DOI:10.1145/3495256

Given the complexity of data science projects and related demand for human expertise, automation has the potential to transform the data science process.

BY TIJL DE BIE, LUC DE RAEDT, JOSÉ HERNÁNDEZ-ORALLO, HOLGER H. HOOS, PADHRAIC SMYTH, AND CHRISTOPHER K.I. WILLIAMS

Automating Data Science

DATA SCIENCE COVERS the full spectrum of deriving insight from data, from initial data gathering and interpretation, via processing and engineering of data, and exploration and modeling, to eventually producing novel insights and decision support systems.

Data science can be viewed as overlapping or broader in scope than other data-analytic methodological disciplines, such as statistics, machine learning, databases, or visualization.[10]

To illustrate the breadth of data science, consider, for example, the problem of recommending items (movies, books, or other products) to customers. While the core of these applications can consist of algorithmic techniques such as matrix factorization, a deployed system will involve a much wider range of technological and human considerations. These range from scalable back-end transaction systems that retrieve customer and product data in real time, experimental design for evaluating system changes, causal analysis for understanding the effect of interventions, to the human factors and psychology that underlie how customers react to visual information displays and make decisions.

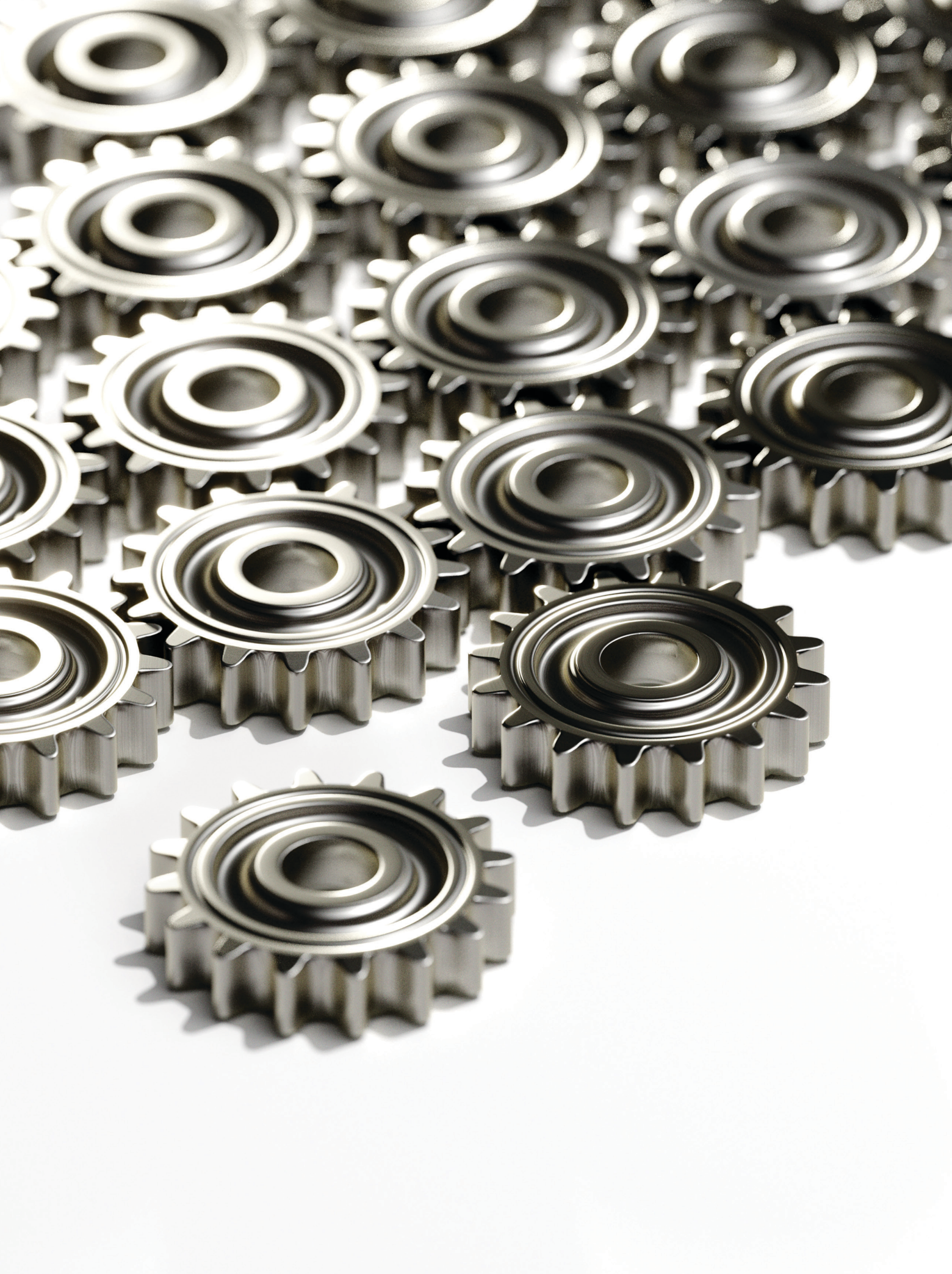
As another example, in areas such as astronomy, particle physics, and climate science, there is a rich tradition of building computational pipelines to support data-driven discovery and hypothesis testing. For instance, geoscientists use monthly global landcover maps based on satellite imagery at sub-kilometer resolutions to better understand how the Earth's surface is changing over time.[50] These maps are interactive and browsable, and they are the result of a complex data-processing pipeline, in which terabytes to petabytes of raw sensor and image data are transformed into databases of automatically detected and annotated objects and information. This type of pipeline involves many steps, in which human decisions and insight are critical, such as instrument calibration, removal of outliers, and classification of pixels.

The breadth and complexity of these and many other data science scenarios means the modern data scientist requires broad knowledge and experience across a multitude of topics. Together with an increasing demand for data analysis skills, this has led to a shortage of trained data scientists with appropriate background and experience, and significant market competition for limited expertise. Considering this bottleneck, it is not surprising there is increasing interest in automat-

» key insights

- **Automation in data science aims to facilitate and transform the work of data scientists, not to replace them.**
- **Important parts of data science are already being automated, especially in the modeling stages, where techniques such as automated machine learning (AutoML) are gaining traction.**
- **Other aspects are more difficult to automate, not only because of technological challenges, but because open-ended and context-dependent tasks require human interaction.**

ILLUSTRATION BY JUSTIN METZ



ing parts, if not all, of the data science process. This desire and potential for automation is the focus of this article.

As illustrated in these examples, data science is a complex process, driven by the character of the data being analyzed and by the questions being asked and is often highly exploratory and iterative in nature. Domain context can play a key role in these exploratory steps, even in relatively well-defined processes such as predictive modeling (for example, as characterized by CRISP-DM[5]) where human expertise in defining relevant predictor variables can be critical.

Figure 1 provides a conceptual framework to guide our discussion of automation in data science, including aspects that are already being automated as well as aspects that are potentially ready for automation. The vertical dimension of the figure reflects the degree to which domain context plays a role in the process. Domain context not only includes domain knowledge but also human factors, such as the interaction of humans with the technology,[1] the side effects on users and non-users, and all the safety and ethical issues, including algorithmic bias. These factors have various effects on data understanding and the impact of the extracted knowledge, once deployed, and are often addressed or supervised with humans in the loop.

