ProjectNotebook

December 5, 2021

1 Project Description

My project is a chatbot where a student (Preferrably UCSD Students) can type in what type of food they are craving (Given the options in the statements) and the chatbot will respond with resturant options on campus, and then for sense of humor, give them off campus options too. If I had more time to complete this project, I would have made my code in my test cell shorter by saving my coding cell, naming it, and then import it into the test cell and then do my asserts. This would have made my test cell easier to read and more concise and neat like mentioned we should try to do in lecture. I would have also made my chatbot more interactive/complex by having it respond to different things rather than having only being able to respond to a few basic inputs. I would have achieved this by including a dictionary of phrases or words the bot can respond back with and importing a random library so I can use the random to pick randomly from the phrases and the bot will respond back with different things. I was also thinking of including a relation fucntion (Similar to the one used in the video "building a chatbot in python for beginners"listed in my sources list) so that the user would be able to type in a mispelling or a shorter phrase and the bot will still understand what it is trying to ask and give the specfific response (Almost feeling as if it is talking to a real person).

1.1 Project Code

If it makes sense for your project, you can have code and outputs here in the notebook as well.

[1]:
```
#from my_module.functions import ...
#from my_module.classes import ...
```

[2]:
```
# Do a bunch of things.

#Overall list of resources I used to help me run this project: A3, class␣
 ↪lecture from 12/1, lecture 11/12
#Website resources I used: https://www.tutorialkart.com/python/
 ↪python-while-loop/python-while-loop-break/
#Website resoirces I used: https://www.tutorialsteacher.com/python/
 ↪python-if-elif
#Website resources I used: https://towardsdatascience.com/
 ↪building-a-chatbot-in-python-the-beginners-guide-2743ad2b4851
#Wesbite resources I used:https://www.w3schools.com/python/ref_string_lower.asp
#Website resources I used: https://realpython.com/python-testing/
```

```
#I imported the string to allow the chatbot to run. It is necessary becasue a␣
 ↪lot of my chatbot is dealing with strings.
#The print statement was made to ask the question once the bot runs.
import string


print("Hi! I am your friendly chatbot here to help you pick a restaurant to eat␣
 ↪at based on your cravings. Finals are rough, so be sure to spoil yourself!")


def foodie_suggestions():
    " " "This function is the main part of the chatbot that allows it to run␣
 ↪and is what circles through the statements based on your inputs" " "
    keepgoing = True
    while keepgoing:
        selection = input("Enter a type of craving,S for sweets, Ve for␣
 ↪Vegetarian, H for health, F for fast food: ")
        craving = parseSelection(selection)
        if craving == "unknown":
            print("Look at the options and think about what you are craving!")
        suggestion = giveASuggestion(craving)
        print(suggestion)
        kg = input("Craving something else? Yes/No: ")
        if kg == "No":
            break
        offCampusFood = input("Want some ACTUAL good food, from offcampus? Y/N")
        if(offCampusFood == "Y"):
            selection = input("Enter a type of craving,S for sweets, Ve for␣
 ↪Vegetarian, H for health, F for fast food: ")
            craving = parseSelection(selection)
            if craving == "unknown":
                print("Look at the options and think about what you are craving!
 ↪")
            suggestion = offCampus(craving)
            print(suggestion)

#this function is what allows the chatbot to know what it should return based␣
 ↪on what the user inputs
#This helps the functions below it be allowed to actually reply with the print␣
 ↪statements, which makes the chatbot answer the user
def parseSelection(txtIn):
    sweets = "s"
    vegetarian = "ve"
    fastFood = "f"
    health = "h"
    if txtIn.lower() == sweets:
```

```
        return "sweets"
    elif txtIn.lower() == vegetarian:
        return "vegetarian"
    elif txtIn.lower() == fastFood:
        return "fastfood"
    elif txtIn.lower() == health:
        return "health"
    else:
        return "unknown"

#The two functions below are what allow for the chatbot to seem like it is␣
 ↪answering the questions.
#With the if and elif statements, it returns a string if the corresponding␣
 ↪statement that matches based on the first two functions.
def giveASuggestion(txtIn):
    sweets = "sweets"
    vegetarian = "vegetarian"
    fastFood = "fastfood"
    health = "health"
    if txtIn.lower() == sweets:
        return "Yogurt World, Starbucks, Burger King"
    elif txtIn.lower() == vegetarian:
        return "Jamba, Lemongrass Farm Fresh Plates,Seed + Sprout,␣
 ↪Croutons,Blue Pepper Asian Cuisine"
    elif txtIn.lower() == fastFood:
        return "Burger King, Dirty Birds, Panda Express, Rubios Coastal Grill,␣
 ↪Santorini Greek Island Grill, Subway, Tapioca Express, Taco Villa"
    elif txtIn.lower() == health:
        return "Jamba, Lemongrass Farm Fresh Plates, Seed + Sprout, Subway,␣
 ↪Blue Pepper Asian Cuisine"

def offCampus(txtIn):
    sweets = "sweets"
    vegetarian = "vegetarian"
    fastFood = "fastfood"
    health = "health"
    if txtIn.lower() == sweets:
        return "85c in Mira Mesa"
    elif txtIn.lower() == vegetarian:
        return "Evolution Fast Food or Loving Hut"
    elif txtIn.lower() == fastFood:
        return "Popeyes"
    elif txtIn.lower() == health:
        return "Whole Foods"

#this below is what is necssary for the whole chatbot to run because to is␣
 ↪calling the first function.
```

```
foodie_suggestions()
```

```
Hi! I am your friendly chatbot here to help you pick a restaurant to eat at
based on your cravings. Finals are rough, so be sure to spoil yourself!
Enter a type of craving,S for sweets, Ve for Vegetarian, H for health, F for
fast food: No
Look at the options and think about what you are craving!
None
Craving something else? Yes/No: No
```

