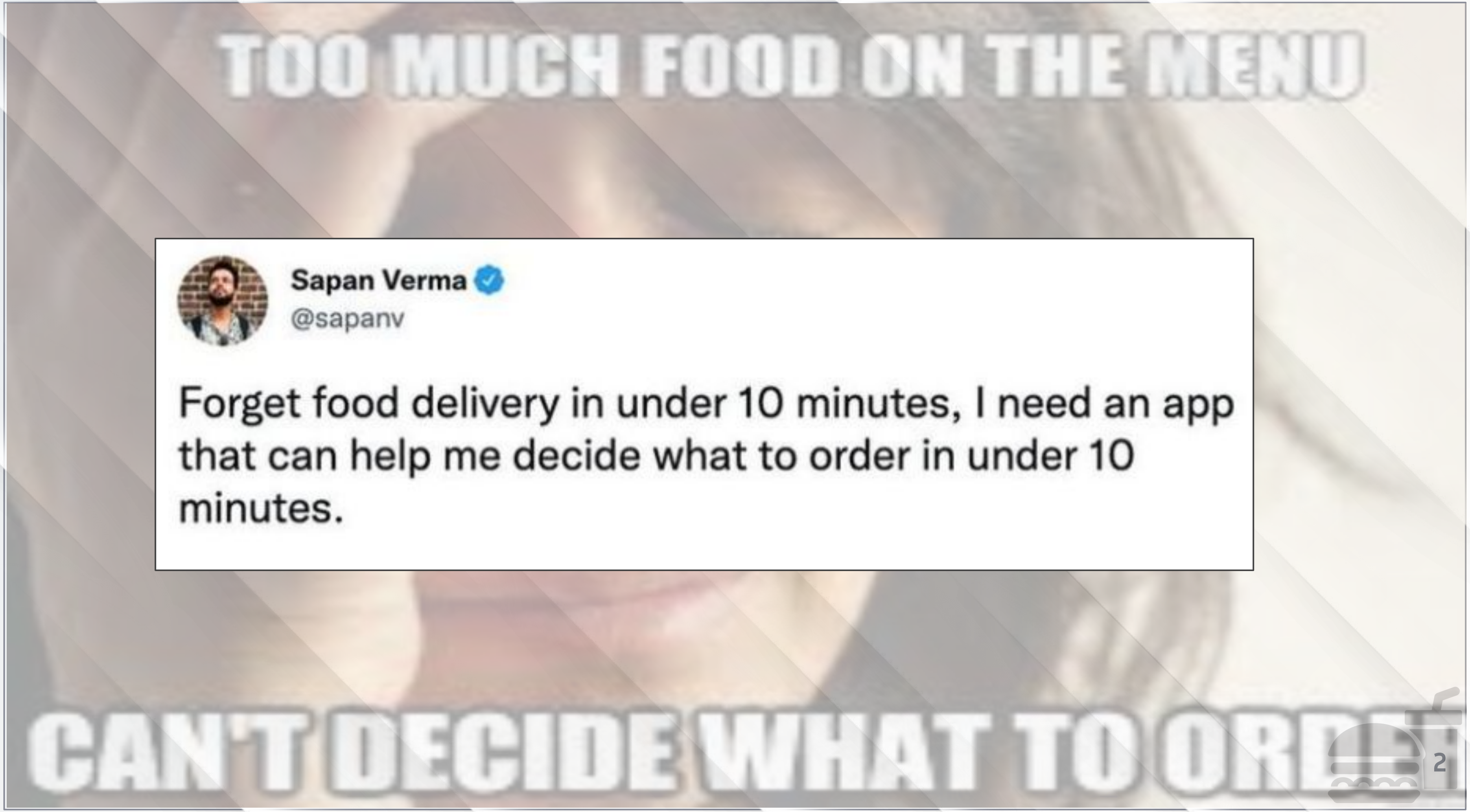
# Food Cart Optimisation

ME 308 Project: Group 17


TOO MUCH FOOD ON THE MENU
Sapan Verma
@sapanv
Forget food delivery in under 10 minutes, I need an app that can help me decide what to order in under 10 minutes.
CAN'T DECIDE WHAT TO ORD

