MVP- Version 1- IshLa

**AIDI 2005 CAPSTONE II**

**Course Facilitator: Marcos Bittencourt**

**Prepared By:**

Izza Godinez - 100556078

Maviya Javed Shaikh - 100766785

Meryl Gabrielle Tubio – 100763231

Nandini Malhotra - 100768797

February 19th, 2021

Contents

[1. Executive Summary 3](#_Toc64645413)

[2. Rationale Statement 3](#_Toc64645414)

[3. Problem Statement 3](#_Toc64645415)

[4. AI Agent 3](#_Toc64645416)

[5. Data 3](#_Toc64645417)

[6. Product Core Functionality 4](#_Toc64645418)

[i. Data Pre-Processing: 4](#_Toc64645419)

[ii. Retrieving User Address: 6](#_Toc64645420)

[iii. Retrieving Places User want to visit and creating a list: 7](#_Toc64645421)

[iv. Filtering Dataset 7](#_Toc64645422)

[v. Shortest Path Dataset 8](#_Toc64645423)

[vi. Calculating Distance from one location to multiple location 8](#_Toc64645424)

[vii. Map of all locations’ user want to visit. 11](#_Toc64645425)

[7. Next Step 11](#_Toc64645426)

[8. Project Plan 12](#_Toc64645427)

# Executive Summary

Optimal path algorithms are a major interest in artificial intelligence. It has many applications in business, including product delivery and shipment processes, as well as in navigation. The aim for this project is to allow the user to create an optimal path between different stores to allow them to complete their shopping lists with various items in the shortest possible time.

# Rationale Statement

To develop a website that takes in user credentials and asks user to select their TO-DO list by clicking on various sub-categories like groceries, electronics, clothing apparels. The user selects items from these sub-categories. Based on the items selected, the website suggests a route from the user's current location that he can take to complete all its errands in one go.

# Problem Statement

“To find the optimal path from the starting point (source), covering all the sub-destinations”.

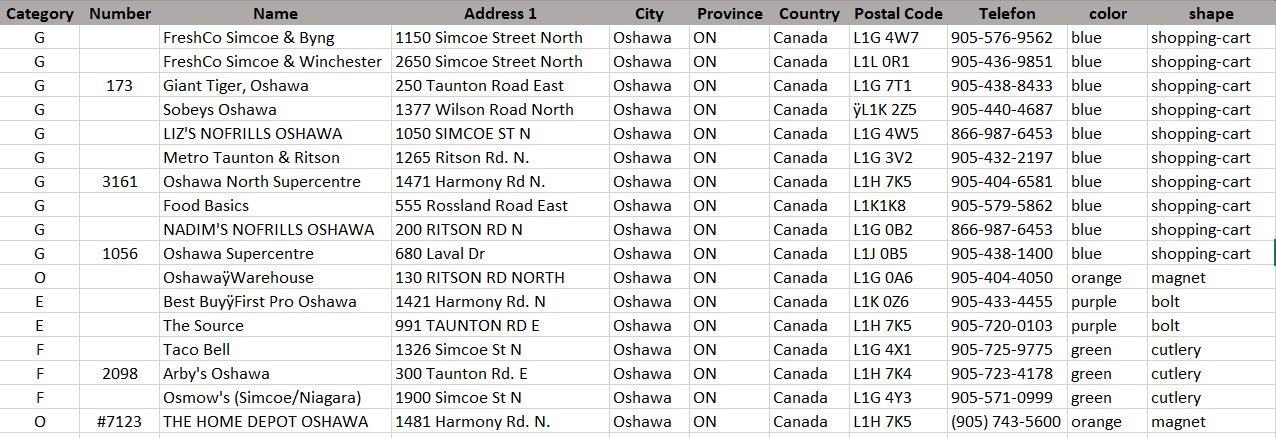
The problem statements aim at determining the closest sub-destination from the current position, the starting point being the user’s current location.

# AI Agent

AI agent is the artificial intelligent component that helps to achieve the goal of our project. Based on the business problem to find the shortest path for the user, we will be using Utility-based AI agents. The reason for choosing Utility-based AI agent is that it not only helps to find the shortest path, but it also provides the best possible way to achieve it.

