UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Complex Data Visualizer

Narcís Terrado

Bachelor thesis
Specialization in Computing

Director: Lluís A. Belanche Muñoz
GEP Tutor: Eguiguren Huerta Marcos

March 2021

Contents

List of Figures

List of Tables

1 Context and scope

1.1 Introduction and contextualization

Given that the human mind is very visual, data visualization has been a need since the times when humans strived for survival. They kept statistics of how many animals and what kind they caught and envisioned hunting strategies [12]. We've come a long way since then, and we created *some* methods to improve the quality of the visualizations that we use, making them more useful. In the last years, we used the computational power to make data visualization available to almost every field imaginable.

The need to visualize data has not declined with time, on the contrary, given the amount of data generated nowadays thanks to computers it may be arguable that it is more needed than ever. We say that because lately, the amount of data generated by organizations around the world has increased greatly, as a result, the amount of data available has also increased dramatically. Data visualization is concerned with the design, development, and application of computer generated graphical representation of the data. This graphical representation is used in a vast amount of ways to make decisions, discover patterns, comprehend information and form an opinion.

As we said earlier, the human mind is very visual, it is much easier for our brain to understand graphically represented data than it is to understand raw numerical data. We use data visualization to form a visual representation of information, taking advantage of the capacity of the human eye to detect information from pictures and illustrations. With data visualization we shift from numerical reasoning to visual reasoning, which we are much more prepared to do and do faster[32].

With the arrival of newer technologies and higher computational power the data generated has become bigger and more complex, there's more rows and more columns in our datasets and there are more types of data being used so the techniques required to graphically represent the data have become more complex too.

To represent data with a high number of dimensions is complicated, and without the proper tools it is hard to make a visualization in which the user can extract the correct information. That's why one of the biggest challenges in data visualization is to find general representations of data that can display multiple variables at the same time.

1.1.1 Context

This is a Bachelor Thesis of the Computer Engineering Degree, specialization in Computing, done in the Facultat d'Informàtica de Barcelona of the Universitat Politècnica de Catalunya.

1.1.2 Concepts

To fully understand the scope of this project we first need to define some concepts. We will talk more in depth about them later in the work, but for now we introduce the concepts.

Data
Generally, data is described as distinct pieces of information formatted and stored in a way that serves a purpose. We could define data as information transformed in a way that we can use.
When we talk about complex data in this work we usually refer to high dimensional datasets. A high dimensional dataset is commonly modeled as a point cloud embedded in a high-dimensional space, with the values of the attributes corresponding to the coordinates of the points[34]. All these factors make the data more difficult to operate, and so, it is more complicated to transform into something we can use and understand. We will also limit our scope to table-based datasets, forgetting about graph or network datasets.

Data Visualization
Data visualization is the presentation of data in a graphical format so that it is easily understandable. It is a way to present information in a certain way so that we can identify and explain patterns. It can be roughly categorized into two applications[18]:

- **Exploration**
  For this application, usually carried out by data analysts, many graphs will be used to reveal interesting and important features in the data. For this part, a high amount of interaction with the dataset is needed,

many plots must be created, many modifications like sorting or rescaling are to be performed and performed fast in order to not disturb the data analyst train of thought.

- **Presentation**
  Once the exploration phase is over and key findings have been done in a dataset, it comes the time to present this findings to a broader audience. This graphics usually aren't interactive as they have usually have to be suitable for printing. High dimensional plots are not often used given the complexity to understand them.

Dimensionality Reduction
Given that it is not feasible to plot more than three dimensions, there are methods that allow to represent more than two dimension graphically and others that reduce the dimensionality of the data to two or three. The later methods objective is to represent the data in a lower dimension while keeping the relevant information. Methods for dimensionality reduction fall into two categories, linear and non-linear methods. As the name implies, the linear dimensionality reduction methods find a linear relation between variables to reduce the dimensionality. Some examples of linear methods are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Multidimensional Scaling (MDS) [19]. The non-linear methods find non-linear relations in the data and are a bit more complicated but the result still the same, reduce the dimensionality of a dataset in order to properly visualize it. Some non-linear methods are the kernel-PCA (a modification of the linear method PCA that uses a kernel to find non-linear relations) and t-distributed Stochastic Neighbour Embedding (t-SNE).

1.1.3 Problem to be solved

As we have seen in previous sections, data visualization is not a new topic. However, it is in constant development and it advances with other fields, as it serves as a support. In this thesis we want to develop a web based complex data visualizer. We also know that this is not new, but we want to make our version tackling some of the problems that we talked about in previous sections. One of this problems would be high dimensionality in data, which we saw that it can be approached in two different ways: User Interaction and

Dimensionality reduction.
One of the goals of this project is to make an interactive, usable and pretty web User Interface for data visualization. This will tackle the part of User Interaction to make the complex data usable. We will also allow the use of dimensionality reduction techniques to the user, making it easier to use methods that are sometimes complex to use.

A comprehensive breakdown of the objectives can be found in section 1.3.1

1.1.4 Stakeholders

There are not many parties involved in this project. They can be categorized into two main groups depending on the interaction with the project in it's development.

The **tutor** and **researcher** are stakeholders that have direct interaction with the project development. The **tutor** it's Lluís A. Belanche Muñoz, professor in Facultat d'Informàtica de Barcelona and part of the Soft Computing Research Group (SOCO). He will be guiding and leading me through the development of the thesis. He is the one who proposed the project, so he has been the one to define most of the requirements. As he is investigating at UPC, this project will probably be of use for him, given his areas of research. The other stakeholder directly involved is the **researcher**, Narcís Terrado González, me, who is responsible for the planning, documenting and developing the project.

For indirect stakeholders we have the **scientific community**, who, just like the tutor of this project, may be interested in having this kind of data visualizing took to aid it's research. We could also count **companies** that use some kind of data visualizer as indirect stakeholders that receive benefits from this project

1.2 Motivation

1.2.1 Previous Studies

In the past decade a variety of approaches have been introduced to express the information found in high dimensional datasets. Some of this approaches may be dimensionality reduction, visual encoding and interactive exploration.

S. Liu et al. have gathered most of the advances in this field in ”Visualizing High-Dimensional Data: Advances in the Past Decade”[34]. We will focus only in some of the advances made because there are a lot of them that are not directly related to this work.
The article talks about the different dimensionality reduction methods that we wrote about earlier and how do they work. We will not explain them as it is not within the scope of this work to explain every method thoroughly. We will talk, however, about some of the improvements and extensions made to this methods. In the case of Principal Component Analysis (PCA), we have interactive-PCA, which introduces a system that visualizes the results of PCA and allows for some interaction and can be used to better understand the PCA method and the dataset itself.

The article also talks about how user interaction is playing an important role in the development of high dimensional data visualization: Interactive filtering, zooming, distortion, linking or a combination of them have been adopted as part of the exploring process of data visualization.

A lot of other methods can also be found in the article, but they are way too many to mention all of them. I will, however, mention some new methods and fields of study that have surged in the last years.

Topological Data Analysis (TDA), which is a new field of study [29],has provided efficient and reliable feature-driven analysis and visualization capabilities. It provides a meaningful abstraction from high dimensional data.

In the field of Machine Learning, data visualization has made efforts to understand the *black box* that Machine Learning models usually are. Visualization systems of neural networks that make it’s design more efficient and understandable[21]. Visualization methods for interactively construct and analyze decision trees[41].

1.2.2 Justification

As we commented many times earlier in previous sections, data visualization is useful in a lot of fields of study. It helps other fields progress and at the same time, other fields make data visualization better.

The methods that we commented so far are the theoretical frame to make complex data visualization possible. We have means to achieve our ends, and so, we must put this theory into practice. In order to make a complex data visualizer we need the methods we talked in previous sections to serve as a base for us to build our visualizer on. The use of some of the methods commented are very important and will make us a better tool for visualizing data.

There are some tools a data analyst could use for data visualizing, but they probably require them to program every chart and graphic themselves in order to be able to extract useful information from it. The *matplotlib* Python library is widely used for data visualization but the interaction that the graphics created allow the user is usually not enough to properly analyse the data given it's complexity. With the result of our project a data analyst could easily interact with the data to extract all the useful information needed.

1.3 Scope

In this section we will talk about the scope that we want to give the project while we enumerate and explain the objectives and requirements that it must have. In the end we will also list the potential risks that we can encounter while doing this project.

1.3.1 Objectives

As we commented briefly in previous sections, the main objective of this project is to develop a functional data visualizer able to visualize complex data to an extent. To accomplish this objective we can divide the objectives into two categories:

Theoretical part

- Research state-of-the-art of data visualization
- Research dimensionality reduction methods. This includes both linear and non-linear dimensionality reduction methods
- Research about User Interface design

Practical part

- Design a web based User Interface for the data visualizer with Dash
  - Learn to use Dash
  - Design the UI

- Use data visualization techniques researched to properly represent data graphically
  - Research data visualization techniques
  - Implement the data visualization methods
- Use dimensionality reduction methods for the proper representation of complex data.
  - Research dimensionality reduction methods
  - Implement dimensionality reduction methods

We can assume that the practical part will be more important than the theoretical part, that's because this thesis is not focused on research but on the practical application of it. We are not going to implement manually any of the methods commented earlier, we must, however, know how they work and what task are they accomplishing in order to use them properly.

1.3.2 Requirements

The tutor for this project has made some requirements in order to ensure the quality of the project. The final program must allow the following points:

- User must be able to upload data (in any of the most typical formats)
- User must have some flexibility to customize and modify the visualizer
- Program must be able to reduce the dimensionality of a dataset via linear and non-linear methods in order to properly visualize it.
- User must be able to choose the kernel function to use from a predefined set of them.
- Program must let choose which graphical representation the user wants from a predefined set of them.
- Program must be able to visualize a predefined subset of complex data.
- Program must allow the user to interact with the data.

1.3.3 Potential obstacles and risks

During the development of this project there may be some risks or obstacles that may prevent the correct development of it.

- **Deadline of the project**

  There is a deadline for delivering the project, hence, the duration is defined and unavoidable so we must define a distribution of the work that allows us to finish the project correctly and adapt in case that's not possible.

- **Inexperience**

  It will be the first time I'll be designing a web interface and a data visualizer and I'll be using a framework (Dash) that I have no experience with. This can complicate the development of the project because it may take more time than I think to learn these topics.

- **Coronavirus**

  The risk of Coronavirus will probably not be gone by the time I finish this project, so the risk of getting confined it's still there.

1.4 Methodology and rigor

Defining the correct methodology to be followed during the development of the project is key to correctly develop it.

1.4.1 Methodology

The methodology chosen will be the Kanban methodology. With it we can manage the state a task is in and how is a task completed.

Kanban is a Japanese word that translates to "visual cards". This methodology is about using visual notes representing a task to do. A task will be in one of the defined stages, and each stage corresponds to a column in a board. So we will be keeping track of tasks with a board with the following stages:

- **To do**

  In this column there will be all the tasks that we have defined already but have not been started yet.

- **In progress**

  All the tasks that are being developed at the moment will be in this column. This column must not contain a lot of tasks, that would defeat the purpose of this methodology.

- **Testing**

  The tasks that have been developed but have not been tested yet will fall in this column. Some tasks won't have a testing stage and will be put directly into the last one.

- **Completed**

  All finished tasks will be here as a log of what we have already completed.

We will be using a virtual Kanban board with Trello, a web application that allows us to simulate the cards and the board in a Kanban. Some other column may be added to the board.

1.4.2 Monitoring tools and version control

We will be using GitHub as a tool for version control. It will allow us to keep a record of all the work being done with the commit records. We will also be able to keep a copy of our project in the cloud, which will allow us to access it easily and also secure the development of the project in case we lose some progress to some unexpected event.

2 Project Planning

The development of this project will last approximately 500 hours until late October approximately. The date for the oral defense is to be determined yet. The work per day is going to be about 4.5 hours, but there may be some discrepancies due to the fact that I may not be able to do all hours some day.

2.1 Task definition

Now we will present all the tasks that will be carried out during the project. We give for each one a description, the estimated duration and dependencies with the other tasks. Table 1 summarizes the task information and Figure 1 illustrates the schedule.

This section of the project is probably one of the most important because we define a temporal scope to work with. We also define the tasks and plan its distribution in time and in relation to each other. This is something that will help us throughout the project because we will be able to have an idea of how we are doing.

Project management

- **ICT tools to support project and team management**

  To make our job easier we will need to use the best technology available so we need to research different types of software for different types of tasks.

- **Context and scope**

We will indicate the general objectives of the project while contextualizing it and justifying the decisions made.

- **Project Planning**

  In order to complete the project in time we will need to make a good planning of the tasks to do. Doing this we will know which tasks need more of our focus and which ones are the most important.

- **Budget and sustainability**

  This will help us know what's the true cost of the development of our project and the impact it can make. We will make a budget and analyze the sustainability of the project.

- **Final project definition**

  We will modify the previous sections with the feedback given to us in order to improve the quality of the project.

Study of the state-of-the-art
Even though this project's theoretical part is not the most important we still need to do research in order to understand the concepts we need to apply later on. For this we will perform the following tasks

- **Data visualization**

  We will need to do research on data visualization in order to be able to develop a data visualizer. We will use the knowledge created while researching to improve the visualizer. In this task we will also test if we understood the concepts by creating some very simple data visualizers. We assigned 36 hours to this task because we need to understand the state-of-the-art to successfully implement later on.

- **Dimensionality reduction**

  We will also need to do research on dimensionality reduction in order to be able to apply it to our program. In this task we will also check our understanding on the field trying some of the methods studied. We've also assigned 36 hours on this task for the same reason, we need to understand correctly how it works in order to use it.

Practical implementation

- **Learn Dash**

  We will be taking some time to learn Dash, the python framework that will help us build the web based visualizer. We have to learn it beforehand in order to familiarize myself with it before starting with the visualizer. This will also include learn CSS, which Dash and I have never used. It is quite a few work, so we assigned 50 hours to this task to ensure we have learned enough to complete the rest of the tasks.

- **Use data visualization techniques**

  To develop the visualizer we will need to apply the techniques we learned during our research in data visualization. This will take a good amount of time because I have no experience in this field, so we assign 50 hours to this task to ensure we end up using these techniques successfully.

- **Use dimensionality reduction methods**

  We will use the dimensionality reduction methods we learned. As we have no experience in this, we will assign 50 hours to properly use this methods.

- **Implement the use of some kernel functions for dimensionality reduction**

  For the dimensionality reduction we will let the user choose between some kernel functions, and this needs to be implemented correctly. Once we learned the dimensionality reduction methods we will need to implement some kernel functions for the user to use. For this, we assigned 25 hours as we will already have some knowledge after researching the state-of-the-art and implementing the dimensionality reduction methods.

- **Implement user interaction**

  The user needs to interact with the data, we need to implement this interaction. As this does not seem extremely complicated to make with Dash, we assigned 25 hours to it.

- **Design the UI**

  In order to make our program better we will be dedicating some time to design a fast, clean and usable UI. We don't think this will take much time, so we assign 25 hours to it even though we have no experience.

- **Testing**

  We need to assign some time to testing the program that we are building in order to make it usable. For this tool testing is quite important to make the experience enjoyable, so 50 hours seems fit.

Given that our thesis is not about research we don't need to extract a direct conclusion from it, so we will have no tasks for experimentation or analysis.

There is two implicit tasks to be done, the documentation and the preparation for the oral defense.

