People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Abderrahmane MIRA University of Béjaïa

Exact Sciences Faculty

Department of Operations Research

Master's thesis in Applied Mathematics

Specialty: Data Science and Decision Support

Prediction and modeling of chronic kidney disease (CKD) progression using machine learning algorithms

Presented by :

Maissa Meridji

Supervised by : Salima Kendi

Defended on 30/06/2025, before a jury composed of:

Larbi Asli	Chair	MCA	University of Bejaïa
Rabah Djabri	Examiner	MCB	University of Bejaïa
Zahra Bouzeria	Examiner	Ph.D. student	University of Bejaïa

Academic year : 2024-2025

Acknowledgments

I would first like to thank Almighty God for granting me the strength and patience to complete this work.

I express my sincere gratitude to Prof. Nabil Bellik for giving me the authorization to carry out this experiment, which was very close to my heart because my grandfather suffered from kidney failure before his death.

I would like to thank Dr. Salima Kendi for her guidance, and availability throughout this project.

I also thank the members of the jury for their time and constructive feedback.

Finally, I extend my heartfelt thanks to everyone who, directly or indirectly, contributed to the success of this work.

Dedications

I dedicate this work to:

My parents, for their unconditional love and unwavering support, especially my mom;
My brothers and sister, especially my bestfriend Sabrina Yala for their love, patience and encouragement;
My teachers, for the knowledge and values they have passed on to us;
My friends in the Operations Research program, for their team spirit.

Contents

List of Figures

General Introduction

Millions of people worldwide suffer from chronic kidney disease (CKD), an irreversible kidney disease that progresses over time. It often goes undetected in its early stages, when kidney function has already been seriously impaired. End-stage kidney failure may result from delayed treatment, requiring replacement therapy (dialysis or transplantation). Therefore, early CKD progression prediction is essential for therapy optimization and better patient outcomes.

As medical informatics has advanced, machine learning (ML) has emerged as a powerful tool for analyzing large clinical datasets, identifying patterns that are invisible to the human eye, and predicting how chronic diseases like chronic kidney disease (CKD) will develop. Using past clinical variables like creatinine, GFR, hemoglobin, phosphorus, calcium, ACR, urinary proteins, and specific antecedents like hypertension, diabetes, or anemia, this work aims to develop a predictive model that can forecast the progression from one clinical stage to the next (from normal to moderate, then to advanced).

The topic is divided into three chapters:

1. The theoretical and contextual background are covered in the first chapter. It covers CKD, risk factors, its stages according to GFR formulae (Cockcroft-Gault, MDRD, CKD-EPI), and a summary of machine learning algorithms used in predictive medicine.

2. The methodology of the study is explained in the second chapter. The database, pre-processing procedures (encoding, normalization, and missing value management), target variable construction, algorithm selection, and evaluation methods (patient cross-validation...) are all covered.

3. Experimental analysis is the main topic of the third chapter. It explains the constraints seen, compares performance with and without variable selection, explains the results generated by multiple models (Random Forest, SVM, XGBoost, KNN, Logistic Regression, Decision Trees), and offers a comparative analysis with earlier research in the field.

Finally, this work is closed by a general conclusion.

1

General information on chronic kidney disease (CKD)

Introduction

When the kidneys gradually lose their capacity to function properly over time, it is known as chronic kidney disease (CKD). It is divided into stages based on the degree of kidney damage and kidney function. Many people around the world are impacted by this serious medical issue. Over time, this illness progresses and affects kidney function, which reduces the kidney's ability to filter waste and extra fluid from the bloodstream, a vital process for urine production.

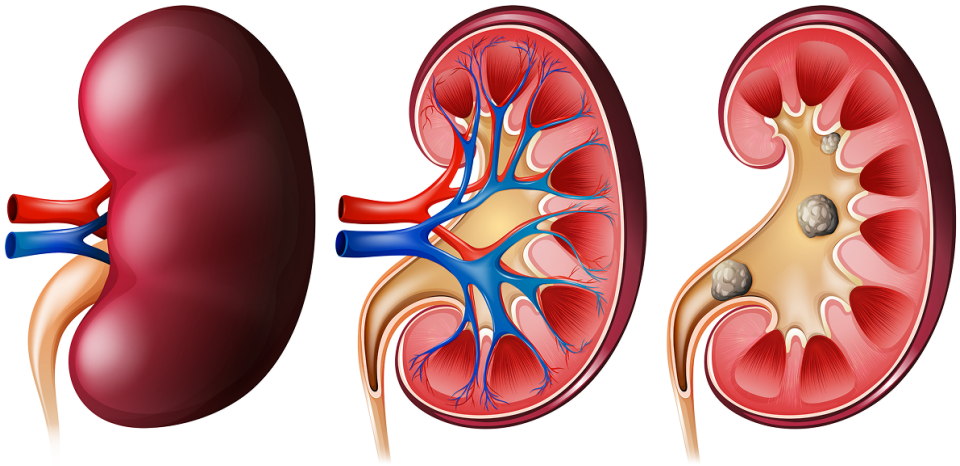


Figure 1.1: Kidney disease

1.1 What is CKD ?

1.1.1 Definition

Chronic kidney disease (CKD) is a chronic disorder that causes gradual loss of kidney function over time. It is commonly defined by an estimated glomerular filtration rate (eGFR) of less than 60 ml/min/1.73 m². CKD is also characterized by urinary abnormalities (proteinuria) and anatomical abnormalities of the kidney [14].

The word "chronic" describes the gradual and persistent progression of kidney damage. The illness is characterized by its silent nature, with symptoms often going undetected until the disease reaches more advanced stages. The fact that CKD affects individuals around the world shows how serious a healthcare issue it is. A serious medical condition known as chronic kidney disease (CKD) gradually impairs kidney function over the course of months or years [2].

1.1.2 Epidemiology

The frequency and incidence of CKD are difficult to determine because it is often unnoticed for a long time. Several studies have attempted to provide estimations [5]. CKD is a widespread disorder, with an estimated prevalence of 10.6% to 13.4% in the United States [22]. Over the past few decades, the prevalence has increased significantly due to rising rates of diabetes, hypertension, cardiovascular disease and a growing elderly population. CKD rises enormously with age and the presence of chronic comorbidities [22, 5].

Prevalence estimates for stages 1 and 2 in Europe range from 5.1% to 7%, for stage 3 from 4.5% to 5.3%, and for stage 4 from 0.1 to 0.4%. More than 1.7 million people in France are affected by stage 3 CKD [5].

Boucquemont [5] focused in her article on the incidence and prevalence of end-stage renal disease (ESRD) due to the challenges in measuring these factors. While patients are receiving treatment, it becomes easier to count cases. For instance, an independent patient association in France created since 1972, which is called REIN (Réseau Epidémiologie et Information en Néphrologie) offers data on transplantation, waiting list access, patient survival, dialysis treatment quality, chronic renal failure, replacement therapy, and patient characteristics. The REIN network gathers all patient data related to dialysis and transplant recipients. It covered every region of France in 2012. With 40,983 dialysis patients and 32,508 transplant recipients as of December 31, 2012, the overall prevalence of ESRD was 1,127 per million. 10,048 cases, or 154 per million, were the incidence rate.

The cost of the ESRD was estimated at around four billion euros in France in 2007. It is crucial to slow the progression of CKD in order to reduce the costs and the impact on the patient's life, even though thousands of people cannot afford to pay this much to save their lives [5].

1.1.3 CKD Staging Overview

Chronic Kidney Disease (CKD) is classified into five stages based on the severity of kidney disease. Figure 1.2 shows that understanding these stages is essential for the management of related health concerns, particularly cardiovascular disease (CVD) [15] :

- **Stage 1:** Normal or high eGFR (>90 mL/min) with kidney damage indicators (e.g., proteinuria).
- **Stage 2:** Mild decrease in eGFR (60-89 mL/min) with kidney damage.

- **Stage 3:** Moderate decrease in eGFR (30-59 mL/min), often subdivided into :
  - Stage 3a: 45-59 mL/min.
  - Stage 3b: 30-44 mL/min.
- **Stage 4:** Severe decrease in eGFR (15-29 mL/min), indicating significant kidney impairment.
- **Stage 5:** End-stage renal disease (eGFR <15 mL/min), often requiring dialysis or transplantation.

Figure 1.2: Various eGFR scores of CKD stages realized by PowerPoint

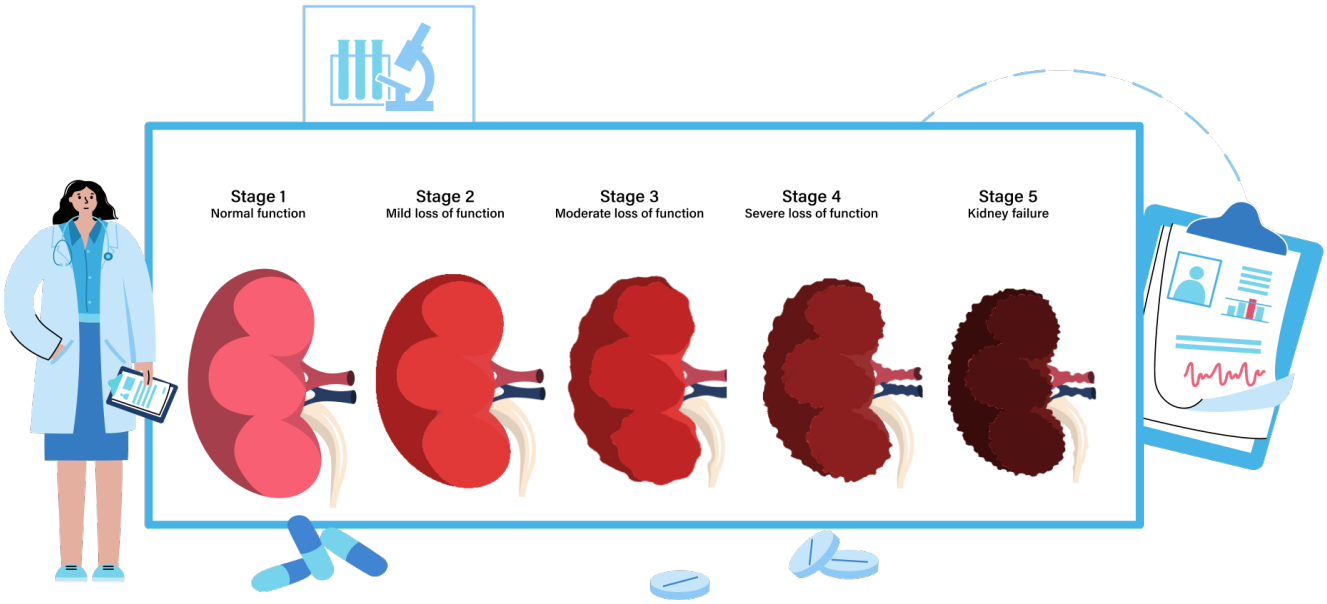


Figure 1.3: The five stages of CKD realized by Photopea

1.2 Risk factors and main causes

Although early stage CKD is common, only a small percentage of patients went to later stages. However, comorbidity and mortality rates are significantly higher in the advanced stages of CKD. There is limited understanding of the factors that impact the evolution of CKD and the transition between phases [22].

Diagnoses were made using clinical criteria, with glomerular and proteinuria disorders requiring a kidney sample [22]. Proteinuria can lead to the development of the disease and it is a sign of kidney impairment. Only 100 to 150 mg of protein are excreted in urine daily under normal kidney conditions. A kidney problem is indicated by an excessive amount of protein in the urine. Because proteinuria impairs kidney function, it may promote the progression of chronic kidney disease [5].

The presence of uncontrolled diabetes increases the risk of kidney damage from proteinuria. Glomerular sclerosis and subsequent loss of kidney function can be caused by elevated blood sugar levels [5].

There is also elevated blood pressure. Untreated blood pressure can lead to kidney artery progressive sclerosis. Blood pressure may rise as a result of hormonal changes and poor kidney function. Thus, arterial hypertension may be considered both a cause and an effect of the CKD progression. Proteinuria interacts with it [5] ;

Ischemic heart disease, peripheral artery disease, cerebrovascular disease, and congestive heart failure were among the comorbidities reported at baseline [22] ;

Smoking is another important risk factor. Whether in the general population or in particular populations like diabetics or hypertensives, smoking has been linked in a number of studies to the development of kidney lesions. According to other studies, stopping smoking slows the progression of chronic kidney disease [5] ;

Other risk factors for progression were age, sex, albuminuria, and underlying kidney disease [23].

1.3 Consequences of CKD growth

Chronic kidney disease (CKD) leads to a variety of serious health effects, impacting both patient quality of life and healthcare systems. As CKD progresses, there is an increased risk of kidney failure, cardiovascular events, and death. The following sections discuss the major effects of CKD progression [23].

A study proved that women experienced a lower death rate than men, both men and women were more likely to die in advanced stages of CKD. Women in

all CKD phases, except G5, had decreased cumulative incidence of fall-cause death prior to KRT. The updated Fine and Gray model used in that study revealed that women were 10% less likely to die than men [22].
Cardiovascular diseases were the leading cause of death in both genders and at all stages of CKD [22].

1.4 Key biological parameters and their units of measurement

Biological parameters play a crucial role in evaluating health and physiological states and can be assessed using a variety of procedures and units. These characteristics include a variety of measures, each with its own unit of measurement [15] :

- **Red cells :** Measured in million per microliter (10^6/µL) their average value is : (4.5 – 5.9) for men and (4.1 – 5.1) for women.
- **White cells :** Measured in cells per microliter (µL) (4,000 – 10,000).
- **Platelets :** Measured in platelets per microliter (µL) (150,000 – 400,000).
- **Hemoglobin :** Measured in milligrams per deciliter (mg/dL) ((13 – 17) for men and (12 – 16) for women).
- **Creatinine :** Measured in milligrams per liter (mg/L) ((7 – 13) for men and (6 – 11) for women).
- **Clearance :** Measured in milligrams per liter (mg/L) (15 – 45).
- **Uric acid :** Measured in milligrams per liter (mg/L) ((35 – 72) for men and (26 – 60) for women).

- **Blood sugar :** Measured in milligrams per liter (mg/L) (7 – 9.9).
- **Phosphorus :** Measured in milligrams per liter (mg/L) (25 – 45)
- **Calcium:** Measured in milligrams per liter (mg/L) (85 – 105).
- **Albumin :** Measured in milligrams per liter (mg/L) (34000 – 50000).
- **Sodium :** Measured in milligrams per liter (mmol/L) (135 – 145).
- **Potassium :** Measured in milligrams per liter (mmol/L) (3.5 – 5.0).
- **Chlorine :** Measured in milligrams per liter (mmol/L) (96 – 106).
- **HbA1c:** It displays a global memory of glycemia over the preceding three months, measured in % (percentage) (< 5.7 (normal) / 5.7–6.4 (pre-diabetes) / > 6.5 (diabetes)).
- **Parathyroid Hormone (PTH) :** Measured in picograms per milliliter (pg/mL) (10 – 65).
- **Urea :** Measured in milligrams per liter (g/L) (0.15 – 0.45).
- **Blood pressure :** Measured in millimeters of mercury (mmHg), eg : (120/80 mmHg)
    - 120 mmHg = Systolic pressure (when the heart contracts)
    - 80 mmHg = Diastolic pressure (when the heart relaxes)
- **24hours Protein :** Measured in milligrams per 24hours (mg/24h).
- **Urinary Albumin-Creatinine Ratio (ACR) :** Measured in milligrams per grams (mg/g).

