To Do or Not To Do Hackathon

B057 - Precious Sukhi
B061 - Harshita Ranka
B062 - Sanjana Surana

TABLE OF CONTENTS

01

The Eisenhower Matrix

02

The Simplex Method

03

Integrating Data Analysis

04

Achieving Machine Learning Objectives

TABLE OF CONTENTS

05

Other Prioritization Techniques

06

Improving the task manager

The Eisenhower Matrix

WHAT IS THE EISENHOWER MATRIX?

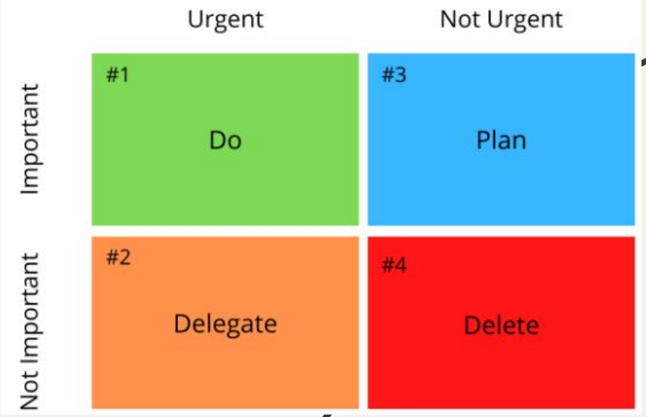
- The eisenhower matrix is a time management tool that helps prioritize tasks based on urgency and importance. It categorizes activities into four quadrants:

  1. *Urgent and Important (Do First)*

  2. *Important, but Not Urgent (Plan)*

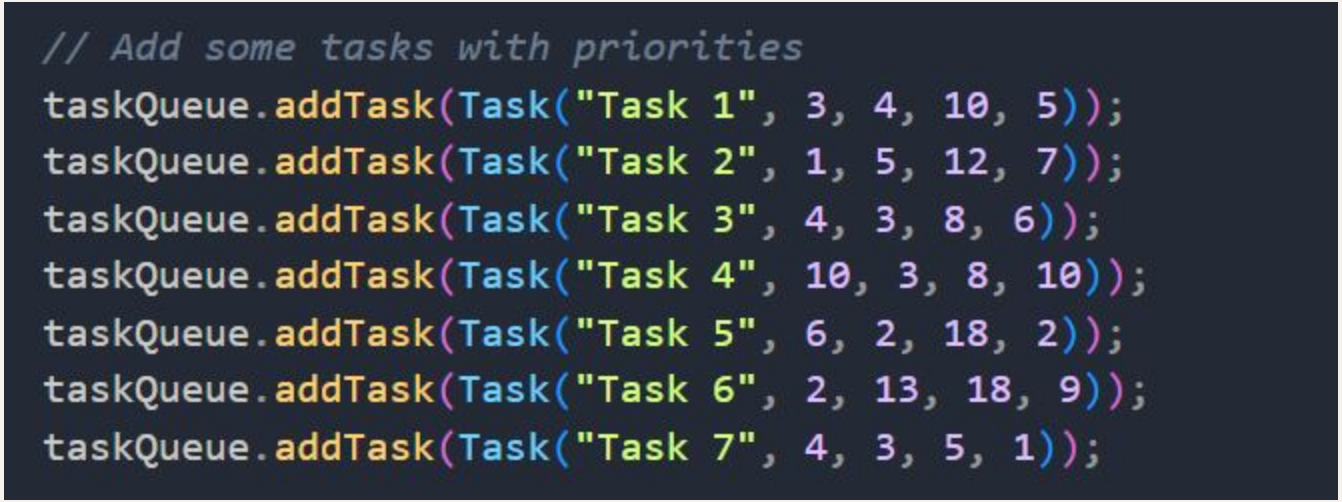
  3. *Urgent, but Not Important (Delegate)*