2.2 Resources

To carry out a correct development of the project we need some resources. This resources have been divided into four groups

2.2.1 Human resources

For this project we have three people that fall into this type of resources. First, we have the researcher, who is responsible for planning, developing and testing the project. There is also the director of the project which is responsible for leading and guiding the researcher for the correct development of the project. Lastly, we have the GEP tutor, who is in charge of helping the researcher do the planning for the project during the first month of its development.

2.2.2 Hardware resources

In this group of resources we have our computer, which is quite essential for the development of the project. We will be using a *TUXEDO InfinityBook Pro 15* with 16 GB of RAM, Intel(R) Core(TM) i7-10510U. We also have to take into account all the resources connecting the laptop to the network (e.g the router). We also use a mouse and maybe some USB memory to share files.

2.2.3 Software resources

We will use multiple resources for this group, and each will fill a specific need in the development. First, if we have to schedule a meeting with the director we will use Google Meet. To manage the code of the project we will use GitHub, where it will be secure and available in case of a local loss.

For the programming part we will use Python as our primary language given that we will be using the Dash framework to create the web interface. We may use other programming languages for other purposes but for the moment we will keep Python as the only language used. For the documentation we will use Overleaf or TeXmaker as our text editor. We may swap to one or another depending on the size of the document.

2.2.4 Material resources

In this kind of project there is always the need to gather knowledge, we will be doing that from books and papers.

2.3 Risk management

We introduced the potential risks for the project in section 1.3.3. Now we will also assign a risk level to each risk and we will present how can we solve them once we encounter them.

- **Deadline of the project [High risk]**

  In case we have made a bad temporal planning of the tasks to complete the project in time, we will need to readjust the planning once we are more deep into the project. If we still don't meet the deadline, we will create a task of 20 hours to solve the problem we face.

- **Inexperience [Medium risk]**

  We do not have expertise in every piece of software that we will use, and that may be a risk. We planned the tasks so that we assign a lot of time for learning the tools that we need to use, but it may not be enough. In case of seeing that we are not reaching the required knowledge to finish the project we will assign more time to learn the software needed. If we cannot assign more time for some reason we will reduce the hours of some other task to make it possible.

- **Coronavirus [Low risk]**

  Given that it's been some time since the start of the pandemic, we are prepared to continue working on the project even if the pandemic causes the country to go under a confinement again.

We have already overestimated the amount of hours in the majority of the tasks in order to avoid some of the risks.

ID	Name	Time(h)	Dependencies
T1	Project Management	**94**	
T1.1	ICT tools to support project and team management	4	
T1.2	Context and scope	25	T2
T1.3	Project Planning	10	
T1.4	Budget and sustainability	15	T1.3
T1.5	Final project definition	20	T1.2, T1.3, T1.3
T2	Study of the state-of-the-art	**72**	
T2.1	Data visualization	36	
T2.2	Dimensionality reduction	36	
T3	Practical implementation	**300**	T2
T3.1	Learn Dash	50	
T3.2	Use data visualization techniques	50	
T3.3	Use dimensionality reduction methods	50	
T3.4	Implement the use of some kernel functions for dimensionality reduction	25	
T3.5	Implement user interaction	25	
T3.6	Design the UI	25	
T3.7	Testing	50	
T4	Project Documentation	**76**	
T5	Bachelor thesis defense preparation	**20**	
Total		**537**	

Table 1: Summary of the information of the tasks

2.4 Gantt diagram

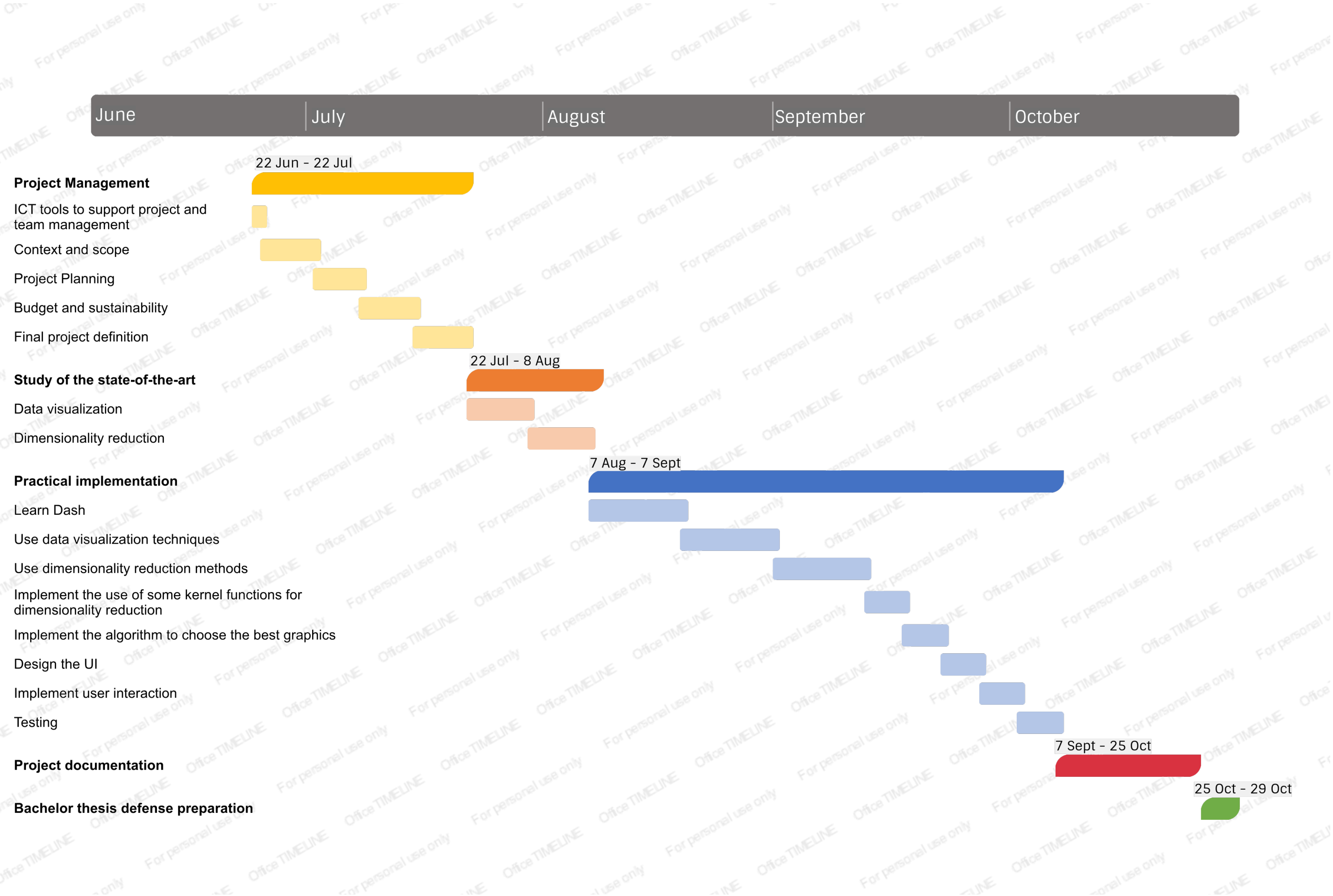


Figure 1: Gantt diagram

3 Budget and Sustainability

In this section we will lay out the economic part of the project. We will talk about the budget for the project, which will include the personnel costs, generic costs and other costs. There will also be some management control mechanisms in order to be prepared for some unforeseen event.

3.1 Budget

3.1.1 Personnel costs per activity

Here we will compute the total cost for each task defined in the previous section. The cost of one task is computed by summing the cost of the workers. Then, for each worker we will compute the cost by multiplying his cost per hour by the amount of time that worker needs to work on that task.

We can define 5 types of workers that would be used to develop this project, however, they will all be impersonated by me, the director of the project or the GEP tutor. First we have the **project manager**, who is responsible for the proper planning and correct development of the project. This role will be carried out by the GEP tutor and me. We also need a **programmer** and a **tester**, which task will consist in programming the code and verifying its correct functioning. This roles will be played only by me. Finally we will need a **technical writer** that will document everything regarding the development and the results. This role will also be played by me.

In table 2 we show the annual salary for each of this roles. And we have computed the total cost for the personnel in table 3

With this information we will be able to compute the cost per task.

Role	**Annual Salary (€)**	**Total including SS (€)**	**Price per hour (€)**	**Role played b**
Project manager	39.004	50.705,2	28,97	GEPT, D, R
Programmer	26.198	34.057,4	19,46	R
Tester	20.592	26.769,6	15,29	R
Technical writer	26.263	34.141,9	19,50	R

Table 2: Annual salary for the different project roles. Estimated hours worked per day is 1750. Information from: [1]
GEPT: GEP Tutor
D: Director
R: Researcher

ID	Name	Total Hours			Hours	
			Project manager	Programmer	Tester	Technic
T1	**Project Management**	**94**	**94**	**0**	**0**	**0**
T1.1	ICT tool to support project and team management	4	4	0	0	0
T1.2	Context and scope	25	25	0	0	0
T1.3	Project Planning	10	10	0	0	0
T1.4	Budget and sustainability	15	15	0	0	0
T1.5	Final project definition	20	20	0	0	0
T2	**Study of the state-of-the-art**	**72**	**0**	**72**	**0**	**0**
T2.1	Data visualization	36	0	36	0	0
T2.2	Dimensionality reduction	36	0	36	0	0
T3	**Practical implementation**	**300**	**0**	**250**	**50**	**0**
T3.1	Learn Dash	50	0	50	0	0
T3.2	Use data visualization techniques	50	0	50	0	0
T3.3	Use dimensionality reduction methods	50	0	50	0	0
T3.4	Implement the use of some kernel functions for dimensionality reduction	25	0	25	0	0
T3.6	Implement user interaction	25	0	25	0	0
T3.7	Design the UI	25	0	25	0	0
T3.8	Testing	50	0	0	50	0
T4	**Project Documentation**	**76**	**0**	**0**	**0**	**76**
T5	**Bachelor thesis defense preparation**	**20**	**20**	**0**	**0**	**0**
Total		**537**	**114**	**322**	**50**	**76**

Table 3: Cost for each member of the personnel

3.1.2 Generic costs

Amortization
We will take into account the amortization of the hardware resources we are using. We considered in the previous section an average of 4,5 hours of work per day during 125 days. The average lifespan of a laptop is 4 years. We will be using the laptop and so the amortization formula is the following:

Amortization(€)=Resource price $\times \frac{1}{4} \times \frac{1}{125} \times \frac{1}{4{,}5} \times$ hours used

All the amortization done in this project is from hardware because all the software that we will use is free to use. As we said in the previous section we will be using this resource approximately 560 hours. If we follow the amortization formula we can obtain the following result:

Amortization(€)=1.200 $\times \frac{1}{4} \times \frac{1}{125} \times \frac{1}{4{,}5} \times 560 =$ **298,67€**

Electric cost
The fare for the kWh is 0,1636 €[2]. We will compute the price of the electricity for the project. We only count the hours that we will be working, as we said, that's approximately 560 hours in total. The power in Watts for the laptop used is 65W So we can compute the cost by

$$\text{cost}(€) = \frac{0.1636 * 560 * 65}{1000} = \textbf{5,95€}$$

Internet cost
The internet rate for us is 54€per month. The project will last about 5 months, and we will be working 4,5 hours a day.

$$\text{cost}(€) = 5\text{ months} \times \frac{54€}{1\text{ month}} \times \frac{4,5\text{ hours}}{24\text{ hours}} = \textbf{50,62€}$$

Water cost
Water in my area costs around 30,5€per month. Using the same operation than before we can obtain the water cost for the whole project.

$$\text{cost}(€) = 5\text{ months} \times \frac{30,5€}{1\text{ month}} \times \frac{4,5\text{ hours}}{24\text{ hours}} = \textbf{28,6€}$$

Travel cost

Due to the coronavirus pandemic the travel costs have been significantly reduced, to the point where we do not need to travel at all in order to develop the project. In case of some unexpected event, we will assign some of the contingency budget to travel costs.

Work space

The project will be developed in my house located in Terrassa. The rent is 600€per month, and given that I am sharing the flat with someone else, the real cost of the space is 300€. Hence, the cost for the work space is 5months × 300€ = 1.500€

Generic cost of the project

In table 3 we can see the generic costs of the project summarized.

Concept	**Cost (€)**
Amortization	298,67
Electric cost	5,95
Internet cost	50,62
Water cost	28,6
Travel cost	0
Work space	1.500
Generic cost	**1.883,84**

Table 4: Generic cost of the project

3.1.3 Other costs

Contingencies

During the development of the project we will face unexpected events that will take part of our budget. To mitigate the impact these events can have in the development we will be assigning a fund of contingency to the project budget. Summing all the costs we have: 11.815,2€ for personnel and 1.883,84 € for the generic costs = 13.699,04€ . For the contingency we will be assigning 15% of that cost, and that will make a contingency of **2.054,8€**

Incidental costs

In previous sections we listed the possible risks that the development of this project can have and we also gave a possible solution for each one. This solutions will increase the cost of the project and so, we need to take them into account to plan our budget. The cost of each incident is computed by multiplying the price it would cost to solve by the probability that the risk occurs. Table 5

Incident	**Estimated cost (€)**	**Risk (%)**	**Cost (€)**
Deadline of the project (20 hours)	547,05	40	218,82
Inexperience (30 - 50 hours)	973	80	778,4
Coronavirus	0	100	0
Total			**997,22**

Table 5: Incidental cost of the project

3.1.4 Total cost

We can find a summary of the total cost for the project in table 6. The total cost is computed by summing all the previous sections costs.

Activity	**Cost (€)**
Personnel cost	11.815,2
Generic cost	883,84
Contingency	2054,8
Incidental cost	997,22
Total	**15.751,06**

Table 6: Total cost of the project

3.1.5 Management control

We cannot assume that we will face no eventualities and that we will fulfill the budget and time estimations perfectly. We must, then, define a model to control the potential deviations.

For every task we must compute the deviation of all the costs. We now list different formulas for computing different deviations we can face.

- **Human resources:** Caused when personnel does not produce the amount of work expected per time unit.

Human resources deviation = $\sum_{i \in pi}$(estimated cost per hour$_i$−real cost per hour$_i$)× total real hours$_i$

where *pi* refers to the personnel involved.

- **Amortization:** Caused by using a resource more or less time than the expected.

  Amortization deviation = $\sum_{i \in hr}$(estimated usage hours$_i$−real usage hours$_i$)× price per hour$_i$

  where *hr* refers to hardware resources.

- **Travel cost:** Caused by making more travels than expected.

  Travel cost deviation = (estimated number of journeys−real number of journeys× journey cost

- **Total cost:** This deviation groups the deviations on the different tasks.

  Total cost deviation = estimated general cost − real general costs

Using this list we can easily comprehend where and why there has been a deviation in the project development and how much it costs. In case of a negative deviation in the total cost we will have to use the contingency budget.

3.2 Sustainability

3.2.1 Self-assessment

Given the situation in the world we live in, where it is clear that capital has a greater importance than human lives, we need a change as soon as possible. The increment of pollution, among other things, is aggravating global warming, making it less and less hopeful for the people of the earth to have a decent future. The lack of meaningful commitments from every country to reduce this precarious situation has made it so we have less and less time to stop it and are more likely to suffer from it.

This raises the absolute need for checking the footprint of a project. This includes the social and environmental dimensions as well as the economic one. This was not clear for me at first, because I did not think that the economic factor should be as important as the others. I still not think they are comparable, given the importance I give to the environment, human lives and society, but I certainly am more conscious about this subject after the poll and doing a bit of research.

