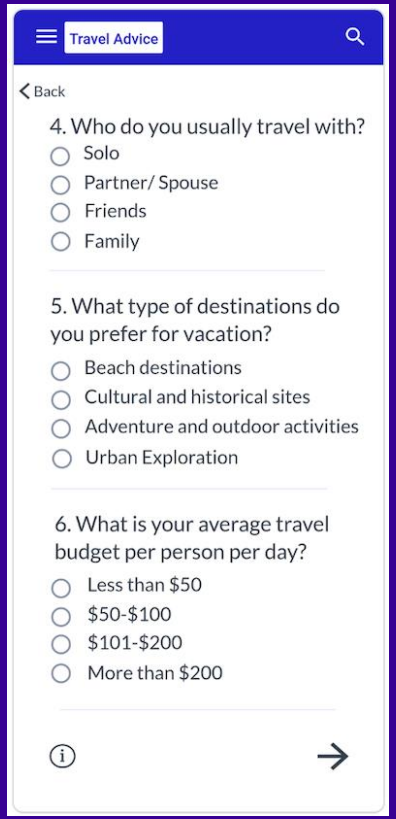
Fuzzy Sets and Systems I

Optimizing Travel Paths: Fuzzy Logic Systems For Tailored Destination Recommendations

Simon Parris
Vedasri Nakka
Michèle Fischer

University of Fribourg
22 Dec 2023

Table of Contents



Concept

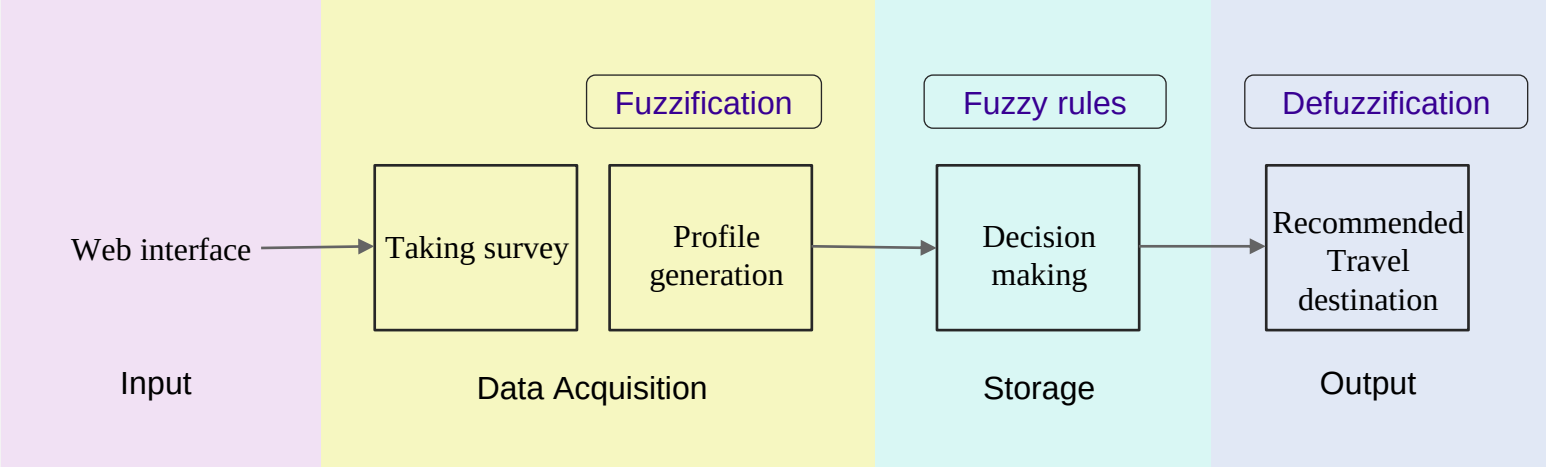
1. The main idea behind this project is to use fuzzy logic technology to recommend a customized travel destination to users by analyzing their responses.

