Ten simple rules for building Master's of Data Science program

Tiffany A. Timbers Michael A. Gelbart

Invalid Date

Abstract

The University of British Columba (UBC) Master of Data Science (MDS) program is a 10-month professional master's program in Data Science. The MDS program was launched in September 2016 and is offered by a collaboration between the UBC Department of Computer Science and Department of Statistics. It involves 24 one-month courses followed by a two-month Capstone Project. It has grown from 22 students with just under 100 applicants at its onset, to 120 students with over 2000 applicants in its most recent application cycle. In this article we document some of the things we think have been key to the success of building this successful program.

Introduction

- describe program and its history (pull from blog)
- document its measurable success
  - student body growth
  - admissions applications growth
  - spin-off programs at UBC (MDS-O, MDS-CL)
- document some testimonials
  - can we get these from the marketing team?

Ten simple rules

Rule 1: Engage (at least) statisticians and computer scientists - Think broadly about the definition of data science & Feed students their vegetables

Think broadly about how you define data science, and feed students a healthy diet of data science dishes, not just dessert. What do we mean with this metaphor? Some students may enter your data science program primarily excited about the latest hot topic, such as deep learning. Others may have broader interests, but still focus on the model-building aspect of data science. Every student brings their own preconceptions about data science. As we know, data science is so much more than training models. In our program we focus on a broad range of topics, based on input from faculty members and industry advisors: topics include ethics, data science communication, data visualization, data cleaning, statistical inference, experimental design, reproducible workflows, collaborative software development, and more.

Our students sometimes complain that they want dessert (e.g. deep learning) before dinner and that they don't want their vegetables (e.g. writing tests and documentation). That said, when our students work with external organizations during our Capstone project, we find our Capstone partners regularly praising the students' code quality, comprehensive documentation, reproducible analysis pipeline in a Docker container, etc. From qualitative results in our alumni surveys, we also find that our alumni appreciate these skills in hindsight as they move forward in their careers. Like vegetables, setting up comprehensive documentation might not be that fun in the moment; it is the type of investment that one can appreciate far into the future, when one revisits a year-old codebase and finds that the project is clearly documented and explained.

NOTE:

the original idea was

1. Engage (at least) statisticians and computer scientists - Think broadly about the definition of data science & Feed students their vegetables

However, I feel like those are two separate things and I had trouble combining them. It would nice to mention stats/cs somewhere though, if possible.

Update: I will try to put this in Rule 3

Rule 2: teach current data science concepts, methods and tools

Teach current data science concepts, methods and tools. Data science has its roots in computer science and statistics, but modern data science can be viewed as its own interdisciplinary field with its own distinct emphasis, applications, workflows, tools, and culture. This means that

building a data science program likely means that new courses will need to be built from scratch. Selecting a subset of pre-existing statistics and computer science courses, to create a data science program, will create one in name only, and will in reality be a combined statistics and computer science degree. Such a program will lack the distinct aspects of data science that have evolved from it's parent fields. In our program we created all of the courses from scratch. This is not to say that we took nothing from courses in the parent fields of data science. A good deal was taken, however, much of the emphasis, workflows and tools needed to be changed. We also needed to add new topics that have arisen from the field of data science, which are absent from the parent fields (e.g., FILL IN ...).

Data science is distinct from it's parent fields is that data science is primarily an applied field, and most students who graduate from a data science program move on to working outside academia. As such, when teaching data science concepts, methods and tools, it is critical to teach them in the context of answering real world questions using rich, real data sets. An additional consequence of students primarily turning into applied, data science practitioners upon graduation is that the curriculum needs to be kept modern and current to help them be successful on the job market. At the time of writing this (and likely for many years hereafter), continual effort is needed to evolve data science educational materials to keep up with the fast evolving pace of the field. In our program, we build this expectation into the culture of our teaching team, as well as budget time and money for this. Culturally, we assign a certain number of courses as targets for redevelopment when it becomes apparent that the field has changed. We also normalize incorporation of new tools methods in our courses, by doing this regularly. When a course needs redevelopment to modernize it with what is happening in the field, we assign the instructor extra teaching credit to provide them the prep time to do this. We also provide instructors funding to attend conferences related to data science, and time to contribute to open source data science projects. Finally, we also listen to our alumni and our academic advisory team (comprised of academic, government and industry representatives) through surveys, and meetings to keep connected with data science practitioners so that we are aware of when the field is moving, and where it is moving too.

