LDATA2010 Project: Visualization Dashboard

Data is becoming increasingly important in our society, while datasets are becoming larger and larger every day. Visualizing those datasets is an important part of data science, and various algorithms have been designed to help the visualization of big data.

In this project, you will have the opportunity to create a *specialized* visualization dashboard. This includes getting familiar with different chart types, exploring different visualization algorithms, depicting multiple views of the data, etc. Obviously, the aim of the project is *not* to develop a generic visualization software. Instead, you are asked to implement only the features necessary for the dataset provided, with a careful look at user experience and user interaction.

1 Datasets

We propose three different datasets for this project. You must **choose one dataset** for which you will create a visualization tool. The datasets are briefly introduced below.

1.1 CelebA with embeddings

Dataset Context

Computer vision is a branch of computer science that aims to develop algorithms that allow programs to extract usable information from images or videos. This information can then be used for various applications such as autonomous driving, assisting radiation delivery in the medical context and facial recognition. Recently, advances in machine learning and in particular with deep neural networks (DNN) brought impressive results. These models transform the input data (ie: an image) into a *latent representation* (or embedding), which is an abstract representation of the image. For instance, when looking at faces, a DNN might extract information such as "has glasses", "is smiling", "colour of the hair" etc... The "reasoning" that DNNs have when doing inference is notoriously hard to understand as humans, the models are generally seen as "black boxes", this can be problematic in the context of sensitive tasks such as medical applications. Understanding the latent space of such models is an active research topic.
This dataset contains latent representations of images extracted from different models, the aim is to study the internal representation of images that these models extract, using various data visualization methods.

Each dataset contains multiple metadata columns and finishes with several "embedding" columns (512). The "image_name" column is the name of an image in the "img_celeba" directory provided with the CSV files. The "id" field contains a unique identifier for each celebrity (you

might want to check if you can cluster the persons using the embeddings provided). The remaining fields categorize each image using binary features (examples: "Eyeglasses", "Beard", "Smiling", ...)

You will receive two datasets, where each dataset contains the latent representations from a different model. We used models provided by InsightFace [1] (small and large) to compute these embeddings.

1.2 Zebrafish

Dataset Context

Advances in measuring methods allow biologists to collect the gene expression levels within single living cells, for hundreds of genes at once. In recent years, these techniques have rise to large transcriptomics datasets, consisting of thousands of observations (cells) and hundreds of genes (variables). To analyze the data, biologists frequently use modern data exploration techniques such as clustering, dimensionality reduction, and more classical statistical analysis conjointly. This dataset was collected in the tissues of developing zebra fishes, to study the differentiation process between cells.

To support your data visualization, two sets of colors/labels are proposed: the primary labels colorize cells depending on their type, and the secondary label indicates the time after fecundation at which the sample was taken. There are 63530 cells and 1000 genes in total. This dataset comes from a papper of Daniel E Wagner et al. [2]

1.3 Breast Cancer Gene Expression Profiles (METABRIC)

Dataset Context

Breast cancer is the most common cancer among women, impacting 2.1 million annually and causing over 600,000 deaths in 2018. Despite similar disease stages, patients often experience different outcomes, which highlights the need for a better understanding of the underlying factors.
This dataset focuses on gene expression in breast cancer tissues, which measures the activity of genes within cells. By comparing the gene expression in cancerous and normal tissues, researchers can gain insights into disease progression and patient prognosis.

This dataset includes 31 clinical attributes, m-RNA levels z-score for 331 genes, and mutation in 175 genes for 1904 breast cancer patients. This dataset aims to study the relation between the patient's pathology (cancer type, cancer stage, ...) based on the different features (attributes, scores, ...). This dataset comes from Kaggle [3]

[1] https://github.com/deepinsight/insightface
[2] https://pmc.ncbi.nlm.nih.gov/articles/PMC6083445/_ad93_
[3] https://www.kaggle.com/datasets/raghadalharbi/breast-cancer-gene-expression-profiles-metabric

2 Exploratory analysis

Your first task after selecting one of the three datasets, is to explore and understand the purpose and contents of the file. Without this knowledge, it will be difficult to create a dashboard appropriate to the dataset. However, there's no need to become an expert!

You should also be able to answer basic questions like: Is the dataset complete? What should you do with missing data? How should you deal with categorical values?

Before starting to design your visualization tool, you should ask yourself what questions a user might ask himself. What does the user want to discover from the data? What are the important relationships to know?

This can be done in various ways, you are free to take any direction that you find suitable for this data set (comparing features from different models, comparing features within the same latent space, relations between the features and the original images, relations between the images by looking at the features, using time-related information, ...)

