Final Project: Applied Modeling in Action

This final project will give you hands-on experience in building, critiquing, and improving quantitative models for real-world decision-making. You will work in teams of up to four students to explore one of three application contexts:

1. **Designing Custom Spotify Playlists**
2. **Improving Security Checkpoints at USC**
3. **A Student-Defined Project** (with instructor approval)

These contexts are described more fully below.

Each project begins with a **base model**, which you will implement using either a mathematical optimization or simulation approach (or both). However, simply running a model and reporting its solution is insufficient. The goal of this project is to think critically about **what the base model is missing**, improve it, and analyze the implications of your refinements. This process mirrors how businesses and organizations use analytics in real-world decision-making—understanding initial models, identifying their limitations, and refining them to better reflect complex realities.

Project Structure

Every project will have 4 parts.

1. **Implementing a Base Model**
   You will receive initial description and data as a starting point for your model. Your first task is to formulate and implement this base model in Python. You will also answer a few questions to demonstrate your understanding of how the model works and what assumptions it makes.

2. **Critiquing the Base Model**
   No model is perfect. In this section, you will evaluate the limitations of the base model and discuss important aspects of the application it fails to capture. It's easy to nitpick and find things that are missing or not well-captured in a model. A good project will not just identify missing elements, but argue clearly why those missing elements significantly impact the quality of the model's recommendations. A GREAT will consider many missing elements and argue which ones are the MOST important to capture and focus on those most important elements. Both good and great projects will justify their choices quantitatively.

3. **Enriching the Model**
   Based on your critique, you will develop an improved version of the model that addresses some of the shortcomings that you saw as most crucial. Addressing shortcomings is not easy! You may have to refine the constraints or objective or change the model entirely. You may need to go off and get different (or better) data to support

your enriched model. The key is clearly explaining how your changes improve the model **by addressing the shortcomings you identified earlier.**

4. **Making Recommendations and Offering Insights**
After running your enriched model, you will analyze the results and provide **actionable recommendations**. You should discuss any limitations of your enriched model and how they affect the strength of your recommendations. (Of course, if you can't recommend any actions with confidence because you're too unsure, it suggests your model might need more refinement.) You should also assess the robustness of your solution to poor data quality, Great projects will quantitatively assess robustness using tools such as sensitivity analysis and simulation.

Deliverables

Each team will submit:

- A **5 page report** (1.5 spacing, 11pt font) detailing your modeling process, critiques, improvements, and insights. This should include clear explanations, figures, and thoughtful discussion. Mathematical optimization formulations should appear in an appendix (outside the 5-page main body), but described clearly in the main body in a manner accessible to a general business audience. Think of the main body as the short document your manager would read and perhaps share with your team or their boss. Think of the appendices as any information the engineering team actually trying to build a production-quality implementation of your model might need to know to reproduce your results.

  While I do want you to respond to all 4 parts of the prompt, please do not organize your write-up by simply answering the 4 parts of the prompt in 4 sections. Spend the time to organize it in the fashion that makes most sense for YOUR specific findings.

- A **runnable Jupyter Notebook (.ipynb)** implementing your models. (This would be for the engineering team.)
- Any **datasets** used or collected.

A Note on AI Tools, Plagiarism, and Effort

While I fully expect you to use AI tools like CoPilot to help you in coding, or perhaps in brainstorming for your project, I do expect that your write-up will be your own and reflect your analysis. I will run write-ups through an AI plagiarism checker. Similarly, if I find someone on Kaggle has posted a project eerily similar to yours (It's sadly happened before), I will send it to the university as an academic integrity violation. (Don't test me on this. I've had to do it before, and it stinks for everyone, me included.)

In a similar vein, I expect your analysis to be DEEP. If the quality of your analysis is not much better than what ChatGPT can suggest in a few minutes, why should a company hire you? You have over 6 weeks to think about and work on your projects. You will be given 7 hours of in-class time to collaborate with your team, talk to other teams for ideas, and consult with guest faculty and doctoral students on modeling questions. That's 28 people-hours of in-class time exclusively! I do expect you to put in time on this project outside of class since the only deliverable you have for the course after exam 2 is a short speaker write-up.