Another advantage of using Utility-based AI agent is that in our project there will be multiple alternatives i.e. multiple paths to choose from and agents must choose the best path amongst all. Below is the architecture for utility-based AI agent.

# Data

A dataset containing the addresses of various destinations is used to help find the optimal path from user’s current location to various sub-destinations. 

Where:

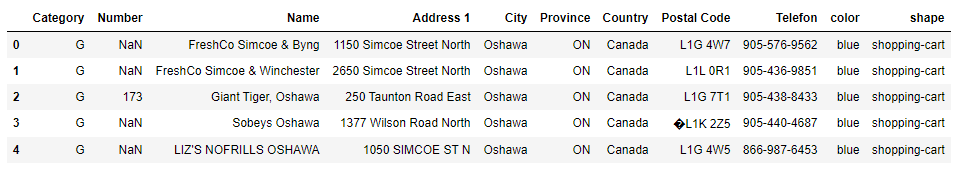
1. **Category:** It is the category of the store for various kinds of errands like Groceries(G),Electronics(E), Food(F), Clothing(C), Others(O), which might include Home Depot, Gas Stations and Car Wash
2. **Number:** It includes the Store Number if valid.
3. **Name:** It includes the name of the store.
4. **Address 1:** This field contains the street number and street name of the store.
5. **City, Province, Country, Postal Code:** These fields would specify the city, province, country and postal code of the store.
6. **Telefon:** This field would contain the telephone number of the store.
7. **color, shape:** These two fields are solely for the purpose of plotting the destinations on the map intuitively.

# Product Core Functionality

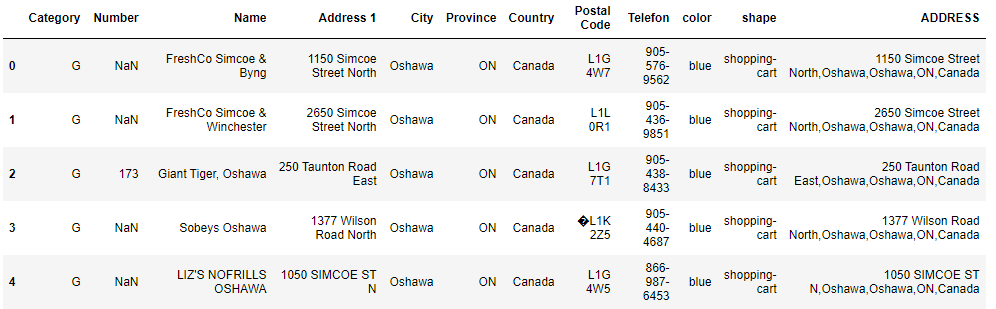
This section would explain the core functionality of the application as coded in backend, in python.

