Feature Selection and Classification for enhancing Chronic Kidney Disease Diagnose

Oussama Nasri, Karam Fayek,Jihen saidi,Malek Frih,Mouhib Dalhoumi

December 15, 2022

1 Introduction

After an attentive reading of two articles about the analysis and the testing of the performance of some communally used classification methods for the prediction of the CKD disease (article 1 : Diagnosis of Chronic Kidney Disease Using Effective Classification , article 2: Boosted Classifier and Features Selection for Enhancing Chronic Kidney Disease Diagnose) we would like to deliver this rapport as a reproduction and a critic for those two articles.

To do so, we started by a careful reading of the articles followed by a profound data exploration then a reproduction of the methods used in the articles. In this reproduction we tried our best to copy the articles way of implementing the feature classification and classification methods. Then we have done a comparison of those methods using clear metrics. Finally, we tried to add our touch and beat the article's scores.

2 Our Methodology

As an attempt to make our work efficient yet rich, we tried to divide the work in an equal manner. As for the approach we chose to work with the CRISP-DM framework.

1-Business understanding
2-Data understanding
3-Data preparation
4-Modeling
5-Evaluation
6-Deployment

3 Business Understanding

Chronic kidney disease or CKD for short, is, as the name suggest, a type of kidney disease in which there is gradual loss of kidney function over a period of months to years. Initially, there are generally no symptoms. Later,however, symptoms may include leg swelling, feeling tired, vomiting, loss of appetite, and confusion. Complications include an increased risk of heart disease, high blood pressure, bone disease, and anaemia. CKD is a global health problem with a high mortality rate, and it induces other diseases. As there are no obvious symptoms during the early stages of CKD, patients often do not notice the disease, which ends up in most cases leading to a complete loss of kidney function.

Early detection of CKD allows patients to receive timely treatment to improve the progression of this disease. For the previously mentioned reasons, the main objective is to design and implement an automatic machine learning model that, based on data from clinical laboratories, makes predictions in the diagnosis of CKD in its initial stages and contributes to the reduction of significant complications in the disease such as dialysis processes, kidney transplantation or reaching death even, and therefore helping reduce the mortality rate and costs for the health system which have been the two main motivations since patients are charged very high amounts in our day and age, coupled with the increasing number of cases diagnosed by CKD and the scarcity of specialist physicians(neurologists and radiologists) .The main criterion of success for this project, with the help of machine learning, is

to identify the behaviors or behavior patterns in the initial stages of CKD to improve the quality of life of patients.

4 Data Understanding

Overview:
By examining our data set we find that we have 400 rows and 25 columns(features) including the target feature(label) which is the last column ' class' that indicates whether the patient has chronic kidney disease or not. We are here in a case of supervised learning and we have as a main goal to adequately approach this data set in order to enhance the CKD diagnosis.

Existing features and their utilities:

age - age
bp - blood pressure
sg - specific gravity
al - albumin
su - sugar
rbc - red blood cells
pc - pus cell
pcc - pus cell clumps
ba - bacteria
bgr - blood glucose random
bu - blood urea
sc - serum creatinine
sod - sodium
pot - potassium
hemo - hemoglobin
pcv - packed cell volume
wc - white blood cell count
rc - red blood cell count
htn - hypertension
dm - diabetes mellitus
cad - coronary artery disease
appet - appetite
pe - pedal edema
ane - anemia
class - class

REMARK:
Our dataset includes numerical type of features and nominal types with two outcomes which give us a hint about the methods to be used when preparing the data .

A BETTER LOOK AT OUR DATASET:

The signs and symptoms of chronic kidney disease are multiple Some of them are hypertension, diabetes, older age, cardiovascular disease, anemia, the body fluids, urine ...
However based on the articles The symptoms of this disease are obscure and the presence of one of these signs doesn't necessarily mean the patient is affected with a CKD.
As a result ,We conducted a further research on the data set and we looked for the relationship between each of these features and our target disease in order to better understand the hidden patterns within our data set and facilitate the data preparation process .
specific gravity :
The normal range for urine specific gravity is 1.005 to 1.030. Normal value ranges may vary slightly among different laboratories.
In fact damage to the kidney's tubules affects the ability of the kidney to re-absorb water. As a result, the urine remains dilute (good indicator for CKD) .

