

# CMPT 120 LECTURE 5-1

Prof. Nick Vincent  
Simon Fraser University

2023-10-04

# QUICK NOTE FOR THOSE READING SLIDES

There's only one set of slides for week 5 (Monday was a holiday).

# 5-1 AGENDA

- Quick set of review questions
- Quick review on accumulator pattern
- Completing our recommender system example
  - Putting together for loops, nested for loops, list slicing, string slicing

# REVIEW QUESTIONS

# QUESTION 1A

What does this snippet output?

```
1 response = "I LOVE RAIN!!"  
2 words = response.lower().strip("!").split(" ")  
3 if "rain" in words or "umbrella" in words:  
4     print("You're a fan of the rain?")
```

# QUESTION 1A

You're a fan of the rain?

# QUESTION 1B

```
1 response = "I LOVE RAIN!!"
2 try:
3     words = response.lower().split(" ").strip("!")
4     if "rain" in words or "umbrella" in words:
5         print("You're a fan of the rain?")
6 except AttributeError:
7     print("Something went wrong")
```

# QUESTION 1B

Something went wrong



# QUESTION 2A

What does this output?

```
1 fruits = "durian, rambutan, lychee"  
2 fruit_list = fruits.split(",")  
3 print("durian" in fruit_list)  
4 print("rambutan" in fruit_list)
```

# QUESTION 2A

True

False

# QUESTION 2B

What does this output?

```
1 fruits = "durian,rambutan,lychee".split(",")  
2 print(fruits[1][0].upper())
```

# QUESTION 2B

R

# QUESTION 3

Find three errors with this code:

```
1 z = int(input["Give me a number, any number: "])
2 if z > 5 and <= 10:
3     print(x)
```

# QUESTION 3 – ANSWER

```
1 z = int(input("Give me a number, any number: "))
2 if z > 5 and z <= 10:
3     print(z)
```

# QUESTION 4

What does this snippet output?

```
1 letters = ['a', 'b', 'c', 'd', 'e', 'f']
2 print(len(letters))
3
4 print(letters[1:3])
5 print(letters[:4])
6 print(letters[3:])
7 print(letters[:])
8 print(letters[-1])
9 print(letters[3:-1])
```

# QUESTION 4

6

`['b', 'c']`

`['a', 'b', 'c', 'd']`

`['d', 'e', 'f']`

`['a', 'b', 'c', 'd', 'e', 'f']`

f

`['d', 'e']`



# TRICK TO REMEMBER SLICING

`letters[:4]` will give you 4 total items in your output! It must start at 0... hence you get 0,1,2, and 3.

# ACCESSING CHARACTERS IN STRINGS

- Just like accessing items in a list
- If you think of strings as a list of characters, you're good to go
- (As long as you know how lists work)

# ACCUMULATOR REVIEW

“Go through all the items in a pile and pick out the red ones”

```
1 items = ["red", "blue", "red", "green"]
2 items_we_grab = []
3 for item in items:
4     if item == "red":
5         items_we_grab += [item]
6 print(items_we_grab)
```

```
['red', 'red']
```

# WEEK 5 LEARNING OBJECTIVES

# OPEN + READ LINES FROM A FILE

```
1 with open('fake_data.csv', 'r') as file:  
2     lines = file.readlines()  
3     print(len(lines))
```

51

# SPLIT A STRING INTO A LIST

```
1 text = "Python is fun"  
2 words = text.split() # ['Python', 'is', 'fun']  
3 print(words)
```

```
['Python', 'is', 'fun']
```

# ACCESS SPECIFIC ELEMENTS OF A LIST USING INDEXING/SLICING

```
1 numbers = [1, 2, 3, 4, 5]
2 first_element = numbers[0]      # 1
3 sliced_elements = numbers[1:4]  # [2, 3, 4]
```

# ACCESS SPECIFIC CHARACTER IN A STRING USING INDEXING/SLICING

```
1 text = "Hello World"
2 first_char = text[0] # 'H'
3 sliced_chars = text[6:11] # 'World'
```



# PERFORM COMPARISONS BETWEEN NUMBERS, TAKING INTO ACCOUNT ORDER OF OPERATORS (OPERATOR PRECEDENCE)

```
1 result = 2 + 3 * 4 > 10 # Evaluates to 2 + 12 > 10 => 14 > 10 => True
2 print(result)
```

