Vladimir Sokolov
Github Link - https://github.com/VladimirASokolov/CS6320Project
Youtube Link - https://youtu.be/mhpy-Hnl1LQ

Task - Selecting a place to eat is a hassle so the task is to create a chatbot to help assist with recommending places to eat.

Approach - The approach was to leverage the Google Maps places API to find restaurants in an area. The main components of a search are the latitude and longitude coordinates, searching radius, as well as type tags that Google assigns to establishments. Using spaCy and regex the program extracts the intent of a user input to make changes to the search parameters. To find coordinates to center a search on the user can request a location which uses Google's Text Search in the Places API. When modifying type tags to include or exclude, difflib package's close_matches helps match a user's input into the list of tags that the Nearby Search API recognizes.

Creativity - Originally the program had its input and output through the terminal but I added a simple gui using the Tkinter python package. Tkinter did the job but was bright white and had an interface that transported the user back multiple decades. I replaced it with Ttkbootstrap which had built in themes including dark ones to choose from as well as added extra information on the side to show the user the different pools of types.

Complexity - The main goal that I had from the beginning was to integrate the Google Maps API into the project as its main data source. Nearby Search from the Places API was the best fit but had limitations and wasn't exactly what I had in mind at the start. Originally I wanted to use Google Maps to gather all of the restaurants within a large area then use the gathered data to train a model to more organically recommend places to go to. The main problem is that it's limited to a couple dozen results at a time so it would take an extra program that calls search on small patches of area in a grid pattern to build up the data which would quickly max out the free tier of the API. Instead I opted

to use the API in its intended way and focus on how I could interpret a user's input to best use it.

Lessons learned and potential improvements - This was my first time using a public facing API and not a simple one created for a tutorial project. Learning about it was more difficult than expected and just finding out what is free or not was time consuming. The program uses less NLP concepts than I would have liked due to how the API works which was disappointing to me. If I had more time I would have definitely tried to improve the intent recognition and replaced the regex with something way more robust. One annoying quirk of the current design is that it needs a detailed pass on the place types blacklist as currently it includes a lot of places Google tags as restaurants that intuitively shouldn't be such as movie theatres and gas stations.