Blood pressure:
Blood pressure should be controlled to less than 130/80 if you have CKD. Albumin:
A normal amount of albumin in your urine is less than 30 mg/g. Anything above 30 mg/g may mean you have kidney disease, even if your GFR number is above 60.
Sugar:
Sugar is not a problem for the kidneys unless the blood sugar level gets too high. This commonly occurs in both Type 1 and Type 2 diabetes. Once the blood sugar level gets higher than 180 mg/dl, the kidneys start to spill sugar into the urine. The higher the blood sugar, the more sugar comes out in the urine.
Red blood cells count:
A normal RBC count would be around: men – 4.0 to 5.9 x 10*12/L. women – 3.8 to 5.2 x 10*12/L.
Anemia:
Most people with kidney disease will develop anemia. Anemia can happen early in the course of kidney disease and grow worse as kidneys fail and can no longer make EPO.
Pus cells and pus cells clumps :
pus cells/ pus cells clumps/bacteria can be both indicators of an existing kidney problem when checking the urine for their presence.
Packed cell volume:
The packed cell volume (PCV) is a measurement of the proportion of blood that is made up of cells. The value is expressed as a percentage or fraction of cells in blood. For example, a PCV of 40 percent means that there are 40 millilitres of cells in 100 millilitres of blood.
Diabetes mellitus:
Refers to a group of diseases that affect how the body uses blood sugar (glucose).
Coronary artery disease:
When the heart is no longer pumping efficiently it becomes congested with blood, causing pressure to build up in the main vein connected to the kidneys and leading to congestion of blood in the kidneys, too.
Pedal edema :
Bilateral pedal edema is the most common symptom in chronic kidney diseased patients. It occurs due to the loss of functioning of the kidney. This may lead to fluid accumulation in the body and also an accumulation of excretory products or waste products like creatinine, uric acid, urea levels are increases in blood.

SOME PATTERS AND RELATIONSHIPS WE FIND WHEN UNDERSTANDING THE DATASET:
-By analysing the data we see that pus cells are highly correlated to pus cells clumps in fact
-A normal PC test results automatically in the non presence of pus cells clumps in the urine (we can use one feature)
-Blood glucose random and sugar levels are the same thing they can be a sign of a diabetes which can be itself an indicator for CKD
-Anemia is highly correlated to the hemoglobin levels and the red blood cells present in the blood so we can use two out of three
-Appetite is not necessarily a sign of CKD , it can refer to the presence of multiple health issues and cant be considered as highly important
-Blood pressure within a higher range than normal is directly associated with hyper tension so there is a high correlation between these features.

5 Data preparation

In the preprossessing part we tried to stay faithful to the articles at first. So what we did is estimating the missing values,deleting the outliers,encoding data,normalization and check for unbalances in the data set.

5.1 Missing Values

For the missing values we found 1012 missing values.Which is considerable amount of data which means deleting them will affect the accuracy of our model. So we imputed them with the KNN.

5.2 Detecting Outliers

We know that the best way to know if your data contains outliers is to consult a specialist who is in this domain.So we consulted a doctor with our data and he told us the data is accurate and contains no outliers.

5.3 Encoding data

Our data contains a lot of categorical data, 14 variables in fact including the label ('Class'). These features are al,su,rbc,pc,pcc,ba,htn,dm,cad,appet,pe,ane,class.
So we used the Lebel Encoder method to make these column containing numerical values. And so we made our these columns take the following values:
'al' {0,1,2,3,4,5}
'su' {0,1,2,3,4,5}
'rbc' {0,1}
'pc' {0,1}
'pcc' {0,1}
'ba' {0,1}
'htn' {0,1}
'dm' {0,1}
'cad' {0,1}
'appet' {0,1}
'pe' {0,1}
'ane' {0,1}
'class' {0,1}

5.4 Data Scaling & Normalization

The goal of scaling data is to transform the data so that it has a consistent scale, meaning that the values have similar ranges and are not skewed by very large or very small values. This is important because many machine learning algorithms use a distance-based measure to compare observations, and if the data is not scaled, then features with large values will dominate the distance calculation and have a disproportionate influence on the model.
Scaling data can also help to improve the performance of the machine learning algorithm, by making the optimization problem easier to solve and speeding up the training process. Additionally, scaling can help to improve the interpretability of the model by making the coefficients more comparable and easier to interpret.
To do that we used the Standard Scaling in article 1* and the Min Max Scaling in the article 2*.
The Standard Scaler will standardize the data by subtracting the mean and dividing by the standard deviation, resulting in a dataset with zero mean and unit variance.
The mathematical formula:

$$z = (x - min)/(max - min)$$

The Min Max Scaler will first determine the minimum and maximum values for each feature. Then, for each feature, you subtract the minimum value from the original value and divide by the range:
The mathematical formula:

$$z = (x - mean)/std$$

5.5 Checking for unbalance

After calculating the number of people with CKD and people with no CKD in this dataset, we can see that we have :
Pourcentage of ckd : 62.5 % 250 samples
pourcentage of $\text{non}_c kd$: $37.5\%150 samples$
$The dataset can be considered balanced.$

