

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 slide for week 5 (Monday was a holiday).

5-1 HOUSEKEEPING

- Updates on marking
- Updates on additional study materials (and how to get feedback)
- Office hours today

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 OBJECTIVE

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 Joh
6         "Mickey Mouse", "Hermione Granger", "Naruto", "Serena Williams", "
7         "Leonardo DiCaprio", "Mario", "Taylor Swift", "Darth Vader", "Jame
8         "Usain Bolt", "Tony Stark", "Bugs Bunny", "The Rock", "Walter Whit
9         "Sheldon Cooper", "Mulan", "Elsa", "Sherlock Holmes", "Jack Sparro
10        "Luffy", "Luke Skywalker", "Marilyn Monroe", "Freddie Mercury", "L
11        "Superman", "Wonder Woman", "Neo", "Beyonce", "Ash Ketchum",
12        "Lara Croft", "Elvis Presley", "Kermit the Frog", "Spider-Man", "H
13        "Indiana Jones", "Simba", "Rihanna", "Dexter Morgan", "Morpheus",
14        "Wolverine", "Cinderella", "Captain America", "Thor", "Katniss Eve
15
16 movies_by_genre = {
17     "Drama": ["The Shawshank Redemption", "Forrest Gump", "The Godfather",
18     "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
19     "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
20     "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
21     "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
22     "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
23     "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
24     "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
25     "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
26     "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
27     "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
28     "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
29     "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
30     "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
31     "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
32     "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
33     "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
34     "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
35     "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
36     "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
37     "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
38     "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
39     "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
40     "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
41     "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
42     "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
43     "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
44     "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
45     "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
46     "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
47     "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
48     "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
49     "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
50     "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
51     "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
52     "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
53     "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
54     "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
55     "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
56     "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
57     "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
58     "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
59     "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
60     "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
61     "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
62     "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
63     "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
64     "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
65     "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
66     "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
67     "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
68     "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
69     "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
70     "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
71     "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
72     "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
73     "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
74     "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
75     "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
76     "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
77     "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
78     "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
79     "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
80     "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
81     "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
82     "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
83     "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
84     "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
85     "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
86     "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
87     "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
88     "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
89     "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
90     "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
91     "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
92     "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
93     "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
94     "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
95     "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
96     "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
97     "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
98     "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
99     "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
100    "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
101    "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
102    "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
103    "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
104    "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
105    "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
106    "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
107    "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
108    "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
109    "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
110    "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
111    "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
112    "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
113    "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
114    "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
115    "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
116    "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
117    "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
118    "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
119    "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
120    "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
121    "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
122    "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
123    "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
124    "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
125    "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
126    "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
127    "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
128    "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
129    "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
130    "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
131    "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
132    "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
133    "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
134    "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
135    "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
136    "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
137    "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
138    "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
139    "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
140    "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
141    "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
142    "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
143    "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
144    "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
145    "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
146    "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
147    "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
148    "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
149    "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
150    "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
151    "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
152    "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
153    "Horror": ["The Exorcist", "The Silence of the Lambs", "The Shining",
154    "Mystery": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
155    "Romance": ["The Notebook", "The Notebook", "The Notebook", "The Notebo
156    "Sci-Fi": ["The Matrix", "The Matrix", "The Matrix", "The Matrix", "The
157    "Sports": ["The Shawshank Redemption", "The Shawshank Redemption", "The
158    "Thriller": ["The Usual Suspects", "The Usual Suspects", "The Usual Sus
159    "War": ["The Shawshank Redemption", "The Shawshank Redemption", "The
160    "Western": ["The Shawshank Redemption", "The Shawshank Redemption", "The
161    "Drama": ["The Shawshank Redemption", "The Shawshank Redemption", "The
162    "Action": ["The Dark Knight", "Avengers: Endgame", "Pulp Fiction", "Jur
163    "Comedy": ["The Hangover", "The 400 Blows", "The Godfather", "The
164    "Fantasy": ["The Lord of the Rings: The Fellowship of the Ring", "The
165    "Horror": ["The Ex
```

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
```

```
Jack Sparrow,Lord of the Rings: The Fellowship of the Ring,Pulp Fiction,right
wing,balanced
```

SIMILARITY

Our initial definition 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 line == user_record:
```

```
['Jack Sparrow', 'Lord of the Rings: The Fellowship of the Ring', 'Pulp
Fiction', 'right wing', 'balanced']
```

```
['Lara Croft', 'Frozen', 'Frozen', 'very right wing', 'dairy-heavy']
```

```
['Hannibal Lecter', 'Back to the Future', 'Star Wars: Episode IV', 'very right
```

```
wing', 'fruit-heavy']  
['Marilyn Monroe', 'Psycho', 'Gone with the Wind', 'right wing', 'fruit-  
heavy']  
['Sheldon Cooper', 'Gone with the Wind', 'Jaws', 'very left wing', 'balanced']  
['James Bond', 'Pulp Fiction', 'Avengers: Endgame', 'right wing', 'balanced']  
['Katniss Everdeen', 'Shrek', 'Frozen', 'very right wing', 'meat-heavy']  
['Lady Gaga', 'Back to the Future', 'Star Wars: Episode IV', 'very left wing',  
'balanced']  
['Arya Stark', 'Avengers: Endgame', 'The Dark Knight', 'very left wing',  
'dairy-heavy']  
['Kermit the Frog', 'The Godfather', 'The Shawshank Redemption', 'very left
```

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.

How to draw an Owl



Fig 1. Draw two circles



Fig 2. Draw the rest of the Owl

Fig 1.
Here are some
Lean concepts

Fig 2.
Now go
practice

How to draw an owl dot jpeg

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 Ash Ketchum,Casablanca,Citizen Kane,very right wing,meat-heavy

You are ['Ash Ketchum', 'Casablanca', 'Citizen Kane', 'very right wing', 'meat-heavy']

After careful consideration, we have found that the most similar user is Katniss Everdeen,Shrek,Frozen,very right wing,meat-heavy

You have a similarity score of 2

We recommend you watch Frozen

OK, LET'S TAKE A STEP BACK

- switch to live coding here

