VANDERBILT | **M.S. DATA SCIENCE**

Capstone Development (DS-5999) Course Syllabus
Instructors: Jesse Blocher and Dana Zhang

Course Learning Objectives

As the title implies, this course should be the culmination of your Data Science education. In this course, you will develop a practical and professional data science project. You should design and implement this project expecting that it will place prominently, perhaps at the top, of your Data Science portfolio of projects. It should set you apart from other job candidates in its scope, detail, and execution.

Learning Objectives
Specifically, by the end of this course, students should:

1. Have a more comprehensive understanding of how the various skills and concepts you have learned work together and interact with each other.
2. Be able to synthesize the various skills you've learned in your classes into a cohesive whole.
3. Communicate your work, process, and conclusions clearly and concisely in a professional setting.

Grading

Grading is simple, as everything centers around your final project. The final project will have a few parts to it, discussed later.

Intermediate deadlines are pass/fail. If your submission is not acceptable, it will be returned with feedback for you to re-submit until you pass. Later submissions will not be graded until you have passed earlier submissions.

Your final project can also be re-submitted if the A-F grade you get is lower than you desire. Take feedback into account and re-submit for regrading. You can do this up to the final deadline for grade submission before graduation.

Class Structure

This class should feel more like a professional experience than a class. There will not be required multiple weekly classes. Instead, you will have periodic check-in meetings with one of the instructors to ensure you are making progress, discuss obstacles you are facing, and help you overcome them.

Regular Check-Ins. Especially early on, you will have frequent (weekly) check ins with an instructor. Once you overcome early obstacles and are making progress, the frequency can be reduced. Either you or the instructor can initiate this reduction in frequency. We want to be available, guide you, and provide accountability, but not smother you with meetings.

Ad hoc Check-ins. The instructor is available for office-hours-like meetings to discuss problems along the way. If you run into a major obstacle, you are strongly advised to initiate an individual meeting quickly and not wait until the next Regular Check-in. These meetings are optional.
Some scenarios when you may want to initiate a meeting:

- You aren't sure your idea is panning out and may want to change it.
- You are stuck with a technical (coding) problem, either functionality, errors, or speed and have already spent a few hours trying to find a solution on your own
  (you may want to contact an appropriate instructor to help you, not necessarily one of the course instructors if we are not an expert on many of these technologies…)
- You don't know how to interpret results you are getting and/or got unexpected results and don't know what to do with them.
- You have a specific milestone you are supposed to meet, and you no longer think you can meet it for some reason.
- If for any reason you find yourself stuck and not sure what to do now.

The key to individual meetings is knowing when to initiate one. You should not quickly initiate a meeting without first scoping out the problem or challenges and possible solutions. Neither should you spend hours and hours over many days stuck before asking for help. Delays at any stage may lead to you missing deadlines (including the final project deadline). This is good professional practice as well, as this is very similar to escalating issues to a supervisor in a professional setting.

Project Deliverables

Your final project will have four deliverables:

1. **GitHub**. The whole project should be available and explained on GitHub (using Readme.md files for example). It should be reproducible and easily forked and run by someone with some basic experience in the technologies you employ.
   (For example, if you use PySpark, someone who has taken an intro course on PySpark and used it themselves at least once should be able to get it running even if they don't understand all that you did). Be sure to use good coding practice (comments, reproducibility, etc.).
2. **Final Report**. Write up your results targeting a professional audience – i.e., your supervisor, an executive, partner, or very senior researcher who is busy enough that they may only scan your report, but smart enough that they may want easy access to details.

   The genre is business writing, so that means that your first page should be an executive summary that has everything in it if that is all someone reads. Subsequently, use headings and subheadings so the organization of your paper is clear, and someone can skip to where they want to go next.