6 Feature Selection

6.1 RFECV

In the article 1, authors used the RFE for the feature selection.
The RFE works by training a model on the whole dataset and then iteratively removing the least important features, until a specified number of features remains. The importance of the features is determined by the model, using a metric such as the feature weights or feature importance values. Then we applied added a layer of Cross Validation to know the significance of the features selected based on a score.
So the optimum number of features: 19
In the article 2, authors used the CFS for the feature selection.

6.2 CFS

CFS is a very powerful feature method. It calculates the correlation between each feature and all the other features and its correlation with the label and then the CFS gives each feature a score and then keeps the ones with the highest scores.
We will take 17 best feature when using the classification methods used in this article, as imposed by the article.

7 Classification

7.1 Definition of metrics

7.1.1 Confusion Matrix

The confusion matrix takes a predicted results from a classification method and calculates the number of TP (True Positive) values , TN (True Negative) values , FP (False Positive) values ,FN (False Negative) values. And then plots a matrix describing these values.
These four values will help define 4 important metrics used to analyse the performance of any classification Algorithm.
Precision
Accuracy
Recall
F1-Score

7.1.2 Precision

Precision is the proportion of positive identifications that were correctly predicted.

$$Precision = TruePositives/(TruePositives + FalsePositives)$$

7.1.3 Accuracy

Accuracy is the proportion of the correctly identified values to the total number of values.

$$Precision = TruePositives + TrueNegatives/(TotalNumberofprediction)$$

7.1.4 Recall

The recall is the proportion of correctly classified positive.

$$Recall = TruePositives/(TruePositives + FalseNegatives)$$

7.1.5 F1-Score

F1-score combines precision and recall into one measure.

$$F1 - Score = 2[(PrecisionxRecall)/(Precision + Recall)]$$

7.2 Article 1

7.2.1 SVM + RBF (Radial Basis Function)

The Support Vector Machine algorithm primarily creates a line to separate the dataset into classes, enabling it to decide the test data into which classes it belongs.

The Radial Basis Function (RBF) was employed as a kernel for classification data.
The usage of the RBF means that we will use this mathematical equation in the calculation of the distance between the points:

$$K(X, X') = e^{\frac{-||X-X'||^2}{2\sigma^2}} \quad (1)$$

For this algorithm we chose as hyper-parameters:
'C': 100, 'gamma': 0.0001
'C': trades off classification of training examples against simplicity of the decision surface.
'Gamma':Defines how much influence a single training example has. We can say it is the distance separating the drawn points in the same class.

7.2.2 SVM + RBF (Radial Basis Function) : Test Results

	precision	recall	f1-score	support
0.0	0.86	0.81	0.83	62
1.0	0.71	0.79	0.75	38
accuracy			0.80	100
macro avg	0.79	0.80	0.79	100
weighted avg	0.81	0.80	0.80	100

7.2.3 Knn Classifier

In a K-Nearest Neighbour classifier, the output for a given input is determined by its k nearest neighbors in the training data. The output is typically the majority class among the k nearest neighbors, although other methods such as weighted voting can also be used.

For this algorithm we chose as hyper-parameters:
'leaf-size': 1
'n-neighbors': 1
leaf-size: means that only the single nearest neighbor or the five nearest neighbors are used to make predictions.
'n-neighbors' : The number of neighbors determines the number of training examples that are used to make predictions for a new input.

7.2.4 Knn Classifier: Test Results

	precision	recall	f1-score	support
0.0	0.87	0.66	0.75	62
1.0	0.60	0.84	0.70	38
accuracy			0.73	100
macro avg	0.74	0.75	0.73	100
weighted avg	0.77	0.73	0.73	100

7.2.5 Decision Tree

A decision tree works by dividing the input space into a set of rectangular regions, called "nodes," and making predictions for a new input by traversing the tree from the root node to a leaf node. The tree is constructed by choosing the split points in the input space that result in the greatest reduction in the uncertainty of the output. This process is repeated recursively for each node in the tree until the tree is fully grown.