1.4.1 Definition

ACR (Albumin to Creatinine ratio): is a ratio obtained from a spot urine sample, preferably the first morning urine. It indicates the amount of albumin (a particular protein) in relation to creatinine in urine, adjusting for hydration-related variations in urine concentration. ACR is primarily used to detect and quantify microalbuminuria, particularly in diabetics or those at risk for kidney disease [7].

Table 1.1 expresses albuminuria categories :

Albuminuria categories (urinary albumin / creatinine) mg/g)		
Categorie	**Value**	**Signification**
A1	< 30	Normal to slightly increased
A2	30–300	Moderately increased (Microalbuminuria)
A3	> 300	Severely increased (Macroalbuminuria)

Table 1.1: Classification of albuminuria categories according to KDIGO criteria [15]

1.4.2 Definition

24-hour proteinuria: it refers to the total amount of protein excreted in the urine during a 24-hour period. This approach provides a direct estimate of the entire amount of protein excreted by the kidneys for a whole day, including albumin and other urine proteins [7].

Table 1.2 expresses proteinuria categories :

24-hour Proteinuria Categories (mg/24h)		
Category	**Proteinuria (mg/24h)**	**Interpretation**
P1	< 150	Normal
P2	150–500	Slightly increased
P3	500–3000	Moderately increased (subnephrotic range)
P4	> 3000	Severely increased (nephrotic range)

Table 1.2: Classification of 24-hour proteinuria levels (mg/24h) [15]

1.5 GFR estimation methods

Estimating glomerular filtration rate (GFR) is critical for assessing kidney function. Several methods are available, each with unique benefits and limits. The accuracy of those tests can vary greatly depending on the patient demographic and therapeutic context. The following are significant GFR estimate methods identified in recent investigations [18].

- **Estimation equations :**
- **CKD-EPI Equation :** This equation accurately estimates GFR in healthy individuals and is recommended for general usage. It outperforms MDRD. It is used most by nephrologists and understood to be the most accurate means of safely evaluating GFR in the united states. However, it continues to underestimate kidney function [18].

$$\begin{aligned} \text{eGFR}_{\text{CKD-EPI}} = 141 \times \min\left(\frac{\text{Scr}}{\kappa}, 1\right)^{\alpha} \times \max\left(\frac{\text{Scr}}{\kappa}, 1\right)^{-1.209} \\ \times\ 0.993^{\text{Age}} \times \text{Sex} \times \text{Race} \end{aligned} \tag{1.1}$$

  - Scr : represents the serum creatinine

- **Creatine-Based Equations:** The Cockcroft-Gault and MDRD equations, commonly used, have been criticized for their limitations, especially in patients with severe illness where direct measurement is recommended [18].
- **Cockcroft-Gault equation :** Its was the first and the most well-known estimating equation. It estimates GFR but may not be applicable to broader demographics and it may be inaccurate depending on a patient's body weight [18].

$$\mathrm{Cl}_{Cr} = \frac{(140 - \mathrm{Age}) \times \mathrm{Weight\ (kg)} \times (0.85 \text{ if female})}{72 \times \mathrm{Scr}} \tag{1.2}$$

- **MDRD equation :** This equation is widely used but tends to misclassify kidney function, particularly in individuals with normal or mildly reduced GFR. It has been criticized for overestimating CKD prevalence [18].

$$\begin{aligned} \mathrm{eGFR}_{\mathrm{MDRD}} = 175 \times \mathrm{Scr}^{-1.154} \times \mathrm{Age}^{-0.203} \\ \times\ (0.742 \text{ if female}) \times (1.178 \text{ if Black}) \end{aligned} \tag{1.3}$$

1.6 Importance of progression prediction

Predicting CKD progression allows healthcare providers to identify people at increased risk of rapidly declining kidney function. This enables earlier and more targeted therapies, which may delay disease development [2] and higher risk patients could receive more intense testing, intervention and early nephrology care [23].

Lower risk individuals might be handled by the primary care by physicians without further testing of CKD problems and by identifying high-risk individuals with CKD stage 3, the cost effectiveness of CKD care will increase [23].

It also helps to avoid emergency situations by planning for dialysis or transplantation, rather than facing unforeseen hospitalizations [2].

1.7 The role of Artificial Intelligence (AI) and Machine Learning

Machine learning (ML) is a subset of artificial intelligence that trains machines to learn, while AI is a discipline that tries to replicate human abilities. Machine learning is a type of artificial intelligence that enables computers to learn from prior events [21], rather than relying on predetermined equations. Increasing the number of learning examples improves the adaptive performance of the algorithms [1].

Machine learning can be considered as an extension of artificial intelligence. Indeed, a system that is incapable of learning cannot be called intelligent. The ability to learn and gain from experience is critical for a system meant to adapt to changing conditions. Artificial intelligence (AI) encompasses cognitive science, neuroscience, logic, electronics, engineering, and other disciplines to create intelligent robots. The phrase "artificial intelligence" is becoming more popular than "machine learning" due to its imaginative connotations [1].

Conclusion

This chapter has explored chronic kidney disease, its various stages, risk factors and causes, the consequences of its progression, and, finally, how to avoid kidney failure. In the following chapter, we will examine various techniques to preventive diagnosis that employ machine learning algorithms to forecast kidney disease progression.

2

Machine Learning methods and application context

Introduction

Computers are successfully tackling complicated learning challenges. They have achieved tasks which were previously thought impossible, such as learning physics from experimental data, and becoming video game experts [9].
As the number of companies specializing in complicated data analysis has increased, it is not surprising that some analytics organizations are focusing on healthcare issues. Deo [9] has investigated in his article whether challenges in medicine might benefit from such learning strategies. His goal was to identify and address impediments to implementing statistical learning methodologies in medical practice, eg : heart failure clinical trial.

2.1 What is Machine Learning ?

2.1.1 Definition

The term "Machine learning" was used for the first time by **Arthur Samuel** on 1959, he defined it as a fascinating scientific field that looks at how computers learn from data. **The checkers software**, which learned to play the game by

competing with itself and got better as the game went on, was the first famous example he developed hence one of the first instances of automatic supervised learning was this one [9].

He also precised that ML combines statistics (the study of data relationships) and computer science, which focuses on efficient computing algorithms. The science of statistics, optimization, algorithmics, and signal processing has evolved into an important element of modern culture. This technology, which has been used for decades in automatic character recognition and spam filters, is now being used to [1] :

- Preventing bank fraud.
- Recommending products based on our preferences.
- Detecting faces in camera viewfinders.
- Translating texts between languages.

Azencott [1] mentioned in her article that Fabien Benureau has proposed a definition that is applicable to a company, a human being, and a computer program :"Learning is the adjustment of behavior based on previous experience."

2.1.2 Why do we use Machine Learning ?

Machine learning can solve problems we couldn't solve before, such as image recognition or language comprehension. It can also solve problems we know how to solve but require resource-intensive procedures, such as predicting interactions between larger molecules [1]. Machine learning is thus applied when there is a lot of data, but the expertise is inaccessible or underdeveloped [9].

In this approach, machine learning can help humans to learn : The models generated by learning algorithms can indicate the relative relevance of different

pieces of information or how they interact with each other to solve a specific problem. This element of machine learning is commonly employed in scientific research, and there are several examples [1]:

- Which genes contribute to the formation of a specific type of tumor, and how ?
- What areas of a brain imaging predict behavior ?
- What properties of a molecule make it an appropriate medication for a specific indication ?
- Which features of a telescopic image can be utilized to identify a specific celestial object?

2.2 The various types of Machine Learning

Machine Learning is a vast field, learning algorithms can be classed based on the learning mode used. This section outlines the key problems that machine learning addresses [1].

2.2.1 Supervised learning

If the classes are predetermined and the examples are known, the system learns to classify using a classification model, which is known as supervised learning [16].

Azencott [1] mentioned in her book that supervised learning begins with the purpose of predicting a known outcome or target. Handwriting recognition is a common supervised learning task in machine learning competitions, where participants are evaluated on performance on shared datasets. Recognition skills

include handwriting recognition, item classification (eg : cat or dog), and document classification (eg : heart failure clinical trial or financial report), It also involves classifying new data instances and predicting unknown parameters (for example, the temperature in San Francisco tomorrow afternoon).

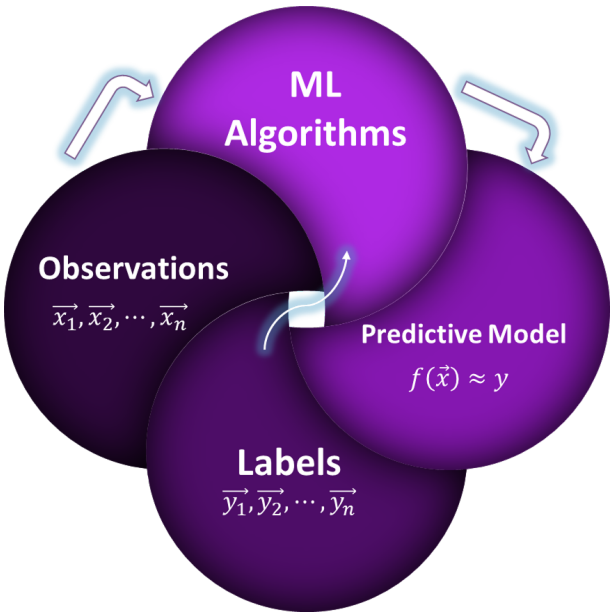


Figure 2.1: Supervised Learning realized by PowerPoint

2.2.2 Unsupervised learning

In contrast, unsupervised learning has no predictable outputs. Instead, Deo [9] mentioned in his topic that he aims to identify natural patterns or groupings in the data. Unsupervised learning groups are generally graded based on their performance in later supervised learning tasks, which can be difficult to judge.

Unsupervised learning, also known as clustering, occurs when a system or operator has just instances without labels and the number and types of classes are not specified. No expert is available or needed [1].

When might such approaches be applied in medicine ?

The "precision medicine" project is a very intriguing opportunity. To address the underlying heterogeneity of common diseases, researchers are working to rede-

fine them based on pathophysiological principles, potentially leading to novel treatment options. But identifying mechanisms for complicated complex illnesses will not be simple [9].

The algorithm must independently find the hidden structure of the data. To classify examples into homogeneous groups, the system must target the data in the description space (the sum of the data) based on their accessible qualities. The distance function is commonly used to calculate similarity between pairs of samples. The operator is responsible for determining the meaning for each group [16].

Example

To identify explanatory hypotheses in a large group of liver cancer victims, an epidemiologist can use a computer to differentiate between groups and associate them with factors such as geographical origin, genetics, alcoholism, or exposure to heavy metals or toxins [9].

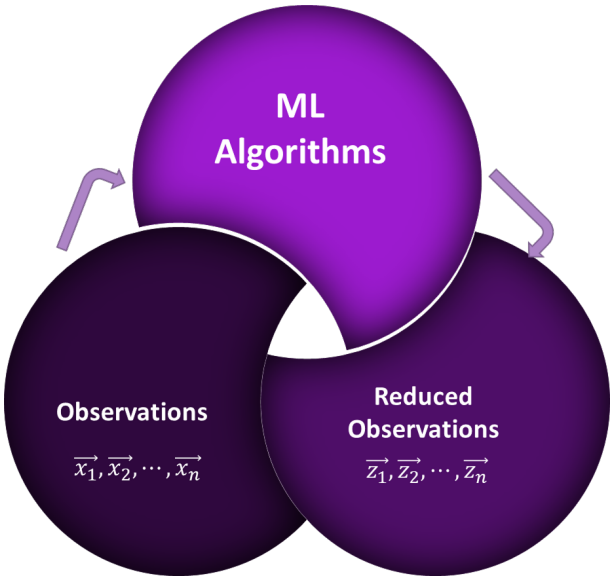


Figure 2.2: Unsupervised Learning realized by PowerPoint

2.3 Methods of supervised learning

2.3.1 Logistic Regression

Logistic regression is a supervised learning method in machine learning, used mainly for binary classification tasks. It calculates the likelihood of a binary result using one or more predictor variables. Recent improvements have demonstrated its effectiveness, particularly in huge datasets, by novel sampling strategies and practical applications in a variety of disciplines [19].

- **Advantages** :

  1. It calculates probabilities for binary outcomes, making it easier to understand the relationship between variables and outcomes. It can handle both continuous and categorical data, allowing the adjustment of numerous predictors, which is critical in observational studies.

  2. Logistic regression is well-established and may be applied to high-dimensional data, outperforming some current algorithms in some sit-

uations.

3. The method is freely available in statistical software, promoting its use across several study disciplines [19].

- **Inconveniences** :

1. Coefficients given in log odds might lead to confusion, especially when compared to risk ratios .
2. Logistic regression findings are not collapsible, which might complicate model comparisons and lead to erroneous conclusions.
3. The approach presupposes a linear relationship between the outcome's log chances and the predictors, which may not always be accurate [19].

Example

Predicting if a patient has type 2 diabetes [9].

2.3.2 Decision Trees

Principal of trees

Decision Trees are a popular and practical supervised learning algorithm that can solve a variety of problems, including classifications and regressions. Each node in the tree represents a condition on a variable, and each of its children is a possible reaction to that situation. of its children represents a possible response to this circumstance and tree leaves correspond to labels [4].

To anticipate an observation's label, we follow test responses from the root of

the tree and return the label of the leaf we arrive at. Mayou and Belhachani [16] mentioned that its structure is similar to a flowchart, with internal nodes representing characteristics, branches representing decision rules, and leaf nodes representing results. This non-parametric algorithm makes no assumptions about data distribution, facilitating decision-making.

Feature selection measurement

The main challenge in building a decision tree is determining the best feature for the root node and how to best separate the data. To address this issue, there is a technique known as Attribute Selection Measuring (ASM), which includes two main and widely used measures [16]:

1. **Gini Index**

$$f(p) = p(1-p) \tag{2.1}$$

2. **Information gain**

$$f(p) = -plog(p) \tag{2.2}$$

An algorithm for creating a decision tree

The algorithm is summarized as follow [4] :

1. Generate the root node N.
2. If all samples belong to the same class C, return node N as a leaf node with the class label C.
3. If no feature is found, return N as the leaf node with the most common class among samples.
4. Use the feature selection measure to identify the best feature.