   Length: 6-10 pages, single spaced, 11pt fonts, 1" margins. Brevity is valued: use up to 10 pages only if necessary. There will not be deductions if these guidelines are slightly violated, so do not stress over getting this precisely right. If you use the defaults on any major word processor, you should be fine. The final report should be submitted in M.S. Word or other word processing format.
3. **Blog Post**. You should turn your executive summary into a blog post, suitable for us to post online. It can/should include the most important tables or plots needed to explain your work to your peers. I can bet a bit more informal in tone. The blog post should be submitted in M.S. Word or other word processing format and will be read only by the instructor. There is no need to actually publish it online (though you are encouraged to do so…). We may contact you later to get permission to use your blog post on our website for promotional purposes.
4. **Oral Presentation.** You should prepare a five-minute talk that explains your project. You can and should use slides. The first slide should be a title slide and the last should clearly be your last

slide saying "Thank You" or "Conclusion," etc. In between you should have no more than 5 slides (one per minute). The content should be largely the same as your Blog Post/Executive Summary. Slides should be submitted in PDF format (so we can combine them together into one big slide deck for quick transitions). Your oral presentation will occur during Final Exam week.

In each of these deliverables, you should discuss:

1. Why your work is important (what is the problem you solve and why does it matter)
2. One or more challenges you faced and how you overcame it (what was your key contribution)
3. The key result of your work (how the problem is solved)
4. Next steps or implications of your findings (think big picture – i.e. another project that could follow yours. This should not be 'fine tuning' or minor additions.)

Assessment

New this year, Capstone is an outcome-based course. Because it is outcome based, *you may get less than a passing grade on your first submission for any of the Exit Criteria below.* This means you should address the deficiencies and resubmit. The paradigm here is more like you are submitting what you think is a finished project to your supervisor for comment. I'll comment and send it back to you until what you submit is well done. This class will be graded A-F, but everyone can get an A in Capstone if you start early enough and resubmit enough. Your final grade will be determined by your grades on three exit criteria:

(Pass/Fail) Phase 1 is to come up with your Project Proposal. This is a written assignment that you should submit in Brightspace.
Exit Criterion: Getting a passing grade on this proposal.
Must be completed by: January 9, 2022 (This means that you must have a passing grade by this date, so submit/resubmit earlier than this.)

Note: You cannot submit Phase 1 until Phase 1 is passed.

(Pass/Fail) Phase 2 is doing the work. This is the longest part of the course.
Exit Criterion: Getting a passing grade on a draft of your final report. To do so, you must be mostly finished with the work of your project and have results in hand. You may wish to re-run something, perform some additional minor analysis, etc. but you are basically done.
Must be completed by: April 15, 2021. This is 3 weeks prior to the end of Exams. If your primary project work (i.e. coding) goes much beyond this, you will be at risk of a poor grade.

Note: You cannot submit Phase 3 until Phase 2 is passed.

(A-F) Phase 3 is communicating your work. Do not underestimate this phase. Brilliant work can be ignored due to poor communication.
Exit Criteria:

1. Present your work in a "lighting round" set of presentations, about 5 min each. This is effectively an oral version of your blog post. This will happen in a small group, and you will both give and receive peer feedback. This will also happen first. *The Deadline for submitting your slides in Brightspace (an indicator that you are ready to schedule your Oral presentation) is May 1, 2022.* Oral presentations can happen early (if a group of students are ready), otherwise they will occur during exam week.
2. Submit a written Final Report of your work. This is described above.

3. Submit a blog post-style write up of your work. This is described above.

You may submit 2 and 3 more than once, and indeed you may not get a passing grade on your first submission so submit early.

Must be completed by: May 4, 2021. This is the middle of Exam week. Any submission after this date is subject to a deduction for being late. Resubmissions may be simply rejected and your grade will be what you received on the previous submission. Failure to submit a final report by the deadline may result in an Incomplete for the class and thus delay your diploma and graduation.

Here is a Rubric to help you understand what is expected of you:

A-level work is defined by the following:

- **Idea/Approach.** This is work that we likely want to highlight on the DSI website and/or use as example projects for future Capstone classes. Your idea is unique (using somewhat standard technical tools) or your technical approach is unique/clever (in solving an existing or common problem) or your data source is unique (you find and make useful a new data source). Your project is at the top of your GitHub and highlighted on your resume.
- **Quality/Depth.** You have been thorough in your solution. This may involve error checking (i.e. when reading in data live for example) and/or robustness (if implementing a clustering/segmentation project) and/or tuning/accuracy (if using machine learning) and/or statistical rigor and experimental design. This also include reproducibility, writing clear and clean code, and commenting (I will spot check your project, I will not read every line of code).
- **Communication/Deliverables.** You can easily articulate your idea for a non-technical audience (try it out on your friends or family). Your Slidedoc is professional and would be an appropriate deliverable in a business setting. There are few, if any, grammar mistakes or language errors. Colors are consistent and used to enhance the document. Plots are excellent and easy to understand. Plots and tables are explained in the text (they never speak for themselves). There is a clear structure and purpose to your communication. You are concise in your message with little repetition or unnecessary points.

