2022-2023

Machine Learning Project's Report

Honoris United Universities

Module :

Machine Learning

Elaborated by :

Mohamed Amine Ben Elazrak

Yassine Ben Cheikh Brahim

Wejden Rejeb

Khalil Trabelsi

Louay Guetat

Riadh Rabti

Contents

General Introduction

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract or extrapolate knowledge and insights from noisy, structured and unstructured data and apply knowledge from data across a broad range of application domains.

The primary aim of health-related AI applications is to analyze relationships between clinical techniques and patient outcomes. AI programs are applied to practices such as diagnostics, treatment protocol development, drug development, personalized medicine, and patient monitoring and care.

Currently, the most common roles for AI in medical settings are clinical decision support and imaging analysis. Clinical decision support tools help providers make decisions about treatments, medications, mental health and other patient needs by providing them with quick access to information or research that's relevant to their patient.

In this context, we are going to conduct our research into the use of AI in the diagnosis of certain illnesses.

This type of research must be done according to a specific methodology in our case it is called CRISP DM.

Project theme

Our project is focused on the detection of possible signs that may indicate the existence of chronic kidney disease(CKD).The end goal is to help treat the disease through early on detection and treatment .As for all applications of AI in the medical the high accuracy and reliability on the final model is of utmost importance .

We are going to simulate two studies that tackled this subject, synthesizing the best techniques used and adding ones of our own in order to have the best possible outcome .

The dataset that we will bec working with contains 25 features that describe general aspects of the patients health e.g blood pressure,age,albumin etc .

The end results quality will be evaluated using performance tools with the end goal of determining whether or not a patient has CKD .

Chronic Kidney Disease (CKD)

Chronic kidney disease is a very serious disease featuring in the top 10 diseases in terms of mortality. It is a disease that is unfortunately frequent and that targets the kidneys and the urinary track in general.

About 37 million US adults are estimated to have CKD, and most are undiagnosed.

CKD is a condition in which the kidneys are damaged and cannot filter blood as well as they should. Because of this, excess fluid and waste from blood remain in the body and may cause other health problems, such as heart disease and stroke.

CKD is a condition in which the kidneys are damaged and cannot filter blood as well as they should. Because of this, excess fluid and waste from blood remain in the body and may cause other health problems, such as heart disease and stroke.

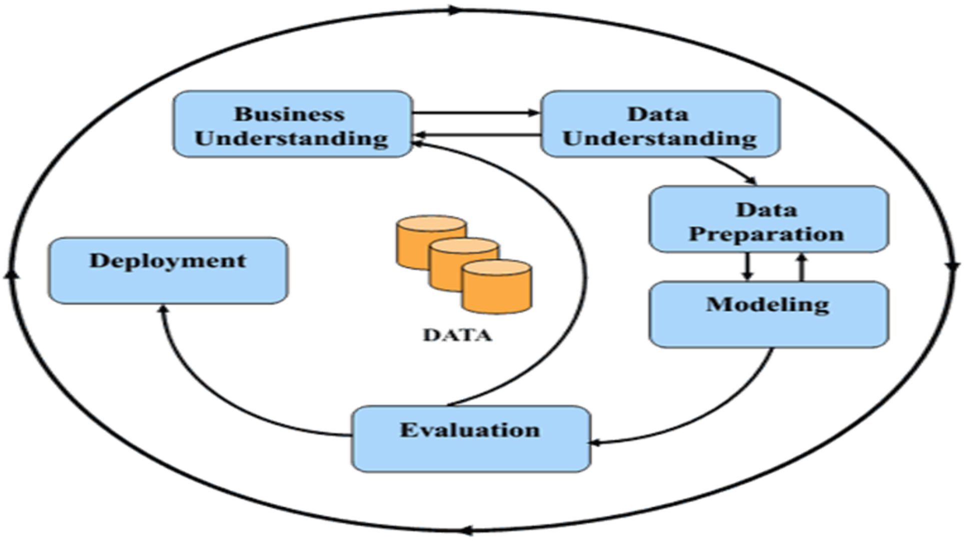
CRISP DM Methodology

The CRoss Industry Standard Process for Data Mining (*CRISP-DM*) is a process model that serves as the base for a data science process. It has six sequential phases:

1. Business understanding – What does the business need?
2. Data understanding – What data do we have / need? Is it clean?
3. Data preparation – How do we organize the data for modeling?
4. Modeling – What modeling techniques should we apply?
5. Evaluation – Which model best meets the business objectives?
6. Deployment – How do stakeholders access the results?

Published in 1999 to standardize data mining processes across industries, it has since become the most common methodology for data mining, analytics, and data science projects.

Data science teams that combine a loose implementation of CRISP-DM with overarching team-based agile project management approaches will likely see the best results.



Business understanding

This phase consists of a very precise specification of the problem together with methods of evaluating the achievement of the goal .While jumping on to data is not bad all time but in most cases we end up with insights that do not get integrated with real-world instances. There are multiple pitfalls of not having an understanding of business and underneath data.

CKD has varying levels of seriousness. It usually gets worse over time though treatment has been shown to slow progression. If left untreated, CKD can progress to kidney failure and early cardiovascular disease. When the kidneys stop working, dialysis or kidney transplant is needed for survival. Kidney failure treated with dialysis or kidney transplant is called end-stage renal disease.

Not all patients with kidney disease progress to kidney failure. To help prevent CKD and lower the risk for kidney failure, control risk factors for CKD, get tested yearly, make lifestyle changes, take medicine as needed, and see your health care team regularly.

