DSC 90

Winter 2020

Presenting Clearly
&
Take Home

Presenting Clearly

Material adapted from:

Should you make a presentation?

Think about the goal.

Presentations are one way to communicate information, but they are often ineffective.

Alternatives:
- Code
- Slides
- Documentation
- Meeting
- ?

4 Steps to Better Presentations

1. **Start with the problem**. Is this a problem your audience knows about already? If not, you'll have to begin by establishing in clear terms that there *is* a problem.

4 Steps to Better Presentations

1. **Start with the problem**. Is this a problem your audience knows about already? If not, you'll have to begin by establishing in clear terms that there *is* a problem.

2. **Have empathy for your audience** and present them with the information they want in a format and in language they can understand.

4 Steps to Better Presentations

1. **Start with the problem**. Is this a problem your audience knows about already? If not, you'll have to begin by establishing in clear terms that there *is* a problem.

2. **Have empathy for your audience** and present them with the information they want in a format and in language they can understand.

3. **Illustrate your conclusions** with data visualizations, but let your own explanation - not the charts - drive your presentation.

4 Steps to Better Presentations

1. **Start with the problem**. Is this a problem your audience knows about already? If not, you'll have to begin by establishing in clear terms that there *is* a problem.

2. **Have empathy for your audience** and present them with the information they want in a format and in language they can understand.

3. **Illustrate your conclusions** with data visualizations, but let your own explanation - not the charts - drive your presentation.

4. **Keep it simple**, and leave out unnecessary detail in both your explanations and your charts. Don't exceed 10 to 15 minutes for the whole presentation.

Who is your audience?

- **Your team manager**
  - Probably the first line of review for any work you do or show to other stakeholders.
  - Your manager may or may not be technical.
  - But they certainly will be communicating with other teams/stakeholders.
- **Line-of-business (LOB) stakeholders**
  - This could be a product manager, business analyst, or a VP of customer support.
  - Data science is amazing because it enables cross-functional work!
  - Just remain aware of how your insights or recommendations influence other teams' workflows.
- **Data engineers/engineering team**
  - Don't forget the team that's working to deploy champion models!
  - Just because these are more technical stakeholders doesn't necessarily mean they should not have any business context in the information

Details.

Making a targeted presentation

Once you understand your audience, you can begin tailoring an effective and targeted presentation.

- With non-technical stakeholders
    - Avoid highly technical terms (e.g. your hyperparameters for your TensorFlow model)
    - Instead, try to frame the machine learning problem into the same terms in which business decisions are made — marginal cost and benefit.
- With your engineering or devOps team
    - They to know details such as how long the model takes to train and GPU/CPU metrics during training.
- The most important thing to recognize is that this should not be the only time these results are communicated.
- Frequent communication and feedback will help alleviate pressure on the final presentation, increase buy-in for your work, and help ease business stakeholders into technical details.
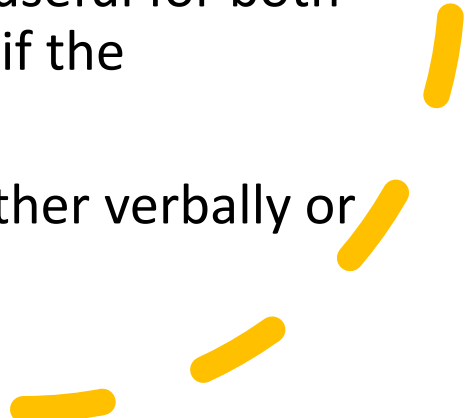
Making a targeted presentation

Here's a useful *starting* framework you can use:

1. Your understanding of the business problem

2. How to measure business impact — what business metrics do your model results align to?

3. What data is/was available — if appropriate, reference what data it would be helpful to collect

4. The initial solution hypothesis

5. The solution/model — use examples and visualizations

6. The business impact of the solution and clear action items for stakeholders

Common Errors

- Omitting/glossing over any key assumptions
- Recycling the same presentation for different audiences
- Showing visualizations like charts and tables without re-iterating the main idea
- Saving all insights until a final presentation (instead of making the process piecemeal and iterative)
- Saving the findings until the end of the presentation — make sure to include an executive summary and recommendations at the beginning of the presentation
- Not having back-up materials of different technical levels — an appendix with supporting details is useful for both the actual presentation itself and context if the presentation is shared
- Not opening up after the presentation (either verbally or via email) for feedback

Data Viz: A whole field!