[3]:
```
# test it out"

#I was honestly unsure how to make a test cell for my first function, but I did␣
 ↪them for the rest.
#I copied and pasted the last 3 functions and put them on to here so that the␣
 ↪test cell runs.
def parseSelection(txtIn):
    sweets = "s"
    vegetarian = "ve"
    fastFood = "f"
    health = "h"
    if txtIn.lower() == sweets:
        return "sweets"
    elif txtIn.lower() == vegetarian:
        return "vegetarian"
    elif txtIn.lower() == fastFood:
        return "fastfood"
    elif txtIn.lower() == health:
        return "health"
    else:
        return "unknown"

def giveASuggestion(txtIn):
    sweets = "sweets"
    vegetarian = "vegetarian"
    fastFood = "fastfood"
    health = "health"
    if txtIn.lower() == sweets:
        return "Yogurt World, Starbucks, Burger King"
    elif txtIn.lower() == vegetarian:
        return "Jamba, Lemongrass Farm Fresh Plates,Seed + Sprout,␣
 ↪Croutons,Blue Pepper Asian Cuisine"
    elif txtIn.lower() == fastFood:
        return "Burger King, Dirty Birds, Panda Express, Rubios Coastal Grill,␣
 ↪Santorini Greek Island Grill, Subway, Tapioca Express, Taco Villa"
    elif txtIn.lower() == health:
```

```
        return "Jamba, Lemongrass Farm Fresh Plates, Seed + Sprout, Subway,␣
 ↪Blue Pepper Asian Cuisine"

def offCampus(txtIn):
    sweets = "sweets"
    vegetarian = "vegetarian"
    fastFood = "fastfood"
    health = "health"
    if txtIn.lower() == sweets:
        return "85c in Mira Mesa"
    elif txtIn.lower() == vegetarian:
        return "Evolution Fast Food or Loving Hut"
    elif txtIn.lower() == fastFood:
        return "Popeyes"
    elif txtIn.lower() == health:
        return "Whole Foods"


#below are my tests. I defined them as test functions and put assert statements␣
 ↪as mentioned in lecture.
#The assert statements have the funtion first, what the call is and what should␣
 ↪come out of that calling, and the statement
#that will result in that calling being done.
def test_parseselection():
    sweets = "s"
    vegetarian = "ve"
    fastFood = "f"
    health = "h"
    assert parseSelection(sweets) == "sweets"
    assert parseSelection(vegetarian) == "vegetarian"
    assert parseSelection(fastFood) == "fastfood"
    assert parseSelection(health) =="health"
def test_giveASuggestion():
    sweets = "sweets"
    vegetarian = "vegetarian"
    fastFood = "fastfood"
    health = "health"
    assert giveASuggestion(sweets) == "Yogurt World, Starbucks, Burger King"
    assert giveASuggestion(vegetarian) == "Jamba, Lemongrass Farm Fresh␣
 ↪Plates,Seed + Sprout, Croutons,Blue Pepper Asian Cuisine"
    assert giveASuggestion(fastFood) == "Burger King, Dirty Birds, Panda␣
 ↪Express, Rubios Coastal Grill, Santorini Greek Island Grill, Subway, Tapioca␣
 ↪Express, Taco Villa"
    assert giveASuggestion(health) == "Jamba, Lemongrass Farm Fresh Plates,␣
 ↪Seed + Sprout, Subway, Blue Pepper Asian Cuisine"
```

```
def test_offCampus():
    sweets = "sweets"
    vegetarian = "vegetarian"
    fastFood = "fastfood"
    health = "health"
    assert offCampus(sweets) == "85c in Mira Mesa"
    assert offCampus(vegetarian) == "Evolution Fast Food or Loving Hut"
    assert offCampus(fastFood) == "Popeyes"
    assert offCampus(health) == "Whole Foods"

#similar to the statement I put at the very bottom of the cell above, I need␣
 ↪these in order to run the test cells
#and see if they are working.
test_parseselection()
test_giveASuggestion()
test_offCampus()
```

[ ]:

Extra Credit (*optional*)

1. Your Python Background: I have had no python experience before taking this class and barely even knew what coding was or what it was used for.
2. How your project went above and beyond the requirements of the project and/or how you challenged yourself to learn something new with the final project: I have had no prior python experience which has caused me to struggle with this class. I have gotten low scores on a lot of assignments but have pushed through the course thus far and was willing to try to learn more python by taking on this project rather than just taking the exam. I think my ability to do this project has shown massive improvement upon myself and my abilities. This project took me hours and I have never been so stressed out in my life until I started doing this project. Although this chatbot is simple, I think I am able to show that I can still make somethign work in python and just needed to be given a lot of time, use of notes, and asking for help when needed.

[ ]: