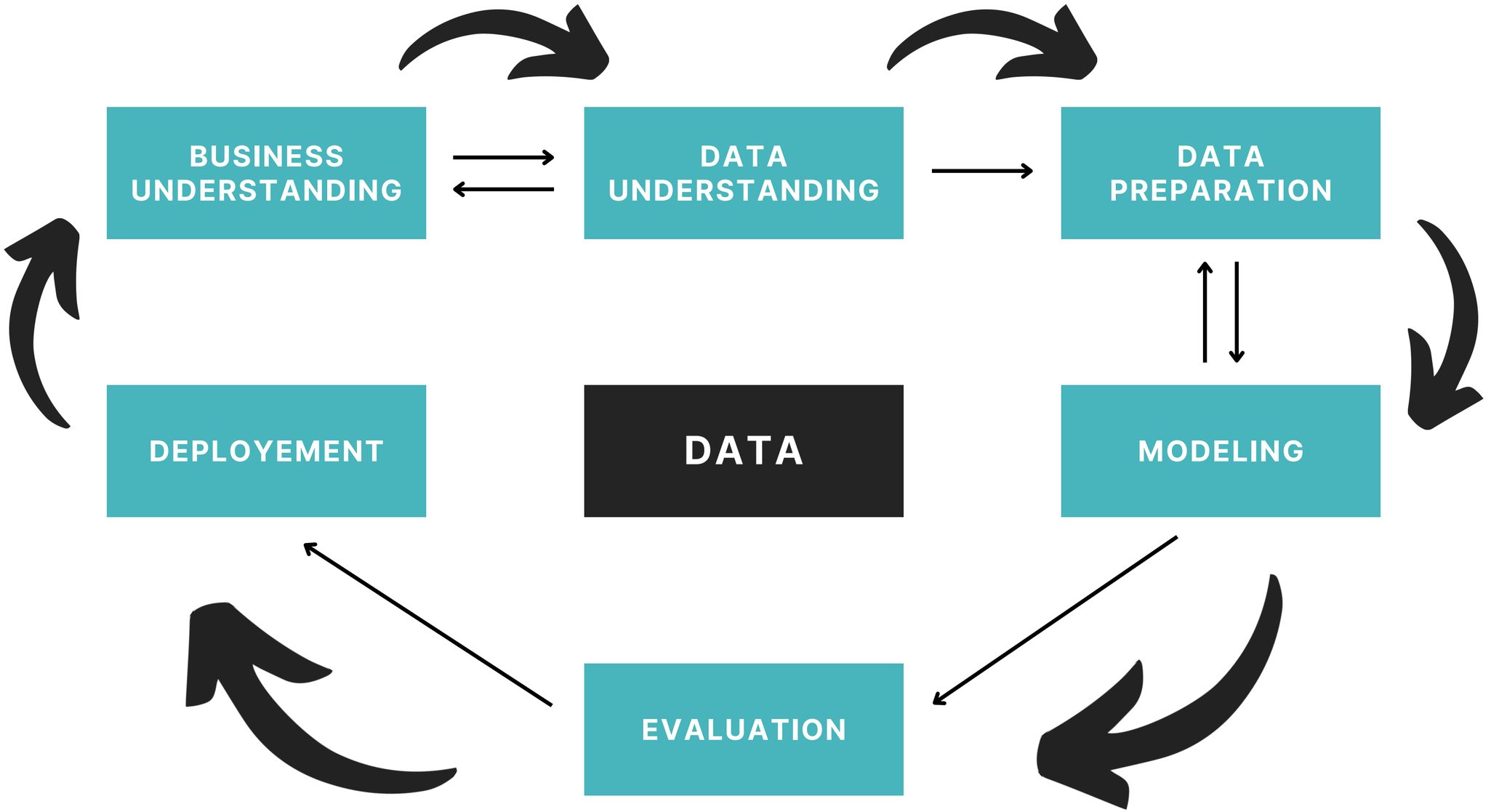
MACHINE LEARNING PROJECT

CHRONIC KIDNEY DISEASE

PLAN

- BUSINESS UNDERSTANDING
- DATA UNDERSTANDING
- DATA PREPROCESSING
- MODELING
- EVALUATION
- DEPLOYMENT

CRISP-DM methodology



BUSINESS UNDERSTANDING

01

Business Understanding

The kidney

Chronic kidney disease

Stadiums

etiology

Risk factors

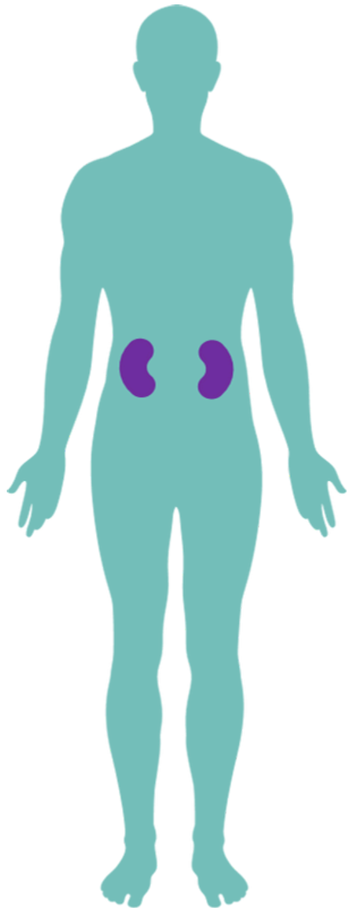
Symptoms from the CKD

The kidney

Definition:

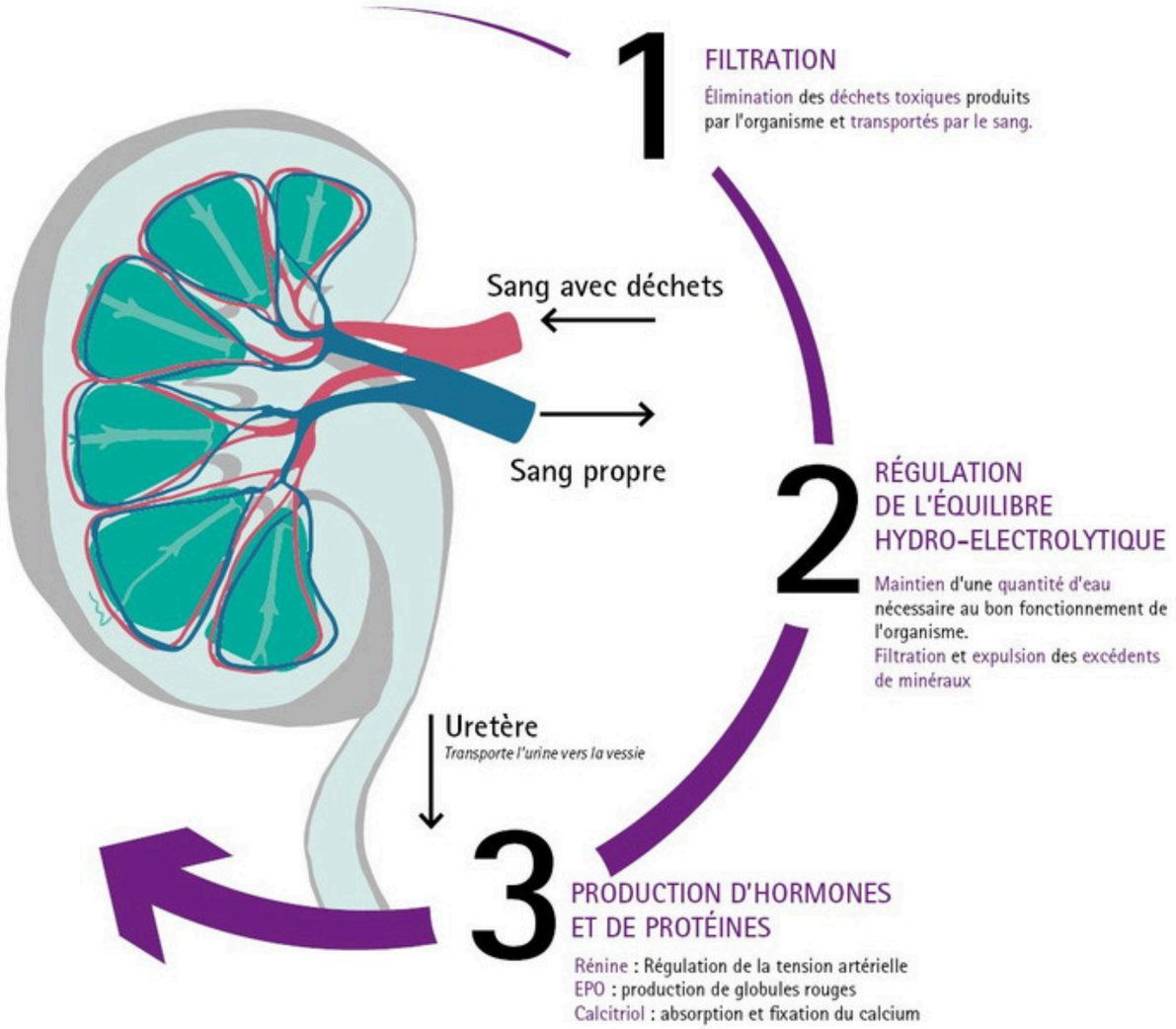
- The kidney is a bean-shaped organ located in the lower mid-back in the retroperitoneal space.

- Our kidneys are vital organs. This means that we need at least one healthy kidney to live.



The kidney

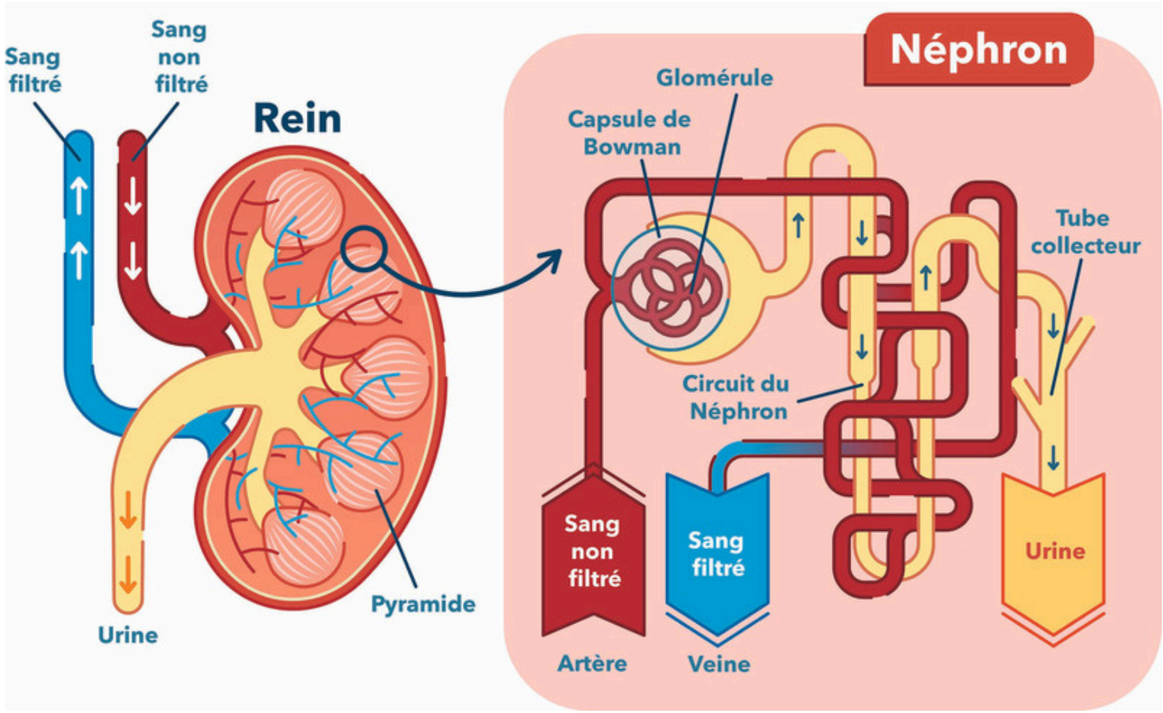
Role:



The kidney

Functioning:

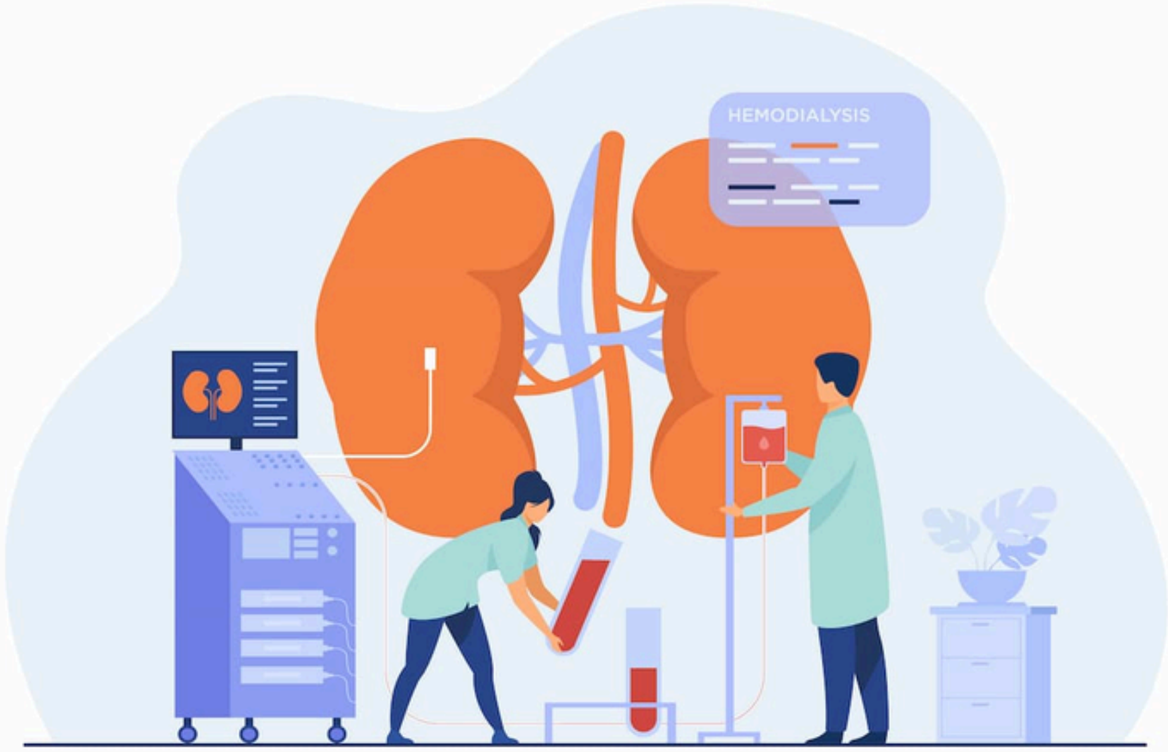
- Blood enters the kidneys which contain millions of filtering units called nephrons; Each nephron includes a filter called a glomerulus and a tubule.
- The glomerulus filters the blood, removing waste and extra fluid to produce urine.The tubule returns necessary substances to the blood.
- Clean, filtered blood leaves the kidneys and returns to the rest of the body.