- **The golden rule: readability comes first**

- It's easy to go down the rabbit hole trying to make your data look pretty, but remember that good design is about *communicating information* first and foremost, and aesthetics are a distant second.

- In a perfect world, your design will be both clear *and* pretty, but if you can only choose one, go for clear every time.

- An ugly but readable chart still conveys information, a beautiful but unreadable chart is useless.

Case Study

a small project that will show the company how well you do data science in practice.

Case Study: common for entry-level positions

- Someone on the data science team will you give a data set, a vague problem to solve with it, and a set time period to do it in.

- This might either be during the on-site interview with an hour or two to work, or a longer period like a weekend so you can do it at home.

- You'll generally be allowed to use the programming languages or tools you are most familiar with, although it's possible the company will limit you to just the tools they use.

- After your time has elapsed, you'll share your results in a short presentation or discussion with a group of people from the data science team.

Case Study: Constraints

- The task should approximate a real task that a qualified candidate should know how to do

- It shouldn't contain "gotchas" and doesn't have to contain a single "right" answer that the candidate must discover in order to advance

- The candidate can use his/her own equipment in a setting of their choosing using their favorite development environment

- Should not require a ton of cleaning or inventing algorithms from scratch.

> The entire point of this exercise to make the work as representative of a "real" work environment as possible

Case Study: Examples

Given data on promotional emails sent by a company and data on orders placed, determine which of the email campaigns did best and how the company should market differently in the future.

Given the text of 20,000 tweets where the company was mentioned, group the tweets into different topics that you think would be useful to the marketing team.

An expensive A/B test was run on the company website, but halfway through the data stopped being consistently collected. Take the experiment data and see if there is any value that can be derived from it.

Case Study: Why Employers Use Them

- **Can you take a vague, open-ended problem and figure out some methods to try to solve it?** It's entirely possible you won't solve the problem, but as long as you make an attempt in a reasonable direction, it shows you have the technical skills and can get things done.

- **Can you work with real-world messy data?** The data you'll be given will likely require filtering, joins, feature engineering, and handling missing elements. By giving you a complex dataset, you'll be doing the kind of work that you would do on the job.

- **Can you structure an analysis?** Will you look at the data in a methodical, well-thought out way, or will you investigate things that don't relate to the task at hand?

- **Can you produce a useful report?** You will have to create a presentation about your work, and possibly documents like Jupyter notebooks or R markdown reports. Can you make something that is useful to the company? Can you structure a useful narrative?

Case Study: Unique Constraints

Limited Time

Use wisely!

Judged on presentation of results

Build it as you go

Very specific audience

Case Study: Steps

1. Read and Plan
   - The first step is to read the directions – not once, but multiple times.
   - Assuming the exercise doesn't have a time limit that starts when accessed, the directions should be read the first time when the exercise is received, a second time before beginning the exercise and a third time before submitting
   - It is also important to start the exercise early and plan multiple work sessions.

2. Choose Your Tools
   - Unless specified in the directions, candidates must choose an appropriate toolkit and/or programming language.
   - Time and skill permitting, it is good practice to choose a tool or language that is used by the employer's team.

3. Keep It Simple
   - Start with a simple but appropriate technique for the problem and then work towards more sophisticated methods.
   - Keeping the analysis basic (at least at the beginning) shows the candidate can think carefully and logically about a problem rather than immediately reaching for the algorithm or method du jour. For some employers, simpler methods are actually more desirable than complex ones, due to their interpretability and ease of use.

4. Organize and Narrate
   - Carefully organize code and annotate it so that a colleague could understand it without much effort. Functions should be documented using a language-appropriate style and ample comments should be provided throughout the code.
   - If a tool like Jupyter notebook is used, make full use of the markdown formatting features. Headings should make it easy to identify key information and answers to exercise questions. Narrative text should explain not only what is happening, but also what was attempted previously, and how the analysis could be further expanded. Finally, demonstrate mastery of the methods utilized by describing their strengths and weaknesses.

Case Study: Presentation

- 20-30 minutes of you presenting your findings
- 10-15 minutes of questions from the audience

Practice your presentation!

During the question and answer section you'll be peppered with questions on all sorts of topics.

- Take a moment to think about the question before answering so that you can get your bearings and think through your response.
- In the event that you don't have an answer you are confident in, it's generally best to give some version of "I am not sure, but…" and give some ideas on how you could potentially find the answer.
- If possible, add the relevant context you do know.
- Be prepared to justify assumptions, choices.