# Contents

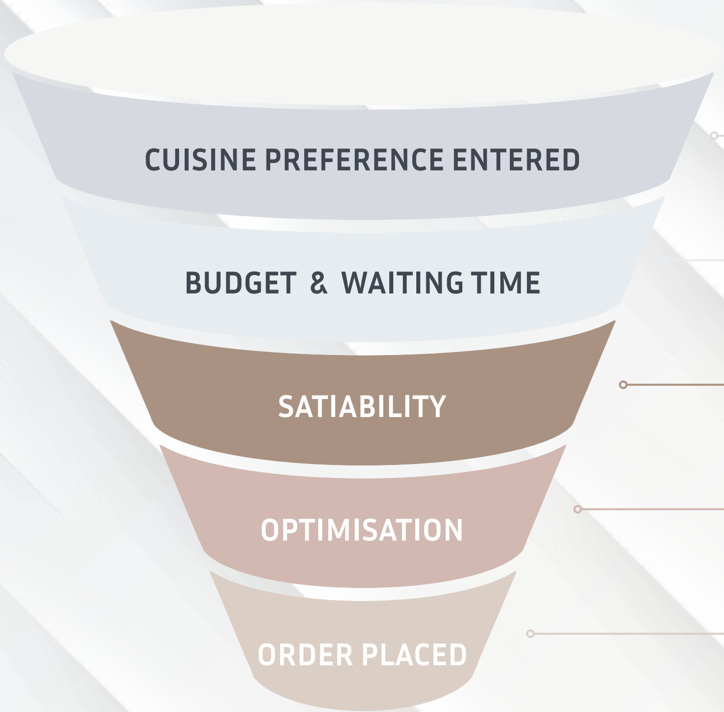
# OVERVIEW

# What is our project about?

- To aid the decision making process of ordering food from an online delivery platform

- Optimises the cart considering various parameters and user preferences

# Flow Of Information




The user enters his cuisine preferences

The user enters his budget & the amount of time he is willing to wait

The user enters his calorie requirements

The food is optimized for utility

Final order(s) based on maximum utility, keeping in mind all the constraints

# FORMULATION

# Modeling the World

### Cost

Assigned for each food item in every outlet

### Rating

Rating of individual food items in different outlets; doesn’t exist if the food item isn’t served in a particular outlet

### Calories

Stores the calories of each food item, changes from outlet to outlet depending on the preparation