True

# PERFORM COMPARISONS (E.G. !=, <, >) WITH STRINGS

```
1 string1 = "apple"
2 string2 = "banana"
3 print(string1 != string2)    # True
4 print(string1 < string2)    # True, because 'a' comes before 'b' alphabetically
```

True

True

# INTERPRET CODE WITH NESTED CONDITIONALS + COMPARISON OPERATORS (E.G. !=, <, >=)

```
1 x = 5
2 y = 10
3 if x != y:
4     if x < y:
5         print("x is less than y")
6     else:
7         print("x is greater than y")
8 else:
9     print("x is equal to y")
```

x is less than y

# FIND THE COMMON ELEMENTS BETWEEN 2 LISTS

```
1 list1 = [1, 2, 3, 4]
2 list2 = [3, 4, 5, 6]
3 common_elements = []
4 for x in list1:
5     if x in list2:
6         common_elements.append(x)
7 print(common_elements) # [3, 4]
```

[3, 4]

# UNDERSTAND + USE NESTED FOR LOOPS

```
1 for i in range(3):  
2     for j in range(2):  
3         print(i, j)
```

```
0 0  
0 1  
1 0  
1 1  
2 0  
2 1
```

# APPLY OPERATOR ORDER TO EVALUATE EXPRESSIONS

```
1 result = 3 + 4 * 2 / (1 - 5) # Evaluates using operator precedence: multip
2 print(result)
```

1.0

# CONCATENATE LISTS

```
1 list1 = [1, 2, 3]
2 list2 = [4, 5, 6]
3 concatenated_list = list1 + list2
4 print(concatenated_list) # [1, 2, 3, 4, 5, 6]
```

```
[1, 2, 3, 4, 5, 6]
```

# APPLY ACCUMULATION PATTERN FOR STRINGS AND LISTS (PREVIOUSLY WAS NUMBERS)

```
1 # For strings
2 words = ["Hello", "World"]
3 sentence = ""
4 for word in words:
5     sentence += word + " "
6 print(sentence.strip()) # Hello World
7
8 # For lists
9 matrix = [[1, 2], [3, 4]]
10 flat_list = []
11 for sublist in matrix:
12     for item in sublist:
13         flat_list.append(item)
14 print(flat_list) # [1, 2, 3, 4]
```

```
Hello World
[1, 2, 3, 4]
```



# CALCULATE THE MAXIMUM AMONG SEVERAL VALUES

```
1 values = [3, 5, 2, 8, 6]
2 maximum = values[0]
3 for value in values:
4     if value > maximum:
5         maximum = value
6 print(maximum) # 8
```

8

# OK, BACK TO CONTENT

- We want to extend our recommender systems example

# LET'S RECAP...

```
1 with open("example.csv", "r") as file:
2     # Skip header
3     header = file.readline()
4
5     # Process and print each record
6     for line in file:
7         columns = line.strip().split(",")
8         nice_output = ""
9         for column in columns:
10             nice_output += column + " | "
11         print(nice_output)
```