2. The user will engage with the travel advice application and provide their travel preferences to the system.

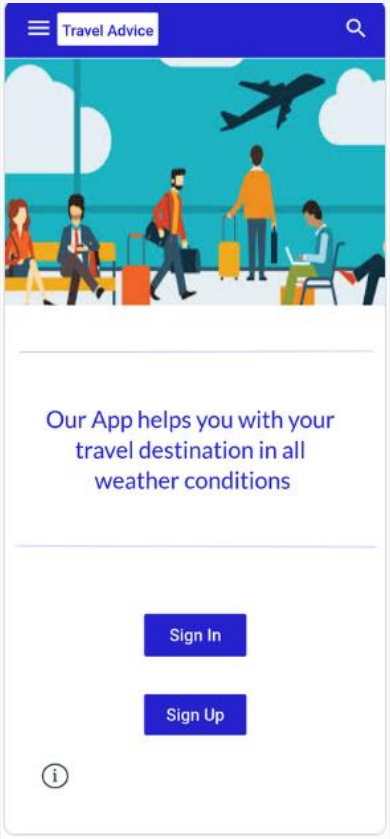
3. Based on user feedback, our designed fuzzy logic will recommend a location.

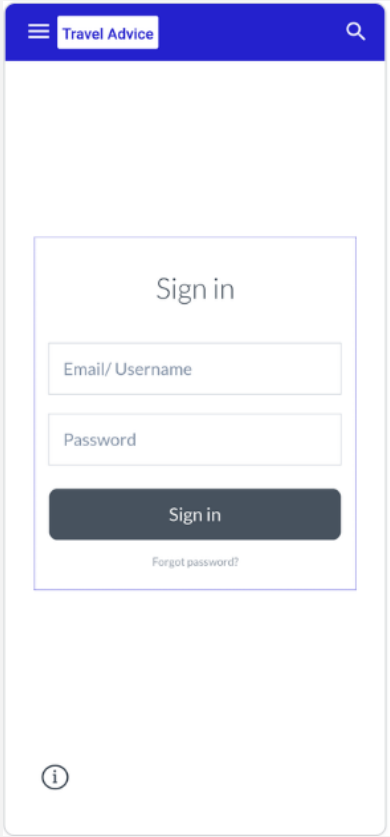
4. To make these recommendations, we will leverage accessible data on tourist trends, destination popularity, and user feedback.