Chronic kidney disease

Chronic kidney failure is caused by a reduction in the number of functional nephrons. The mechanism of this reduction is twofold:

- Initial destruction linked to the causative disease.
- Hyperfunction of the remaining nephrons will lead to glomerulosclerosis.



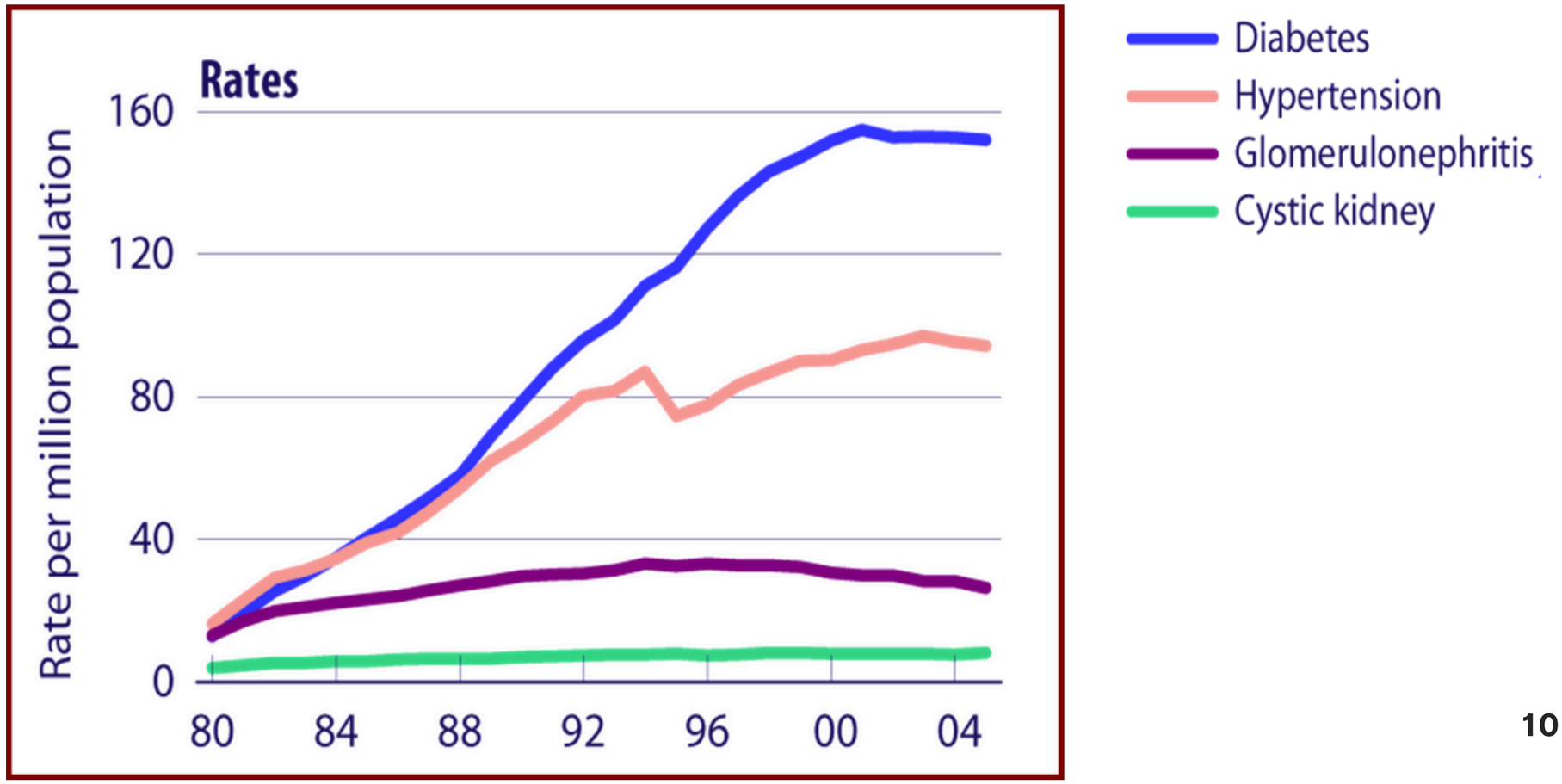
Stadiums

Renal failure can be classified according to the level of renal filtration:

Stade	Definition	DFG (ml/min/1.73 m²)
1	Chronic kidney disease with normal or increased filtration	>=90
2	Chronic kidney disease with slightly decreased filtration	60-89
3	Moderate chronic renal failure	30-59
4	Severe chronic renal failure	15-29
5	End-stage chronic renal failure	<15

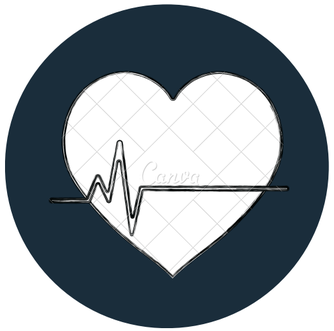
Etiology of CKD:

Diabetes and hypertension are the main causes of kidney failure:



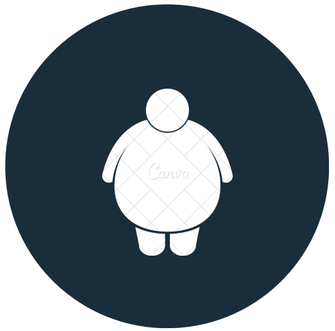
Risk factors:

The person is more likely to have CKD if:



Cardiac disease

Having a family member with kidney disease



Obesity

Upper age at 60

SYMPTOMS OF CKD

yellowish dry skin and itchy

muscle cramps

gastrointestinal bleeding

Dyspepsia: Digestive disorder

Drop

Decreased urination

Dyspnea: difficulty breathing

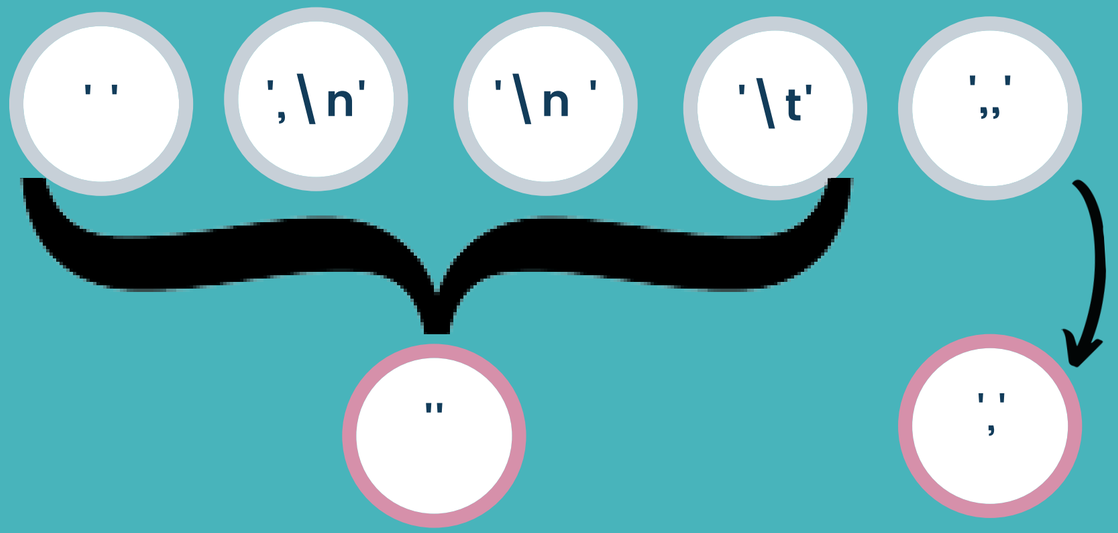
Anemia

DATA UNDERSTANDING

02

Import Data

Problems



Code

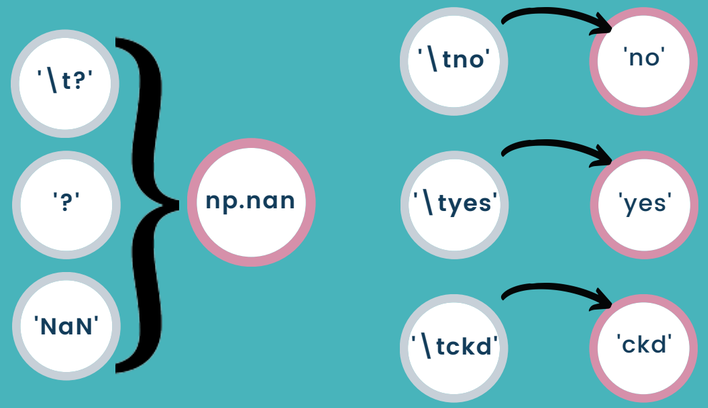
```
data = []
with open('chronic_kidney_disease.arff', "r") as f:
    for line in f:
        line = line.replace(' ', '')
        line = line.replace(',\n', '')
        line = line.replace('\n', '')
        line = line.replace('\t', '')
        line = line.replace(',,', ',')
        data.append(line.split(','))


names = ['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba',
         'bgr', 'bu',  'sc', 'sod', 'pot', 'hemo', 'pcv', 'wbcc',
         'rbcc', 'htn', 'dm', 'cad', 'appet', 'pe', 'ane',
         'class']

df = pd.DataFrame(data[29:], columns=names)
```

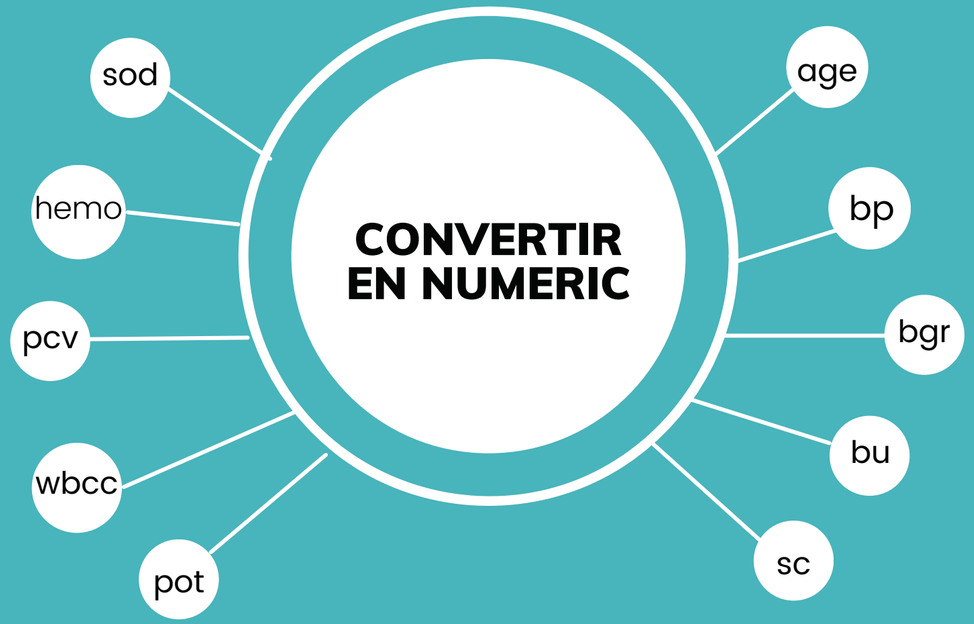
```
df.shape
#24 features and 1 target Class
```

```
(401, 25)
```