  4. *Not Urgent and Not Important (Delete)*



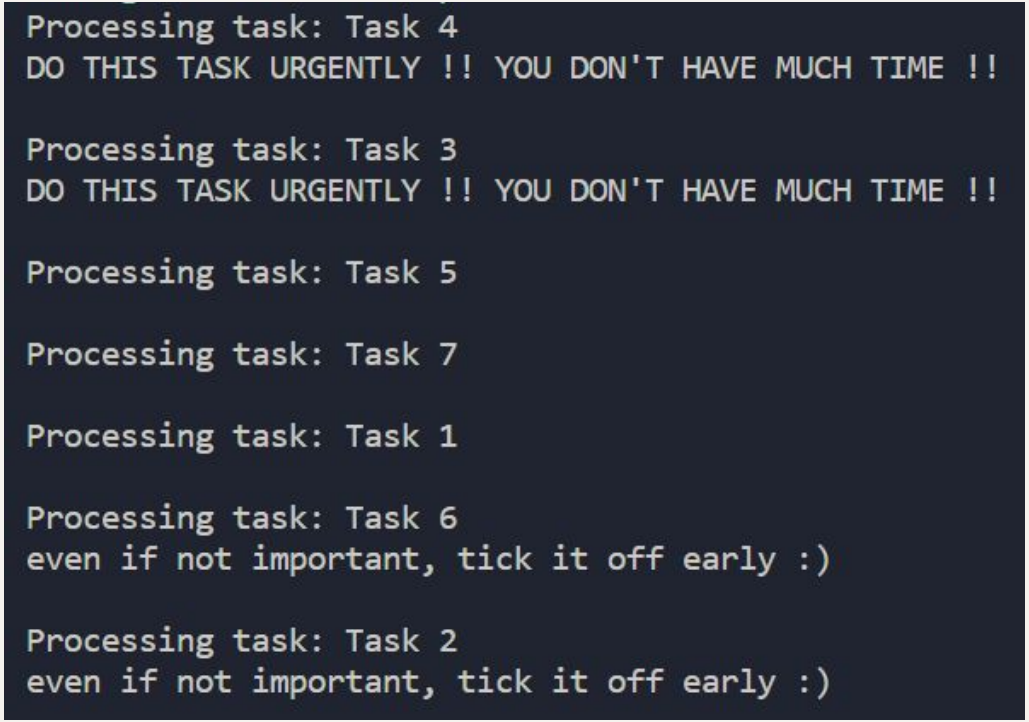
	Urgent	Not Urgent
Important	#1 Do	#3 Plan
Not Important	#2 Delegate	#4 Delete

The Eisenhower Matrix - Input



```
// Add some tasks with priorities
taskQueue.addTask(Task("Task 1", 3, 4, 10, 5));
taskQueue.addTask(Task("Task 2", 1, 5, 12, 7));
taskQueue.addTask(Task("Task 3", 4, 3, 8, 6));
taskQueue.addTask(Task("Task 4", 10, 3, 8, 10));
taskQueue.addTask(Task("Task 5", 6, 2, 18, 2));
taskQueue.addTask(Task("Task 6", 2, 13, 18, 9));
taskQueue.addTask(Task("Task 7", 4, 3, 5, 1));
```