This is all to say – if you plan on "knocking this out" with your team the week it's due, you're unlikely to get a good grade. You should plan to spend as many hours as you would preparing for a final exam. So if you would have studied hard for the 2 days before an exam (pulling all-nighters), that's something like 36 hours of work PER PERSON that you should probably be putting into the project.

Why This Project Matters

This project is designed to reflect how I imagine you using the tools from this course in your first job or internship. Whether in consulting, operations, finance, or technology, business analysts rarely use models in isolation. Instead, they must be able to:

- Understand and implement quantitative models
- Identify gaps in initial approaches
- Adapt and refine models based on real-world complexity
- Communicate findings effectively to decision-makers

By the end of this project, you should not only have a stronger grasp of the technical skills we've covered in class but also a deeper appreciation for the **judgment, creativity, and critical thinking** required to apply these methods effectively in practice.

Project Option 1: Designing Custom Spotify Playlists

Overview

Music streaming platforms generate personalized playlists to match users' tastes, moods, and activities. In this project, your team will act as data-driven playlist curators, designing custom playlists for a given user based on a dataset of songs, artists, and user ratings. You may use the attached data to get started, but you are also welcome (encouraged!) to collect additional relevant data from the web or other sources. Please document/cite your sources in your final report and include any extra data in your submission.

We've all listened to a playlist, and we all know simply collecting songs that are highly rated is not enough—a truly great playlist must balance variety, discovery, and flow.

Your task is to create one or more personalized playlists for the user using mathematical optimization, simulation, or a combination of both. You will start with a **base model** and then refine it to better reflect real-world playlist curation.

Part I: Base Model Implementation

Using the dataset provided, you will create a playlist of length at least **15 minutes** that maximizes the total sum of user ratings, subject to the following basic constraints:

- **No artist should appear more than once.**
- **No song should be repeated.**

This base model provides a simple framework, but it does not capture important aspects of good playlist design.

Part II: Critiquing the Base Model

After implementing the base model, you will analyze its limitations. Some guiding questions include:

- **Is 15 min the right length for a playlist?** How should playlist length be determined? Should we consider "adaptive" playlists,e.g., if, after your 30-minute run, you want to extend your workout another 10 min?
- **How do we ensure diversity?** Variety is the spice of life! A good playlist isn't just highly rated songs—it balances well-known tracks, deep cuts, and discoveries. It might also blend together genres, languages, artist origin, release decade, or musical style, or niche indie releases with Billboard Top 40. How will you measure and ensure diversity in your playlists?
- **How do we introduce the user to new songs?** How many times can you listen to Selena Gomez's "Love You Like a Love Song" before you stop loving it like a love song?

Simply maximizing known ratings may reinforce past preferences and limit the discovery of new songs. For a subscription platform like Spotify, we might lose customers in the long run. How can we model and account for this?

- **Just vibing.** Factors like beats per minute (BPM), musical key, and energy level can create a natural flow to a playlist. Similarly, suddenly changing these factors might create variety, or might disrupt the flow. How will you think about this?
- **What trade-offs exist?** Ratings aren't everything! How will you balance competing goals that make up a great playlist?
- **One Hit Wonder?** A platform like Spotify will likely rerun your model repeatedly on the same user over time to generate new playlists, and may update it as the users rate more songs. How will your model perform if used in this way? Can you simulate **long-term playlist generation** by assuming that users repeatedly rate songs and update their preferences? How do variety and playlist ratings evolve? Does it depend on how accurate the AI model is?

Remember: A strong critique will not only identify missing elements but also justify which are the most important to address.

