Problem analysis

Foreword

The first step I always take before actually sitting down and start coding is to think about the problem. Many people are conditioned just to solve the problem as it is given, without realizing a way more obvious and better solution. I can give you an example who is one of my friend's personal experiences: she was working at a company which manages warehouses. One of the data science jobs they offered some interns to do is the following: their conveyor belts sometimes get jammed and the management will need to send people to unjam it whenever it is discovered. Some people noticed from CCTV footage, that there will be some patterns of the parcel forming before the conveyor belts get jammed. Her task was to develop a vision algorithm to identify those patterns, and then send a message to the management so they could go unjam the conveyor belts. This is a very well-defined data science task, and one just has to tag some images, train a convolutional neural network, and then write a bot to send the management the message whenever those patterns are detected. They hired some interns to do it, took some money, and took some months, then it worked. Is this not a great victory for data science again? A data science intern these days is not exactly cheap, so developing such a solution did cost the company a non-trivial amount of money. If you were running the company, how would you solve it? Instead of spending some months developing a vision algorithm, I would just either read out the conveyor belt motor information or buy a gauge to measure that, and whenever it is 0, I send a message to the management. Probably cost me less than 100 dollars instead of 100k salary for the intern.

The moral of the story is to just think before you commit to a big plan. **What are you trying to achieve in this project?** Right here is the most important question you have to ask yourself every time you come back to this project. Is it to sell it to millions? Is it to get a promotion? Is it just for fun? The answer to this question should be dedicated to the answer to the following questions we will go through in this lecture. And every time you have to make a decision or whenever you are lost, ask yourself this question again. The more clear your answer is, the more clear which choice you should go for is, and the rest is detail. The objective of this course is to train you to be an independent data scientist, I want you to be able to do a project on your own as well as work with others. So instead of going through just a list of technical skills, I will also put my focus on making decisions: what programming language should I use? What should my documentation look like? How do I want to manage the code base? Should I rewrite yet another javascript framework? These are examples of questions you will ask yourself throughout this course.

Plan for today's session

By the end of today's session,
1. A github repository for your project.
2. A README.md made by following the guide below.

The logistics of today's session is going to go as the following:
1. First, we are going to brainstorm some potential project ideas, just defining the problem and the objective.
2. The next step is work out the details of some of these project.
3. Then we are going to plan a timeline for these projects.
4. Finally, we have to choose one project to work on.

Problem analysis

Semester project brainstorming

Since this is a one-semester project, it will be helpful to keep the scope of the project relatively small.

There are a couple of milestones I would like you to hit during this course, which are
1. Implmenting the core logic of your project.
2. Design an interface to serve your project.
3. Release the code and the service to the world.

Step 1: Go wild

The first step of finding your potential project is to come up with a list of dream projects. Here are some questions that will help you brainstorm what could be interesting:

1. What are your hobbies? Is there anything that you want related to your hobbies that can benefit from data science?
2. What chores do you dislike? Can it benefit from automation?
3. Ask you friends to rant about something, what could possibly make their life better?
4. Is there a dataset that you think it is cool and always wants to try your hands on it but never find the time?

Come up with at least 5 projects that might interest you, discuss with your peers, and come up with more ideas!

Step 2: Come back to reality

After you have finished brainstorming a list of potential projects, here is a list of questions to help you filter for the most suitable projects:

1. **Define one major objective for your project.** Be as concise as possible.
2. **Is this goal acheivable in 13 weeks timescale, with 1 day a week commitment mainly from you?**
3. **Where are you sourcing the data?** Downloading public dataset? Generating the data yourself? Make sure there is not privacy issue or license issue when you are using public dataset.
4. **What is main computation?** It is not required in this course you build a complicated model, I am not even going to require you to use any machine learning in your model. Make sure you have access to enough compute at every stage of the project, from collecting data to deployment.
5. **What are the features you want your product to have?** Is it to serve a dataset with some interactibility? Does it have to be very accurate?
6. **Are you excited about the project?** Choosing something you are excited about is almost always beneficial.

Examples

Here is a list of sample projects to give you some ideas what is in the scope of this course. Disclaimer: These projects are closely related to my personal interests, which may put them out of the possible projects list because of conflict of interest. As mentioned, the only assignment that is relevant to your grade is the labs, so even if you pick any of the projects listed below, it will not affect your grade at all. But to stay on the safe side, I will not approve any project that is **exactly the same as described below**. Outside of this course, I am truly passionate about these projects. If people are interested in

these projects, we can discuss what we can do together, but notice this is straightly outside the scope of this class.