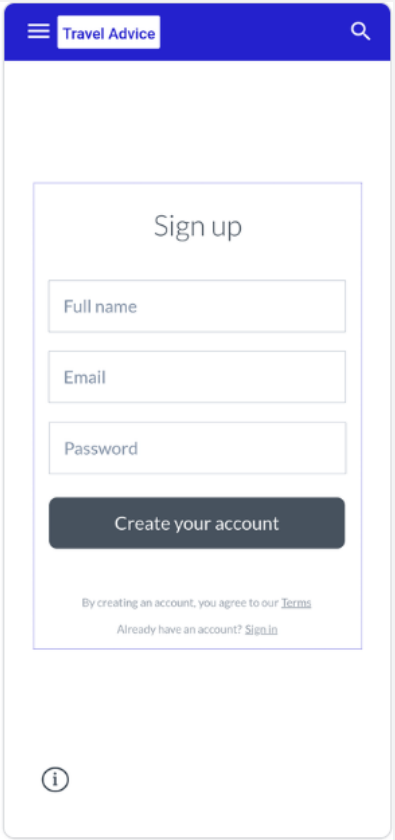
Architecture

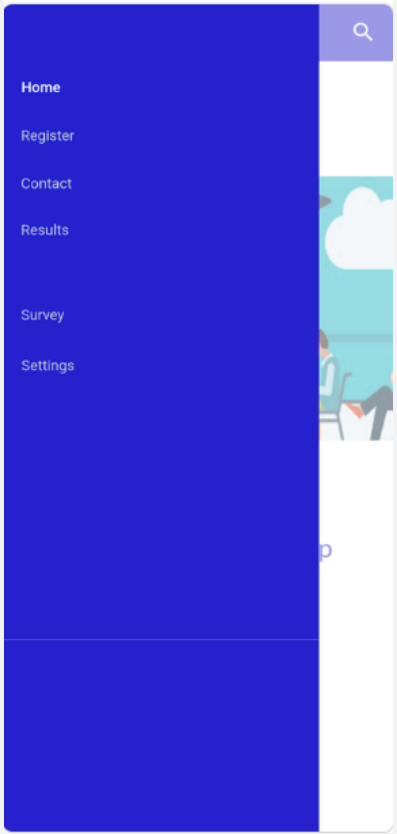


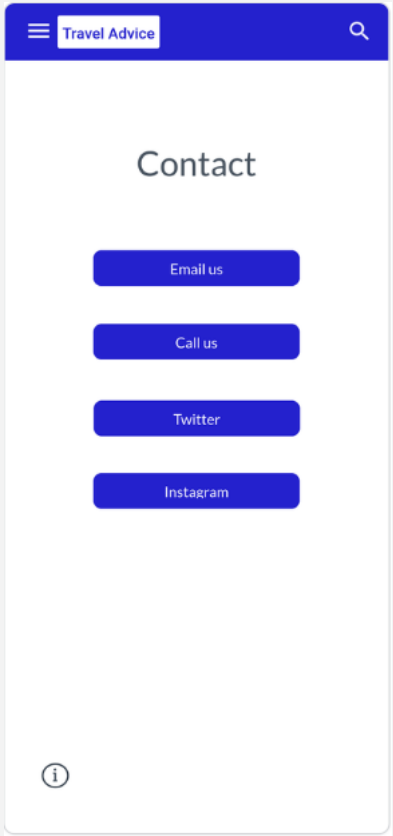
User/Fuzzy profile

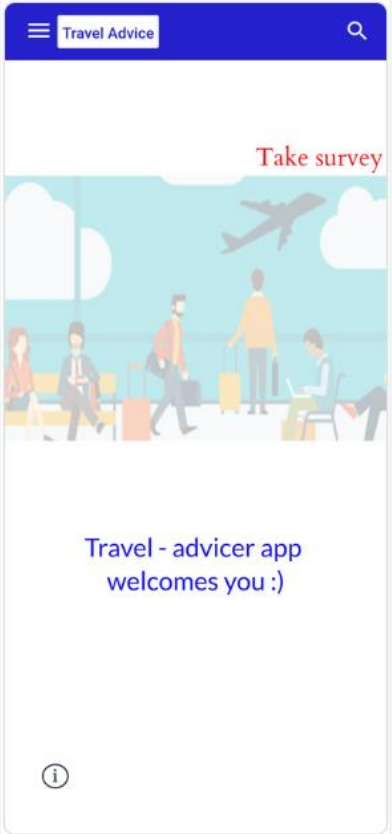

Travel Advice
Our App helps you with your travel destination in all weather conditions
Sign In
Sign Up

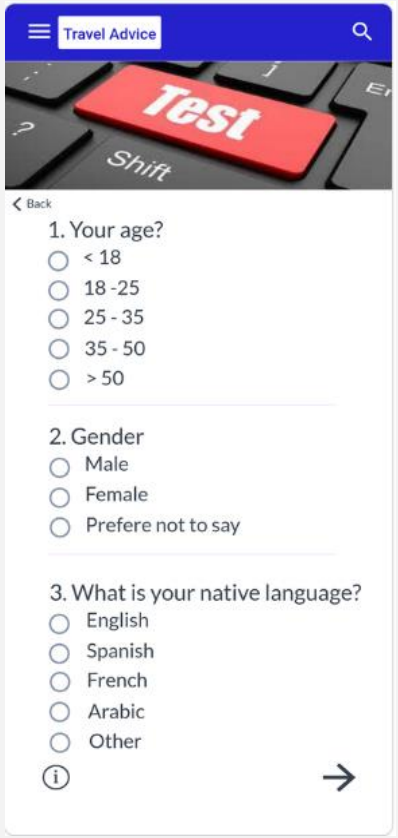

Travel Advice
Sign in
Email/ Username
Password
Sign in
Forgot password?