Import

Problems

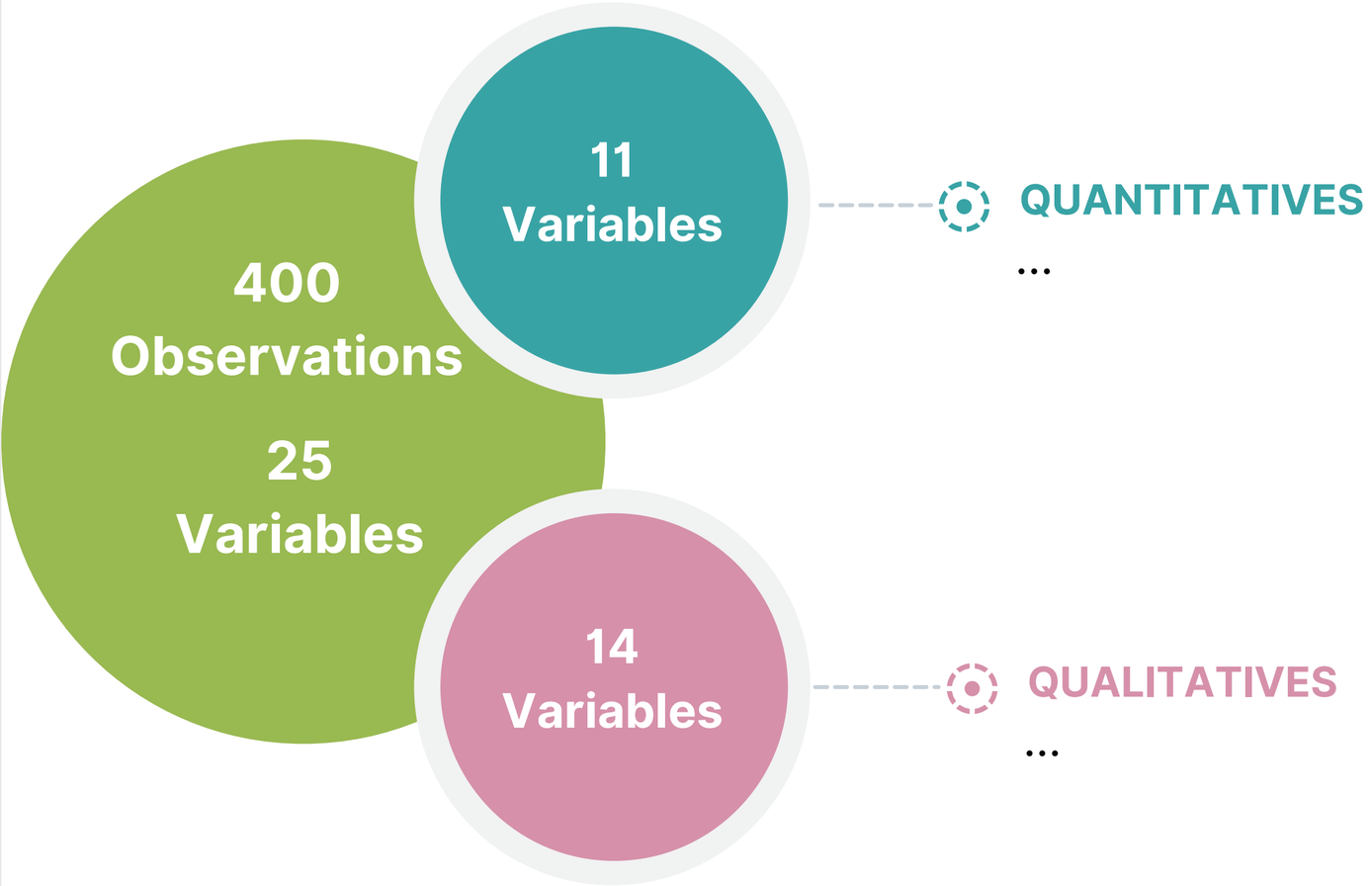




Code

```
df.replace(to_replace = {'\t?':np.nan,'?':np.nan,
                         '\tno':'no','\tyes':'yes',' yes':'yes',
                         'NaN':np.nan, 'ckd\t':'ckd'},inplace=True)
df['age']=pd.to_numeric(df['age'], errors='coerce')
df['bp']=pd.to_numeric(df['bp'], errors='coerce')
df['bgr']=pd.to_numeric(df['bgr'], errors='coerce')
df['bu']=pd.to_numeric(df['bu'], errors='coerce')
df['sc']=pd.to_numeric(df['sc'], errors='coerce')
df['sod']=pd.to_numeric(df['sod'], errors='coerce')
df['hemo']=pd.to_numeric(df['hemo'], errors='coerce')
df['pcv']=pd.to_numeric(df['pcv'], errors='coerce')
df['wbcc']=pd.to_numeric(df['wbcc'], errors='coerce')
df['rbcc']=pd.to_numeric(df['rbcc'], errors='coerce')
df['pot']=pd.to_numeric(df['pot'], errors='coerce')
```

Importing data:




```
df.dtypes.value_counts()
```

```
object    25
dtype: int64
```

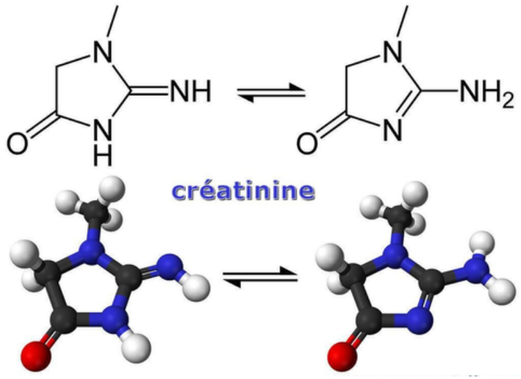
```
df.dtypes.value_counts()
```

```
object     14
float64    11
dtype: int64
```

Previews:

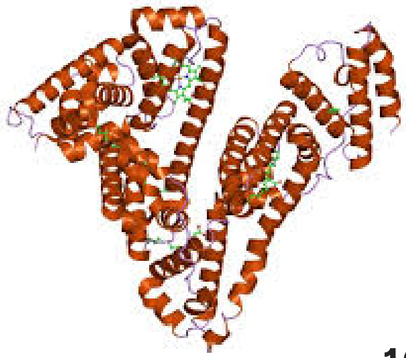
1. Serum Creatinine (SC):

- Creatinine is a waste product in the blood that comes from your muscles.
- Serum creatinine level indicates how well your kidneys are functioning.
- A normal result is:
  - Men: 0.7 to 1.3 mg/dL (61.9 to 114.9 µmol/L)
  - Women: 0.6 to 1.1 mg/dL (53 to 97.2 µmol/L)



2. Albumin (AL):

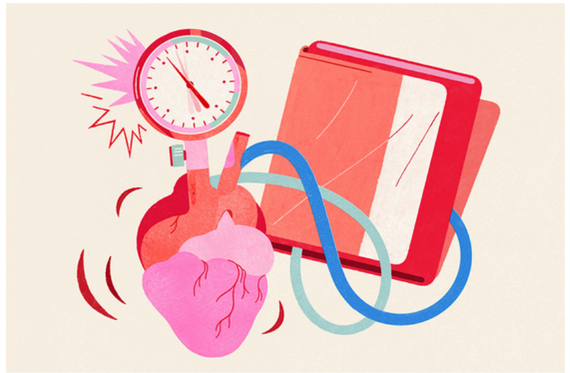
- Albumin is a protein present in blood plasma.
- The normal level of albumin in the blood between 3.5 and 5.5 g/dL



Previews:

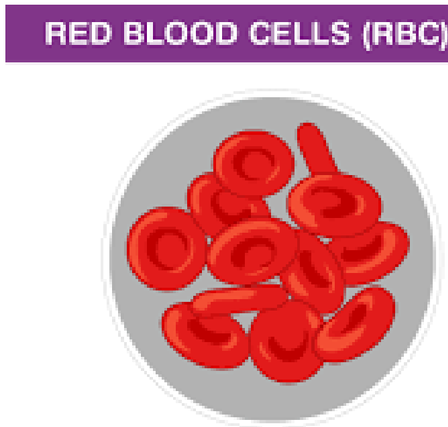
3. Blood Pressure (BP):

- Blood pressure is the force exerted by circulating blood against the walls of the arteries.
- Blood pressure :Ideal: between 90/60 mmHg et 120/80 mmHg.
    - High: > 140/90 mmHg.
    - Low : < 90/60mmHg.



4. Red Blood Cells (RBC):

- Red blood cells carry fresh oxygen throughout the body.
- A normal red blood cell count is around:
    - Men: 4,0 à 5,9 x 10*12/L
    - Women: 3,8 à 5,2 x 10*12/L

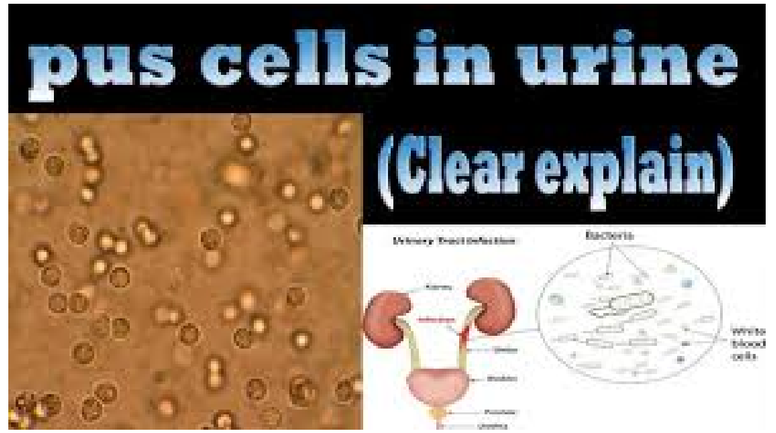


Previews:

5. Pus cells (PC) :

- Pus, formed from a mixture of dead cells (bacteria, white blood cells)
- The sighting of pus in urine is known as Pyuria.
- The presence of pus cells in urine:
    - 0-5: normal range
    - 8-10: pus cells suggest bacterial infection

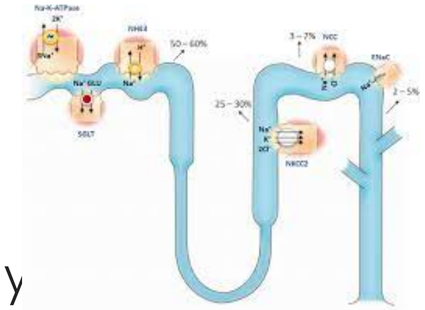
    => a urinary tract infection (UTI)



6. Blood Urea (BU) :

- The liver produces urea in the urea cycle as a waste product of protein digestion.
- A normal adult's blood should contain 6 to 20 mg/dL
- (2.1 to 7.1 mmol/L) urea nitrogen.

Previews:



7. Sodium (SOD) :

- Sodium is a mineral that plays an important role in the hydration state of the body
- Sodium is filtered by the kidneys,
- Excess sodium can be dangerous for people with kidney disease because your kidneys cannot remove excess sodium and fluid from your body.
- CDK => this elimination may decrease.

8. Potassium (POT) :

- When the kidneys fail, they can no longer eliminate excess potassium, which therefore accumulates in the body.
- Your blood potassium level is normally between 3.6 and 5.2 millimoles per liter (mmol/L).

Previews:

9. Hypertension (HTN):

- Hypertension is when blood pressure is too high.
- Hypertension is diagnosed when it is measured on two different days:
    - systolic blood pressure (the first number): ≥140 mmHg.
    - diastolic blood pressure (the second number): ≥90 mmHg

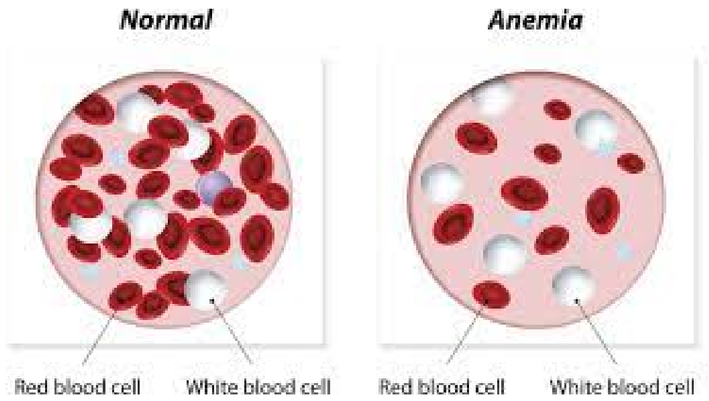
10. Specific Gravity (SG):

- It is a commonly used urinalysis parameter in the assessment of kidney function.
- It indicates the concentration of all chemical particles in urine.
- The normal specific gravity of urine is 1.003.

Previews:

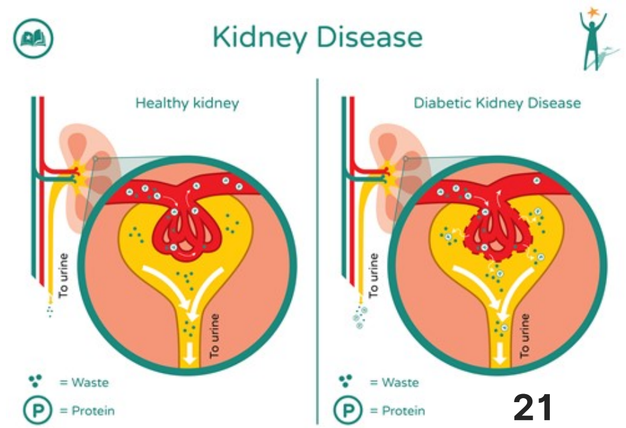
11. Anemia (ANE) :

- Anemia corresponds to an abnormal drop in the level of hemoglobin in the blood.
- Kidney failure can also cause anemia, as the kidneys secrete erythropoietin, the hormone that stimulates the production of red blood cells.