Part III: Enriching the Model

Based on your critique, you will develop a richer playlist-generation model that captures some of the limitations you identified. Addressing these issues may require gathering additional data (e.g., genre labels, BPM values, or song popularity metrics). You should justify why your enriched model improves upon the base model and discuss any trade-offs introduced. See the **Overview** portion of this document as well for other tips!

Part IV: Making Recommendations and Offering Insights

After running your enriched model, you will analyze the results and provide business recommendations to the Spotify playlist generation team. A strong analysis will:

- Explain **why the final playlist(s) meet the user's needs better than the base model's output.**
- Discuss the **limitations of the enriched model** and whether any key factors are still omitted, or whether your model of some key factors is too coarse of an approximation.
- Evaluate the **robustness of the solution to poor data quality or changing user preferences.** This is a huge concern in this problem, particularly around user preferences given by the AI model. This might be an ideal place to incorporate sensitivity analysis or simulation for a quantitative discussion of robustness.
- Justify any trade-offs made between competing priorities like diversity, familiarity, and listening flow.

Project Option 2: Improving Entry Checkpoints at USC

Overview

USC's campus entry points serve thousands of people every day, and the design of these checkpoints significantly affects the flow of foot traffic, student experience, and staffing costs. In this project, your team will analyze and improve operations at one specific location: the **Pardee Way pedestrian checkpoint** near JFF and Popovich Hall.

Your goal is to **optimize the layout and operations of this checkpoint**, considering various design elements including the number and arrangement of scanners, the structure of queues, staff deployment, signage, and more. Your project should not rely solely on theoretical modeling—you are **strongly encouraged to visit the Pardee Way checkpoint**, observe operations, and use your observations to inform your modeling choices and assumptions.

If you do visit Pardee Way, please remember:

- Be polite and respectful to all staff and fellow students.
- Do not disrupt the checkpoint operations in any way.
- Do not film or photograph anything without explicit permission.

Currently, I believe the checkpoint has **6 scanners and 3 staff members**, though this may have changed recently. Your project will begin with a simplified simulation model (that assumes this layout) and build toward a richer, more realistic representation of checkpoint operations.

Part I: Base Model Implementation

In the base model, you will simulate operations at the checkpoint using Python. Your initial model will include the following assumptions:

- **Pedestrian arrivals follow an exponential distribution** (i.e., a Poisson arrival process).
- **Service times at the scanners are also exponentially distributed.**
- There is a **single queue feeding into six identical servers (scanners)**.
- Arriving individuals join the queue and are served by the next available scanner.

Note, I have not provided you with explicit data for this part. You will need to do some estimation / make some assumptions to get started.

You will use this simulation model to estimate:

- The **average waiting time** for a pedestrian in the queue.

- The **average number of people in the queue at any given time**.

You should also analyze how these metrics would change if the number of scanners (servers) were increased or decreased.

Part II: Critiquing the Base Model

After building your base simulation, it's time to step back and critically assess its assumptions and limitations. Consider the following questions to guide your critique:

- **Should we have more than one queue?** What if the scanners were split into two separate lines (e.g., three scanners per line)? How would this affect performance and flow?

- **What happens when people switch queues?** Should your model account for individuals switching from a slow-moving queue to a faster one?

- **How do people choose which scanner to use?** If scanners are arranged in a row and people pick the first available scanner, does this block other users from accessing other scanners? How does this affect performance?

- **What about disruptions?** Real-life operations include irregular events—someone forgot their ID, someone needs extra help to register a guest, or someone's large items block others. How might you model these disruptions? Are there other disruptions that the staff point out as particularly problematic?

- **Are arrival and departure lanes the same?** Should we coordinate inbound and outbound traffic to improve flow? How?

- **What happens *after* people pass through?** Pedestrians typically go in one of three directions after entering Pardee Way. Should this influence how the checkpoint is arranged?

- **How do peak periods affect performance?** Students often arrive right before class or after lunch—how would your model perform during peak hours versus off-peak times?