Travel Advice
Sign up
Full name
Email
Password
Create your account
By creating an account, you agree to our Terms
Already have an account? Sign in

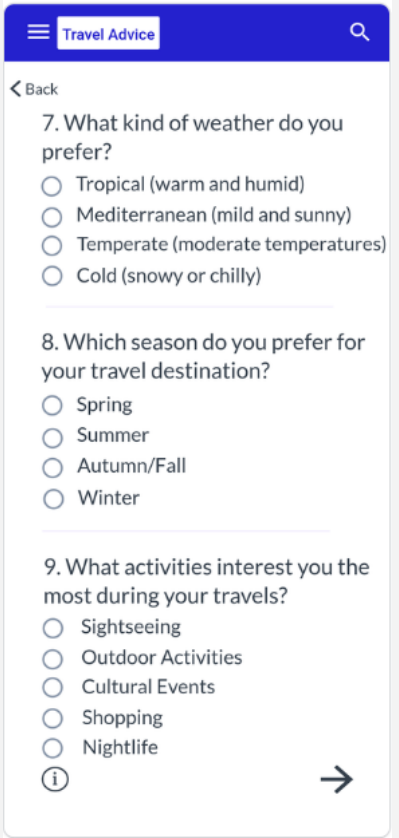

Home
Register
Contact
Results
Survey
Settings

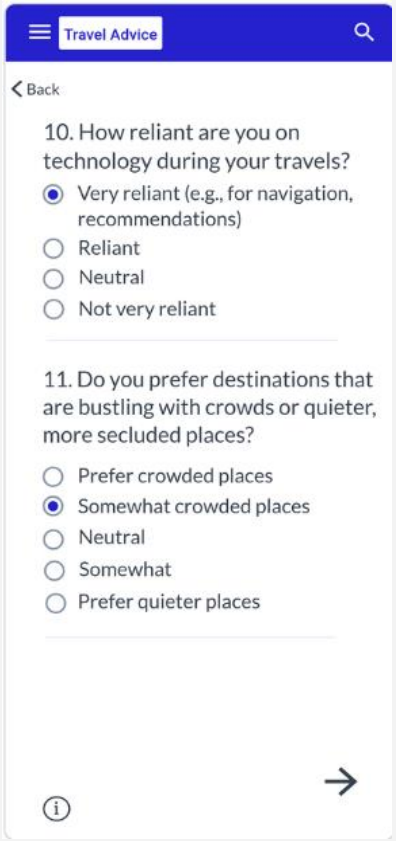

Travel Advice
Contact
Email us
Call us
Twitter
Instagram

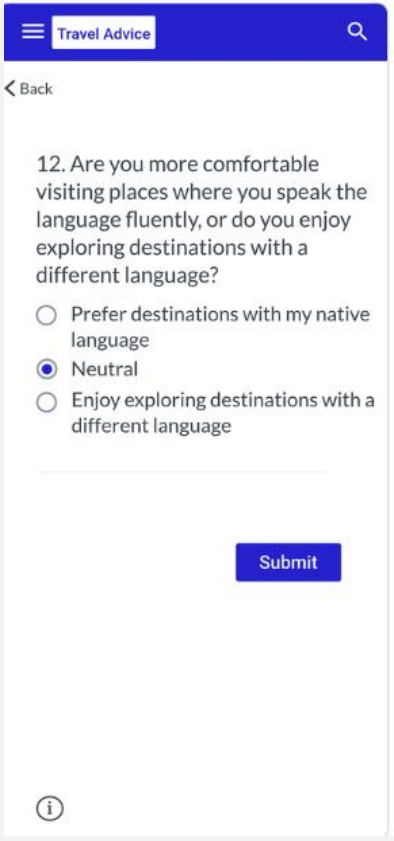

Travel Advice
Take survey
Travel - advicer app
welcomes you :)

