Scalable Interdisciplinary Data Science Teaching at the University of Twente

M. van Keulen C. Seifert M. Poel C.G.M. Groothuis-Oudshoorn

Faculty of EEMCS, University of Twente, Enschede, The Netherlands

{m.vankeulen; c.seifert; m.poel; c.g.m.oudshoorn}@utwente.nl

Abstract

Data scientists are in high demand in many disciplines and domains. This paper describes the data science course open to all master students of the University of Twente. We outline the main challenges of teaching a large and heterogeneous population of non-computer science students about data science and how we addressed them, as well as a historical perspective on how the course grew and evolved.

I. Introduction

Data Science plays an increasingly important role in many disciplines and domains. As a consequence, data scientists are in high demand [1]. Therefore, it is important that besides teaching the next generation *data science specialists*, that we also teach a wide variety non-computer science students *about data science*. The paper describes how the CS department of the University of Twente took the lead in developing a course to teach all master students of the university about data science regardless of their master study. The paper describes the challenges such a large and heterogeneous target population poses and how we addressed them. The main objectives of the data science course are

- Teach basic data science skills and knowledge to a large and heterogeneous population of master students.
- Focus on application skills and methodology [3] as well as continued learning in the work place.

These pose several main challenges

- Heterogeneity in pre-knowledge
- Differences in topical needs
- Scalability

Section II describes our course design and Section III the thematic content. Section IV explains how the challenges were addressed in terms of motivation behind major design choices. Section V describes the growth and evolution over the years.

II. Course design

The course design is outlined in figure 1. The course consists of technical topics and projects. A *topic* is an educational unit of about 1.5 EC focusing on one particular kind of technology in the broad field of Data Science (see Figure 2). Each topic consists of a lecture, study material, such as book chapters or selected papers and assignments designed to learn and practice. Students choose two topics. The assignments for topics are not graded, but need to be signed off as sufficient. Each topic has one or two *topic teachers*, who are responsible for the lecture(s) and maintenance of the topic material.

A *project* is composed of a real-world data

Figure 1: *Course content overview*

DPV: Data Preparation and Visualization
DM: Data Mining
IENLP: Information Extraction and Natural Language Processing
SEMI: Semi-structured data
TS: Feature extraction from Time-Series
PDBDQ: Probabilistic Databases and Data Quality
PM: Process Mining

Figure 2: *Current set of available topics*

set with all its data quality problems and a *challenge*: what problem needs to be answered or what knowledge can potentially be extracted. Each project has a *project owner*: a teacher who formulated the project and who is usually the owner of the dataset. Students choose one project. The final project deliverable is a presentation and a scientific report of the project. For the project report, we provide a template for both LaTeX and Word based on the ACM proceedings template. We also allow 5 pages maximum, excluding the appendix. This template was introduced after we received some reports of 30+ pages. Limiting the page number helped students to better estimate the required amount of work and focus their project. It also improved project assessment for grading.

Students work in pairs for both the topic assignments as well as the project. They are expected to apply at least one of the topics, but are invited to also apply a second topic and/or deepen their knowledge and skills by self-study. Project challenges are formulated broadly to allow students to define their own focus, contribution and sound methodology towards solving the challenge.

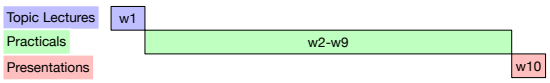

Figure 3: *Study timeline overview*

The course concept is grounded in case-based and project-based learning geared towards self-study in an assignment-driven and project-driven manner. The topics are for learning basic skills, the projects for deepening and assessment. For each topic several assignments are formulated that have to be submitted before the students can start with the project. Those assignments are evaluated by the teaching assistants and should be evaluated as sufficient, i.e. the students need to show that they obtained a general understanding. The two, weekly supervised practical sessions are shared with all topics and projects. The project is assessed by two teachers typically the project owner and a topic teacher according to the assessment criteria of Figure 4. The project grade is the grade for the course. In the end of the course the students need to present their project work for a teacher (topic and/or project owner) and can get feedback before the final submission of the project result.