I've been a bit surprised by the large number of indicators to evaluate the different factors present in sustainability. I now think it is very important to measure the different impacts that a project does to different factors. This way we can detect problems regarding sustainability and research tools to solve them.

3.2.2 Economic impact

Have you estimated the cost of undertaking the project (human and material resources)?

In previous sections, the reader can find the economic costs of this project, including the material costs and the personnel costs. It can also be found the potential deviations there can be.

How is the problem that you wish to address resolved currently (state of the art)? In what ways will your solution economically improve existing solutions?

Data visualization can affect an enormous set of fields of study, by providing an efficient and useful data visualization tool we can improve the economic impact in many other fields.

3.2.3 Environmental impact

Have you estimated the environmental impact of undertaking the project? Have you considered how to minimise the impact, for example by reusing resources?

We have not estimated the environmental cost of the project. However, we consider this project to have a low environmental impact, given that it requires no new material resources (the ones that we will use I already own and use for personal use). It can have a high electric cost given the amount of hours I will need to be using the laptop among other things we commented in the previous sections.

How is the problem that you wish to address resolved currently (state of the art)? In what ways will your solution environmentally improve existing solutions?

As we said before, data visualization can affect many different fields, and a good tool for data visualization can reduce the costs and resources needed.

For example, if a researcher can make a good induction out of a visualization, it will not be needed to test a large amount of methods, wasting time and electricity.

3.2.4 Social impact

What do you think undertaking the project has contributed to you personally?

This will be my first big project and it will be the first time I work in the field of data visualization. The fact that I face a somewhat big amount of hours of work, has made me more organized and less prone to procrastination. It is also an opportunity to put into practice all the knowledge gathered through this years.

How is the problem that you wish to address resolved currently (state of the art)? In what ways will your solution socially improve (quality of life) existing solutions? Is there a real need for the project?

As we said repetitively, data visualization affects many fields and can indirectly address issues in other fields of study.

4 Complex Data

We described data earlier in this work as distinct pieces of information formatted and stored in a way that serves a purpose.

With this description, data could be as diverse as wanted, so we need to get a bit more specific. In this work, what we are interested about is visualizing datasets. A dataset is just a collection (a set) of data. When we talk about datasets we usually refer to a set of collected samples, organized in a table where each row is a sample and each column represents a particular variable of a sample. We list some of the properties that may make the data complex:

- **High dimensionality**
  This refers to datasets with a lot of variables, which makes it very hard to visualize because we can only properly visualize data in 2 or 3 dimensions.

- **Missing Values**
  Datasets with variables of some rows without values. We explain better why is this a concern in subsection 4.1

- **Data Types**
  Data can come in very different shapes and treating each of them is quite a task. In subsection 4.2 we focus on talking about categorical and numerical variables and how we treat them. Other data types would be dates, which are hard to deal with because they are often circular and may come in different formats.

- **Text data**
  We can find datasets that contain documents, and we cannot process documents in the same way we process numerical and categorical variables and certainly we can not plot the data using the same methods. Documents need to be treated differently, and that's beyond the scope of this work.

- **Time Series**
  Time series is a sequence of variables collected at a regular interval during a certain period of time. This dataset type comes with at least

1 variable as a date, which is part of what makes it difficult to work with. Visualizing this kind of dataset is not hard, but 1 axis will always have to be the date for it to make sense.

Data complexity is a bit of an ambiguous term, it can refer to a lot of things. In this work we narrowed our scope to care mostly about high-dimensional datasets. This means datasets with more than a trivial number of columns. We also tackle missing values and mixed data types (datasets containing numerical and categorical variables) in order to make the work usable, because almost any real-world dataset is going to contain missing values and it's variables may not all be the same type. Of course we can talk about other properties of the data that make them complex, such as the sample size of a dataset, or datasets with types of data besides numerical and categorical, but these are not within the scope of this project and tackling them would require time and effort. This work then, can be considered a step towards a tool for visualizing arbitrarily complex data, where some of the issues commented above are tackled.

4.1 Missing data

In real wold data we cannot always assume to have every value set. When we collect data from the world, we usually have *missing values*. Missing values is what we call when we have no value for a variable of a given sample. Missing data can have a significant impact on the conclusions that we draw from the data and they need to be treated carefully when working with datasets with a high rate of missing values.

Missing values can also be used to draw conclusions, for example, if we are analysing the data of a survey, we might use the missing values to conclude that when a certain variable lays around certain value, the individuals are less likely to respond.

We cannot always use them, and then we need to handle the missing values. In our web application we give the user two options for treating the missing data. The rows containing missing data can either be deleted or they can be imputed. If there are just a few missing values, deleting these samples might be the right choice, as not much information is lost. However, we can also impute them. Imputing means trying to *guess* which value corresponds to that sample, usually this is done by looking at the other samples. A very simple imputation might be setting the missing value to the value of the median of that variable. In our web application, if the user uploads a dataset with missing value is given the option to impute them by using K-Nearest Neighbor Imputer (KNN Imputer)[3]. With KNN imputer each sample's missing values are imputed using the mean value from the nearest neighbors found in the training set. Two samples are considered close if the variables that neither of them is missing are close. For the case of imputing categorical variables we use a simple imputer [4] which resolves the value for a category using the most frequent category for that variable taking into account the other variables as well.

4.2 Numerical and Categorical data

Although there are more types of data than these two, in this work we only focus on numerical and categorical data because they are the most used. We also focused on them because it is easy to convert one type to the other.

In our web application the user is given the option to convert the categorical variables to numerical and vice versa. We recommend, however, that the user converts the data to numerical given the nature of the dimensionality reduction methods used, which most of them accept only numerical data. Also, less information is lost when converting from categorical to numerical than it is the other way around.

To convert the numerical data to categorical data we need to discretize it, create intervals for the numerical data to fit in, and convert a continuous set of data to a discrete one. In our visualizer, the user can choose to convert the numerical variables to categorical ones with the K-Bins discretizer[5], which handles this process by grouping the numerical values into bins, like in a histogram, creating this way the categories. For example: If we have a numerical variable like the score of a test that we want to discretize with value 8.5, we can create 5 categories (5 intervals of length 2). Our score of 8.5 will end up in the category [8-10].

To convert the categorical variables into numerical ones we need to encode the categories. For this we can use One Hot Encoding[6]. One Hot Encoding is a method that converts each category in the categorical variable into a new column, so each original categorical variable spans a set of new variables. Each of this sets acts as a binary array where on a given sample or row the new variables are all 0 except the one that represents the category on which the sample belongs. For example: If we have a categorical variable with two possible values: red and blue. When we apply One Hot Encoding in them, we create a binary array of size two, the first value for the array will be 1 and the second 0 if that row was red and 0 first and 1 second if it was blue.

We leave the high-dimensionality part for the next section, where we hope to make understand why we focused most of our work in it.

5 Dimensionality Reduction

Dimensionality reduction techniques are important in many applications related to machine learning, data mining, bioinformatics, biometric and information retrieval. As we commented earlier in the work, dimensionality reduction is a tool that allows us to map the points of a higher dimensional data to a lower dimension while keeping relevant information within the data, allowing us to visualize it. [23]

5.1 Curse of Dimensionality

To understand why we need dimensionality reduction techniques to visualize complex data we need to understand the curse of dimensionality.

The essence of it it's that a small increase in the dimensionality requires a large increase in the volume of the data to maintain performance in tasks such as clustering or regression. As the dimensionality increases, the probability of more sparse data increases due to the input space growing exponentially with respect to the dimensionality. This causes the degradation of the performance in problems depending on said input space such as classification as we can see in Figure 2. [43]

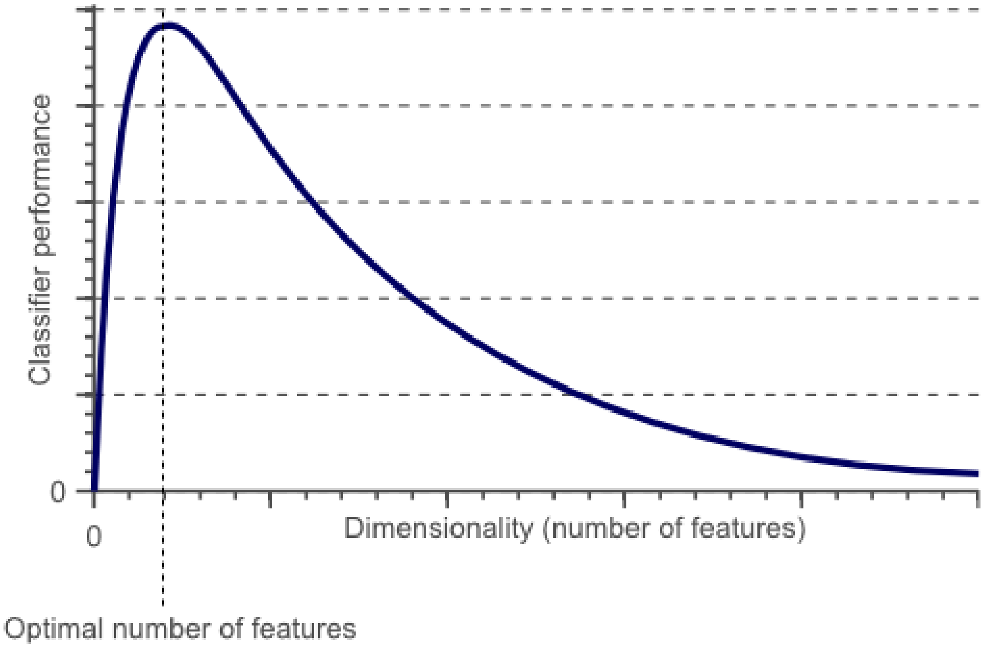


Figure 2: Classifier performance as a function of the number of dimensions

This problem starts at the data gathering, where we obtain noisy or redundant dimensions that barely bring any information, thus resulting in an increase in the dimensionality without any significant benefit.

This work will explain some of the methods used to deal with this *curse*.

Dimensionality reduction techniques are divided into two main categories depending on the function to optimize. Convex techniques optimize an objective function that does not contain any local optima, whereas non-convex techniques optimize objective functions that do contain local optima. In this work we did not focus on these two categories and we divided the methods used into linear and non-linear methods, this way we do not focus on the function to optimize and we can focus on the transformation of the data, which is a more practical approach.

5.2 Linear Dimensionality Reduction

Linear dimensionality reduction methods have been developed throughout statistics, machine learning, and applied fields for over a century, and these methods have become indispensable tools for analyzing high dimensional, noisy data. These methods produce a low-dimensional linear mapping of the original data while preserving most of the information. As such, it can be used for visualizing or exploring structure in data, *denoising* or compressing data.

Cunningham and Ghahramani [30] provided a very general definition of the linear dimensionality reduction as: Given a data matrix $X \in R^{d \times n}$ and a choice of dimensionality $r < d$, optimize an objective function $f_X(\cdot)$ to produce a linear transformation $P \in R^{r \times d}$, and call $Y = PX \in R^{r \times n}$ the low-dimensional transformed data.

To simplify *a lot* without doing a massive deep dive into the mathematics behind linear algebra, to reduce the dimensionality of a matrix A we need to multiply it by a vector/matrix X such that the product is B. So $AX = B$ will have a solution if and only if B is a linear combination of A.

5.3 Non-linear Dimensionality Reduction

The non-linear dimensionality reduction methods can be explained similarly as the linear methods. The difference, as the name implies, is that the transformation applied on the data is not a linear one. So we are not looking for a linear function to transform our data, and that makes it a bit more complicated. The end goal is still the same, we need to transform a given dataset $\mathbf{X}$ with dimensionality D into a new dataset $\mathbf{Y}$ with dimensionality d (with $d < D$, and most times $d \ll D$) while retaining the geometry of the original data as much as possible. [35]

There are a lot of different kinds of dimensionality reduction techniques and we will be working with a few of them. In the next chapter we will be seeing some methods used for linear dimensionality reduction and how we used them for high-dimensional data visualization.

6 Data Visualization Methods

For this thesis we used six different methods that will allow us to visualize high-dimensional data by reducing the dimensionality. We will explain briefly how the methods work.

6.1 Principal Component Analysis

Principal Component Analysis (PCA) is a useful statistical unsupervised technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension. [40] As we can imagine, PCA is used for dimensionality reduction. It is a technique for reducing the dimensionality of datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance, preserving as much variability as possible. This means that PCA finds new variables that are linear functions of those in the original data that are uncorrelated to each other. Finding these new variables (Principal Components) reduces to solving an eigenvalue/eigenvector problem and (more recently) Singular Value Decomposition (SVD) is used as a more general solution. As the Principal Components are dependent on the data and are not predefined and makes no assumptions on the distribution of the data, this method is highly adaptive but at the same time, it makes it very sensitive to the presence of outliers. It also is a *non-parametric* analysis, meaning There are no parameters to tweak and no coefficients to adjust based on user experience the answer is unique and independent of the user. [39]

Although PCA is more than a 100 years old, it resurfaced again due to the available computational power of modern computers, which make it feasible to use on non-trivial datasets. Since the creation, a large number of variants and improvements have been developed in many different disciplines.

PCA can be used with the covariance matrix or the correlation matrix. As it is defined by variance, which depends on units of measurement, a Principal Component based on the covariance matrix will change if we change the units of measurements of one or more variables. As this is not desirable, it is common practice to standardize variables, so the original data matrix is changed to contain standardized values. Since the covariance matrix of a standardized dataset is merely the correlation matrix of the original dataset, a PCA on the standardized data is also known as a correlation matrix PCA. The correlation matrix Principal Components are not the same and are not related to the original covariance matrix PC. The percentage variance accounted for by each PC will differ and frequently more correlation matrix PCs than covariance matrix PCs are needed to account for the same percentage of total variance [31]. Correlation matrix PCs are invariant to linear changes in units of measurement, then they are the appropriate choice for datasets with changes of scale.

6.1.1 Eigenvectors and eigenvalues in PCA

As you know, you can multiply two matrices together, provided they are compatible sizes. Eigenvectors are a special case of this. If you multiply a matrix on the left of a vector, the result is another vector that is transformed from it's original position. All the eigenvectors of a matrix are orthogonal to each other (in a $n \times n$ matrix there are at most n eigenvectors). As the length of a vector does not not determine if it is an eigenvector, so we usually use the length 1 or unit eigenvector. Every eigenvector comes with an eigenvalue associated which tells us the factor by which the eigenvalue is scaled. This process is called eigendecomposition, and we can only perform it in square matrices. The next subsection talks about a generalization of this process.

We can use the eigenvectors of a covariance or correlation matrix to be able to extract useful information that can allow us to characterise the data. And then we can express the data in terms of those eigenvectors by transforming it.