For this algorithm we chose as hyper-parameters:
'ccp-alpha': 0.001
'criterion': 'gini'
'max-depth': 5
'max-features': 'sqrt'
'ccp-alpha': controls the amount of pruning that is performed on the tree.
'criterion':the Gini impurity (used in this case) is a measure of the purity of the data at a given node. It is used to determine the best split point for the data at each node in the tree.

$$Gini = 1 - (p1^2 + p2^2 + ... + pn^2) \tag{2}$$

'max-depth': of the tree. The depth of a tree is the number of levels of nodes in the tree, starting from the root node at level 0.
'max-features': the number of features that are considered when determining the best split point at each node.

7.2.6 Decision Tree: Test Results

	precision	recall	f1-score	support
0.0	0.95	0.97	0.96	61
1.0	0.95	0.92	0.94	39
accuracy			0.95	100
macro avg	0.95	0.95	0.95	100
weighted avg	0.95	0.95	0.95	100

7.2.7 Random Forest

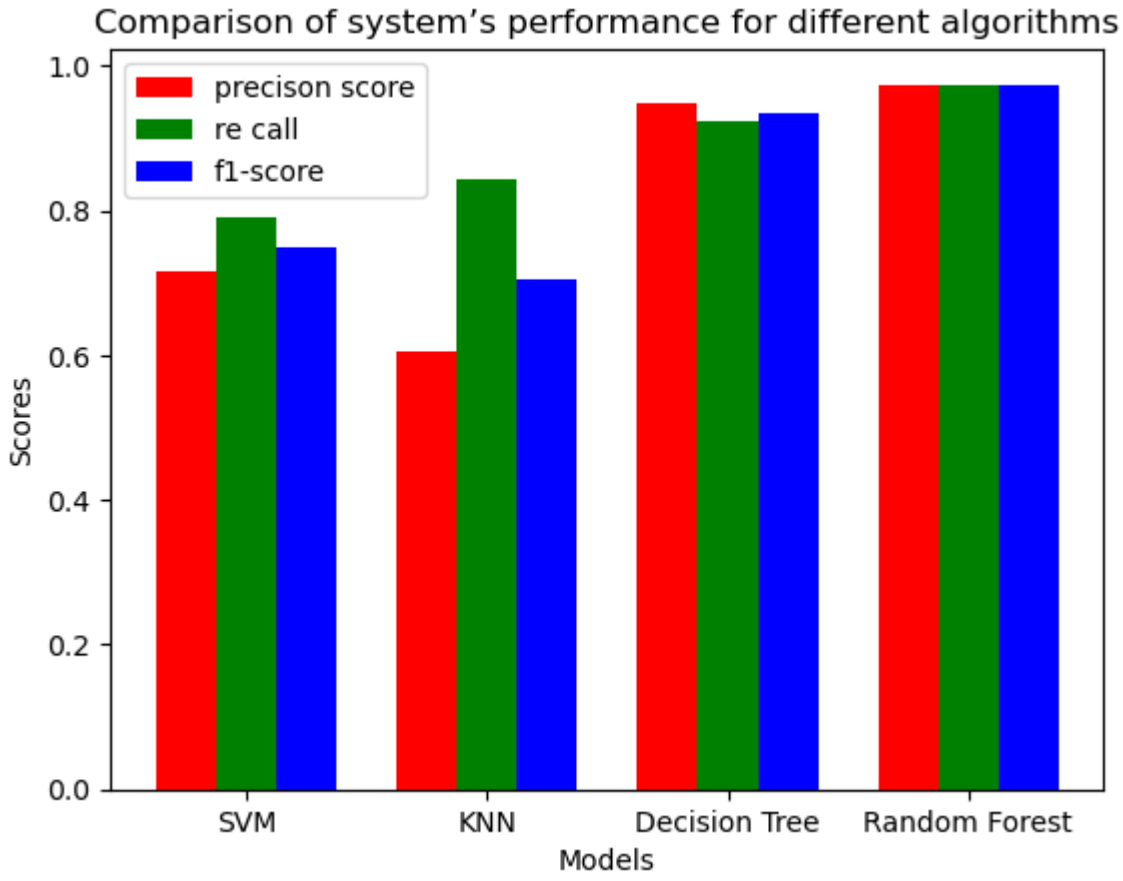
A random forest is a type of machine learning algorithm that uses ensembles of decision trees to make predictions. It works by training a large number of decision trees on a dataset, and then combining the predictions made by each individual tree in order to make a more accurate overall prediction.
'criterion': 'gini'
'max-depth': 4
'max-features': 'auto'

'n-estimators': 200
'n-estimators' : determines the number of decision trees that will be trained in the ensemble.