5. Label node N with the feature found in step 4 (test feature).
6. For each value v_i of the test feature.
7. Divide samples and create sub-trees for each value based on test feature.
8. Let a_i be the collection of tuples for which the test feature is v_i.
9. If a_i is empty, attach a leaf node with the most common class from the samples.
10. Otherwise, attach the node returned by Generate decision tree

- **Advantages** :
  1. Decision Trees are extremely simple and fast.
  2. Does not require domain expertise or parameter settings, and can handle high-dimensional data.
  3. High accuracy (depending on available data) [4].
- **Inconveniences** :
  1. Training takes a lengthy time since each level of the tree requires a single run over the training tuples in the dataset to find the best split.
  2. Insufficient memory while working with huge databases.
  3. Decision Trees can be overly complex for certain concepts due to replication issues [4].

Example

Decision Trees could be used to predict the state of heart attacks [9].

2.3.3 Random Forest

Random Forest is an extremely popular supervised learning technique. It is also useful for regression and classification difficulties. Based on a collection of learning algorithms, which is the act of mixing multiple algorithms to solve a complex problem and enhance model performance. The technique constructs multiple decision trees (thus the name "forest") on different subsets of the data [16].

The algorithm analyzes each tree's forecasts and predicts the eventual outcome based on their votes. The figure below explains the algorithm's functioning and structure [16].

Random Forest construction algorithm

The algorithm is summarized as follow [16] :

1. Take random samples from a training dataset.
2. Create a decision tree for each sample (subset). Then compute the prediction results for each decision tree.
3. To earn new points, vote for each expected result.
4. Choose the prediction that received the most votes as the final outcome.

Random Forest advantages :

1. It is among the most accurate learning algorithms known. It generates a highly accurate classifier for a wide range of data sets.
2. It works well with huge databases.
3. It efficiently estimates missing data and maintains accuracy even when a significant amount of data is missing [16].

Random Forest inconveniences :

The random forest approach has a problem of being slow and inefficient for real-time forecasts due to the enormous size of trees. These algorithms are quick to train, yet sluggish in making predictions after training. Adding additional trees to improve prediction accuracy leads to poorer performance [16].

2.3.4 Support Vector Machine (SVM)

SVM is a widely used supervised method for classification and regression. However, it is mostly used for categorization in machine learning. The SVM algorithm aims to generate a decision line that divides the n-dimensional space into classes, making it easier to assign additional data points to the appropriate class in the future. The optimal choice boundary is known as a hyperplane [16]. SVM selects the extreme points/vectors that contribute to the hyperplane. The algorithm known as a support vector machine is named after the severe scenarios it addresses [4]. Diagrams 2.3 show two classes (blue points and purple triangles) that are classified with a hyperplane :

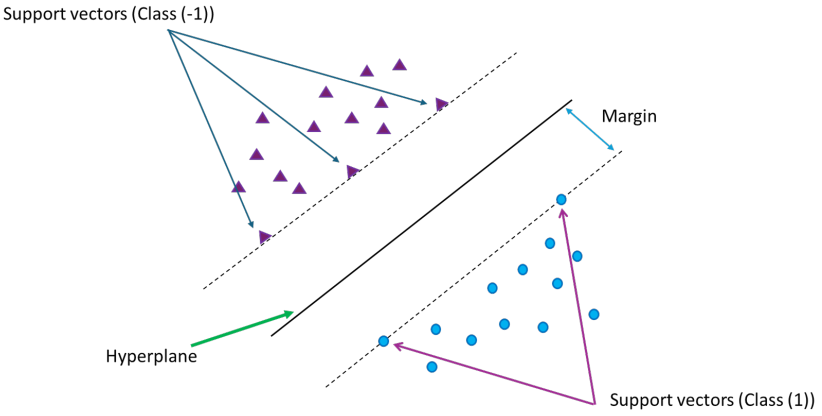


Figure 2.3: Perfect separation of two classes with a hyperplane

SVM advantages :

1. SVM is less likely to overfit than other approaches.
2. Supports unstructured and semi-structured data, including text, pictures, and trees.
3. It determines the optimal classification function to distinguish between two classes in the training data [4].

SVM inconveniences :

1. It's susceptible to noise.
2. SVMs are slow to learn and require extensive training time.
3. Classification of more than two classes is problematic [16].

Example

Deo [9] mentioned an example in his article that SVM can be used to predict the risk of myocardial infarction by using clinical variables such as blood pressure, cholesterol level, age, body mass index, and family history.

2.3.5 K-Nearest-Neighbor (KNN)

Bhavsar and Ganatra [4] mentioned that the K-Nearest-Neighbors technique is one of the simplest machine learning algorithms. It's is non-parametric, instance-based learning method. Instance-based classifiers, also known as lazy learners, store all training examples and generate a classifier only when a fresh unlabeled sample is needed for classification.

KNN performs classification or prediction based on a set number of data points nearest to the input point. This means that for a given value of K, an entrance point would be classed or belong to the same class as the closest class of the number of nearby K points [16].

KNN construction algorithm

The algorithm is summarized as follow [16] :

1. Choose the number K of neighbors.
2. For each example in the dataset:
    - Determine the distance between the query example and the current example using the data.
    - Add the example's distance and index to the ordered collection.
3. Sort the distances and indices from smallest to largest (in increasing order) based on distance.
4. Choose the first entries from the k collection.
5. Assign the example query to the class with the highest number of neighbors (more frequent).

KNN advantages :

1. Can be applied to classification and regression [16].
2. The classification technique is simple to understand and implement.
3. Perform effectively in applications with many class labels [4].

KNN inconveniences :

1. They have high storage requirements.
2. Classification performance is slower due to delayed computing.
3. They lack a logical approach to selecting k, instead relying on computationally expensive cross-validation techniques [4].

Example

Deo [9] mentioned that k-nearest neighbors estimate outcomes based on similar cases rather than developing a model. To predict a patient's likelihood of having a heart attack, it's helpful to look at similar cases in other patients.

2.3.6 XGBoost

XGBoost is a powerful decision tree group that uses gradient boosting. It, generates an additive extension of the objective function by minimizing a loss function. To manage the complexity of decision trees, XGBoost uses a variant of the loss function [3].

$$\mathscr{L}_{\text{xgb}} = \sum_{i=1}^{M} L(y_i, F(x_i)) + \sum_{m=1}^{K} \Omega(h_m) \tag{2.3}$$

1. $\mathscr{L}_{\text{xgb}}$: XGBoost global objective function,
2. $L(y_i, F(x_i))$: loss function between true value y_i and prediction $F(x_i)$,
3. M : total number of examples in the training set,
4. $\Omega(h_m)$: regularization term for the complexity of the m-th tree,

5. K : total number of trees built by the model.

Here is the regularization expression :

$$\Omega(h) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.4}$$

- T : number of leaves,
- w_j : leaf weight ,
- γ, λ : regularization hyper-parameters,

XGBoost advantages :

1. High performance.
2. Automated handling of missing values.
3. Advanced regularization algorithms help to prevent over-learning [3].

XGBoost inconveniences :

1. Not easily interpretable.
2. Long computing time for large datasets.
3. Parameter-sensitive [3].

2.4 Modeling process for Machine Learning

To apply supervised learning we divide the process into six steps [16] :

- Collect data that includes our examples.
- Preparation of data.
- Select the appropriate model(s).

- Training the model.
- Testing the model.
- Improving the model.

2.4.1 Data collection

First, we need to collect the data required for machine learning, consolidated into a single flat table [16].

2.4.2 Data preparation (wrangling)

Data wrangling refers to preparing data for machine learning algorithms [16].

- **Data cleaning :**

  Locate nulls, missing values, and duplicate data. Replace or delete null and missing values and avoid duplicates. To decompose data, we have to separate text columns that contain multiple pieces of information into specialized columns [16].

  Before evaluating the data, we have to deal with missing value analysis to avoid biased or inaccurate results. Emmanuel et al. [10] mentioned in their topic several approaches in the literature to address missing values, such as :

  1. **Simple imputation :** this strategy involves replacing missing data for each individual with a quantitative or qualitative property of all non-missing values. With simple imputation, missing data is handled by different approaches such as mode, mean, or median, of the available values [10].

     They may lead to biased or unrealistic conclusions in high-dimensional

datasets, with the rise of big data, this approach appears to perform badly and may not be suitable for large data sets [10].

2. **Advanced imputation :** the KNN technique identifies the nearest neighbors of missing data and uses a distance metric between instances for imputation. Its downside is that it looks through the entire dataset, which increases computing time [10].

   Euclidean distance is the most extensively utilized distance metric, as it is reported to provide efficiency and productivity when categorizing nearest neighbors of missing values for imputation [10]. We will describe KNN imputation using the Euclidean distance metric below :

$$\text{Dist}_{xy} = \sqrt{\sum_{k=1}^{m} (X_{ik} - X_{jk})^2} \tag{2.5}$$

   - Dist_{xy} : is the squared euclidean distance between instances x and y,
   - m : is the total number of features (variables),
   - X_{ik}, X_{jk} : are the values of the k-th feature for instance i and j [10].

- **Preliminary statistical tests :**

  Several statistical tests were performed in this work to examine variable relationships, compare means, and determine categorical data dependencies. The following tests were chosen: the Chi-squared test, Pearson correlation test, Student's t-test, and Analysis of Variance (ANOVA). Each test was chosen based on the type of variables and the hypotheses being investigated [13].

1. **Chi-squared test (Chi²)** :

   The Chi-squared test is a statistical test that determines whether there

is a significant relationship between two category variables. It compares the actual frequencies in a contingency table to the anticipated frequencies if the variables were independent [13].

Type of variables : categorical variables (gender, diabetes, stage classification...) [13].

Test statistic : the Chi-squared statistic is calculated as:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - T_{ij})^2}{T_{ij}} \tag{2.6}$$

where :

- n_{ij} is the observed frequency in the i-th row and j-th column,
- T_{ij} is the expected frequency under the assumption of independence,
- r and c are the number of rows and columns in the contingency table [13].

2. **Pearson correlation test** :

The Pearson correlation coefficient assesses the degree and direction of a linear link between two continuous quantitative variables. We'll use it to determine if there is a substantial linear relationship between two numerical variables [11].

Test statistic:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.7}$$

3. **Student's t-test** :

The Student's t-test is used to see if there is a significant difference

in the means of two independent groups. It is assumed that the data follows a normal distribution with equal variances between the two independent groups [11].

Type of variables : one quantitative variable and one binary categorical variable (e.g., comparing creatinine levels between male and female patients) [11].

Test statistic:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \tag{2.8}$$

With :

$$v_c^2 = \frac{(n_1 - 1)v_1^2 + (n_2 - 1)v_2^2}{n_1 + n_2 - 2} \tag{2.9}$$

Application in this study : comapring the mean glomerular filtration rate (eGFR) between two groups of patients classified by disease stage [13].

4. **Analysis of variance (ANOVA)** :

   It is a statistical approach for comparing the averages of three or more independent groups. It assists in determining whether at least one group mean differs statistically from the others [17].

   Test statistic:

$$F = \frac{\text{variance between treatments (MSB)}}{\text{variance within treatments (MSW)}} = \frac{\text{SSB}/(k-1)}{\text{SSW}/(n-k)} \tag{2.10}$$

   Application in this study : comparing mean creatinine levels across different stages of chronic kidney disease [17].

- **Data scaling :** obtain data on a common scale, if not already available. Data scaling does not apply to labels, categories, or their columns. It is necessary when there is significant variance in feature ranges [16].
- **Data shaping and transformation :** from categorical to digital [16].
- **Data visualization :** analyzing our data to identify connections between columns. Charts allow for comparing features and identifying connections between them, as well as between features and labels using different tests that we mentioned before [16].
- We may wind up with a massive number of columns. To pick which columns to use as features, a dimensional reduction is necessary for large datasets [16].

2.4.3 Data separation

To separate our dataset we must divide it into three categories: training, testing, and validation [16].

The algorithms that we mentioned before will be trained using training data, while test data will be used to verify performance. Validation data will only be used at the end of the process to prevent bias [16].

- **Training the model**

  Here's where the magic happens ! The dataset is linked to an algorithm that uses advanced mathematical modeling to learn and make predictions [16]. These algorithms are often classified into one of three categories:

  1. **Binary** : Divide into two categories.
  2. **Classification** : Divide into multiple categories.

3. **Regression** : Predict a numeric.

A Machine Learning model differs from the previous one as it is not based on a mathematical demonstration or equation. Instead, it's constructed using data, similar to a statistical model [16].

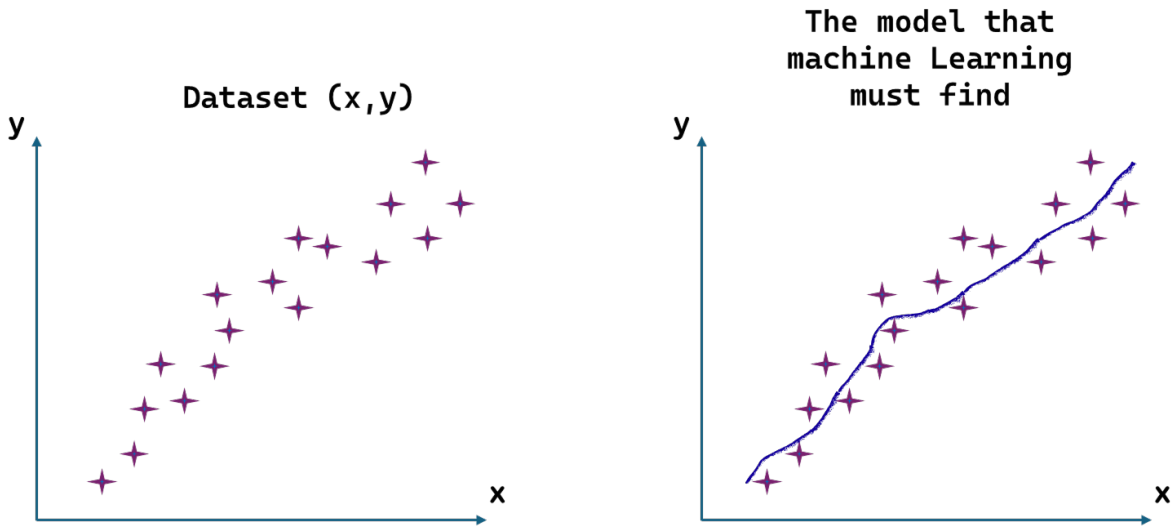


Figure 2.4: Work training overview

In practice, it is up to us to select the type of model (i.e. the mathematical function), and it is up to the machine to determine the coefficients of this function that will be used to calculate the model function. By convention, we refer to these coefficients as model parameters [16].

For example, we could create a linear model and let the machine determine the value of machine such as Logistic regression and SVM, which produces the best results. Alternatively, we may use a non-linear model (Random forest or XGBoost), such as one with parameters, they are able to capture complicated nonlinear interactions due to their tree-like structure. The alternatives are limitless, but we'll learn later in this course how to choose one model over another [16].

- **Testing the model**

It's time to validate our trained model. Using the test data, we evaluate the model's accuracy. To select the optimal model, the machine must be capable of measuring the performance of existing models [16].

