

Ankur Garg  
12/12/2023

## Building a Crowd-Sourced Travel Itinerary Recommender System

For my final project, I decided to merge my educational passion for innovation with my passion for travel. Throughout the years, I have planned many trips, and the worst part about it is the need for more organization regarding real people and their recommendations. Given everything we have learned in this course, such as metadata, sensemaking, structure, and databases, I figured I had the perfect toolset to take on the challenge. Having completed it and now writing this report, I am happy with the result and looking forward to improving it as I get more time in the coming months.

To develop this implementation project, I used Python and the pandas library, leveraging pandas' data manipulation power to create well-curated results. I also used Microsoft Excel to custom-build the dataset and organize the data to make it easier to manage inside Python. In the final result, users could specify travel preferences such as destination, rating, cost, transportation mode, type of stay, and preferred season. They were also able to dial these features using hierarchical structures.

### Technical Implementation

We are starting with the technical implementation. Given that this project requires the shortest paper of the three, we must cover precisely the implementation that took place and how you can use it to its full capabilities.

Before I even entered Python, I was struck by a significant problem. I had a set of standards I wanted to meet regarding the variables I wanted to consider, and I needed a source to give me a reliable dataset to match those needs. By combining datasets such as the ones linked here:

Traveler data: <https://www.kaggle.com/datasets/rkiattisak/traveler-trip-data/>

Airport source: <https://www.kaggle.com/datasets/rajkumarpandey02/list-of-busiest-airports-by-passenger-traffic/data>

Car Rental Source: <https://brandirectory.com/rankings/car-rental-services/>

I was able to do a combination of merging and generating data. This ended up very on theme because when I was later reviewing the Traveler data I used, I found a note about how it was also randomly generated. As you go through the data, you will also find the experimental tab, where I was able to bring in and note some of the variables I used to generate the data.

When looking through the data, I organized it as if the entries on the primary page are the main categories, and each page dives deeper into the subcategories of each of their respective areas.

Once I started coding, much of it was sifting through and retrieving data based on user input. With thirteen variables that can be played with, I had my work cut out. I created functions that

would each sift through the data frame and return results and indexes. These indexes would be fed into the driver code for the final work.

The user can decide which factors to consider in the driver code, as you will soon experience. Based on their response, I call the appropriate functions. One of the beauties of this format is that it allows the user to control just how far into the hierarchy goes. They can stay as surface level as looking at every itinerary for a single destination, or they can independently dictate the budget for their stay, transport, and activities to find the perfect itinerary for them. The use cases are endless, and I will demonstrate them soon.

The final step after the user provides their input is to conclude and present the information to them. The program outputs the itineraries that most match the users' needs and gives the users a comprehensive view of potential trips they can take. It filters out unnecessary columns to keep things concise and easy on the eyes.

#### Use cases: How to Use an Examples

There are a few essential things to keep in mind for the program.

- Responses are case-sensitive; please respond precisely how it shows. Usually, it will be standard, first letter capital followed by all other lowercase.
- As only 140 data points are being cut down by 13 different variables, some attempts you make may lead to no result.
- To help combat this frustration, I have provided verified inputs so you can play around with them.

#### Use case 1:

- The user lives in Denver, Colorado, and has yet to decide where to go for their next summer vacation. They need a surefire trip after countless past failed trips from random articles on Google. Something that's been validated by an actual person. They come to us. By inputting just their Origin city and season, they get three perfect itineraries, including a road trip and two international trips with varying budgets. They can also see the demographics of those who went on the trip before and if they liked it.
- Input to see this result (in order):
  - o BG
  - o Denver
  - o C

#### Use case 2:

- The user is a lot more picky. They are open to leaving from anywhere in Colorado, so they input the state instead of the city. Again, they don't know where they want to go, but their main focus is the journey; while they don't care about transport or hotel costs, they want to keep their activities costs under \$100. They don't want to drive and prefer the comfort of room service, so they select a hotel. They don't care about the chain since they don't have points anywhere.

- Input to see this result (in order):

- ☐ BDEF
- ☐ CO
- ☐ D
- ☐ 100
- ☐ B
- ☐ No
- ☐ A
- ☐ No