12. Diabetes Mellitus(DM) :

- Diabetes is the leading cause of chronic kidney disease.
- High blood sugar due to diabetes can cause damage inside your kidneys.
- Controlling blood sugar helps reduce the risk of kidney disease.



Previews:

13. blood glucose random (BGR) :

- This is a measurement of blood sugar at the time you are tested.
- A blood sugar level of 200 mg/dL or higher indicates that you have diabetes.

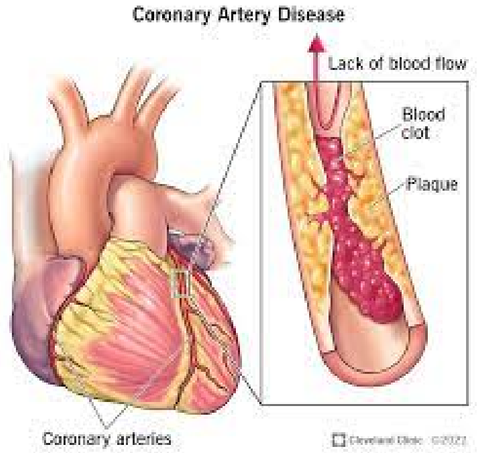
14. Bacteria (BA) :

- Kidney infections are usually caused by bacteria that spread to the kidneys from the urinary tract
- Increased urea concentration during CKD leads to increased risk of infection

Previews:

15. Coronary Artery Disease (CAD) :

- The main blood vessels that supply the heart (coronary arteries) struggle to send enough blood, oxygen and nutrients to the heart muscle.
- Cholesterol deposits in the arteries of the heart and inflammation are usually the cause of coronary heart disease.



16. Appetite (APPET) :

- Appetite is a person's desire to eat food. It is distinguished from hunger, which is the body's biological response to a lack of food. A person may have an appetite even if their body does not show signs of hunger, and vice versa.

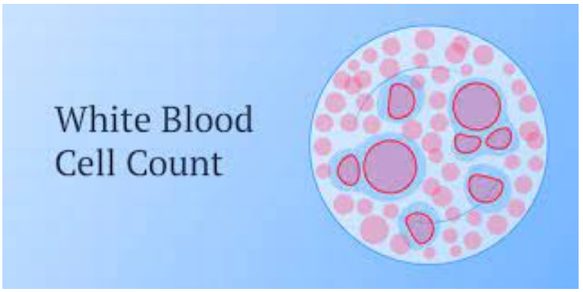
Previews:

17. Pedal Edema (PE) :

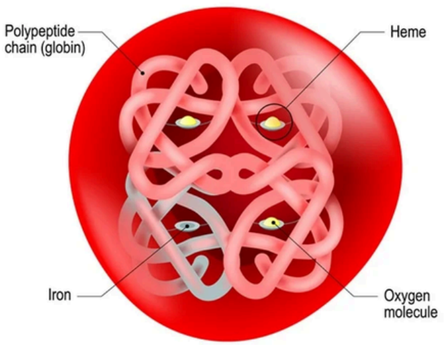
- Pedal edema causes an abnormal buildup of fluid in the ankles, feet, and lower legs, leading to swelling in the feet and ankles.
- Edema can be caused by medications, pregnancy, infections, and many other medical problems. Edema occurs when small blood vessels leak fluid into nearby tissues.

18. white blood cells (WC) :

- White blood cells, also called leukocytes, are responsible for protecting your body against infections. As part of your immune system, white blood cells circulate in your blood and respond to injury or illness.
- The normal number of leukocytes in the blood is 4,500 to 11,000 leukocytes per microliter (4.5 to 11.0 × 109/L).

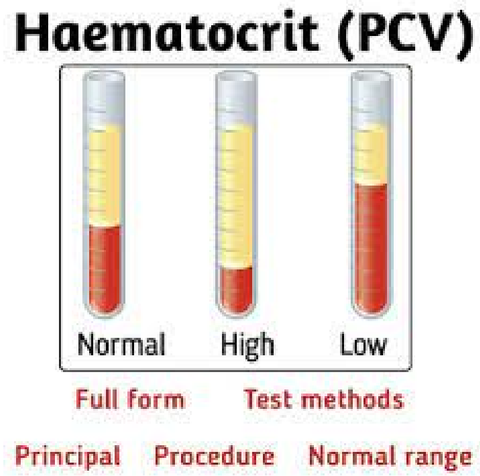


Previews:



19. Hemoglobin (HEMO) :

- Hemoglobin is the protein that allows red blood cells to carry oxygen from your lungs to the rest of your body. If there are fewer red blood cells or less hemoglobin, your tissues and organs, such as your heart and brain, may not receive enough oxygen to function properly.



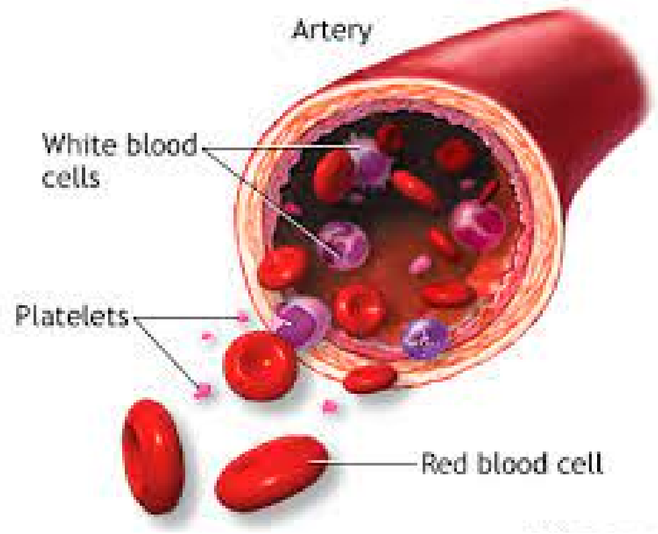
20. Packed Cell Volume (PCV) :

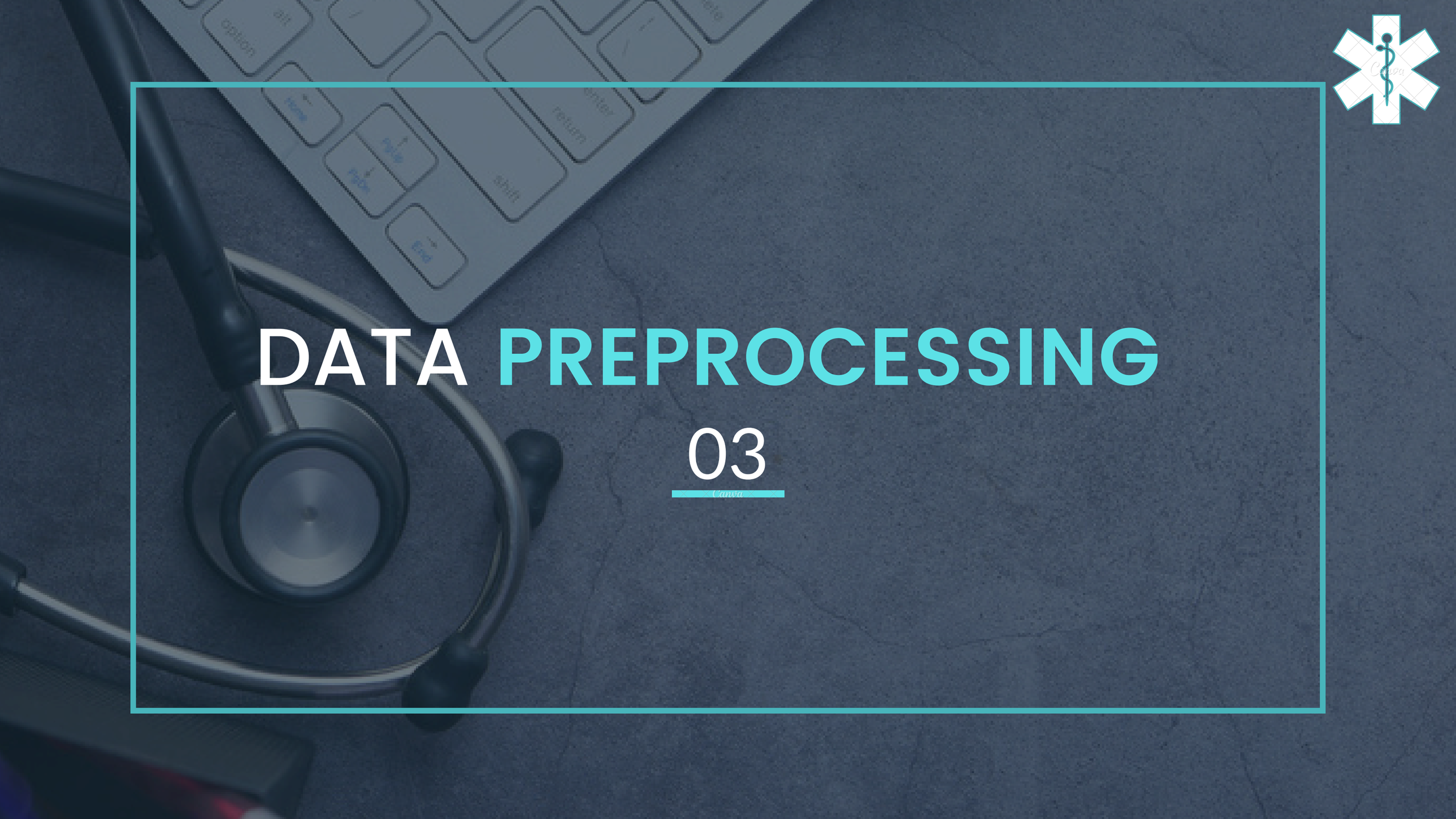
- PCV, reticulocyte count, and reticulocyte index appear to worsen with increasing severity of kidney disease. There is a need to provide erythropoietin for evidence-based management of anemia in renal patients.

Previews:

21. Red Blood Cell Count （RC）:

- Red blood cell, cellular component of blood, millions of which in the circulation of vertebrates give blood its characteristic color and carry oxygen from the lungs to tissues.
- The normal range is generally considered to be:
    - for men: 4.35 to 5.65 million red blood cells per microliter of blood
    - for women: 3.92 to 5.13 million red blood cells per mcL of blood
    - For children: The threshold for a high red blood cell count varies depending on age and gender.