3 Basic Features

You will develop a software with the following features:

1. A user will be able to display multiple views of the data, to explore different facets of its structure using different chart types.
2. You will enable the user to compute some basic properties and metrics of the dataset. The user will have the possibility to highlight these properties on one or multiple plots.
3. The user will be able to filter the data according to some values or attributes. This should update all the visible plots, metrics and properties. (You might want to force the user to select filters before computing some heavy algorithms)
4. The user should be able to select some data in one plot to update the other plots accordingly.
5. The user should be able to visually see how the variables are linked.

User Interface

While the user interface (the placement of views, charts, buttons, ...) is left up to you, be careful that it is an integral aspect in the design of the application.

4 Clustering and Dimensionality Reduction

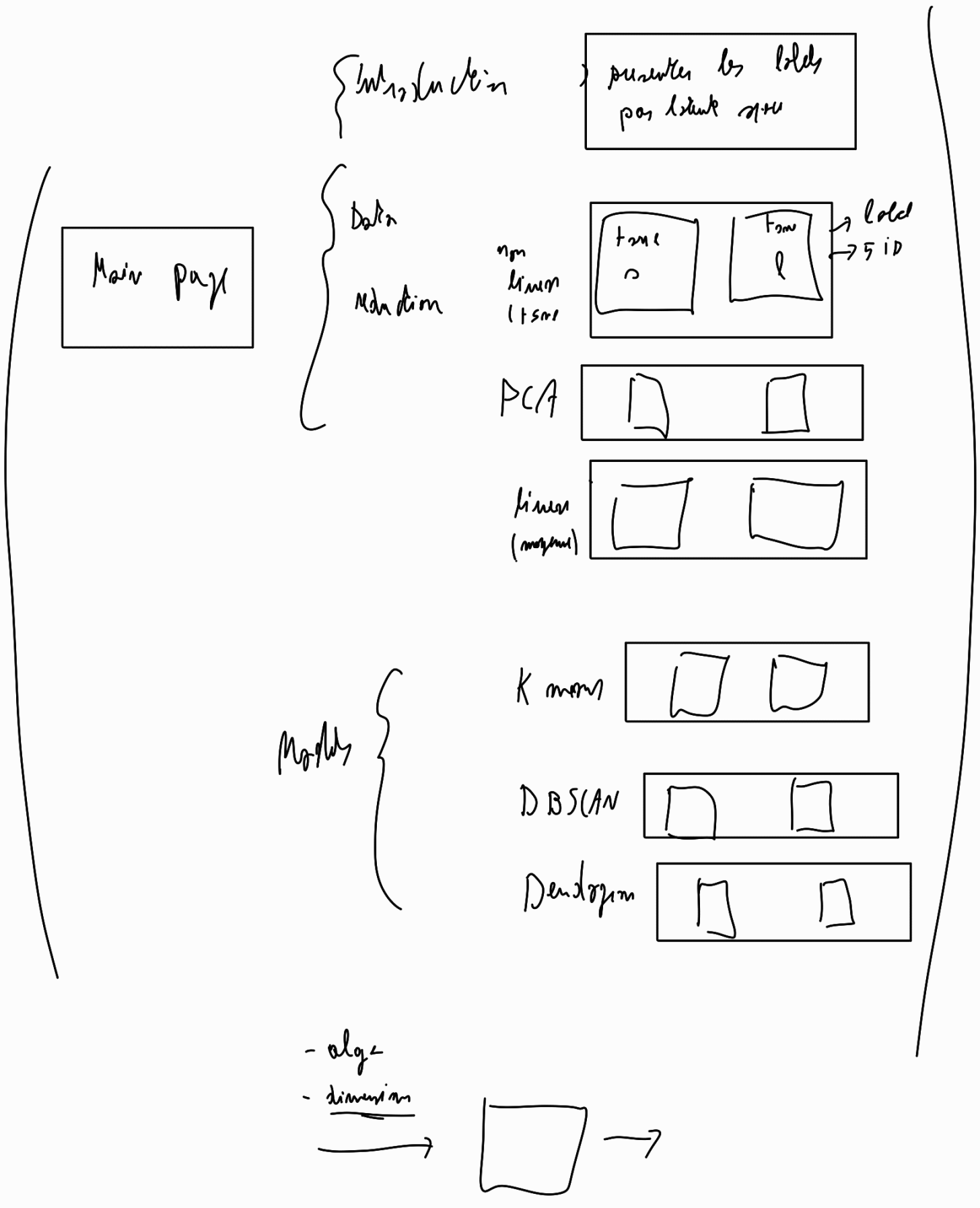
Once you have implemented the global plotting capabilities of your interface, you will now have to dive deeper in an advanced aspect of information visualization. For this, we ask that you implement different views of the data using clustering and dimensionality reduction algorithms:

- Clustering : What can you use clusters for? How will you show them in your interface? Can you use cross-filtering with a hierachical clustering? How does changing the parameters of the algorithms change the clustering?