# GENERATING DATA (ADVANCED)

```

1 import csv
2 import random
3
4 # Lists of names and diets
5 names = ["Michael Jordan", "Tom Hanks", "Arya Stark", "Goku", "Scarlett Johansson",
6          "Mickey Mouse", "Hermione Granger", "Naruto", "Serena Williams", "Leonardo DiCaprio",
7          "Mario", "Taylor Swift", "Darth Vader", "James Cameron", "Usain Bolt", "Tony Stark",
8          "Bugs Bunny", "The Rock", "Walter White", "Sheldon Cooper", "Mulan", "Elsa",
9          "Sherlock Holmes", "Jack Sparrow", "Luffy", "Luke Skywalker", "Marilyn Monroe",
10         "Freddie Mercury", "Lord of the Rings: The Fellowship of the Ring", "Superman",
11         "Wonder Woman", "Neo", "Beyonce", "Ash Ketchum", "Lara Croft", "Elvis Presley",
12         "Kermit the Frog", "Spider-Man", "Indiana Jones", "Simba", "Rihanna", "Dexter Morgan",
13         "Morpheus", "Wolverine", "Cinderella", "Captain America", "Thor", "Katniss Everdeen"]
14
15 movies_by_genre = {
16     "Drama": ["The Shawshank Redemption", "Forrest Gump", "The Godfather", "The Godfather Part II",
17              "The Silence of the Lambs", "Schindler's List", "The Usual Suspects", "Good Will Hunting",
18              "The Green Mile", "The Departed", "Inglourious Basterds", "The Prestige", "The Incredibles"],
19     "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jurassic Park", "The Matrix",
20               "Die Hard", "John Wick", "Mission: Impossible - Fallout", "Gladiator", "The Expendables",
21               "The Fast and the Furious", "The Hitman's Bodyguard", "The Hitman's Wife's Bodyguard", "The Hitman's Regretful Wife",
22               "The Hitman's Unlucky Wife", "The Hitman's Annoyed Wife", "The Hitman's Disappointed Wife",
23               "The Hitman's Frustrated Wife", "The Hitman's Irritated Wife", "The Hitman's Miffed Wife",
24               "The Hitman's Pissed Wife", "The Hitman's Riled Wife", "The Hitman's Ticked Wife",
25               "The Hitman's Tired Wife", "The Hitman's Upset Wife", "The Hitman's Vexed Wife",
26               "The Hitman's Winded Wife", "The Hitman's Exhausted Wife", "The Hitman's Spent Wife",
27               "The Hitman's Defeated Wife", "The Hitman's Dejected Wife", "The Hitman's Downcast Wife",
28               "The Hitman's Discouraged Wife", "The Hitman's Disheartened Wife", "The Hitman's Dispirited Wife",
29               "The Hitman's Dismayed Wife", "The Hitman's Disturbed Wife", "The Hitman's Troubled Wife",
30               "The Hitman's Aggrieved Wife", "The Hitman's Annoyed Wife", "The Hitman's Angry Wife",
31               "The Hitman's Fuming Wife", "The Hitman's Furious Wife", "The Hitman's Incensed Wife",
32               "The Hitman's Indignant Wife", "The Hitman's Irate Wife", "The Hitman's Irritable Wife",
33               "The Hitman's Petulant Wife", "The Hitman's Pouting Wife", "The Hitman's Sulky Wife",
34               "The Hitman's Sullen Wife", "The Hitman's Teary Wife", "The Hitman's Tearful Wife",
35               "The Hitman's Tragic Wife", "The Hitman's Unhappy Wife", "The Hitman's Unlucky Wife",
36               "The Hitman's Unwell Wife", "The Hitman's Upset Wife", "The Hitman's Utterly Fed Up Wife",
37               "The Hitman's Vexed Wife", "The Hitman's Winded Wife", "The Hitman's Exhausted Wife",
38               "The Hitman's Spent Wife", "The Hitman's Defeated Wife", "The Hitman's Dejected Wife",
39               "The Hitman's Downcast Wife", "The Hitman's Discouraged Wife", "The Hitman's Disheartened Wife",
40               "The Hitman's Dispirited Wife", "The Hitman's Dismayed Wife", "The Hitman's Disturbed Wife",
41               "The Hitman's Troubled Wife", "The Hitman's Aggrieved Wife", "The Hitman's Annoyed Wife",
42               "The Hitman's Angry Wife", "The Hitman's Fuming Wife", "The Hitman's Furious Wife",
43               "The Hitman's Incensed Wife", "The Hitman's Indignant Wife", "The Hitman's Irate Wife",
44               "The Hitman's Irritable Wife", "The Hitman's Petulant Wife", "The Hitman's Pouting Wife",
45               "The Hitman's Sulky Wife", "The Hitman's Sullen Wife", "The Hitman's Teary Wife",
46               "The Hitman's Tearful Wife", "The Hitman's Tragic Wife", "The Hitman's Unhappy Wife",
47               "The Hitman's Unlucky Wife", "The Hitman's Unwell Wife", "The Hitman's Upset Wife",
48               "The Hitman's Utterly Fed Up Wife", "The Hitman's Vexed Wife", "The Hitman's Winded Wife",