Talk to your doctor about getting tested if you have any of these risk factors:

- Diabetes

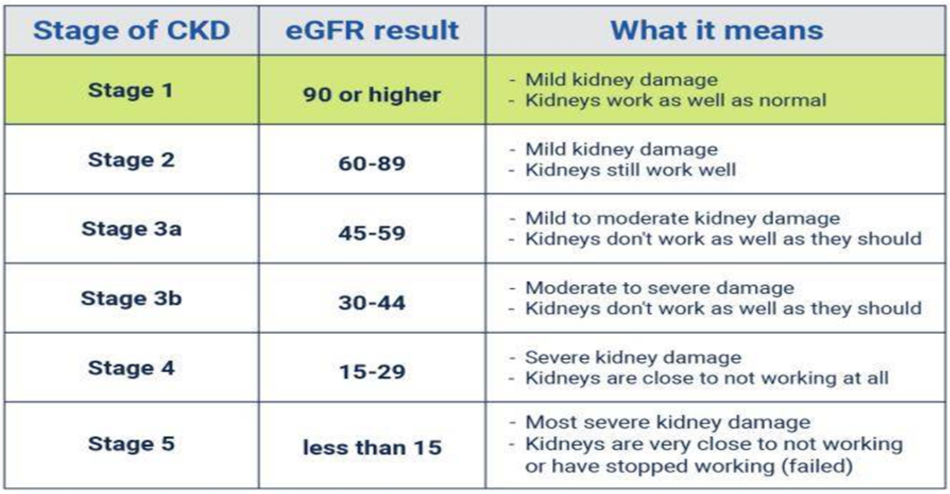
- High blood pressure
- Heart disease
- Family history of CKD
- Obesity

Some other health consequences of CKD include:

- Anemia or low number of red blood cells
- Increased occurrence of infections
- Low calcium levels, high potassium levels, and high phosphorus levels in the blood
- Loss of appetite or eating less
- Depression or lower quality of life

Different stages of CKD

RThe evolution of CKD can be determined through the following figure



Stage of CKD	eGFR result	What it means
Stage 1	90 or higher	- Mild kidney damage - Kidneys work as well as normal
Stage 2	60-89	- Mild kidney damage - Kidneys still work well
Stage 3a	45-59	- Mild to moderate kidney damage - Kidneys don't work as well as they should
Stage 3b	30-44	- Moderate to severe damage - Kidneys don't work as well as they should
Stage 4	15-29	- Severe kidney damage - Kidneys are close to not working at all
Stage 5	less than 15	- Most severe kidney damage - Kidneys are very close to not working or have stopped working (failed)

Certain key elements were used as guidance in the upcoming steps of our experiment such as the risk factors and symptoms of CKD many of which were present as features in our dataset this helped to shed more light and properly deal with these insights.

Data understanding

In this study, we will evaluate a dataset collected from 400 patients from the University of California, Irvine Machine Learning Repository. The dataset consists of 24 features divided into 11 numerical features and 13 categorical features, and our target feature is ckd

Categorical features

Red Blood Cells, Cell Pussy, Puss Cell Clumps, Bacteria, White Blood Cell Count, Red Blood Cell Count, Hypertension, Diabetes Mellitus, Coronary Artery Disease, Appetite, Pedal Edema, Anemia,ckd

Numerical features

Age, Blood Pressure, Specific Gravity, Packed Cell Volume, albumin, Sugar, Blood Glucose Random, Blood Urea, Serum Creatinine, Sodium, Potassium, Hemoglobin And we searched for the intervals of each numerical value these intervals will be detailed in the treatment of outliers

Most features contain missing values and outliers. The dataset is unbalanced because it contains 250 boxes of class **"ckd"** by **62.5%** and 150 boxes of **"notckd"** by **37.5%**.

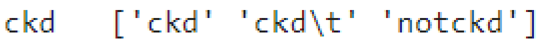
Data preprocessing

Data preprocessing is the process of preparing the dataset to ensure the best results possible for the modeling phase.

Anomaly detection and treatment :

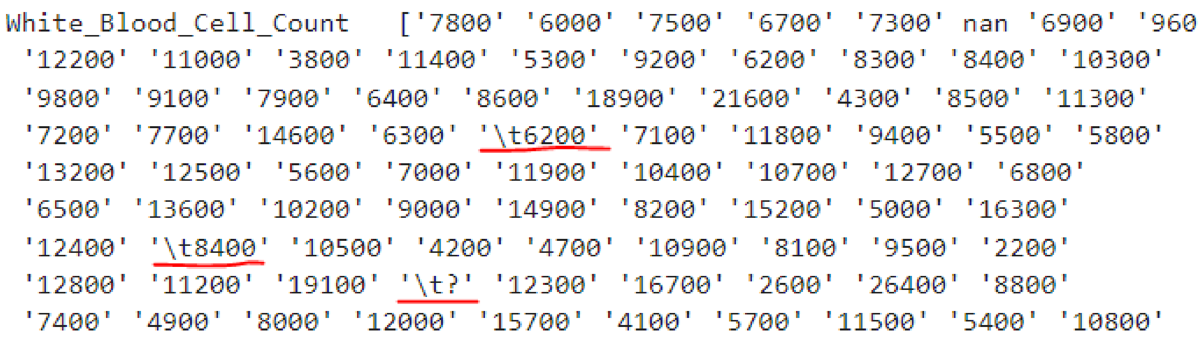
During the phase of exploratory data analysis, we came across different anomalies that needed to be treated beforehand .
The types of anomalies encountered were :

- **Typing errors :** Categorical and numerical features were found to have typing errors e.g 'CKD' was found to have 3 types of values instead of 2 with the extra being '\tckd'.



```
ckd    ['ckd' 'ckd\t' 'notckd']
```

- <u>Wrong type of data attribution</u> where for instance the feature **'white_blood_cell_count'** was given the type object where in fact it was numerical as well as the previously mentioned error,this was treated with a simple **to.numeric()**

