

## Worksheet 2 – Chapter 2

---

Team: \_\_\_\_\_

Up to 60XP

This worksheet is on the recommender class of ch2.

1. What would you recommend for Chan?

---

2. Who is Hailey's nearest neighbor

---

3. Who are the closest two neighbors?

---

4. k-nearest neighbor

Find recommendations for Hailey using the k-nearest neighbor algorithm with  $k = 3$ .

**Band**

**Rating**

---

---

---

Explain in detail how the algorithm determined these recommendations and ratings. (for example, who were the nearest neighbors. (use back side of this sheet)

## 5. Looking at Python code.

In the Python shell I type the following:

```
rec = {'Phoenix': 5, 'Black Keys': 4.5, 'Weird Al': 3, 'Vampire': 2}
```

Explain what the following does:

```
list(rec)[:3]
```

---

---

---

What does the following do and how is it different from the above?

```
list(rec.items())[:3]
```

---

---

---

Using that knowledge, explain what lines 195-8 do in the recommender.py code.

---

---

---

---

---

---

---

## 6. Use the fastest laptops.

Using the fastest laptops in your team, load the book crossing dataset. You will be looking at recommendations for user 171118.

### 6.1 With $k=1$ what are the top 5 recommendations?

---

---

---

---

---

### 6.2 With $k=2$ what are the top 5 recommendations?

---

---

---

---

---

### 6.3 With $k=3$ what are the top 5 recommendations?

---

---

---

---

---

**6.4 With  $k=4$  what are the top 5 recommendations?**

---

---

---

---

---

**6.5 Explain the differences. Which  $k$  do you think is best to use?**

---

---

---

---

---

**7. Where does user 171118 live?**

Hint: look at code beginning at line 101. (The point of this question is to have people look at the code)

**8. Implement Cosine Similarity**

Implement Cosine Similarity.

Using that algorithm, what would you recommend for user 278442 from Fredericksburg?

---

---

---

---

---

## 9. Implement Manhattan Distance.

Cosine similarity and Pearson both have a rating between 1 (maximally similar), 0 (no similarity) and -1 (maximally negatively related). I would like you to implement Manhattan distance in such a way that 1 indicates most similar and 0 indicates dissimilar. In the green real-world textbook there was mention about normalization. That may help.

### 9.1 Describe your method in English

---

---

---

---

---

### 9.2 Compare algorithms

Compare the performance of Manhattan Distance, Cosine Similarity, and Pearson on any dataset you want. The music dataset (users in the code) is nice because it is small, but it has made-up values. Perhaps you could augment that with ratings from your team.

---

---

---

---

---