Criterion	Description
Communication (20%)	Oral and written
Technical depth (40%)	(1) Proper application of technology (2) Depth: going beyond what is taught in the topic(s), combination with other technology or technology of other topic. (3) Understanding of the technology.
Method (40%)	(1) Proper interpretation of results. (2) Interesting insights. (3) Critical attitude towards source data and results. (4) Clear goal, proper goal oriented methodology and priorities/choices with argumentation. (5) Relevance of the steps taken.

Figure 4: *Assessment criteria*

The course outline from a student's perspective is shown in figure 5.

- You work in pairs and are graded in pairs.
- You have to complete 2 topics and 1 project.
- You need to have 2 topic assignments signed off. They will be checked and are mandatory, but will not contribute to the final grade.
- You need to finish 1 project. This will be graded.
- You need to present your project in a presentation at the end of the course.

Figure 5: *Data Science course in a nutshell*

III. Thematic Course Content

In this section we will go into more details about the available topics (cf. figure 2) and projects.

i. Topics

The topics of the course span a broad area of Data Science, covering methods for specific data (e.g., time series, text, event logs), representation techniques (e.g., semi-structure data) and practical issues, such as data quality.

Data Mining (DM): Data mining is about discovering patterns in large data sets involving methods from artificial intelligence, machine learning, statistics, and database systems. The topic teaches supervised methods (classification and regression), and unsupervised methods (clustering).

Data Preparation and Visualization (DPV): The topic teaches (i) data warehousing techniques for extracting and transforming data (ETL, extract-transform-load), (i) modeling data for analytic purposes using the multidimensional modeling approach of OLAP, and (iii) data visualization techniques. This topic comes in two flavours: you can do it in a tool-based fashion as well as in a programming language-based fashion (using the programming language R).

Information Extraction and Natural Language Processing (IENLP): Most information is available in a form rather unsuitable for processing by computers, namely natural language text. This topic teaches (i) text mining (analyzing text directly), (ii) rule-based techniques for information extraction, and (iii) statistical techniques for information extraction and natural language processing. This topic combines well with the topic "Data Mining".

Feature Extraction from Time Series (TS): Sensors and other measurements increasingly produce massive amounts of data with space and time dimensions. The analysis of spatio-temporal data has many applications. The topic focuses on key techniques for preparing time series data for analysis, such as peak detection, filtering, Fourier analysis, dynamic time warping, and prediction models.

Semi-structured Data (SEMI): There exist several data exchange and knowledge representation standards. This topic teaches the most important standards and skills to manipulate data in these standards: (i) XML and its associated standards SQL/XML, XPath and XQuery for publishing and manipulation with both relational as well as XML databases, (ii) JSON storage and manipulation in relational databases, and (iii) Semantic Web standard RDF with its associated standards SPARQL for remote querying, also known as "Linked Open Data".

Probabilistic DataBases and Data Quality (PDBDQ): Much effort in data preparation is devoted to dealing with data quality problems. Probabilistic database technology has the potential of representing data quality problems as uncertainty in the data, and storing and querying it [4]. The topic teaches the most important skills for (i) using probabilistic database technology, and (ii) how to represent several kinds of data quality problems as uncertainty in the data.

Process Mining (PM): Process mining aims to improve understanding and efficiency of business processes by analysing event logs with specialized data-mining algorithms. The topic teaches the most important concepts and skills for applying and understanding Process Mining: (i) Petri nets: the theoretical foundation of process mining, (ii) concepts like event log, causal trace, and the Alpha algorithm, (iii) using the ProM tool for process discovery, (iv) answering analytical questions for a discovered process, and (v) using the ProM tool for process conformance checking.

ii. Example Projects

Similarly to the topics, the projects span a broad spectrum of Data Science. The projects are chosen based on their suitability to the taught topics, the availability of data and their potential appeal to different study programs. Some projects handle anonymised, but still sensitive data. To work with these data sets, students have to sign a non-disclosure agreement before the data set is handed over.