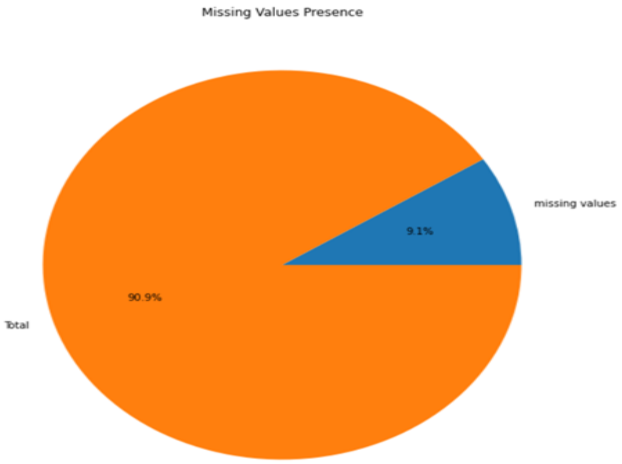


```
White_Blood_Cell_Count   ['7800' '6000' '7500' '6700' '7300' nan '6900' '960
 '12200' '11000' '3800' '11400' '5300' '9200' '6200' '8300' '8400' '10300'
 '9800' '9100' '7900' '6400' '8600' '18900' '21600' '4300' '8500' '11300'
 '7200' '7700' '14600' '6300' '\t6200' '7100' '11800' '9400' '5500' '5800'
 '13200' '12500' '5600' '7000' '11900' '10400' '10700' '12700' '6800'
 '6500' '13600' '10200' '9000' '14900' '8200' '15200' '5000' '16300'
 '12400' '\t8400' '10500' '4200' '4700' '10900' '8100' '9500' '2200'
 '12800' '11200' '19100' '\t?' '12300' '16700' '2600' '26400' '8800'
 '7400' '4900' '8000' '12000' '15700' '4100' '5700' '11500' '5400' '10800'
```

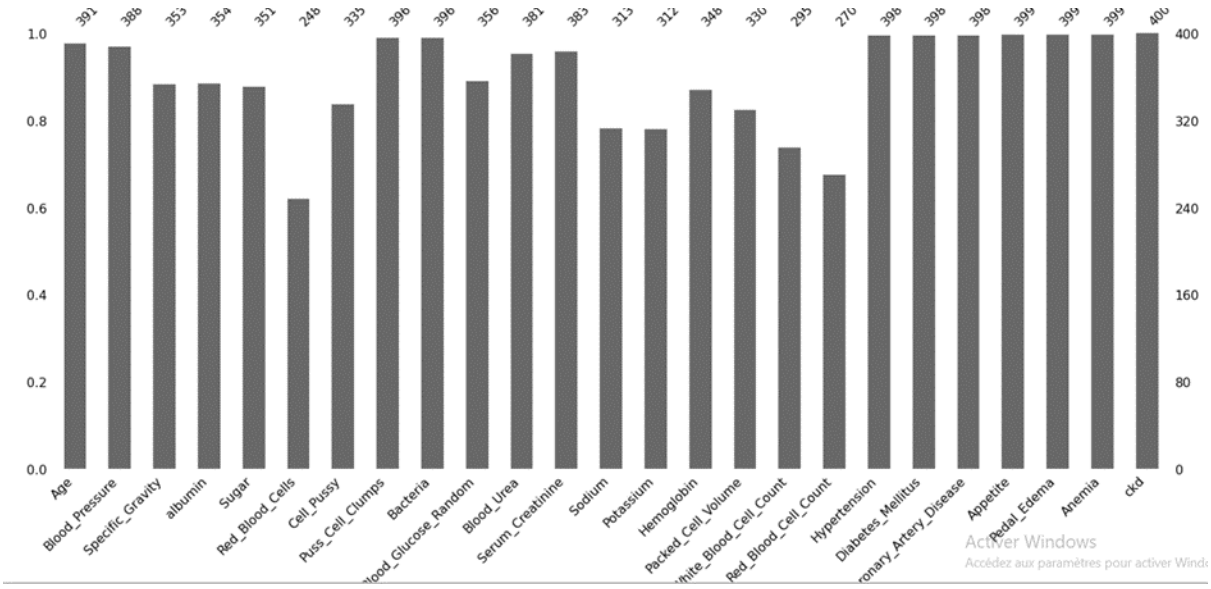
Encoding categorical features

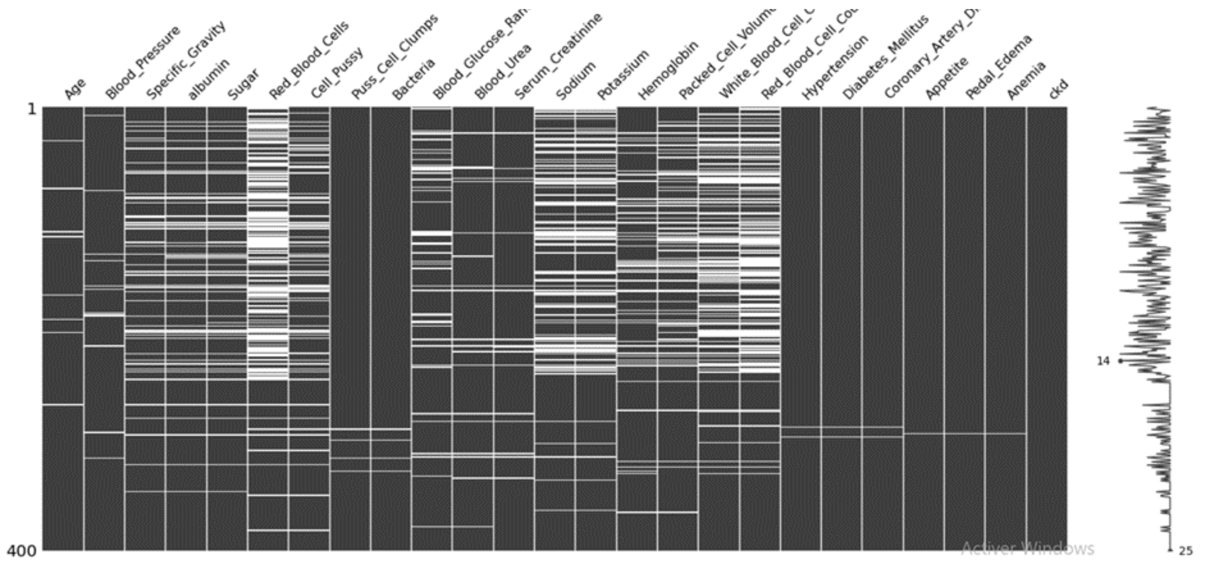
Since the machine learning model works entirely on mathematics and numbers, we must encode these categorical variables in numbers. We use Label encoding since our features are nominal.