Example 1: Use computer vision model to extract joint positions from video

Example 2: Using LLM to parse relevant grants information

Example 3: Build a drone that can track athlete well while keeping them in frame

Other than these examples, I have a list of projects related to a couple of people I chatted with in JHU. These will be real research projects, so they may not necessarily be suitable for this course due to the complexity of the project. Nevertheless, if you are really interested in looking into these projects, we can discuss in more detail.

Filling in more specific details

Once we have the brief scope of some projects, let's narrow it down to three potential candidates and work out more specific details, here are a list of bullet points to help you with that:

Data

1. **Data type**: What kind of data are you going to use? Text? Image? Video? Tables?
2. **Storage**: How much data are you going to store? Where are you planning to store them?
3. **Data collection**: How are you planning to collect the data?
4. **Privacy issues**: Will your data collection process violate any privacy issues?

Compute

1. **Compute**: Do you need a lot of computing power? Do you have access to it?
2. **Platform**: Are you going to host the project on Web? Mobile? Desktop?
3. **Latency**: Does you project need low latency response?
4. **Network**: Do you need to communicate with other services? Do you plan to host a server?

Developement cycle

1. **Development tools**: What environment you are going to develop in?
2. **CI/CD**: What kind of continous integration you will want to set up?
3. **Rewrite**: Do you anticipate major rewrite of the code?

User Exerpience

1. **User group**: Who are your target users?
2. **User interface**: What is the user interface going to look like?
3. **User interaction**: Do you want to interact with your users?

Now you should have a pretty good grasp of the specification of each project. While there is some chance you might want to change your project during the semester for whatever reason, let's pick one to start with. If it does not work out as well as you thought, the other two should be in a ready state to be picked up.

Making a timeline

Now we have figured out the scope of the project and what features we want the ideal product to have, let's make a timeline for the project so we can stick to the plan.

Problem analysis

Setting milestones

Jumping on top of mount Everest is straightly impossible to any human being, but with the right training and dedication, considerable amount of people have hiked it. The same goes for your project, any interesting project should sound like impossible if you are only thinking about the end result. The importance of milestones is to breakdown this impossible task and make a pathway which you can eventually hike up to the top of the mountain. As a starter, let's imagine you are walking up a multistory building, the goal is to reach the top floor, and your task now is to figure out what are the floors in between, and how should you get to the next floor. Here are some questions to guide you through this process:

1. Look through your feature lists, which ones are easier to implement, and which ones are more difficult? What are the relationship between features? For example, if I am trying to build a drone that will track me moving around, first I need to assemble the drone before I can flash its software.
2. After you figure out where are the floors, now what are the steps to go from one floor to another? Going back to the drone example, say I now know I need to assemble the drone first, the steps I will need to take are first list all the parts I need, find ways to buy them, them assemble them.

How much time are you willing to commit to this?

Everyone has different capacity and constraints, so it is important to make sure you are honest to yourself how much time you are willing to put into building this project. Is it 2 hrs per day for 7 days a week? Or 2 days a week? Is that going to interfere with your other commitments? Is this time commitment realistic given the scope of the project? The most important part of this question is to be honest and realistic, you don't have to tell me you are going to dedicate 8 days a week and 25 hours a day to work on this project.

Consistency is key

The main reason I am asking you to construct a detail timeline and break it down into acheivable steps is so that you can keep making differential progress, and after a while, just like climbing a mountain or a tall building, you will find yourself in places where you couldn't have possible been if you are trying to make one big jump. And in this process, consistency and habit are you best friends.

Decide which day(s) in the week you are going to work on this, and what time in the day you are going to work on the project. Try to stick with the plan as much as possible. If you find yourself not able to stick with the plan, then that needs to be addressed.

What about set backs?

As much as we all want to stick with the plan, life happens. It is inevitable that something will go wrong and you have to move back the schedule. That's fine and that's why we need a clear timeline. If you cannot achieve what are supposed to do this week, no big deal, just move it to next week. However, it is important to recognize setbacks should not happen every week, or more often than the rate you are making progress.