To determine which model is superior among two options, we must evaluate them. To do so, we calculate the error between a model and the data, which we refer to as the **Cost Function** [16].

In the case of **regression**, for example, we can quantify the difference between the model's prediction and the value associated with it in our data. This is related to the concept of calculating the distance between our arrow and the center of the target, which is where it is supposed to arrive. In the case of **classification**, we can construct our Cost Function by quantifying the number of samples in the data set that our model will misclassify with our decision frontier [16].

2.4.4 Cross-validation

Cross-validation is a crucial method in machine learning for assessing classification model performance and generalization and address concerns such as overfitting [20]. It entails dividing the dataset into training and testing subsets, allowing for a systematic assessment of model accuracy across various data splits. Various approaches exist, each with unique advantages and downsides, which are critical for picking the optimal methodology depending on individual dataset features and the modeling goal but in this paper we'll only introduce one approach which is **k-fold cross-validation** [24].

The k-fold cross-validation approach is widely utilized. In k-fold cross validation, data is divided into k subgroups of equal size. The model is trained with $k-1$ folds and tested on the final fold. The process is repeated k times

using a different fold as the test set each time, here is the steps of realizing the algorithm [20] :

Algorithm 1 K-fold Cross-Validation Algorithm

Require: Integer K such that K divides n

1: Construct a partition $V_1, V_2, \ldots, V_K$ of $\{1, \ldots, n\}$

2: **for** $k = 1$ to K **do**

3: Define $A_k = \{1, \ldots, n\} \setminus V_k$

4: Train a model $\hat{f}_{D_{A_k}}$ on the dataset $D_{A_k} = \{(x_i, y_i) \mid i \in A_k\}$

5: Compute risk:

$$R_k = \frac{K}{n} \sum_{i \in V_k} \ell(y_i, \hat{f}_{D_{A_k}}(x_i))$$

6: **end for**

7: **Output:** Estimated risk

$$\frac{1}{K} \sum_{k=1}^{K} R_k$$

2.4.5 Selection of evaluation metrics

The selection of assessment metrics in machine learning is critical for effectively assessing model performance, particularly in classification tasks. Common measures include accuracy, precision, recall, F1-score, and AUC-ROC, each of which provides distinct insights into model performance. However, the metrics used should be context-dependent, taking into account the task's specific qualities and requirements. This answer investigates the measures' strengths and limits, as well as their applicability in various contexts [25].

- **Accuracy :**

  Accuracy is a simple metric that calculates the percentage of correctly identified cases out of all instances. It is extensively used since it is simple. However, accuracy can be misleading in imbalanced datasets where

one class dominates, because it may not reflect the true performance of the model on minority classes [25].

- **Precision :**

  Precision calculates the fraction of true positive cases among those anticipated as positive, which is useful in situations where false positives are costly [25].

- **Balanced accuracy :**

  Balanced Accuracy is a version that addresses the class-frequency-weighted aspect of accuracy. In the binary case, it is comparable to re-scaling informedness [25].

- **Recall :**

  Recall, or sensitivity, estimates the fraction of real positive instances among all actual positive instances, which is important when missing positive instances is costly [25].

- **F1-Score :**

  The F1-score is the harmonic mean of precision and recall, offering a balance between the two, especially in imbalanced datasets [25].

- **AUC-ROC :**

  The AUC-ROC statistic assesses the model's ability to differentiate between classes at various threshold values, giving a complete picture of performance. It is especially useful in binary classification problems and dealing with imbalanced datasets, as it is less affected by class distribution [25].

2.4.6 Optimizing and adjusting hyperparameters

A hyperparameter adjusting step was used to improve the predicted performance of each algorithm. This means experimenting with different combinations of the previously mentioned parameters to see which ones minimize the risk of overfitting while maximizing cross-validation scores [12].

Pre-learned settings known as hyperparameters affect the structure and functionality of the model. Instead of being learned instantly from data, they need to be defined by the user, in contrast to the model's fundamental parameters [12].

To select combinations with the highest validation scores, 5-fold cross-validation and grid search (GridSearchCV) were used for hyperparameter improving for each model. Here are some basic optimization models such as :

- **Grid search :**

  This approach that we will use in our study, thoroughly examines a preset list of hyperparameter values for a specific algorithm. This entails determining relevant hyperparameters to consider. Some have real values, while others have category values (like the kind of SVM kernel). The choice of hyperparameter values might depend on other choices [6].

  Following the definition of the different hyperparameter configurations, the algorithm's performance is evaluated for each configuration. Lastly, the configuration that performs the best is given back [6].

  The configuration that is produced by using a grid search is frequently better than the default configuration. Usually, designers restrict the number of possible values and pre-define the grid. The goal is to cut down on the quantity of searches and the amount of time spent on them. Thus,

we can argue that these systems autonomously perform a simple form of hyperparameter optimization [6].

- **Random search :**

  Random search randomly explores the configuration space. The space of alternatives needs to be defined in advance, just like in the previous scenario. It is not necessary to discretize real-valued hyperparameters, though. Just indicate the distribution and interval to be used in order to sample values [6].

2.4.7 Limits and challenges of Machine Learning in healthcare

Even with the remarkable advancements in machine learning (ML) in the healthcare sector, specifically in the areas of diagnostic assistance, medical imaging, and the prediction of chronic illnesses, there are still many unanswered questions regarding the clinical application of these technologies. These limitations cover organizational, legal, ethical, and human issues in addition to technical ones [8].

The quality of the data used to train an algorithm and the capacity of experts to comprehend and evaluate its output are closely related to the algorithm's effectiveness. According to the Chaire Santé Sciences Po [8] and other academic publications, this section attempts to outline the main challenges related to the application of ML in healthcare.

- **Health data challenges :**

  Better treatment outcomes are made possible by empowering people to actively participate in their healthcare journey through personal health data. Therefore, it is essential to promote users' active participation. Data en-

hances research and system efficiency in addition to improving individual care. Combes [8] mentioned in her book that the gathering of sizable databases needed to train algorithms is hampered by confidentiality issues, legal and ethical restrictions continue to limit the exchange of data between hospitals or institutions without forgetting that medical data is often noisy, biased, inconsistent, or incomplete.

- **Ethical and legal challenges :**
  1. If an AI decides on a wrong diagnosis or course of treatment, then there will be a responsible person that would pay for that.
  2. Ethical issues arise because patients don't always understand how AI systems work [8].
- **Challenges of integration into the healthcare system :**
  1. The current hospital information systems must be integrated with AI tools.
  2. Algorithms cannot be validated by the same standards as pharmaceuticals.
  3. Typically, caregivers do not receive training in how to use or comprehend machine learning tools [8].

Conclusion

Machine Learning methods for identifying chronic kidney failure are presented in this chapter, lowering the possibility of negative outcomes for patients. K-Nearest-Neighbors, Decision Trees, Random Forest, Support Vector Machine, XGBoost, and Logistic Regression are some of the algorithms that will be applied to our dataset in this study.

3

The practice of prediction and modeling through machine learning

Introduction

This last chapter will outline our topic's purpose, objectives, duration, and location. In order to correct outliers and choose the best model to use, we set up our data collection, including its features and pre-processing procedures (clean, explore, and model selection).
Finally, in order to shed light on how each task was performed, we will evaluate and contrast our findings with those of other studies, as well as offer criticism.

3.1 Materials and methods

3.1.1 Study's location and duration

This is a retrospective study that examines historical data to determine the evolution of chronic kidney disease in **124** patients admitted to the nephrology consultation service at the Frantz-Fanon Hospital in Bejaia.
During a 2-month period from January 06, 2025 to the end of March 31, 2025.

3.1.2 Folders

During this crucial brief time, we examined a significant number of folders and came to the conclusion that they included a variety of significant pieces of information, such as :

- Patient's information such as full name, e-mail address, phone number, age, home...
- Personal and family history.
- Biological tests.
- Management plans or clinical monitoring that expresses medical actions or decisions to be considered or implemented for patients.
- Renal ultrasound and renal MRI (magnetic resonance imaging).

3.1.3 Excel files

In purpose of realizing our study we based on implementing our medical data in an excel folder in the ".xlsx", format that contains several rows and columns to help us organize our data.

3.1.4 Tools and libraries used

We statistically processed the acquired data using the Python programming language (version 3.11), a powerful statistical data analysis tool that contains several libraries that are useful for the next steps.

- **Jupyter notebook :**

  Jupyter Notebook supports multiple programming languages, including

Python. It allows us to build documents that include code, equations, visualizations, and text. It has numerous applications, including data purification and transformation, numerical simulation, statistical modeling, data visualization, and machine learning.

- **Python :**

  Python is a multi-paradigm programming language and a dominating programming language in data science, with multiple implementations that add to its appeal. Python is a top choice for machine learning due to its numerous high-quality libraries that offer both ease of use and powerful functionality. These are the following libraries that we used in our study :

  - **NumPy :**

    NumPy is a Python extension that allows us to manage multidimensional arrays.

  - **Pandas :**

    Pandas is another Python library for data manipulation and analysis. Its data cleaning feature reduces the time spent cleaning data in machine learning projects, as many datasets have empty or null fields that can negatively impact models.

  - **Matplotlib :**

    Matplotlib is a Python package that supports static, animated, and interactive visualizations.

  - **Seaborn :**

    Seaborn is a Python data visualisation library based on matplotlib. It provides a high-level interface for creating appealing and informative statistical graphs.

- **Scikit-learn :**

  She is the most essential Python library for automatic learning, as it contains numerous algorithms such as random forests, logistic regressions, classification algorithms, and support vector machines.

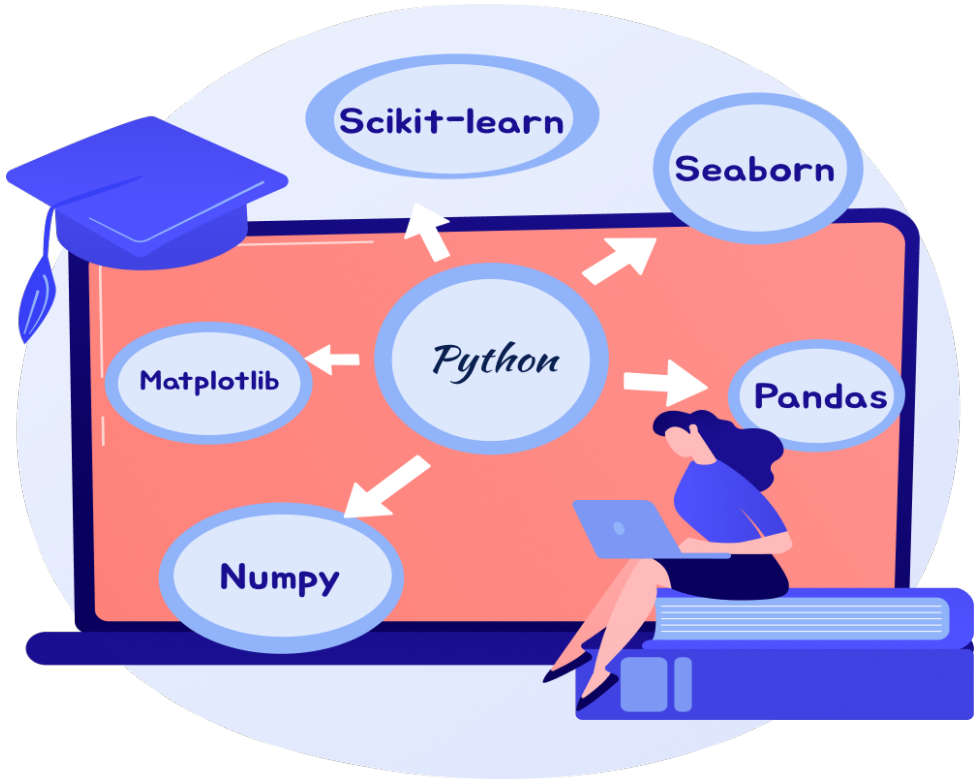


Figure 3.1: Python's libraries realized by Figma

3.1.5 Statistical analysis

We conducted quantitative and qualitative studies to address the research hypotheses given in the study.

Pearson's correlation test will be used to determine the significance of correlations between two continuous quantitative variables (parameters), followed by simple regression analysis to establish linear correlation curves and track the evolution of the variable to be explained as a function of the explanatory variable (if the correlation is significant).

The Chi-square independency test is used to assess the dependencies between categorical variables. The significant p-value was set to the standard 0.05. Differences in averages and correlations between variables are considered significant at $p < 0.05$.

To determine the impact of a qualitative element on a quantitative variable, we compare numerous averages using Student's test if there are only two groups, and the ANOVA test if there are more than two groups.

3.2 Database presentation

3.2.1 Data source and description

The nephrology consultation service was the original source of this dataset. Since chronic kidney failure affects every member of the population we have defined. Based on specific diagnostic measures included in the dataset, the dataset's goal is to diagnostically predict whether a patient may advance from a moderate to a severe stage of the disease. Several limitations were applied to the selection of these cases from a bigger database such as time.

In fact, all of the patients (124) are Algerian men and women over the age of 16. The sample size is **1069** rows, so **20** rows can represent a single patient, which means 20 consecutive measures (visits, days, months, and years) and **29** columns for variables.

3.2.2 Variables description

Our database is composed of three types of variables which are numerical variables, categorical variables, and temporal ones, described as follows:

- **Categorical variables :**

1. **Sex :** male or female.
2. **Diabetes :** which is represented by yes or no (if that patient has diabetes or not).
3. **HBP :** it represents high blood pressure.
4. **Anemia**.

- **Numerical variables :**

  The numerical variables are age, weight, length, red.c (red blood cells), white.c (white blood cells), Hb (hemoglobin), creatinine, clearance, uric acid, blood sugar, phosphorus, calcium, albumin, sodium, potassium, chlorine, HbA1c, PTH, urea, BP (blood pressure), protein (24h protein), and ACR.

- **Temporal variables :**

  We have only one which is "Date".

There are also some other variables that would be created and used as **targets** such as : eGFR-CKD-EPI and Stage-CKD.