## Data Pre-Processing:

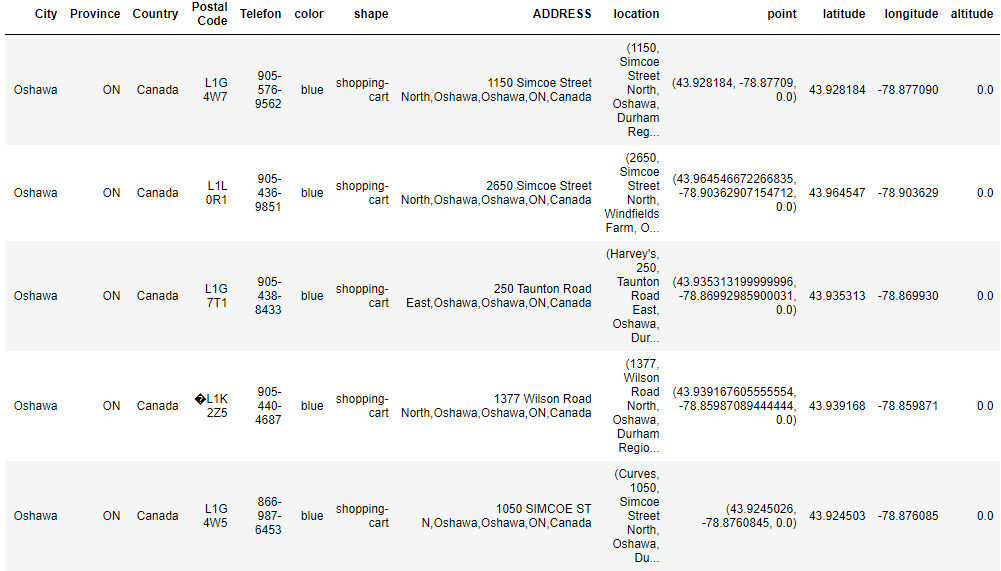
1. Loading the dataset



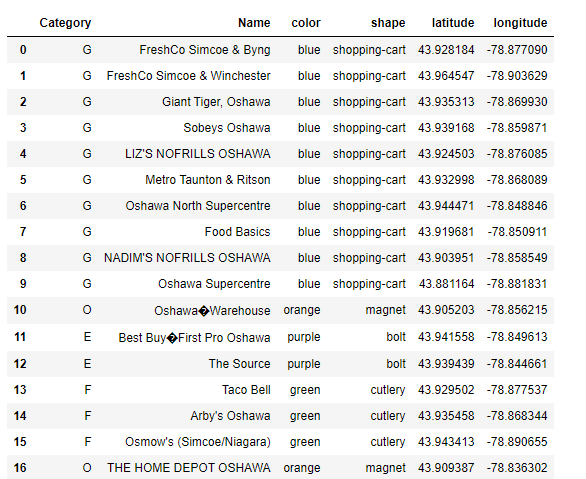
1. Combining the addresses in one column



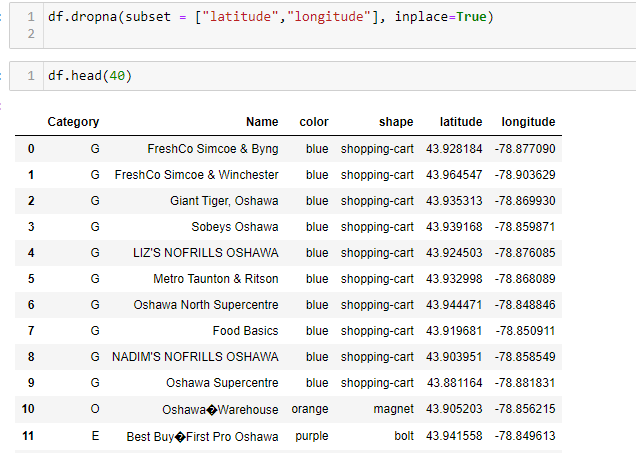
1. Performing feature engineering by calculating the Latitude and Longitude of the stores based on the full address



1. Removing the irrelevant fields.

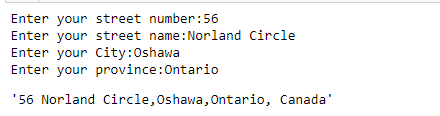


1. Dropping the rows with ‘na’ values in longitude and latitude column.



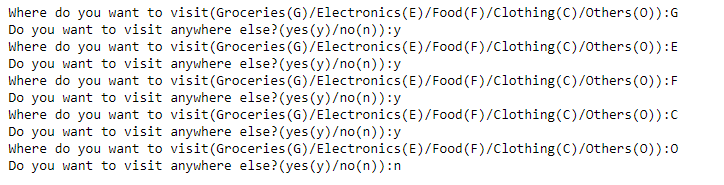
## Retrieving User Address:

Asking user for their address to get the user’s street number, street name, city, and province.