- **What about special days?** During the first week of class or finals, you might expect higher foot traffic or more students with ID issues/guests logging in/etc. How should you plan for such events?

- **What are the costs of the system?** Scanners, staff, and laptops all cost money. Can you estimate annual costs at Pardee Way and consider improvements that could reduce costs while maintaining quality of service?

- **Can you optimize the system design?** Could you formulate an optimization model recommending an ideal configuration of staff and scanners under a cost-performance trade-off?

- **Are there other performance metrics worth tracking?** Beyond average wait time and queue length, what else could you quantify—throughput, staff utilization, cost-efficiency, etc.?

A good critique will not just point out limitations, but argue which ones are most consequential and should be addressed in an improved model.

Part III: Enriching the Model

In this part of the project, you'll design a **richer, more realistic model** that addresses the key limitations you identified. This could involve:

- Modeling **multiple queues and different server arrangements.**
- Adding **disruption events** and analyzing their effect on queue dynamics.
- Incorporating **time-varying arrival rates** to reflect peak and off-peak periods.
- Introducing **alternative queueing disciplines or priority systems.**
- Adding **cost-based decision models** to optimize staffing or equipment levels.

A strong enriched model will not just be more complex—it will be thoughtfully designed to improve performance or insight meaningfully.

> **You must justify your modeling choices, assumptions, and any parameters you introduce.** If you gather data through observation or estimation, describe clearly how and why you made those choices.

Also, keep in mind:

- **Qualitative discussion is not enough.** You must build and analyze a new model.
- If you make simplifying assumptions, **explain why they are reasonable** and test the **robustness** of your recommendations using sensitivity analysis or simulation.

Part IV: Making Recommendations and Offering Insights

After running your enriched model, you will provide actionable recommendations for improving checkpoint operations. Your report should include:

- A comparison between the base and enriched models.
- An explanation of how your recommendations would improve the experience for students and reduce operational costs.
- A discussion of **limitations** in your enriched model and **trade-offs** involved in your recommendations.
- A **robustness analysis**, showing how sensitive your results are to changes in assumptions or data.

Great projects will crisply characterize tradeoffs and suggest a reasonable configuration, or perhaps multiple configurations that could be used in different circumstances.

Project Option 3: Student-Defined Modeling Project

Overview

This is an opportunity for your team to pursue a modeling project that is **personally meaningful**, creatively designed, and grounded in a topic that excites you. Your group will propose your own project topic and develop a quantitative model to analyze and improve a decision-making or operational process using **linear optimization, simulation, or both**.

The goal of this option is to give you flexibility to apply the tools from this course to a domain that matters to you. However, to ensure the project remains appropriate in scope and rigor, **all student-defined projects must be approved by the instructor**.

> **Project approval deadline: April 15**
> To schedule a meeting for approval, please send me a direct message (DM) that includes:
> a) A brief description of your proposed project
> b) The key questions you want to answer
> c) How you intend to use optimization, simulation, or both to address those questions
> d) What data do you already have and what data you think you can reasonably acquire
> *(Note: If your proposal lacks data or relies only on vague plans to collect it later, it will likely not be approved.)*

Important Guidelines

- This is an opportunity for **creativity**, but it must also be **concrete and grounded**.
- I will not approve projects that are **too broad**, **banal**, or clearly replicable from existing internet examples.
- Your project must have a **specific, well-defined goal** and a **reasonable path forward** in modeling and analysis.
- Data availability is critical—if it's not clear where your data is coming from, your project won’t be approved.

Project Structure

Your project deliverables will be **very similar to the other two options**, with one important difference in structure:

- Instead of beginning with a base model and a critique, your report should begin by **motivating your refined model** directly.

- You should explain **why you chose to model the system in the way that you did**, what alternative modeling choices you considered, and why certain features were included or omitted.

- From there, your project should proceed as before:

    - Model implementation in Python
    - Thoughtful analysis of results
    - Actionable recommendations
    - Robustness checks and limitations discussion