3.3 Data pre-processing

3.3.1 Importing libraries

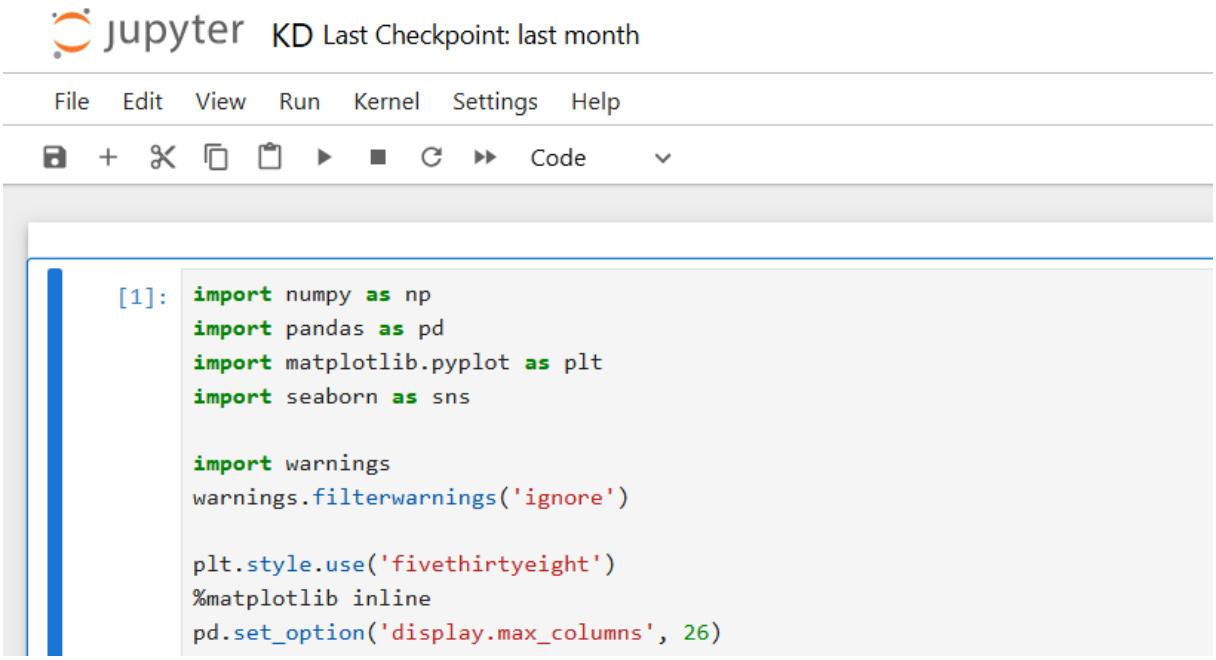
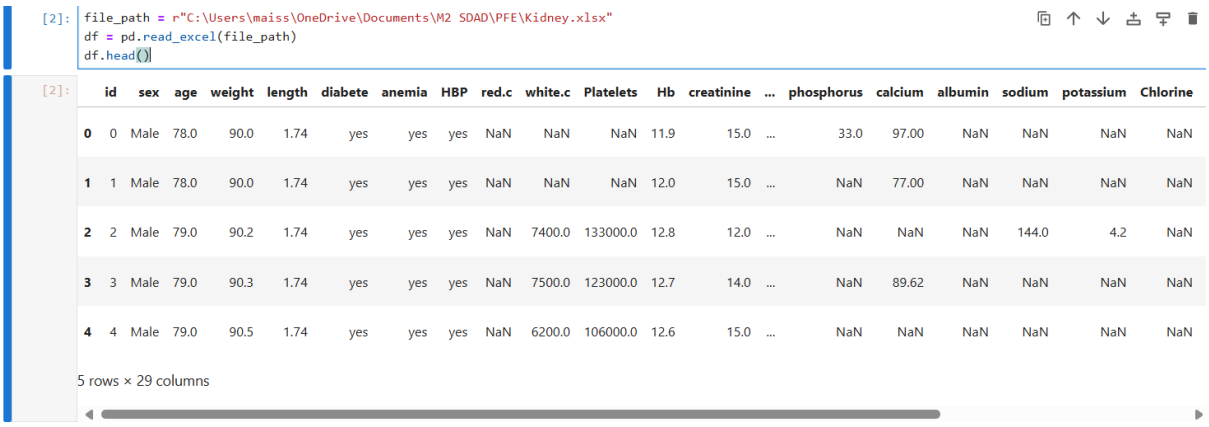


Figure 3.2: Python's libraries captured from Jupyter notebook

3.3.2 Data uploading



```
[2]: file_path = r"C:\Users\maiss\OneDrive\Documents\M2 SDAD\PFE\Kidney.xlsx"
df = pd.read_excel(file_path)
df.head()
```

[2]:

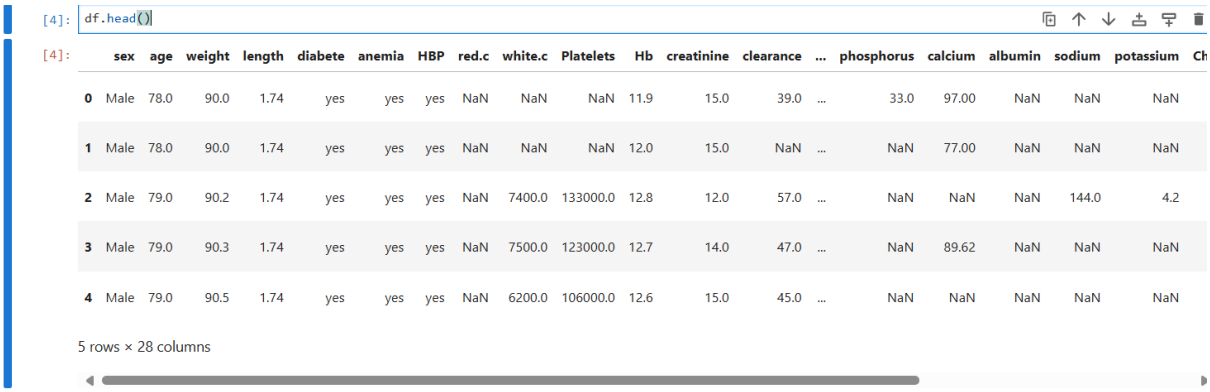
	id	sex	age	weight	length	diabete	anemia	HBP	red.c	white.c	Platelets	Hb	creatinine	...	phosphorus	calcium	albumin	sodium	potassium	Chlorine
0	0	Male	78.0	90.0	1.74	yes	yes	yes	NaN	NaN	NaN	11.9	15.0	...	33.0	97.00	NaN	NaN	NaN	NaN
1	1	Male	78.0	90.0	1.74	yes	yes	yes	NaN	NaN	NaN	12.0	15.0	...	NaN	77.00	NaN	NaN	NaN	NaN
2	2	Male	79.0	90.2	1.74	yes	yes	yes	NaN	7400.0	133000.0	12.8	12.0	...	NaN	NaN	NaN	144.0	4.2	NaN
3	3	Male	79.0	90.3	1.74	yes	yes	yes	NaN	7500.0	123000.0	12.7	14.0	...	NaN	89.62	NaN	NaN	NaN	NaN
4	4	Male	79.0	90.5	1.74	yes	yes	yes	NaN	6200.0	106000.0	12.6	15.0	...	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 29 columns

Figure 3.3: Uploading the database

3.3.3 Data manipulation

After uploading our database, we are going to start by showing only the first five rows after deleting the column "id" hence to have a better visualization :



[4]: df.head()

[4]:

	sex	age	weight	length	diabete	anemia	HBP	red.c	white.c	Platelets	Hb	creatinine	clearance	...	phosphorus	calcium	albumin	sodium	potassium	Ch
0	Male	78.0	90.0	1.74	yes	yes	yes	NaN	NaN	NaN	11.9	15.0	39.0	...	33.0	97.00	NaN	NaN	NaN	
1	Male	78.0	90.0	1.74	yes	yes	yes	NaN	NaN	NaN	12.0	15.0	NaN	...	NaN	77.00	NaN	NaN	NaN	
2	Male	79.0	90.2	1.74	yes	yes	yes	NaN	7400.0	133000.0	12.8	12.0	57.0	...	NaN	NaN	NaN	144.0	4.2	
3	Male	79.0	90.3	1.74	yes	yes	yes	NaN	7500.0	123000.0	12.7	14.0	47.0	...	NaN	89.62	NaN	NaN	NaN	
4	Male	79.0	90.5	1.74	yes	yes	yes	NaN	6200.0	106000.0	12.6	15.0	45.0	...	NaN	NaN	NaN	NaN	NaN	

5 rows × 28 columns

Figure 3.4: Showing the first five rows

```
df.shape
```

```
(1069, 28)
```

Figure 3.5: Exploring rows and columns

As we saw in figure 3.4 where we just uploaded our dataset, it was somehow full of missing values "NaN", and we need to clean them all by filling those gaps instead of deleting the rows to avoid reducing the dataset.

3.3.4 Imputation of missing values

Figure 3.6 indicates the percentage of missing data for each variable, which corresponds to a certain group. Then we will perform two types of imputations (basic using the median and complex ones using the KNN imputer) .

As figure 3.7 shows, all the missing values had been treated.

HBP	0.0	0-20%
red.c	87.27783	80-100%
white.c	26.099158	20-50%
Platelets	22.918616	20-50%
Hb	17.867166	0-20%
creatinine	10.757717	0-20%
clearance	57.998129	50-80%
uric acid	72.029935	50-80%
Blood sugar	87.184284	80-100%
phosphorus	63.891487	50-80%
calcium	58.465856	50-80%
albumin	84.190833	80-100%
sodium	28.811974	20-50%
potassium	28.063611	20-50%

Figure 3.6: Grouping the different missing values

✔ Pourcentage de valeurs manquantes après imputation :

sex	0.0
age	0.0
weight	0.0
diabete	0.0
anemia	0.0
HBP	0.0
white.c	0.0
Platelets	0.0
Hb	0.0
creatinine	0.0
clearance	0.0
uric acid	0.0
phosphorus	0.0
calcium	0.0
albumin	0.0
sodium	0.0
potassium	0.0
PTH	0.0
Urea	0.0
Protein	0.0
Date	0.0

Figure 3.7: The imputation algorithm with the final check

3.4 Descriptive analysis

To better understand the structure of the clinical variables, a descriptive statistical analysis is necessary in our study before using any predictive and modeling technique.

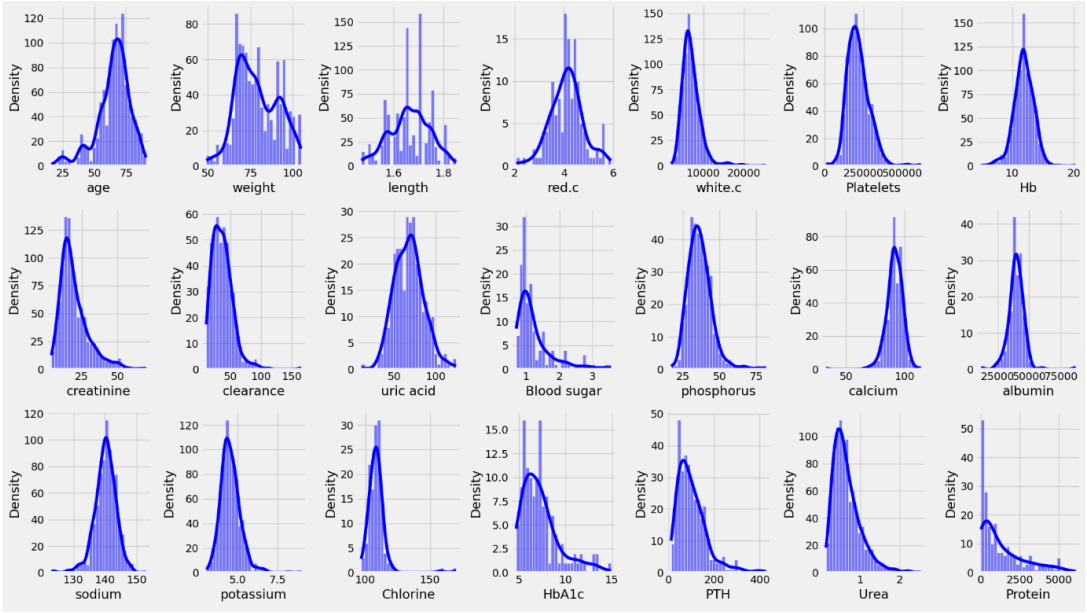


Figure 3.8: Distribution of numerical variables

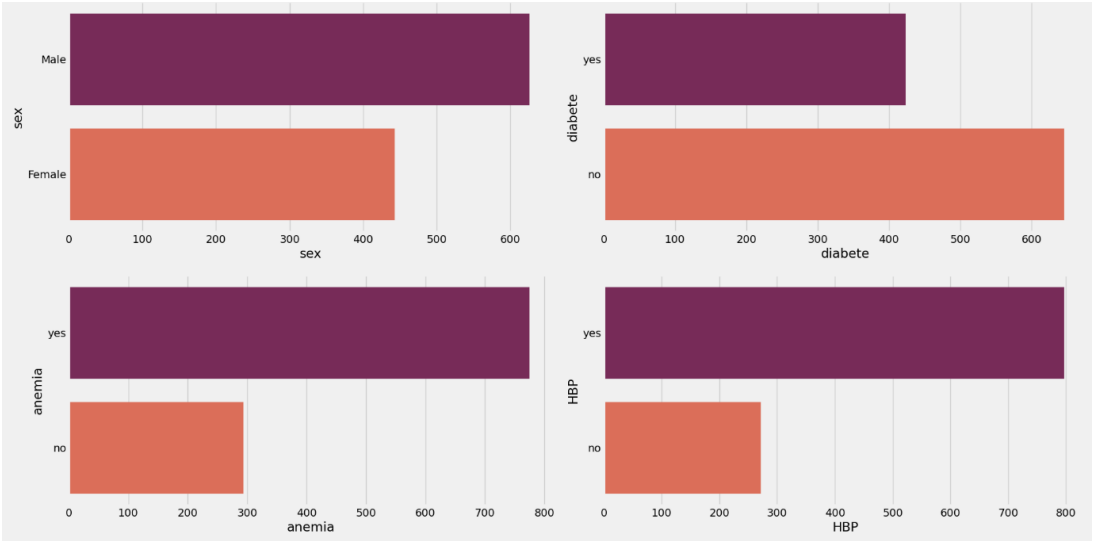


Figure 3.9: Distribution of categorical variables

3.4.1 Uni-variate descriptive analysis

- **Numerical variables :**

  Figure 3.10 shows the distribution of serum creatinine , one of the clinical biomarkers for evaluating kidney function, it shows that the min value is 3.9 mg/L and the max value is 70 mg/L, we have done the same thing with the rest of other variables.

```
Analyse de creatinine :
count    954.000000
mean      19.743753
std        9.976581
min        3.900000
25%       13.000000
50%       17.000000
75%       24.000000
max       70.000000
Name: creatinine, dtype: float64
```