## Retrieving Places User want to visit and creating a list:

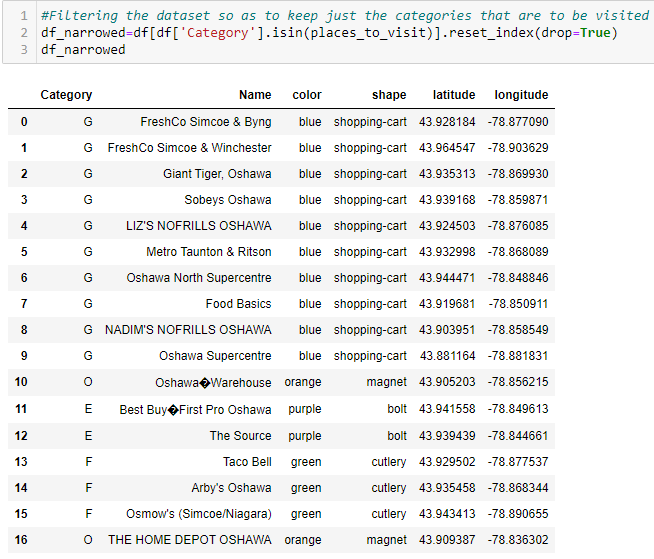
Asking user to select the category of the product user want to buy



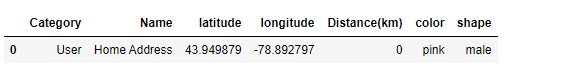


## Filtering Dataset

The cleaned dataset that we have is filtered based on the categories of the stores that the user wants to visit.



## Shortest Path Dataset

This dataset will be the final dataset that would contain the sub-destinations along with the user’s address. This dataset would also be updated every time the user checks out from one store and is planning to travel to another nearest store.   


## Calculating Distance from one location to multiple location

We are using two methods to calculate the distance. First is **Open Source Routing Machine** and second is **Haversine Distance.** It was found that Open Source Routing machine method provides more accurate distance than Haversine distance method by validating the result with Google Map.

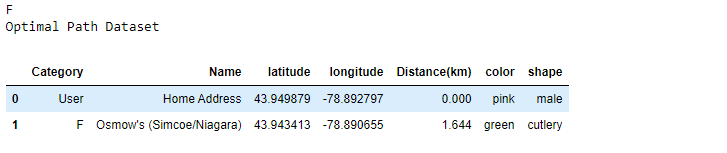
The distances are calculated iteratively depending on the current location

**Iteration 1:**

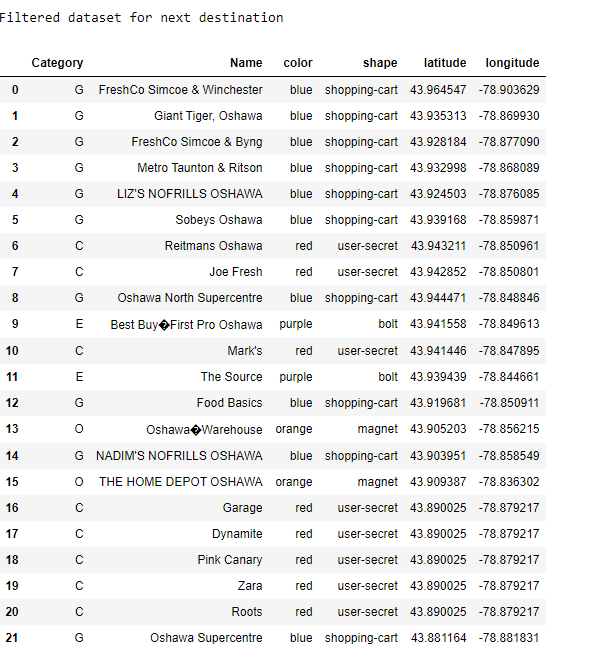
First, the distance of all the stores are calculated with respect to the current location (i.e. User address)



The shortest path is identified according to the selected category of the user and stored it in the final optimal path dataset. (Here category F(food) is nearest, so it is added to the optimal path dataset).

