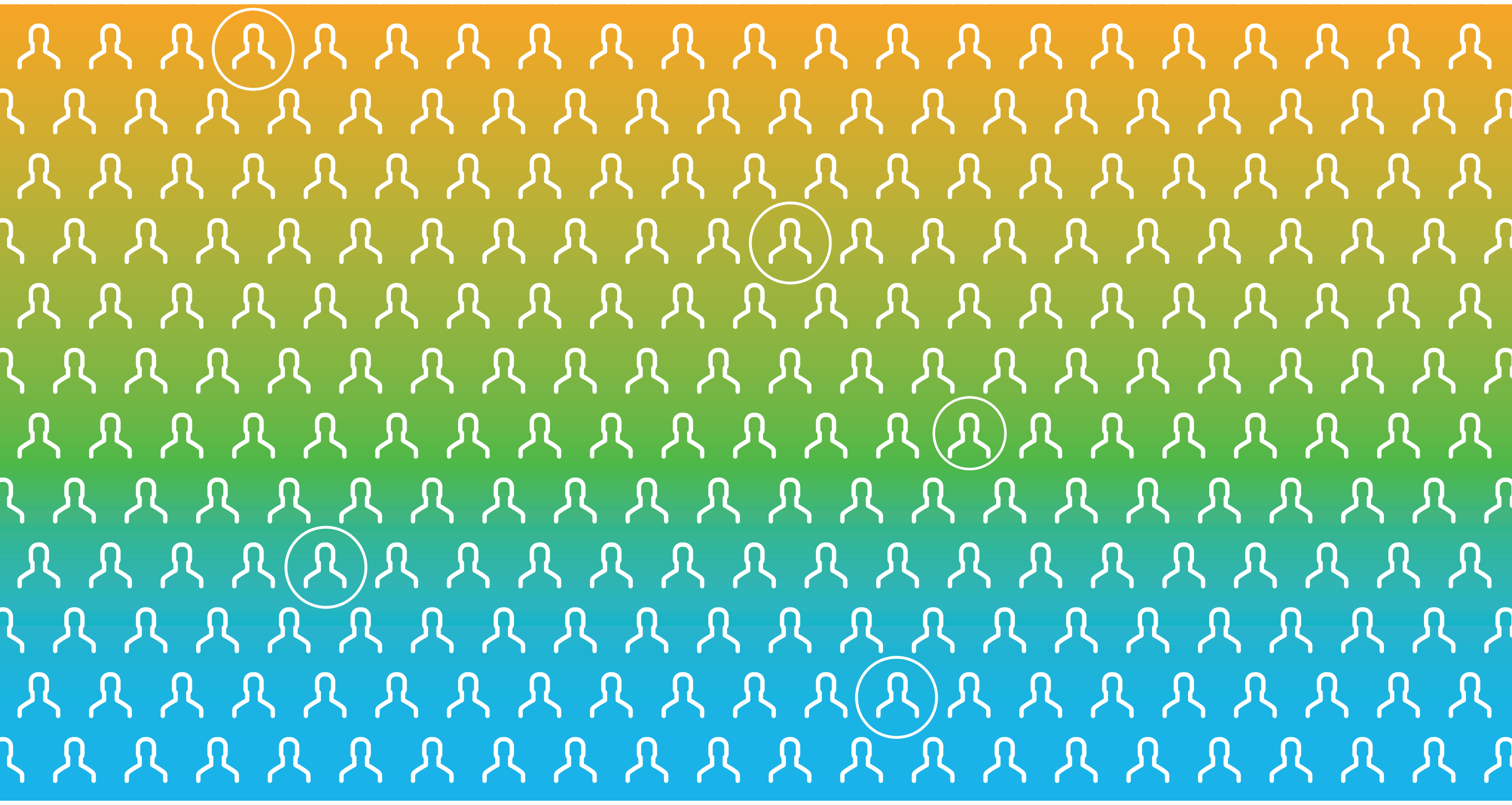
Automated decisions and Artificial Intelligence in human resource management

Guideline for reviewing essential features of AI-based systems for works councils and other staff representatives

Prof. Dr. Sebastian Stiller (TU Braunschweig)
Jule Jäger (TU Braunschweig)
Sebastian Gießler (AlgorithmWatch)
May 18, 2021



Published by

ALGORITHM WATCH

Funded by

Hans Böckler Stiftung

The automation of human resources management (HR) will continue to advance in companies. On the one hand, this offers new opportunities; on the other, it harbors the potential for conflict. Not only will HR automation change the way work is organized, but it will also have a profound impact on the structure of companies. Therefore, the introduction and further development of these systems is an important topic for works councils and other employee representatives. The purpose of this guide is to enable members of these interest groups to shape the process competently.

