Data Science For All

Jácome Cunha
jacome@di.uminho.pt
University of Minho & INESC Tec/HASLab, Portugal

José Dias,
Paula Pereira
{a78494,a77672}@alunos.uminho.pt
University of Minho, Portugal

João P. Fernandes
jpf@dei.uc.pt
University Coimbra, Portugal

Rui Pereira
ruipereira@di.uminho.pt
INESC Tec/HASLab, Portugal

ABSTRACT

Data is everywhere and in everything we do and in many cases in massive amounts. While on its own data has little value its analysis under the lenses of data science currently supports valuable functions and systems. The problem is that the amount of data generated and the fast-growing need to analyze it is not compatible with the number of workers with the necessary skills. A possible way to mitigate this issue is to propose methodologies and tools that more people, with less programming skills can still use. In this paper we propose our vision to create such methodologies and tools.

1. INTRODUCTION

Data by itself, even if massive, has little value [11, 23, 30]. Indeed, it is the extracted information from data that has the potential to keep changing and improving our lives. However, the extracting process is quite complex and requires several tasks[1] [8], such tasks make up what is called *data science* [5]. These steps are challenging as they require a variety of skills including mathematics, statistics, machine learning, algorithms, correlation or causation [6], with experts with all such skills being hard to come by. In fact, data science related job openings stay unfilled about 10% more time than the market average [12]. Studies have shown that by 2020 the number of positions for data scientists in the USA will be of 2.7 million [12], while also advocating that academia must ensure for data literacy for all in any field of education. In fact, many countries have defined national strategies regarding data science [5]. As widely suggested by companies and governments [5, 9, 12] academia should prepare courses and degrees to capacitate the next generation of data scientists with data

[1]Such tasks include cleaning, transforming, understanding, analyzing and interpreting data

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project UIDB/50014/2020.

science skills. Additionally, researchers and industry should create methodologies and tools for non-programmers or end users to be capable of performing such activities. In this paper we pursue the latter proposal discussing and proposing ways to achieve – **data science for all** (DS4All).

2. WHO ARE THE DATA SCIENTISTS?

A *Data Scientist*'s job is relatively recent, being recognized as such for little over a decade (although many have performed tasks similar to what data scientists nowadays do) [5]. Many working on the topic have a CS background as they have the skills and know the tools (e.g. PLs) to manipulate data. Indeed, in many job offers, employers ask for skills such as SQL, Java, or Unix [12] which are tools common to be known among workers with CS background. However, given the rapid growth of data science job openings [12], many hire workers with different backgrounds [21]. For instance, physicists and other highly quantitative disciplines are being hired for financial quantitative analysts [12].

Given this eclectic scenario we argue that the research community needs a better understanding of the people doing data science. CHI 2019 held a workshop raising this concern [19], with other researchers studying the field in detail [20, 27, 29]. We have also recently presented results of interviews with data science professions in order to further understand their skills, methodologies, difficulties and needs [21]. Such studies are fundamental to understand who are today's data scientists.

3. MINDING THE GAP BETWEEN HUMANS AND TOOLS

Beyond understanding data scientist it is also necessary to know the gap between them (skills and needs) and the existing tools (capabilities and abstraction).

For a CS background worker, the best tool s/he uses may be a PL like Python. Nevertheless, it may be possible to create abstractions to help improve her/is productivity. In fact, many Python programmers use Pandas [18] for data science related tasks, as this library has abstractions to help programmers become more productive. For non-CS workers the gap and need for stronger abstractions may be wider. The industry has proposed tools such as RapidMiner [22] or KNIME [1] allowing users to define data science tasks using control flow visual languages. Tableau [25] goes even further by allowing users to manipulate data through drag-and-drop actions and other intuitive graphical interactions. However, there is little scientific knowledge about the effectiveness of these approaches.

We thus propose to study the gap between data science workers and their currently used tools. It is necessary to create users' profiles characterized by skills and tool knowledge. Depending on their purpose different solutions may arise. General purpose tools, i.e. PLs, require very different skills and abstraction power when compared to tools that are closer to direct manipulation of the data, such as Tableau. We thus need to understand the users skills' and needs, as well as the tools and their requirements, and map these two sets to be able to propose impactful solutions.