Referral Advice: Low back pain is the most common cause for activity limitation and has a tremendous socioeconomic impact in Western society. In primary care, low back pain is commonly treated by general practitioners and physiotherapists. In the Netherlands, patients can opt to see a physiotherapist without referral from their general practitioner (so called 'self-referral'). Although self-referral has improved the choice of care for patients, this also requires that a patient knows exactly how to select the best next step in care for his or her situation (general practitioner, physiotherapist or self-care), which is not always evident. We would like to automatize the referral advice (no human made decision) and want to know which features are relevant in this referral advice, since this could shorten the questionnaire.

The provided data set is based on a scientific vignette study and contains 1288 fictive patient cases on low back pain that were judged by healthcare professionals on referral advises on a 5-point scale.

Web Harvesting for Smart Applications: There exists technology that can autonomously and robustly harvest data from the web. Such data can be a suitable source for developing smart applications. For example, finding indications of possible unknown side effects of medicines. One could harvest all messages from a web forum for a certain disease, extract information about (a) medicines people report using and (b) which side effects they report having, and compare that with the leaflets of these medicines to determine if some reported side effects are unknown (i.e., not mentioned in the leaflet). There is no given data set for this project, but students are expected to choose a website themselves and harvest data from it.

The challenge is to demonstrate the potential of data harvested from the web for developing smart applications by (i) integrating data from at least two sources (at least one is harvested from the web), and (ii) designing and implementing a proof of concept of a smart web/mobile app.

Automatic detection of Atrial fibrillation episodes: Atrial fibrillation (AF) occurs as a complication postoperatively form cardiac surgery. AF results in stasis of the blood. In the postoperative period AF can induce delirium and neurocognitive decline, thereby prolonging the hospital stay. On the long term serious complications like thromboembolic diseases, stroke and heart failure can be induced by AF. These complications result in increased morbidity and mortality and prolonged hospital stays. Precise ECG monitoring is important to detect AF as soon as possible. Then complications caused by AF can be obviated due to a fast intervention. The challenge of this project it is to develop an algorithm/method that can detect automatically episodes of AF (minimum of 30 seconds) from (preprocessed) ECG data. For the project, a data set with pre-processed ECG readings from a Dutch medical center is provided alongside with its description.

Process discovery and analysis: The data set of the project is from Dutch municipalities. The event data covers building permit applications over a period of approximately four years. The cases in the log contain information on the main application as well as objection procedures in various stages. Furthermore, information is available about the resource that carried out the task and on the cost of the application. The municipalities want to find possible points for improvement on the organizational structure. Moreover, if some of the processes will be outsourced then they should be removed from the process and the applicant needs to

have these activities performed by an external party before submitting the application. The management wants to know will outsourcing affect the organizational structures of municipalities? Moreover, can municipalities learn from each other's processes? The organization would like to streamline their business process. Students are asked to write an advisory report with performance statistics, bottlenecks, etc. and present recommendations for the organization on how to improve and enhance their business process.

IV. Addressing the challenges

There are a few main challenges that drove the course design and organization. In this section, we discuss how we addressed them.

Student population heterogeneity

Initially data science knowledge and skills were taught to a CS population only.[1] One challenge of opening a course to a population of students from all master studies, is how to effectively deal with the different non-CS backgrounds.

An obvious first aspect is that no *programming experience* can be assumed. Initially, we addressed this issue by using tools (Pentaho Data Integration/Kettle, MySQL Workbench, and Weka). It turned out, however, that many students would really like to learn how to program and that they are willing to invest more time to achieve it. Data Science is actually a good opportunity for learning to program, because one only needs limited programming knowledge: mainly control structures, data structures, and using functions from libraries, but no user-interfaces, for example.

Therefore, we provide the students with pointers to tutorials and an indication of which parts are important for data science. Furthermore, we allow the students to make the topic assignments of DPV and DM in a tool-based as well as a programming-based flavour. We make sure there is both tool as well as programming expertise among the teaching assistants (TAs) such that students can benefit from supervision also for their own learning-to-program ambitions. Finally, since the course is given 3 times per year, it is easy for us to allow students to continue the course in the next quarter, so they can pursue their ambitions without risk of failing the course due to lack of time.