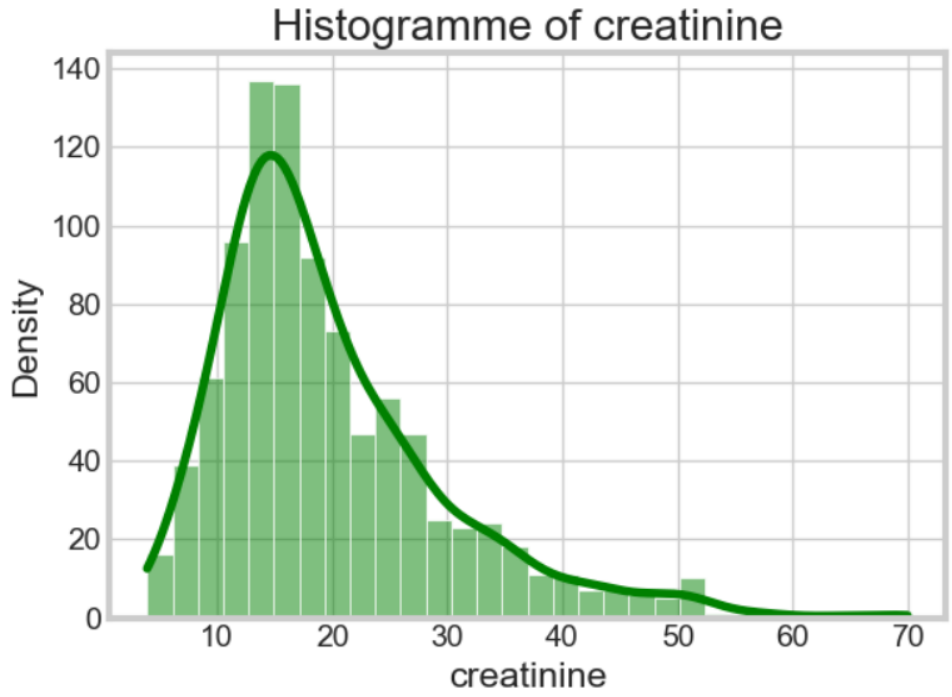


Figure 3.10: Distribution of categorical variables

Due to its essential role, creatinine is used to estimate GFR using equations, we chose the CKD-EPI equation since it is most commonly used by doctors, and we calculated this equation using two other variables which

are : age and sex, as shown in figure 3.11.

```
df['eGFR_CKD_EPI'] = df.apply(egfr_ckd_epi, axis=1)
# Affichage du DataFrame final
print(df[['creatinine', 'creatinine_missing', 'age', 'sex','eGFR_CKD_EPI']])
```

```
      creatinine  creatinine_missing   age   sex  eGFR_CKD_EPI
0          1.500                   0  78.0  Male     43.958752
1          1.500                   0  78.0  Male     43.958752
2          1.200                   0  79.0  Male     57.168765
3          1.400                   0  79.0  Male     47.448244
4          1.500                   0  79.0  Male     43.651041
```

Figure 3.11: eGFR CKD-EPI

And we used this numerical variable to create another target variable which is called "Stage-CKD", which allows us to determine each patient's stage of kidney failure as mentioned in figure 3.12.

```
df['Stage_CKD'] = df['eGFR_CKD_EPI'].apply(classer_stade_ckd)
df['Stage_CKD']
```

```
[23]: 0       Stage 3b
      1       Stage 3b
      2       Stage 3a
      3       Stage 3a
      4       Stage 3b
              ...
      1064    Stage 3a
      1065    Stage 3b
      1066     Stage 4
      1067    Stage 3b
      1068     Stage 4
      Name: Stage_CKD, Length: 1069, dtype: object
```

Figure 3.12: eGFR CKD-EPI

- **Categorical variables :**

  Figure 3.13 shows the distribution of the categorical variable "sex", the result says that most of our population are men, and the same procedure was done for the rest of variables, as shown in table 3.1 :

```python
for var in ['sex', 'diabete', 'HBP', 'anemia']:
    plt.figure(figsize=(6, 4))
    ax = sns.countplot(data=df, x=var, palette=palette_custom.get(var, 'pastel'))
    plt.title(f"Repartition of {var.capitalize()}", fontsize=14)

    # Affichage des valeurs au-dessus des barres
    for p in ax.patches:
        height = p.get_height()
        ax.annotate(f'{int(height)}', (p.get_x() + p.get_width() / 2, height),
                    ha='center', va='bottom', fontsize=11)

    plt.xlabel(var.capitalize())
    plt.ylabel("Effectif")
    plt.tight_layout()
    plt.show()
```

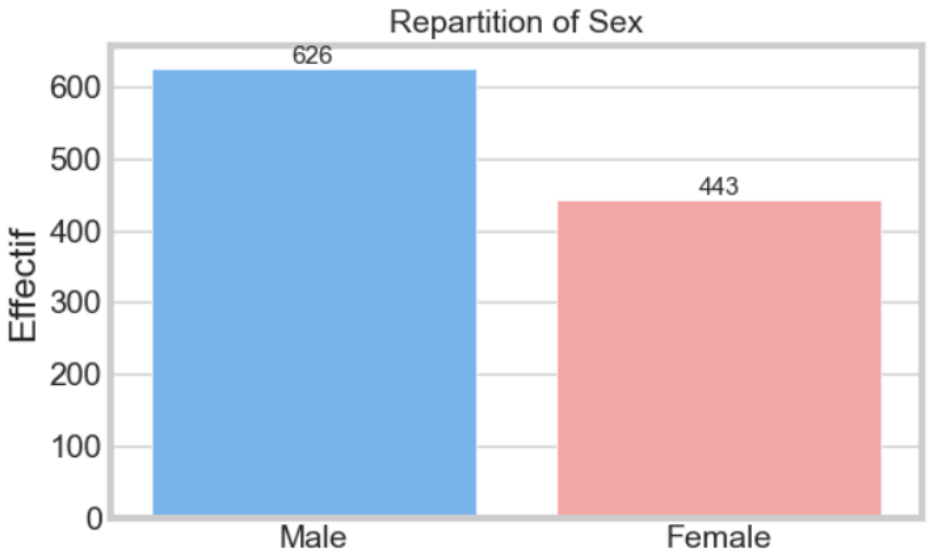


Figure 3.13: Distribution of categorical variables

Table 3.1: Distribution of categorical variables

Categorical Variable	**Distribution**
Sex	Male : 626, Female : 443
Diabetes Status	Diabetic : 423, Non-diabetic : 646
HBP	Yes : 797, No : 272
Anemia	Yes : 775, No : 294

Figure 3.14 shows that the majority of patients were classified as having stage 3b chronic kidney disease, indicating that most individuals are already in a moderate-to-advanced phase of CKD progression.

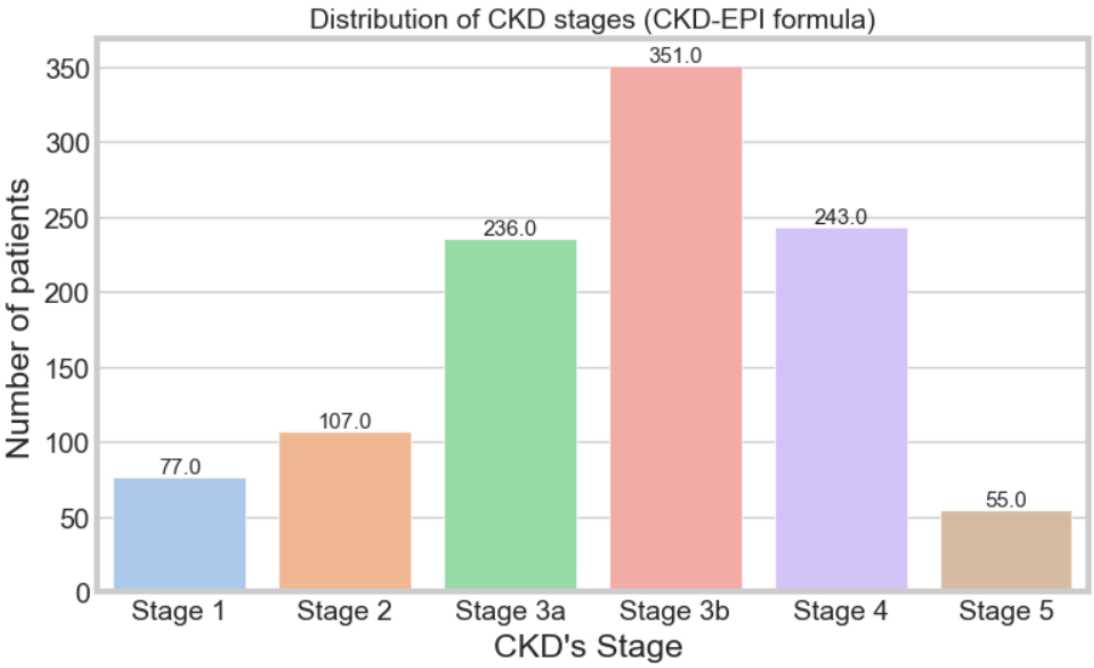


Figure 3.14: Distribution of CKD stages

3.4.2 Bi-variate descriptive analysis

This analysis was performed to investigate possible correlations between selected variables, particularly those that may influence the course of chronic kidney disease. The relationships between numerical and categorical variables were investigated using statistical tests such as the Chi-squared test, Pearson's test, Student's test and ANOVA, depending on the nature of the data.

- **Categorical vs Categorical :**

  In order to measure the relationship between sex and hypertension (BPH). Figure 3.15, indicates a statistically significant association between gender and HBP ($p = 0.004$).

Women seem to have a higher prevalence of hypertension (79.23%) than men (71.25%).

```
Tableau de contingence (Sex vs HBP) :

HBP      no  yes
sex
Female   92  351
Male    180  446

Résultat du test du Chi² :
Chi2 = 8.31, p-value = 0.0040
```

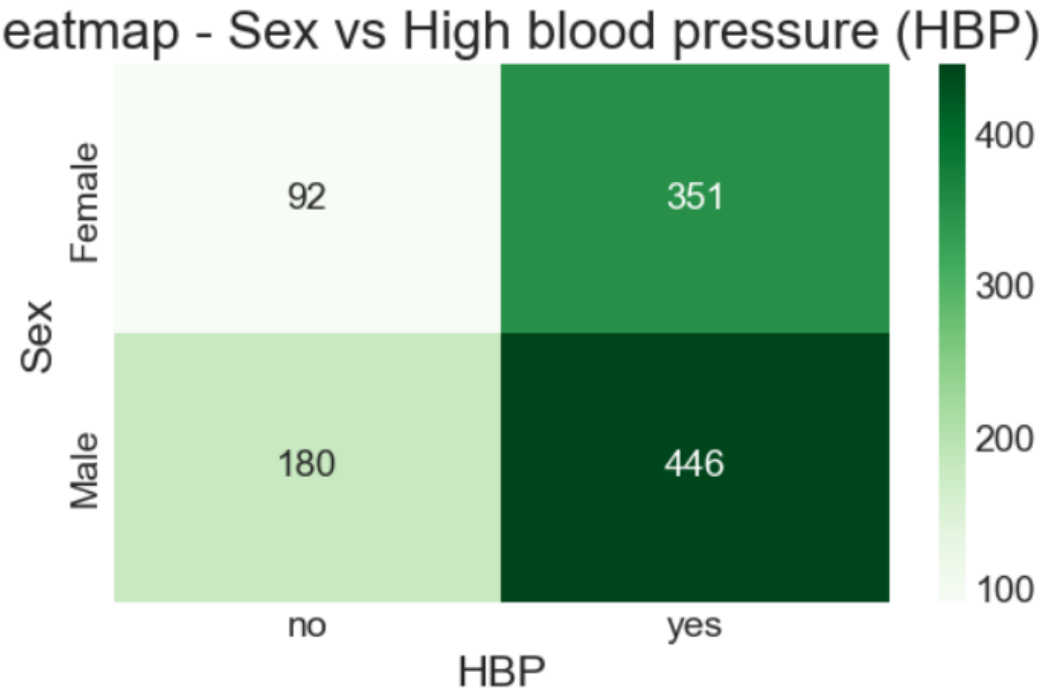


Figure 3.15: Relation between sex and HBP

And we did the same steps with other variables and we discovered three new significant associations which are :

- **Sex vs Stage-CKD** : more women are in the milder stages (Stage 1 and 2), while more men are in the more advanced stages (Stage 3b, 4, and 5) ($p < 0.0001$).
- **HBP vs Stage-CKD** : as CKD stages progress, the prevalence of hypertension increases markedly, particularly from stage 2 through to

stage 4 ($p < 0.0001$).

- **Diabetes vs Stage-CKD** : the proportion of diabetic patients clearly increases as the stages progress. Diabetic patients dominate non-diabetics after stage 3b ($p < 0.0001$).

- **Numerical vs Numerical :**

Figure 3.16 visualized the correlation between serum creatinine and eGFR

```python
from scipy.stats import pearsonr
x = df['creatinine']
y = df['eGFR_CKD_EPI']
# Test de corrélation de Pearson
corr, p_value = pearsonr(x, y)
print(f"Correlation of Pearson : r = {corr:.2f}, p-value = {p_value:.4f}")
# Visualisation
plt.figure(figsize=(6, 4))
sns.regplot(x=x, y=y, scatter_kws={'alpha':0.6}, line_kws={'color':'red'})
plt.title("Correlation between creatinine and GFR")
plt.xlabel("Creatinine")
plt.ylabel("eGFR")
plt.tight_layout()
plt.show()
```

```
Correlation of Pearson : r = -0.78, p-value = 0.0000
```

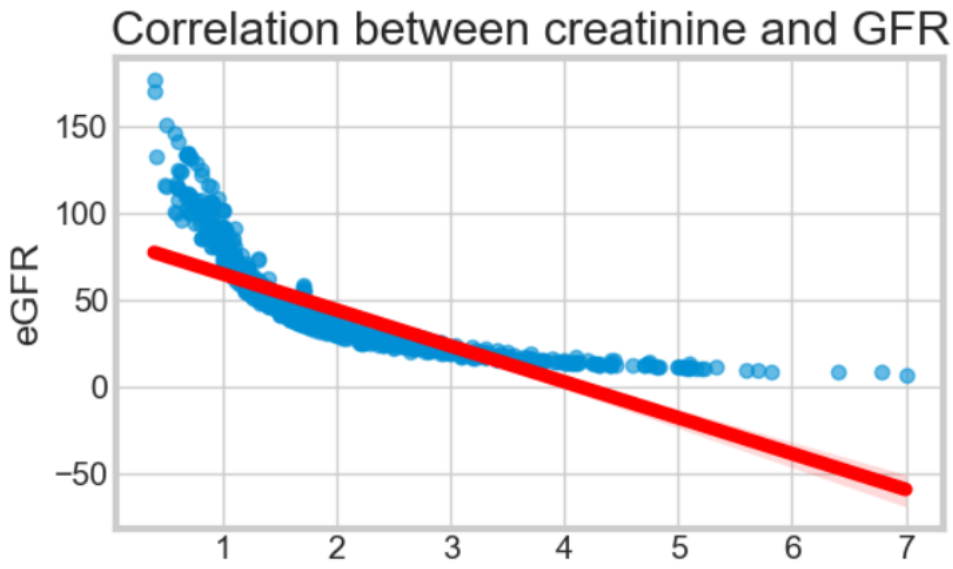


Figure 3.16: Correlation between serum creatinine and eGFR

which means that there is a negative correlation between them ($r = -0.78$), and also strong ($p < 0.0001$).

As creatinine levels increase, GFR decreases, which is totally logical from

a pathophysiological point of view.

