This is the main document for the Capstone, everything links from here

Capstone instructions

1. Overview

In the capstone, you will no longer act as a student, but as an employee of a renowned consulting firm.

The scenario is the following: you've been hired by a consulting firm, and have just had the first exchange of emails with your new boss.

You already know that you have 3 deliverables:

- A proposal describing the plan of analysis, metrics and modeling approaches
- An API endpoint with your model
- A report with an analysis of the dataset and the model description and outcomes

In order to simulate a real world scenario, the requirements of the problem may be ambiguous. A big part of your job as a data scientist in the real world will be turning business requirements into clear cut data science requirements. In the capstone, the starting point is the emails. You will have to send a clarification email to your client with any follow up questions to fully understand what you need to do.

Once you feel comfortable that you understand what is required of you, you will write a proposal with a detailed plan of the analysis and the modeling approach you'd like to use. Your employer (the instructors) will review the proposal and suggest improvements to your plan. You will then proceed with the analysis, create a model and an API endpoint. We will test the API by sending data to it. Finally, you will write a report with an analysis of the dataset and the description of your model, justifying the approach you have chosen.

2. Timeline

- **6 - 12 May** start analyzing the dataset and ask questions about any ambiguities, start preparing the proposal.
- **13 - 15 May** your employer reviews the questions and prepares answers.
- **19 May** deliver the proposal.
- **20 - 22 May** your employer reviews the proposal and provides feedback .
- **23 May - 16 June** prepare the model and the API, write the report. One week before the delivery date, we will do a trial round for you to test the API.
- **16 June** API and report delivery deadline.
- **17 - 23 June** your employer tests the API with the deployed model by sending data to it and reviews the report.
- **24 June - 7 July** address any feedback from the instructors and add it to the report.
- **7 July** submit corrections to the report and the code of your model.
- **7 - 14 July** final review of the corrected report and the API testing.
- **15 July** graduation.

3. Workflow

1. Carefully read **the client briefing**.
2. Disambiguate any requirements, by sending an email to your client via the provided google form. Ask as many questions as you can.
3. Get very familiar with the training set. Expect to spend quite a few hours experimenting, exploring, and getting to know it. Focus on the analysis needed to answer the business questions.
4. Prepare a proposal using **this template**.
5. Train the model that you will require for your API, and understand its limitations.
6. Produce a report that satisfies your client's requirements, using the suggested structure (will be provided later)**.**
7. Deploy your model, using **these instructions.**
8. Deal with the data as it arrives and ensure that your API is responding successfully. The API should deal also with inappropriate data, it should not crash.

4. Sending an email to your client

In the real world, the requirements are never fully defined at the start of the project. You get some instructions, but it's your responsibility to understand what is under-scoped and what information is missing.

In this part of the capstone, you will understand the instructions carefully, start exploring and analyzing data, and realize where you have questions and where information is missing. You will then compile all of this into a professional "email", and send it to your client. You should consider who you are talking to (what hints have been given about them) and make sure that you use language at the appropriate tech level.
Hint: *Do you think they will know the machine learning jargon, or do you have to break down what you need any further?*

To "send" your email, please use **this form**. Your client will answer most questions on 15 May. Your last date for sending them an email is 12 May. Try to put all your questions into one email.

5. The data

You will receive data in 3 moments:

Receiving the train set:
The first is when you receive these instructions. The email from your client will already link to your training dataset.

Receiving the test set 1:
Later (see timeline), the data will start flowing from the client via HTTP. First, you will only receive the labels (not the target).

After this data has stopped flowing, you will receive the respective targets. At this time you will be able to adjust your model and re-deploy if you feel that it's worth updating. This model update is optional.

Receiving the test set 2:
Finally, the data will restart flowing, and the second test set will arrive via HTTP. You will never receive the true labels for this dataset.

	Training Data	Testing Data	
y	`y_train` This portion of the data will be given to you all at once and is what you will use to write their first report and train their model.	`y_test_1` You will receive this portion of y one day after providing a prediction for the corresponding entry in X	`y_test_2` This portion of y you will never receive the true outcome
X	`X_train`	`X_test_1` For this portion of X you will need to provide predictions the same way as in a kaggle challenge except the observations will arrive via HTTP over the course of a week or more.	`X_test_2`

6. The proposal

Before starting to work on your analysis and API, the client wants to see a project proposal. This is to ensure that the work you will do goes in the right direction. The proposal is basically a roadmap of what you're going to do.
The capstone project has two parts - analyzing the existing data and setting up and deploying a model based on the outcomes of the analysis and the client requirements. You will need to define the objectives and outcomes. You will need to think about what to measure to address the client's concerns and narrow down the metrics you want to use for your model. After setting up the model, you need to show that the model works the way it's expected.
To write the proposal, it is important to understand the data very well. You should spend time analyzing the data before writing the proposal. Make sure to include sufficient and relevant details of your plan. A useful proposal should not be too general. A good proposal results in good feedback from the client that will guide you in future work.
Use this template to write your proposal.

6.1 How to deliver the proposal

1. Write the report
2. Name the document ***Proposal_<your email>***
3. Ensure you've read and complied with the technical specifications described in the section above
4. Export the report as PDF
5. Go to the Portal → Capstone → Proposal and upload it there

7. The report

In real life, your company will probably have a template or an older report that you can follow. We provide this report template (to be updated) to you. You can choose to directly use it or adapt it, but we ask that you respect the following:

- Keep to the sequence and titles as indicated in the template. Note that leaving out a section will be an automatic fail of the capstone project.

- The number of pages (listed in the report structure) is a guideline, not a hard rule, but please don't deviate too much from it. Knowing what to leave out is an important skill. In the annexes, however, feel free to go much more overboard.

- Don't include code in the report. You will deliver the code separately.

- Size 11, Arial or some other normal font.
    - Comic Sans will be the reason for immediate fail.

- Don't forget to provide a table of contents on the first page and make sure it is up to date.

7.1 How to deliver the report

6. Write the report
7. Name the document ***Report_<your email>***
8. Ensure you've read and complied with the technical specifications described in the section above
9. Export the report as PDF
10. Go to the Portal → Capstone → Report and upload it there

7.2 How to deliver your code

TBD

7.3 Success Criteria

The passing criteria is also similar to that in the professional world. We expect you to deliver something that would be acceptable by a client. There isn't a single number we are expecting you to hit, nor is there a grader to tell you if you are right.

That will lead to a bit of subjectivity. In general, if you deliver on all the requirements with an acceptable level of quality you will pass. If you deliver something that would get you a bad performance review, you won't.

8. Hints and advice

This is a capstone. It contains data science, data engineering, and project management. Don't worry if it feels a bit overwhelming at first, take a breath and read everything twice. Make a plan for how you will approach each challenge. Ask questions. This is going to be difficult, but you can do it!

You may find that part of this assignment contains some pretty tricky questions. For instance, you may find that every model you train discriminates against some protected group. You will most likely find it impossible to completely remove this effect. That's how the real world works.

You may also discover that there are trade-offs where diminishing one type of discrimination actually increases another. Or that your model performance would go down on some metrics as you attempt to fix others. You may also find that as you attempt to fix true positive rates, your true negative rates will become unequal. To be clear, there is no perfect solution.

Any solution will be subjective, and we are not expecting you to find the "right one". What we are expecting is that you are able to do your best to deal with this, and then support your decisions in an informed way.

There is of course an objective truth, but here is a hint: make sure that you always answer either true or false, and that you aren't caught off and answer np.nan. Look for edge cases. Be skeptical and don't assume things will just work, and look hard at your predictions on the training set, not just as aggregate numbers.