

E09 – Recommender System (Python)

Business Intelligence

Exercise

Winter Term 2025/2026

Introduction - Recommender Systems

- A recommender systems (also recommendation systems) are part of our daily life. Almost any IT system that works with users or products uses some sort of recommender system. You can read more on recommender systems here (see section with “credentials and materials” for further information) - <https://builtin.com/data-science/recommender-systems>
- In this assignment, you will
 - calculate similarities
 - implement simple recommender system



Notes on Software (I/III) - Python

- The current exercise uses Python / Jupyter Notebook, which is an open platform-as-a-service that provides a very popular opportunity to work with Python directly in your browser. Python as a programming language is quite widespread nowadays and got an amazing community. Despite availability of a *Natural Language Processing (NLP)* capabilities in form of various packages in Python ecosystem, your task will be to implement sentiment analysis on your own and then try to perform sentiment analysis by using provided inside the database methods.
- **NOTE:** If you are familiar with any similar tool/software, which you could potentially use to implement the current assignment you are free to use it



Notes on Software (I/III) - Jupyter Notebook

Use one of the options below to setup working environment

- Option 1. Locally installed Python + further requirements
 - Python
 - <https://www.python.org/downloads/release/python-3128/>
 - Information on tools (IDE, packages, etc.)
 - <https://github.com/vdmitriyev/uol-data-analytics?tab=readme-ov-file#gear-tools>
 - Option 2. Locally installed Jupyter Notebook
 - You can install it on your PC and then use it, here you are going to find more information about the installation of this tool (as part of the Anaconda Distribution) -
 - <https://www.anaconda.com/products/individual>
 - Options 3. Remote SaaS
 - You can also use online service that offer Python environment (e.g., Google Colab)

Notes on Software (III/III) - Infrastructure

Additional instructions on the infrastructure (online platform) for this course can be found in

E00 - Infrastructure

If you are new to Python, here you are going to find all required for this assignment materials to master Python programming

<https://github.com/vdmitriyev/uol-data-analytics>



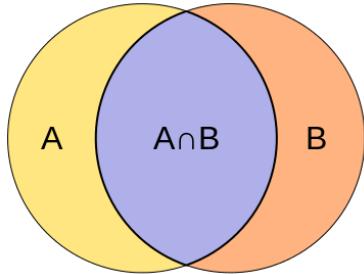
Task 1 (I/II)– Similarity

A common problem to be solved while working with a recommender system is to design (or select) a metric function, such as it could be used to calculate similarities between two and more items.

In this assignment you need to compare two given texts using different similarity methods – “Jaccard similarity” and “Cosine similarity”. See the description of the methods below.

Jaccard similarity used to find similarities between sets. It is defined as divisions of the cardinality of the intersection of sets with the cardinality of the union of the sample sets.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$



Cosine similarity calculates similarity by measuring the cosine of angle between two vectors, where resulting similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same
(A and B are vectors)

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Task 1 (II/II)– Similarity

Your task is to compare abstracts of the given below paper and find out the most similar papers.

Papers you should compare:

- <https://arxiv.org/abs/2007.08978>
- <https://arxiv.org/abs/2002.03389>
- <https://arxiv.org/abs/2006.16964>
- <https://arxiv.org/abs/2007.04095>
- <https://arxiv.org/abs/2008.00315>
- <https://arxiv.org/abs/1901.08151>
- <https://arxiv.org/abs/1901.10555>
- <https://arxiv.org/abs/1511.03085>
- <https://arxiv.org/abs/1905.03061>
- <https://arxiv.org/abs/2002.01759>

You could also prepare the the data before calculating similarity. The preparations steps include following: removing stop words, change to lower case, stemming, normalization, etc.

Task 2 (I/II)– Fill the gaps for ratings

In this task you are going to used movies rating database, which you already used while doing “E03 - SQL Review (Part 2)” exercise.

As long as not all users rated all movies, your task is to select and then use appropriate algorithm to fill these rating gaps. You need to select an algorithm, which is usually used with a recommender system

	Gone with the Wind	Star Wars	The Sound of Music	E.T.	Titanic	Snow White	Avatar	Raiders of the Lost Ark
Sarah Martinez	4							
Daniel Lewis						4		
Brittany Harris			2					4
Mike Anderson	3							
Chris Jackson			3	2				4
Elizabeth Thomas						5	3	
James Cameron							5	
Ashley White				3				

Task 2 (II/II)– Fill the gaps for ratings

1. Which algorithm from the recommender system's pool you selected to solve the task 2? Why?
2. Which 2 movies are the most similar to each other according to your analysis / algorithm?
3. Which 2 users are the most similar to each other according your analysis / algorithm?
4. Rate all movies, which you personally watched (no more than 3 in total) and let your algorithm “guess” missing ratings for you. Are results generated by the algorithm comparable with your personal preferences / ratings?



Submission

- Use StudIP to upload your solution (PDF report)
- You should upload the following document
 - Report as a PDF file with your solutions and explanations
 - You could also use Jupyter Notebook to directly export your solutions and explanations as HTML and convert it into PDF before submission
 - Each answer (e.g. screenshot, source code) should be annotated by you
 - Original question (+ number of the task)
 - Answer
- Name convention for your submission file (without extension)
 - **E09_FIRSTNAME LASTNAME**
- Submission deadline (it is a “soft” deadline)
 - 10 days after this exercise starts
 - Some exercises could take a bit more time and could be submitted later
 - NOTE: to receive feedback, you should first submit your progress



Credits and Materials

- <https://builtin.com/data-science/recommender-systems>
- <https://www.datacamp.com/community/tutorials/recommender-systems-python>
- https://www.researchgate.net/publication/323000727_Recommender_systems_An_overview_of_different_approaches_to_recommendations
- https://en.wikipedia.org/wiki/Recommender_system
- <https://medium.com/towards-artificial-intelligence/recommendation-system-in-depth-tutorial-with-python-for-netflix-using-collaborative-filtering-533ff8a0e444>
- <https://github.com/microsoft/recommenders>
- <https://core.ac.uk/download/pdf/231064079.pdf>
- https://en.wikipedia.org/wiki/Cosine_similarity
- https://en.wikipedia.org/wiki/Jaccard_index
- <https://www.kaggle.com/rounakbanik/the-movies-dataset/data>