The lower quadrants of *Data Exploration* and *Exploitation* are typically closely coupled to the application domain, while the upper quadrants of *Data Engineering* and *Model Building* are often more domain agnostic. The horizontal axis characterizes the degree to which different activities in the overall process range from being more open-ended to more precisely specified, such as having well-defined goals, clear modeling tasks and measurable performance indicators. *Data Engineering* and *Data Exploration* are often not precisely specified and are quite iterative in nature, while *Model Building* and *Exploitation* are often defined more narrowly and precisely. In classical goal-oriented projects, the process often consists of activities in the following order: Data exploration, data engineering, model building, and exploitation. In practice, however, these trajectories can be much more diverse and exploratory, with practitioners navigating through activities in these quadrants in different orders and in an iterative fashion (for example, Martínez-Plumed et al.[31]).

From the layout of Figure 1 we see, for example, that *Model Building* is where we might expect automation to have the most direct impact—which is indeed the case with the success of automated machine learning (AutoML). However, much of this impact has occurred for modeling approaches based on supervised learning, and automation is still far less developed for other kinds of learning or modeling tasks.

Continuing our discussion of Figure 1, *Data Engineering* tasks are estimated to often take 80% of the human effort in a typical data analysis project.[7] Consequently, it is natural to expect that automation could play a major role in reducing this human effort. However, efforts to automate *Data Engineering* tasks have had less success to date compared to efforts in automating *Model Building*.

Data Exploration involves identifying relevant questions given a dataset, interpreting the structure of the data, understanding the constraints provided by the domain as well as the data analyst's background and intentions, and identifying issues related to data ethics, privacy, and fairness. Background knowledge and human judgement are key to success. Consequently, it is not surprising that *Data Exploration* poses the greatest challenges for automation.

Finally, *Exploitation* turns actionable insights and predictions into decisions. As these may have a significant impact, some level of oversight and human involvement is often essential, for example, new AI techniques can bring new opportunities in automating the reporting and explanation of results.[29]

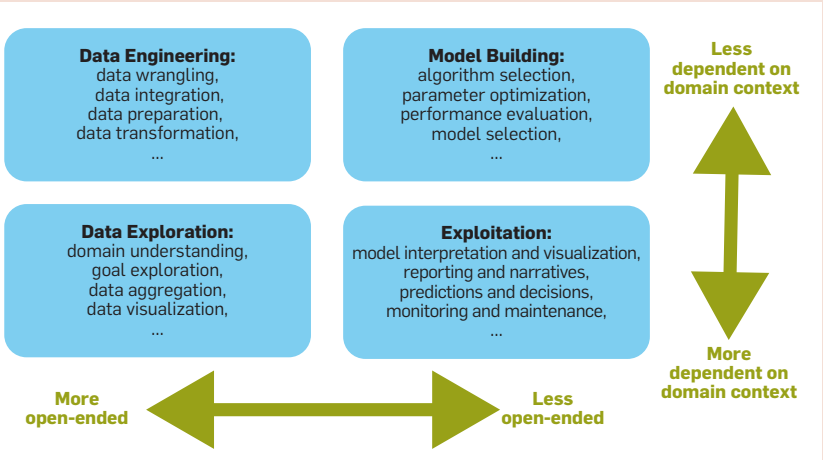
Broadly speaking, automation in the context of data science is challenging depending on the form it takes, ranging in complexity depending on whether it involves a single task or an entire iterative process, or whether partial or complete automation is the goal.

1. A first form of automation—*mechanization*—occurs when a task is so well specified that there is no need for human involvement. Examples of such tasks include running a clustering algorithm or standardizing the values in a table of data. This can be done by functions or modules in low-level languages, or as part of statistical and algorithmic packages that have traditionally been used in data science.

2. A second form of automation—*composition*—deals with strategic sequencing of tasks or integration of different parts of a task. Support for code or workflow reuse is available

Figure 1. The four data science quadrants used in this article to illustrate different areas where automation can take place.

The vertical dimension determines the degree of dependence on domain context, usually introduced through human interaction. The horizontal dimension determines the degree to which a process is open-ended. Some activities, such as data augmentation and feature engineering, are situated in data engineering near the boundary with data exploration.



in more sophisticated tools that have emerged in recent years, from interactive workflow-oriented suites (such as KNIME, RapidMiner, IBM Modeler, SAS Enterprise Miner, Weka Knowledge Flows and Clowdflows) to high-level programming languages and environments commonly used for data analysis and model building (such as R, Python, Stan, BUGS, TensorFlow, and PyTorch).

3. Finally, a third form of automation—*assistance*—derives from the production of elements such as visualizations, patterns, explanations, among others, that are specifically targeted at supporting human efficiency. This includes a constant monitoring of what humans are doing during the data science process, so that an automated assistant can identify inappropriate choices, make recommendations, and so on. While some limited form of assistance is already provided in interactive suites such as KNIME and RapidMiner, the challenge is to extend this assistance to the entire data science process.

Here, we organize our discussion into sections corresponding to the four quadrants from Figure 1, highlighting the three forms of automation where relevant. Because the activities are arranged into quadrants rather than stages following a particular order, we begin with *Model Building*, which appears most amenable to automation, and then discuss the other quadrants.

Model Building: The Success Story of AutoML

In the context of building models (Figure 1), machine learning methods feature prominently in the toolbox of the data scientist, particularly because they tend to be formalized in terms of objective functions that directly relate to well-defined task categories.

Machine learning methods have become very prominent over the last two decades, including relatively complex methods, such as deep learning. Automation of these machine learning methods, which has given rise to a research area known as AutoML, is arguably the most successful and visible application to date of automation within the overall data science process (for example, Hutter et al.[22]). It assumes, in many cases, that sufficient amounts of high-quality data are available; satisfying this assumption typically poses challenges, which we address in later sections of this article (see Ratner et al.[34]).

While there are different categories of machine learning problems and methods, including supervised, unsupervised, semi-supervised and reinforcement learning, the definition of the target function and its optimization is most straightforward for supervised learning (as discussed in "From Machine Learning to Automated Machine Learning"). Focusing on supervised learning, there are many methods for accomplishing this task, often with multiple hyperparameters, whose values can have substantial impact on the prediction accuracy of a given model.

Faced with the choice from a large set of machine learning algorithms and an even larger space of hyperparameter settings, even seasoned experts often must resort to experimentation to determine what works best in each use case. Automated machine learning attempts to automate this process, and thereby not only spares experts the time and effort of extensive, often onerous experimentation, but also enables non-experts to obtain substantially better performance than otherwise possible. AutoML systems often achieve these advantages at rather high computational cost.

It is worth noting that AutoML falls squarely into the first form of automation, *mechanization*, as discussed in the introduction. At the same time, it can be seen as yet another level of abstraction over a series of automation stages. First, there is the well-known use of programming for automation. Second, machine learning automatically generates hypotheses and predictive models, which typically take the form of algorithms (for example, in the case of a decision tree or a neural network); therefore, machine learning methods can be seen as meta-algorithms that automate programming tasks, and hence "automate automation." And third, automated machine learning makes use of algorithms that select and configure machine learning algorithms—that is, of meta-meta-algorithms that can be understood as automating the automation of automation.