Dealing with missing values



Since we are dealing with a small dataset we can't just delete rows with missing values, that's why we imputed the numeric values with means and the categorical values wit modes.

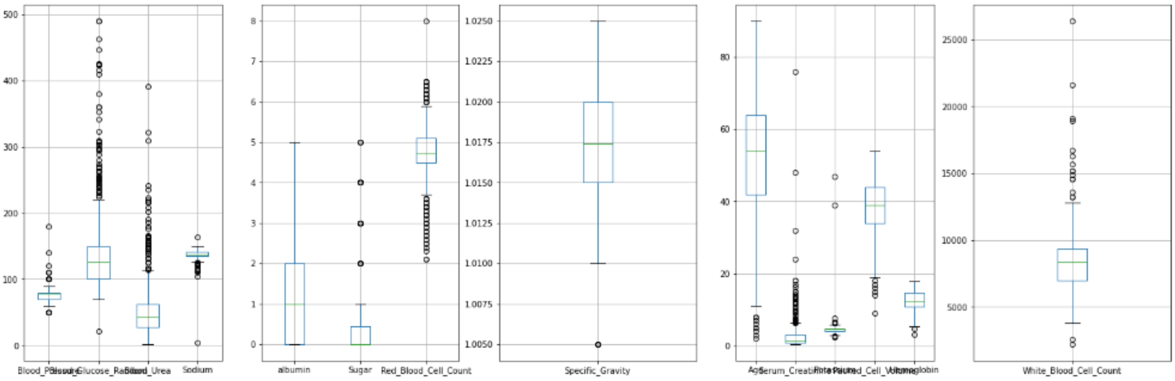




Dealing with outliers

Detecting the outliers :

We used boxplots to detect the presence of outliers in our dataset as shown :

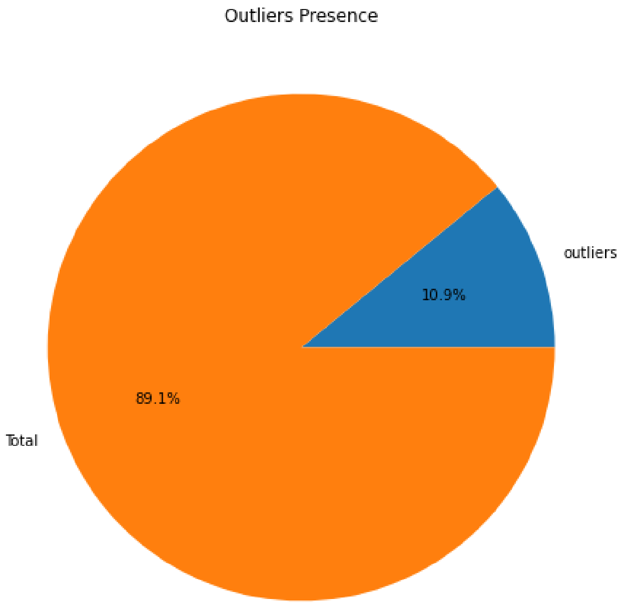


Since we are working in a medical field where precision is of most importance,we spent a lot of time looking for the exact range with a correct scale for each numeric value in order to detect outliers. the results are portrayed below:

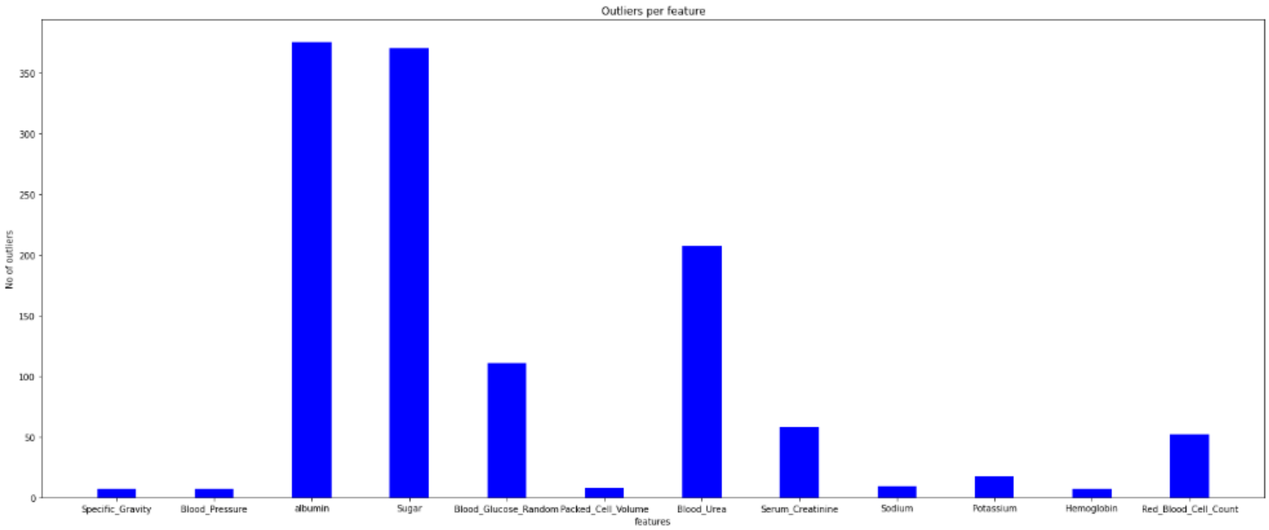
Feature	Lower value	Higher value
Blood Pressure	60	120
albumin	3,4	5
Sugar	2,8	6,2
Blood_Glucose_Random	100	300
Blood Urea	5	43
Serum Creatinine	0,3	5