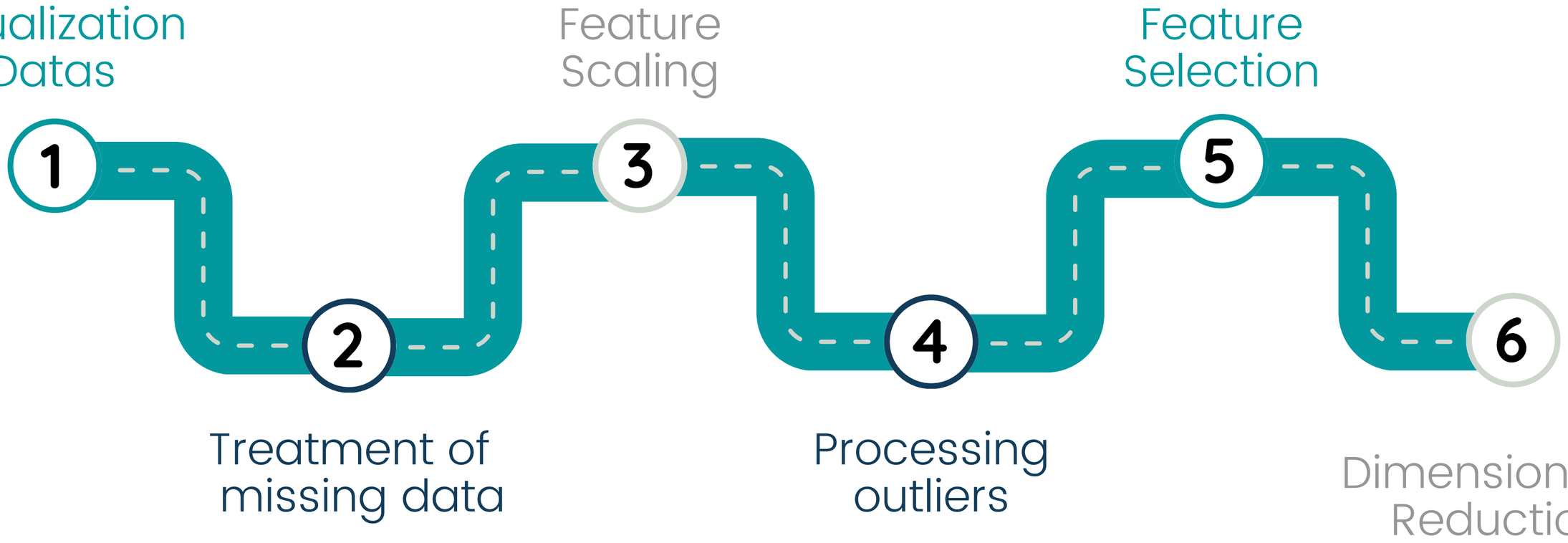


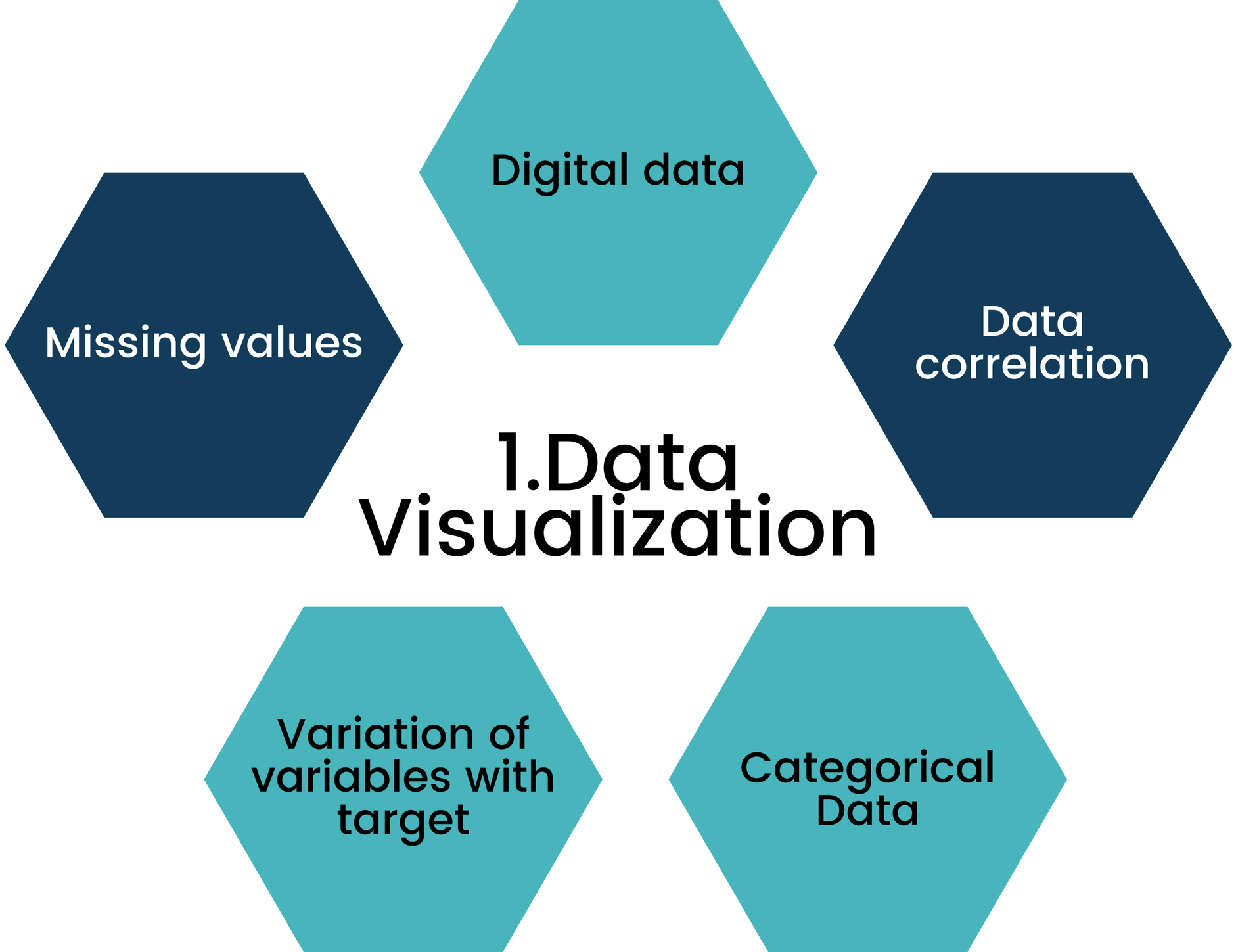


DATA PREPROCESSING

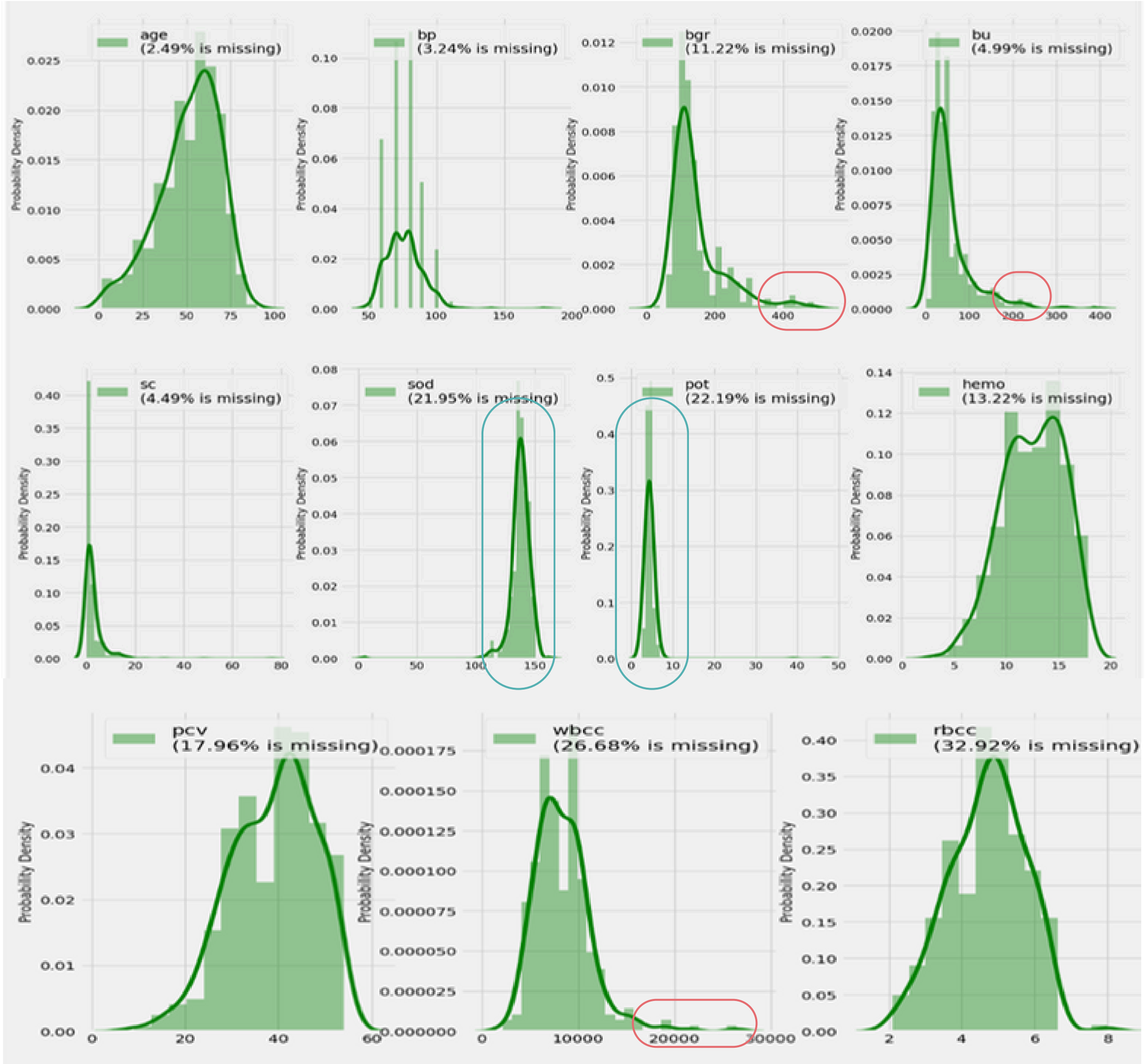
03

PLAN

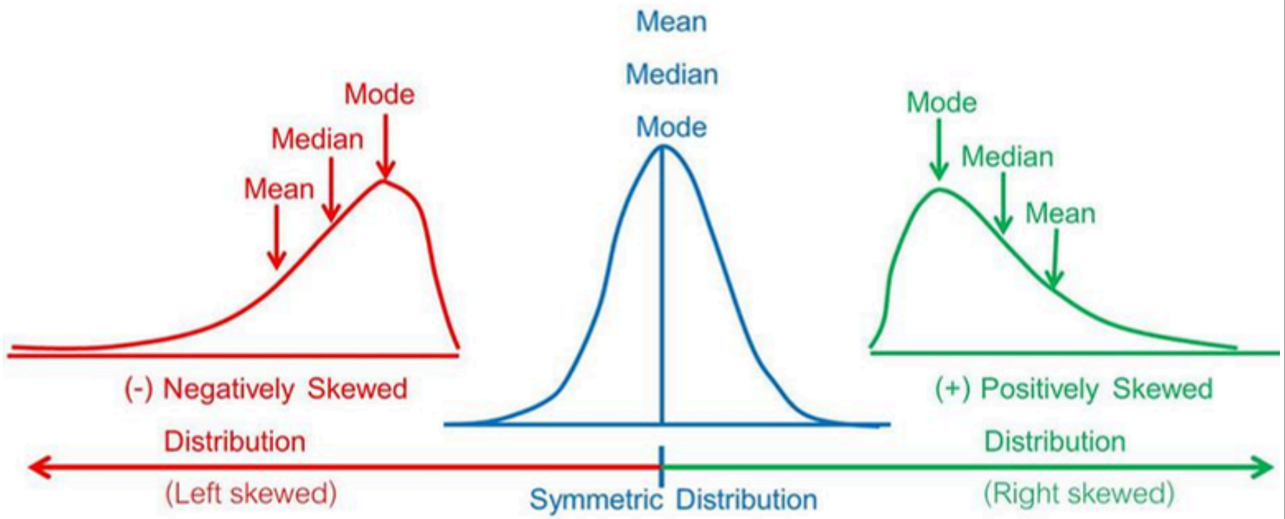

Visualization Datas
1
Treatment of missing data
2
Feature Scaling
3
Processing outliers
4
Feature Selection
5
Dimensionality Reduction
6


Digital data
Missing values
Data correlation
1.Data Visualization
Variation of variables with target
Categorical Data

Digital data



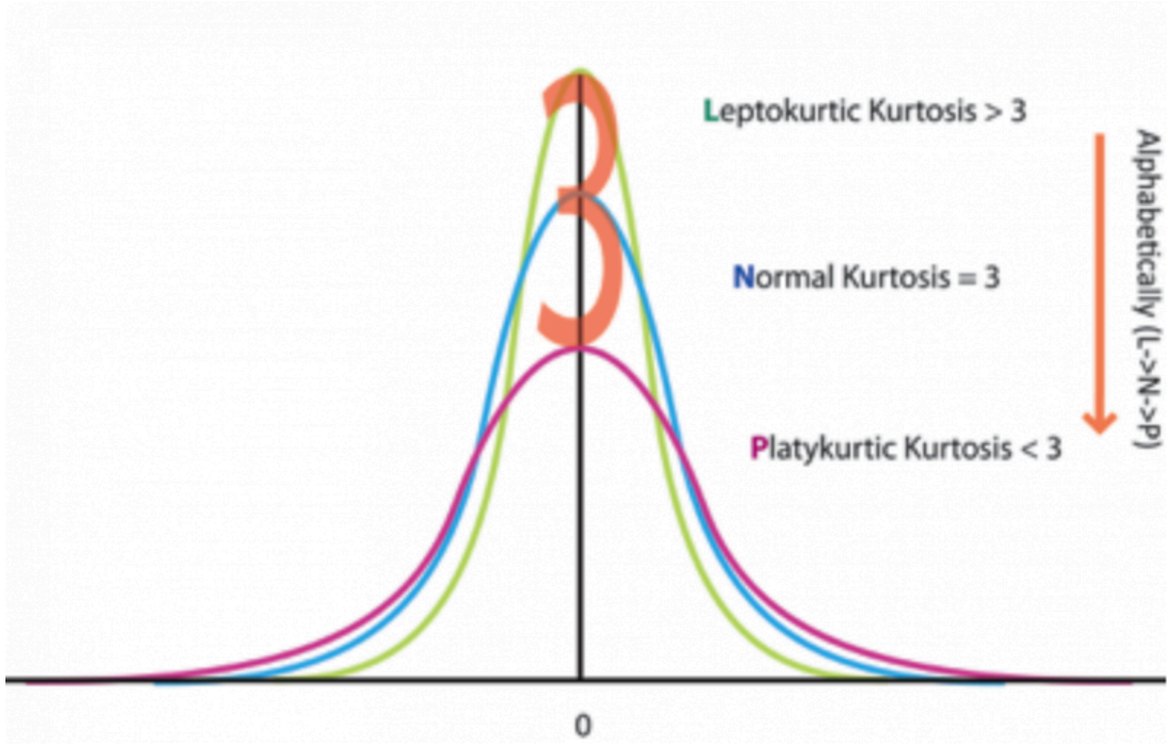
Asymmetry coefficient : Skewness




```python
#Skewness of each numerical feature
for col in numerical_features:
    print(f"{col} has {df[col].skew()} values\n")
```

```
age has -0.6682594691593559 values

bp has 1.6054289569770592 values

bgr has 2.010773172514955 values

bu has 2.6343744585903863 values

sc has 7.509538252140634 values

sod has -6.996568560937047 values

pot has 11.582955561754394 values

hemo has -0.3350946791593011 values

pcv has -0.4336785974434392 values

wbcc has 1.621589371911243 values

rbcc has -0.1833293207517324 values
```