49               "The Hitman's Exhausted Wife", "The Hitman's Spent Wife", "The Hitman's Defeated Wife",
50               "The Hitman's Dejected Wife", "The Hitman's Downcast Wife", "The Hitman's Discouraged Wife",
51               "The Hitman's Disheartened Wife", "The Hitman's Dispirited Wife", "The Hitman's Dismayed Wife",
52               "The Hitman's Disturbed Wife", "The Hitman's Troubled Wife", "The Hitman's Aggrieved Wife",
53               "The Hitman's Annoyed Wife", "The Hitman's Angry Wife", "The Hitman's Fuming Wife",
54               "The Hitman's Furious Wife", "The Hitman's Incensed Wife", "The Hitman's Indignant Wife",
55               "The Hitman's Irate Wife", "The Hitman's Irritable Wife", "The Hitman's Petulant Wife",
56               "The Hitman's Pouting Wife", "The Hitman's Sulky Wife", "The Hitman's Sullen Wife",
57               "The Hitman's Teary Wife", "The Hitman's Tearful Wife", "The Hitman's Tragic Wife",
58               "The Hitman's Unhappy Wife", "The Hitman's Unlucky Wife", "The Hitman's Unwell Wife",
59               "The Hitman's Upset Wife", "The Hitman's Utterly Fed Up Wife", "The Hitman's Vexed Wife",
60               "The Hitman's Winded Wife", "The Hitman's Exhausted Wife", "The Hitman's Spent Wife",
61               "The Hitman's Defeated Wife", "The Hitman's Dejected Wife", "The Hitman's Downcast Wife",
62               "The Hitman's Discouraged Wife", "The Hitman's Disheartened Wife", "The Hitman's Dispirited Wife",
63               "The Hitman's Dismayed Wife", "The Hitman's Disturbed Wife", "The Hitman's Troubled Wife",
64               "The Hitman's Aggrieved Wife", "The Hitman's Annoyed Wife", "The Hitman's Angry Wife",
65               "The Hitman's Fuming Wife", "The Hitman's Furious Wife", "The Hitman's Incensed Wife",
66               "The Hitman's Indignant Wife", "The Hitman's Irate Wife", "The Hitman's Irritable Wife",
67               "The Hitman's Petulant Wife", "The Hitman's Pouting Wife", "The Hitman's Sulky Wife",
68               "The Hitman's Sullen Wife", "The Hitman's Teary Wife", "The Hitman's Tearful Wife",
69               "The Hitman's Tragic Wife", "The Hitman's Unhappy Wife", "The Hitman's Unlucky Wife",
70               "The Hitman's Unwell Wife", "The Hitman's Upset Wife", "The Hitman's Utterly Fed Up Wife",
71               "The Hitman's Vexed Wife", "The Hitman's Winded Wife", "The Hitman's Exhausted Wife",
72               "The Hitman's Spent Wife", "The Hitman's Defeated Wife", "The Hitman's Dejected Wife",
73               "The Hitman's Downcast Wife", "The Hitman's Discouraged Wife", "The Hitman's Disheartened Wife",
74               "The Hitman's Dispirited Wife", "The Hitman's Dismayed Wife", "The Hitman's Disturbed Wife",
75               "The Hitman's Troubled Wife", "The Hitman's Aggrieved Wife", "The Hitman's Annoyed Wife",
76               "The Hitman's Angry Wife", "The Hitman's Fuming Wife", "The Hitman's Furious Wife",
77               "The Hitman's Incensed Wife", "The Hitman's Indignant Wife", "The Hitman's Irate Wife",
78               "The Hitman's Irritable Wife", "The Hitman's Petulant Wife", "The Hitman's Pouting Wife",
79               "The Hitman's Sulky Wife", "The Hitman's Sullen Wife", "The Hitman's Teary Wife",
80               "The Hitman's Tearful Wife", "The Hitman's Tragic Wife", "The Hitman's Unhappy Wife",
81               "The Hitman's Unlucky Wife", "The Hitman's Unwell Wife", "The Hitman's Upset Wife",
82               "The Hitman's Utterly Fed Up Wife", "The Hitman's Vexed Wife", "The Hitman's Winded Wife",
83               "The Hitman's Exhausted Wife", "The Hitman's Spent Wife", "The Hitman's Defeated Wife",
84               "The Hitman's Dejected Wife", "The Hitman's Downcast Wife", "The Hitman's Discouraged Wife",
85               "The Hitman's Disheartened Wife", "The Hitman's Dispirited Wife", "The Hitman's Dismayed Wife",
86               "The Hitman's Disturbed Wife", "The Hitman's Troubled Wife", "The Hitman's Aggrieved Wife",
87               "The Hitman's Annoyed Wife", "The Hitman's Angry Wife", "The Hitman's Fuming Wife",
88               "The Hitman's Furious Wife", "The Hitman's Incensed Wife", "The Hitman's Indignant Wife",
89               "The Hitman's Irate Wife", "The Hitman's Irritable Wife", "The Hitman's Petulant Wife",
90               "The Hitman's Pouting Wife", "The Hitman's Sulky Wife", "The Hitman's Sullen Wife",
91               "The Hitman's Teary Wife", "The Hitman's Tearful Wife", "The Hitman's Tragic Wife",