Sodium	**120**	**150**
Potassium	**3**	**6**
Packed_Cell_Volume	**20**	**60**
Hemoglobin	**5**	**17,5**

we created two different outlier detection functions depending on the nature of the feature , the total number of outliers detected was 1228 which represents 10.9% :



This outlier distribution per feature :



Imputing the outliers

Missing data imputation :

In data analytics, missing data is a factor that degrades performance. Incorrect imputation of missing values could lead to a wrong prediction. In this era of big data, when a massive volume of data is generated every second, and utilization of this data is a major concern to the stakeholders, efficiently handling missing values becomes more important.

- Imputation using KNN :

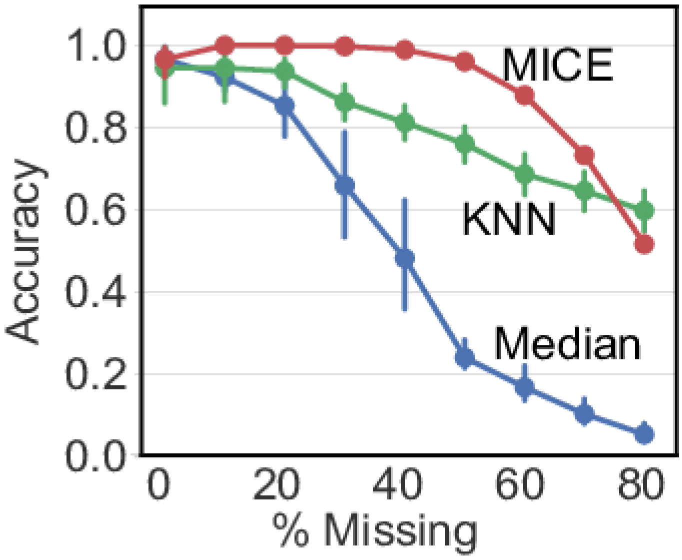
  kNN imputation is **a lazy and instance-based estimation method**. Different from model-based algorithms (building estimators from all complete instances and then filling in a missing datum with the estimators), kNN needs to search all complete instances and select k instances most relevant to a given missing data

- Imputation using MICE :

  Multiple Imputation by Chained Equations is a robust, informative method of dealing with missing data in datasets. The procedure ‘fills in’ (imputes) missing data in a dataset through an iterative series of predictive models. In each iteration, each specified variable in the dataset is imputed using the other variables in the dataset. These iterations should be run until it appears that convergence has been met.

Conclusion :

Results show that the multiple imputations by using chained equations (MICE) outperformed the other imputation methods. **The mean and k nearest neighbor (KNN) performed better relative to sample and median imputation methods**.



Scaling

Having features varying in scale and range could be an issue when the model we are trying to build uses distance measures such as Euclidean Distance. Such models could be K-Nearest Neighbours, SVM .
This is an essential part because we have different value scales and we will be implementing machine learning algorithms that are affected by these differences .

Modeling :1st Article Boosted Classifier and Features Selection

Introduction

In this chapter, we will be treating the first article named “Article Boosted Classifier and Features Selection”. In this article, we will treat our data with 3 different Methods to compare our results and choose the best method for our modeling process. First of all, we will use our models on our data pre-feature selection and before any data treatment. For the second part, we will use a feature selection for our data and treat the new data with our models so we can compare the new results and the improvement of our model. Last part is the application of AdaBoost using our different models and we will be cleaning outliers of the new data after the feature selection case by case and then comparing the results to the 2 different methods used before.

Feature Selection

Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data.

It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve. We do this by including or excluding important features without changing them. It helps in cutting down the noise in our data and reducing the size of our input data.

The correlation-based feature selection (CFS)

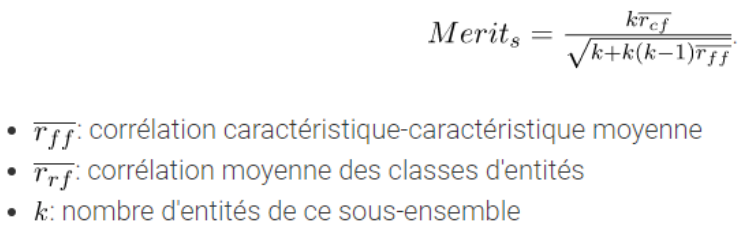
Including feature selection methods as a preprocessing step in predictive modeling comes with several advantages. It can reduce model complexity, enhance learning efficiency, and can even increase predictive power by reducing noise.

General Principle

The *correlation-based feature selection* (CFS) method is a filter approach and therefore independent of the final classification model. It evaluates feature subsets only based on data intrinsic properties, as the name already suggest: correlations.

The goal is to find a feature subset with *low feature-feature correlation*, to avoid redundancy, and high *feature-class correlation* to maintain or increase predictive power.