From Machine Learning to Automated Machine Learning

The problem of supervised machine learning can be formalized as finding a function f that maps possible input instances from a given set X to possible target values from a set Y such that a loss function is minimized on a given set of examples, that is, as determining $\arg\min_{f \in F} L(f, E)$, where F, referred to as hypothesis space, is a set of functions from X to Y, L is the loss function, and E is the set of examples (or training data), comprised of input instances and target values.

When Y is a set of discrete values, this problem is called (*supervised*) *classification*; when it is the set of real numbers, it is known as *(supervised) regression*. Popular loss functions include cross-entropy for classification and mean squared error for regression.

In this formulation, different hypothesis spaces F can be chosen for a given supervised machine learning task. In addition to the parameters of a given model (such as the connection weights in a neural network) that determine a specific $f \in F$, there are typically further parameters that define the function space F (such as the structure of a neural network) or affect the performance of the model induction process (such as learning rates). Generally, these *hyperparameters* can be of different types (such as real numbers, integers or categorical) and may be subject to complex dependencies (such as certain hyperparameters only being active when others take certain values). Because the performance of modern machine learning techniques critically depends on hyperparameter settings, there is a growing need for hyperparameter optimization techniques.

At the same time, because of the complex dependencies between hyperparameters, sophisticated methods are needed for this optimization task. Human experts not only face the problem of determining performance-optimizing hyperparameter settings, but the choice of the class of machine learning models to be used in the first place, and the algorithm used to train these. In *automated machine learning* (AutoML) all these tasks, often along with feature selection, ensembling and other operations closely related to model induction, are fully automated, such that performance is optimized for a given use case, for example, in terms of the prediction accuracy achieved based on given training data.

AutoML systems have been gradually automating more of these tasks: model selection, hyperparameter optimization and feature selection. Many of these systems also deal with automatically selecting learning algorithms based on properties (so-called metafeatures) of given datasets, building on the related area of meta-learning.[4] In general, AutoML systems are based on sophisticated algorithm configuration methods, such as SMAC (sequential model-based algorithm configuration),[21] learning to rank and Monte-Carlo Tree Search.[33]

So far, most work on AutoML has been focused on supervised learning. Auto-WEKA,[41] one of the first AutoML systems, builds on the well-known Weka machine learning environment. It encompasses all the classification approaches implemented in Weka's standard distribution, including many base classifiers, feature selection techniques, meta-methods that can build on any of the base classifiers, and methods for constructing ensembles. Auto-WEKA 2[25] additionally deals with regression procedures and permits the optimization of any of the performance metrics supported by Weka through deep integration with the Weka environment. The complex optimization process at the heart of Auto WEKA is carried out by SMAC. Auto-sklearn[12] makes use of the Python-based machine learning toolkit scikit-learn and is also powered by SMAC. Unlike Auto-WEKA, Auto-sklearn first determines multiple base learning procedures, which are then greedily combined into an ensemble.

These AutoML methods are now making their way into large-scale commercial applications enabling, for example, non-experts to build relatively complex supervised learning models more easily. Recent work on AutoML includes neural architecture search (NAS), which automates key aspects of the design of neural network architectures, particularly (but not exclusively) in deep learning (for example, Liu et al.[28]). Google Cloud's proprietary AutoML tool, launched in early 2018, falls into this important, but restricted class of AutoML approaches. Similarly, Amazon SageMaker, a commercial service launched in late 2017, provides some AutoML functionality and covers a broad range of machine learning models and algorithms.

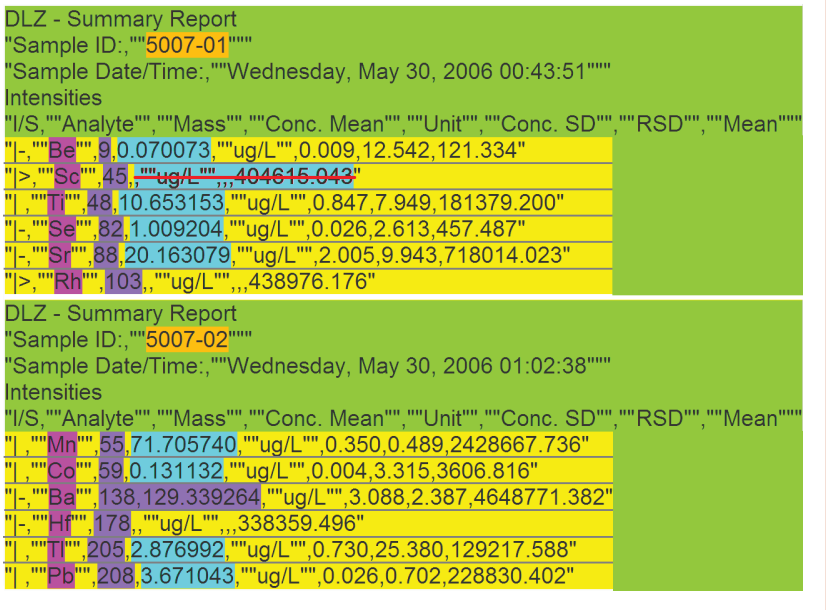
The impressive performance levels reached by AutoML systems are evident in the results from recent competitions.[17] Notably, Auto-sklearn significantly outperformed human experts in the human track of the 2015/2016 ChaLearn AutoML Challenge. Yet, results from the same competition suggest that human experts can achieve significant performance improvements by manually tweaking the classification and regression algorithms obtained from the best AutoML systems. Therefore, there appears to be considerable room for improvement in present AutoML systems for standard supervised learning settings.

Other systems, such as the Automatic Statistician,[29] handle different kinds of learning problems, such as time series, finding not only the best form of the model, but also its parameters. We will revisit this work in the section on *Exploitation*.

The automation of model building tasks in data science has been remarkably successful, especially in supervised learning. We believe the main reason for this lies in the fact that these tasks are usually very precisely specified and have relatively little dependence on the given domain (see also Figure 1), which renders them particularly suitable for *mechanization*. Conversely, tasks beyond standard supervised learning, such as unsupervised learning, have proven to be considerably harder to automate effectively, because the optimization goals are more subjective and domain-dependent, involving trade-offs between accuracy, efficiency, robustness, explainability, fairness, and more. Such machine learning methods, which are often used for feature engineering, domain understanding, data transformation, and so on, thus extend into the remaining three quadrants, where we believe that more progress can be obtained using the other two kinds of automation seen in the introduction: *composition* and *assistance*.

Figure 2. FlashExtract.[27]

After separating attributes by colors, FlashExtract can recognize examples (such as Be, 9 and 0.070073; and Ti, 48 and 10.653153) and counter examples (such as the part struck through in red), in order to induce a program that is able to identify other occurrences of these fields and put them in a spreadsheet or table for further processing.



Data Engineering: Big Gains, Big Challenges

A large portion of the life of a data scientist is spent acquiring, organizing, and preparing data for analysis, tasks we collectively term *data engineering*.[a]

a Data wrangling and data cleansing are terms that are also associated with many of these stages.

The goal of data engineering is to create consolidated data that can be used for further analysis or exploration. This work can be time-consuming and laborious, making it a natural target for automation. However, it faces the challenge of being more open-ended, as per its location in Figure 1.