### Distance

Distance of the outlet from the user

### Preparation time

Standard prep time of a particular food item

### Outlet efficiency

Efficiency of the shop in preparing and serving a particular food item

# User Inputs

### Cuisine

User's cuisine preference

### Budget

The maximum amount the user is willing to pay

### Satiability Requirement

The user's minimum calorie requirement from his order

### Delivery Time

The maximum time the user is willing to wait

# Calculations

## Total delivery time

$$T_{ij} = P_j \cdot E^{-1}_{ij} + D_i \cdot v^{-1}$$

$T_{ij}$ = total time of $j^{th}$ food item from $i^{th}$ shop

$P_j$ : preparation time for for $j^{th}$ food item

$E_{ij}$: $i^{th}$ shop's Efficiency in preparation of $j^{th}$ food item

$D_i$: Distance of $i^{th}$ shop from IITB

v : Average vehicular speed in Mumbai

## Desirability of a food item

$$R_{ij} = 0.6 \cdot L_{ij} + 2 \cdot Y_J$$

$R_{ij}$ : desirability of $j^{th}$ food item in shop i

$Y_I$ : user preference of cuisine of food j

$L_{ij}$ : rating of $j^{th}$ food item in $i^{th}$ shop

# The Problem

## Parameters:

Cost Matrix = $C_{ij}$

Calorie Matrix = $S_{ij}$

Desirability Matrix = $R_{ij}$

Time Matrix = $T_{ij}$

For food item j in $i^{th}$ shop

Maximum wait time = t

Budget = b

Calorie Requirement = s

**Decision Variable:**

Decision = X

**Objective Function:**

Utility = $\Sigma^{n}_{i=1,j=1} \; R_{ij} \cdot X_{ij}$

**Constraints:**

Cost : $\Sigma^{n}_{i=1,j=1} \; C_{ij} \cdot X_{ij} \leq b$

Time : $\forall_{i,j} \; T_{ij} \cdot X_{ij} \leq t$

Satiability: $(1.2 \cdot s + 200) \geq \Sigma^{n}_{i=1,j=1} \; S_{ij} \cdot X_{ij} \geq s$

# HEURISTIC METHOD

# Greedy Algorithm

- **Metric 1 - Rating / Cost**

  Sort all items on the basis of this metric

  Keep adding the items starting at the top, keeping in mind-

  - Adding an item shouldn't lead us to spend more than the user's budget
  - The time taken for the food to be delivered should be lesser than the limit imposed by the user
  - The item must exist, which means the rating of the item should be > 0

Ideally, we should end up with an order that maximises average rating.

**Issue -** The calorie requirement might not be met. Solution?

# Greedy Algorithm

- **Metric 2 - (Calories ▪ Rating$^2$ ) / Cost**

  Sort all items on the basis of this metric

  From the initial order, remove the item that performs the worst on the basis of this metric

  Search for the items that perform the best according to this metric which are not in the cart

  Add these items, keeping in mind-

  - Adding an item shouldn't lead us to spend more than the user's budget
  - The time taken for the food to be delivered should be lesser than the limit imposed by the user
  - The item must exist, which means the rating of the item should be > 0

  Keep replacing worse performing items in the cart with better performing ones subject to above conditions till we cross the minimum satiability threshold

# Why the Heuristic Approach?

## Reliable

Will provide an optimal solution over widely different operating conditions

## Efficient

Known time complexity O(nlogn)

## Flexible

Possible to try out various combination of metrics for improving results

## Easy to Use

Easier to run diagnostics for someone who is not well-versed in LP optimisations

# RESULTS AND ANALYSIS

# Dataset

## Cost

50 shops x 26 items

Range : Rs 150 - Rs 650

## Rating

50 shops x 26 items

Average : 3

Standard Deviation:1.28

## Calories

50 shops x 26 items

5 cuisines: 5, 5, 5, 5, 6 items each

Average : 511.39

Standard Deviation: 189.97

## Processing Time

26 items x 1

Average : 28.825

Standard Deviation: 9.47

## Distance

50 shops x 1

Average : 11.09 Km

Standard Deviation: 5.78 Km

## Shop Efficiency

50 shops x 26 items

Average : 0.8

Standard Deviation: 0.08

# Linear Program

<u>User Preferences:</u>

budget = Rs 1200

time = 41 minutes

capacity = 3200 Cal

<u>Optimal Order:</u>

Cost: Rs 1198

Average rating: 4.038

Calories: 3700

| Shop Number | Item Number | Cost | Rating | Calories |
|---|---|---|---|---|
| 20 | 24 | 157 | 4.3 | 350 |
| 34 | 24 | 191 | 4.8 | 340 |
| 29 | 25 | 164 | 4.1 | 420 |
| 5 | 5 | 106 | 3.2 | 400 |
| 16 | 5 | 124 | 3.8 | 480 |
| 29 | 5 | 133 | 3.5 | 470 |
| 50 | 5 | 147 | 4.6 | 480 |
| 50 | 1 | 176 | 4.0 | 760 |

