

**University of Isfahan**

**FOL TOUR RECOMMENDER Project**

Authors:

Melika Shirian

Kianoosh Vadaei

**Fundamentals and Applications of Artificial Intelligence**

Dr. Hossein Karshenas

2023-11-21

**Introduction**

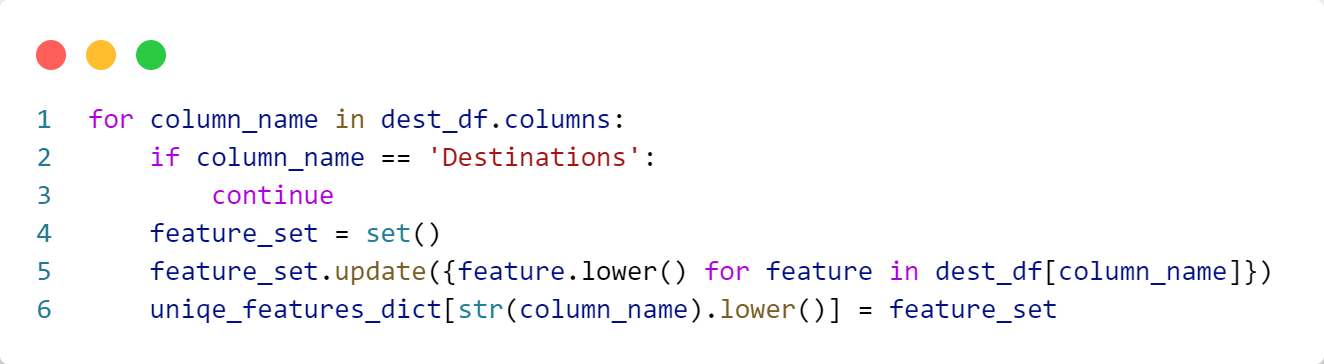
This project explores the combination of Prolog and Python to build a Recommender System for Tour Trips, complete with an interactive map display using APIs. By blending Prolog's logic with Python's practicality, the aim is to create an effective system for personalized travel recommendations. The report will delve into the theoretical and collaborative aspects of this integration, shedding light on its real-world implications in travel planning.

**Code Explanation**

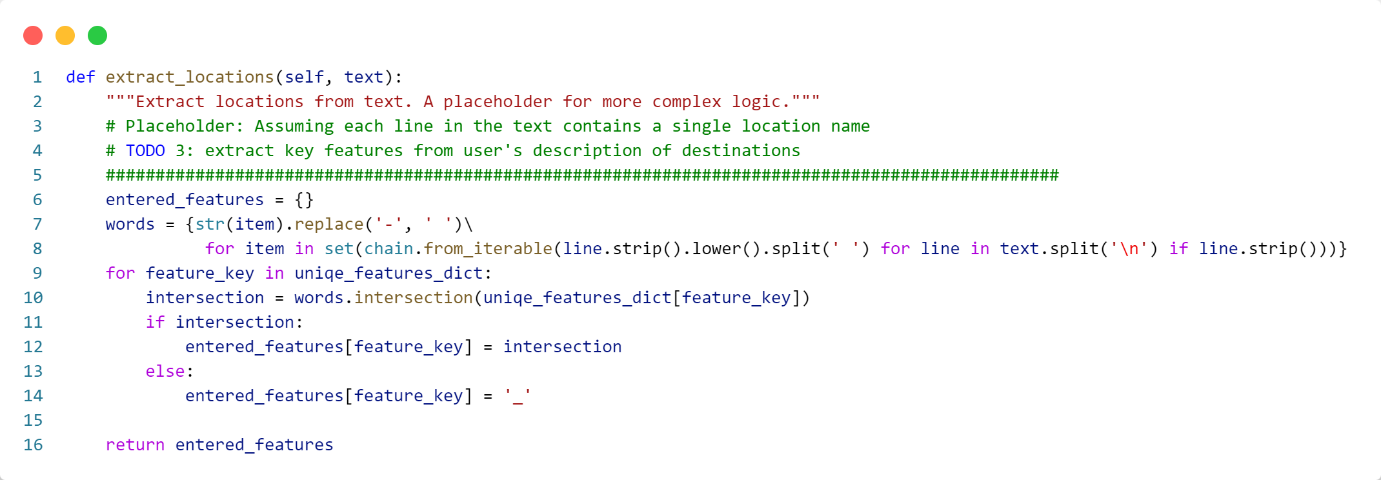
In this project, there exists a file containing destinations along with their respective features.