The Eisenhower Matrix - Output



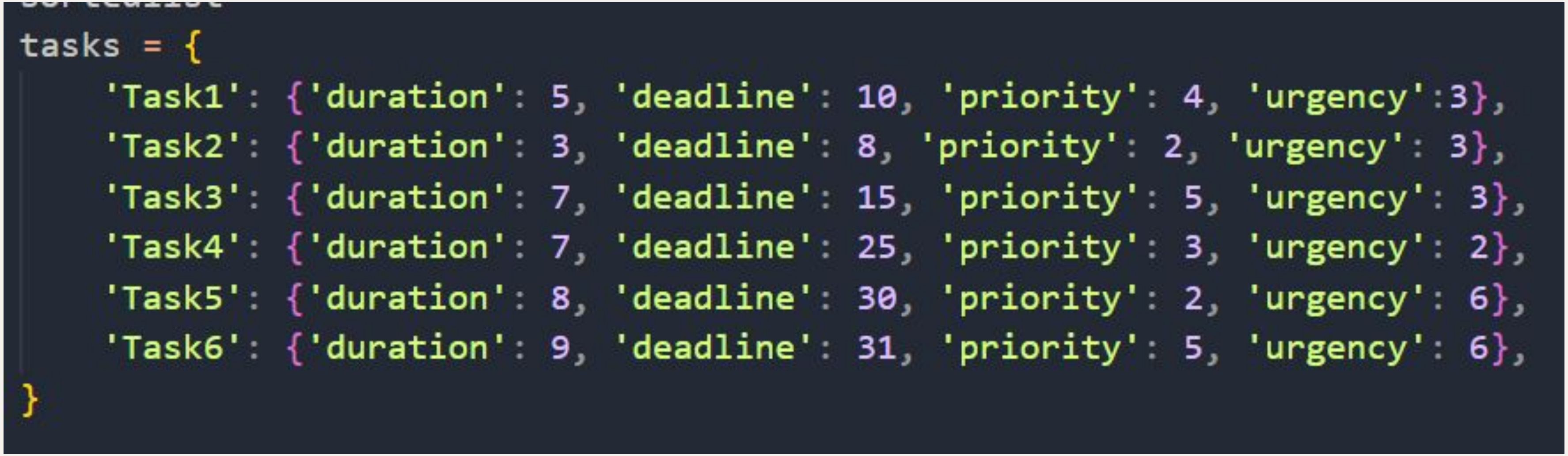
```
Processing task: Task 4
DO THIS TASK URGENTLY !! YOU DON'T HAVE MUCH TIME !!

Processing task: Task 3
DO THIS TASK URGENTLY !! YOU DON'T HAVE MUCH TIME !!

Processing task: Task 5

Processing task: Task 7

Processing task: Task 1

Processing task: Task 6
even if not important, tick it off early :)

Processing task: Task 2
even if not important, tick it off early :)
```

The Simplex Method

WHAT IS THE SIMPLEX OPTIMIZATION METHOD?

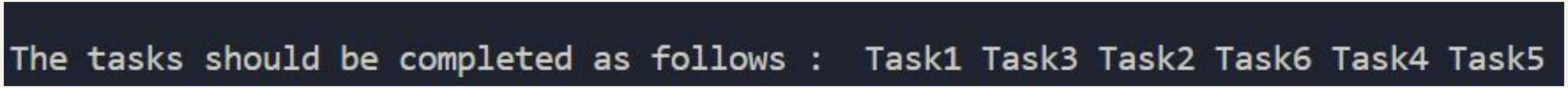
- The simplex method is a algorithm for solving linear programming problems, where the goal is to maximize or minimize a linear objective function subject to linear equality and inequality constraints.
- It starts with an initial feasible solution and iteratively moves towards the optimal solution by pivoting through vertices of the feasible region.
- It explores the space of feasible solutions, adjusting variable values to improve the objective function while maintaining feasibility.
- The algorithm continues until an optimal solution is found or it is determined that no improvement is possible.