# Greedy Algorithm

**User Preferences:**

budget = Rs 1200

time = 41 minutes

capacity = 3200 Cal

**Optimal Order:**

Cost: Rs 1173

Average rating: 4

Calories: 3230

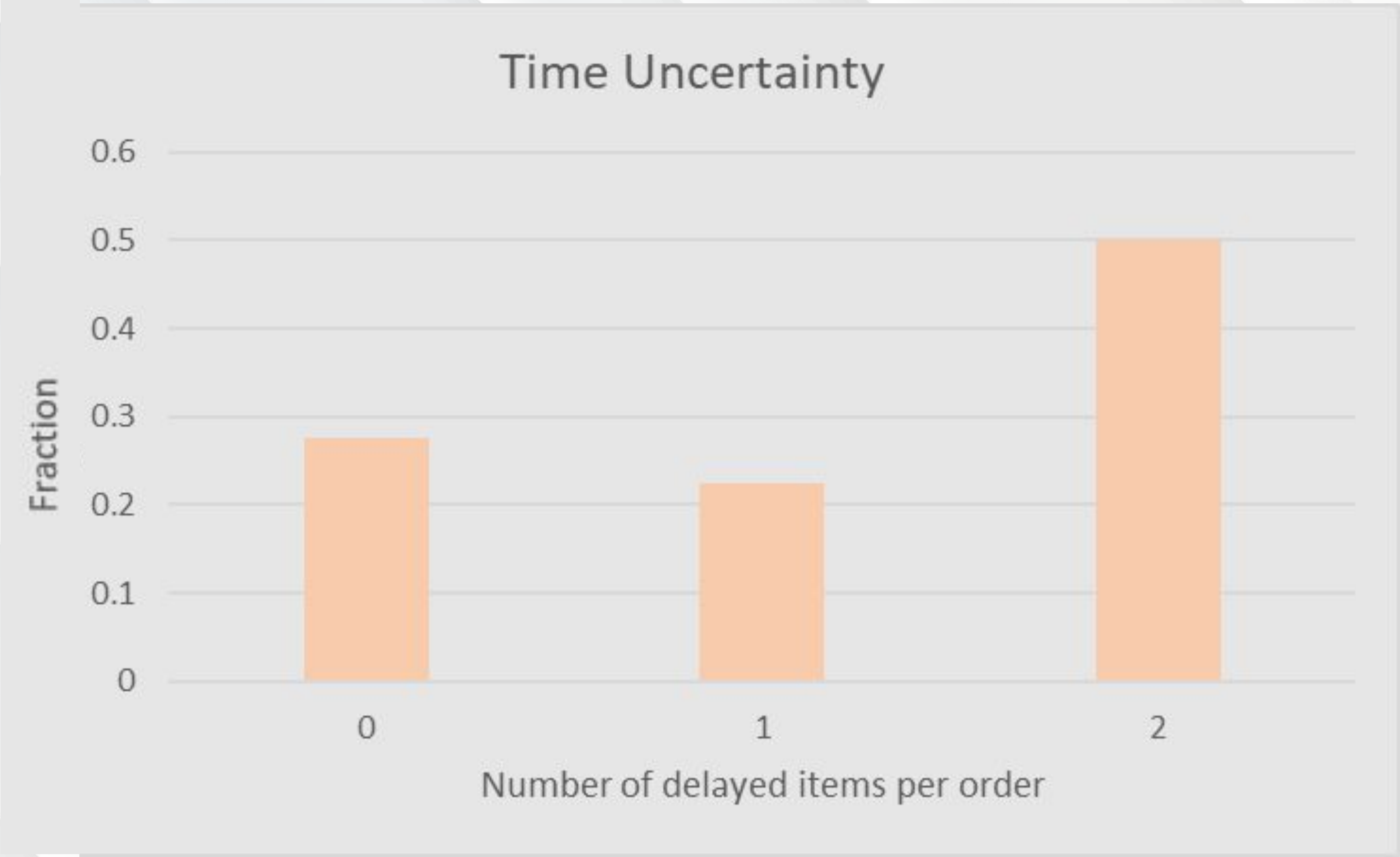
| Shop Number | Item Number | Cost | Rating | Calories |
|---|---|---|---|---|
| 20 | 24 | 157 | 4.3 | 350 |
| 46 | 20 | 86 | 4.3 | 240 |
| 50 | 1 | 176 | 4.0 | 760 |
| 34 | 24 | 191 | 4.8 | 340 |
| 29 | 25 | 164 | 4.1 | 420 |
| 5 | 5 | 106 | 3.2 | 400 |
| 1 | 5 | 103 | 3.6 | 390 |
| 12 | 8 | 190 | 3.7 | 330 |