Structure of the guide

After a brief introduction to the topic, a series of questions about automated HR systems is presented. If possible, these are questions that works councils should ask management before implementing new systems, although they can also be asked about systems already in place. Along with the questions, this guide also explains how to classify answers and what they should include.

Introduction

Many people have a vague idea of what is meant by "Artificial Intelligence" (AI): computer systems that think in roughly the same way as humans. A more precise understanding is generally considered to be the domain of experts. From an expert's point of view, AI is actually quite different: systems that are comparable to human thinking, so-called "strong" AI, do not exist, and there is no binding definition for "weak" AI. The term "AI" is used so arbitrarily and for such different procedures that one should always ask how the procedure works. The same is true for other terms used in automated HR management, such as "talent analytics," "workforce analytics," "people analytics," or "human resources analytics". It is misleading and unhelpful to assume that these systems are similar to human thinking.

How a process works is only to a lesser, often insignificant extent, a matter for experts. It is always a subject of procedures that arrive at statements on the basis of data. Why and in what sense these statements are true or useful depends on the justification for the procedure. Substantial parts of these justifications are not mathematical and can be discussed equally by users and those affected, as by developers.

In addition, many procedures do not simply make true or false statements, but rather they provide hints and recommendations. To use such hints responsibly, one must know how they came about. Such an understanding is possible. And ultimately, the reasoning behind a statement must be good enough to convince those who bear the consequences and responsibility for it.

Therefore, in the context of workplace co-determination, it must be possible to discuss how AI processes are used in HR functions. This guide contains questions that make this possible by helping to drill down to the justificatory, discussion-worthy properties and contexts of a software system.

The question is: What distinguishes modern HR management systems with the addition of "analytics" (e.g., Human Resources Analytics (HRA)) from classic enterprise software, the so-called Human Resources Information Systems (HRIS)? Previous systems provided information about directly available data (e.g., employee training days, grading of job applicants) or derived simple, clearly defined statements from the data (e.g., the average value of the number of training days of all employees in a department, the average grade of job applicants). However, new systems make additional statements whose connection to the underlying data is not immediately apparent. These statements range from evaluations and recommendations (e.g., custom-fit suggestions for further training, pre-selection of applications) to automatic or strongly predetermined decisions.

Thus, HRA offer above all an extension by components, which make statements possible, whose justifications are not mathematically provable, but shaped by rationales, theories, and insights from psychology, business administration, human resources, and behavioral economics. Whether the underlying theories are accepted and whether the translation of the theories into computable criteria seem plausible are both up for discussion.

Of course, such HRA systems are not infallible. This is due to the fact that there are no clear and unambiguous answers to many questions in personnel management (assessment of application documents, decisions on organizational structure, compensation). Nevertheless, it can be advantageous to use HRA systems with the appropriate expertise. HRA systems can often consider much larger contexts than human decision-makers. Human intuition, experience, and judgment are subject to many weaknesses from which HRA systems are free. Making the criteria of HRA systems transparent can lead to greater reliability and fairness in an organization.

However, to derive their decision from data, HRA systems often use inappropriate or narrow criteria and are poor at handling individual cases. In addition, systems that use so-called machine learning (ML) – i.e., inferring their rules for the future from a sample of past decisions – depend on how good the past decisions were and how well they fit the future. A basic assumption of ML methods is that the future looks essentially the same as the past that is being learned from. For natural sciences like physics, this assumption is very reasonable. Where human decisions are involved, this assumption is often problematic.

The weaknesses of HRA systems are especially critical when a system remains opaque and cannot be assessed, questioned, controlled, overridden, and changed. Conversely, the strengths of HRA systems are particularly effective when a system is transparent and meaningfully integrated into the organization. Therefore, this guide includes questions about integrating the HRA system into the organization.

It is not possible for manufacturers and experts to assess and weigh the advantages and disadvantages of an HRA system universally for all organizations . What suits some companies will interfere significantly in others. Instead, this guide compiles questions that experts would also ask to discuss whether and how an HRA system should be used within a particular organization.

A system that is well understood and prudently integrated into operational processes can benefit everyone in the operation. This guide is intended to help ask the right questions to gain this understanding.