For that, the algorithm estimates the merit of a subset **s** with **k** features with the following equation:



$$Merit_s = \frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}}.$$

- $\overline{r_{ff}}$: corrélation caractéristique-caractéristique moyenne
- $\overline{r_{rf}}$: corrélation moyenne des classes d'entités
- k: nombre d'entités de ce sous-ensemble

Hall [1] proposes a best first search approach using the merit as heuristic. The search starts with an empty subset and evaluates for each feature the merit of being added to the empty set. For this step the feature-feature correlation can be neglected, as the denominator of the equation above simplifies to 1, due to **k=1**.

$$\frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}} = \frac{1\overline{r_{cf}}}{\sqrt{1+1(1-1)\overline{r_{ff}}}} = \frac{\overline{r_{cf}}}{\sqrt{1}} = \overline{r_{cf}}$$

So for the first iteration the evaluation is solely based on the feature-class correlation. The feature with the highest feature-class correlation is added to the so far empty subset. In the next step again all features, except for the one already added, are evaluated and the one that forms the best subset with the previously added one is kept.

This process is iterative and whenever an expansion of features yields no improvement, the algorithm drops back to the next best unexpanded subset. Without a limitation this algorithm searches the whole feature subset space. Hence, the number of backtracks must be limited. After reaching this limit the algorithm returns the feature subset that yielded the highest merit up to this point.

[1] Hall, M. A. (2000). Correlation-based feature selection of discrete and numeric class machine learning.

Models

KNN

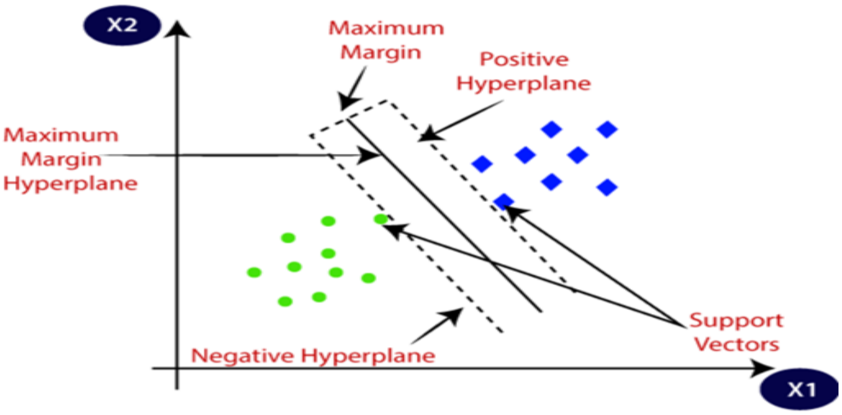
The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

SVM

Support Vector Machine or SVM is a Supervised Learning algorithm, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence algorithm is termed as Support Vector Machine.



Naive Bayes

Naive Bayes is a classification approach that adopts the principle of class conditional independence from the Bayes Theorem. This means that the presence of one feature does not impact the presence of another in the probability of a given outcome, and each predictor has an equal effect on that result. There are three types of Naïve Bayes classifiers: Multinomial Naïve Bayes, Bernoulli Naïve Bayes, and Gaussian Naïve Bayes. This technique is primarily used in text classification, spam identification, and recommendation systems.

Logistic regression

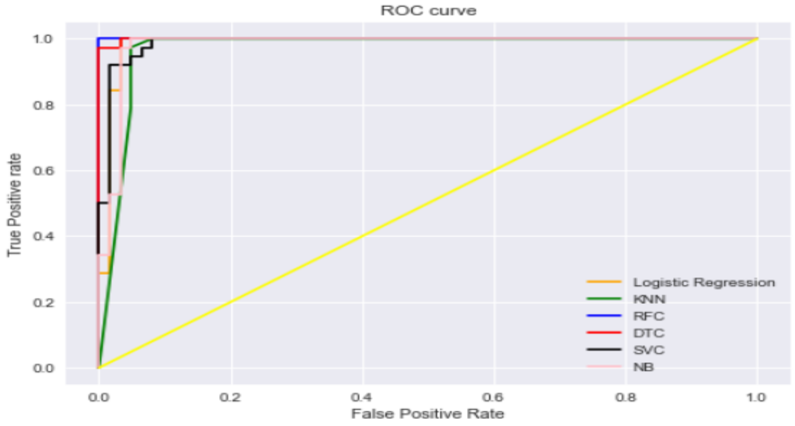
Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