To illustrate the variety of tasks involved in data engineering, consider the study[2] of how shrub growth in the tundra has been affected by global warming. Growth is measured across a number of traits, such as plant height and leaf area. To carry out this analysis, the authors had to: integrate temperature data from another dataset (using latitude, longitude and date information as keys); standardize the plant names, which were recorded with some variations (including typos); handle problems arising from being unable to integrate the temperature and biological data if key data was missing; and handle anomalies by removing observations of a given taxon that lay more than eight standard deviations from the mean.

In general, there are many stages in the data engineering process, with potential feedback loops between them. These can be divided into three high-level themes, around *data organization*, *data quality* and *data transformation*,[32] as we will discuss in turn. For a somewhat different structuring of the relevant issues, see Heer et al.[19]

Beginning with the first stage, *data organization*, one of the first steps is typically *data parsing*, determining the structure of the data so that it can be imported into a data analysis software environment or package. Another common step is *data integration*, which aims to acquire, consolidate and restructure the data, which may exist in heterogeneous sources (for example, flat files, XML, JSON, relational databases), and in different locations. It may also require the alignment of data at different spatial resolutions or on different timescales. Sometimes the raw data may be available in unstructured or semi-structured form. In this case it is necessary to carry out *information extraction* to put the relevant pieces of information into tabular form. For example, natural language processing can be used for information extraction tasks from text (for example, identifying names of people or places). Ideally, a dataset should be described by a data dictionary or *metadata* repository, which specifies information such as the meaning and type of each attribute in a table. However, this is often missing or out-of-date, and it is necessary to infer such information from the data itself. For the data type of an attribute, this may be at the syntactic level (for example, the attribute is an integer or a calendar date), or at a semantic level (for example, the strings are all countries and can be linked to a knowledge base, such as DBPedia).[6]

The automation of model building tasks in data science has been remarkably successful, especially in supervised learning.

FlashExtract[27] is an example of a tool that provides assistance to the analyst for the information extraction task. It can learn how to extract records from a semi-structured dataset using a few examples; see Figure 2 for an illustration. A second assistive tool is DataDiff,[39] which integrates data that is received in installments, for example, by means of monthly or annual updates. It is not uncommon that the structure of the data may change between installments, for example, an attribute is added if new information is available. The challenge is then to integrate the new data by matching attributes between the different updates. DataDiff uses the idea that the statistical *distribution* of an attribute should remain similar between installments to automate the process of matching.

In the second stage of data engineering, *data quality*, a common task is standardization, involving processes that convert entities that have more than one possible representation into a standard format. These might be phone numbers with formats like "(425)-706-7709" or "416 123 4567," or text, for example, "U.K." and "United Kingdom." In the latter case, standardization would need to make use of ontologies that contain information about abbreviations. *Missing data* entries may be denoted as "NULL" or "N/A," but could also be indicated by other strings, such as "?" or "-99." This gives rise to two problems: the identification of missing values and handling them downstream in the analysis. Similar issues of identification and repair arise if the data is corrupted by anomalies or outliers. Because much can be done by looking at the distribution of the data only, many data sci-

Five Data Exploration Subtasks in Social Network Analysis

Computational social scientists may wish to explore a social network to gain an understanding of the social interactions it describes. For example, an analyst may decide to look for community patterns, formalized as subsets of the nodes and the edges connecting them. In the broad context of data exploration, five subtasks that can potentially be automated are outlined as follows:

1. *Form of the pattern.* Options include the network's high-level topology, degree distribution, clustering coefficient, or the existence of dense subnetworks (communities) as considered here by way of example.

2. *Measuring pattern 'interestingness.'* Interestingness can be quantified as the number of edges or the average node degree within the community, the local modularity, or subjective measures that depend on the analyst's prior knowledge, or measures developed from scratch.

3. *Algorithmic strategy.* Optimizing the chosen measure can require numerical linear algebra, graph theory, heuristic search (for example, beam search), or bespoke approaches.

4. *Pattern presentation.* The most interesting communities can be presented to the analyst as lists of nodes, by marking them on a suitably permuted adjacency matrix, or using other visualizations of the network.

5. *Interaction.* Almost invariably, the analyst will want to iterate on some of the subtasks, for example, to retrieve more communities, or to explore other pattern forms.

ence tools include (semi-)automated algorithms for data imputation and outlier detection, which would fall under the *mechanization* or *assistance* forms of automation.

Finally, under the *data transformation* heading, we consider processes at the interface between data engineering and model building or data exploration. Feature engineering involves the construction of features based on the analyst's knowledge or beliefs. When the data involves sensor readings, images or other low-level information, signal processing and computer vision techniques may be required to determine or create meaningful features that can be used downstream. Data transformation also includes instance selection, for example, for handling imbalanced data or addressing unfairness due to bias.

As well as the individual tasks in data engineering, where we have seen that *assistive* automation can be helpful, there is also the need for the *composition* of tasks. Such a focus on composition is found, for example, in Extraction, Transformation and Load (ETL) systems, which are usually supported by a collection of scripts that combine data scraping, source integration, cleansing and a variety of other transformations on the data.

An example of a more integrated approach to data engineering, which shows aspects of both compositional and assistive automation, is the *predictive interaction* framework.[18] This approach provides interactive recommendations to the analyst about which data engineering operations to apply at a particular stage, in terms of an appropriate domain specific language, ideas that form the basis of the commercial data wrangling software from Trifacta. Another interesting direction is based on a concept known as *data programming*, which exploits domain knowledge by means of programmatic creation and modeling of datasets for supervised machine learning tasks.[34]

Methods from AutoML could potentially also help with data engineering. For instance, Auto-sklearn[12] includes several pre-processing steps in its search space, such as simple missing data imputation and one-hot encoding of categorical features. However, these steps can be seen as small parts of the data quality theme, which can only be addressed once the many issues around data organization and other data quality steps (for example, the *identification* of missing data) have been carried out. These earlier steps are more open ended and thus much less amenable to inclusion in the AutoML search process.

While many activities related to storage, aggregation and data cleaning have been significantly automated by recent database technology, significant challenges remain, since data engineering is often an iterative process over representation and integration steps, involving data from very different sources and in different formats, with feedback loops between the steps that trigger new questions (for example, Heer et al.[19]). For instance, in the Tundra example, one must know that it is important to integrate the biological and temperature data, that the data must already be in a close-enough format for the transformations to apply, and that domain knowledge is needed to fuse variant plant names.

As all these data engineering challenges occupy large amounts of analyst time, there is an incentive to automate them as much as possible, as the gains could be high. However, doing this poorly can have a serious negative impact on the outcome of a data science project. We believe that many aspects of data engineering are unlikely to be fully automated soon, except for a few specific tasks, but that further developments in the direction of both assistive and compositional semi-automation will nonetheless be fruitful.

Data Exploration: More Assistance Than Automation

Continuing our discussion of the quadrants in Figure 1, we next focus on data exploration. The purpose of data exploration is to derive insight or make discoveries from given data (for example, in a genetics domain, *understanding* the relation between particular genes, biological processes, and phenotypes), often to determine a more precise goal for a subsequent analysis (for example, in a retailing domain, discovering that a few variables explain why customers behave differently, suggesting a segmentation over these variables). This key role of human insight in data exploration suggests that the form of automation that prevails in this quadrant is *assistance*, by generating elements that can help humans reach this insight. We will collectively refer to all these elements that ease human insight as *patterns*, capturing particular aspects or parts of the data that are potentially striking, interesting, valuable, or remarkable

for the data analyst or domain expert, and thus worthy of further investigation or exploitation. Patterns can take many forms, from the very simple (for example, merely reporting summary statistics for the data or subsets thereof), to more sophisticated ones (communities in networks or low-dimensional representations).