Support, autonomy and predictions

When evaluating HRA systems, it is helpful to distinguish between systems that support decisions and those that decide autonomously (on their own):

- A software system is a decision support system, if it provides suggestions to a person on a screen – e.g., who should be promoted – yet the decision may turn out differently since it remains fully in the hands of that person.

- If software is used that can make decisions on its own to some extent without human involvement – for example, in creating a binding shift schedule – it is an autonomous system.

HRA systems can be descriptive, predictive, or prescriptive. While the first two variants only represent existing data or derive predictions from them, prescriptive systems provide recommendations for action. Hybrid forms of these three types of systems also occur.

Guiding questions

The guideline is divided into four blocks of questions. The first block aims to clarify as concretely as possible what statements the software makes and what tasks it has in operation. The second block focuses on how these statements are justified. The answers to these questions should be as clear and plausible as would be expected when introducing non-electronic regulations and procedures – for example, for compensation or work management. The third block contains questions on the quality of algorithms and programming, which must be assessed by experts. The answers to questions in the third block, unlike in the second block, are, therefore, not primarily about understanding and plausibility, but about binding commitments and test results from experts. The fourth block of questions serves to clarify whether, against the background of the first three blocks, the system is being used appropriately and is integrated correctly into the processes of the organization.

Q1 What statements does the software make and what decisions does it affect?

Q1.1 Which software and which components of it are involved?

Software systems often consist of several components. Components and functionalities may be added in the course of deployment via updates or in a software-as-a-service (SaaS) model. Therefore, as a starting point for a dialog with management, it is important to gain an overview of the software system along with clarification of how adding components will be handled in the future.

It is also important to clarify where the company's data, especially employee data, is stored and who has access to it and for what purpose.

Q1.2 What statement does the software make and with what degree of truth?

The task of software systems is to make statements that are further used by users or even have direct consequences. Thereby a software system usually offers many functionalities, which make different kinds of statements.

Statements should be named as concretely as possible. For example: The software package provides results such as “Candidate X has personality structure Y”. In contrast, the following answer is too unspecific: “This software product finds the ideal employees for the company”. The more concretely the statement of a software package is defined, the better it can be checked later to see whether the software does what was promised.

Not all statements are simply true or false. Statistical statements, for example, are true with a certain probability. Other statements are more like guesses or suggestions. In this sense, every statement has a level of validity. The truth value depends on how well-founded the statement is. Conversely, the truth value affects how the statement can be used. It is important to be explicit about the truth value of the statements made by the software. If the truth value is high, it must be grounded very well. If it is low, one can only use the results in a limited way.

Q1.3 In which area does the software system prepare decisions or decide autonomously (on its own)?

The answer to this question should indicate the specific purpose for which the company wants to acquire the software system. Are existing processes to be (partially) automated, or are entirely new ones to be introduced? Which class is the software to be assigned to: Descriptive, predictive, prescriptive (see introduction)?

Q2 How does the software arrive at its statements?

Q2.1 What data does the software have access to?

For several reasons, it is important to clarify which data (especially employee data) the software has access to. Which data is used and which decisions are made based on which data concerns data protection and labor law.

Furthermore, it should be assessed in this question whether it seems conclusive to derive the statements of the software from the data used. Feedback effects should also be discussed here: For example, if employees know that their email behavior is being evaluated, how does it affect the way they work? Are the statements of the system still meaningful or are they invalidated by the possible reactions of the employees?

Q2.2 What criteria does the software use to make decisions?

At first glance, a software program seems too complicated to be understood by a layperson. But, before one starts programming, one considers criteria that can be used to generate the desired statement. Such considerations include justifications for the criteria used that are understandable to all people. The discussion of these criteria does not require programming knowledge. It can be conducted by users and stakeholders at least as well as by those who create the software.

Some software systems use criteria that directly justify what the software says. The criterion of a navigation system, for example, is the length of a route from start to destination. Among all possible routes, it selects the one that is the shortest according to this criterion. The statement of the navigation system: "This route is the shortest route" is, therefore, well-founded. In modern HR software, statements are often made for which no direct criterion can be given. An example of such a statement would be, "Employee XY is especially important to the project." Since "importance" is not a value that can be uniquely determined, vendors use a substitute criterion. For example, they calculate how central employee XY is to the network of email communications. This criterion is then used to rate the importance of employee XY. In order for the software to be used responsibly, such substitute criteria must be discussed.