7.2.8 Random Forest: Test Results

	precision	recall	f1-score	support
0.0	0.95	0.97	0.96	61
1.0	0.95	0.92	0.94	39
accuracy			0.95	100
macro avg	0.95	0.95	0.95	100
weighted avg	0.95	0.95	0.95	100

7.2.9 SVM Vs Knn Vs Decision Tree Vs Random Forest



We can actually see that Random Forest is better than the SVM,KNN,Decision Tree in terms of Precision, recall,F1-Score.

7.3 Article 2

7.3.1 Knn Classifier: Test Results

	precision	recall	f1-score	support
0	1.00	0.93	0.97	74
1	0.90	1.00	0.95	46
accuracy			0.96	120
macro avg	0.95	0.97	0.96	120
weighted avg	0.96	0.96	0.96	120

7.3.2 SVM: Test Results

	precision	recall	f1-score	support
0	1.00	0.97	0.99	74
1	0.96	1.00	0.98	46
accuracy			0.98	120
macro avg	0.98	0.99	0.98	120
weighted avg	0.98	0.98	0.98	120

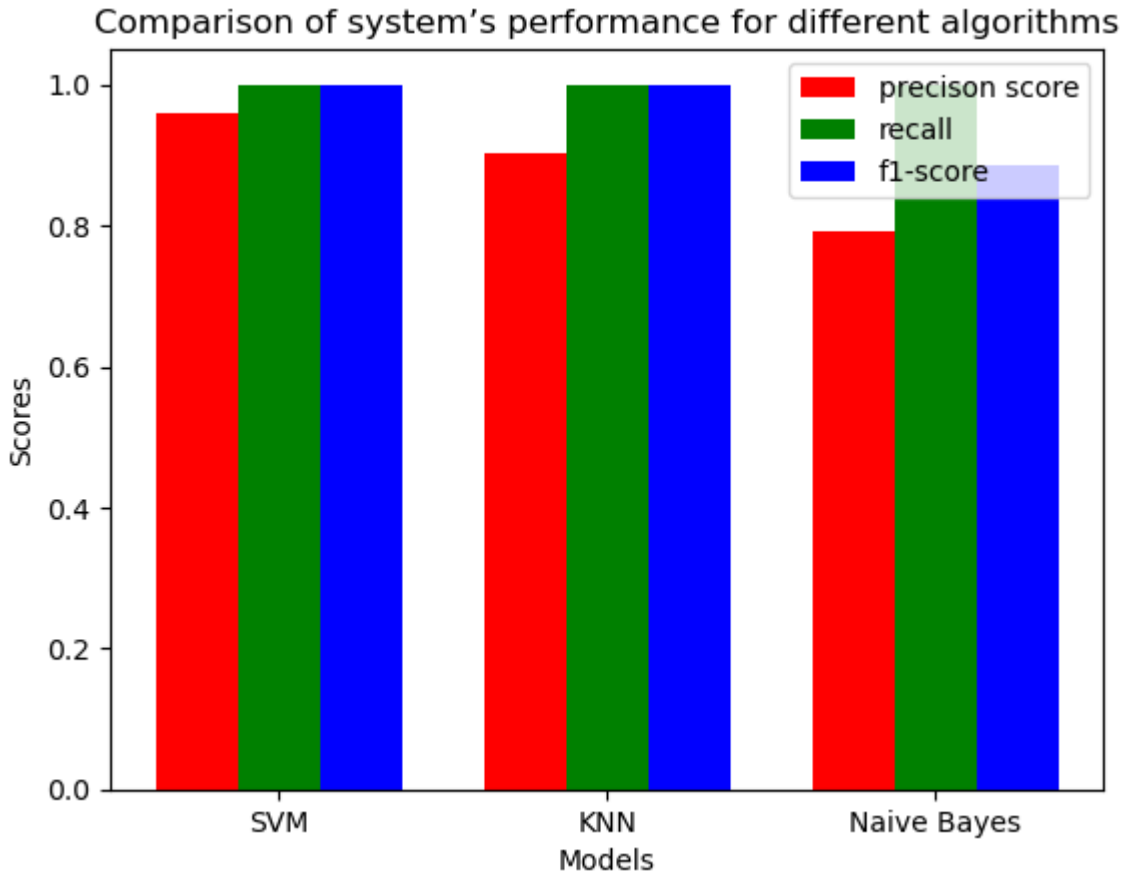
7.3.3 Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm that is often used for classification tasks. It is called "naive" because it makes the assumption that all of the features in a dataset are independent of each other, which is not always the case in real-world data.
The algorithm works by calculating the probability that a given input belongs to each class, based on the features in the input. It then assigns the input to the class with the highest probability.