In PCA, once we get the eigenvectors with it's eigenvalues we will choose the eigenvector with the highest eigenvalue, and we will call that the Principal Component 1. And the same will be done with the other components. By doing this we are taking the eigenvalue that explains the data best and this way we will be minimizing the information loss. This can translate as finding a linear basis of reduced dimensionality in which the amount of variance in the data is maximal. We are creating a *new coordinate system* with the principal components, we then translate our data into this *new coordinate system* with a lower dimensionality that preserves most of the information. As we can only perform eigendecomposition with square matrices, we cannot explore most of the existent *new coordinate systems*

6.1.2 Singular Value Decomposition in PCA

We mentioned that to obtain an eigenvector the way we described earlier we need an square matrix in order to extract them. In order to overcome this issue, we use Singular Value Decomposition (SVD). Just like we were decomposing a square matrix to obtain the eigenvalues and eigenvectors with the eigendecomposition, we are now able to obtain *Singular Values*. SVD is a generalization of the case we talked above. The main intuition behind Singular Value Decomposition is, that Matrix A transforms a set of orthogonal vectors v to another set of orthogonal vectors u just like the eigendecomposition did, but this time we do not need the matrix to be square. That means we can explore more of the possible spaces where we project our data, usually achieving better results. The way SVD achieves this is by applying eigendecomposition differently. It is based on the theorem that a rectangular matrix A can be expressed as the product of 3 other matrices. So $A_{mn} = U_{mm} S_{mn} V_{nn}^T$, where U and V are orthogonal matrices and where the columns of U are orthonormal eigenvectors of AA^T and the columns of V are the orthonormal eigenvectors of $A^T A$ and S is the diagonal matrix containing the square roots of eigenvalues from V or U in descending order. We are able to find the three matrices V, U and S applying the eigendecomposition two different times for U and V from the original matrix A. [14] The Python implementation that we used for the program uses SVD.

6.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a very common supervised technique for dimensionality reduction problems as a preprocessing step for machine learning and pattern classification applications. The LDA is developed to transform the features into a lower dimensional space, which maximizes the ratio of the between-class variance to the within-class variance, thereby guaranteeing maximum class separability. As with the PCA, the goal is to project the original data onto a lower dimensional space. It does so by first computing the separability between the different classes (it computes the distance between the means of the different classes). This is called the *between-class variance* S_{B_i}. Then it computes the distance between the mean and the samples of each class, which is called the *within-class variance* S_{W_i}. The third step is to construct the lower dimensional space. Figure 3 shows a simple visual representation of LDA does. There are two different methods to find the LDA lower dimensional space, *class-dependent* and *class-independent*, that we will talk about in a bit.

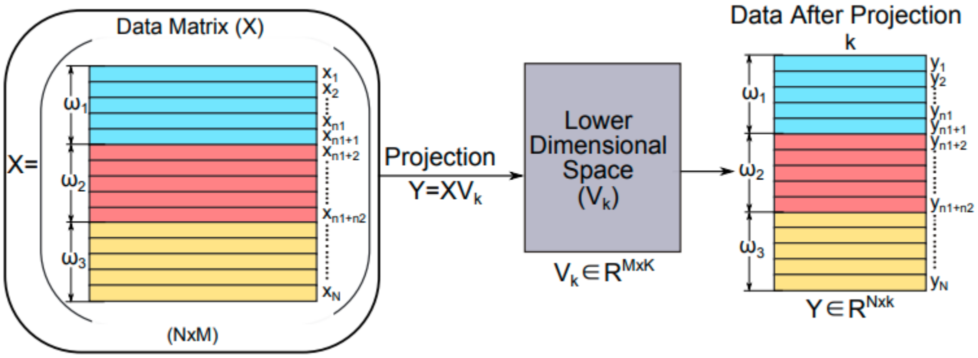


Figure 3: Visual outline of the projection of the original samples on the lower dimensional space found by LDA. X is the original data. V_k is the lower dimensional space. ω_i is class i

The between-class variance of the i^{th} class (S_{B_i}) represents the distance between the mean of the i^{th} class (μ_i) and the total mean (μ). LDA searches for a lower-dimensional space, which is used to maximize the between-class variance, or simply maximize the separation distance between classes. The within-class variance of the i^{th} class (S_{W_i}) represents the difference between the mean and the samples of that class. [23]

In the class-independent method, after computing the between-class variance (S_{B_i}) and within-class variance (S_{W_i}), we can compute the transformation matrix (W) of the LDA as:
$S_{B_i}W = \lambda S_{B_i}W$, where λ represents the eigenvalues of W. As we said in the previous chapter, we are able to compute the eigenvalues and eigenvectors of a matrix with eigendecomposition. The eigenvectors represent the directions of the new space, and the corresponding eigenvalues represent the scaling factor or length of the eigenvectors. Each eigenvector represents one axis of the LDA space, and the eigenvalue represents the robustness of this eigenvector. The robustness of the eigenvector is directly related to its ability to discriminate between different classes, increasing the between-class variance, and decreasing the within-class variance of each class. In this method, all classes are projected onto the same lower dimensional space.

In the case of class-dependant method, we compute a different lower dimensional space for each class:
$W_i = S_{W_i}^{-1}S_B$, where W_i is the transformation matrix for class i. This means that we get eigenvalues and eigenvectors for each transformation matrix, projecting the elements of its class on its own lower dimensional space. This comes at a price, it consumes more resources and it may not get good results if the sample size is small, because the number of elements in each class affects W_i.[23]

Now we are going to discuss the two main problems of LDA and present some solutions. As LDA is used to find the lower dimensional space using a linear transformation that discriminates between classes, we run into problems when the classes are non-linearly separable because we are not able to find a linear transformation to project our data onto the lower dimensional space. We can say that LDA struggles to find the lower dimensional space when the discriminatory information used to separate classes is not on the means. We can find problems which LDA cannot solve because the information to discriminate is in the variance instead of the mean. This can happen when the classes of a problem have a very similar mean, then we have that S_B and W are zero ans the lower dimensional space cannot be computed.

Because the problem is the linearity in our data we can imagine a non-linear transformation used. We will talk about non-linear transformation when we're talking about the non-linear dimensionality reduction methods. For now we can understand it as mapping the original data into a higher dimensional space where we can find a linear transformation for LDA to project the data onto a dimensional space lower that the original.

Another problem LDA can run into is the *Small Sample Size problem.* This problem results from high-dimensional pattern classification tasks or a low number of training samples available for each class compared with the dimensionality of the sample space. This happens when the S_W matrix is singular (A matrix is singular if it is square, does not have a matrix inverse, the determinant is zeros; thus, not all columns and rows are independent).

There are some solutions to the Small Sample Size problem that LDA encounters. We can use Regularization LDA, which multiplies the identity matrix by η and then it's added to the S_W matrix to make it non-singular, making it possible to compute it's inverse. Then, $S_W = S_W + \eta I$. This, however, amplifies the parameter tuning and can get pretty bad results if not chosen carefully. We can also use a sub space to transform the original data into a lower dimensional space equal to the rank of S_W so we can invert it. We can use PCA to achieve this lower dimensional sub-space and then applying LDA method. The use of this method is, however, not fully recommended as we often lose information about how to discriminate between classes.[28]

6.3 Multiple Correspondence Analysis

Multiple Correspondence Analysis (MCA), like PCA, is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. We, however, assumed the variables to be numerical in the case of PCA. MCA solves this problem, as it is considered a PCA-like method for categorical variables.

6.3.1 Correspondence Analysis

Before we dive deep into MCA, we need first to understand its parent method: Correspondence Analysis (CA). Correspondence analysis (CA) is a generalized principal component analysis made for the analysis of qualitative data. CA was created to analyze contingency tables, but, CA is so versatile that it is used with a lot of other data table types. It can only be used on datasets with two discrete or categorical variables.

The goal of correspondence analysis is to transform a data table into two sets of factor scores: One for the rows and one for the columns. The factor scores give the best representation of the similarity structure of the rows and the columns of the table. Just as principal components account for maximum variance of quantitative variables in PCA, CA finds scores for the row and column categories on a small number of dimensions which account for the greatest proportion of the χ^2 for association between the row and column categories. The scores provide a quantification of the categories, and they maximize the correlation between the row and column variables.

One simple development of CA would be to find the scores of the row categories X and the column categories Y of a given matrix data with the help of Singular Value Decomposition. Applying SVD to the residuals from independence, trying to account for the largest proportion of χ^2 in a smaller number of dimensions. Each row point is the weighted average of the scores for the column categories, and each column point is the weighted average of the scores for the row observations. [22]

This way, CA is designed to show the deviation of the data from the expectation when the two variables are independent. Correspondence Analysis shows only row and column categories as points in the two or three dimensions which account for the greatest proportion of deviation from independence. The pattern of the associations can then be inferred from the positions of the row and column points.

As we said, CA can only be used with datasets with at most two categorical variables. As this seems to restrict a lot our options, given that most datasets worth looking at have more that two columns. To solve this problem we get back to our original method and the extension of CA, Multiple Correspondence Analysis.

Multiple Correspondence Analysis allows us to display the relationships between categorical variables. Provides an optimal scaling of the variables, giving a score to the categorical variables which can then be plotted to visualize the relationships between the categorical variables. Unfortunately, the generalization of the CA we've seen to use it in more than two variables follows a different path, so CA is not really a special case of MCA where the data only have two variables. The intuition behind it is somewhat similar, so having knowledge of correspondence analysis may help us understand how the MCA works.

The typical development of MCA starts by defining indicator variables for each category and express the contingency table in the form of a cases by variables indicator matrix Z. Then, MCA can be described as the application of the simple correspondence analysis algorithm to the indicator matrix Z. By doing this we would get scores for the rows (samples) of Z, but not the columns (categories).

This way we obtain a point for each category, and that point is the centroid of of all the samples pertaining in that category for a given categorical variable, and the origin represents the weighted average of the categories for each variable. Thus, categories with low frequencies will be located further from the origin in the new lower dimensional space, and categories with high frequencies will be located closer to the origin.

To extend the use of CA for datasets with more than two categorical variables we can use what is called a *Burt Matrix* B. B can be computed as: $B = Z^T Z$, where Z is the indicator matrix we have. In B, the diagonal blocks contain the one-way marginal frequencies. The off-diagonal blocks contain the bivariate marginal contingency tables for each pair of variables.[11]

We can define MCA now as the singular value decomposition of matrix B, from which we get the scores for the categories of all variables so that the greatest proportion of bivariate associations is accounted for in a lower dimensional space.

6.4 Kernel PCA

We now start with the non-linear dimensionality reduction methods. We begin with Kernel PCA, an extension of the method already explained in Section 6.1. With PCA we could find a lower dimensional space which was a linear combination of the previous space. However, if the data does not lay in a linear space PCA cannot really find the relations between the data. This is where Kernel PCA is useful.

The standard steps for Kernel PCA follow as: First, we assume a non-linear transformation $\phi(x)$ that transforms from our original D-dimensional space to an M-dimensional space where $D \ll M$. Then, the naive approach would be to apply PCA to this new dimensional space and find linear combinations there. This can be costly and inefficient. We can use kernel methods to simplify the computations needed. We first assume that the projected new features have zero mean. This means:

$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i) = 0$$

Then we compute the covariance matrix of the projected features C as:

$$C = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\phi(x_i)^T$$

And as we know, we can compute the eigenvalues and eigenvectors as:

$$Cv^k = \lambda_k v_k$$

Now, if we define a kernel function $\kappa(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ and multiply by $\phi(x_l)^T$ both sides of the previous equation we can use matrix notation to get:

$$K^2 a_k = \lambda_k N K a_k$$

where K_{ij} is the kernel function defined earlier for i and j, and a_k is the N-dimensional column vector.

We can now compute the kernel principal components as:

$$y_k(x) = \phi(x)^T v_k = \sum_{j=1}^{N} a_{ki}\kappa(x, x_i)$$

We can see that this is *a bit* more complicated than the linear version, but it is powerful, and it can be understood better with a correct understanding of what kernel functions are. We can see that the steps on the this section are not trivial, and it may seem that we are not getting where we want, but we are basically transforming the data to a higher dimensional space where we can find linear separations for our data.

If we were to apply PCA to a $N \times N$ dot product matrix we would still need to evaluate the dot products in a very high dimensional space to compute the entries of the matrix even when a diagonalization may be tractable, it would not be enough. Kernel PCA does not compute all dimensions in the feature space so we can avoid this problem. It chooses an adequate subspace inside the feature space, taking advantage of this lower dimensionality by working only on a relevant subset of the feature space (we could say that it only works with the features that it deems most important.

Kernel PCA has the property of unitary invariance, due to the fact that both the eigenvalue problem and the feature extraction depend on only kernel values. This ensures that the features extracted do not depend on which orthonormal coordinate system we use for representing our input data.

6.4.1 Kernel Functions

So, we need to transform our data to a higher dimensional space where the data can be linearly separated. This seems like a very complex operation, given that we are converting points in the input space of the data to a much higher and much complex dimensional space. Kernel functions allow us to make this mapping of the input to a higher dimensional space with very reduced computations by only computing the dot product between the mapped patterns, and not the patterns itself. We represent the computation of the dot product as a kernel function like: $\kappa(x, y) = \phi(x) \cdot \phi(y)$, which allows us to compute the dot product in the feature space without explicitly having to use ϕ. Through a kernel function we can represent the dot product between elements in the feature space in terms of elements in the input space. Often, the feature space has a very high dimension, so we would like to find an expression for κ that can be efficiently computed because all the computations needed are done in input space instead of the much higher dimensional feature space.

Choosing a κ without being concerned with the mapping of ϕ into the feature space might correspond to a dot product between patterns mapped with a suitable ϕ. This means we need to work solely in terms of the dot products without needing ϕ. This is useful because ϕ is probably very high dimensional, and so, computations are costly. We can see that this has some restrictions, as we are hoping to find a kernel function that allows us to skip the computations to do the mapping ϕ and we can assume that most kernel functions will not have this property. However, *Mercer's Theorem* [38] implies that if k is a continuous kernel of a positive integral operator, there exists a mapping into a space where κ acts as a dot product. So we need to be clever about choosing a kernel function for our data.

6.5 t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) is popular nonlinear dimensionality reduction and data visualization method. It was presented very recently (2008[36]) and it has become a very popular procedure and a state-of-the-art technique in a lot of applications. t-SNE is a iterative algorithm that allows us to visualize high dimensional data by applying dimensionality reduction, just like the methods we've seen until now. t-SNE is able of capturing much of the local structure of the high-dimensional data very well, while also revealing global structure such as the presence of clusters at several scales. This is something that most non-linear dimensionality reduction methods are not able to achieve in real world datasets.

t-SNE starts by computing a joint probability distribution for all pairs of points in the data, creating a symmetric $n \times n$ matrix P. In a similar fashion, it creates a two-dimensional symmetric matrix Q with the joint probability distribution over all pairs $\{(y_i, y_j)_{1 \leq i \neq j \leq n}\}$.

P and Q are similarity matrices that contain information about the distances between pairs of points in the high dimensional space (P) and the two-dimensional space (Q). Each element in P is computed as:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

where $p_{j|i}$ is:

$$p_{j|i} = \frac{exp(-||X_i - X_j||_2^2/2\tau_i^2)}{\sum_k exp(-||X_i - X_k||_2^2/2\tau_i^2))}$$

And we define the elements of Q as:

$$q_{j|i} = \frac{(1 - ||y_i - y_j||_2^2)^{-1}}{\sum_{k,t,k \neq t}(1 + ||y_k - y_t||_2^2)^{-1}}$$

The goal of t-SNE is to find n points in the lower-dimensional space (y_i) by solving an optimization problem where we try to minimize whats called the Kullback–Leibler divergence(KL-divergence). KL-divergence is a measure of how a probability distribution is different from a second. We can solve the optimization problem through many different methods, the most widely used is a variation of the gradient descent algorithm that includes a *momentum term*. This term is used to speed up the convergence and reducing the risk of getting stuck in a local minimum[17]. Even with the momentum parameter we can be stuck in a position where we are very slow to find the convergence or even we are not able to achieve it in some cases. As a solution, the standard practice for the use of this method is to use a *early exaggeration* technique that's applied to the early stages of the optimization which speeds up the convergence.