And there are also some other significant correlations which are :

- **Age vs eGFR** : there is a significant and negative correlation ($r = -0.43$, $p < 0.0001$). As the patient's age increases, his CKD-EPI eGFR tends to decrease.
- **Urea vs eGFR** : High urea is strongly associated with reduced kidney function ($r = -0.63$, $p < 0.0001$).

- **Categorical vs Numerical :**

We can use Student's test to examine the correlation between a categorical and numerical variable if there are only two groups, and the ANOVA test if there are more than two groups.

- **Sex and eGFR** :

  Figure 3.17 shows that there is a moderate and positive correlation between them ($p < 0.0001$) .

  Figure 3.18 shows that mean renal function is higher in women, suggesting better mean renal function in this group.

```
# Violin plot comparing eGFR by sex
fig = px.violin(
    df,
    x="sex",
    y="eGFR_CKD_EPI",
    color="sex",
    box=True,
    template="plotly_dark",
    title="Distribution of eGFR (CKD-EPI) by Sex"
)
fig.update_layout(xaxis_title="Sex", yaxis_title="eGFR CKD-EPI (mL/min/1.73m²)")
fig.show()
```

```
T-test : t = 5.49, p-value = 0.0000
```

Figure 3.17: Student's test between sex and eGFR

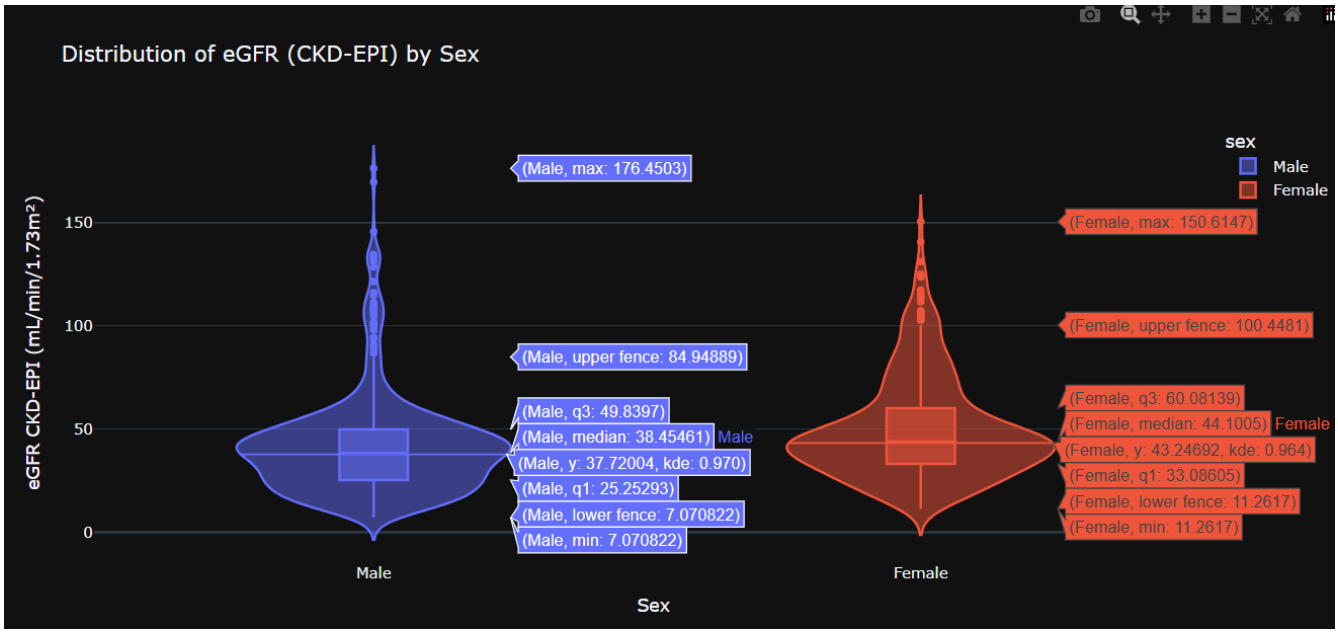


Figure 3.18: Violin's plot

- **Stage-CKD and eGFR** : There is a highly significant difference in eGFR according to CKD stages ($p < 0.0001$). This validates the relevance of CKD stages as a tool for stratifying renal function.

3.5 Predictive analysis

3.5.1 Prediction's goal

The goal of this study is to identify whether a patient is at risk of progressing from a moderate to a severe stage of the illness.

The topic is treated as a multiclass classification task, with three categories: normal, moderate, and severe.

3.5.2 Feature encoding

```python
# 💡 À adapter : liste exacte des noms des variables dans ton DataFrame
vars_binaires = ['HBP', 'diabete', 'anemia', 'sex']  # oui/non
vars_nominales = ['category_hb']
var_ordinale = 'Stage_CKD'

# 🔹 1. One-hot encoding appliqué à df directement
df = pd.get_dummies(df, columns=vars_binaires + vars_nominales, drop_first=True)

# Forcer tous les nouveaux booléens en entiers (0/1)
for col in df.columns:
    if df[col].dtype == 'bool':
        df[col] = df[col].astype(int)

# ✅ Vérification
print(df.dtypes)
print(df.head())
```

Figure 3.19: Encoding categorical variables

3.5.3 Target variable building

[69]:
```python
# Créer la cible : décalage temporel du stage regroupé
df['Stage_CKD_next'] = df.groupby('patientID')['Stage_CKD_grouped'].shift(-1)

# Supprimer les lignes sans cible (dernière ligne de chaque patient)
df = df.dropna(subset=['Stage_CKD_next'])
```

[70]:
```python
print(df['Stage_CKD_next'].value_counts())
```

```
Stage_CKD_next
Moderate    519
Advanced    259
Normal      167
Name: count, dtype: int64
```

Figure 3.20: Target variable

3.5.4 Data separation

We have longitudinal data and want to intelligently split the training and test sets, therefore **GroupShuffleSplit** is the best approach because it assures that all lines from the same group are either totally in the train set or entirely in the test set. This ensures a true separation between individuals as shown in figure 3.21.

Initial split

```
from sklearn.model_selection import GroupShuffleSplit, GroupKFold

# Split initial avec GroupShuffleSplit
gss = GroupShuffleSplit(n_splits=1, test_size=0.2, random_state=42)
train_idx, test_idx = next(gss.split(X, y, groups=groups))

# Séparation des données
X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]

# 👇 CORRECTION ICI : récupérer les groupes associés aux lignes d'entraînement
groups_train = df.loc[X_train.index, 'patientID']  # IDs patients alignés avec X_train
```

```
print("Shapes :")
print(f"- X_train : {X_train.shape}")
print(f"- y_train : {y_train.shape}")
print(f"- Groupes (patientID) : {len(groups.unique())} patients uniques")
```

```
Shapes :
- X_train : (714, 25)
- y_train : (714,)
- Groupes (patientID) : 114 patients uniques
```

Figure 3.21: Smart division by using GroupShuffleSplit

3.5.5 Encoding and normalization of numerical variables

We apply the encoding and normalization step after splitting the data into a **train** and **test** set thus to avoid data leakage, it occurs when information from the test set "leaks" into the training model, affecting its evaluation as demonstrated in figure 3.22.

The normalization step is very crucial, it prevents certain variables from dominating others and improves algorithm convergence, as shown in figure 3.23.

```
[77]: from sklearn.preprocessing import OrdinalEncoder, StandardScaler
encoder = OrdinalEncoder(categories=[['Normal', 'Moderate', 'Advanced']])
X_train['Stage_CKD_encoded'] = encoder.fit_transform(X_train[['Stage_CKD_grouped']])
X_test['Stage_CKD_encoded'] = encoder.transform(X_test[['Stage_CKD_grouped']])
```

Figure 3.22: Numerical encoding

```
[79]: # b. Normalisation
scaler = StandardScaler()
num_features = ['creatinine', 'Hb','weight', 'white.c', 'Platelets', 'clearance' , 'uric acid',
                'phosphorus',  'calcium',  'sodium',  'potassium',  'PTH', 'Urea',
                  'eGFR_CKD_EPI','Protein', 'albumin', 'age']  # À adapter selon le dataset
X_train[num_features] = scaler.fit_transform(X_train[num_features])
X_test[num_features] = scaler.transform(X_test[num_features])
```

Figure 3.23: Numerical normalization

3.5.6 Tested models

- **Training the model**

  The Random Forest algorithm's predictive performance in classifying chronic kidney disease progression was evaluated through comprehensive training and validation, ensuring class balance through *k*-fold cross-validation, preserving class balance.

  The model was optimized using a grid search (figure 3.24) to adjust key hyperparameters such as tree number, depth, split number, and feature consideration.

```
# 2. GridSearchCV (optimisation)
grid_search = GridSearchCV(model, param_grid, cv=GroupKFold(20), scoring='balanced_accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train, groups=groups_train)

print("✅ Meilleurs hyperparamètres :", grid_search.best_params_)
```

Figure 3.24: GridSearchCv

- **Testing the model**

  The model was tested using GridSearchCV and cross-validated to identify the best combination for performance on unseen data, then retrained on the full training set for accuracy, precision, recall, and F1-score (figure 3.25).

```
[82]: from sklearn.metrics import classification_report, accuracy_score, confusion_matrix

# 1. Prédiction sur le jeu de test
y_test_pred = best_model.predict(X_test)

# 2. Accuracy globale
acc_test = accuracy_score(y_test, y_test_pred)
print(f"\n✅ Accuracy sur le jeu de test : {acc_test:.3f}")

# 3. Rapport détaillé
print("\n📋 Rapport de classification sur le test :")
print(classification_report(y_test, y_test_pred, target_names=['Normal', 'Moderate', 'Advanced']))
```

```
✅ Accuracy sur le jeu de test : 0.732

📋 Rapport de classification sur le test :
              precision    recall  f1-score   support

      Normal       0.59      0.64      0.61        66
    Moderate       0.73      0.71      0.72       112
    Advanced       0.92      0.91      0.91        53

    accuracy                           0.73       231
   macro avg       0.75      0.75      0.75       231
weighted avg       0.74      0.73      0.73       231
```

Figure 3.25: Testing the model

As we can see in figure 3.25 the accuracy of the Random Forest algorithm is 73%, after using various hyper parameters and (GroupKFold) to avoid overfitting, we can say that Random Forest is trustworthy.

Table 3.2 contains the the evaluation of each algorithm, XGBoost and Logistic Regression performed the most evenly out of all the models, especially when

it came to identifying severe cases. All algorithms, however, struggle to accurately classify the 'Normal' category. The varying F1-scores across models suggest that the moderate stage is the most ambiguous class.

Table 3.2: Algorithm performance on the three CKD classes

Algorithm	**Class**	**Precision**	**Recall**	**F1-score**
Logistic Regression	Normal	0.74	0.44	0.55
	Moderate	0.70	0.88	0.77
	Severe	0.92	0.89	0.90
Decision Tree	Normal	0.49	0.76	0.59
	Moderate	0.64	0.42	0.51
	Severe	0.76	0.79	0.78
KNN	Normal	0.62	0.45	0.53
	Moderate	0.67	0.80	0.73
	Severe	0.92	0.85	0.88
SVM	Normal	0.68	0.41	0.51
	Moderate	0.69	0.85	0.76
	Severe	0.92	0.92	0.92
XGBoost	Normal	0.66	0.32	0.43
	Moderate	0.64	0.85	0.73
	Severe	0.88	0.85	0.87

Table 3.3: Accuracy of Each Algorithm on the Test Set

Algorithm	**Accuracy (Test Set)**
Logistic Regression	75%
Decision Tree	60%
Random Forest	73.2%
Support Vector Machine (SVM)	74%
XGBoost	70%
KNN	71.4%

As we can see in table 3.3, Logistic Regression tend to slightly outperform

SVM, demonstrating a higher capability for accurate CKD progression prediction, but there is another way to improve the results and it is by reducing the number of variables (working only with the most important features), Bah et al. [2] used that method in their study and they had better results. They pulled those characteristics from the Random Forest technique, which is exactly what we are doing (figure 3.26).

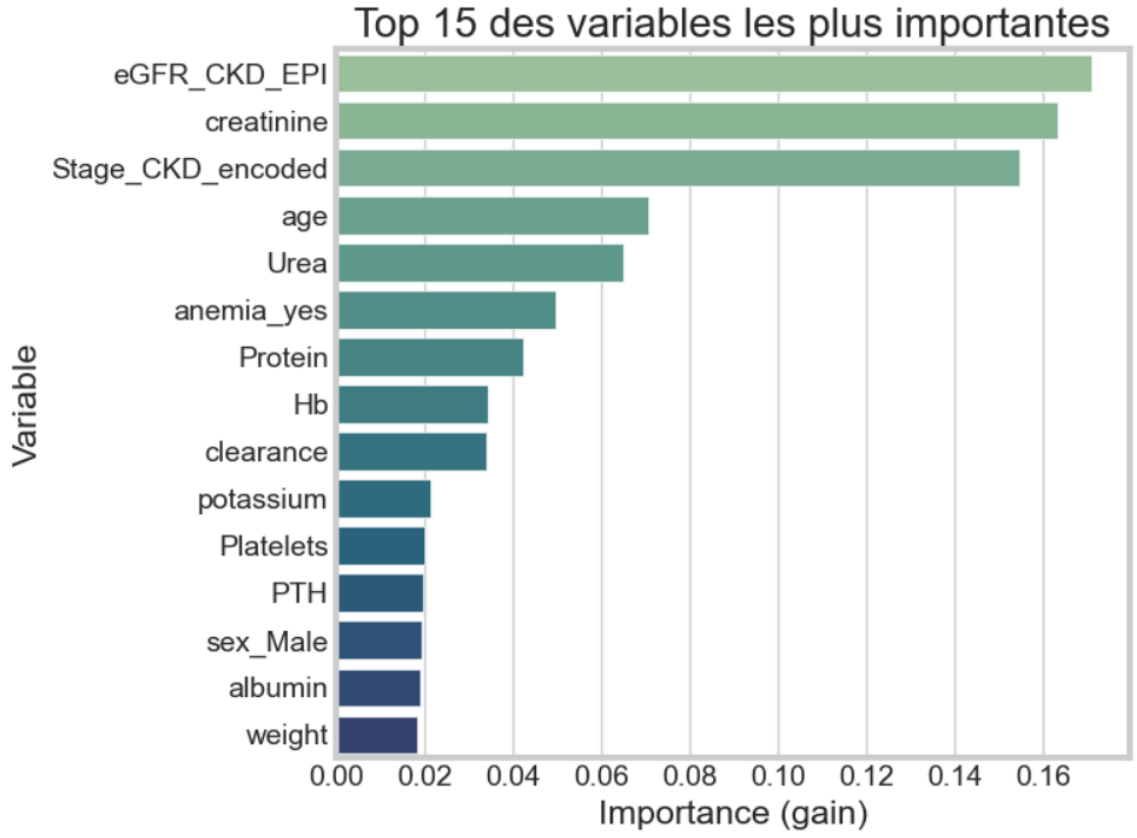


Figure 3.26: 15 most important features

Now we replace the variables with the top 15 features only, hence having a better result.