The origins of contemporary data exploration techniques can be traced back to Tukey and Wilk,[43] who stressed the importance of human involvement in data analysis generally speaking, and particularly in data analysis tasks aiming at 'exposing the unanticipated'—later coined Exploratory Data Analysis (EDA) by Tukey[42] and others.

The goal of EDA was described as hypothesis generation, and was contrasted with confirmatory analysis methods, such as hypothesis testing, which would follow in a second step. Since the early days of EDA in the 1970s, the array of methods for data exploration, the size and complexity of data, and the available memory and computing power have all vastly increased. While this has created unprecedented new potential, it comes at the price of greater complexity, thus creating a need for automation to assist the human analyst in this process.

As an example, the 'Queriosity' system[48] provides a vision of automated data exploration as a dynamic and interactive process, allowing the system to learn to understand the analyst's evolving background and intent, to enable it to proactively show 'interesting' patterns. The FORSIED framework[8] has a similar goal, formalizing the data exploration process as an interactive exchange of information between data and data analyst, accounting for the analyst's prior belief state. These approaches stand in contrast to the more traditional approach to data exploration, where the analyst repeatedly queries the data for specific patterns in a time- and labor-intensive process, in the hope that some of the patterns turn out to be interesting. This vision means that the automation of data exploration requires the identification of what the analyst knows (and does not know) about the domain, so that knowledge and goals, and not only patterns, can be articulated by the system.

To investigate the extent to which automation is possible and desirable, without being exhaustive, it is helpful to identify five important and common subtasks in data exploration, as illustrated for a specific use case (social network analysis) in the associated box. These five problems are discussed in "Five Data Exploration Subtasks in Social Network Analysis."

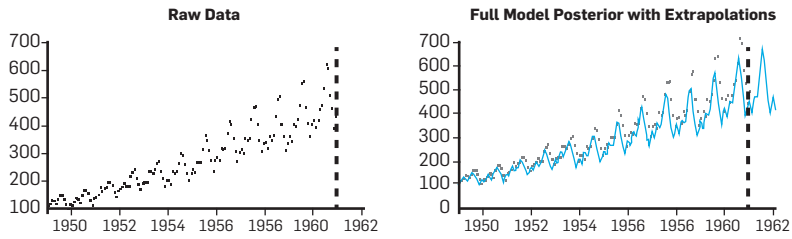
The *form of the patterns* (subtask 1) is often dictated by the data analyst, that is, user involvement is inevitable in choosing this form. Indeed, certain types of patterns may be more intelligible to the data analyst or may correspond to a model of physical reality. As illustrated in the box, a computational social scientist may be interested in finding dense subnetworks in a social network as evidence of a tight social structure.

There are often too many possible patterns. Thus, *a measure to quantify how interesting any given set of patterns of this type is to the data analyst* is required (subtask 2). Here, 'interestingness' could be defined in terms of coverage, novelty, reliability, peculiarity, diversity, surprisingness, utility, or actionability; moreover, each of these criteria can be quantified either objectively (dependent on the data only), subjectively (dependent also on the data analyst), or based on the semantics of the data (thus also dependent on the data domain).[14] Designing this measure well is crucial but also highly non-trivial, making this a prime target for automation. Automating this subtask may require understanding the data analyst's intentions or preferences,[35] the perceived complexity of the patterns, and the data analyst's background knowledge about the data domain—all of which require interaction with the data analyst. The latter is particularly relevant for the formalization of novelty and surprisingness in a subjective manner, and recent years have seen significant progress along this direction using information-theoretic approaches.[8]

The next stage (subtask 3) is to identify the *algorithms needed to optimize the chosen measure*. In principle, it would be attractive to facilitate this task using higher-level automation, as done in AutoML. However, considering the diversity of data across applications, the diversity of pattern types, and the large number of different ways of quantifying how interesting any given pattern is, there is a risk that different data exploration tasks may require different algorithmic approaches for finding the most interesting patterns. Given the challenges in designing such algorithms, we believe that more generic techniques or declarative approaches (such as inductive databases and probabilistic programming, covered in the final section of the paper) may be required to make progress in the *composition* and *assistance* forms of automation for this subtask.

The user interface of a data exploration system often *presents the data*, and *identifies patterns within it*, *in a visual*

Figure 3. A fragment of the Automatic Statistician report for the "airline" dataset, which considers airline passenger volume over the period from 1949 to 1961.[29]



The structure search algorithm has identified four additive components in the data. The first 2 additive components explain 98.5% of the variation in the data as shown by the coefficient of determination (R2) values in accompanying table. Short summaries of the additive components are as follows:

- A linearly increasing function.
- An approximately periodic function with a period of 1.0 years and with approximately linearly increasing amplitude.
- A smooth function.
- Uncorrelated noise with linearly increasing standard deviation.

manner to the analyst (subtask 4). This makes it possible to leverage the strong perceptual abilities of the human visual system, as has been exploited and enhanced by decades of research in the visual analytics community.[23] At the same time, the multiple comparisons problem inherent in visual analysis may necessitate steps to avoid false discoveries.[51] Automating subtask 4 beyond some predefined visualizations (as in the Automatic Statistician, see Figure 3) requires a good understanding of the particular perception and cognition capacities and preferences of each user, a question that also features prominently in the related area of explainable artificial intelligence, which we will discuss.

Such visualizations and other kinds of tools for navigating the data must allow for *rich and intuitive forms of interaction* (subtask 5), to mitigate the open-endedness of typical data exploration tasks. They must allow the analyst to follow leads, verify or refine hypotheses by drilling deeper, and provide feedback to the data exploration system about what is interesting and what is not. A huge challenge for automation is how a novice data analyst could be given hints and recommendations of the type of an expert might use, *assisting* in the process of data navigation, from the combinatorial explosion of ways of looking into the data and possible kinds of patterns. For instance, the SeeDB[45] and Voyager[49] systems interactively recommend visualizations that may be particularly effective, and *Interactive intent modeling*[35] has been proposed to improve information seeking efficiency in information retrieval applications.

Each of the five subtasks is challenging on its own and contains many design choices that may require expert knowledge. We argue that the limitations of current AI techniques in acquiring and dealing with human knowledge in real-world domains are the main reason why automation in this quadrant is typically in the form of *assistance*. Meanwhile, we should recognize that the above subtasks are not independent, as they must combine, through the *composition* form of automation, to effectively assist the data analyst, and non-expert users, in their search for new insights and discoveries.

It is important to raise awareness of the potential pitfalls and side effects of higher levels of automation in data science.

Exploitation: Automation within the Real World

The bottom right quadrant in Figure 1 is usually reached when the insights from other tasks must be translated back to the application domain, often—but not always—in the form of predictions or, more generally, decisions. This quadrant deals with extracted *knowledge* and less with data, involving the understanding of the patterns and models, publishing them as building blocks for new discoveries (for example, in scientific papers or reports), putting them into operation, validating and monitoring their operation, and ultimately revising them. This quadrant is usually less open-ended, so it is no surprise that some specific activities here, such as reporting and maintenance, can be automated to a high degree.