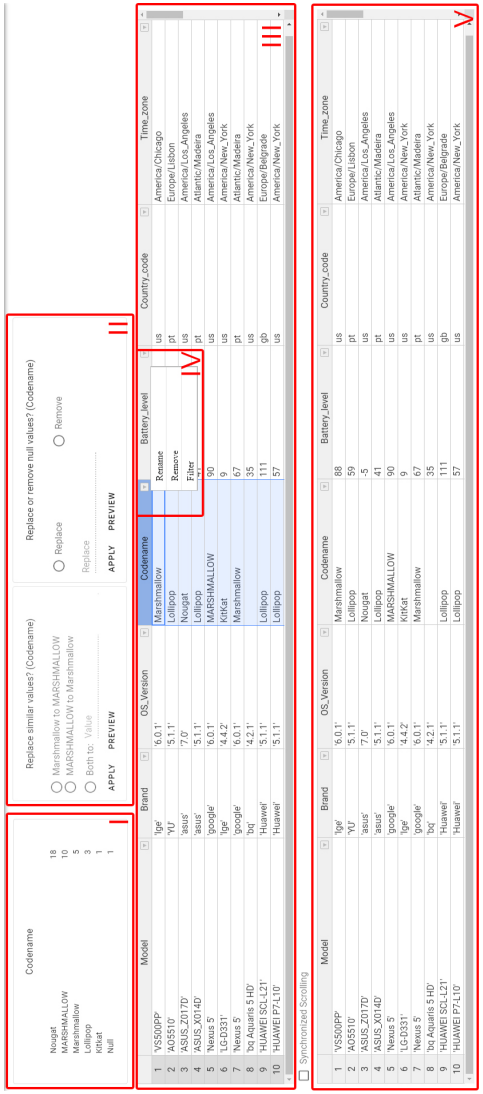

Figure 1: Humanized Data Cleaning DS4ALL Interface

4. TOWARDS HUMANIZED DATA SCIENCE

Data scientists must master several tools which can be challenging. This is even more of an issue for no-CS background workers as having end users developing software is a very well known problem [4, 14].

Computer science is challenging for several reasons and in particular because of its need for abstraction, something paramount for students yet very difficult [16]. In general there does not seem to exist a course specifically on abstraction, but it is mastered by practising it in software development and math courses [16]. In fact, not all computer users are willing to create abstractions as it heavily involves both investment and risk, yet programmers tend to do it more than end users [2].

On the other hand, direct manipulation offers "visibility of the object of interest; rapid, reversible, incremental actions; and replacement of complex command language syntax by direct manipulation of the object of interest" [24]. This makes users feel they master the system under use, ease the learning process, and increases the desire to explore more powerful aspects [24].

Another interesting approach to include in a data science tool for all is to allow users to define their tasks by example. Programming by example has been extensively explored by the research community with very good results [17]. It allows users to give a set of examples of the results one wishes to achieve and have a program synthesized that generalizes the results for any given input.

More generally, all these approaches have the common characteristic of easing the development of software, specially for end users or non-programmers, which is the case of many data science workers.

Based on the following facts: a) programming is difficult in part due to abstraction [2]; b) learning to abstract is difficult [16]; c) direct manipulation allows for mastering a system in use [24]; and d) visual programming, PBE, and live programming intend to ease programming [3, 10, 26]; **we advocate that a visual environment for direct manipulation of data is the best tool one can desire for allowing anyone to perform data science, that is, to achieve a data science for all**.

5. TOWARDS A HUMANIZED DATA SCIENCE TOOL

We have proposed a prototype, shown in Figure 1, for a humanized data cleaning interface [7]. We believe data should be represented in a way users can see and manipulate it using a tabular format. Indeed, presented in Figure 1-V, is the original and unaltered dataset shown at all times, allowing the user to better accompany their transformations. All such transformations are previewed in Figure 1-III. This side-by-side look at the dataset before and after applying changes aims to help remove a level of abstraction of how data will be changed, and directly present such actions. At any point, the user may directly manipulate the data within the *preview dataset*, such as updating cell values, or through a drop down menu (Figure 1-IV) to allow changes or filtering data on a specific column.