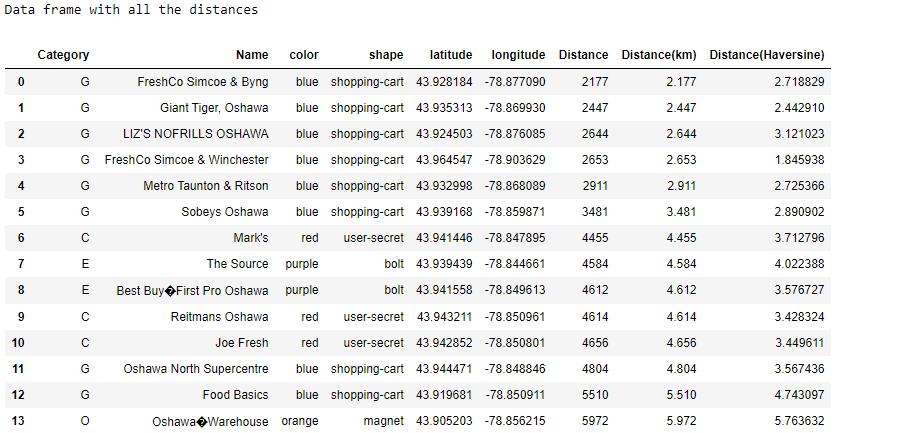


As one store of specific category is already selected and stored in the dataset, we are removing the rows with that category and create the filtered dataset. (Here, category F(food) is already stored in the optimal path dataset, therefore we are removing the data of category F(food) that can be seen in the below figure).

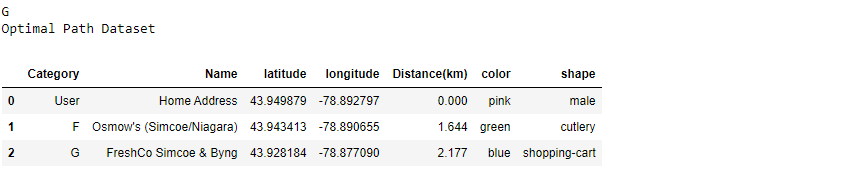


**Iteration 2:**

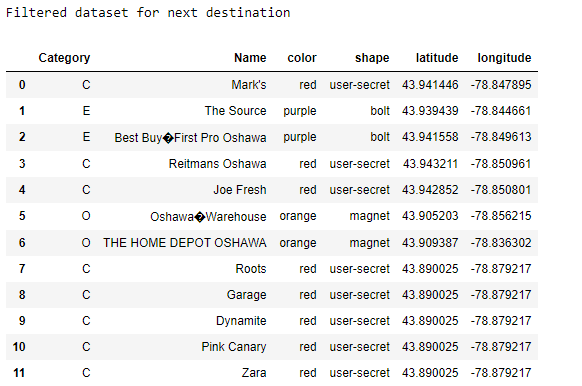
Now, the distances of all the stores are calculated with respect to the current location (i.e. store with category F which is stored in the optimal path dataset).



From the store that user visited first i.e. category F(food), the shortest path calculated is the grocery store (i.e. store with the category G (grocery)) which is then stored in the optimal path dataset.

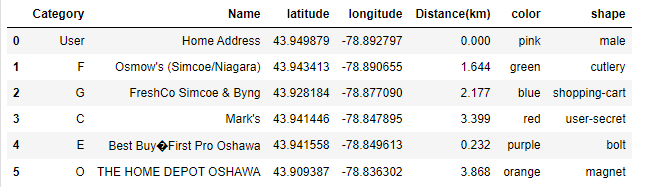


After visiting the G(grocery) store, the user would now want to visit the nearest store and the dataset now filters out Grocery store from the places to visit. As seen, the dataset now only contains C(Clothing), E(electronics) and O(Other) category.