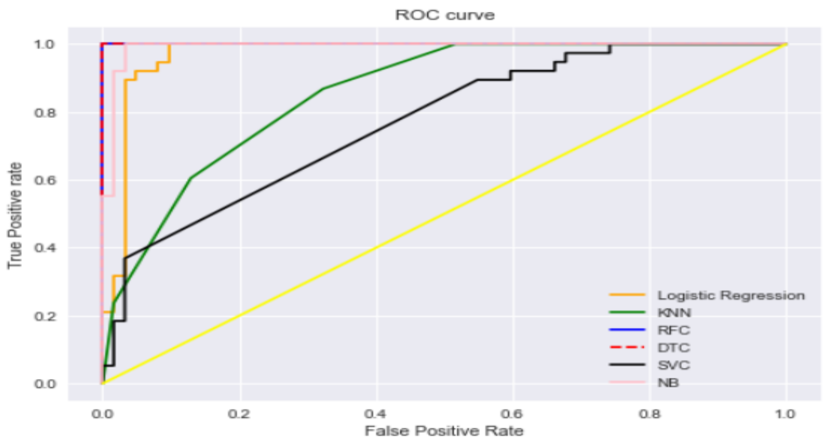
Results

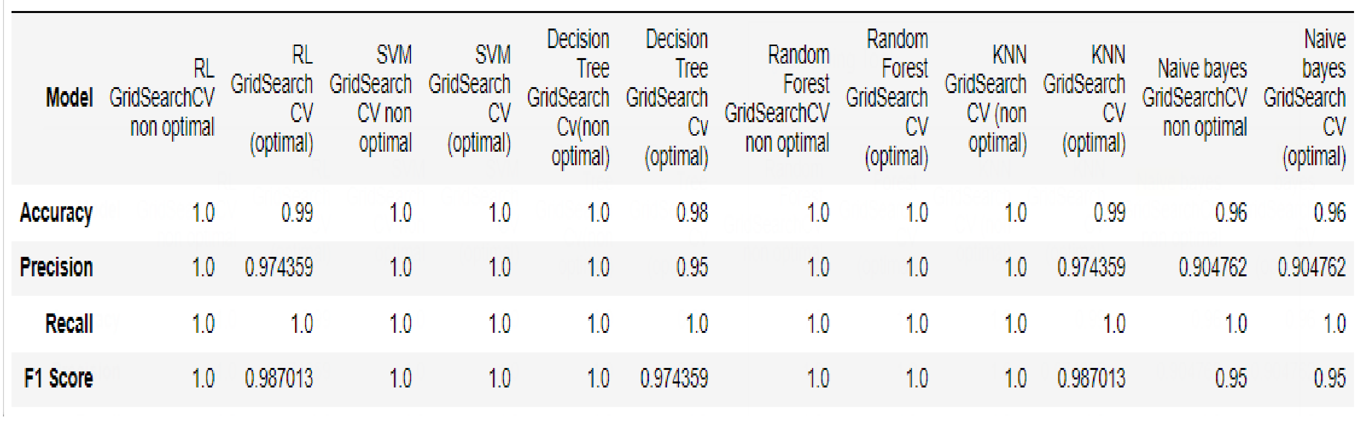
This Following table shows different results for different models that we used to classify our data, we applied these modals before and after optimizing our dataset:

- We have 5 features in Optimal modals, The features are : ['Hemoglobin','Specific_Gravity','Diabetes_Mellitus', 'albumin','Packed_Cell_Volume']



- We have 12 features in non Optimal modals (if the features are >12, the modal performance decreases), The features are : ['Hemoglobin','Specific_Gravity','Hypertension','albumin','Diabetes_Mellitus','Packed_Cell_Volume','Red_Blood_Cell_Count','Sugar','Appetite','Sodium','White_Blood_Cell_Count','Anemia']



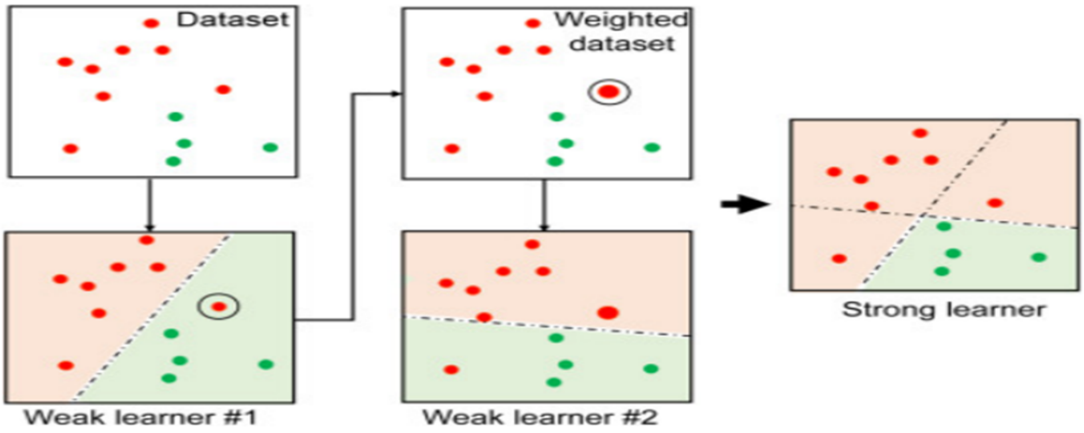


Model	RL GridSearchCV non optimal	RL GridSearch CV (optimal)	SVM GridSearch CV non optimal	SVM GridSearch CV (optimal)	Decision Tree GridSearch Cv(non optimal)	Decision Tree GridSearch Cv (optimal)	Random Forest GridSearchCV non optimal	Random Forest GridSearch CV (optimal)	KNN GridSearch CV (non optimal)	KNN GridSearch CV (optimal)	Naive bayes GridSearchCV non optimal	Naive bayes GridSearch CV (optimal)
Accuracy	1.0	0.99	1.0	1.0	1.0	0.98	1.0	1.0	1.0	0.99	0.96	0.96
Precision	1.0	0.974359	1.0	1.0	1.0	0.95	1.0	1.0	1.0	0.974359	0.904762	0.904762
Recall	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
F1 Score	1.0	0.987013	1.0	1.0	1.0	0.974359	1.0	1.0	1.0	0.987013	0.95	0.95

AdaBoost with Selected Features

Ada-boost or Adaptive Boosting is an ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get a high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and train the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as a base classifier if it accepts weights on the training set. Adaboost should meet two conditions:

- The classifier should be trained interactively on various weighted training examples.
- In each iteration, it tries to provide an excellent fit for these examples by minimizing training error.



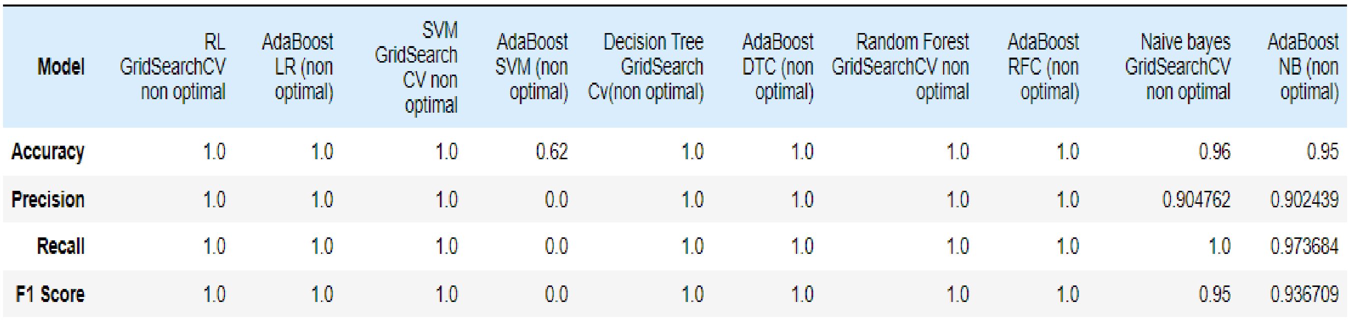
Comparing results

For this section, we will be comparing our results. In this process we will compare the accuracy for each model through the **5 methods** used. That means for each model we will compare the accuracy of the AdaBoost with optimal and non optimal data.

