Project 2 Overview

21 September 2016

Lecture overview

Last lecture

- HTML and Python

Today

- Overview of Project 2

Project 1

- We are collating the marks. Should be out soon

Project 2

Data Analytics in Commerce and Marketing

- Imagine you want to produce an online restaurant guide
- There are other restaurant guides out there, so how do you make your guide special?
- By making use of people's ratings of restaurants:
  - How do you find people with similar tastes in food?
  - What are the most popular types of cuisines?
  - How can you make personalised restaurant recommendations to people?
- Armed with these snazzy services, your restaurant guide can capture the market

Data Analytics in Commerce and Marketing

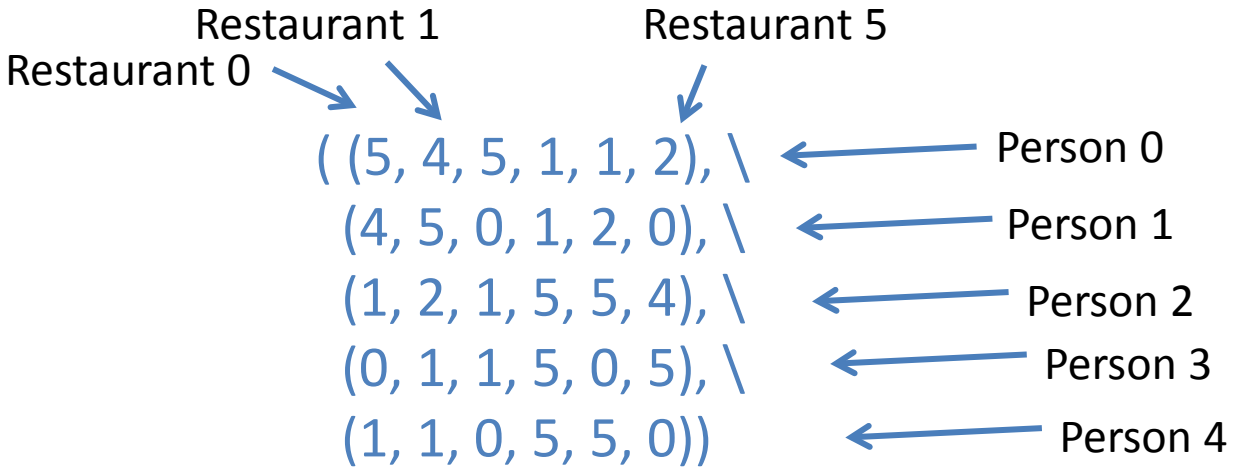
- We will look at three different approaches to analysing people's ratings of restaurants
  - Calculate the similarity between two people based on their restaurant ratings (question 1)
  - Find which type of restaurant cuisine gets the highest average rating (question 2)
  - Recommend a restaurant to a person based on preferences of other similar people (question 3)

Skills developed

- This project will develop your skills with tuples, and multi-dimensional data sets
- You will get more practice with nested loops
- You will start to implement some real-life functions

Question 1

Each person is represented by a tuple containing the restaurant ratings of that person, e.g., here are the ratings by 5 people for 6 restaurants



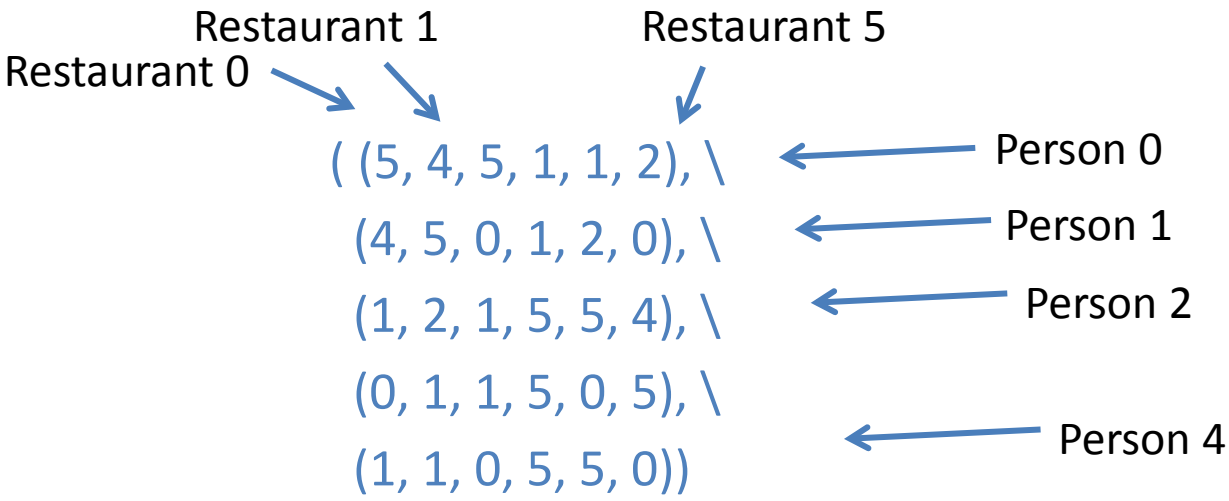
where 5 is the most favourable rating,
1 is the least favourable rating, and
0 means the person has not rated the restaurant