For some statements, it is hard for us humans to give exact criteria. Examples are our taste in music or our ability to recognize friends and relatives without being able to describe exactly what we base this on. Machine learning methods are used to generate such statements with software. An ML method automatically generates a criterion based on examples, which is often too complicated to be understood by humans. If such methods are used, then separate questions, which follow here, should be asked. These questions are preceded by a short insertion for a better understanding of ML.

Software criteria must be mentioned. It is usually possible to explain the criteria sufficiently well without going into such detail that it would reveal so much about the product that intellectual property rights or trade secrets would be jeopardized. If the criteria are not mentioned, it is not possible to understand whether statements are justified, and ultimately the software cannot be used responsibly.

A short insertion: The most important things about machine learning

We can answer many questions without being able to give a clear criterion. For example: How do you recognize people you know well in photos? The criterion for this must somehow be composed of what you see in the photos. But people cannot list exactly what the criterion is composed of. ML finds such a complicated criterion if you give it enough example pictures to "learn" and tell it approximately what to look for in them.

So machine learning always consists of two phases: In the first phase, a criterion is learned using lots of example data (called "training data"). In the second phase, the criterion is used to evaluate new data. The indispensable bases for all these procedures are the training data and the data used during deployment, which should be approximately – not necessarily exactly – the same type. If the training data only contains photos that show a person from the front, then the criterion will not recognize that person in photos that show him or her from the back.

How does machine learning work? ML methods are not directly comparable to human learning. Often you hear that ML processes are only driven by data. This is also not true. Every ML process starts with a (very large) set of possible criteria, the hypotheses. So you do not specify the criterion itself, but the type of criterion. Without such a specification, the process does not work. With the help of this specification, one puts prior knowledge – or better: a prior opinion – into the procedure.

The so-called training algorithm selects a criterion from this type of criteria that fits the training data particularly well.

The word “learning” gives the impression that an ML procedure always leads to the truth sooner or later. However, very many attempts to solve a problem with ML procedures fail. Of course, people do not like to talk about this in public. Whether an ML procedure works or not can be judged almost exclusively in practical trials.

The quality of an ML procedure is measured with another set of data, the so-called test data. The ML procedure may provide a criterion that always makes correct statements relative to its training data, but fails when applied to the test data. It is important that the test data has not already been used in training. It is equally important that the test data match the data to which the learned criterion is to be applied. If an ML procedure is good at matching the musical tastes of executives, the same procedure may fail with apprentices.

Today, there is freely available software that allows you to build ML procedures yourself quite easily, without understanding much about how and why these procedures work. You assemble a procedure, train it on a training data set, and try it out to see if it produces good results when applied to test data. Programming an ML procedure today does not require a deep understanding of what difficulties their application may have.

With this in mind, the following questions should be discussed when using ML procedures:

Q2.3 What assumptions and scientific theories underlie the ML procedure used and why was this procedure chosen?

Q2.4 What training data was used for the ML procedure?

Careful consideration must be given to whether the training data and the use cases in operation match well enough to meet the basic requirements for the use of ML procedures. The training data serve as examples for the ML procedure to base its decisions on. Are the decisions in the examples exemplary for the operation? For example, if mainly men were hired in the past , an ML procedure that has been trained with the application data of successful individuals might discriminate against women.

In dialog with management, the management should demonstrate that an ML procedure is non-discriminatory, rather than conversely assuming that the system does not discriminate until discrimination has been demonstrated. Therefore, the following question should be asked:

Q2.5 How has the ML procedure been safeguarded against discrimination and other unintended influences from the training data?

For example, it is not sufficient here to keep gender out of the training data for the pre-selection of applicants. The ML procedure could more or less unambiguously infer gender from other characteristics and thereby still adopt gender discrimination as a pattern from the old training data.

Q2.6 How has the ML procedure been tested?

In practice, the quality of ML procedures is almost always checked by testing. What are the probabilities of correct and incorrect judgment when the procedure is applied to test data? Test data must be completely separate from the training and development of the ML procedure. Preferably, tests should have been conducted independently of the systems’ vendors. In addition, the test data must fit the use cases of the system in operation.

Q3 How is the quality of the system guaranteed?

Q3.1 Does the algorithm implement the criteria accurately?

The algorithm in the strict sense is a (mathematical) procedure to calculate the chosen criterion as fast as possible. The quality of systems can vary here. It may well be, for example, that the algorithm for a scheduling system does not produce the schedule that would be the best according to the specifications. It is difficult to obtain reliable information here. As a rule, only the judgment of experts or a comparison with competing products can help.