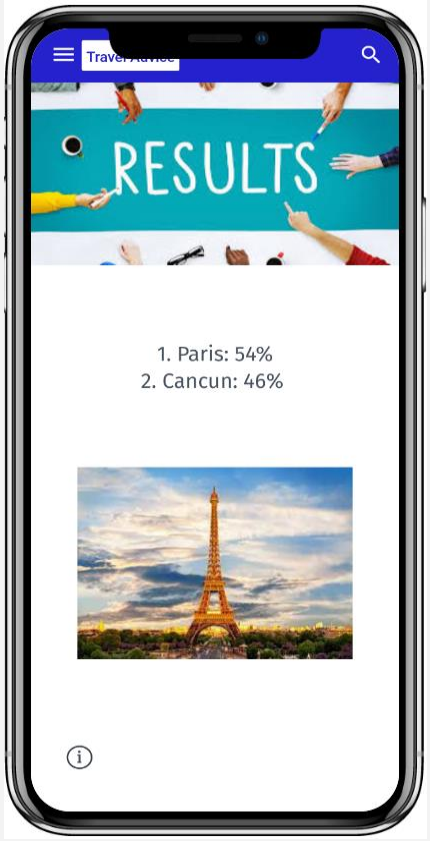

Travel Advice
Test
Shift
Back
1. Your age?
< 18
18 -25
25 - 35
35 - 50
> 50
2. Gender
Male
Female
Prefere not to say
3. What is your native language?
English
Spanish
French
Arabic
Other


Travel Advice
Back
4. Who do you usually travel with?
Solo
Partner/ Spouse
Friends
Family
5. What type of destinations do you prefer for vacation?
Beach destinations
Cultural and historical sites
Adventure and outdoor activities
Urban Exploration
6. What is your average travel budget per person per day?
Less than $50
$50-$100
$101-$200
More than $200


Travel Advice
Back
7. What kind of weather do you prefer?
Tropical (warm and humid)
Mediterranean (mild and sunny)
Temperate (moderate temperatures)
Cold (snowy or chilly)
8. Which season do you prefer for your travel destination?
Spring
Summer
Autumn/Fall
Winter
9. What activities interest you the most during your travels?
Sightseeing
Outdoor Activities
Cultural Events
Shopping
Nightlife


Travel Advice
Back
10. How reliant are you on technology during your travels?
Very reliant (e.g., for navigation, recommendations)
Reliant
Neutral
Not very reliant
11. Do you prefer destinations that are bustling with crowds or quieter, more secluded places?
Prefer crowded places
Somewhat crowded places
Neutral
Somewhat
Prefer quieter places


Travel Advice
Back
12. Are you more comfortable visiting places where you speak the language fluently, or do you enjoy exploring destinations with a different language?
Prefer destinations with my native language
Neutral
Enjoy exploring destinations with a different language
Submit


RESULTS
1. Paris: 54%
2. Cancun: 46%

Defining the features

```
age = np.arange(0,5, 1) # ['<18', '18-25', '26-35', '36-50', '>50']
gender = np.arange(0, 3, 1) # ['M', 'F', 'X']
native_language = np.arange(0, 5, 1) # ['English', 'Spanish', 'French', 'Arabic', 'X']
travel_with = np.arange(0, 4, 1) # ['solo', 'duo', 'friends', family]
destination_type = np.arange(0, 4, 1) # [beach, culture, adventure, urban]
budget = np.arange(0, 4, 1) # [<50$, 50$-100$, 101$-200$, >200$]
weather = np.arange(0, 4, 1) # [tropical, mediterranean, temperate, cold]
season = np.arange(0, 4, 1) # [spring, summer, autumn, winter]
activities = np.arange(0, 5, 1) # [sightseeing, outdoor, cultural, shopping, nightlife]
technology = np.arange(0, 5, 1) # [very reliant, reliant, neutral, not very reliant, not reliant at all]
crowd = np.arange(0, 5, 1) # [crowded,somewhat crowded, neutral, somewhat quieter, quiet]
destination_language = np.arange(0, 3, 1) # [native, neutral, different]
```