Most of the implementations are based on an early exaggeration phase followed by the embedding stage that iterates through the gradient descent algorithm.

t-SNE is made with SNE as a basis. SNE is the previous method that aimed for the same goals, but had a slightly different, less efficient approach. Specially the cost function. In t-SNE, we use a symmetric version of the cost function that SNE uses, with simpler gradients, allowing for easier, faster computations and overall optimization. It also uses t-Student distribution instead of a Gaussian distribution to compute the similarity between the points in the lower dimensional space.

There's another problem that SNE has and t-SNE solves, it is known as the crowding problem. The crowding problem occurs when we map high dimensional data into a two or three dimensional space. As t-SNE works with the distance between pairs of points, it cannot faithfully represent the high dimensional data onto a lower dimensional space. For example, if we had 10 dimensions and we were trying to map to a two dimensional space, we would run into the problem that it is possible to have 11 points that are mutually equidistant and we have no way of representing them in two dimensions. This happens because the area of the two-dimensional map that is available to accommodate moderately distant points will not be nearly large enough compared with the area available to accommodate nearby points[36]. This produces unwanted results in SNE: this problem makes it so that a lot of points will be close to each other (crowded), which prevents gaps forming between natural clusters of the data. The solution proposed is to inject a slight *repulsion* between points so that any distance between a pair of points cannot be below a certain threshold.

Although t-SNE solves problems from SNE, it is not a perfect method and can suffer some drawbacks. The first one is how it works for dimensionality reduction not done for data visualization (i.e. dimensions > 3). That is because of the heavy tails of the t-Student distribution. In high-dimensional spaces, the heavy tails comprise a relatively large portion of the probability mass under the Student-t distribution, which might lead to d-dimensional data representations that do not preserve the local structure of the data. This does not bother us much because in our web application we will we only using t-SNE for data visualization.

Another problem that t-SNE can encounter is when we use it with data with a high intrinsic dimensionality. This means that if we use a dataset that needs at least 4 dimensions to be properly represented, t-SNE will not bear results as good. This is caused because of the assumption that t-SNE makes that the data will be locally linear. By definition this is a problem that we will not be able to avoid fully, because it is impossible to represent the structure of the data with less dimensions than the intrinsic dimensionality of the data.

The major weakness of t-SNE can be found in it's cost function not being convex. This can mean that the method does not find a global minimum, and so, we can not get the best representation of the data in a lower dimensional space. This draws us to choose several optimization parameters and does not guarantee that every execution is the same, because we start with a random configuration of map points.

6.6 Multi-Dimensional Scaling

Multidimensional scaling (MDS) is a classical method for generating meaningful non-linear low dimensional embeddings of high dimensional data. MDS has a long history in the statistics, machine learning, and graph drawing communities. Similar to t-SNE, MDS tries to provide a lower dimensional embedding of high dimensional data by trying to represent properly the distance between points. It's used for the analysis of similarity of data.

MDS tries to represent the *proximities* by distance among the points of an m-dimensional space. The most frequently used and the most natural distance function is the Euclidean distance. It corresponds to the length of the straight line segment that connects two points. MDS uses an expression as a metric of how well is the data represented in the lower dimensional space. It's called *stress* and it's computed like:

$$stress = \sqrt{\frac{\sum(d_{ij} - \hat{d}_{ij})^2}{\sum d_{ij}^2}}$$

where $\hat{d}$ is the predicted disparities by the MDS, and d_{ij} is the distance in the original high dimensional space. Disparities are a transformation of the proximities that we deem admissible. Depending on how we compute these disparities we will be able to use the metric method or the non-metric method. This is useful because we can not always assume that the geometric distance is the one that is important to us. MDS could be useful for finding genetic distances between populations, and we can compute the dissimilarity of pairs of populations as we deem convenient.

6.6.1 Metric MDS

Metric MDS tries to model the similarity of the data as distances between pairs of points. The distances are computed using the geometric coordinates of the data. When mapping the data from the high dimensional space to a lower dimensional space we try to preserve the distances between points as much as possible.

Constructing the distance matrix of the data allows us to perform eigendecomposition or Singular Value Decomposition to compute the eigenvalues and eigenvectors to construct the lower dimensional space and embed the points in it. As we were trying to minimize the stress, the lower dimensional space found is fit to create a good representation of the data.

6.6.2 Non-Metric MDS

In non-metric MDS, rather than using a distance metric for the distances between points in the embedding space, we use a non-parametric monotonic function. This means that only the order of dissimilarities is important rather than the amount of dissimilarities. This changes the optimization function to use the result of the function as the dissimilarity predicted by the MDS. Non-metric models represent only the ordinal properties of the data

We can see that this procedure is very similar of what we have seen in PCA, and indeed classical MDS using euclidean distances is identical to PCA [24]. The difference is subtle then, it depends on how we choose the distance. We can find the lower dimensional space of non-linearly related data by optimizing the stress formula, which as we said is determined by our way of describing disparities, and so, we can find a way of measuring distances that is not necessarily linear.

Also, PCA assumes linearity in the data. MDS does not need to make any assumption of that kind, and so is very suitable for a wide variety of data. PCA also has the restriction that uses the covariance or correlation as a measure for similarity, while in MDS we are able to choose any measure that suits us.

Although MDS suffers from inconveniences. It is a numerical optimization technique, and as such it can fail to find the best solution possible because it gets stuck in a local minimum. There are some ways to overcome this issue, which involves initializing the method randomly some amount of times in order to prevent it to get stuck in the same place every time while optimizing the function.

7 Visualization

Now that we have seen the methods we can see how are the visualizations in our web application. All the visualizations that we see have some degree of interaction in the web application, hovering over points allow us to see the exact coordinates of said point and we can also see the index of that point. In the case of 3 dimensional plots it is also possible to rotate the plot in order to see from all the angles. This kind of interactivity is removed here so we will try to include graphics which are representative enough.

7.1 MNIST dataset

We have first chosen the MNIST dataset[7], which consists of handwritten digits. The dataset is labeled, and every variable corresponds to a pixel of a 28×28 image, which means we end up with 784 variables for each row. For each row we have the digit labeled. This dataset is very famous and widely used, and that is in part why who chose it, and even knowing that is considered a simple dataset we think it is a good dataset to show dimensionality reduction, because we will be going from 784 dimensions to 2 or 3. We will be using only a part of the dataset, because the original data is over 60.000 rows and the operations become quite time consuming when we are trying to process and render so many points. We are aware that choosing to use only a part of the dataset we probably end up with worse results, but it will give a nice idea of how the methods perform.

We are going to start by plotting the result of the PCA. In Figure 4 we see that each class or digit is painted in a different colour, which allows us to visually see which classes the PCA has been able to separate better. It seems obvious that for most of the classes it did not do a great job, but we can see how it seems that it recognizes the digit *1*. It also seems to group some other classes together, like classes 7 and 9 are in the same region in the plot.

At the axis of the plot we find the explained variance of the Principal Component. In total, a 17.5% of the total variance is explained, which is not much, and that's the reason it does not seem to perform well.

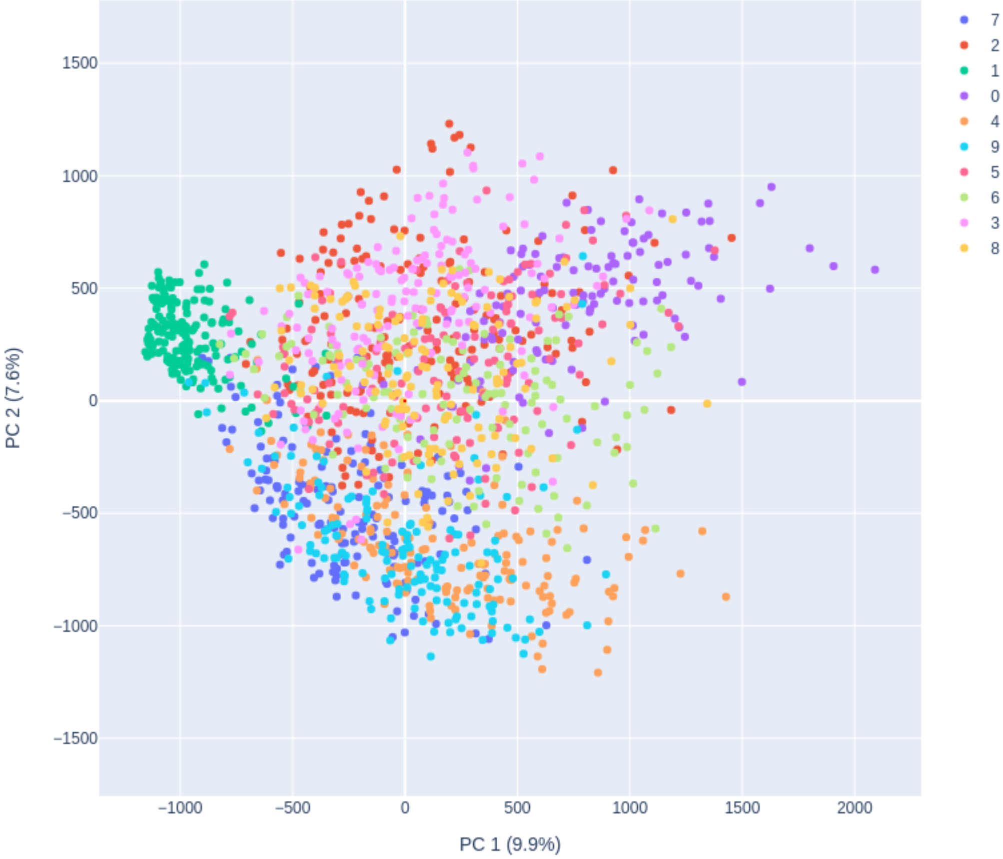


Figure 4: PCA result of the MNIST dataset in 2 dimensions

In Figure 5 we have the plot for the LDA on the MNIST dataset. Now we can clearly see how the classes have been separated better than they did with PCA. The digit 1, which was the only one that really stood out in PCA is much more clearly separated from the rest. We also wee that digits 0 and 2 are quite separated in comparison with PCA, but that's somehow expected given that we are explaining more than 40% of the variance.

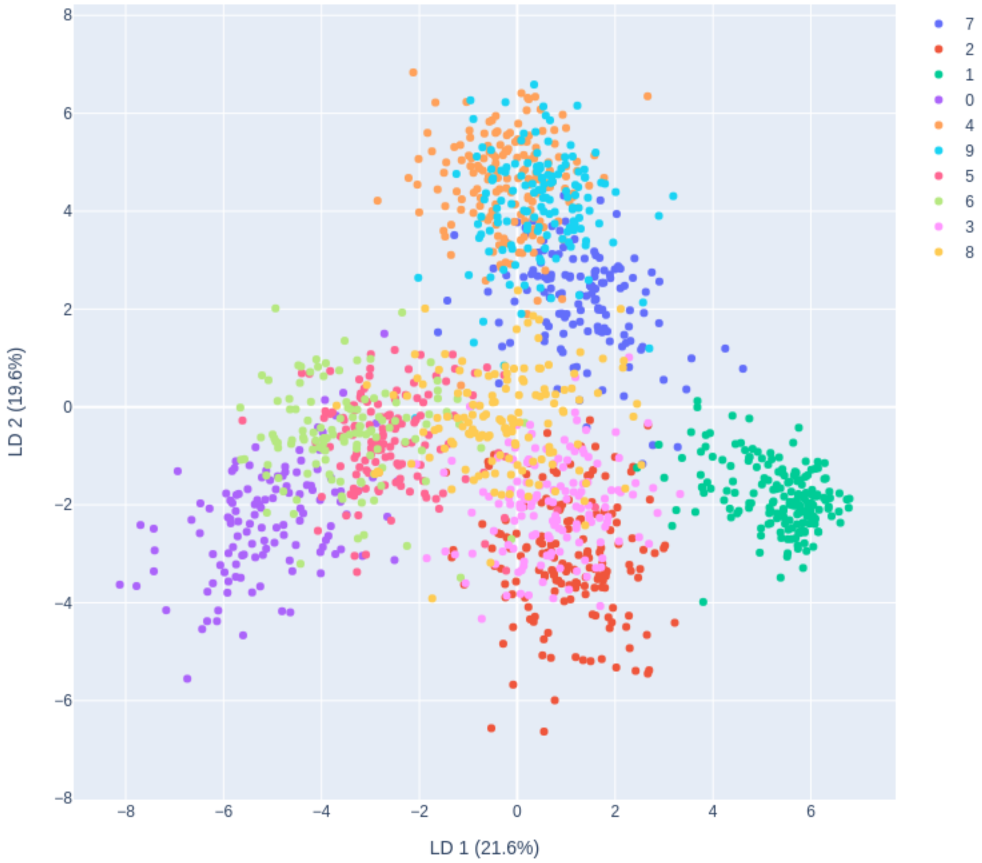


Figure 5: LDA result of the MNIST dataset in 2 dimensions

We will now try to see if we can have a better separation if we plot in 3 dimensions in Figure 6. In this plot we explain 56,9% of the total variance. We are now able to visualize how the classes are separated fairly clearly, and we can see the change in our perception when changing from 2 to 3 dimensions. Even when we only explain a little over 50% of the variance, we could assume, by seeing Figure 6 that we would not need much more to have all the classes separated enough.

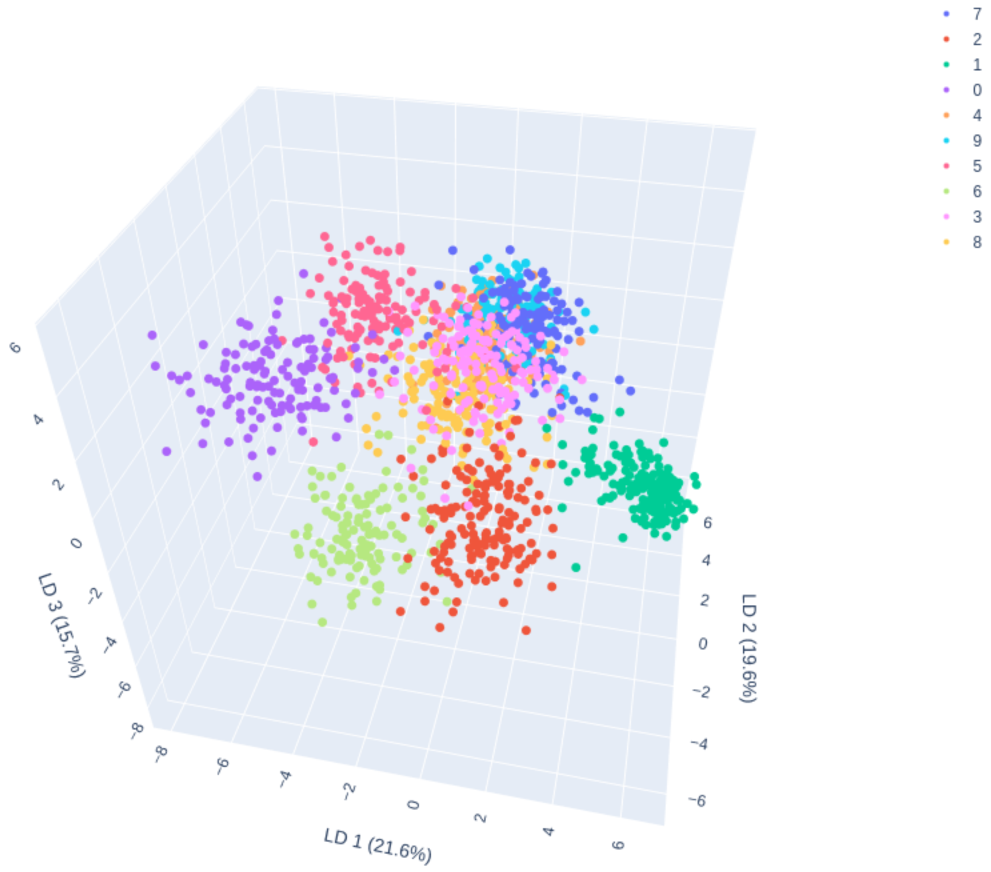


Figure 6: LDA result of the MNIST dataset in 3 dimensions

Now with the non-linear methods in Figure 7 is the plot of the Kernel PCA. We've used the cosine kernel which was the one which has given the best separation between the digit classes. And we can see that it's fairly similar to PCA, it separates the digit class 1 but all of the other classes have a massive overlapping. In the case of non-linear methods we do not have the total explained variance as we had with the linear methods. The axis explain the ratio of the variance explained between the Principal Components, so they will always add up to 100%. That's because when we are transforming the data linearly we are projecting it to some orthogonal basis, and we are able to extract the variance explained from there. But in the case of non-linear projections, like in this case, we cannot decompose the variance in the same way, and thus, we only can try to explain the ratio of the variance explained by each axis.