When selecting a column, a *statistics card* is displayed to help summarize the contents of the chosen column, as shown in Figure 1-I, where the Codename column is selected and details of the different data

Figure 2: Numerical statistics and suggestion card example

[2]For example when a new member comes to the team

entries are shown. Additionally, a collection of *suggestion cards* are automatically displayed (shown in Figure 1-II), where each presents a data transformation action, based on statistics and data inference. For example, here the system detects two very similar values: `Marshmallow` and `MARSHMALLOW`, and thus suggests replacing one data value by the other or by a new value. Figure 2 shows another example of such cards if choosing the `Battery_level` numerical column.

Such *statistic cards* and *suggestion cards* aim to remove a layer of complexity in data cleaning by automatically presenting common statistical information, which users otherwise have to calculate, and by suggesting transformations. In both cases, the user would have to resort to either programming or using complex tools to gather the statistical information and apply their transformation.

6. COMMUNICATION, DOCUMENTATION, REUSE AND MORE

Data science tends to be an iterative activity [28]. For instance, after a first cleaning phase, one may still find data quality issues during analysis, and thus the need for more cleaning. If the cleaning and analysis phases are done by different teams, then they need to **communicate** with each other. On one hand, the analysis team needs to tell the cleaning team that some issues still need work, while on the other hand, the cleaning team needs to describe what was changed from the original data set. Additionally, the performed operations must be documented if at any point one may need to debug or understand how the data reached the current state. This is specially relevant if someone other who performed such changes wants to understand what happened[2]. Thus, a data scientist needs to additionally document the transformations. In fact, PL Notebooks mix code, execution results and text annotations. However, studies show that such Notebooks have been found to be messy by users [13]. Thus, a tool designed for end users should provide a proper way (e.g. a language, possibly visual) to easily allow the description of data changes to be communicated back and forth between different teams. Studies have also shown that data scientists tend to **reuse** code [15]. However, in some tools this is the common copy&paste [15] which is dangerous as one duplicates code. Thus, good tools need to provide support for proper reuse by allowing users to define some kind of reusable function.

In fact, we argue these issues – communicate, document, and reuse – are quite connected. In fact, they can be seen as different perspectives over the same needs. To communicate between the different teams, documentation is needed. If the documentation of the operations is performed using a language with proper semantics, then these operations can be reused. Thus, we propose that a tool for end user data scientists should provide a language to document such operations, so everyone can understand what was done to the data. The language should have a semantics so it can be used to re-execute the operations. Such a language could be inspired by block-based languages which are being used with quite success among novice programmers. As the operations are being executed in the data, the tool could build a block-based program with the operation. This could be used by everyone to read what happened, specify what should be done, or even re-execute a set of operations.

REFERENCES

[1] KNIME AG. last visited 1/6/2020. KNIME. www.knime.com.

[2] Alan F. Blackwell. 2001. See What You Need: Helping End-users to Build Abstractions. *Journal of Visual Languages & Computing* 12, 5 (2001), 475 – 499. https://doi.org/10.1006/jvlc.2001.0216

[3] Margaret M Burnett. 2001. Visual programming. *Wiley Encyclopedia of Electrical and Electronics Engineering* (2001).

[4] Margaret M. Burnett and Brad A. Myers. 2014. Future of End-User Software Engineering: Beyond the Silos. In *Proceedings of the on Future of Software Engineering* (Hyderabad, India) *(FOSE 2014)*. ACM, New York, NY, USA, 201–211. https://doi.org/10.1145/2593882.2593896

[5] Longbing Cao. 2017. Data Science: A Comprehensive Overview. *ACM Computing Surveys (CSUR)* 50, 3 (June 2017), 42. https://doi.org/10.1145/3076253

[6] Vasant Dhar. 2013. Data Science and Prediction. *Commun. ACM* 56, 12 (Dec. 2013), 64–73. https://doi.org/10.1145/2500499

[7] Jose Dias, Jacome Cunha, and Rui Pereira. 2020. Data Curation: Towards a Tool for All. In *Proceedings of the 22nd HCI International Conference on Human-Computer Interaction* (Compenhagen, Denmark) *(HCII '20)*.