The Simplex Method - Input



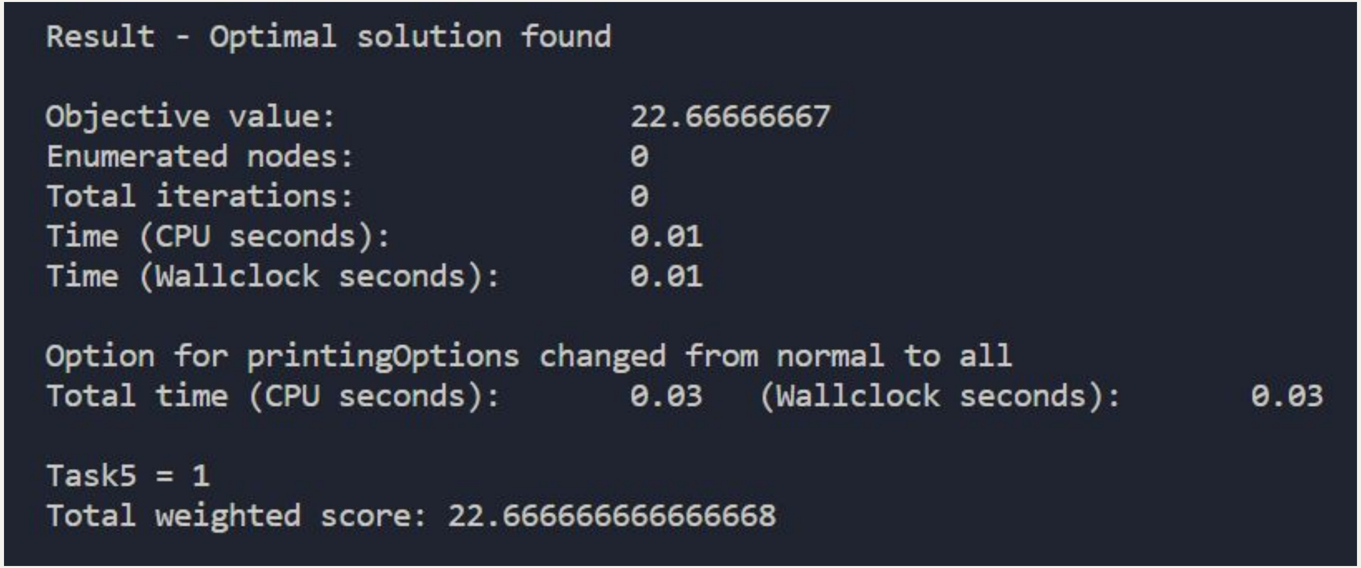
```
tasks = {
    'Task1': {'duration': 5, 'deadline': 10, 'priority': 4, 'urgency':3},
    'Task2': {'duration': 3, 'deadline': 8, 'priority': 2, 'urgency': 3},
    'Task3': {'duration': 7, 'deadline': 15, 'priority': 5, 'urgency': 3},
    'Task4': {'duration': 7, 'deadline': 25, 'priority': 3, 'urgency': 2},
    'Task5': {'duration': 8, 'deadline': 30, 'priority': 2, 'urgency': 6},
    'Task6': {'duration': 9, 'deadline': 31, 'priority': 5, 'urgency': 6},
}
```

The Simplex Method - Output



```
The tasks should be completed as follows :  Task1 Task3 Task2 Task6 Task4 Task5
```

For each task it also gives a weighted score (our problem statement)



```
Result - Optimal solution found

Objective value:                22.66666667
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.01
Time (Wallclock seconds):       0.01

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.03   (Wallclock seconds):       0.03

Task5 = 1
Total weighted score: 22.666666666666668
```

Integrating Data Analytics

- **Anomaly detection algorithms** can be used for time tracking and efficiency metrics - to track & monitor the time spent by employees on tasks
- **Resource allocation algorithms** can be used for employee workload distribution - ensure balanced allocation of tasks
- **Sentiment analysis** can be used to map employee engagement and prevents burnout
- **K-means or hierarchical clustering** can be used for employee skill profiling

Achieving Machine Learning Objectives

- **Linear regression** can be used to predict time required for time completion and analyze temporal dependencies in the task completion times
- **Time Series Forecasting** for Workload Prediction
- **Natural Language Processing** for Task descriptive analysis
- **Deep Reinforcement learning** for adaptive task scheduling - The Simplex algorithm can use it to adaptively adjust task schedules based on real-time feedback, employee performance, and changing task priorities.

Other Prioritization Techniques

- **MoSCoW Prioritization:** This technique categorizes tasks into Must-have, Should-have, Could-have, and Won't-have categories. It's beneficial for distinguishing between essential tasks and those that are less critical.
- **Eat the Frog:** The eat the frog strategy is a prioritization and productivity method used to help people identify difficult tasks. The idea is that you identify one challenging task (the frog) and complete the task first thing in the morning (eating it).

Improving the task manager

Since this task manager is a very basic application, here are the ways we can improve it:

- A weekly/monthly review of stuff completed versus left (stored in user's history) in the form of graphs.
- Limit multitasking.
- Ensuring sync across devices.
- Keep track of the time a user spends on a task depending on its importance.
- A result bar for the user wherein they can input how effective the completion of the task was.
- Include rewards for completing tasks before time.
- A collaboration with a mental health chat bot like wysa to ensure employees are not too stressed.

THANK YOU