# Uncertainty Analysis

- To take into account variability in food preparation time  and uncertainties introduced by varying traffic conditions

- Element wise multiplication of processing time array and distance array with random samples from a Gaussian distribution with mean 1 and standard deviation 0.12

- Found the number of items delayed per order for 200 trials due to time uncertainties


Time Uncertainty
0.6
0.5
0.4
0.3
0.2
0.1
0
Fraction
0
1
2
Number of delayed items per order

# Uncertainty Analysis

- To take into account uncertainties introduced in calorie count during food preparation
- Element wise multiplication of processing time array and distance array with random samples from a uniform distribution with lower bound 0.8 and upper bound 1.2
- Found the number of times items go above or below the calorie specification, 2200 in our test

Calories Uncertainty
Fraction
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
<2200
2200-2300
>2300

# CHALLENGES & LEARNING

# Roadmap of our project

1. Understood the problem and choose prospective decision variables
2. Simplified the original problem by removal of independent variables, shrinkage of matrices
3. Formulated the problem statement, started coding on AMPL
4. Obtained some results, coded a greedy approach on Python, heuristic approach for delivery time
5. Expanding the problem to take into account bigger matrices to emulate a real-world analysis
6. Analysis and Amalgamation of data and results

# Key Challenges

- Simplifying and formulating the data set
- Accounting for variability
- Debugging in AMPL solver

# Our Takeaways & Learning

- Learnt how to apply course work into a real-life application
- Learnt AMPL and dove deeper into python programming
- Learnt how to work in a team and learnt how to pitch our ideas

# GOING FORWARD

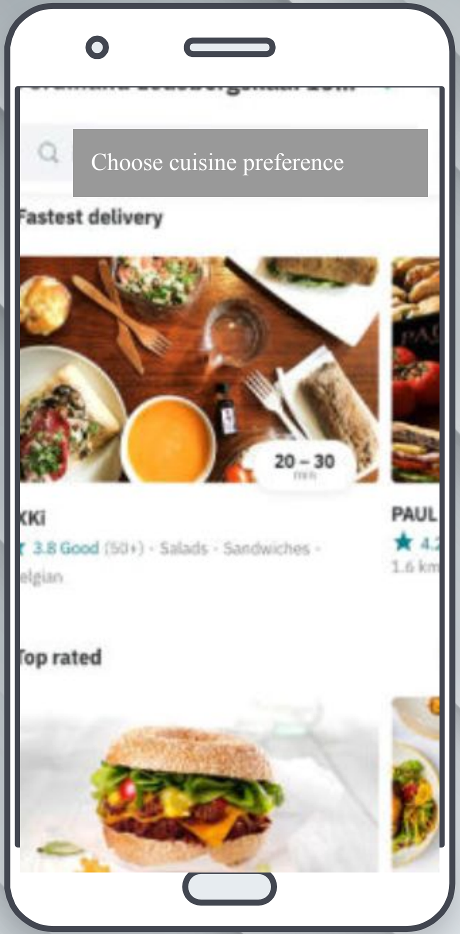
# Further Improvement and Addition

- Extensive tweaking of the metrics in the greedy algorithm to maximize accuracy and improve performance
- Using a larger, more realistic dataset for our analysis
- Integrate this model into a GUI to make it user-friendly

# Mobile App

Integration of the code into an app : a one-stop app for all the foodies out there!

.



# Our Team

**Ritwik Gupta**

190110075

**Aruja Khanna**

190100024

**Anagha Savit**

190100014

**Shiv Modi**

19D100011

A special note of thanks to Prof. Avinash for his invaluable guidance which helped us throughout the course of our project, and our TA, Aditya Iyengar, for his advice

# Thank You!

Have a nice day!