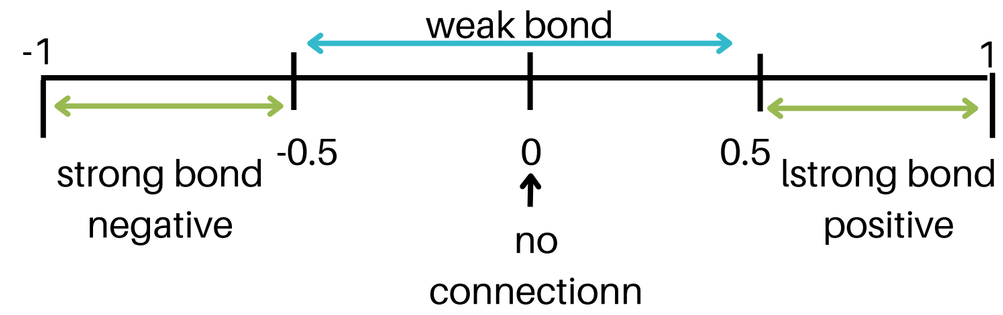
Flattening coefficient: kurtosises

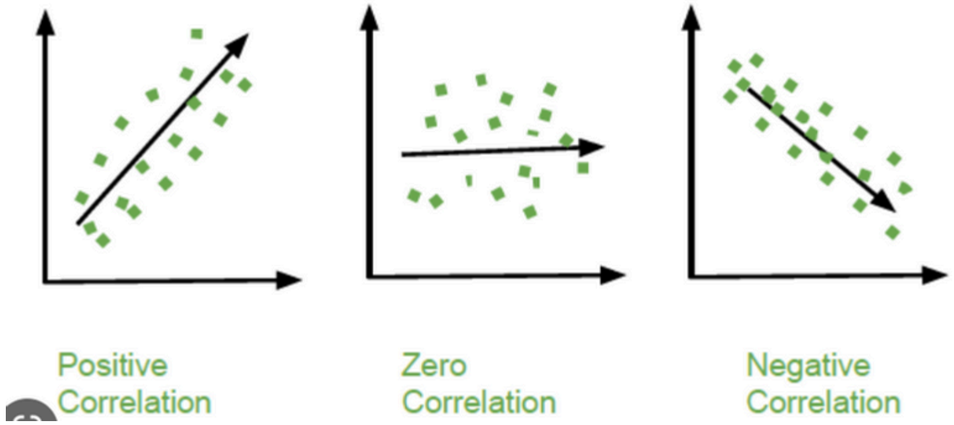


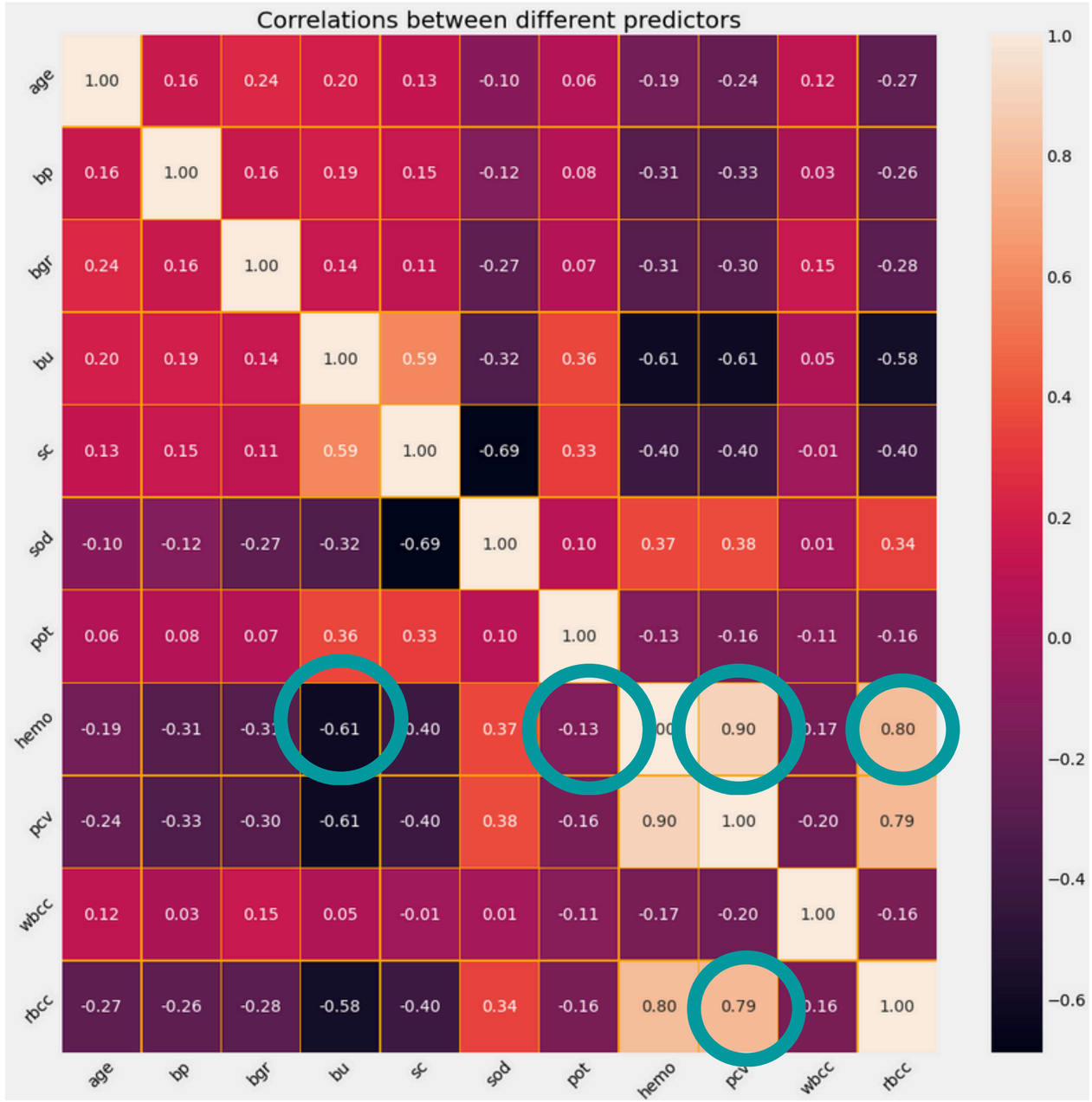

```python
#kurtosis of each numerical values
for col in numerical_features:
    print(f"{col} has {df[col].kurt()} values\n")
```

```
age has 0.0578404946021962 values

bp has 8.646095188994593 values

bgr has 4.225593588175688 values

bu has 9.34528857643826 values

sc has 79.30434545204058 values

sod has 85.53436961995402 values

pot has 142.5059115472415 values

hemo has -0.47139804366030624 values

pcv has -0.3205616837789842 values

wbcc has 6.150639815202153 values

rbcc has -0.27004248129474906 values
```