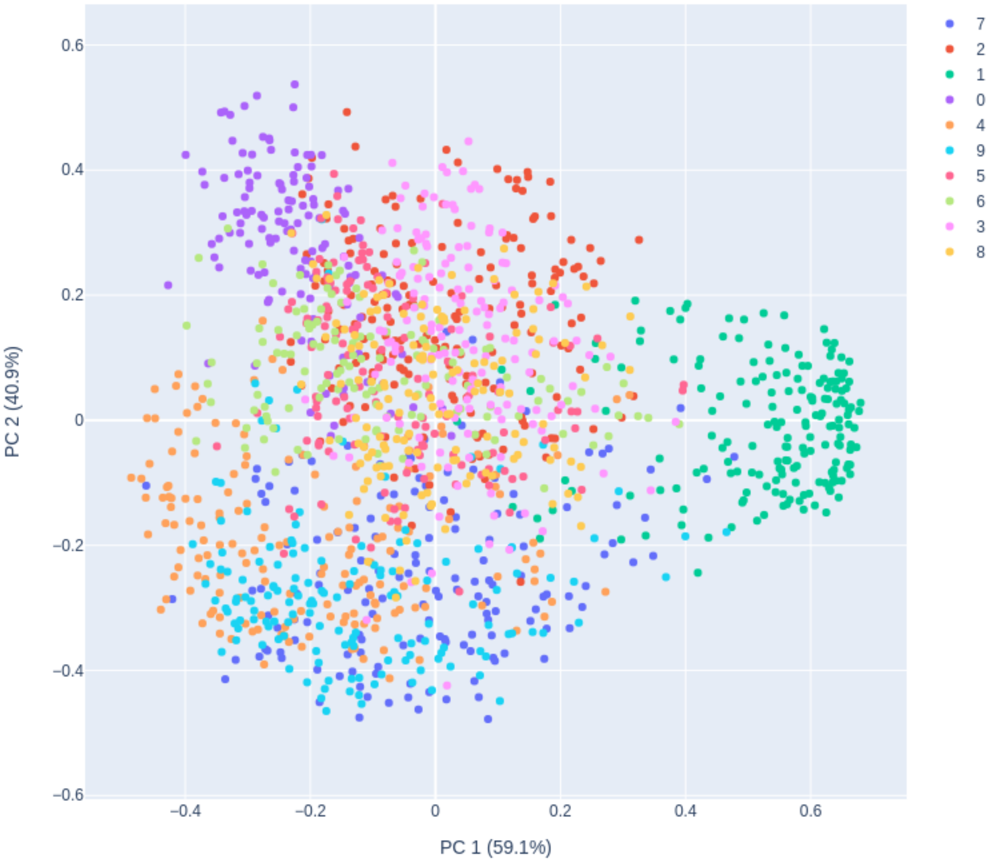


Figure 7: Kernel PCA result of the MNIST dataset in 2 dimensions

In Figure 9a we have the plot made by t-SNE using the parameters in the caption of the figure. Perplexity makes reference to the number of nearest neighbours and the other two parameters are for the gradient descent algorithm. This is clearly the best class separation we've seen, we can clearly see how the method grouped the data in clusters. We can see some of the issues with t-SNE, which is that it can get stuck in local minimum, which prevents the classes to be totally separated. We can conclude that because there are some classes that get some of the points separated by another class. This can also be the product of not using enough samples of the original dataset, but we can clearly see how t-SNE performs with this dataset, allowing us to visualize the topology of the data in 2 dimensions (remember the original 784). As a metric we have the Kullback–Leibler divergence which t-SNE tries to minimize. It is not an extremely high value, so we could say that it is a fairly good representation of the original topology of the data.

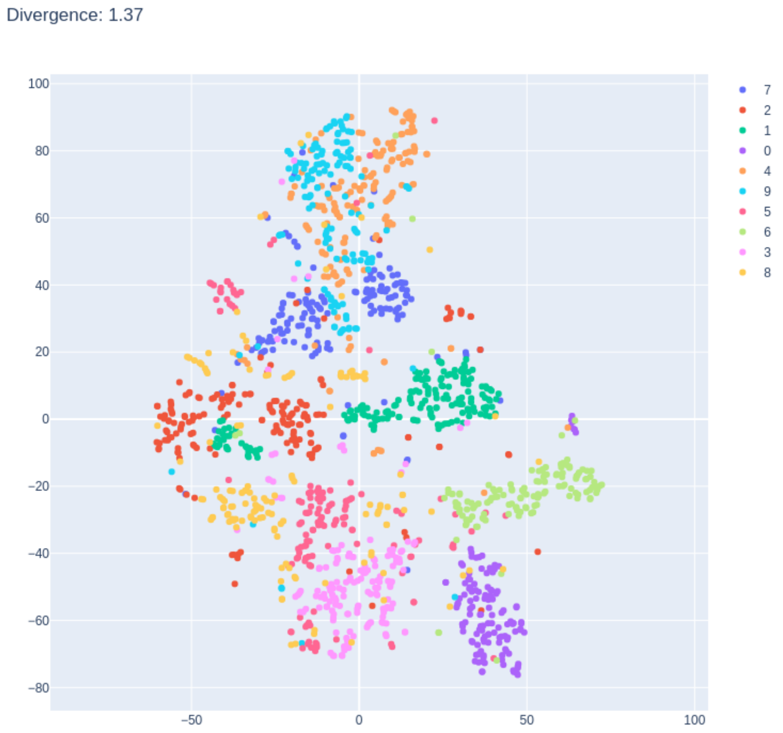


Figure 8: t-SNE result of the MNIST dataset in 2 dimensions

(a) Perplexity: 30, Early exaggeration: 12, Learning rate: 200

Finally we are going to plot the result of MDS in Figure 10. We see that again the digit class 1 is the only one which is clearly separated (or grouped at least). Because of how MDS works it tends to group the points this kind clusters, and we can see some classes (digit 0 for example) that are beginning to separate, but most of the digit classes have a really big overlapping.

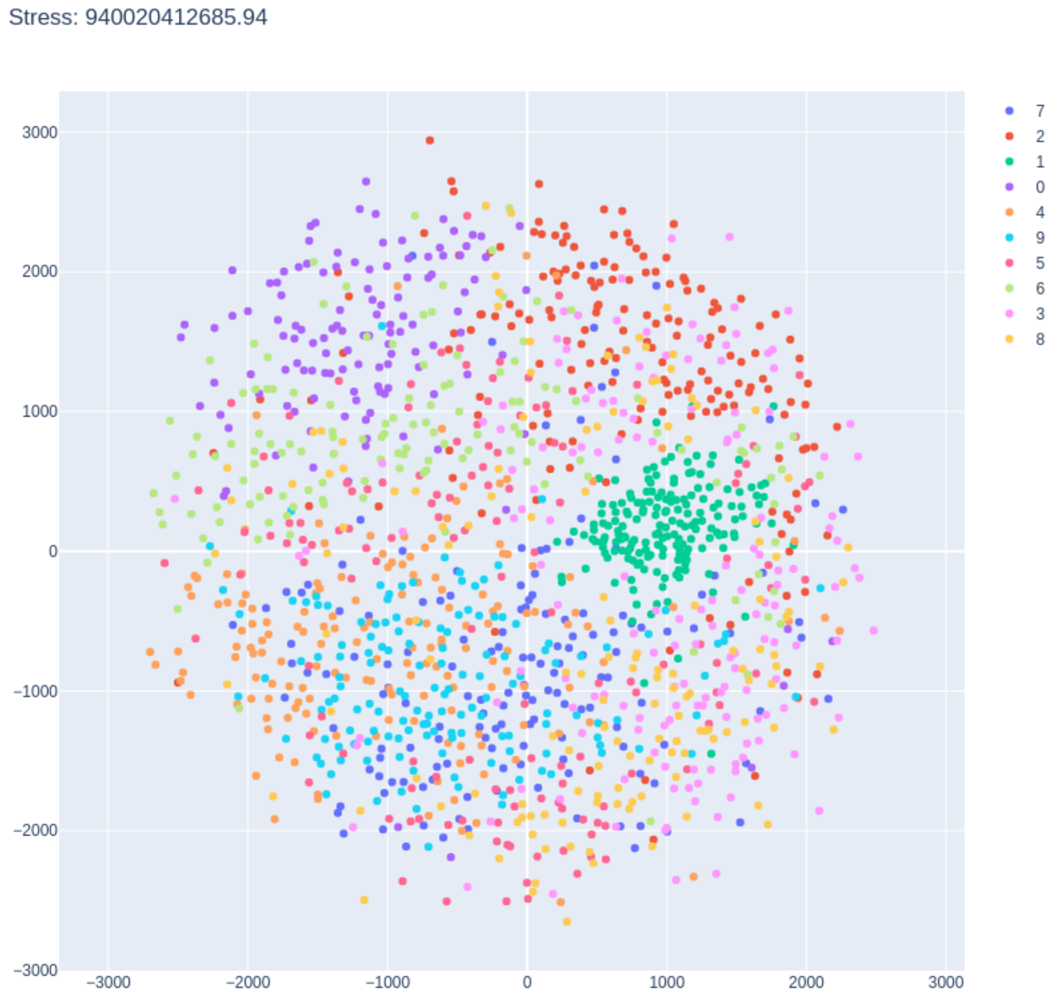


Figure 10: MDS result of the MNIST dataset in 2 dimensions

7.2 Indian Pines dataset

We will now try out the Indian Pines dataset[15]. This time we will try to separate into different types of land-use types with the data from hyperspectral sensor data from the soil from two parts of USA land. It contains 221 variables and we are using only a portion of the dataset again for the same reasons as before.

With PCA in Figure 11 we can see that it does a relatively good job at separating the land-use type classes. The classes in the right side are almost entirely separated from one another, but the classes for corn and alfalfa are overlapping a lot. We could say that PCA performs better in this dataset, this is obvious when looking at the plots, but also, with the Indian Pines dataset the PCA is able to explain more than 80% of the total variance, so we know we have a good representation of the data.

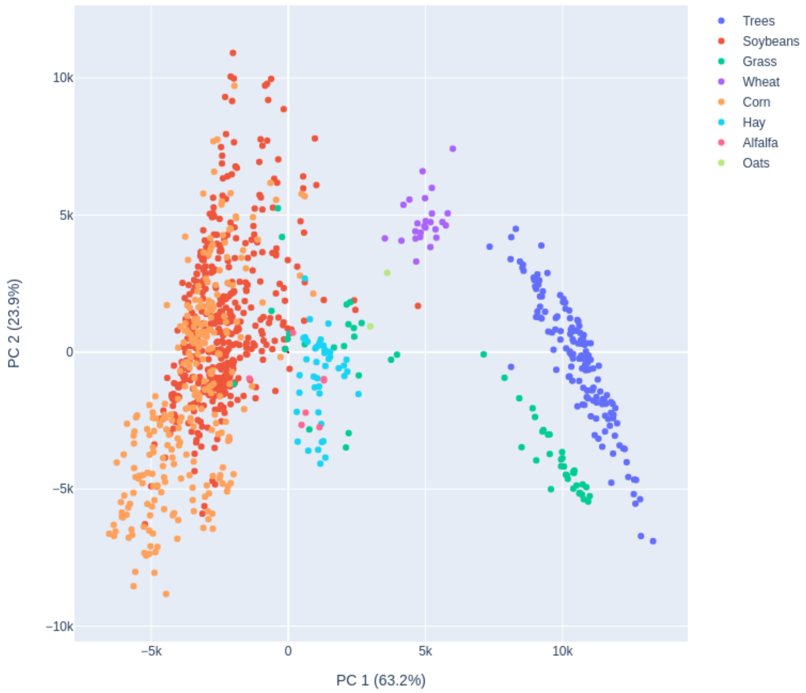


Figure 11: PCA result of the Indian Pine dataset in 2 dimensions

LDA performed very good with the last dataset in Figure 12, and with this one we can see that it has separated all of the classes almost perfectly, with very little overlap with the classes Oats and Hay. LDA is very good at forming clusters because of how it works, pulling elements from the same class together and separating different classes. This time LDA has explained more than 75% of the variance.

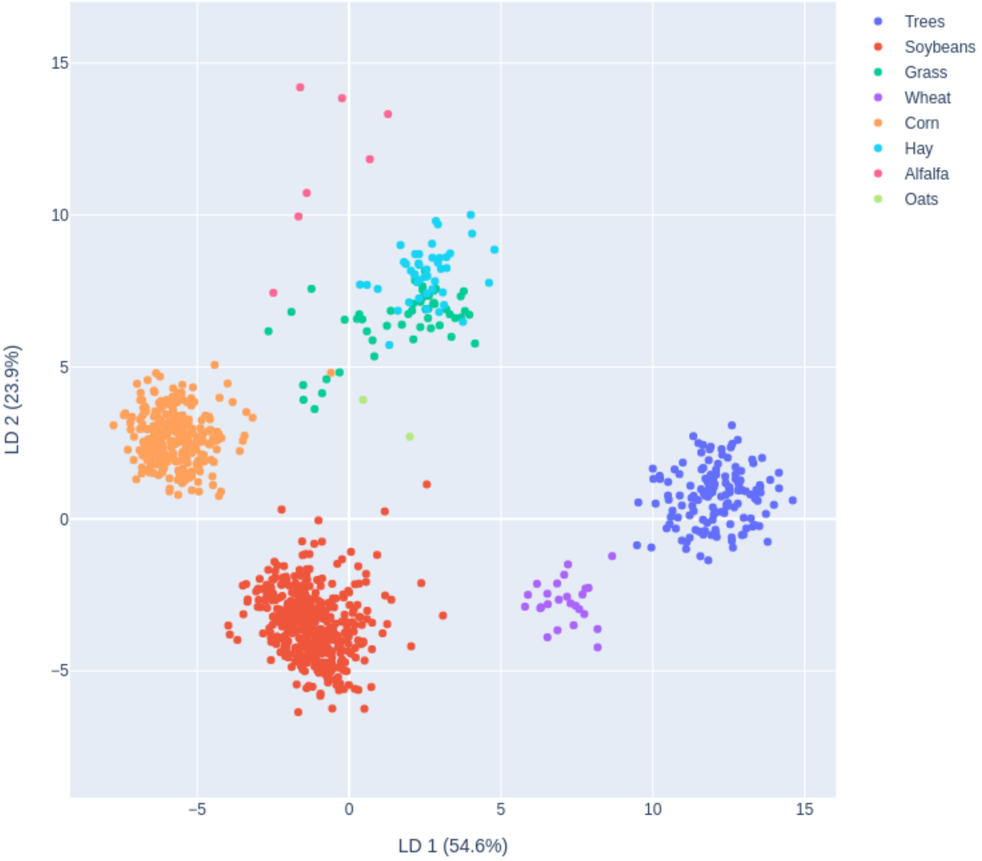


Figure 12: LDA result of the Indian Pine dataset in 2 dimensions

This time with Kernel PCA we tried to use the polynomial kernel with degree 3 to see how it would perform. The result is in Figure 13 Like in the previous dataset, PCA and Kernel PCA separate the classes in a similar manner. We can see that the same classes that PCA was struggling to separate, Kernel PCA does not separate either. Comparing the performance with these two datasets for the methods, PCA and Kernel PCA did not perform as well as some others. This may be due the datasets not being the best for (Kernel)PCA.