Initially, the relevant file is read using the Pandas library and stored in a dataframe. Subsequently, Prolog is instantiated, and the entire initial knowledge base is cleared. Following this, each row of the dataset is read and stored in the database.



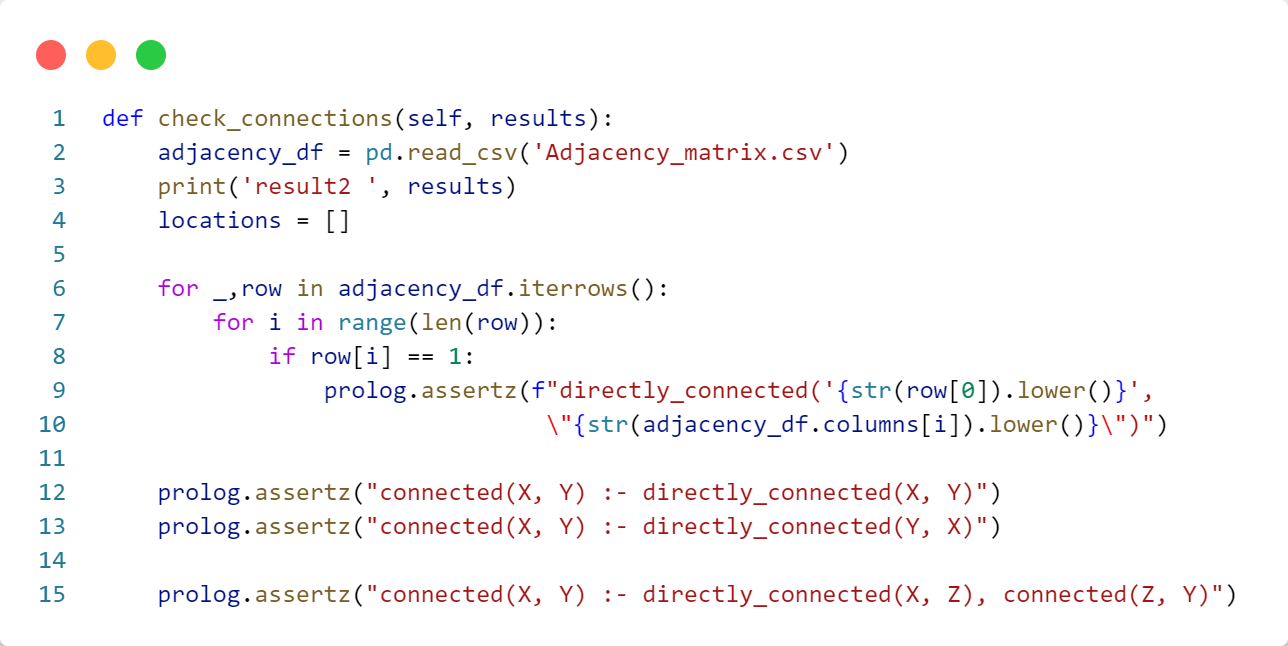
Subsequently, we determine the unique values for each feature and store them in a dictionary. In this dictionary, the keys represent the feature names, and the values are sets containing unique values for each feature.



After the knowledge base is formed and unique values of features are extracted, the next step involves processing user input. The mentioned function maintains all words from the user's text in a set. Subsequently, utilizing the dictionary of unique feature values, it identifies which values of each feature are present in the text and stores them in a new dictionary. It's worth noting that if a feature has no corresponding values in the text, then '\_' is retained in the dictionary.