Data correlation

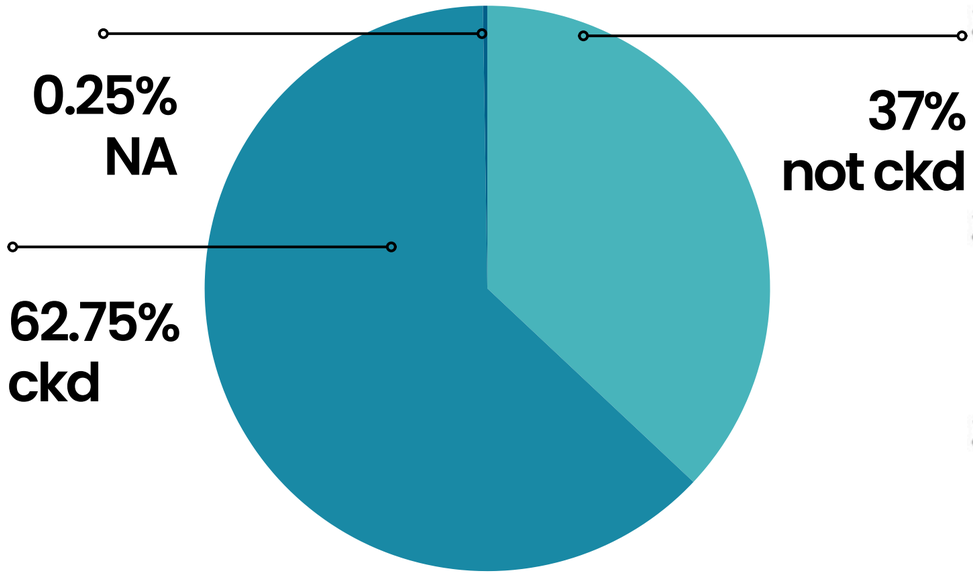




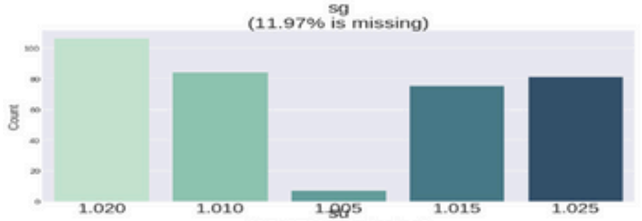


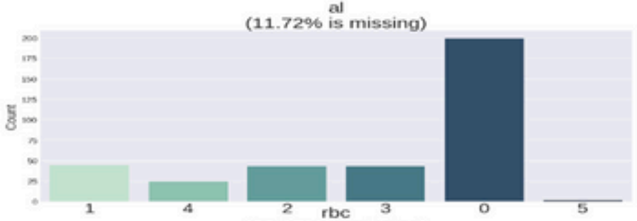
Categorical data

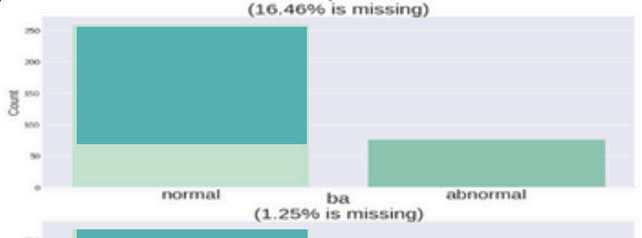
Repartition

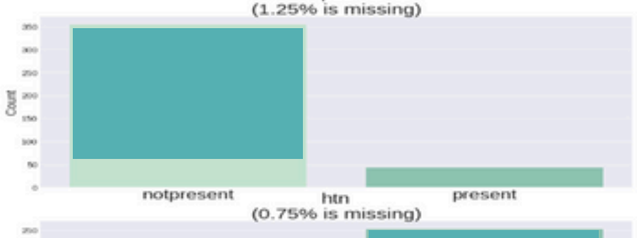


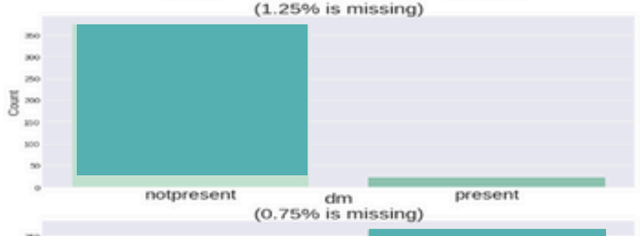
- **UNBALANCED DATA**

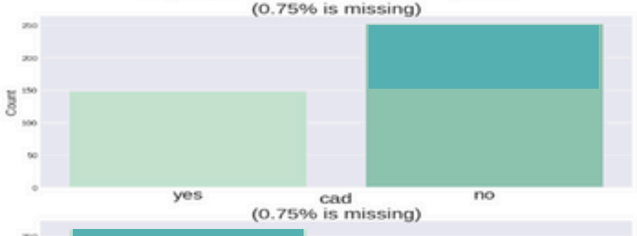


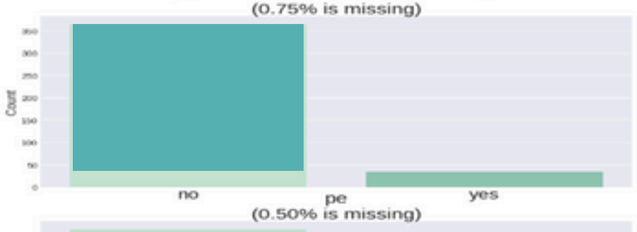


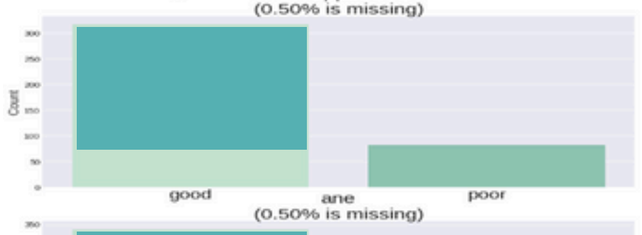


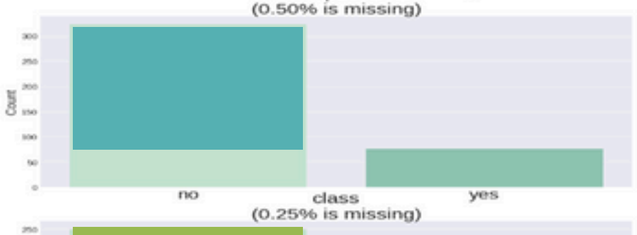


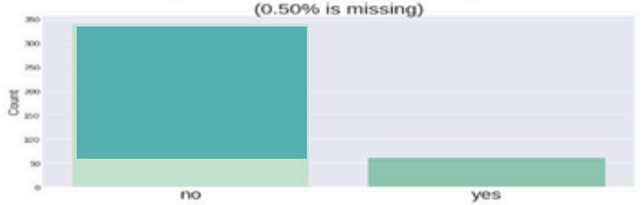


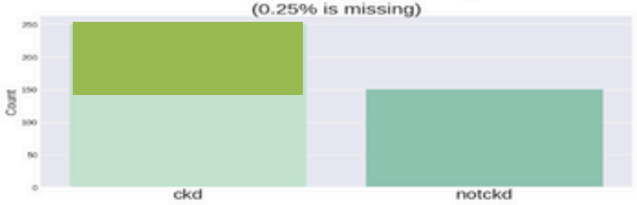




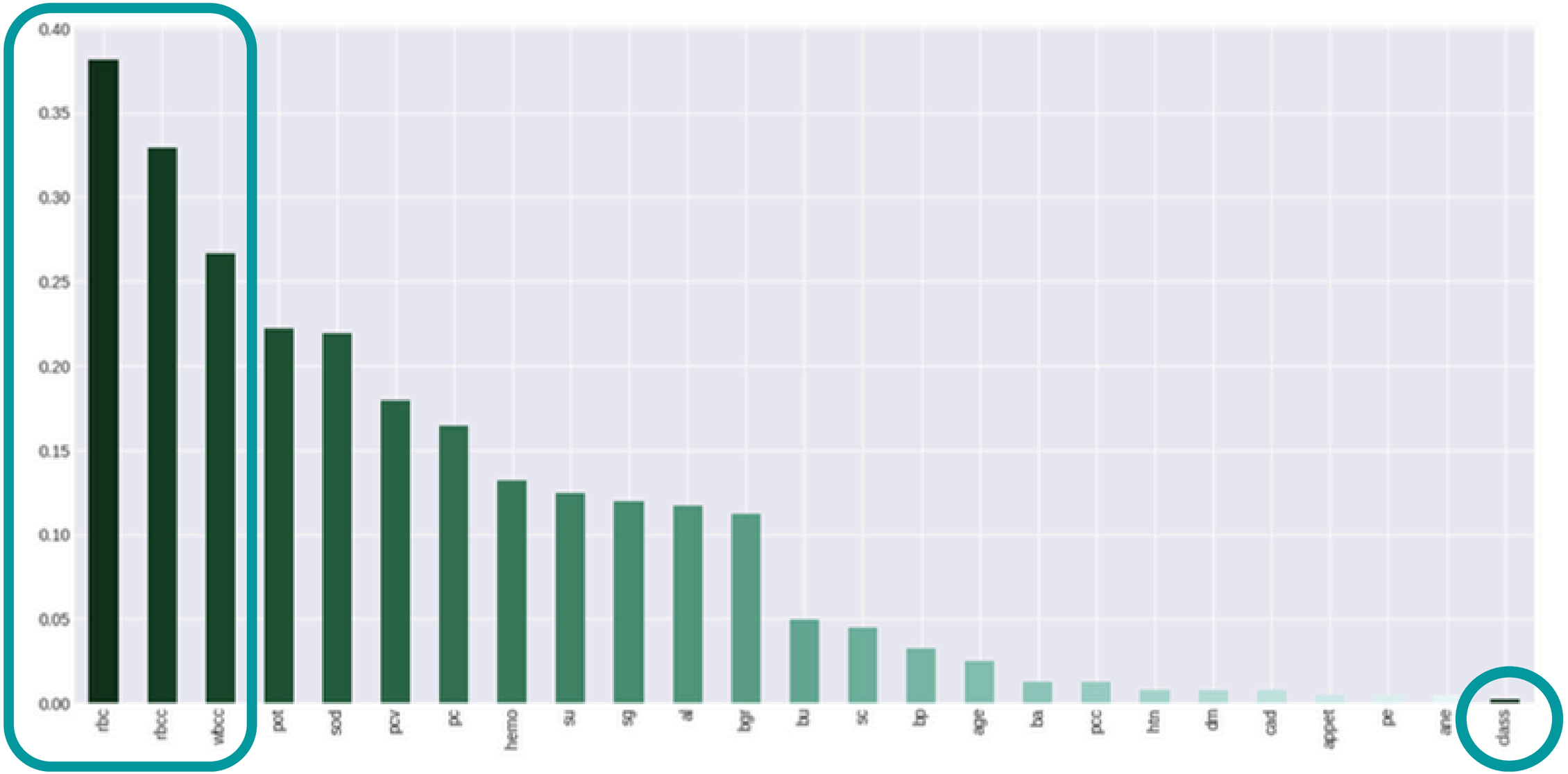








Percentages of missing values

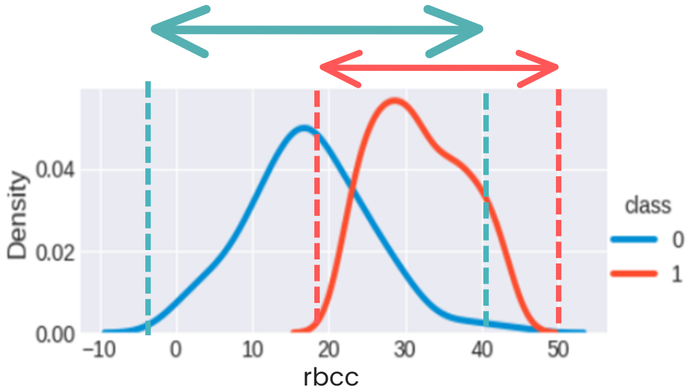


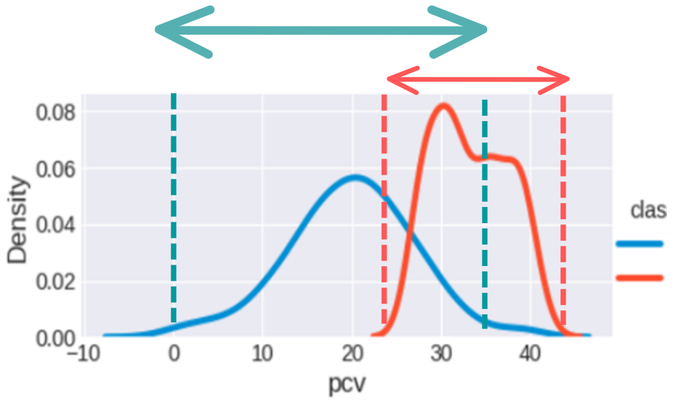
Variation of each feature VS Target

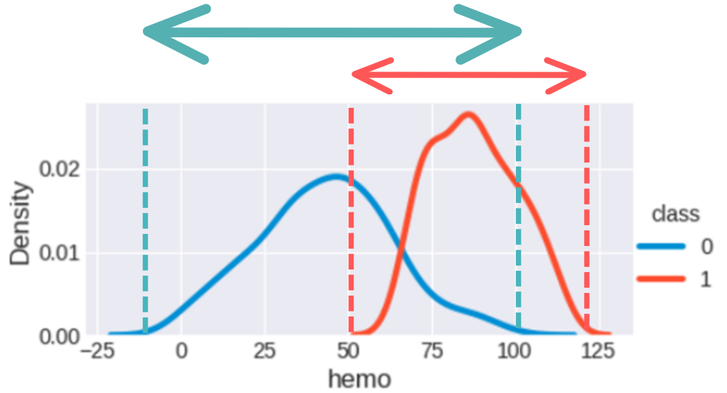
RBCC, PCV, Hemoglobin

ckd

notckd







class	mean	median
0	17.357143	17.0
1	31.790210	31.0

class	mean	median
0	19.961749	20.0
1	33.335616	33.0

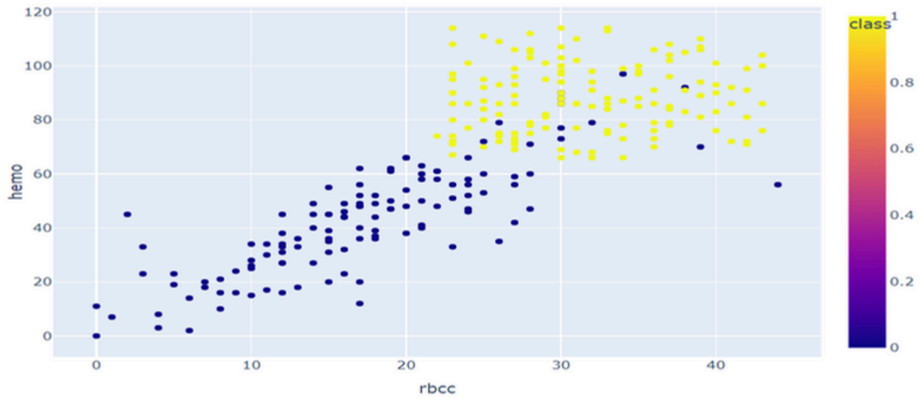
class	mean	median
0	19.961749	20.0
1	33.335616	33.0

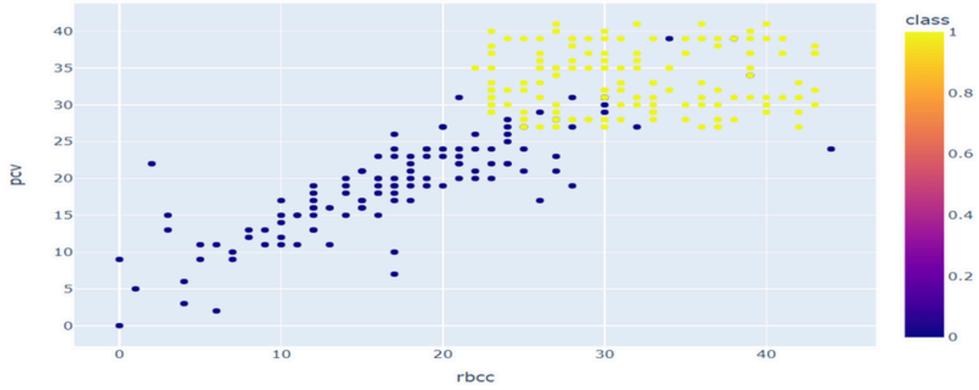
hemo=f(rbcc)

pcv=f(rbcc)