92               "The Hitman's Unhappy Wife", "The Hitman's Unlucky Wife", "The Hitman's Unwell Wife",
93               "The Hitman's Upset Wife", "The Hitman's Utterly Fed Up Wife", "The Hitman's Vexed Wife",
94               "The Hitman's Winded Wife", "The Hitman's Exhausted Wife", "The Hitman's Spent Wife",
95               "The Hitman's Defeated Wife", "The Hitman's Dejected Wife", "The Hitman's Downcast Wife",
96               "The Hitman's Discouraged Wife", "The Hitman's Disheartened Wife", "The Hitman's Dispirited Wife",
97               "The Hitman's Dismayed Wife", "The Hitman's Disturbed Wife", "The Hitman's Troubled Wife",
98               "The Hitman's Aggrieved Wife", "The Hitman's Annoyed Wife", "The Hitman's Angry Wife",
99               "The Hitman's Fuming Wife", "The Hitman's Furious Wife", "The Hitman's Incensed Wife",
100              "The Hitman's Indignant Wife", "The Hitman's Irate Wife", "The Hitman's Irritable Wife",
101              "The Hitman's Petulant Wife", "The Hitman's Pouting Wife", "The Hitman's Sulky Wife",
102              "The Hitman's Sullen Wife", "The Hitman's Teary Wife", "The Hitman's Tearful Wife",
103              "The Hitman's Tragic Wife", "The Hitman's Unhappy Wife", "The Hitman's Unlucky Wife",
104              "The Hitman's Unwell Wife", "The Hitman's Upset Wife", "The Hitman's Utterly Fed Up Wife",
105              "The Hitman's Vexed Wife", "The Hitman's Winded Wife", "The Hitman's Exhausted Wife",
106              "The Hitman's Spent Wife", "The Hitman's Defeated Wife", "The Hitman's Dejected Wife",
107              "The Hitman's Downcast Wife", "The Hitman's Discouraged Wife", "The Hitman's Disheartened Wife",
108              "The Hitman's Dispirited Wife", "The Hitman's Dismayed Wife", "The Hitman's Disturbed Wife",
109              "The Hitman's Troubled Wife", "The Hitman's Aggrieved Wife", "The Hitman's Annoyed Wife",
110              "The Hitman's Angry Wife", "The Hitman's Fuming Wife", "The Hitman's Furious Wife",
111              "The Hitman's Incensed Wife", "The Hitman's Indignant Wife", "The Hitman's Irate Wife",
112              "The Hitman's Irritable Wife", "The Hitman's Petulant Wife", "The Hitman's Pouting Wife",
113              "The Hitman's Sulky Wife", "The Hitman's Sullen Wife", "The Hitman's Teary Wife",
114              "The Hitman's Tearful Wife", "The Hitman's Tragic Wife", "The Hitman's Unhappy Wife",
115              "The Hitman's Unlucky Wife", "The Hitman's Unwell Wife", "The Hitman's Upset Wife",
116              "The Hitman's Utterly Fed Up Wife", "The Hitman's Vexed Wife", "The Hitman's Winded Wife",
117              "The Hitman's Exhausted Wife", "The Hitman's Spent Wife", "The Hitman's Defeated Wife",
118              "The Hitman's Dejected Wife", "The Hitman's Downcast Wife", "The Hitman's Discouraged Wife",
119              "The Hitman's Disheartened Wife", "The Hitman's Dispirited Wife", "The Hitman's Dismayed Wife",
120              "The Hitman's Disturbed Wife", "The Hitman's Troubled Wife", "The Hitman's Aggrieved Wife",
121              "The Hitman's Annoyed Wife", "The Hitman's Angry Wife", "The Hitman's Fuming Wife",
122              "The Hitman's Furious Wife", "The Hitman's Incensed Wife", "The Hitman's Indignant Wife",
123              "The Hitman's Irate Wife", "The Hitman's Irritable Wife", "The Hitman's Petulant Wife",
124              "The Hitman's Pouting Wife", "The Hitman's Sulky Wife", "The Hitman's Sullen Wife",
125              "The Hitman's Teary Wife", "The Hitman's Tearful Wife", "The Hitman's Tragic Wife",
126              "The Hitman's Unhappy Wife", "The Hitman's Unlucky Wife", "The Hitman's Unwell Wife",
127              "The Hitman's Upset Wife", "The Hitman's Utterly Fed Up Wife", "The Hitman's Vexed Wife",
128              "The Hitman's Winded Wife", "The Hitman's Exhausted Wife", "The Hitman's Spent Wife",
129              "The Hitman's Defeated Wife", "The Hitman's Dejected Wife", "The Hitman's Downcast Wife",
130              "The Hitman's Discouraged Wife", "The Hitman's Disheartened Wife", "The Hitman's Dispirited Wife",
131              "The Hitman's Dismayed Wife", "The Hitman's Disturbed Wife", "The Hitman's Troubled Wife",
132              "The Hitman's Aggrieved Wife", "The Hitman's Annoyed Wife", "The Hitman's Angry Wife",
133              "The Hitman's Fuming Wife", "The Hitman's Furious Wife", "The Hitman's Incensed Wife",
134              "The Hitman
```