What is the similarity between two people in terms of their ratings?

Question 1

For example, people 0 and 1 like restaurants 0 and 1, but not 3 and 4.

In contrast, people 2 and 3 like restaurants 3 and 5, but not 0 and 2.



We will quantify this similarity using the cosine similarity measure

Question 1

Let the tuple of ratings for person 0 be $(v_{0,0}, v_{0,1}, \ldots, v_{0,n-1})$

and for person 1 be $(v_{1,0}, v_{1,1}, \ldots, v_{1,n-1})$

and let M be the set of restaurants rated by **both** people

The cosine similarity is defined as:

$$C = \frac{\sum_{m \in M} v_{0,m} v_{1,m}}{\sqrt{\sum_{m \in M} {v_{0,m}}^2} \sqrt{\sum_{m \in M} {v_{1,m}}^2}}$$

where $\sum_{m \in M} {v_{0,m}}^2$ means the sum of ${v_{0,m}}^2$ for all values of *m* in *M*

From the last slide, person 0 is (5, 4, 5, 1, 1, 2) and person 1 is (4, 5, 0, 1, 2, 0),

so $$C = \frac{5\times4 + 4\times5 + 1\times1 + 1\times2}{\sqrt{5^2 + 4^2 + 1^2 + 1^2}\sqrt{4^2 + 5^2 + 1^2 + 2^2}} = 0.97$$ (high similarity)

Between person 0 and person 3

$$C = \frac{4\times1 + 5\times1 + 1\times5 + 2\times5}{\sqrt{4^2 + 5^2 + 1^2 + 2^2}\sqrt{1^2 + 1^2 + 5^2 + 5^2}} = 0.49$$ (lower similarity)

Question 1

Write a function similarity(p1, p2, ratings)

that takes the index p1 of one person in ratings,

the index p2 of another person in ratings,

and a tuple of tuples ratings corresponding to the ratings of a set of people,

and returns the float value of the cosine similarity

of the two specified people in terms of their ratings.

```
>>> print(similarity(0, 1, ((5, 4, 2, 1), (1, 2, 5, 5), (4, 5, 0, 0))))
0.56
```

Question 2

Given a type of cuisine (e.g., “italian”),
a collection of restaurants and their cuisines, and
a collection of ratings for the restaurants,
calculate the average rating for the given cuisine
by averaging all the (**non-zero**) ratings by customers
for restaurants with that type of cuisine

For example,

```
>>> print(ave_rating('thai', \
                    (('top thai','thai'), ('pizza palace','italian'), \
                     ('krabi hut','thai'), ('tacky thai','thai')), \
                    ((5, 4, 2, 1), (1, 2, 5, 5), (4, 5, 0, 0))))
3.3
```

Type of cuisine

Tuple of restaurants,
each restaurant described by a tuple
(‘name’, ‘cuisine’)

Tuple of ratings, one tuple per person,
each containing the ratings by that person for all restaurants

Question 3

- How do we recommend restaurants to a person?
- Look at that person's restaurant ratings,
  look for similar people (with similar ratings),
  and recommend restaurants that those similar people liked
- Using the example from Question 1,
  person 0 (5, 4, 5, 1, 1, 2) is very similar to
  person 1 (4, 5, 0, 1, 2, 0)
- Person 1 hasn't tried restaurants 2 or 5
- Since person 0 is similar to person 1,
  and person 0 liked restaurant 2 more than restaurant 5,
  we could recommend restaurant 2 to person 1