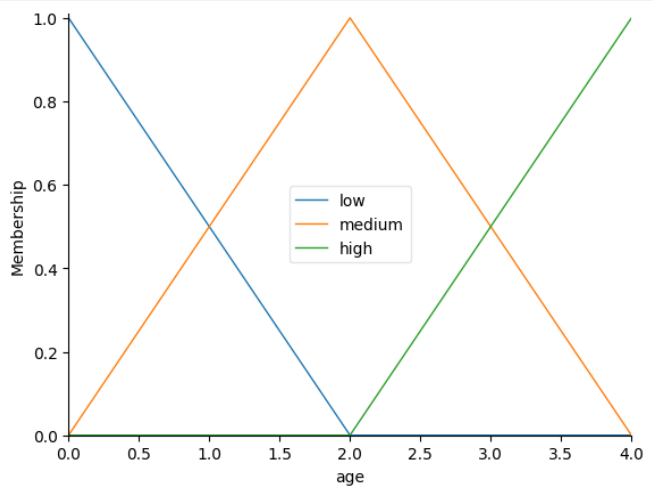
To implement fuzzy logic we used the SciKit-Fuzzy package

Defining the features

```
# Fuzzy logic variables
feature1 = ctrl.Antecedent(age, 'age')
feature2 = ctrl.Antecedent(gender, 'gender')
feature3 = ctrl.Antecedent(native_language, 'native_language')
feature4 = ctrl.Antecedent(travel_with, 'travel_with')
feature5 = ctrl.Antecedent(destination_type, 'destination_type')
feature6 = ctrl.Antecedent(budget, 'budget')
feature7 = ctrl.Antecedent(weather, 'weather')
feature8 = ctrl.Antecedent(season, 'season')
feature9 = ctrl.Antecedent(activities, 'activities')
feature10 = ctrl.Antecedent(technology, 'technology')
feature11 = ctrl.Antecedent(crowd, 'crowd')
feature12 = ctrl.Antecedent(destination_language, 'destination_language')
output = ctrl.Consequent(np.arange(0, 5, 1), 'Output') # Dubai, London, Paris, Cancun, Dominican Republic
```

Defining the Membership Functions

```python
# Define membership functions
# age
feature1['low'] = fuzz.trimf(feature1.universe, [0, 0, 2])
feature1['medium'] = fuzz.trimf(feature1.universe, [0, 2, 4])
feature1['high'] = fuzz.trimf(feature1.universe, [2, 4, 4])
# gender
feature2.automf(names=['M', 'F', 'X'])
#native_language
feature3.automf(names=['English', 'Spanish', 'French', 'Arabic', 'X'])
# travel_with
feature4.automf(names=['solo', 'duo', 'friends', 'family'])
# destination_type
feature5.automf(names=['beach', 'culture', 'adventure', 'urban'])
```



...

```python
output.automf(names=['Dubai', 'London', 'Paris', 'Cancun', 'Dominican Republic'])
```