The interpretation of the extracted knowledge is closely related to the area of explainable or interpretable machine learning. Recent surveys cover different ways in which explanations can be made, but do not analyze the degree and form of automation (for example, Guidotti et al.[16]). Clearly, the potential for automation depends strongly on whether a generic explanation of a model (global explanation) or a single prediction (local explanation) is required, and whether the explanation has to be customized for or interact with a given user, by adaptation to their background, expectations, interests and personality. Explanations must go beyond the inspection or transformation of models and predictions, and should include the relevant variables for these predictions, the distribution of errors and the kind of data for which it is reliable, the vulnerabilities of a model, how unfair it is, and so on. A prominent example following the *mechanization* form of automation is the Automatic Statistician,[b,29] which can produce a textual report on the model produced (for a limited set of problem classes). Figure 3 shows a fragment of such a report, including graphical representations and textual explanations of the most relevant features of the obtained model and its behavior.

We believe that fully understanding the behavior and effect of the models

b https://www.automaticstatistician.com/

and insight produced in earlier stages of the data science pipeline is an integral part of the validation of the entire process, and key to a successful deployment. However, 'internal' evaluation, which is usually coupled with model building or carried out immediately after, is done *in the lab*, trying to maximize some metric on held-out data. In contrast, validation *in the real world* refers to meeting some goals, with which the data, objective functions and other elements of the process may not be perfectly aligned. Consequently, this broad perspective of the 'external' validation poses additional challenges for automation, as domain context plays a more important role (Figure 1). This is especially the case in areas, where optimizing for trade-offs between accuracy and fairness metrics may still end up producing undesirable global effects in the long term, or areas such as safety-critical domains, where experimenting with the actual systems is expensive and potentially dangerous, for example, in medical applications or autonomous driving. A very promising approach to overcome some of these challenges is the use of simulation, where an important part of the application domain is modeled, be it a hospital[11] or a city. The concept of 'digital twins'[40] allows data scientists to deploy their models and insights in a digital copy of the real world, to understand and exploit causal relations, and to anticipate effects and risks, as well as to optimize for the best solutions. Optimization tools that have proven so useful in the AutoML scenario can be used to derive globally optimal decisions that translate from the digital twin to the real world, provided the simulator is an accurate model at the required level of abstraction. The digital twin can also be a source of simulated data for further iterations of the entire data science process.

Deployment becomes more complex as more decisions are made, models are produced and combined, and many users are involved. Accordingly, we contend that automating model *maintenance* and *monitoring* is becoming increasingly relevant. This includes tracing all the dependencies between models, insights and decisions that were generated during training and operation, especially if re-training is needed,[36] resembling software maintenance in several ways. Some aspects of monitoring trained models seem relatively straightforward and automatable, by re-evaluating indicators (metrics of error, fairness, among others) periodically and flagging important deviations, as a clear example of the assistive form of automation, which allows for extensive reuse. Once models are considered unfit or degraded, retraining to some new data that has shifted from the original data seems easily mechanizable (repeating the experiment), but it depends on whether the operating conditions that were used initially still hold after the data shift. Reliable and well understood models can often be reused even in new or changing circumstances, through domain adaptation, transfer learning, lifelong learning, or reframing;[20] this represents a more compositional form of automation.

Data science creates many patterns, models, decisions, and meta-knowledge. The organization and reuse of models and patterns can be automated to some degree via inductive databases, via specialized *databases of models* (for example, machine learning model management[46]), or by means of large-scale experimentation platforms, such as OpenML.[c] In the end, we believe the automation of *knowledge* management and analysis for and from data science activities will be a natural evolution of the automation of *data* management and analysis.

Perspectives and Outlook

The quest for automation, in the broad context of data analysis and scientific discovery, is not new, spanning decades of work in statistics, artificial intelligence (AI), databases, and programming languages. We now visit each of these perspectives in turn, before drawing some final conclusions.

c www.openml.org[44]

Selected research challenges in automating data science, with their associated quadrants and likely forms of automation (mechanization, composition, and assistance).

Quadrant	Challenge	Mechanization	Composition	Assistance
Generic	Enhancing human-AI collaboration, by incorporating domain context for interactively defining and refining the goal of data science activities.		×	×
Generic	Addressing ethical, privacy, and legal issues in the automation of data science.	×	×	×
Model Building	Extending AutoML to tasks beyond supervised learning.	×		
Model Building/Data Engineering	Generating meaningful features, considering domain context and task.	×		×
Data Engineering	Streamlining the ETL (Extract, Transform, Load) process by using pipeline schemas and reusing preprocessing subcomponents.		×	
Data Engineering	Expediting the data cleaning, outlier detection and data imputation processes.	×	×	×
Data Exploration	Supporting the design of interactive data and pattern visualizations.			×
Data Exploration	Developing human-AI collaborative systems for data and pattern exploration.		×	×
Exploitation	Generating collaborative reports and presentations, facilitating the interrogation, validation and explanation of models and results.	×		×
Exploitation	Dealing with concept drift, monitoring the interaction of several data science models, and assessing their effects more globally.	×		×

First, there is a long tradition in AI of attempts to automate the scientific discovery process. Many researchers have tried to understand, model, and support a wide range of scientific processes with AI, including approaches to leverage cognitive models for scientific discovery (such as Kepler's laws).[26] More recent and operational models of scientific discovery include robot scientists,[24] which are robotic systems that design and carry out experiments in order to find models or theories, for example, in the life sciences. While these attempts included experimental design and not only observational data, they were also specialized to particular domains, reducing the challenges of the domain context (the vertical dimension in Figure 1). Many important challenges remain in this area, including the induction or revision of theories or models from very sparse data; the transfer of knowledge between domains (which is known to play an important role in the scientific process); the interplay between the design of methodology, including experiments, and the induction of knowledge from data; and the interaction between scientists and advanced computational methods designed to support them in the scientific discovery process.

Second, there were efforts in the 1980s and 1990s at the interface of statistics and AI to develop software systems that would build models or explore data, often in an interactive manner, using heuristic search or planning based on expert knowledge (for example, Gale[13] and St. Amant et al.[38]). This line of research ran up against the limits of knowledge representation, which proved inadequate to capture the subtleties of the statistical strategies used by expert data analysts. Today, the idea of a 'mechanized' statistical data analyst is still being pursued (see the Automatic Statistician[29]), but with the realization that statistical modeling often relies heavily on human judgement in a manner that is not easy to capture formally, beyond the top right quadrant in Figure 1. It is then the *composition* and *assistance* forms of automation that are still targeted when modular data analytic operations are combined into plans or workflows in current data science platforms, such as KNIME and Weka, or in the form of intelligent data science assistants.[37]

Third, in a database context, the concept of inductive query languages allows a user to query the models and patterns that are held in the data. Patterns and models become "first-class citizens" with the hope of reducing many activities in data science to a querying process, in which the insights obtained from one query led to the next query, until the desired patterns and models have been found. These systems are typically based on extensions of SQL and other relational database languages (for example, Blockeel et al.[3]). Doing data science as querying or programming may help bridge the composition and mechanization forms of automation.

