

Foundations of Data Analysis : Lab Exercise A2
WiSe 2021

General Remarks:

- The deadline for the submission is at 09:45 a.m. on 19.01.2022. Please upload your solutions on Moodle. No deadline extension is possible.
- After the above deadline, the peer review process on Moodle will start and you have to evaluate two other students' work until 30.01.2022 at 11:59 p.m., this means your own work will be reviewed as well. You can find all the further details about the peer review process on Moodle.
- For answering the questions, please use Python. Your code must be well-documented and readable; see the Style Guide for Python¹ for common style conventions.
- Upload your solutions as a zip archive with the following naming scheme **matrikelnumber_A2.zip**. The archive should contain your code as well as a PDF report (or HTML file if you work with jupyter notebooks) with your assumptions, results and a description how to compile and run your code.
- Mark and cite external sources you are using in the code and report.
- If you have problems do not hesitate to contact the tutors or post a question on the dedicated Moodle forum.

Task -1 Dimensionality Reduction, Principal Component Analysis (30 P)

Principal Component Analysis (PCA) is a dimensionality reduction method that is often used to reduce the dimensionality of datasets, by transforming a large set of variables into a smaller one, while preserving as much variance as possible. There are two methods for performing a Principal Component Analysis: using Eigenvector Decomposition (EIG) and the more generalized Singular Value Decomposition (SVD). In this exercise you will use the Wheat Seeds² dataset which involves the prediction of species given measurements of seeds from different types of wheat. You can find the *seeds.csv* file on Moodle.

- (a) Implement from scratch *PCA* via *SVD*. Project the data onto the first two principal components and compare the results with `sklearn.decomposition.PCA`³. (15 points).
- (b) Use “Principal Component 1” and “Principal Component 2” to create a scatterplot. Color the points by their corresponding *Type* label⁴. Include the plot in your write-up. (10 points)
- (c) Briefly explain the importance of PCA in the “curse of dimensionality” problem. (5 points).

¹<https://www.python.org/dev/peps/pep-0008/>

²<https://archive.ics.uci.edu/ml/datasets/seeds>

³<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

⁴You may use *seaborn* (<https://seaborn.pydata.org>) and/or *matplotlib* (<https://matplotlib.org/>).

Task -2 Clustering (30 P)

In this task you try different clustering algorithms and evaluate them using Normalized Mutual Information (NMI) and the Silhouette Score. Use four different datasets: *dataset1_noCluster7.csv*, *dataset2_noCluster6.csv*, *dataset3_noCluster2.csv*, and *dataset4_noCluster2.csv*.

- (a) Import and plot the data and if necessary preprocess it using *sklearn.preprocessing*.
- (b) Try four different clustering techniques such as DBSCAN, KMeans, Expectation Maximization (EM), and Average Link, which are already implemented in *scikit-learn*. Please state which algorithms you have used and what parameters you have chosen and why you have chosen them.
- (c) Evaluate each clustering technique using Normalized Mutual Information⁵ as well as the Silhouette Score. For the evaluation, use the *sklearn.metrics* package.
- (d) Briefly discuss your results.

Task -3 Apriori Algorithm for Recommender Systems (40P)

Task Definition:

In this programming assignment, you are required to implement the Apriori algorithm and apply it to mine frequent itemsets for movie recommendation. You are required to implement the algorithm from scratch only using native Python libraries and numpy.

Input: The provided input file (movies.txt) is based on the MovieLens dataset[1]. It contains the favourite movies of 8892 users. Each line in the file corresponds to a user and represents a list of movies the user likes. An example:

Avengers: Infinity War - Part II;Jurassic World: Fallen Kingdom

In the example above, the corresponding user likes the movies “Avengers: Infinity War - Part II” and “Jurassic World: Fallen Kingdom”.

Output: You need to implement the Apriori algorithm and use it to mine sets of movies that are frequent in the input data. After implementing the Apriori algorithm, please set the relative minimum support to 0.05 and run it on the 8892 lists of movies. In other words, you need to extract all the itemsets that have an absolute support larger than 444.

- (a) Output all the length-1 frequent movies with their absolute supports into a text file named “oneItems.txt” and place it in the root of your zip file. Every line corresponds to exactly one frequent movie and should be in the following format:

Support:movie

For example, suppose a movie (e.g. Bohemian Rhapsody) has an absolute support 503, then the line corresponding to this frequent item set in “oneItems.txt” should be:

503:Bohemian Rhapsody

- (b) Please write all the frequent itemsets along with their absolute supports into a text file named “patterns.txt” and place it in the root of your zip file. Every line corresponds to exactly one frequent itemset and should be in the following format:

support:movie_1;movie_2;movie_3;...

⁵<https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html>

For example, suppose an itemset (Bohemian Rhapsody;Aquaman) has an absolute support 446, then the line corresponding to this frequent itemset in “patterns.txt” should be:

446:Bohemian Rhapsody;Aquaman

- (c) Imagine you should recommend a movie to a user. You know that the user likes the movies “Ant-Man and the Wasp” and “Spider-Man: Far from Home”. Based on the result of the Apriori algorithm, give a movie recommendation for this user by maximizing the confidence that the user will like the movie. Explain your choice and report the confidence score for your recommendation.

References

- [1] Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4) (dec 2015). <https://doi.org/10.1145/2827872>, <https://doi.org/10.1145/2827872>