7.3.4 Naive Bayes: Test Results

	precision	recall	f1-score	support
0	0.84	1.00	0.91	62
1	1.00	0.79	0.88	58
accuracy			0.90	120
macro avg	0.92	0.90	0.90	120
weighted avg	0.92	0.90	0.90	120

7.3.5 SVM Vs Knn Vs Naive Baayes



SVM and KNN give similar results in terms of recall and F1-score but SVM has a slight advantage in the precision score

7.3.6 Knn Classifier + CFS: Test Results

	precision	recall	f1-score	support
0	1.00	0.99	0.99	74
1	0.98	1.00	0.99	46
accuracy			0.99	120
macro avg	0.99	0.99	0.99	120
weighted avg	0.99	0.99	0.99	120

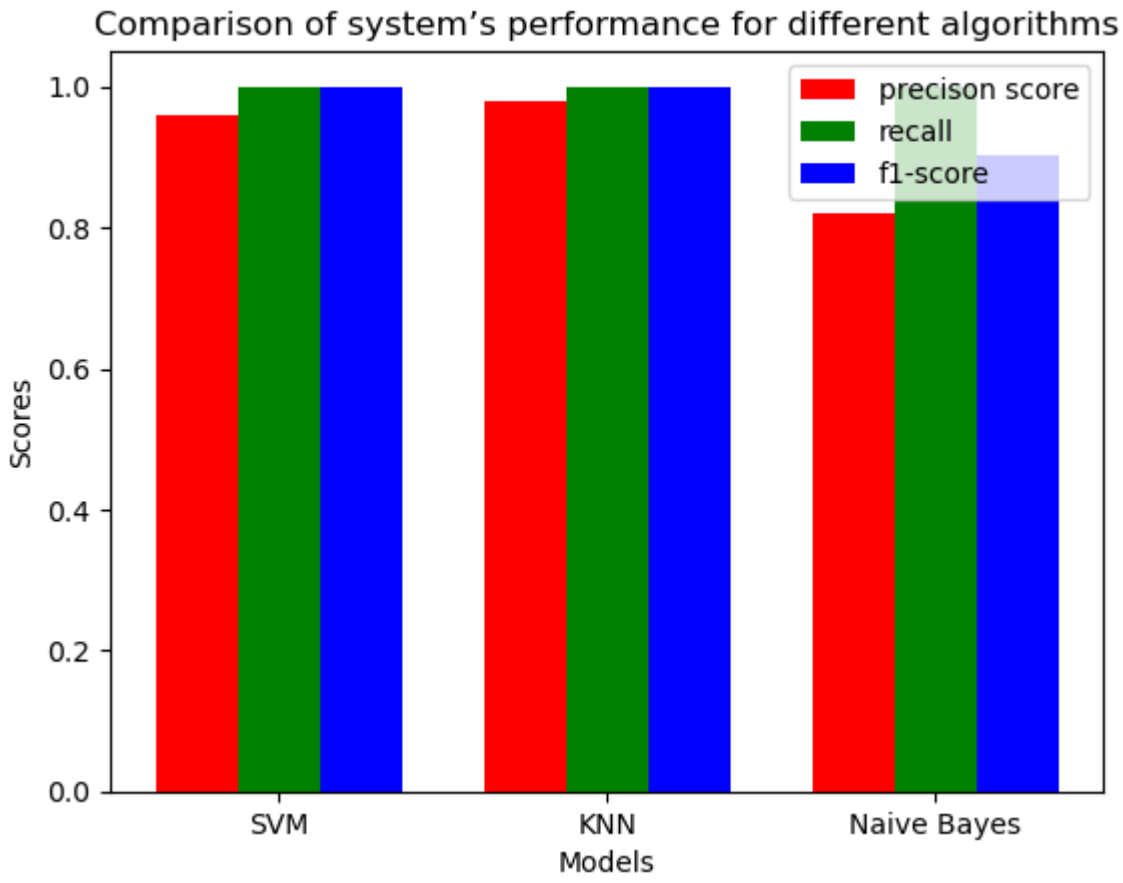
7.3.7 SVM + CFS: Test Results

	precision	recall	f1-score	support
0	1.00	0.97	0.99	74
1	0.96	1.00	0.98	46