[8] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. 1996. The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Commun. ACM* 39, 11 (Nov. 1996), 27–34. https://doi.org/10.1145/240455.240464

[9] Portuguese Government. 2019. Contrato para a Legislatura com o Ensino Superior para 2020–2023. https://www.portugal.gov.pt/download-ficheiros/ficheiro.aspx?v=d2607a18-51c9-489c-a61c-1ff420dab2f0.

[10] Sumit Gulwani. 2016. Programming by Examples - and its applications in Data Wrangling. *Dependable Software Systems Engineering* 45 (2016), 137–158. https://doi.org/10.3233/978-1-61499-627-9-137

[11] Tim Hoyland, Chris Spafford, and Andrew Medland. 2016. Oliver Wyman's 2016 MRO Survey. https://www.oliverwyman.com/our-expertise/insights/2016/apr/mro-survey-2016.html.

[12] IBM, Business-Higher Education Forum, and Burning Glass. 2017. The Quant Crunch: How the Demand for Data Science Skills Is Disrupting the Job Market. https://www.ibm.com/downloads/cas/3RL3VXGA.

[13] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. 2018. The Story in the Notebook: Exploratory Data Science Using a Literate Programming Tool. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3173748

[14] Andrew J. Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, and et al. 2011. The State of the Art in End-User Software Engineering. *ACM Comput. Surv.* 43, 3, Article 21 (April 2011), 44 pages. https://doi.org/10.1145/1922649.1922658

[15] A. P. Koenzen, N. A. Ernst, and M. D. Storey. 2020. Code Duplication and Reuse in Jupyter Notebooks. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 1–9.

[16] Jeff Kramer. 2007. Is abstraction the key to computing? *Commun. ACM* 50, 4 (2007), 36–42.

[17] Henry Lieberman. 2001. *Your wish is my command: Programming by example*. Morgan Kaufmann.

[18] Wes McKinney. last visited 1/6/2020. Pandas - Python Data Analysis Library. https://pandas.pydata.org.

[19] Michael Muller, Melanie Feinberg, Timothy George, Steven J. Jackson, Bonnie E. John, Mary Beth Kery, and Samir Passi. 2019. Human-Centered Study of Data Science Work Practices. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, Article W15, 8 pages. https://doi.org/10.1145/3290607.3299018

[20] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q. Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How Data Science Workers Work with Data: Discovery, Capture, Curation, Design, Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for

Computing Machinery, New York, NY, USA, Article 126, 15 pages. https://doi.org/10.1145/3290605.3300356

[21] Paula Pereira, Jácome Cunha, and Joao Paulo Fernandes. 2020. On Understanding Data Scientists. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 1–5.

[22] RapidMiner. last visited 1/6/2020. RapidMiner. rapidminer.com.

[23] David Reinsel, John Gantz, and John Rydning. 2018. The Digitization of the World – From Edge to Core. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf.

[24] Ben Shneiderman. 1983. Direct Manipulation: A Step Beyond Programming Languages. *Computer* 16, 8 (August 1983), 57–69.

[25] Tableau Software. last visited 1/6/2020. Tableau Desktop. https://www.tableau.com/products/desktop.

[26] Steven L. Tanimoto. 2013. A Perspective on the Evolution of Live Programming. In *Proceedings of the 1st International Workshop on Live Programming* (San Francisco, California) *(LIVE '13)*. IEEE Press, 31–34.

[27] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 211 (Nov. 2019), 24 pages. https://doi.org/10.1145/3359313

[28] Kanit Wongsuphasawat, Yang Liu, and Jeffrey Heer. 2019. Goals, Process, and Challenges of Exploratory Data Analysis: An Interview Study. *arXiv* 1911.00568 (2019). http://idl.cs.washington.edu/papers/eda-goals-process

[29] Amy X. Zhang, Michael Muller, and Dakuo Wang. 2020. How do Data Science Workers Collaborate? Roles, Workflows, and Tools. arXiv. arXiv:2001.06684 http://arxiv.org/abs/2001.06684

[30] Paul Zikopoulos, Dirk Deroos, Krishnan Parasuraman, Thomas Deutsch, James Giles, and David Corrigan. 2012. *Harness the power of big data The IBM big data platform.* McGraw Hill Professional.