Part 1: Problem Set Up

To find the critical path for the restaurant rating prototype, I used multiple files to approach the best case, expected case, and worst-case scenarios for building out the project.

In Excel, I planned out the time estimates (based on personal experience with projects like these and in hours) and individuals required (along with their hourly rates). Using the time estimates, I also created a Gantt chart for the best, expected, and worst-case scenarios. In the Gantt chart, each block represents a two-hour window. As a call out for cost, some tasks require more than one person to work on them. For example, Software Design requires the frontend developer, backend developer, and data scientist. In addition to the tables and Gantt charts in the Excel, there is also a diagram (PM Diagram) that shows how the activities flow from one to the next. The pink boxes represent all the development activities associated with the building the prototype. In the Gantt and tables, you will notice that activity D has no time, due to it being made up of activities D1-D8.

The Excel and PM Diagram would be useless without putting into practice how to solve the critical path. In the Python file, you will find adapted code from a Python PuLP solution prepared by Thomas W. Miller - Revised April 20, 2023. The code is based on run time in hours.

Areas of uncertainty include issues with data ingestion (due to missing data or mismatched data format), people becoming sick (and not being able to work), or other unforeseen circumstances.

Part 2: Model Specification

In this scenario we want to minimize time to build out the product.

- Objective Function: minimizing the end times for all tasks and therefore the project.

- Decision Variables:
    - End Time
    - Start Time
- Constraints:
    - End Time = Start Time + Task Time
    - Start Time = Σ Predecessor Activity Times
        - For example: Activity D4 Start Time = Activity D2 Time + Activity D3 Time
- Cost Function (not included in code):
    - Hourly rate of each employee
        - projectManager: $130.00
        - frontendDeveloper: $125.00
        - backendDeveloper: $130.00
        - dataScientist: $140.00
        - dataEngineer: $130.00
    - Cost by employee (each letter represents the time taken on the task)
        - projectManager: 130*(A+B+C+D1+D5+D7+D8+E+F+G+H)
        - frontendDeveloper: 125*(D2+D4+D6)
        - backendDeveloper: 130*(D2+D3+D4+D5+D6+D8)
        - dataScientist: 140*D2
        - dataEngineer: 130*D3
    - Total Cost
        - 130*(A+B+C+D1+D5+D7+D8+E+F+G+H)+125*(D2+D4+D6)+130*(D2+D3+D4+D5+D6+D8)+140*D2+130*D3

Part 3: Programming

See .py file for code.

Part 4: Solution

- A and B are in parallel (starting at 0)

- C is in parallel with D1
- E starts after B and C end
- D2 and D3 are in parallel
- D5 and D6 are in parallel
- D7 starts after D6 ends
- D8 starts after D7 ends
- F and G are in parallel
- H starts after F and G end

Given the dependencies, it is not surprising that all activities are included in the critical paths. The only difference is duration which clearly relies on preceding tasks. In the best-case scenario, the project would take 118 hours (or about 15 days based on an 8-hour working day). In the expected case, the project would take 224 hours (or about 28 days based on an 8-hour working day). In the worst-case scenario, the project would take 356 hours (or about 45 days based on an 8-hour working day).

Part 5: Overview

At the most, the project will take about a month and a half to be completed (based on the worst-case calculations). This includes time factored in for unpredictable issues arising (such as someone being sick or problems with the code). Based on the worst-case scenario, a fully functioning prototype could be expected in week 5. At most in the worst-case scenario, the project would cost $119,140, and $41,470 in the best-case scenario. If we were able to add more employees (backend developers and front-end developers) we would be able to minimize time spent on the project. Given there is such a large discrepancy between costs, it could help

to run analyses using Monte Carlo or Stochastic Programming to account for uncertainties and understand what could be avoided to remain on the lower end of the cost spectrum. While Monte Carlo wouldn't optimize the schedule it would provide variations to it such as potential delays or unforeseen costs. Stochastic programming would work to find the optimal solution given the other uncertainties so that the process for the product is more dynamic and help minimize time and cost.

References

Dinwiddie, George. *Software Estimation Without Guessing: Effective Planning in an Imperfect World.* Dallas, TX: Pragmatic Bookshelf. [ISBN-13:978-1680506983]

Falk, Kim. 2019. *Practical Recommender Systems.* Shelter Island, NY: Manning. [ISBN-13: 9781617292705]

Hillier, Frederick S., and Gerald J. Lieberman. 2021. *Introduction to Operations Research* (eleventh edition). New York: McGraw-Hill. [ISBN-13: 978-125987299-0] Online resources, including Chatper 22 Project Management withPERT/CPM at https://highered.mheducation.com/sites/dl/free/1259872998/1126268/Hillier_IOR_11e_Ch022_WebChapter.pdfLinks to an external site.

Majumdar, Partha. 2021. *Linear Programming for Project Management Professionals: Explore Concepts, Techniques, and Tools to Achieve Project Management Objectives.* India: BPB Publications. [ISBN-13: 978-9355511164]

Williams, H. Paul. 2013. *Model Building in Mathematical Programming* (fifth edition). New York: Wiley. Section 5.3.7 Critical Path Analysis, pages 94–98. [ISBN-13: 978-1-118-44333-0] We provide a Python PuLP solution to Williams' home building example using an activities dictionary and derived start_time and end_time decision variables: Williams-Critical-Path-Analysis.pyDownload Williams-Critical-Path-Analysis.py

Wisniewski, Mik, and Jonathan H. Klein. 2001. *Linear Programming/Critical Path Analysis.* New York: Palgrave. [ISBN-13:978-0-333-76355-1] Chapter 8, Critical Path Network Analysis Techniques, pages 119–147. Available from Course Reserves.