accuracy			0.98	120
macro avg	0.98	0.99	0.98	120
weighted avg	0.98	0.98	0.98	120

7.3.8 Naive Bayes + CFS: Test Results

	precision	recall	f1-score	support
0	0.86	1.00	0.93	64
1	1.00	0.82	0.90	56
accuracy			0.92	120
macro avg	0.93	0.91	0.91	120
weighted avg	0.93	0.92	0.92	120

7.3.9 SVM + CFS Vs Knn + CFS Vs Naive Baayes + CFS



SVM and KNN give similar results and are in fact better than the Naive Bayes.

7.3.10 Knn Classifier + CFS + Adaboost: Test Results

	precision	recall	f1-score	support
0	0.80	1.00	0.89	74
1	1.00	0.61	0.76	46
accuracy			0.85	120
macro avg	0.90	0.80	0.82	120
weighted avg	0.88	0.85	0.84	120

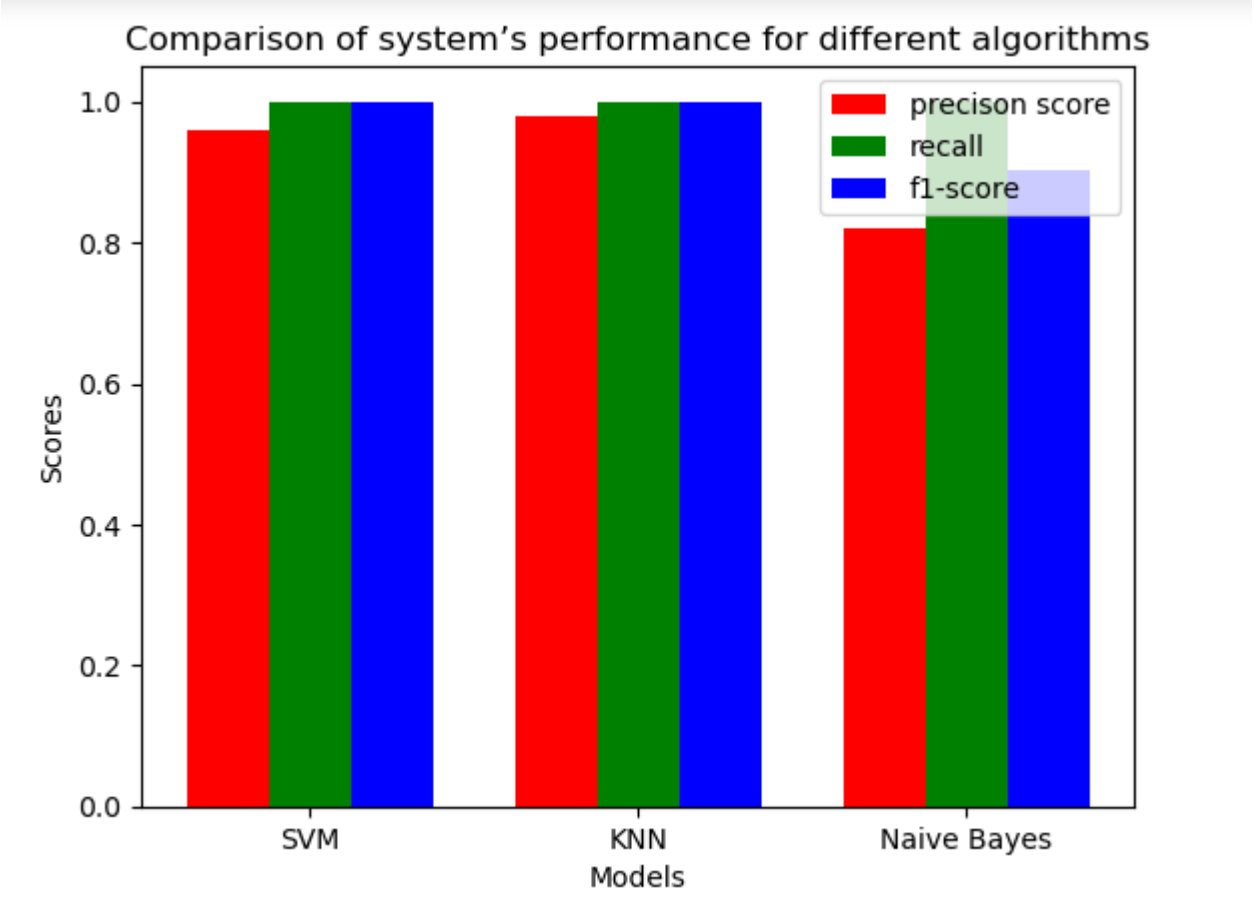
7.3.11 SVM + CFS + Adaboost: Test Results