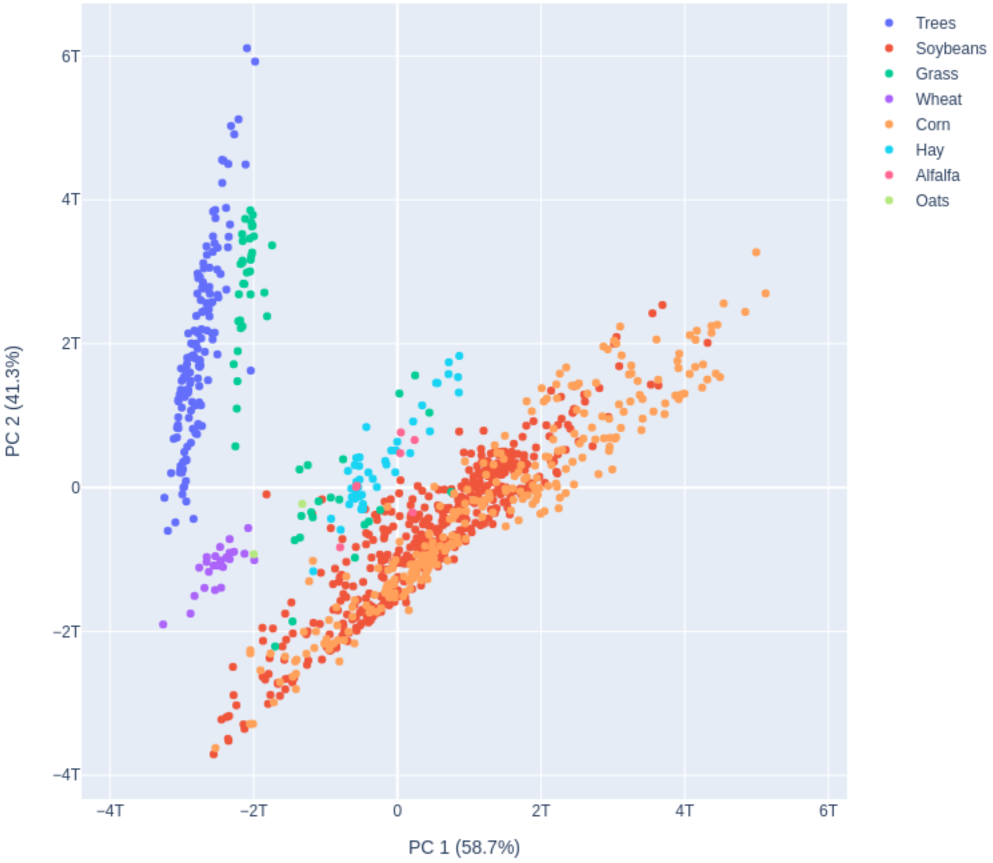


Figure 13: Kernel PCA result of the Indian Pine dataset in 2 dimensions

In Figure 14 we see the result of t-SNE. Again, it has separated the classes almost perfectly, we can see how there is no overlap in most of the clusters and close to none overlap in some of them. We could say this is a good representation of how the data looks like given the low value for the

KL-divergence which t-SNE tries to minimize.

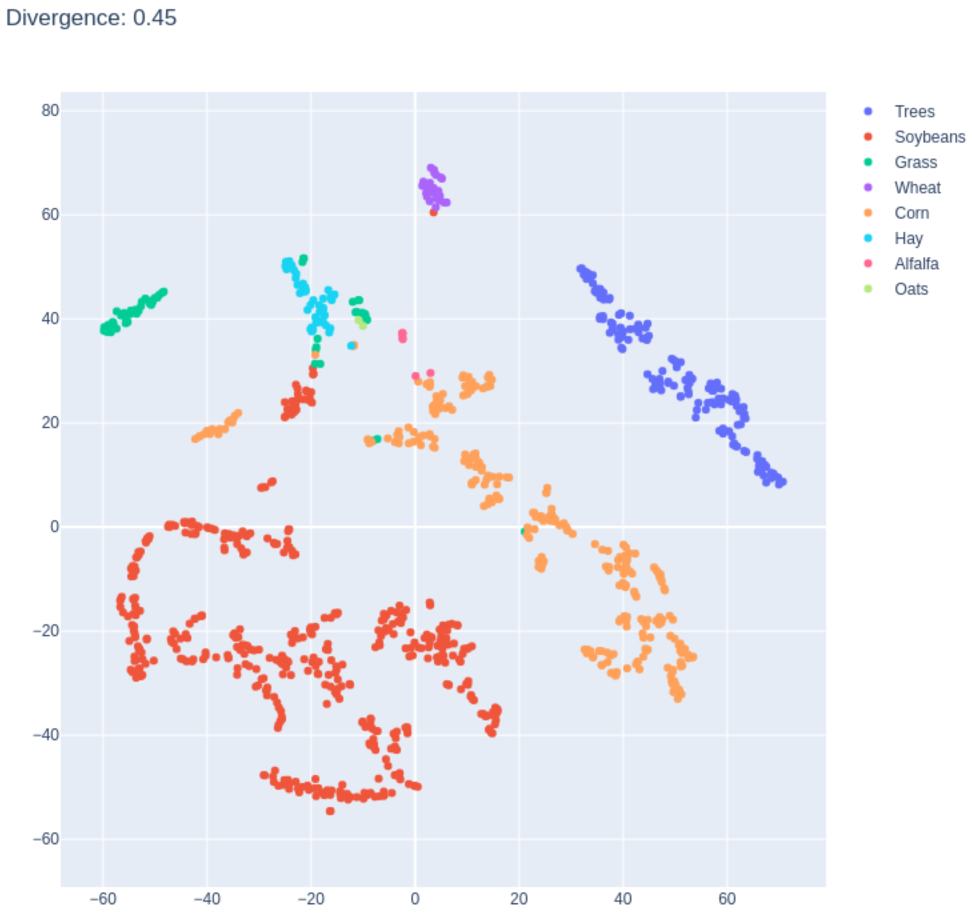


Figure 14: t-SNE result of the Indian Pine dataset in 2 dimensions

In Figure 15 we can see how MDS has improved the separation of the classes significantly with this dataset. It separates most of the classes like PCA or Kernel PCA does, but the ones they were struggling to separate, MDS achieved quite better. While there is still a bit overlapping between the classes, we can easily see the separation between the classes.

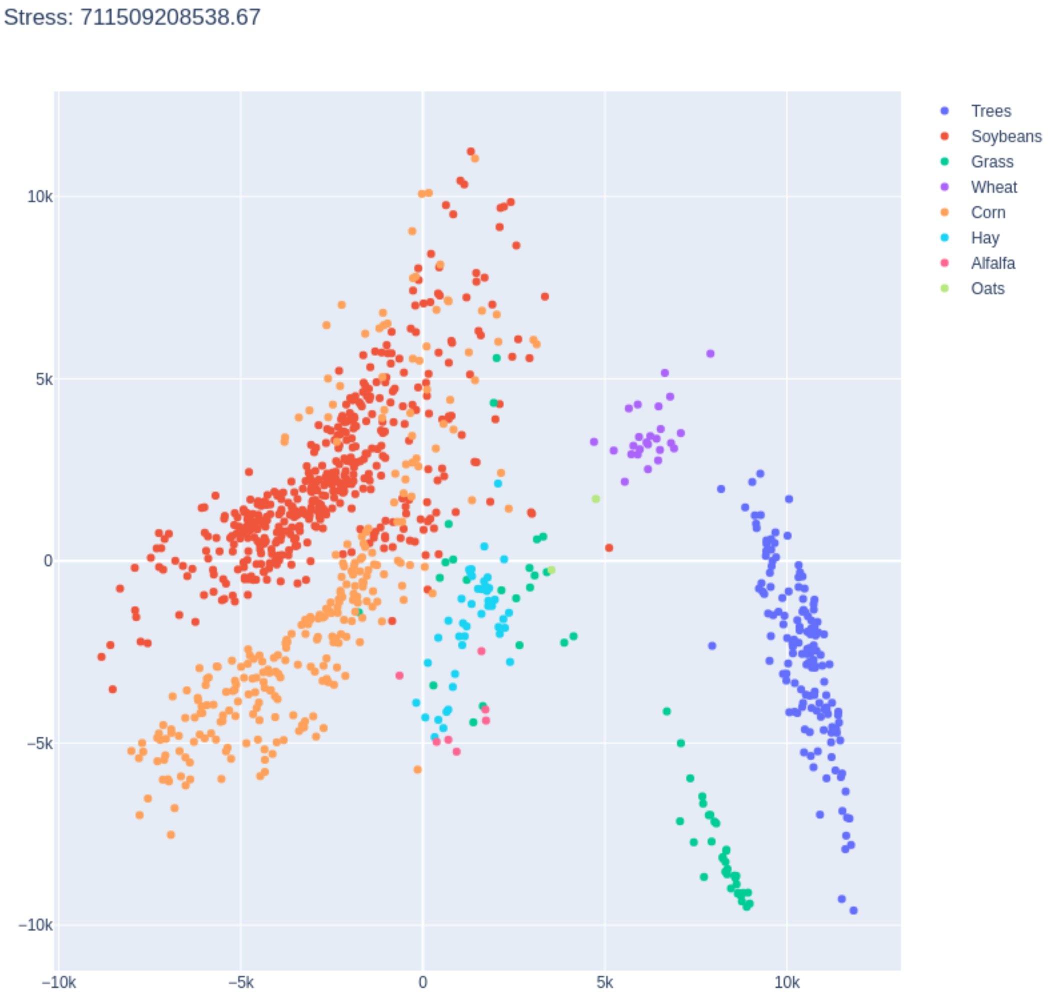


Figure 15: MDS result of the Indian Pine dataset in 2 dimensions

7.3 Categorical Variables with MCA

To try MCA we've looked for categorical datasets to visualize. We are going to visualize the Mushroom dataset[8] and the Soybeans dataset[9]. First we are going to see the Mushroom dataset. It contains information from about 1300 mushrooms in 22 variables that have to help us decide if a given mushroom is edible or poisonous. In Figure 16 we have plotted the result in 3 dimensions in order to maximize the amount of separation that we are able to see. We are able to determine that some of the clusters that MCA has created are very separated, and each of this clusters is probably a different mushroom species, and at the bottom we can see how the two classes overlap by a bit. We can deduce that some mushrooms that were analysed were fairly similar in terms of the attributes but correspond to different classes.

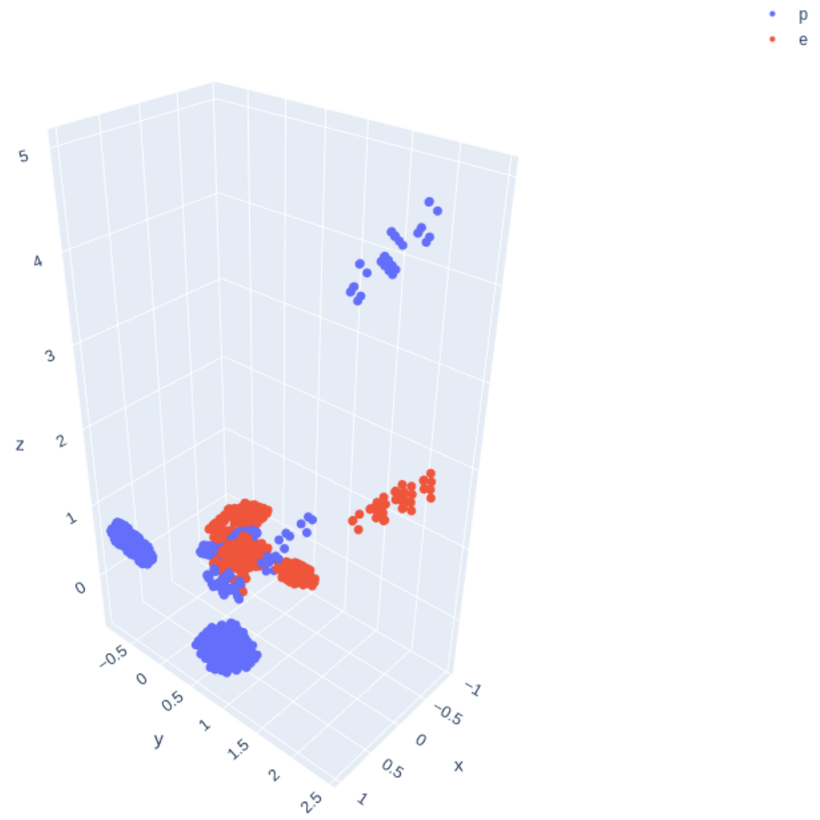


Figure 16: MCA result of the Mushroom dataset in 3 dimensions

The second dataset is about diagnosing soybeans. It contains 35 variables from which we could extract which disease a given soybean has and it contains a relatively low sample size, 307. At first glance we can see that MCA did not perform well separating the classes in this particular dataset, most of the samples are grouped in the same space, overlapping massively

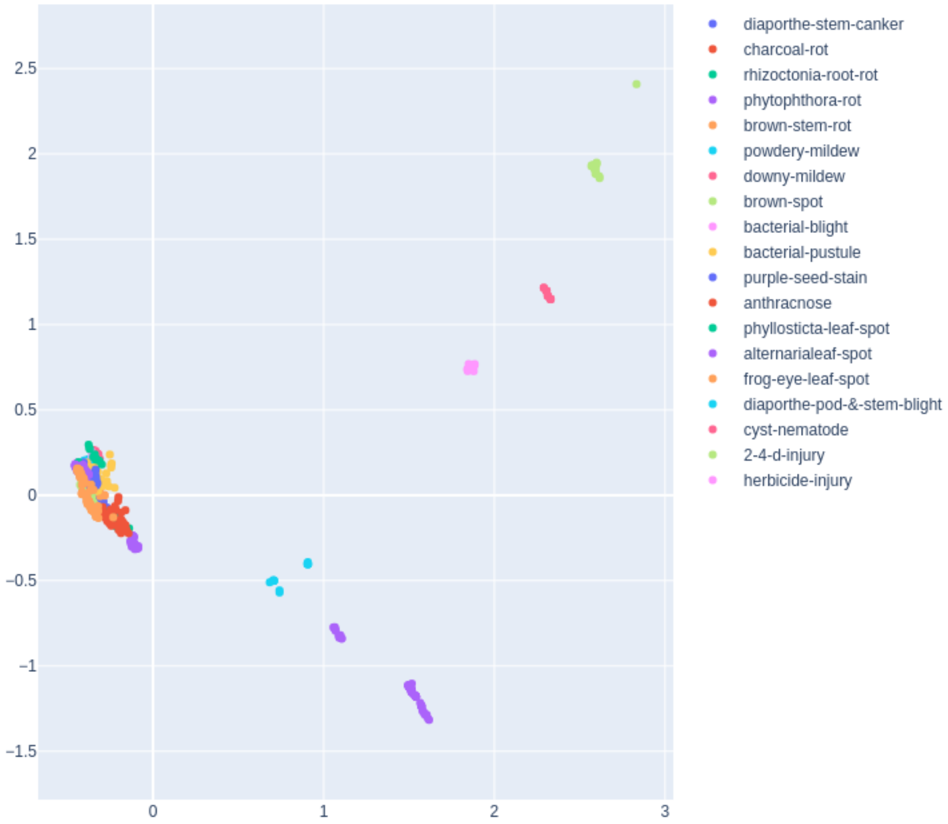


Figure 17: MCA result of the Soybeans dataset in 2 dimensions

8 Conclusions

In this work we've presented the problem of visualizing complex data, and presented a few solutions to one of the properties of complex data, high dimensionality. To overcome the Curse of Dimensionality we've seen how six different dimensionality reduction methods work, including linear and non-linear methods that allow us to transform our original data matrix into a lower dimensional matrix that keeps most of the important information needed. Each method has it's own way of dealing with this and as we've also seen, each of them produces very different visualizations when applied on the same dataset.