A second aspect of the heterogeneity is that we observed a few obstructive attitudes and approaches not usually found among CS students that we explicitly address in our teaching:

- **Small DO-TEST cycles**. We observed that students well-taught in design for their own discipline, had the tendency to design an entire ETL-flow or write large programs before running and testing and then getting into problems with finding and solving bugs. With a small DO-TEST cycle, one knows exactly where to look for the cause of a bug, namely the small addition or change one just made.
- **Error messages** are often largely incomprehensible (e.g., stack traces). Students need to be explicitly told that also CS students and teachers do not comprehend all of it, but that it is beneficial to not dismiss them altogether but to find those few lines that hold a clue.
- **GIYF: Google Is Your Friend**. In some sciences, Google is considered non-academic. Students need to be explicitly told that using Google for finding possible causes and suggestions for solutions is simply effective to get the job done.

Note that it proved important that the teachers themselves came from different backgrounds and were actively involved in the supervision of the topic assignments and projects. We continuously adapt our teaching based on observations of how students with different backgrounds approach things, analyse how they think, and what is difficult for them.

[1] Predecessor courses were part of two master programs that we both consider here as "CS population", namely the master Computer Science and the master Business and IT (`http://www.utwente.nl/en/education/master/programmes`).

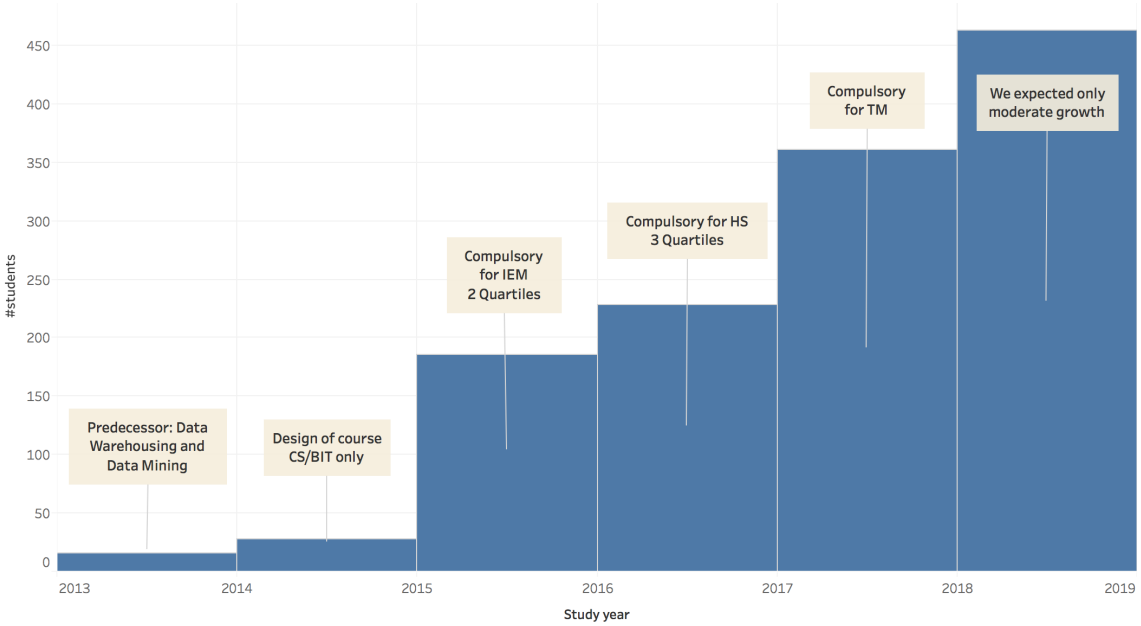

Figure 6: *Growth over the years including main events. The bar between 2013 and 2014 refers to the number of inscribed students in the study year 2013/2014.*

Scalability

The course grew from approx. 50 to 460 students in 5 years (see Section V). An important challenge hence is to scale without significantly affecting quality. The *modular design* with different persons responsible for their own topic(s) or project(s) reduces complexity. Furthermore, the self-guided learning approach reduces much overhead because students manage a lot themselves. These allow coordination to be in the hands of one main teacher who takes care of assigning TAs to supervise sessions, defining clear procedures for the teachers, maintaining sign-offs and grades, communicating with educational management and support, and taking the lead in improvements.

