
Machine Learning 2 Assessment

“Please read the following tasks carefully. Try to benefit as much as you can from the resources. Read the [Deliverables](#) section carefully and then the [Grading](#). The [learnt Lessons](#) section will give you an overview of what you should be able to do after finishing ML 2. This task is designed to assess your skills in ML 2 techniques and to guide you towards building a good portfolio. So, please have fun and try to enjoy yourself.”

Coding Tasks	2
Recommender systems	2
Topic Modelling	2
Anomaly detection	3
Matrix Decomposition	4
K-means suffers a lot when the dataset is not flat	4
Deliverables	4
Grading	5
Learnt Lessons	6
Additional Resources	6

Coding Tasks

1. Recommender systems

- a. Given the following resources
 - i. [MovieLens](#) “dataset”
 - ii. [Python Recommender Systems: Content Based & Collaborative Filtering Recommendation Engines](#)
 - iii. [rposhala/Recommender-System-on-MovieLens-dataset: Knowledge-based, Content-based and Collaborative Recommender systems are built on MovieLens dataset with 100.000 movie ratings. These Recommender systems were built using Pandas operations and by fitting KNN, SVD & deep learning models which use NLP techniques and NN architecture to suggest movies for the users based on similar users and for queries specific to genre, user, movie, rating, popularity.](#)
 - iv. [Build A Movie Recommendation System on Your Own](#)
 - v. [Recommender Systems Python-Methods and Algorithms](#)
 - vi. [A Python scikit for building and analyzing recommender systems](#)
- b. Please use the above resources to build two recommender systems. The first one is to recommend the top 5 movies based on a given movie title. The second model will estimate the rating of a movie based on the user behaviour.
- c. You are allowed to use the Surprise python package, Scikit-Learn, or your own implementation. “NMF is sufficient or whatever you desire”
- d. It’s expected that your justification will be clear about the chosen hyperparameters and the mechanism of the chosen algorithm.

2. Topic Modelling

- a. Given the following resources
 - i. [Gensim Topic Modeling - A Guide to Building Best LDA models](#)
 - ii. [Topic Modelling in Python](#)
 - iii. [Topic Modeling Articles with NMF. Extracting topics is a good... I by Rob Salgado](#)
 - iv. [I notebook.community](#)

- b. Choose either the twitter dataset or the newsgroup dataset or combine them used in the above resources “i” and “ii”. It’s your own choice but you should describe the merits and demerits of combining the two datasets if you do so.
- c. Build a topic modelling algorithm using NMF model either using Scikit-Learn or your implementation or any other library.
- d. Given a sentence, create a function that will return the top 3 topics using NMF.
- e. Compare the performance of the NMF model against LDA “Latent Dirichlet Allocation”. Choose the appropriate metric. “Perplexity, Pearson, or Coherence, ...”
- f. You are not asked to describe the inner steps of LDA at all.

3. Anomaly detection

- a. Given the following resources
 - i. [ODDS – Outlier Detection DataSets](#)
 - ii. [Credit Card Fraud Detection](#)
 - iii. [2.7. Novelty and Outlier Detection – scikit-learn 1.0 documentation](#)
 - iv. [Welcome to PyOD documentation! – pyod 0.9.4 documentation](#)
 - v. [Anomaly Detection in Python with Gaussian Mixture Models. I by Agasti Kishor Dukare](#)
 - vi. [A Simple Way to Detect Anomaly. When the number of observations in one... I by Abby Yeh](#)
- b. Using a GMM model, please detect the anomalies in either of the following datasets: “Obligatory”
 - i. A multidimensional dataset from the above resources “i”.
 - ii. The credit card fraud detection dataset in “ii”.
- c. You can use Scikit-Learn outlier detection models or PyOD models. “Bonus”
- d. Please quickly describe the chosen algorithm.
- e. Benchmark your outlier detection model against a supervised approach of your own choice. The AUC will be a good metric however choose your own metric if you want to with a sensible justification.

4. Matrix Decomposition

- a. Given the following resources
 - i. [In Depth: Principal Component Analysis | Python Data Science Handbook](#)
 - ii. [Linear Discriminant Analysis](#)
 - iii. [Principal Component Analysis](#)
 - iv. [How to add noise \(Gaussian/salt and pepper etc\) to image in Python with OpenCV](#)
- b. Create a notebook and experiment with PCA if it can handle gaussian noise and salt and pepper noise.
- c. It would be good to plot the number of required components against the level of impurity or noise.
- d. Write down your conclusions and insights and of course you can use the set of images that you captured or available image datasets such as MNIST or any easy image.
- e. Linear discriminant analysis suffers from multicollinearity. Generate a highly correlated dataset with its labels - this should be fun - and try to decorrelate the features using matrix decomposition techniques then fit the data against a linear discriminant model.

5. K-means suffers a lot when the dataset is not flat

- a. Please apply the necessary techniques - perhaps such as kernels - on the moons dataset found in Scikit-Learn to make it possible to cluster the dataset successfully. A bonus question: is there a method in Scikit-Learn other than Spectral Clustering - like the DBSCAN method which depends on the Nearest Neighbour techniques or an Agglomerative method - that can do this trick for you? Explain please. "Read the user guide of the clustering page on Scikit-Learn website"
- b. Choose two unsupervised metric techniques and evaluate the K-means output before and after applying the techniques in point "a". One of the techniques should not require a label to be evaluated.