# GET YOUR OWN FAKE\_DATA.CSV

- Download csv file from Canvas -> GitHub
- Download python file from Canvas -> GitHub and run it
- Bonus question: what will be different?

# CHECK THAT YOU CAN READ IT

```
1 with open("fake_data.csv", "r") as file:  
2     header = file.readline()  
3     print(header)  
4     line = file.readline()  
5     print(line)
```

name,favorite\_movie,second\_favorite\_movie,preferred\_political\_party,ideal\_diet

Arya Stark, Lord of the Rings: The Fellowship of the Ring, Lord of the Rings:  
The Fellowship of the Ring, left wing, balanced

# SIMILARITY

Our initial definition: “common interests counter”

- If we have the same favorite movie, that’s one common interest
- If we have the same preferred political party, that’s one common interest.
- If we also have the same preferred diet, that’s one common interest.
- Let’s see who is the most similar!

# MANY DEFINITIONS OF SIMILAR

In CS and math, there are many ways we can see how “similar” or “close” two items are

- (cosine distance, manhattan distance, etc.)
- So we need to specify a definition and algorithm



# OUR MOVIE/POLITICAL PARTY/DIET RECOMMENDER SYSTEM

To recommend a movie to someone, we take the following approach:

- We'll look at favorite movie, preferred political party, and ideal diet
- Each match counts as one similarity score. So score between any two people ranges from 0 to 3.
- When we find the “highest” score, suggest our user watch that person's second favorite movie!

# OUR ALGORITHM

## INITIALIZE

- Set variables to hold all the things we need (i.e., get our buckets ready)

## LOOP

- Run through every record and process it

# INITIALIZE

- Pick the user we'll give a recommendation to
- Define a variable to hold similarity score (starts at zero)
- Define a variable to keep track of which person is most similar
- Define a variable to keep track of what what we'll recommend

# LOOP

## FOR EACH PERSON OTHER THAN OUR CHOSEN ONE

# OUR PSEUDO CODE

```
1 # select a user (who will receive recommendation)
2 # init variables to hold top similarity score, top person, recommendation
3
4 # load data
5     # use a loop to go through all other record lines (one record = one use
6     # make sure to skip chosen user
7     # split each record line into a list of items
8     # use a nested loop to go through each item
9     # make sure we compare columns correctly
10    # check if similarity score is higher
11    # update if so
12
13 # print top score, record, and recommendation
```