Fourth, in recent years, there has been an increasing attention on probabilistic programming languages, which allow the expression and learning of complex probabilistic models, extended or combined with first-order logic.[9] Probabilistic programming languages have been used inside tools for democratizing data science, such as BayesDB[30] and Tabular,[15] which build probabilistic models on top of tabular databases and spreadsheets. Probabilistic programming can also, for example, propagate uncertainty from an imputation method for missing data into the predictive analysis and incorporate background knowledge into the analysis. This may support a more holistic view of automation by increasing the integration of the four quadrants in Figure 1, which may mutate accordingly.

All four of these approaches have had some success in specific domains or standard situations, but still lack the generality and flexibility needed for broader applications in data science, as the discipline incorporates new methods and techniques at a pace that these systems cannot absorb. More scientific and community developments are needed to bridge the gap between how data scientists conduct their work and the level of automated support that such approaches can provide. The accompanying table presents a series of indicative technical challenges for automating data science.

While AutoML will continue to be a flagship example for automation in data science, we expect most progress in the following years to involve stages and tasks other than modeling. Capturing information about how data scientists work, and how data science projects evolve from conception to deployment and maintenance, will be key for more ambitious tools. Progress in areas of AI such as reinforcement learning can accelerate this.

It is important to raise awareness of the potential pitfalls and side effects of higher levels of automation in data science. These include over-reliance on the results obtained from systems and tools; the introduction of errors that are subtle and difficult to detect; and cognitive bias towards certain types of observations, models and insights facilitated by existing tools. Also, data science tools in the context of human-AI collaboration are seen as displacing the work practice of data scientists, leading to new roles.[47] Similarly, this collaborative view suggests new forms of interaction between data scientists and machines, as these become proactive assistants rather than tools.[1]

With all of this in mind, we cautiously make the following predictions. First, it seems likely that there will continue to be useful and significant advances in the automation of data science in the three most accessible quadrants in Figure 1: data engineering (for example, automation of inference about missing data and of feature construction), model building (for example, automated selection, configuration and tuning beyond the current scope of AutoML), and exploitation (for example, automated techniques for model diagnosis and summarization). Second, for the most challenging quadrant of data exploration, and for tasks in the other quadrants where representation of domain knowledge and goals is needed, we anticipate that progress will require more effort. And third, across the full spectrum of data science activities, we see great potential for the assistance form of automation, through systems that complement human experts, tracking and analyzing workflows, spotting errors, detecting and exposing bias, and providing high-level advice. Overall, we expect an increasing demand for methods and tools that are better integrated with human experience and domain expertise, with an emphasis on complementing and enhancing the work of human experts rather than on full mechanization. C

References

1. Amershi, S. et al. Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conf. on Human Factors in Computing Systems*, 2019, 1–13.
2. Bjorkman, A. et al. Plant functional trait change across a warming tundra biome. *Nature 562*, 7725 (2018), 57.
3. Blockeel, H., Calders, T., Fromont, É., Goethals, B., Prado, A., and Robardet, C. An inductive database system based on virtual mining views. *Data Mining and Knowledge Discovery 24*, 1 (2012), 247–287.
4. Brazdil, P., Carrier, C., Soares, C., and Vilalta, R. *Metalearning: Applications to Data Mining*. Springer Science & Business Media, 2008.
5. Chapman, P. et al. CRISP-DM 1.0 Step-by-step data mining guide, 2000.
6. Chen, J., Jimenez-Ruiz, E., Horrocks, I., and Sutton, C. ColNet: Embedding the semantics of Web tables for column type prediction. In *Proceedings of the 33rd AAAI Conf. on Artificial Intelligence*, 2019.
7. Dasu, T. and Johnson, T. *Exploratory Data Mining and Data Cleaning*. Wiley, 2003.
8. De Bie, T. Subjective interestingness in exploratory data mining. In *Proceedings of the Intern. Symp. Intelligent Data Analysis*. Springer, 2013, 19–31.
9. De Raedt, L., Kersting, K., Natarajan, S., and Poole, D. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning 10*, 2 (2016), 1–189.
10. Donoho, D. 50 years of data science. *J. Computational and Graphical Statistics 26*, 4 (2017), 745–766.
11. Elbattah, M. and Molloy, O. Analytics using machine learning-guided simulations with application to healthcare scenarios. *Analytics and Knowledge Mgmt. Auerbach Publication*s, 2018, 277–324.
12. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., and Hutter, F. Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems 28*, 2015, 2962–2970.
13. Gale, W. Statistical applications of artificial intelligence and knowledge engineering. *The Knowledge Engineering Rev. 2*, 4 (1987), 227–247.
14. Geng, L. and Hamilton, H. Interestingness measures for data mining: A survey. *ACM Computing Surveys 38*, 3 (2006), 9.
15. Gordon, A., Graepel, T., Rolland, N., Russo, C., Borgstrom, J., and Guiver, J. Tabular: A schema driven probabilistic programming language. *ACM SIGPLAN Notices 49*, 1 (2014), 321–334.
16. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. A survey of methods for explaining black box models. *ACM Computing Surveys 51*, 5 (2018). 93.
17. Guyon, I., et al. A brief review of the ChaLearn AutoML Challenge: Any-time any-dataset learning without human intervention. In *Proceedings of the Workshop on Automatic Machine Learning 64* (2016), 21–30. F. Hutter, L. Kotthoff, and J. Vanschoren, Eds.
18. Heer, J., Hellerstein, J., and Kandel, S. Predictive interaction for data transformation. In *Proceedings of Conf. on Innovative Data Systems Research*, 2015.
19. Heer, J., Hellerstein, J., and Kandel, S. Data Wrangling. *Encyclopedia of Big Data Technologies*. S. Sakr and A. Zomaya, Eds. Springer, 2019.
20. Hernández-Orallo, J., et al. Reframing in context: A systematic approach for model reuse in machine learning. *AI Commun. 29*, 5 (2016), 551–566.
21. Hutter, F., Hoos, H., and Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the Intern. Conf. on Learning and Intelligent Optimization*. Springer, 2011, 507–523.
22. Hutter, F., Kotthoff, L., and Vanschoren, J., Eds. *Automated Machine Learning—Methods, Systems, Challenges*. Springer, 2019.
23. Keim, D., Andrienko, G., Fekete, J., Görg, C., Kohlhammer, J., and Melançon, G. Visual analytics: Definition, process, and challenges. *Information Visualization*. Springer, 2008, 154–175.
24. King, R., et al. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature 427*, 6971 (2004, 247.
25. Kotthoff, L., Thornton, C., Hoos, H., Hutter, F., and Leyton-Brown, K. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *J. Machine Learning Research 18*, 1 (2017), 826–830.
26. Langley, P., Simon, H., Bradshaw, G., and Zytkow, J. *Scientific Discovery: Computational Explorations of the Creative Processes*. MIT Press, 1987.
27. Le, V. and Gulwani, S. FlashExtract: A Framework for data extraction by examples. In *Proceedings of the 35th ACM SIGPLAN Conf. on Programming Language Design and Implementation*, 2014, 542–553.
28. Liu, C., et al. Progressive neural architecture search. In *Proceedings of the European Conf. on Computer Vision*, 2018, 19–34.
29. Lloyd, J., Duvenaud, D., Grosse, R., Tenenbaum, J., and Ghahramani, Z. Automatic construction and natural-language description of nonparametric regression models. In *Proceedings of the 28th AAAI Conf. on Artificial Intelligence*, 2014.
30. Mansinghka, V., Tibbetts, R., Baxter, J., Shafto, P., and Eaves, B. BayesDB: A probabilistic programming system for querying the probable implications of data. 2015; *arXiv:1512.05006*.
31. Martínez-Plumed, F. et al. CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Trans. Knowledge and Data Engineering* (2020), 1; doi 10.1109/TKDE.2019.2962680.
32. Nazabal, A., Williams, C., Colavizza, G., Smith, C., and Williams, A. Data engineering for data analytics: A classification of the issues, and case studies. 2020; arXiv:2004.12929.
33. Rakotoarison, H., Schoenauer, M., and Sebag, M. Automated machine learning with Monte-Carlo tree search. In *Proceedings of the 28th Intern. Joint Conf. on Artificial Intelligence 7*, (2019); doi: 10.24963/ijcai.2019/457; https://doi.org/10.24963/ijcai. 2019/457.
34. Ratner, A., De Sa, C., Wu, S., Selsam, D., and Ré, C. Data programming: Creating large training sets, quickly. In *Proceedings of the 30th Intern. Conf. on Neural Information Processing Systems*, 2016, 3574–3582.
35. Ruotsalo, T., Jacucci, G., Myllymäki, P., and Kaski, S. Interactive intent modeling: Information discovery beyond search. *Commun. ACM 58*, 1 (Jan. 2014), 86–92.
36. Sculley, D. et al. Hidden technical debt in machine learning systems. *Advances in Neural Info. Processing Systems 28*, (2015), 2503–2511.
37. Serban, F., Vanschoren, J., Kietz, J., and Bernstein, A. A survey of intelligent assistants for data analysis. *ACM Computing Surveys 45*, 3 (2013), 1–35.
38. St. Amant, R. and Cohen, P. Intelligent support for exploratory data analysis. *J. Computational and Graphical Statistics 7*, 4 (1998), 545–558.
39. Sutton, C., Hobson, T., Geddes, J., and Caruana, R. Data diff: Interpretable, executable summaries of changes in distributions for data wrangling. In *Proceedings of the 24th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 2018.
40. Tao, F. and Qi, Q. Make more digital twins. *Nature 573* (2019), 490–491.
41. Thornton, C., Hutter, F., Hoos, H., and Leyton-Brown, K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining*, 2013, 847–855.
42. Tukey, J. Exploratory Data Analysis. Pearson, 1977.
43. Tukey, J. and Wilk, M. Data analysis and statistics: An expository overview. In *Proceedings of 1966 Fall Joint Computer Conf.* (Nov. 7–10, 1966), 695–709.
44. Vanschoren, J., Van Rijn, J., Bischl, B., and Torgo, L. OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter 15*, 2 (2014), 49–60.
45. Vartak, M., Rahman, S., Madden, S., Parameswaran, A., and Polyzotis, N. SeeDB: Efficient data-driven visualization recommendations to support visual analytics. In *Proceedings of the Intern. Conf. on Very Large Data Bases 8* (2015), 2182.
46. Vartak, M., et al. ModelDB: A system for machine learning model management. In *Proceedings of the ACM Workshop on Human-In-the-Loop Data Analytics*, 2016, 14.
47. Wang, D., et al. Human-AI collaboration in data science: Exploring data scientists' perceptions of automated AI. In *Proceedings of the ACM Conf. on Human-Computer Interaction 3*, 2019, 1–24.
48. Wasay, A., Athanassoulis, M., and Idreos, S. Queriosity: Automated data exploration. In *Proceedings of the 2015 IEEE Intern. Congress on Big Data*, 716–719.
49. Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., and Heer, J. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Visualization and Computer Graphics 22*, 1 (2015), 649–658.
50. Wulder, M., Coops, N., Roy, D., White, J., and Hermosilla, T. Land cover 2.0. *Intern. J. Remote Sensing 3*9, 12 (2018), 4254–4284.
51. Zgraggen, E., Zhao, Z., Zeleznik, R., and Kraska, T. Investigating the effect of the multiple comparisons problem in visual analysis. In *Proceedings of the 2018 CHI Conf. on Human Factors in Computing Systems*, 1–12.