B-level work is defined by the following:

Start with the A-level work, but there is a substantial weakness in one of the three areas.

- For example, the idea is creative and communication is solid, but perhaps it shows less depth, rigor, robustness, or thorough tuning/accuracy. Perhaps there is a flaw in your statistical design or something you forgot to control for. Maybe your solution is only slightly more complicated than what you did in previous class or was available online as a tutorial.
- Alternatively, maybe the quality of work is very good and in-depth and looking at an important or interesting problem, but the communication is muddled or unclear. It is hard to follow or not clear what you are doing, perhaps the language/grammar has many errors or confusing.
- Instead, maybe you have tuned, accurate, and in-depth work that is communicated well, but the idea is not all that novel and implementation has been done before (or is not that different from what you'd see in a tutorial). You essentially replicated what someone else did, but you did it a lot more in-depth using similar methods (If you had used a novel, different method to solve the problem robustly, this would be A work).
- Finally, B-level work could also result if all three areas are not quite A-level but pretty good.

C level work is defined by the following:

Start with the A-level work, but two of the three areas have a substantial weakness. Substantial weaknesses are described above in the B-level section for each of the three areas.

Alternatively, it could be that one area has a substantial weakness, and the other two are good, but not

A level.

Failing the class.
You will fail the class if it is clear that you are not taking the project seriously enough. Your failure likely will not come as surprise as we will have discussed the possibility before the end of the semester.

Specifically, if there are very large weaknesses in two or more of the areas above, you will be at risk of failing the class.

- For example, you could have a pretty good idea, but a very thin implementation and poorly communicated results.
- Another scenario would be an idea that is not very original, but your code/approach is pretty good (though perhaps not that original), and then your communication/deliverables are poor. If you have solid depth to your solution, you could move to C level (but poor communication may obscure this so don't count on it...)
- Finally, don't think that you can "put lipstick on a pig" as we say in the U.S. If you have an unoriginal idea and a thin or simplistic approach to the idea, no amount of quality communication will help you pass.

Likely, the biggest temptation in this course is to attempt to take someone else's project, tweak it a little, and then pass it off as your own. If you do this, you are very likely to fail, either because it is so close to an existing project that it is an Honor Code violation, or simply because this represents poor work. If you are inspired by another project, please check with the instructor to be sure your unique contribution is large enough. Do this early enough in the semester that you have time to adjust as needed.

Deadlines and Extensions. It is often possible to get extensions on deadlines, but ask well ahead of the deadline, not just before. ***Asking for an extension within 24hrs of a deadline will be treated the same as being late.*** You should treat the deadlines given as dates given to you by a supervisor for when you should have a project deliverable. Delivering early is fine but being late with no explanation is not. A lack of professionalism in your course conduct will be reflected in your final grade.

Class Technologies

We will use **Brightspace** for assignments and grading, but you will likely not access it very frequently, perhaps only to submit assignments. I do this because Brightspace is FERPA compliant (i.e. I can security communicate your grades to you electronically to maintain your privacy).

I will use **Slack** for all class communication. That includes Announcements to the class (via the class channel) as well as any questions you have (via direct message or on the same channel).

Participation/Attendance

I expect you to attend our meetings unless you are sick or have another unavoidable conflict. A job interview is an acceptable excuse as long as you have made some attempt to avoid the conflict. Please give me as much advance notice as possible if you need to miss a meeting. Treat this class as you would your job. You would never skip a status meeting with your supervisor without a good excuse, clearly communicated in advance.