Deliverables

- Each coding task should be delivered in a jupyter notebook.

- Your results must be reproducible.
- Each coding cell should be described in an html or a markdown cell.
- The code should be formatted using either black or flake8. **“Important”**
- Your code should be readable and follow **the single-responsibility principle**.
- Each function should be documented. For example, two sentences that describe the function, description of the input variables with their types, and what the function returns.
- Don't use classes unless necessary and use functions to keep your code maintainable.
- In each notebook you should explain the pipeline quite extensively. Moreover, you should be aware of each line.
- Your notebooks should look clean and ready for publication on github.
- Regarding the Topic Modelling task:
 - You should learn how to preprocess and prepare a corpus for an ML pipeline.
 - You should learn to use the re package.
 - You should learn how to use pandas advanced methods such as apply.
 - You should expect questions about the preprocessing steps you chose for this task. Anyway you will describe them in your notebook.
- In the anomaly detection task, if you choose the credit card dataset, It's okay to cheat and choose a good supervised learning pipeline with an extremely good fitting on the testing data from the submitted tasks on kaggle. **But you should wonder about the possibility of creating a pipeline of a GMM and a supervised learning model such as the Logistic Regression per se. Will it actually help? This is a bonus question.**
- Combine the notebooks in a zip file named ML_2_{your name}.zip
- Send me an email that contains the zip file and the presentation as described below. The subject should be ML 2 Assessment ITI Suez Canal Branch.
- The **Deadline** is 14 days starting from the moment you receive the task.

Grading



- Task 5, 4, and 3 are mandatory. They will contribute a total of 45%. Each will be weighted equally.
- Choose one task from {1, 2}. It will contribute a total of 30%.


- You should prepare a presentation - along with your jupyter notebooks - that contains an overview of what you have learned and your findings. The presentation will be used in your oral delivery of the tasks. It will contribute a total of 25%.
- The grading will be based on the validity, clearness, reproducibility, soundness, and the justification of the jupyter notebook contents.
- Any answer to the bonus questions will give you a 10% increase over your final score.
- You are allowed to copy code. Actually, **the resources contain enough information to start coding right away**. But any lack of information regarding any line used in the code will be met with severe penalties!
- The topic modelling task is a bit hard and it will be evaluated carefully so don't worry if you are overwhelmed by it.
- You should score 70% or higher to pass the assessment.

Learnt Lessons

1. Developing and evaluating clustering techniques on flat or irregular datasets.
2. Developing and evaluating Recommender Systems.
3. Developing and evaluating Topic Modelling techniques.
4. Be familiar with NLP preprocessing steps.
5. Understand and explain Matrix Decompositions techniques such as PCA, LDA.
6. Explain why PCA can reduce noise and decorrelate the features.
7. Describe why NMF can learn latent variables.
8. Understand why GMM can provide a soft margin for clustering algorithms as SVM does in the case of supervised learning problems.
9. Be familiar with the PyOD library and the anomaly detection concept.
10. Explain briefly the EM algorithm and its usage.
11. By finishing ML 1 and ML 2 you should be easily able to finish this specialization: [Machine Learning Specialization](#).

Additional Resources

1. [Stanford CS229: Machine Learning I Autumn 2018](#)
 - a. Lecture 14, 15
2.  Expectation Maximization: how it works
3.  Clustering (4): Gaussian Mixture Models and EM

4.  Lecture 15.1 – From PCA to autoencoders – [Deep Learning | Geoffrey Hinton | UofT]
5. [CSC 2515 Lecture 9: Expectation-Maximization](#)
6. [Lecture 18: Mixture modeling](#)
7. [The Elements of Statistical Learning](#)
8. [Collaborative Filtering: Matrix Factorization Recommender System](#)
9. [Non-Negative Matrix Factorization for Dimensionality Reduction – Predictive Hacks](#)
10. [Hierarchical Clustering of Countries based on Eurovision Votes – Predictive Hacks](#)
11. [Python: Advanced Guide to Artificial Intelligence](#), code:
[PacktPublishing/Python-Advanced-Guide-to-Artificial-Intelligence](#)
12. [Non-negative matrix factorization \(NMF\) | Python](#)
13. [A TUTORIAL ON NONNEGATIVE MATRIX FACTORISATION WITH APPLICATIONS TO AUDIOVISUAL CONTENT ANALYSIS](#)
14. [The why and how of nonnegative matrix factorization | the morning paper](#)

“Please remember that I don’t want perfection. I only want to see your professionalism and determination.”

“Please contact me and ask me a lot. It’s my duty to help you step forward.”

“By Solving all the tasks you’ll have a very nice portofolio. So, please try them all. It’s worth it!”

Here’re my contact details:

Kareem H. El-Safty

Research Assistant in Quantum ML | Wigner Research Centre for Physics

AI Team Leader | DevisionX

LinkedIn: [Kareem H. El-safty](#)

Email: [kareem el-safty](#)

Good Luck