Q3.2 How is the quality of the implementation (the program code) ensured?

Once a criterion and a suitable algorithm have been designed, they must finally be programmed. Errors occur in the process, especially since software systems comprise very large amounts of programming code. Systematic testing is extremely difficult, even for experts. In general, there are three possibilities:

1. 1.The company creating the software can organize its work processes in accordance with specifications designed to reduce errors and find them during development.

2. 2.Code can be verified after the development process has been completed. However, this is only possible to a very limited extent.

3. 3.The manufacturer fixes occurring errors on an ongoing basis. Here, too, direct insight is not possible. Instead, the question is aimed at what quality control measures have been taken by the manufacturer.

Q3.3 Who created the software and which components were taken over from third parties?

Software packages are often extensive and consist of a large number of components. For example, it is common for component development to be outsourced to third parties or for certain software services (e.g., Amazon Web Services (AWS), IBM Watson) to be incorporated from an external source. The answer should provide information on whether the entire software package was programmed solely by the provider company and, if this is not the case, which components of which third-party providers were integrated.

Q4 How is the system integrated in the company?

Q4.1 What skills and knowledge are required by the users of the software?

In addition to a precise description of the qualifications that are required, the answer could also describe the need for further education and training that is expected from the employer.

Q4.2 Where does the responsibility for the software product lie within the company?

The answer to this question should identify, as precisely as possible, which department(s) and/or individuals have responsibility for the operation of the software. Is there a procedure for reporting suggestions, complaints, or concerns? Are there opportunities to change procedures or add features that are desired after the implementation?

Q4.3 How and by whom is it decided which functionality of the software will be used?

The answer should indicate who in the company decides what data should be made available for use with the software and what analyses should be performed. This may involve the use of sensitive personal data (e.g., contents of emails or the number of sick days). The answer should also include whether there is a clearly defined decision-making process on these issues and how co-determination rights are taken into account in the process.

Q4.4 Who determines the metrics used to define goals in the software?

People analytics and related software systems need a benchmark to make judgments in their analyses, such as when a target has been met or when something should be classified as “good,” “appropriate,” or “successful”. The answer to the question should indicate whether the company itself can define and change the relevant metrics or whether it is solely the company that produces the software that defines these metrics without any possibility of influence from the outside.

Q4.5 How transparent is the decision-making process? [..of the procedure in question.]

Is the process by which the system arrives at decisions transparent? Is it possible to check whether the decision is plausible and understandable? The answer should describe, for example, whether and how the user interface of the software shows intuitively and in a way that can be understood by humans, with which certainty the software makes a statement.

Q4.6 Are potential subtle influences excluded by the design of the software interface?

The type and manner of user guidance (UX, usability) and the presentation of user interface elements can encourage certain usage behavior. For example: a confirmation button is displayed large and green, while a rejection button is small and inconspicuous. Keywords here are “nudging” and “dark patterns”. The answer should state that the software product, especially where decisions can be confirmed or made, does not exert any subliminal influence through such methods.

Q4.7 Can automated decisions be corrected?

The company’s answer should state whether there is a possibility to report and intervene in case of doubts about (partially) automated decisions. Does the software product provide for this? How is this intervention designed? Is it realistically feasible in day-to-day work and process flows? Can the software be “overruled” manually? Who in the company has the right to do this?

Q4.8 Has a risk assessment been performed?

When introducing software, the company management should have the risks assessed not only in terms of technical aspects, but also in terms of data protection aspects. If possible, the answer to the question should contain the results of the risk assessment or a justification as to why none was carried out.

Automated decisions and Artificial Intelligence in human resource management

Guideline for reviewing essential features of AI-based systems for works councils and other staff representatives

Prof. Dr. Sebastian Stiller (TU Braunschweig)
Jule Jäger (TU Braunschweig)
Sebastian Gießler (AlgorithmWatch)
May 18, 2021

Publisher:
AW AlgorithmWatch gGmbH
Linienstr. 13
10178 Berlin

Contact: info@algorithmwatch.org

Proofreading:
Graham Holliday

Layout:
Beate Autering
Tiger Stangl
www.beworx.de

Published as a part of the research project
Automatisiertes Personalmanagement und Mitbestimmung

Website: https://algorithmwatch.org/en/auto-hr/

Funded by

Hans Böckler
Stiftung