It is important that the coordinator is tech-savvy using IT to streamline coordination. We extensively use Google Apps to automate tasks like scheduling TAs and presentations, collecting assessment results and deriving grade overviews, and monitoring student and assessment progress. We use the university's Canvas learning platform for communication to the students, managing groups, and collecting submissions. And we developed a Latex/SVN-based system for maintaining study materials.

The least scalable aspect of the course is assessment: each project needs assessment by two persons. A fixed template and page limit for the report proved essential. We are actively discussing on how to improve the scalability of the course on this aspect.

Continued learning in work practice

One of the main goals of the course is to provide an effective basis for continued learning. The project hence simulates work practice as much as possible: real-world data and challenges, professional presentation and reporting, and the explicit expectation that one develops oneself continuously. The assessment criteria emphasize this (see Figure 4): going beyond what is learned in the topics is awarded both technically as well as methodically.

V. Historical perspective

The data science course, as many others, was not designed in one go: it evolved over time (see Figures 6, 7, and 8). In this section, we describe its evolution and provide context for the growth. Note that Figure 6 shows numbers of participants, while Figures 7 and 8 show numbers of students *finishing* the course. The reason for this difference is that only for students finishing the course we have information about their study, while for the participant numbers we have two more years of data.

The course originated from a re-design of a course called "Data Warehousing and Data Mining" teaching business intelligence to a CS/BIT student population which attracted in its final year 2013/2014 16 students. Our idea was that an adaptation of ETL, multidimensional modeling [2], and data mining would be a good basis for data science. This gave rise to the topics DPV and DM to which we added more advanced topics on XML databases (XMLDB), Semantic Web (SW), and Natural Language Processing for Information Extraction (IENLP). We started in 2014/2015 with the redesign and initially restricted ourselves to the CS/BIT student population although we admitted also a few very interested students of the master program Industrial Engineering and Management (IEM). The course attracted 28 students in that study year.

For 2015/2016, we opened the course to students of other study programs and added the topic PDBDQ. Moreover, we pro-actively brought the course to the attention of the master program IEM. The management of the study programs was quickly convinced of the importance of data science for their field. The few IEM-students admitted the year before played an important role by acknowledging the added value of the course for their study program as well as the quality of the course. We agreed that DPV and DM would be compulsory for IEM-students and that we would run the course for two quarters to better fit in their curriculum. The course grew to 186 students of which 83 came from IEM. As can be seen in Figure 8, IEM remained the largest study program participating in the course. Also about 25 students of 10 other study programs already found their way to the course as an elective, for example from the master program Human-Media Interaction (7 students; M-ITECH in the figures).

In 2016/2017, we added the TS topic and separated the projects from the topics, because different technologies could be used in addressing the challenges of the 9 real-world projects we defined so far. We focused our attention on the master programs Health Science (HS) and Civil Engineering and Management (CEM), especially their specialisation Transport Engineering and Management. To fit as compulsory into the 1-year master HS, we had to agree to giving the course in 3 quartiles. The course grew to 228 students from 25 different study programs that year.

In 2017/2018, the master Technical Medicine put the course as compulsory in one of their specializations and as optional in the other. The study programs IEM, HS, TM all three had a higher participation in the course than Computer Science (CS) which illustrates that the course really evolved into one that focuses on a non-CS target audience. The course grew to 361 students.

For 2018/2019, we did not pro-actively seek to have the course included in more curricula. We did combine topics XMLDB and SW into one topic SEMI and added the topic PM. The course grew by itself to about 460 students while we expected only a moderate growth. We currently have about 15 projects from a wide variety of domains.

VI. Conclusions

The main challenge of teaching data science to a non-CS student population are heterogeneity in backgrounds, scalability, and achieving an attitude of continued learning. This paper presents our teaching methodology to address these challenges. Moreover it gives insight in how teaching the data science course evolved over time.