# INITIALIZE

```
1 # select a user + init variables to keep track of scores
2 index_of_user = 17
3 top_score = 0
4 top_record = ""
5 recommendation = ""
6
7 # load data
8     # use a loop to go through all other record lines (one record = one use
9     # make sure to skip chosen user
10    # split each record line into a list of items
11        # use a nested loop to go through each item
12        # make sure we compare columns correctly
13    # check if similarity score is higher
14        # update if so
15
16 # print top score, record, and recommendation
```

# LOAD DATA

```
1 # select a user + init variables to keep track of scores
2 index_of_user = 17
3 top_score = 0
4 top_record = ""
5 recommendation = ""
6
7 # load data
8 with open("fake_data.csv", "r") as file:
9     # Skip header
10    header = file.readline()
11    print(header)
12    # use a loop to go through all other record lines (one record = one use
13    # make sure to skip chosen user
14    # split each record line into a list of items
15        # use a nested loop to go through each item
16        # make sure we compare columns correctly
17    # check if similarity score is higher
18        # update if so
19
```

name, favorite\_movie, second\_favorite\_movie, preferred\_political\_party, ideal\_diet

# FIRST FOR LOOP

```
1 # select a user + init variables to keep track of scores
2 index_of_user = 17
3 top_score = 0
4 top_record = ""
5 recommendation = ""
6
7 # load data
8 with open("fake_data.csv", "r") as file:
9     # Skip header
10     all_lines = file.readlines()
11     header = all_lines[0]
12     records = all_lines[1:]
13
14     user_record = records[index_of_user]
15
16     # use a loop to go through all other record lines (one record = one use
17     for record in records:
18         if record == user_record:
```

```
['Arya Stark', 'Lord of the Rings: The Fellowship of the Ring', 'Lord of the
Rings: The Fellowship of the Ring', 'left wing', 'balanced']
['Superman', 'Inception', 'The Matrix', 'right wing', 'plant-heavy']
['Spider-Man', 'Pulp Fiction', 'The Dark Knight', 'very left wing',
```



```
'balanced']  
['Rihanna', 'Citizen Kane', 'Citizen Kane', 'very right wing', 'balanced']  
['Luffy', 'Finding Nemo', 'The Lion King', 'left wing', 'dairy-heavy']  
['Ash Ketchum', 'Shrek', 'Shrek', 'very right wing', 'plant-heavy']  
['Wonder Woman', 'Avengers: Endgame', 'Avengers: Endgame', 'very left wing',  
'fruit-heavy']  
['Lara Croft', 'Star Wars: Episode IV', 'The Matrix', 'very right wing',  
'plant-heavy']  
['Walter White', 'Shrek', 'The Lion King', 'very right wing', 'meat-heavy']  
['Walter White', 'Citizen Kane', 'Gone with the Wind', 'very left wing',  
'balanced']
```

# NOTE ON PRINTING AS WE GO

As we write this code, at each step let's make our code print something out so we know we're making progress!

When you're feeling stuck, trying printing something for every new line of code you write.

# NESTED LOOP

```
1 # select a user + init variables to keep track of scores
2 index_of_user = 17
3 top_score = 0
4 top_record = ""
5 recommendation = ""
6
7 # load data
8 with open("fake_data.csv", "r") as file:
9     all_lines = file.readlines()
10    header = all_lines[0]
11    records = all_lines[1:]
12
13    user_record = records[index_of_user].strip().split(",")
14
15    # use a loop to go through all other record lines (one record = one use
16    for record in records:
17        columns = record.strip().split(",")
18
19        if columns[0] == user_record[0]:
```

# NESTED LOOP

Skipping Wonder Woman, Avengers: Endgame, Avengers: Endgame, very left wing, fruit-heavy

Skipping Wonder Woman, Shrek, The Lion King, right wing, fruit-heavy

Skipping Wonder Woman, The Godfather, Forrest Gump, left wing, plant-heavy

You are ['Wonder Woman', 'The Godfather', 'Forrest Gump', 'left wing', 'plant-heavy']

After careful consideration, we have found that the most similar user is Wolverine, Shrek, Shrek, left wing, plant-heavy

You have a similarity score of 2

We recommend you watch Shrek