Figure 3.4 shows that Random Forest performs best overall, particularly when it comes to identifying severe situations (F1-score >88%) and it is more balanced in all three classes. Regardless of the algorithm, the normal class performs poorly, suggesting that it is difficult to distinguish between people without renal

problems.

Table 3.4: Simplified algorithm performance on the three CKD classes

Algorithm	**Class**	**Precision**	**Recall**	**F1-score**
Random Forest	Normal	0.62	0.59	0.60
	Moderate	0.73	0.75	0.74
	Severe	0.92	0.92	0.92
Logistic Regression	Normal	0.73	0.36	0.48
	Moderate	0.77	0.88	0.76
	Severe	0.92	0.89	0.90
Decision Tree	Normal	0.70	0.58	0.63
	Moderate	0.71	0.71	0.71
	Severe	0.73	0.89	0.80
KNN	Normal	0.64	0.42	0.51
	Moderate	0.68	0.82	0.74
	Severe	0.92	0.89	0.90
SVM	Normal	0.63	0.29	0.40
	Moderate	0.66	0.87	0.75
	Severe	0.92	0.92	0.92
XGBoost	Normal	0.70	0.35	0.46
	Moderate	0.66	0.88	0.75
	Severe	0.92	0.85	0.88

As we can see in figure 3.5, the Random Forest algorithm had a better result when we used only the most important features comparing to others.

Table 3.5: Accuracy of each simplified algorithm on the Test set

Algorithm	**Accuracy (Test Set)**
Logistic Regression	74%
Decision Tree	71.4%
Random Forest	74.5%
Support Vector Machine (SVM)	71.4%
XGBoost	72%
KNN	72.3%

3.6 Results and models comparison

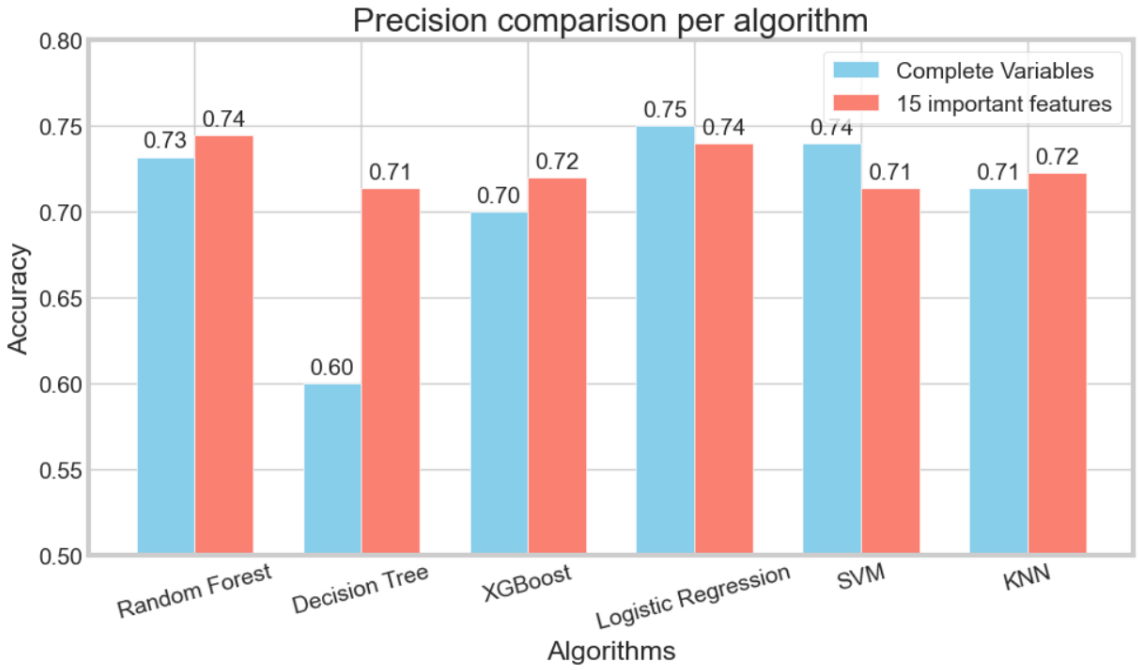


Figure 3.27: Precision comparison

Figure 3.27 indicates that most of the algorithms have a better precision after reducing the number of variables instead of using them all, The outcomes are interpreted as follows :

- Random forest, XGBoost and KNN have a slightly better precision.
- Decision Tree gain in precision with fewer variables.

- Logistic Regression and SVM remain robust even when simplified.

3.7 Discussion

The goal of this work is to predict the progression of chronic kidney disease by classifying it into three clinical stages: normal, moderate, and advanced, using a history of clinical factors. This method is compared to the "Prognostic Modeling of Chronic Kidney Disease Progression" study realized by Bah et al. [2], which focuses on a binary categorization (mild vs severe stages).
Despite the increased difficulty of the task, the proposed model achieves a precision of 74%, which is quite commendable in a multi-class setting.

3.7.1 Performance comparison

Table 3.6 indicates that the performance level reached is lower than that of the reference paper, but that article is based on a simpler challenge (binary prediction), while our model must identify three unique clinical levels, which complicates the problem.

Table 3.6: Performance comparison between the article and our study

Model	**Article (binary)**	**Current project (3 classes)**
Random Forest	85 %	74 %
Logistic Regression	84 %	75 %

In addition, the improved performance of the article can be attributed to the quality and amount of the dataset, which helps learning. Due to the lack of data and the huge number of missing values, especially in the most important features that are considered as key biological parameters of CKD progression, such as **ACR** (97%), we didn't have another choice but to delete it, because the result didn't improve, there is also another important parameter which is **24h**

protein, we had to apply an advanced method of simulation to fill the missing values to avoid biased results.
However, the work done here has allowed us to construct a strong model despite the more restrictive requirements.

Furthermore, it is clinically inappropriate to classify "normal" patients as "moderate" under the guise of binary simplicity. Indeed, while some clinical indicators may overlap between these two groups, the treatment implications and risks of evolution are markedly different. Despite having a lower precision than the binary article, we chose to stay straight and avoid mingling between two different classes. Maintaining a three-level classification allows for a more refined analysis that is more suited to clinical reality.

3.7.2 Limitations and Future Research Recommendations

Why does the “Normal” class often have the worst scores (precision, recall, F1) ?

This typically occurs from a combination of two key reasons. First, the distribution of clinical transitions is imbalanced, even if our patients are classed as "Normal," many of them immediately progress to "Moderate" as soon as a minor variation occurs (modest drop in GFR or a slight rise in serum creatinine) which means that the model identifies relatively little stability in the Normal stage. Second, we have clinical ambiguities, even if the number of "Normal" instances appears considerable, this group is clinically quite heterogeneous, some are nearly moderate and other are very stable. That’s why it frequently confuses "slightly normal" with "moderate" situations.
Using a larger dataset and additional techniques, such as neural networks or

deep learning algorithms, could further increase the accuracy and robustness of the models.

Conclusion

In this chapter we have presented the multiple pre-processing steps, including data exploration, visualization, and cleaning up missing values, without forgetting the statistical analysis including univariate and bivariate ones. Using evaluation methods, we found that Random Forest was the most accurate model in the simplified dataset, and we had better precisions, comparing to the entire dataset, and we ended by a discussion.

General conclusion

This study of CKD progression identified several characteristics related with worsening stages, including male gender, advanced age, higher creatinine, and lower GFR. These correlational clinical data confirm the relevance of the selected variables and the notion that particular patient profiles have a higher probability of progressing to severe stages.

In the purpose of realizing this prediction, we have employed six machine learning algorithms : Decision Trees, Random Forest, Logistic Regression, K-Nearest-Neighbors, XGBoost, and Support Vector Machine. The findings reveal that Random Forest tend to slightly outperform the others in terms of accuracy.

Despite the problem's complexity, we achieved a maximum score of 74% accuracy using a rigorously approved model. This result is clinical, stable, and trustworthy, even if it falls below some scores reported in the literature, which are frequently acquired on simpler tasks (binary classification). The work done has allowed us to better understand the elements that influence CKD progression and propose a solid foundation for future advancements.

So we accomplished our goal of applying intelligent models to real-world data and simulating patient progression in order to better adjust the treatment of patients.

Bibliography

[1] Chloé-Agathe Azencott. *Introduction au Machine Learning*. Cours de l'ENS Paris-Saclay, 2018.

[2] Bah K, Jallow AW, Bah AN. Prognostic Modeling of Chronic Kidney Disease Progression: Bridging Mild and Severe Stages through a Machine Learning Approach. Austin J Clin Case Rep. 2023;10(9):1310

[3] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. *A Comparative Analysis of XGBoost*. arXiv preprint arXiv:1911.01914, 2019.

[4] Bhavsar, H., & Ganatra, A. (2012). A Comparative Study of Training Algorithms for Supervised Machine Learning. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(4), 2231–2307.

[5] Boucquemont J. Modèles statistiques pour l'étude de la progression de la maladie rénale chronique [thèse de doctorat]. Bordeaux : Université de Bordeaux; 2014

[6] Brazdil, P., van Rijn, J. N., Soares, C., & Vanschoren, J. *Metalearning for Hyperparameter Optimization*. In *Metalearning*, Cognitive Technologies, pages 103–122. Springer, Cham, 2022.

[7] Brocheriou C., Fakhouri F. Comment j'explore… une protéinurie. Revue du Praticien, vol. 69, n° 5, 2019, pp. 524–527.

[8] Combes, S. Chaire Santé Sciences Po. *Enjeux de l'intelligence artificielle en santé*. Mai 2023.

[9] Deo RC. Machine learning in medicine. Circulation. 2015 Nov 17;132(20):1920–1930. doi:10.1161/CIRCULATIONAHA.115.001593.

[10] Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Banyatsang, M., & Tabona, O. *A survey on missing data in machine learning*. Procedia Computer Science, 180 (2021), 1177–1184.

[11] Fellahi, Z. A. *Les statistiques inférentielles*. Université de Bordj Bou Arreridj, Faculté des Sciences de la Nature et de la Vie, 2018.

[12] Franceschi, L., Donini, M., Perrone, V., Klein, A., Archambeau, C., Seeger, M., Pontil, M., & Frasconi, P. *Hyperparameter Optimization in Machine Learning*. arXiv preprint arXiv:2410.22854v2, April 2025.

[13] Houde, L. *Module 12 : Tests du Khi-deux*. Université du Québec à Trois-Rivières, Département de mathématiques et d'informatique, 2018.

[14] Kaur KK, Allahbadia G, Singh M. An update on the approaches of avoidance of propagation of chronic kidney disease resulting in reversal or possible need or avoidance of kidney transplantation: A systematic review. J Clin Nephrol. 2022;6:040–057. doi:10.29328/journal.jcn.1001089

[15] Kidney Disease: Improving Global Outcomes (KDIGO) Diabetes Work Group. KDIGO 2022 Clinical Practice Guideline for Diabetes Management in Chronic Kidney Disease. Kidney Int. 2022;102(5S):S1–S127. doi:10.1016/j.kint.2022.06.008

[16] Mayou NasserEddine, Belhachani Mohammed. *Une application web pour la prédiction précoce du diabète basant sur les algorithmes d'apprentissage*

automatique. Mémoire de Master en Informatique, Spécialité Informatique Industrielle, encadré par Dr Abdelmadjid Youcefa, Université Kasdi Merbah Ouargla, Algérie, 2023.

[17] Mindrila, D., & Balentyne, P. *One-Way ANOVA Lecture Notes*. University of West Georgia. Based on Chapter 25 of "The Basic Practice of Statistics" (6th ed.), 2013.

[18] Palacio-Lacambra ME, Comas-Reixach I, Blanco-Grau A, Suñé-Negre JM, Segarra-Medrano A, Montoro-Ronsano JB. Comparison of the Cockcroft–Gault, MDRD and CKD-EPI equations for estimating ganciclovir clearance. Br J Clin Pharmacol. 2018;84(9):2120–2128. doi:10.1111/bcp.13647

[19] Ricco Rakotomalala. *Régression logistique (approche machine learning)*. Université Lumière Lyon 2, ERIC Lab, 2018.

[20] Salmon, J., & Verzelen, N. *Validation Croisée*. Université de Montpellier / INRAE, 2018.

[21] Shaveta. A review on machine learning. *International Journal of Science and Research Archive*, 2023, 09(01), 281–285.

[22] Swartling O, Rydell H, Stendahl M, Segelmark M, Trolle Lagerros Y, Evans M. CKD progression and mortality among men and women: A nationwide study in Sweden. Am J Kidney Dis. 2021;78(2):190–199. doi:10.1053/j.ajkd.2020.11.026

[23] Tangri N, Stevens LA, Griffith J, et al. "A Predictive Model for Progression of Chronic Kidney Disease to Kidney Failure." JAMA. 2011;305(15):1553-1559. doi:10.1001/jama.2011.451

[24] Ünalan, S., Günay, O., Akkurt, I., Gunoglu, K., & Tekin, H. O. *A comparative study on breast cancer classification with stratified shuffle split and K-fold cross validation via ensembled machine learning*. Journal of Radiation Research and Applied Sciences, 17(4), 101080, 2024. 10.1016/j.jrras.2024.101080

[25] Vickers, P., Barrault, L., Monti, E., & Aletras, N. *We Need to Talk About Classification Evaluation Metrics in NLP*. In Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 498–510, Nusa Dua, Bali, November 2023. Association for Computational Linguistics.

Résumé

Au cœur de ce mémoire, on a conçu et développé un modèle fiable et assez robuste pour un problème complexe de trois classes majoritaires pour la prédiction de la maladie rénale chronique (MRC), afin d'éviter les risques de complications de cette maladie sur la santé du patient. Pour atteindre cet objectif, on a utilisé des algorithmes d'apprentissage automatique supervisé (K-Nearest-Neighbors, Decision Trees, Random Forest, Support Vector Machine, XGBoost, Logistic Regression) et le data set extrait à partir des dossiers médicaux du service de consultation en nephrologie situé à Bejaia. Les performances des classifieurs ont été comparées en fonction du taux de précision. Le plus haut taux de classification obtenu par l'application de Random Forest est de 74% en appliquant la méthode d'évaluation train/test .

Mots clés: IA , ML , CKD, Decision Trees, Random Forest, Support Vector Machine.

Abstract

In this work, we created an accurate and robust model for predicting chronic kidney disease (CKD) in a difficult situation with three majority classes. This helps patients avoid complications and improves their health. To achieve this objective, we employed efficient machine learning methods (K-Nearest-Neighbors, Decision Trees, Random Forest, Support Vector Machine, XGBoost, Logistic Regression) and retrieved data from medical records of the nephrology consulting service in Bejaia. The classifier's performance was compared based on accuracy rate. The Random Forest application achieved the greatest classification rate of 74% using the train/test evaluation approach.

Keywords: AI, Machine Learning, CKD, Decision Trees, Random Forest, Support Vector Machine.