AIM OF THE PROJECT:

↳ is the latent space of the large model better than the small model to discriminate non-latent labels?

- K means ✓
- hierarchical clustering
- density based model ✓ → DBSCAN → assigns the labels
- linear data reduction → mean of the labels
- non linear data reduction → TSNE


Main page
Data
Reduction
PCA
Kmeans
DBSCAN
Dendrogram
- algo
- dimension

  - Implement K-Means
  - Implement a hierarchical clustering algorithm of your choice
  - Implement a density-based clustering algorithm of your choice

- Dimensionality reduction : Can a clustering algorithm help you in choosing an effective DR algorithm? Do you detect visual clusters when you visualize a DR algorithm? What parameter should you choose and which should you enable the user to change?
  - Implement a linear DR algorithm
  - Implement a non-linear DR algorithm of your choice

Interactivity between the views

Data exploration is an interactive process: there is a back and forth between the user and the algorithms via the UI. Thus, interactivity is a central component of your software: were it is adequate, we expect the various views to be interactive and linked between one another (for instance, if showing a conditional distribution, the user could change the condition by selecting points or variables in another graph). How can you make sure interactivity doesn't have a too high impact on the responsiveness of your interface ? What strategies can you use ?

5 Report

In addition to your software, you are asked to provide a small report (maximum 7 pages) detailing the features that you have implemented. In particular,

- You can write your report as a user guide for your software.
- Explain and justify your design choices.
- Cite the toolboxes, sources and papers that you employed. There is no restriction on the libraries and sources you use nor on the papers that you read, but you have to cite them.
- Reasonably detail the algorithms that you have employed (e.g. by providing an overview of each one of them without the practical implementation details) and justify why you chose them.
- Provide examples on how to use your software, illustrating its capabilities in terms of scaling, interaction and visualization.
- Give some ideas on how to improve your software. Which features might be worth implementing in future versions? How could you make your software more scalable?

6 Project Structure

The project will be structured in three parts, with a deadline and a deliverable for each part.

6.1 Global analysis and first draft of final UI

- Deadline : week 9
- Deliverable : An first analysis of the data using standard plots as shown in the exercise session, as well as a draft of your final software. We do not ask for an actual software implementation for your draft, a drawing showing the various planned features of your software suffice.
- One-on-one sessions : In order to start on good foundations, each group will have a (15 minute) one-on-one with a teaching assistant during week 9. (Registration will be on Moodle). During this meeting, you should already have :
  - An understanding of the dataset selected;
  - A User Interface helpful for exploratory analysis
  - A sketch/draft of the final user interface. You might for example use pen-and-paper or `https://excalidraw.com/` to draw the layout of your application.

6.2 Final Deliverable

- Deadline : 13/12/2024
- Deliverable : Complete project and report. (Global analysis and Clustering and dimensionality reduction)
- Project presentation : During the exam, you will be asked to present your visualization tool.

7 Practical information

- **Groups:** You can complete the project alone or by groups of 2 students. Please join a group on Moodle (even if you're alone).
- **Programming language:** you can use the one you prefer (Python, Matlab, R, etc.), but Python is recommended. You can use all the toolboxes, packages, modules, etc., that you find relevant. You can use toolboxes to help you design the interactive user interface.
- **Final Deadline for the project submission on Moodle:** Friday December 13, 15pm. Submit one .zip file per group, containing report and code.
- Do not hesitate to ask questions to the teaching assistants before or after your planned one-on-one, for instance to define what you plan to implement, the model metrics that you could evaluate, etc.

installation of the software

Don't forget to add a requirements.txt file containing the necessary libraries, one way to build requirements.txt is by using *pipreqs*. The software should be easy to install at the moment of evaluation.
Also, all file paths should be relative, not absolute!

- bar charts
- line charts
- scatter plots

⚠ no unnecessary elements:

↳ excessive colors
↳ 3D effects
↳ distractions

→ axis labeled
→ legends
→ annotation
→ titles

- effective use of colors

- Data - Ink ratio: no overload with non-essential graphical elements

interactions:

- no lag !!!
- no memory
- cool without overcrowding

⚠ color blind:

→ green and red / green brown / blue purple
→ underline not bold

↳ WCAG used !