	precision	recall	f1-score	support
0	1.00	0.99	0.99	74
1	0.98	1.00	0.99	46
accuracy			0.99	120
macro avg	0.99	0.99	0.99	120
weighted avg	0.99	0.99	0.99	120

7.3.12 Naive Bayes + CFS + Adaboost: Test Results

	precision	recall	f1-score	support
0	1.00	0.93	0.97	74
1	0.90	1.00	0.95	46
accuracy			0.96	120
macro avg	0.95	0.97	0.96	120
weighted avg	0.96	0.96	0.96	120

7.3.13 SVM + CFS + Adaboost Vs Knn + CFS + Adaboost Vs Naive Baayes + CFS + Adaboost



SVM and KNN give similar results and are in fact better than the Naive Bayes.

7.4 Comparison of the performance of the algorithm of Article 1 and Article 2

Parameter	Classifiers	
	SVM	KNN
	Article 1	
	RFE	
Accuracy	0.8	0.73
Precision	0.79	0.74
Recall	0.8	0.75
F1-Score	0.79	0.73
	Article 2	
	CFS	
Accuracy	0.98	0.85
Precision	0.98	0.9
Recall	0.99	0.8
F1-Score	0.98	0.82

It is obvious that the results obtained after implementing the CFS lead to better metrics!

7.5 Our try to enhance the performance of the classification

To enhance the performance of our models we tried instead of implementing a feature selection algorithm we opted for a Dominantly Reduction algorithm. So we used the PCA.

7.5.1 Principal component analysis (PCA)

PCA is a statistical technique that is used to analyze the variations in a dataset. It is a way of reducing the dimensionality of the data by identifying the underlying patterns and structure in the data.
'logistic-C': 1.0
'pca-n-components': 13
'logistic-C':The 'C' parameter in logistic regression is a regularization parameter that controls the strength of the regularization. It determines how much the model should be penalized for having large coefficients, which can help to prevent overfitting.
'pca-n-components':specifies the number of principal components to use in the transformed dataset

7.5.2 Knn + PCA

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	74
1.0	0.98	0.96	0.97	46
accuracy			0.97	120
macro avg	0.98	0.97	0.97	120
weighted avg	0.98	0.97	0.97	120

7.5.3 SVM + PCA

	precision	recall	f1-score	support
0.0	0.99	0.99	0.99	74
1.0	0.98	0.98	0.98	46
accuracy			0.98	120
macro avg	0.98	0.98	0.98	120
weighted avg	0.98	0.98	0.98	120

7.5.4 Knn + PCA

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	74
1.0	0.98	0.96	0.97	46
accuracy			0.97	120
macro avg	0.98	0.97	0.97	120
weighted avg	0.98	0.97	0.97	120

7.5.5 Decision Tree + PCA

	precision	recall	f1-score	support
0.0	0.96	0.96	0.96	74
1.0	0.93	0.93	0.93	46
accuracy			0.95	120
macro avg	0.95	0.95	0.95	120
weighted avg	0.95	0.95	0.95	120

7.6 Conclusion

We can clearly see an imporvement in the results after doing a dimensinalty reduction instead of a feature selection.

[1] [2]

[1]Article 1:Diagnosis of Chronic Kidney Disease Using Effective Classification
[2]Article 2:Boosted Classifier and Features Selection for Enhancing Chronic Kidney Disease Diagnose