Rule 3: create a teaching team that is connected to each other and the entire curriculum & break down the walls between courses

Build a interdisciplinary, cohesive core teaching team. Why interdisciplinary? Data science spans many disciplines, and it's very hard to be an expert in everything. We like to joke that the only people who know everything in the MDS curriculum are the MDS alumni! Therefore, an interdisciplinary teaching team is critical in providing expertise across data science. Our MDS program is an equal partnership between the UBC Departments of Computer Science and Statistics; thus, our teaching team draws heavily from these two fields. However, our core team members come from a much wider array of backgrounds: neuroscience, biomedical engineering, climate science, economics, and more. It is through this interdisciplinary teaching

team that we are able to deliver such a broad curriculum. In addition, it is a joy and privilege to learn from one another within the team.

Why cohesive? The above paragraph speaks to the breadth of data science. When a curriculum is broad, it is particularly susceptible to feeling fragmented. When students learn about feature selection in machine learning, have they already seen that topic in a statistics course, perhaps under a different name? When students log in to remote machines in their cloud computing course, have they already learned best practices of authentication and security in their databases course? The list of connections goes on. To deliver an excellent data science program, instructors need to be aware of what was taught in earlier courses, and furthermore instructors need an awareness of the big picture to make connections where applicable. In our MDS program, we achieve this with a core teaching team of 6-8 faculty and postdocs from whom MDS teaching is our primary work. Our team meets as a group weekly and we stay in close contact regarding all the courses. In addition to cohesion across disciplines, this approach also creates a more cohesive experience for the students in terms of teaching style and course organization.

Rule 4: reflect and iterate (time for redevelopment, academic retreats, not being scared of paperwork, capstone)

Create space and time for reflection on course and program effectiveness, and when weaknesses are identified, act on improving them. Just like a data science project, iteration is critical process for improvement. It is naive to think that a data science course or program would be optimally effective in its first instance.

> "No lesson withstands first contact with learners." – Greg Wilson

Furthermore, given how the field rapidly evolves and changes, iteration might be required even after a course or program has settled into a period of effectiveness. Reflection is critical for identifying when iteration is needed, and then willingness to act is critical to implement the necessary changes. In the UBC MDS program, we have three main avenues for collecting feedback and reflecting on what course and program improvements we might need to make:

1. student-elected student representatives
2. student surveys
3. academic retreats

One way we obtain feedback is through student-elected student representatives. During each course block in MDS, the student body is asked to elect two student block representatives. The role of these student representatives is to facilitate and encourage communication and feedback from students to the teaching team in a more informal and student-driven way. The student representatives are encouraged to bring forward any issues/concerns the student body has with the MDS program or courses to the teaching team if/as they arise through a private

messaging channel on our Slack course discussion forum, and each block they are invited to a meeting with the MDS teaching and operations team (including the program directors) midway through the block. Most student block representatives design and carryout their own surveys ahead of these block meetings to gather feedback from the wider student body to bring to the attention of the teaching team. The feedback received from student block representatives in MDS was extremely useful during the first several years of the program, when we launched 24 newly developed one-credit courses. Through this feedback we quickly learned when courses were presenting material at the wrong pace, when workloads exceeded student capacity given the deadlines, what topics students found most challenging, as well as what pedagogies were most and least effective from the student perspectives.

Finally, we send out surveys to our students upon graduation and at several other time periods post graduation (GET CORRECT TIMES AND INSERT HERE). These surveys provide a different perspective than the feedback from the student-elected block representatives, as these surveys are performed at the end of the program, and while students are (ideally) working in the field of data science. The survey feedback has been most useful in identifying areas where we initially did not place enough emphasis as well as new/arising topics in the field of data science.

Twice a year, MDS holds a half-day academic retreat where the instructors who taught courses over the previous term meet and reflect on what went well with their courses, what could be improved. Time for discussion around what could be improved, and how that cuts across courses to make a more cohesive program is emphasized. Through these retreats we have identified missing or ill-assumed prerequisite knowledge, course ordering issues, pedagogical inconsistencies and more. Course instructors have valuable insight into what is taught and how it is delivered, that is distinct and complementary to that of students. These academic retreats also help build relationships and community within our course instructors. At the end of each retreat, we build an actionable list of action items to be worked on in individual courses, and at the program-level.