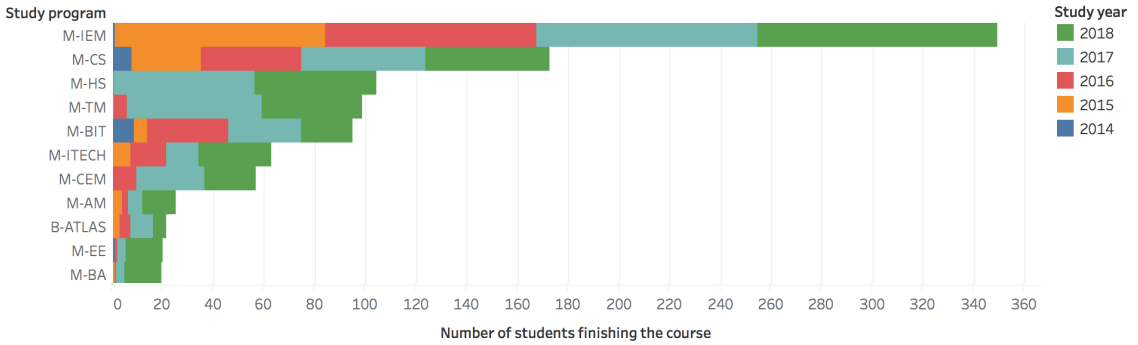

Figure 7: *Overall student participation per study program. Reporting students that finished the course.*

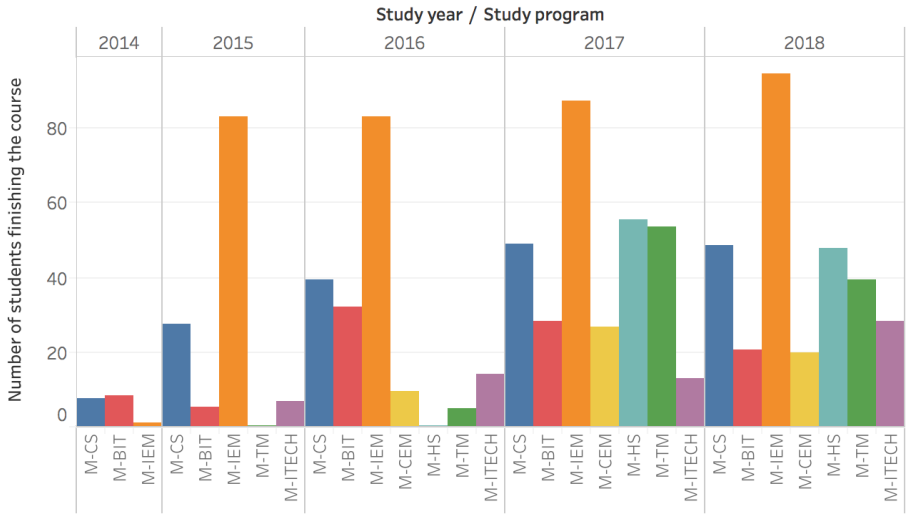

Figure 8: *Evolution of student participation of the study programs with the highest overall participation. Reporting students that finished the course.*

References

[1] Forbes. Data scientist is the best job in America according Glassdoor's 2018 rankings, 2018. `https://www.forbes.com/sites/louiscolumbus/2018/01/29/data-scientist-is-the-best-job-in-america-according-glassdoors-2018-rankings/#5a4d9a765535`, accessed 2019-10-12.

[2] C. Jensen, T. Pedersen, and C. Thomsen. *Multidimensional Databases and Data Warehousing*. Morgan & Claypool, 2010. ISBN 978-1608455379.

[3] KDNuggets. Putting the Science Back in Data Science, 2017. `https://www.kdnuggets.com/2017/09/science-data-science.html`, accessed 2019-10-12.

[4] M. van Keulen. Probabilistic data integration. In S. Sakr and A. Zomaya, editors, *Encyclopedia of Big Data Technologies*. Springer, Feb. 2018.