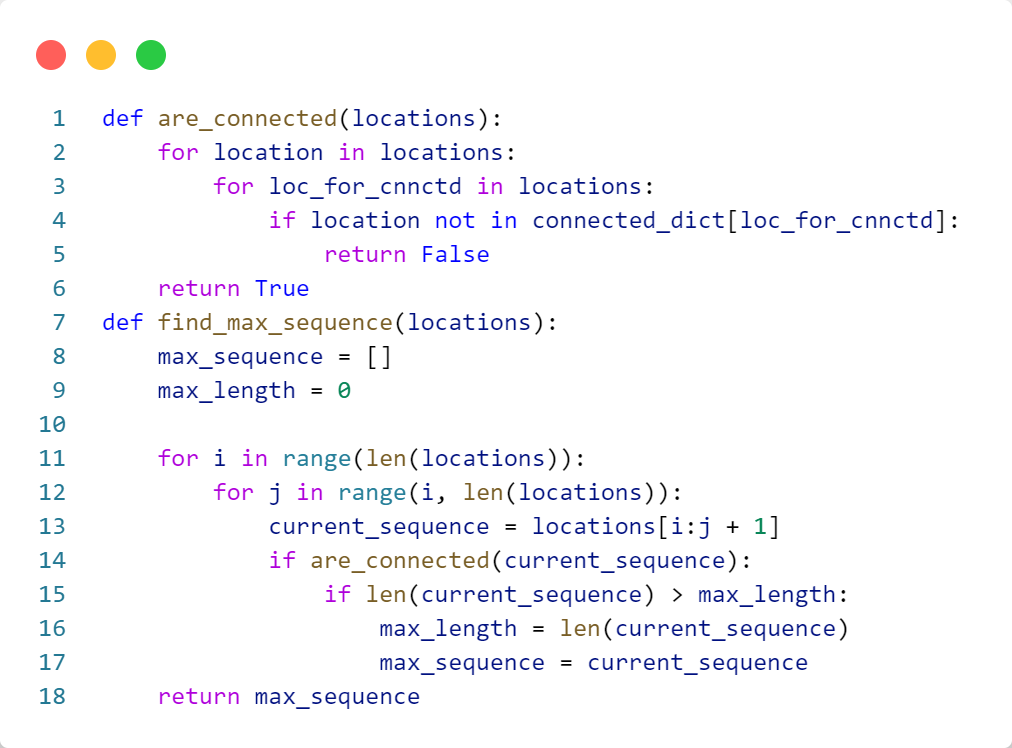
Now, in this function, based on the features extracted from the user's input text, we formulate a query to the knowledge base. This query retrieves destinations with the specified features and returns them in a list. It's noteworthy that given the possibility of multiple values for each feature, logical 'and' operators are used between features, and logical 'or' operators are used between different values of each feature in the query.



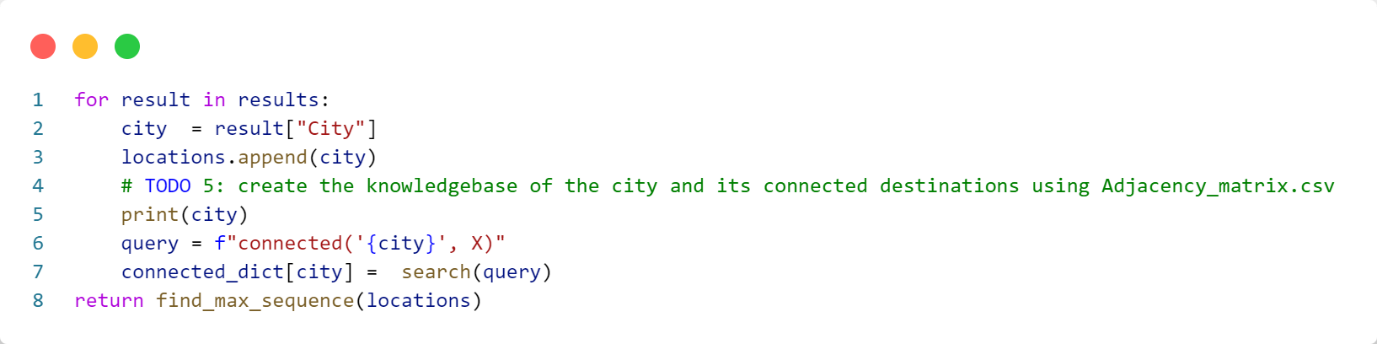
In this section, we initially construct the knowledge base based on a dataset containing an adjacency matrix, considering up to level 2 connections. By doing so, if two cities are connected, they will be bidirectionally connected in the knowledge base.



In this function, the formed query based on a city, which returns connected cities up to a maximum of 2 degrees, is sent. The cities connected to a particular city are stored one by one in a dictionary with the city itself as the key.



These two functions first validate the connection between two cities and then return a sequence of the largest connected cities.



Finally, the ultimate dictionary is maintained here in the described manner, and the sequence of connected cities with the maximum length is returned in a list.



In the end, the condition for the number of cities is examined, their connections are verified, and the final list is displayed on the map.

**References**

* OpenAI. "ChatGPT." <https://www.openai.com/>
* Russell, Stuart, and Norvig, Peter. "Artificial Intelligence: A Modern Approach." (Book)