Finally, we create time/space in our teaching team's workload to act on the identified needs for course and program redevelopment. Currently, we identify 2-3 courses that are in need of iteration each year, and during our teaching load assignments, give extra time to the instructors assigned to teaching those courses. This means they teach less courses that year. We also make sure to do this well before the academic year starts so that the instructor can find reasonable time to do this important work. It is a cost for the program, in terms of resources, but it pays off in dividends in regards to the benefits it brings for student learning.

Rule 5: use evidence-based pedagogies for learning data science (live coding, flipped classroom, experiential learning)

The field of data science arose in part to help people make descisions based on evidence. Thus, it follows that when teaching data science, evidence should also be used to select effective

pedagogies. In the beginning of our MDS program in 2016, there was not yet a rich literature with evidence on how to teach data science. Instead, we borrowed from the pedagogical literature from related fields, such as statistics and computer science. Many of the pedagogies we have employed fall under the umbrella of active learning. Active learning is almost the opposite of traditional university lectures where students passively watch, listen and sometimes take notes. It is an approach to teaching and learning where students are consistently and frequently engaged with their course studies by doing things (e.g., discussing with classmates, working on solving problems, etc) (Bonwell & Eison, 1991). There is evidence that student performance, at least on summative assessments, can be increased through active learning (Freeman et al. 2014).

Next discuss these common active learning strategies by describing and citing an original paper, and giving a concrete example of how we use them in MDS. - live coding & demonstrations - flipped classroom - in-class worksheets - iClicker - group projects (don't talk about this too much in detail as we have an whole other section on it)

Rule 6: use (and create?) open educational resources

Rule 7: Include meaningful group projects (group + projects = do together)

Include several group projects in your Data Science program. Group work is important, and project work is important; combining them is often particularly effective.

Why group work? This is important because Data Science is an inherently cross-disciplinary role. Data Scientists often must interact with other data scientists, domain experts, data engineers, software engineers, management, and other stakeholders. Including group work in a Data Science program both trains students for collaborative work, and also signals to students that collaboration is valued by the program and necessary for success.

Why projects? Short homework assignments are very effective at reinforcing specific concepts, but they are not representative of real-world Data Science. By "projects" we mean assignments that are both more open-ended, but also longer in duration than traditional homework assignments. For example, our program includes four 4-week projects (completed part-time at the same time as other courses) and one 8-week Capstone project (completed full-time). Projects allow students to practice the critical meta-skills needed by a Data Scientist, such as choosing which method or tool to apply given a problem, or when to consider a project "done".

Why group projects? Although it doesn't have to be this way, in our program all projects are group work and almost all group work is projects. Projects are generally challenging and projects with a larger, more realistic scope can be assigned to groups than individually. Furthermore, providing grades and feedback to projects is very important but very time-intensive – grouping students reduces the total number of projects that need to be assessed by the teaching team.

Additional information on these projects can be found in our blog post, Project courses in MDS.

Rule 8: Scaffold a respectful and supportive community for learners

Put significant effort into creating a respectful and supportive community. Any group of learners has some level of heterogeneity, and establishing a respectful and supportive community for learners is fundamental for levelling the playing field across a group of heterogeneous learners. Learner heterogeneity may be particularly higher in a Master of Data Science program given that students come from a wide variety of academic backgrounds. In a classroom where a respectful and supportive community is not built, only some learners who have certain privileges are likely to succeed. In such an unsafe environment, many learners lacking these certain privileges will become demotivated, frustrated, and feel alone. Their learning will suffer because of this, no matter how well the content is delivered by the teaching team.

One way to scaffold a respectful and supportive community is to use and enforce a code of conduct that clearly outlines what behaviors are not acceptable in our learning spaces, and a process for reporting a code of conduct violation. Importantly, providing contact details for a secondary reporting person is important, in case the person listed as the reporting person (unintentionally) violates the code of conduct. Furthermore, when a code of conduct violation is reported, it is critical that the violation be addressed. Upholding a code of conduct can be challenging, and sometimes requires difficult and uncomfortable conversations. However, without addressing violations, the Code of Conduct merely becomes a facade.