Tijl De Bie (tijl.debie@ugent.be) is a professor in the Internet and Data Lab (IDLab) at Ghent University, Belgium.

Luc De Raedt (luc.deraedt@kuleuven.be) is a professor at the Department of Computer Science and Director of the KU Leuven Institute for AI at KU Leuven, Belgium, and Wallenberg Guest Professor at Örebro University, Sweden.

José Hernández-Orallo (jorallo@upv.es) is a professor at the Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Spain.

Holger H. Hoos (hh@liacs.nl) is Professor of Machine Learning at the Leiden Institute of Advanced Computer Science (LIACS) at Leiden University, The Netherlands, and adjunct professor of computer science at the University of British Columbia in Vancouver, Canada.

Padhraic Smyth (smyth@ics.uci.edu) is Chancellor's Professor in the Computer Science and Statistics Departments at the University of California, Irvine, USA.

Christopher K.I. Williams (ckiw@inf.ed.ac.uk) is a professor of Machine Learning in the School of Informatics, University of Edinburgh, U.K, and a Turing Fellow at the Alan Turing Institute, London, U.K.

Acknowledgments

The authors thank the anonymous referees for their comments, which helped to improve the article.

Funding information.

TDB: The European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517. The Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie Vlaanderen" programme. The Fund for Scientific Research–Flanders (FWO–Vlaanderen), project no. G091017N, G0F9816N, 3G042220.

LDR: The research reported in this work was supported by the European Research Council under the European Union's Horizon 2020 research and innovation programme (grant agreement No [694980] SYNTH: Synthesising Inductive Data Models), the EU H2020 ICT48 project "TAILOR" contract #952215; the Flemish Government's "Onderzoeksprogramma Artificiële Intelligentie Vlaanderen" programme and the Wallenberg AI, Autonomous Systems and Software Program funded by the Knut and Alice Wallenberg Foundation.

JHO: EU (FEDER) and the Spanish MINECO, Grant: RTI2018-094403-B-C3. Generalitat Valenciana, Grant: PROMETEO/2019/098. FLI, Grant RFP2-152. MIT-Spain INDITEX Sustainability Seed Fund. Grant: FC200944. EU H2020. Grant: ICT48 project "TAILOR" contract #952215.

HHH: The research reported in this work was partially supported by the EU H2020 ICT48 project "TAILOR" contract #952215; by the EU project H2020-FETFLAG-2018-01, "HumanE AI," contract #820437, and by start-up funding from Leiden University.

PS: This material is based on work supported by the US National Science Foundation under awards DGE-1633631, IIS-1900644, IIS-1927245, DMS-1839336, CNS-1927541, CNS-1730158, DUE-1535300; by the US National Institutes of Health under award 1U01TR001801-01; by NASA award NNX15AQ06A.

CKIW: This work is supported in part by grant EP/N510129/1 from the UK Engineering and Physical Sciences Research Council (EPSRC) to the Alan Turing Institute. He thanks the Artificial Intelligence for Data Analytics team at the Turing Institute for many helpful conversations.

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/automating-data-science