Question 3

- For a given person (such as person 1), we want to compute the recommended rating for each restaurant that has not been rated by that person
- For example, person 1 has not rated restaurants 2 or 5
- What is the recommended rating $\hat{v}_{1,2}$ for person 1 for restaurant 2, and $\hat{v}_{1,5}$ for restaurant 5 for person 1?
- We can then recommend whichever of restaurant 2 or restaurant 5 has the highest recommended rating

Question 3

The recommended rating $\hat{v}_{i,j}$ of restaurant *j* for person *i* is the weighted average of the ratings given by all other people with a non-zero rating for restaurant *j*, where the weighting of the rating from each person is based on the cosine similarity of that person to person *i*

$$\hat{v}_{1,2} = \frac{s_{0,1} \times v_{0,2} + s_{2,1} \times v_{2,2} + s_{3,1} \times v_{3,2}}{s_{0,1} + s_{2,1} + s_{3,1}}$$

$$\hat{v}_{1,5} = \frac{s_{0,1} \times v_{0,5} + s_{2,1} \times v_{2,5} + s_{3,1} \times v_{3,5}}{s_{0,1} + s_{2,1} + s_{3,1}}$$

Question 3

To compute the recommended ratings,
we need the cosine similarities between all pairs of people.

We will give you these similarities in Question 3.

Ratings:

```
( (5, 4, 5, 1, 1, 2), \
  (4, 5, 0, 1, 2, 0), \
  (1, 2, 1, 5, 5, 4), \
  (0, 1, 1, 5, 0, 5), \
  (1, 1, 0, 5, 5, 0))
```

Similarities:

```
( (1.0, 0.97, 0.50, 0.49, 0.40), \
  (0.97, 1.0, 0.58, 0.38, 0.49), \
  (0.50, 0.58, 1.0, 0.98, 0.99), \
  (0.49, 0.38, 0.98, 1.0, 1.0), \
  (0.40, 0.49, 0.99, 1.0, 1.0))
```

Question 3

Putting all this together:

$$\hat{v}_{1,2} = \frac{0.97\times5+0.58\times1+0.38\times1}{0.97+0.58+0.38} = 1.32$$

$$\hat{v}_{1,5} = \frac{0.97\times2+0.58\times4+0.38\times5}{0.97+0.58+0.38} = 1.67$$

So we would tend to recommend restaurant 5 for person 1
(in this case the results were a bit *biased* because there were *more ratings* for restaurant 5 than restaurant 1 – this is a problem with small data sets)

Question 3

Given an integer person,
a tuple ratings containing the ratings
from a collection of people, and
a tuple similarities containing the similarities
between the collection of people,
returns an integer corresponding to the index
of the recommended restaurant for the person

Index of person

Tuple of ratings, one tuple per person,
each containing the ratings by that person
for all restaurants

For example,

```
>>> print(recommendation(2, \
        ((5, 1, 4, 2), (1, 4, 2, 5), (1, 5, 0, 0)), \
        ((1.0, 0.59, 0.38), \
         (0.59, 1.0, 0.99), \
         (0.38, 0.99, 1.0))))
```

Table of similarities
between all pairs of people

Academic Honesty

- All assessment items (worksheets, projects, test and exam) must be **your own, individual, original work**.
- For example, you must not copy the code of other students, and you must not make your code available to others to see. Do not give other students your login id and password, do not share USB memory drives, do not post your code on public forums, or any other activity that would make your code available to others. Likewise, do not ask other students to see their code. If other students ask to see your code, please say "no", as copying (collusion or plagiarism) is considered academic misconduct, and all students involved may face penalties (both the student who copied, and the student who made their code available).
- Any code that is submitted for assessment may be automatically compared against other students' code and other code sources using sophisticated similarity checking software, and cases of potential copying may lead to a formal academic misconduct hearing.
- For further information, please see the university's Academic Honesty and Plagiarism website, or ask your lecturer.

Conclusion

- The project questions will be submitted via Grok
- The specification of the project questions will be in Grok later this week
- The deadline will be in the first week back after the mid-semester break (see specification when released in Grok)
- This project is worth 10% of the final subject
- We will be marking the correctness, quality, readability and commenting of your code
- Make progress submission early and often
- Even if your submission doesn’t work, you might still get partial marks