Another way to scaffold a respectful and supportive community in the data science classroom is through careful choice of data sets. Consciously avoiding the use of data sets that may cause negative emotional reactions in learners is creates a space where all learners can focus on the data science concept, tool, workflow being taught in the lesson, without having to suppress or cope with negative emotions. For example, avoid data sets that involve violent and/or sex crimes, body weight, gender coded as a binary variable and eugenics. Yes, these topics can be important to research, however most of the time they are not needed in data science classroom when teaching concepts such as classification, version control, et cetera. If such as data set cannot be avoided for some reason, it is better to call awareness to it's challenges or short-comings and address why they are so.

There is one exception to this suggestion to avoid data sets that cause negative emotional reactions, which is when teaching data science ethics, we often need to use real-world case studies to motivate and engage students in learning about how to carryout out data science in an honest, ethical and fair way. However, even when teaching data science ethics, selection and presentation of these cases need care and effort. Choosing the case that will elicit the minimally needed amount of negative emotions to motivate the topic, as well as providing a rich description of the context of the case, should be the chosen course of action.

Another way is to engage students in formal/professional training on how to work and communicate with others in a respectful manner. Many universities have an equity and inclusion office that runs training courses, and where those do not exist, there are independent organizations that run such courses (e.g., Ally Skills Workshop by Framework Consulting). We recommend offering these courses to students during orientation days/weeks before the beginning of the program, when enthusiasm for the program is high, the culture of your cohort is being formed.

Rule 9: Spend time with the students (contact hours, student-teacher ratios)

Make sure students are given ample opportunity to interact with faculty. One of the main reasons students enroll in a Data Science program is to interact with faculty. While it is resource-intensive to allocate large amounts of "contact hours" between faculty and students, it is critical. In our program, each course has 3 hours of associated lectures per week and 2 hours of associated lab time per week, for a total of 5 hours/week. Our labs are staffed by a faculty instructor and multiple TAs. Furthermore, the cohort of ~100 students is divided into two lab sections, with ~50 students in each. From the student perspective then, each week of class consists of 12 lecture hours, 8 lab hours, 4 optional faculty office hours, and 15-20 optional TA office hours.

Beyond simply the number of hours, be sure to designate part of the time with a label other than "lecture". Although "lecture" time may be used in a variety of ways, including active flipped classroom approaches, the label itself sets expectations for a highly structured, efficient class time. It is important to have times that do not feel rushed, when students can take time to formulate their questions, ask them, seek clarification, consult their peers, and also simply get to know each other and the faculty. To facilitate these types of interactions, if possible seek to schedule your non-lecture time in flat-floor classrooms with tables and chairs. The sloped lecture theatre configuration is not conducive to collaboration or mingling.

Rule 10: Engage your alumni

Leverage your program's alumni community by connecting them with current students. Often, the best person to give advice to a student is someone who was previously in their shoes. In our program, we engage alumni in a variety of ways. All alumni (including faculty/staff alumni) are members of the UBC MDS Alumni Slack workspace, where alumni post jobs ads, ask/answer data science questions, and generally stay in touch. Beyond that, our alumni generously volunteer their time to give talks to current students (e.g. at orientation, or at an employer seminar or Capstone seminar), to participate in career-building events (e.g. conducting mock interviews with current students), to pair up with a current students as part of our mentorship program, or to partner with us for Capstone projects (it's always a delight to have MDS alumni on the other side of the table). These interactions enrich the experience for current students while also benefitting alumni. Current students receive mentorship, wisdom, and employment

opportunities; alumni have an avenue to stay up-to-date on data science techniques, partner on Capstone projects, and hopefully hire some great data scientists. Once your program reaches steady state, the number of current students will stay roughly the same from year to year, but your community of alumni will always be growing. Your alumni hopefully had a great experience with your program, and will often be happy to stay connected. Eventually, the vibrant alumni community, and its associated career opportunities, may even be a draw for prospective students to apply to and select your program over other options.

Conclusion

- TBD

References

Freeman, Scott, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. "Active Learning Increases Student Performance in Science, Engineering, and Mathematics." *Proceedings of the National Academy of Sciences* 111 (23): 8410–15.