Fuzzy rules

```
rule1 = ctrl.Rule(((feature3['Arabic'] & feature12['native']) |
                   (feature3['French'] & feature12['different']) |
                   (feature3['Spanish'] & feature12['different']) |
                   (feature3['English'] & feature12['different'])),
                   output['Dubai'], label='rule language Dubai')
rule2 = ctrl.Rule(((feature3['French'] & feature12['native']) |
                   (feature3['Arabic'] & feature12['different']) |
                   (feature3['Spanish'] & feature12['different']) |
                   (feature3['English'] & feature12['different'])),
                   output['Paris'], label='rule language Paris')
rule3 = ctrl.Rule(((feature3['Spanish'] & feature12['native']) |
                   (feature3['French'] & feature12['different']) |
                   (feature3['Arabic'] & feature12['different']) |
                   (feature3['English'] & feature12['different'])),
                   output['Cancun'], label='rule language Cancun')
rule4 = ctrl.Rule(((feature3['Spanish'] & feature12['native']) |
                   (feature3['French'] & feature12['different']) |
                   (feature3['Arabic'] & feature12['different']) |
                   (feature3['English'] & feature12['different'])),
                   output['Dominican Republic'], label='rule language Dominican Republic')
rule5 = ctrl.Rule(((feature3['English'] & feature12['native']) |
                   (feature3['French'] & feature12['different']) |
                   (feature3['Arabic'] & feature12['different']) |
                   (feature3['Arabic'] & feature12['different'])),
                   output['London'], label='rule language London')
```

Fuzzy Output

```
# Create fuzzy system
fuzzy_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9, rule10, rule11, rule12, rule13, rule14,
 rule15])
fuzzy_system = ctrl.ControlSystemSimulation(fuzzy_ctrl)
```

```
responses = [[1, 1, 2, 3, 0, 1, 2, 0, 1, 3, 1, 0],
             [0, 1, 1, 2, 1, 2, 1, 2, 0, 3, 1, 0]]
```

```
# Run the system for each set of features
for i in range(len(responses)):
    #fuzzy_system.input['age'] = responses[i][0]
    #fuzzy_system.input['gender'] = responses[i][1]
    fuzzy_system.input['native_language'] = responses[i][2]
    #fuzzy_system.input['travel_with'] = responses[i][3]
    fuzzy_system.input['destination_type'] = responses[i][4]
    fuzzy_system.input['budget'] = responses[i][5]
    fuzzy_system.input['weather'] = responses[i][6]
    fuzzy_system.input['season'] = responses[i][7]
    fuzzy_system.input['activities'] = responses[i][8]
    #fuzzy_system.input['technology'] = responses[i][9]
    #fuzzy_system.input['crowd'] = responses[i][10]
    fuzzy_system.input['destination_language'] = responses[i][11]
```

Fuzzy Output

```
    # Compute the fuzzy output
    fuzzy_system.compute()

    # Get the crisp output (defuzzification)
    fuzzy_output = fuzzy_system.output['Output']
    # Get the destinations in order
    results = []
    for name in output.terms:
        results.append(fuzz.interp_membership(output.universe, output[name].mf, fuzzy_output))
    result_destination = dict(sorted(dict(zip(destinations, results)).items(), key=lambda item: item[1], reverse=True))

    print(f"Survey Response {i + 1}: Fuzzy Output = {result_destination}")
```

```
Survey Response 1: Fuzzy Output = {'Paris': 0.5376344086021505, 'Cancun': 0.4623655913978495, 'Dubai': 0.0, 'London': 0.0, 'Dominican Republic': 0.0}
Survey Response 2: Fuzzy Output = {'Cancun': 0.7847222222222223, 'Paris': 0.21527777777777768, 'Dubai': 0.0, 'London': 0.0, 'Dominican Republic': 0.0}
```

Conclusion

- Helps travellers with limited budgets find affordable destinations.
- Makes dream trips possible for those on a tight budget.
- Introduces frequent travellers to new and exciting destinations.
- Encourages exploration beyond familiar places.
- Eases the process of finding destinations that everyone in a group will enjoy.
- Reduces the hassle of group travel planning.

Future work

- Add more rules and destinations
- User feedback Integration for continuous improvement of recommended locations.
- Using fuzzy sets to Adjust to real-time changes such as weather or events in the nearby destination.

Demo

Thank you