We use these 5 models:

- SVM
- Decision Tree
- Logistic Regression
- Random Forest
- Naive Bayes

This table shows the different results for the different models.



Model	RL GridSearchCV non optimal	AdaBoost LR (non optimal)	SVM GridSearch CV non optimal	AdaBoost SVM (non optimal)	Decision Tree GridSearch Cv(non optimal)	AdaBoost DTC (non optimal)	Random Forest GridSearchCV non optimal	AdaBoost RFC (non optimal)	Naive bayes GridSearchCV non optimal	AdaBoost NB (non optimal)
Accuracy	1.0	1.0	1.0	0.62	1.0	1.0	1.0	1.0	0.96	0.95
Precision	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.904762	0.902439
Recall	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	0.973684
F1 Score	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.95	0.936709

Conclusion

In this chapter, we went in detail through our process for modeling according to the first article and applying different methods so we can get the best results.

Modeling: 2nd Article Diagnosis of Chronic Kidney Disease Using Effective Classification

Introduction

Feature Selection

Recursive Feature Elimination (RFE)

Given an external estimator (we chose KNN estimator) that assigns weights to features (e.g., the coefficients of a linear model), the goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through any specific attribute or callable. Then, the least important features are pruned from the current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached

Data Split

Following the steps performed by the article we chose to divide our dataset into 75% training set and 25% test set .

Models

we implemented the following algorithms in our work:

- KNN
- SVM
- Logistic regression
- Decision Tree
- Random Forest Classifier

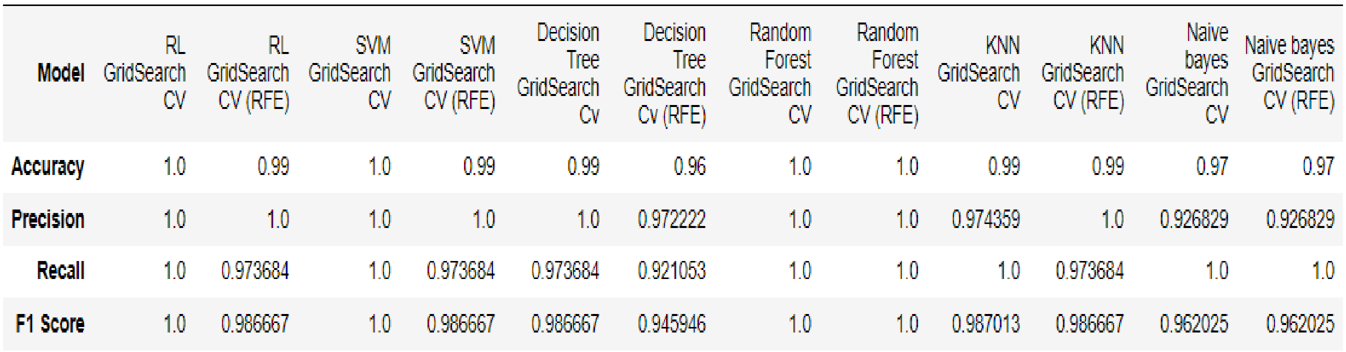
Decision Tree

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model

Results



Model	RL GridSearch CV	RL GridSearch CV (RFE)	SVM GridSearch CV	SVM GridSearch CV (RFE)	Decision Tree GridSearch Cv	Decision Tree GridSearch Cv (RFE)	Random Forest GridSearch CV	Random Forest GridSearch CV (RFE)	KNN GridSearch CV	KNN GridSearch CV (RFE)	Naive bayes GridSearch CV	Naive bayes GridSearch CV (RFE)
Accuracy	1.0	0.99	1.0	0.99	0.99	0.96	1.0	1.0	0.99	0.99	0.97	0.97
Precision	1.0	1.0	1.0	1.0	1.0	0.972222	1.0	1.0	0.974359	1.0	0.926829	0.926829
Recall	1.0	0.973684	1.0	0.973684	0.973684	0.921053	1.0	1.0	1.0	0.973684	1.0	1.0
F1 Score	1.0	0.986667	1.0	0.986667	0.986667	0.945946	1.0	1.0	0.987013	0.986667	0.962025	0.962025

Comparing results

It is worth noting that our dataset is an unbalanced one therefore, the only measure of comparison between the performance of the different models is f1_score.
From the table above,we noticed that all of the models that we have implemented have had very high performance reaching in some cases 100%.
However, all the models except random forest have witnessed a slight decrease in all performance indicators with the usage of feature selection techniques.
This is understandable because random forest is a rule-based model.

Furthermore, the objective of feature selection techniques is to help reduce computational cost and time without sacrificing accuracy, but in our case we noticed on one hand a slight decrease in performance on the other hand we gained a lot in terms of run time and complexity.
We concluded that such a slight decrease in performance is outweighed by the amount of computational power gained.

Conclusion

In this chapter, we went in detail through our process for modeling according to the second article and applying different methods so we can get the best results.while applying specific dimensionality reduction techniques such as RFE .