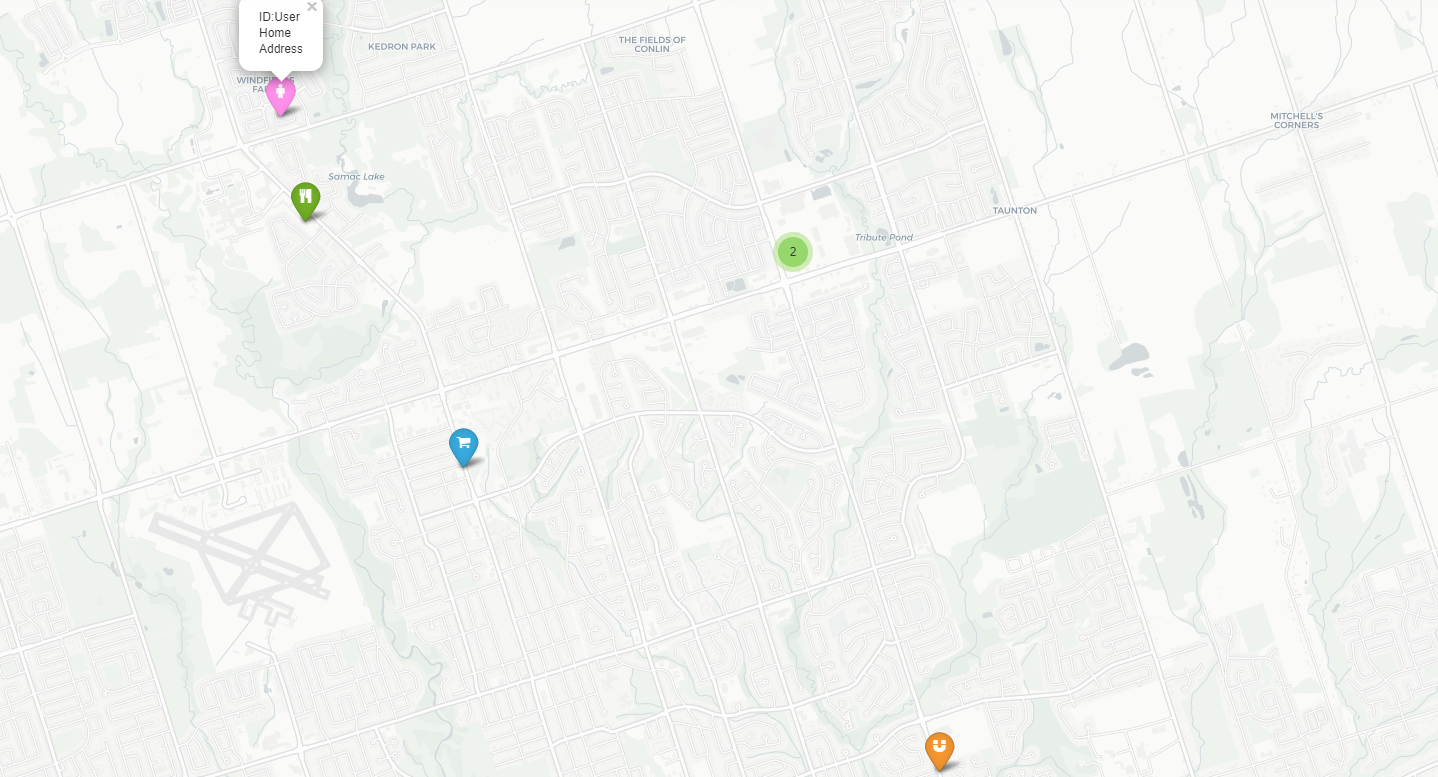
**Final Dataset**

After evaluating all the stores, the final optimal path dataset would be as follows:



## Map of all locations’ user want to visit.

The location in Optimal path dataset is plotted on map using folium plugin.



# Next Step

The next steps to be taken are as follows:

* Troubleshooting the issues if user provides incorrect input for selecting the categories.
* Work on improving the map by plotting the route between sub-destinations.
* Constructing the user interface
* Integrate user interface with backend.

# 

# Project Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Milestone** | **Submission Date** | **# of hours** | **# of Resources** |
| 1 | Submit Project proposal (including: SOW, ppt slides, github profile) | Feb 1, 2021 | 10 | 4 |
| 2 | Minimum Viable Product (MVP)  (version 1) | Feb 19, 2021 | 10 | 4 |
| 3 | MVP (version 2) | Mar 12, 2021 | 10 | 4 |
| 4 | Minimum Marketable Product (MMP) | March 26, 2021 | 10 | 4 |
| 5 | Product to launch | April 11, 2021 | 10 | 4 |
| 6 | Final Evaluation | April 16, 2021 | 10 | 4 |