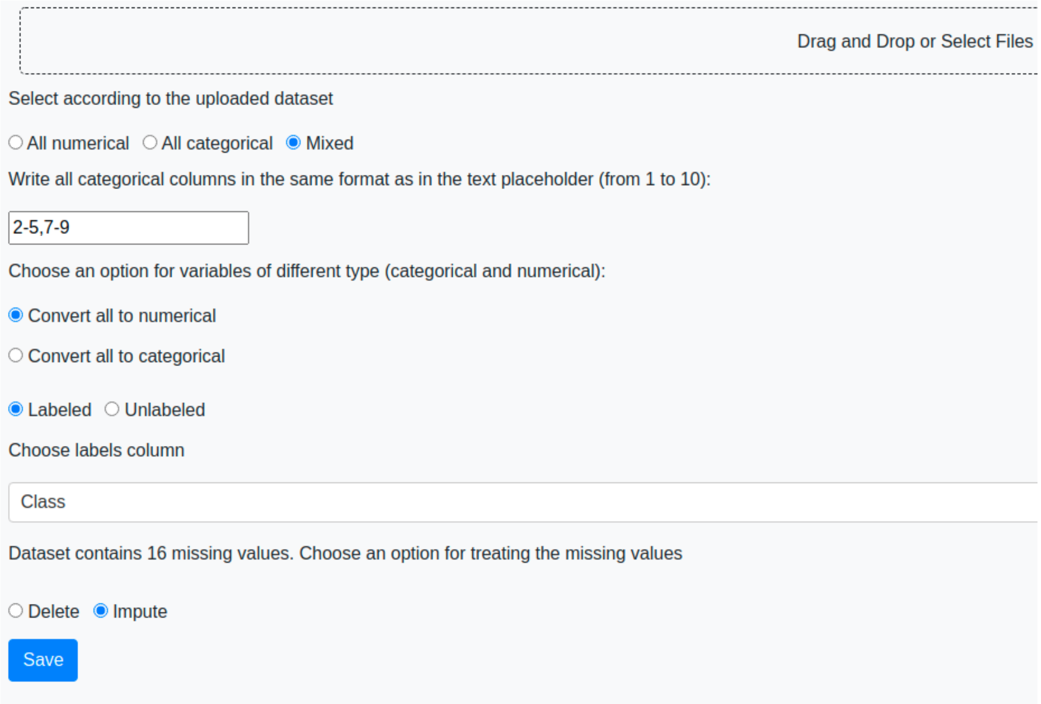
We've created this web application because we have not found any other similar tool that allows a user to visualize easily complex data. It serves a need that is not otherwise met by any other tool. This is, however, the very first step to being able to visualize truly complex data, as high dimensionality is only one of the *dimensions* of the complexity of data. More work is needed if we want to be able to visualize any arbitrary dataset in any way. For example more work should be done with the different data types given that in this work we've only seen numerical and categorical variables, but the real world data is not always like this. Even in the field of dimensionality reduction more work can be done, as we've only explored a small part of all the methods that exist for that purpose. Even so, this work is meant as an introduction, a first step towards the visualization of truly complex data.

A User Manual

In this appendix we're going to write a short user manual of the web application so any person reading this work can use it.

The project is Dockerized so it can be ran in any system without compatibility problems. A container with all the necessary packages is created and the web application can be open in any modern web browser.

Once we've opened it, the first interesting thing to do is to upload a dataset. In Figure 18 we have an example of how that will go. We first need to upload the dataset and then choose the options for it. It needs a bit of user inputs because there is information about the dataset that cannot be extracted automatically. We first need to tell the program if the dataset contains mixed data types or if it's all numerical or categorical. In case that we have a mixed dataset we will need to indicate all the categorical columns in the Input text box. We can write them in the form of intervals or individual numbers or both, separated by commas. Then we shall choose if we want to convert the categorical to numerical or the numerical to categorical. We recommend the former given that most of the methods used work with numerical values. After that we choose if the dataset uploaded is labeled or not, and if it is, we need to indicate the variable that contains the labels. Finally, if our dataset has missing values, we can choose what to do with them. The options are to delete them or to impute them. The method for imputing is discussed in section 4.1.



Drag and Drop or Select Files

Select according to the uploaded dataset

○ All numerical ○ All categorical ● Mixed

Write all categorical columns in the same format as in the text placeholder (from 1 to 10):

2-5,7-9

Choose an option for variables of different type (categorical and numerical):

● Convert all to numerical

○ Convert all to categorical

● Labeled ○ Unlabeled

Choose labels column

Class

Dataset contains 16 missing values. Choose an option for treating the missing values

○ Delete ● Impute

Figure 18: Upload view of the web application

After uploading the dataset, we can jump directly to visualize the data by choosing the desired method in the sidebar we see in Figure 19. The methods are the same we've seen in section 6. Aside from those options we have a view to see the dataset in table form in case that's convenient for anyone. The user can also choose the *Visualization* option if they want to create a visualization of one variable of the dataset against another.

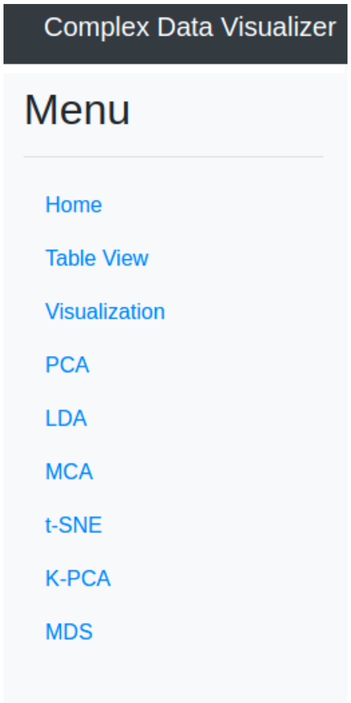
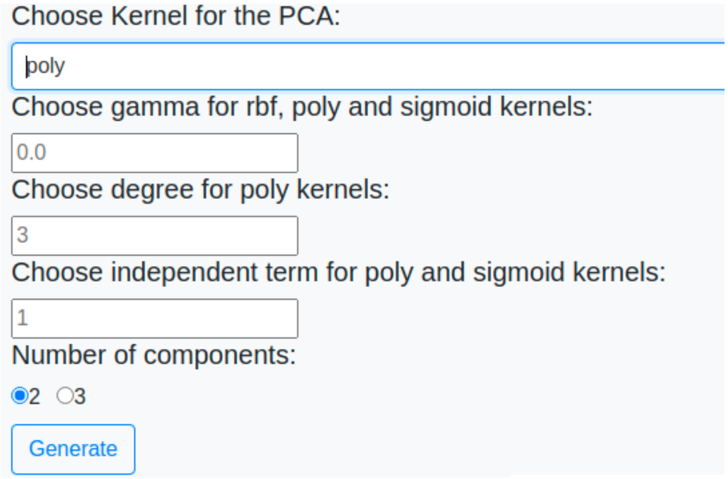


Figure 19: Sidebar links of the web application

Inside any given parametric method, we are given the options to fill the parameters of that method. For example in Figure 20 we see the options available for the Kernel PCA. We can choose the kernel to use with a Dropdown menu, and then, depending on the kernel chosen, parameters for said kernel will appear.



Choose Kernel for the PCA:

poly

Choose gamma for rbf, poly and sigmoid kernels:

0.0

Choose degree for poly kernels:

3

Choose independent term for poly and sigmoid kernels:

1

Number of components:

●2 ○3

Figure 20: Parameters for the Kernel PCA method

A fixed parameter will always be the number of components. That's what indicates how many dimensions we want to visualize, and consequently, the dimensionality at which we need to reduce the dataset. After we've chosen everything, we simply press the *Generate* button, and in a short time (depending on the method and on the dataset) we will have our visualization.

After generating a successful plot, we can chose to download it as a *png*. We can also download the points plotted and the matrix used to transform the original data.

In order to know how to tweak the parameters of the methods we can consult the documentation of the *sklearn* implementation[10].

References

[1] URL: https://www.glassdoor.es.

[2] URL: https://tarifasgasluz.com/comercializadoras/endesa/precio-kwh.

[3] URL: https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html.

[4] URL: https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html#sklearn.impute.SimpleImputer.

[5] URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.html.

[6] URL: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html.

[7] URL: https://www.kaggle.com/avnishnish/mnist-original.

[8] URL: https://archive.ics.uci.edu/ml/datasets/Mushroom.

[9] URL: https://archive.ics.uci.edu/ml/datasets/Soybean+%28Large%29.

[10] URL: https://scikit-learn.org/stable/user_guide.html.

[11] Hervé Abdi and Lynne J. Williams. "Correspondence Analysis". In: *Encyclopedia of Research Design.* (2010).

[12] Manuela Aparicio and Carlos Costa. "Data Visualization". In: *Communication Design Quarterly* (2015). URL: https://dl.acm.org/doi/10.1145/2721882.2721883.

[13] Sanjeev Arora, Wei Hu, and Pravesh K. Kothar. "An Analysis of the t-SNE Algorithm for Data Visualization". In: *Princeton University and Institute for Advanced Study* (2018).

[14] Kirk Baker. "Singular Value Decomposition Tutorial". In: (2005).

[15] Marion F. Baumgardner, Larry L. Biehl, and David A. Landgrebe. *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3*. 2015. DOI: doi:/10.4231/R7RX991C. URL: https://purr.purdue.edu/publications/1947/1.

[16] Ingwer Borg and Patrick J.F. Groenen. *Modern Multidimensional Scaling Theory and Applications.* 2005. ISBN: 0387251502.

[17] T. Tony Cai and Rong Ma2. “Theoretical Foundations of t-SNE for Visualizing High-Dimensional Clustered Data”. In: *Department of Biostatistics, Epidemiology and Informatics2* (2021).

[18] Chun houh Chen and Wolfgang Hardle. *Handbook of Data Visualization*. 2008. ISBN: 3540330364.

[19] Suchismita Das and Nikhil R. Pal. *Nonlinear Dimensionality Reduction for Data Visualization: An Unsupervised Fuzzy Rule-based Approach*. 2020. URL: https://arxiv.org/pdf/2004.03922.pdf.

[20] ERIK DEMAINE et al. “MULTIDIMENSIONAL SCALING: APPROXIMATION AND COMPLEXITY”. In: (2021).

[21] Tzeng F.-Y. and MA K.-L. “Opening the black box - data driven visualization of neural networks”. In: *Proceedings of IEEE Visualization* (2005).

[22] Michael Friendly and David Meyer. *Visualizing Categorical Data with R*. 2015. ISBN: 9781498725859.

[23] Ibrahim A Gaber T Tharwat A and Hassanien AE. “Linear Discriminant Analysis: A Detailed Tutorial”. In: (2017).

[24] Benyamin Ghojogh et al. “Multidimensional Scaling, Sammon Mapping, and Isomap: Tutorial and Survey”. In: (2020).

[25] Michael Greenacre. *Theory and applications of Correspondence Analysis*. 1993. ISBN: 0122990501.

[26] Thomas Hofmann and Joachim Buhmann. “Multidimensional Scaling and Data Clustering”. In: (1995).

[27] Thomas Hofmann, Bernhard Scholkopf, and Alexander J. Smola. “KERNEL METHODS IN MACHINE LEARNING”. In: *Darmstadt University of Technology, Max Planck Institute for Biological Cybernetics and National ICT Australia* (2008).

[28] R Huang et al. “Solving the Small Sample Size Problem of LDA”. In: *Proceedings of 16th International Conference on Pattern Recognition* (2002).

[29] Zomorodian A. J. “Topology for Computing (Cambridge Monographs on Applied and Computational Mathematics)”. In: *Cambridge University Press* (2005).

[30] Zoubin Ghahramani John P. Cunningham. “Linear Dimensionality Reduction: Survey, Insights, and Generalizations”. In: *Journal of Machine Learning Research 16* (2015).

[31] Cadima J Jolliffe IT. “Principal component analysis:a review and recent development”. In: *http://dx.doi.org/10.1098/rsta.2015.0202* (2016).

[32] Zhao Kaidi. “Data Visualization”. In: *National University of Singapore* (). URL: https://www.cs.uic.edu/~kzhao/Papers/00_course_Data_visualization.pdf.

[33] Anna Little, Yuying Xie, and Qiang Sun†. “Exact Cluster Recovery via Classical Multidimensional Scaling”. In: (2020).

[34] S. Liu et al. “Visualizing High-Dimensional Data: Advances in the Past Decade”. In: *EuroVis* (2015). URL: http://www.sci.utah.edu/~beiwang/publications/Vis_HD_STAR_BeiWang_2015.pdf.

[35] Jaap van den Herik Laurens van der Maaten Eric Postma. “Dimensionality Reduction: A Comparative Review”. In: *Tilburg centre for Creative Computing* (2009).

[36] Laurens van der Maaten and Geoffrey Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research 9* (2008).

[37] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Kernel Principal Component Analysis”. In: (1998).

[38] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Nonlinear Component Analysis as a Kernel Eigenvalue Problem”. In: (1998).

[39] Jon Shlens. “A tutorial on Principal Components Analysis, Derivation, Discussion and Singular Value Decomposition”. In: (2003).

[40] Lindsay I Smith. “A tutorial on Principal Components Analysis”. In: (2002).

[41] Teoh S. T. and Ma K.-L. “Paintingclass: interactive construction, visualization and exploration of decision trees”. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003).

[42] Martin Theus. “High Dimensional Data Visualization”. In: *University of Augsburg, Department of Computational Statistics and Data Analysis* (2008).

[43] Naveen Venkat. “The Curse of Dimensionality: Inside Out”. In: *Indian Institute of Science* (2018).

[44] Quan Wang. “Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models”. In: (2014).