Technology and Mobile Devices

This is not that related to class, but more of a public service announcement. We know that all of the notifications, badges, and noises our phones and computers make are intentionally trying to draw our attention that specific app. Data scientists are designing these functions to drive "user engagement" with their own app or software, and thus draw you away from your work. You should be very, very intentional about what apps are allowed to interrupt you, and this should be a very short list. Suppress notifications for everything else. You do not need to immediately know about the latest news headlines or a friend's new social media post. Your attention is your scarcest resource! Guard it! To succeed in this program, you will need long stretches of uninterrupted time to work and you should ensure that your technology does not interfere with that. You own your technology, not the other way around.

During Check-In meetings or one-on-one meetings, I expect you to silence your phone or computer notifications and/or ignore them if something does buzz or ring. You should be fully present in the meeting. I commit to do the same for you.

Honor Code

Your final project should be your own work. What does this mean?

Project Idea. Either the idea/problem or the approach to solve/analyze it should be unique. For example, you may find an interesting project idea online, but you come up with a new way to approach it. Alternatively, you may use existing methods to solve a new problem. If you use someone else's project idea and implement code you mostly found online (from one source) to analyze or solve the problem, that is an Honor Code violation.

Code Reuse. You can and should reuse code from old projects, books, the internet (Stack Overflow, Medium, etc.), and even professors or friends/colleagues in the data science program. This is the way the real world works. This code should help you solve an isolated problem in your project, or perhaps help speed up a certain operation with a more efficient implementation. Perhaps you can find or get a module that does a certain task for you.

What you should **not** do is use a large portion of code all from one source and then represent it as your own. For example, you should not download code someone else has used for a project really similar to yours, and then change it (perhaps read in some different input data) and claim it as yours. This is a violation of the honor code.

It may be that a large portion of you code is reused. This is fine if it comes from a variety of places and your primary contribution is to bring it all together to solve your specific problem. You could argue that this is the very essences of a good applied Data Scientist. Be sure to give credit to all the people who wrote that code and helped you.

Citing others – code. If you use a large portion of code from someone else, you should give them credit. You should post this in the landing page Readme.md file on your GitHub page. Something like "I could not have done this project without Jesse Spencer-Smith's webcrawler module (link) that helped me download all the data I needed with minimal changes." Or "I am indebted to Diego Mesa's implementation of TensorFlow (link), which saved me hours of tuning work." If it is someone you do not know, cite them on your page and consider contacting them (or posting in comments or social media) pointing them to your work and highlighting how they helped you.

Code you do not need to cite. Stack Overflow posts that helped you with specific syntax, tutorials that showed you the basics of how to implement a certain class, your own reused code or utility classes/scripts from a previous project. Anything that is small (either in lines of code or in its unique contribution) and/or would not be considered helping you achieve a major milestone in your project timeline.
Exception: If you were stuck on a specific problem for a while and/or someone helped you figure out exactly how to code something really helpful (say, a speed enhancement that made your code 4x faster), then it is good form to cite/thank them, even if it is just one or two lines of code. Here, the citation/thank you is for how much they helped you, not how much work it was for them. If a Stack Overflow post was really helpful, consider upvoting it and/or commenting on it thanking the person.

Citing other ideas. You probably read articles somewhere that inspired your idea. You should do your best to cite them as well, again on your GitHub page. Remember, part of the project is explaining why it matters and is important. You can't do this with computer code.

A lot of this is actually more in the realm of simply thanking and acknowledging others, not really the Honor Code, but the two are related. Remember that Data Science is a team sport, and you should enjoy promoting others' excellent work that helped you. You should begin to develop this as a habit and discipline as it will help your networking. It will connect you to more people and people with better skills who will themselves want to promote your work. It will help solidify existing relationships when colleagues know that you are someone who recognizes others' help.

Violations of the Honor Code will result in a zero on the assignment in question and a referral to the Vanderbilt University Graduate Honor Council. Repeated violations may result in failure for the course and possibly dismissal from the program.

Past Projects – Your capstone project must be a new, fresh project. It cannot be a continuation of an existing project that you have already used for another class nor can it be a project that you will use for another class this semester. The capstone is a unique data science project. If you have concerns about this, or you wish to continue using a data set you are already familiar with, you must discuss the extent of overlap with me before getting started. To be possible, it must be a substantial improvement or extension of the existing project and you must be clear about what was done in the prior class and what you plan on doing in the Capstone class.

Textbooks

None. You should use your texts, readings, and resources from previous classes.