ckd

not ckd



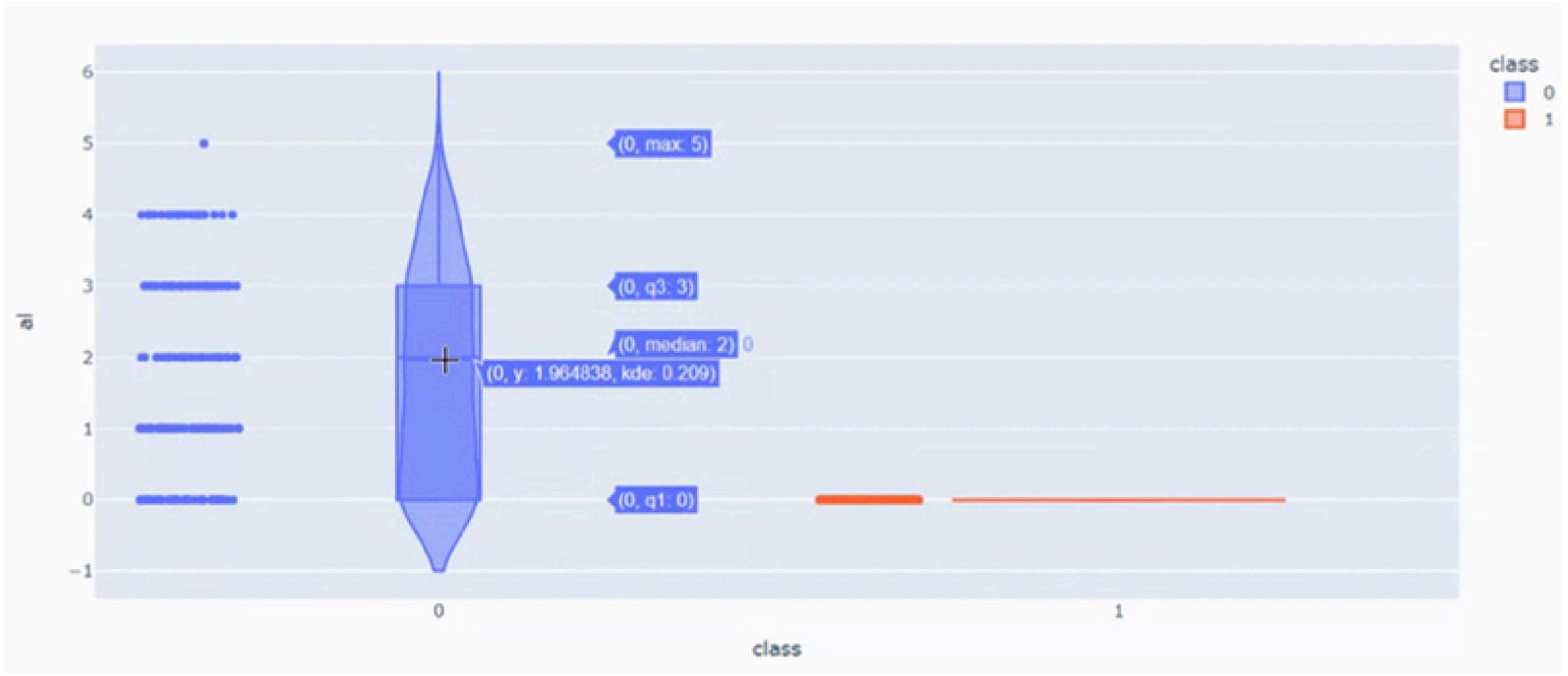


Variation of each feature VS Target

Albumin

ckd

notckd

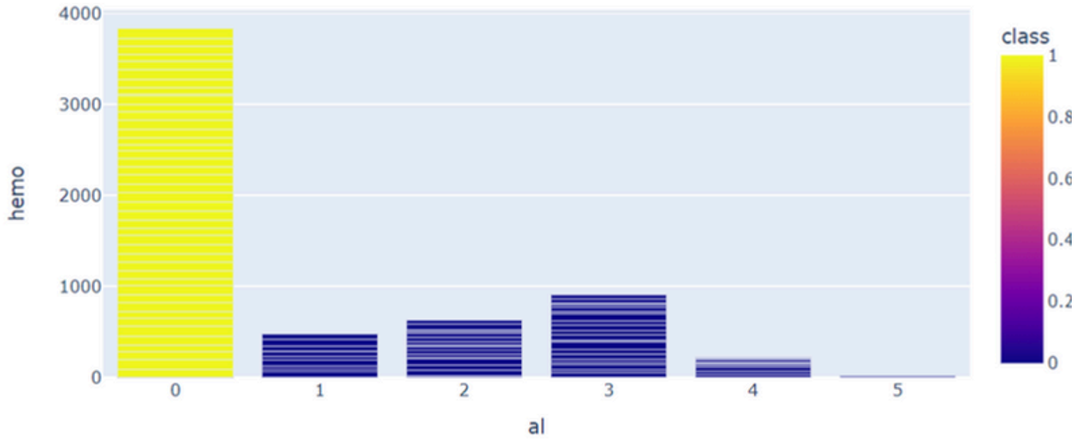


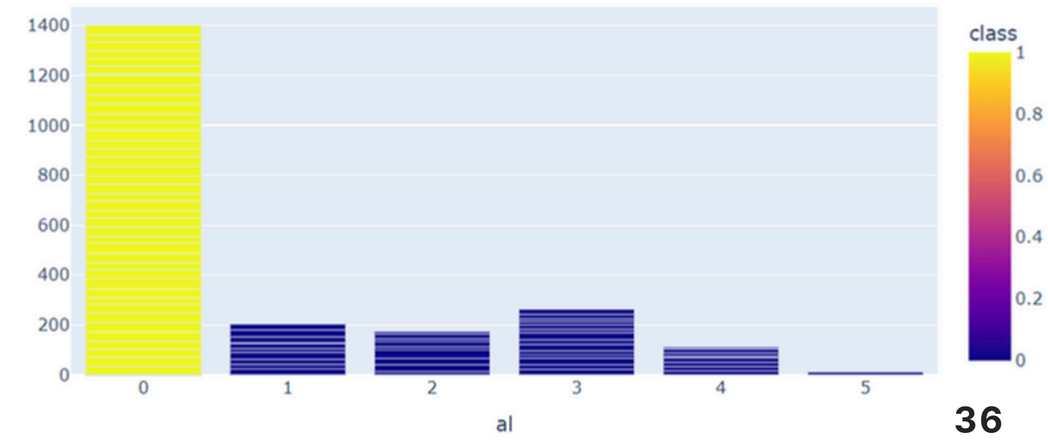
hemo=f(al)

pcv=f(al)

ckd

notckd





Variation of each feature VS Target

Specific Gravity

ckd

notckd

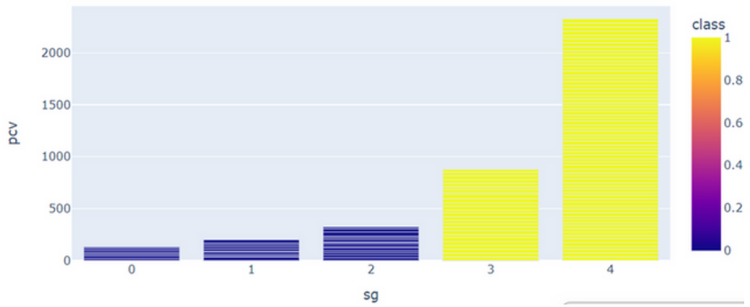
pcv=f(sg)

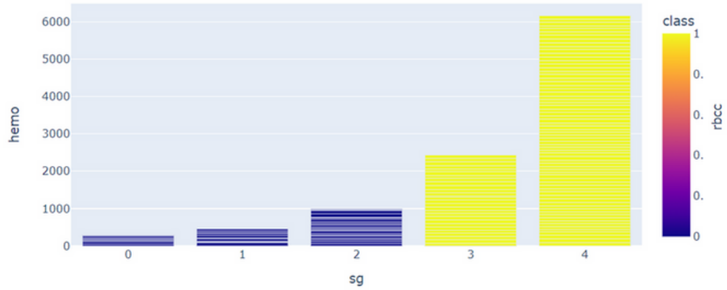
hemo=f(sg)

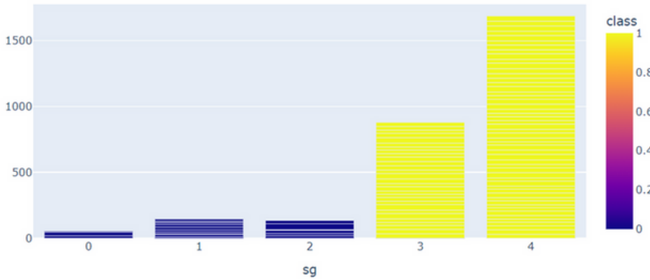
rbcc=f(sg)

ckd

notckd





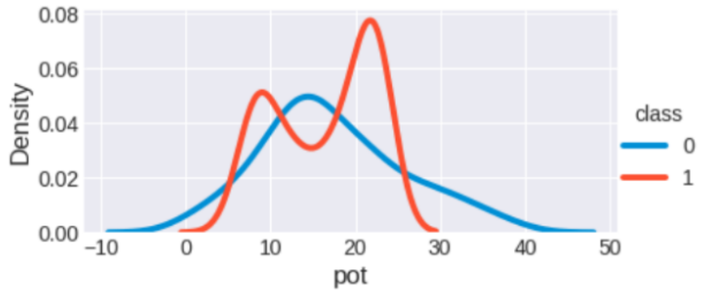


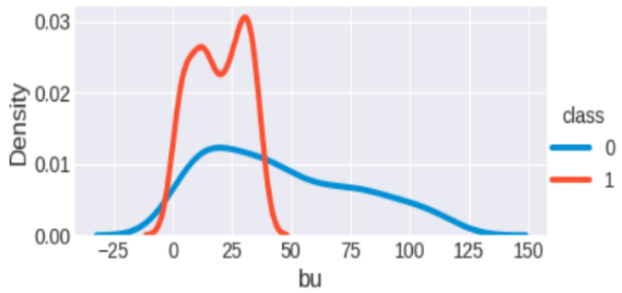
Variation of each feature VS Target

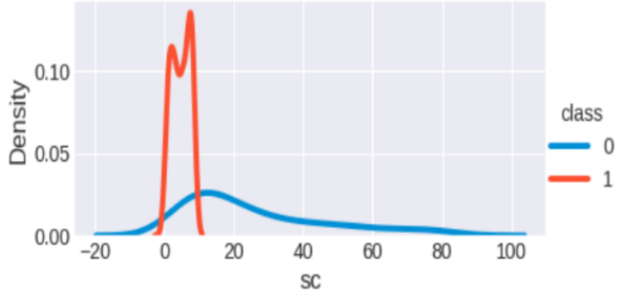
Potassium, blood urea and serum_creatinine

ckd

notckd



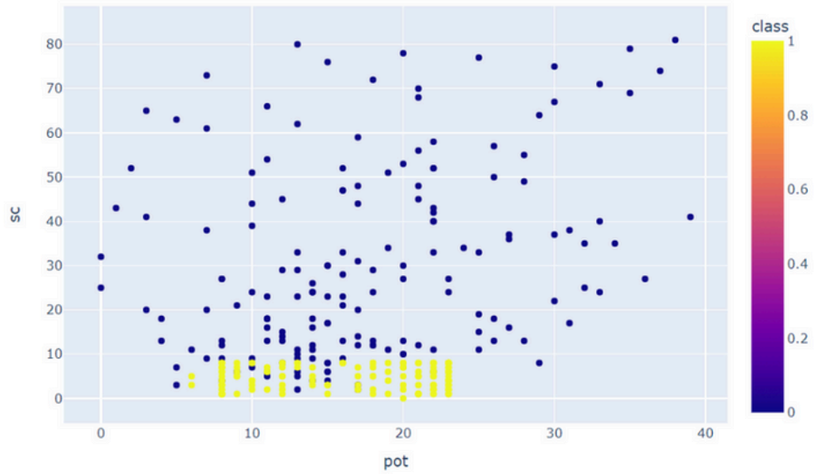




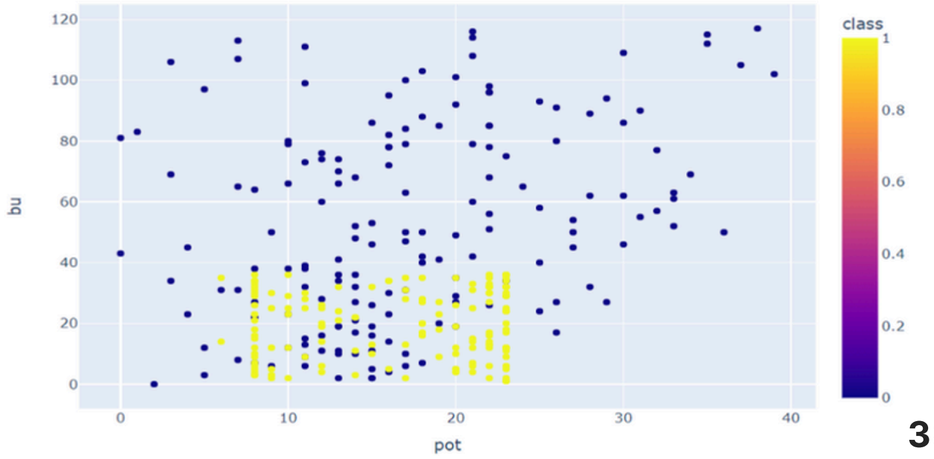
sc=f(pot)

ckd

notckd



bu=f(pot)

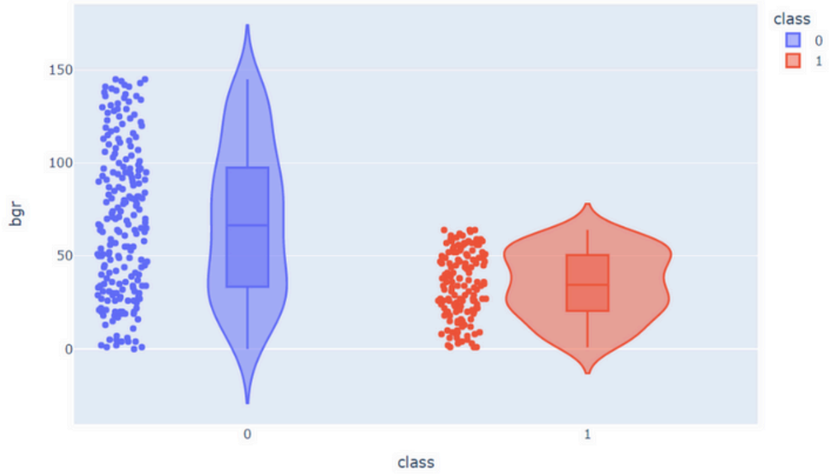


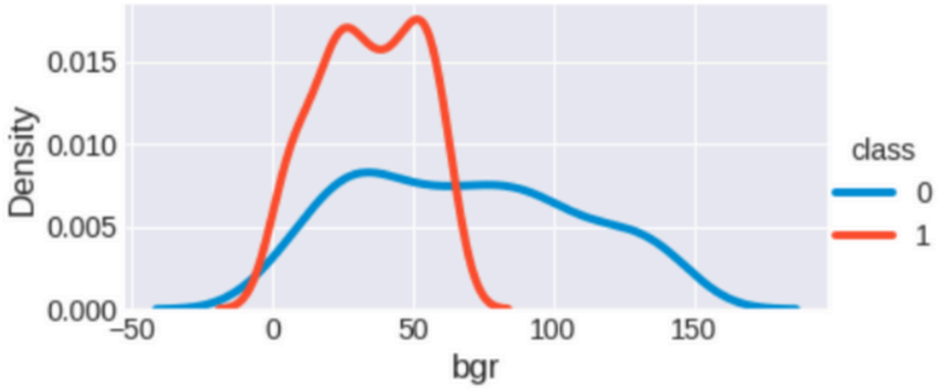
Variation of each feature VS Target

blood glucose random, sugar, diabetes_mellitus

ckd

notckd

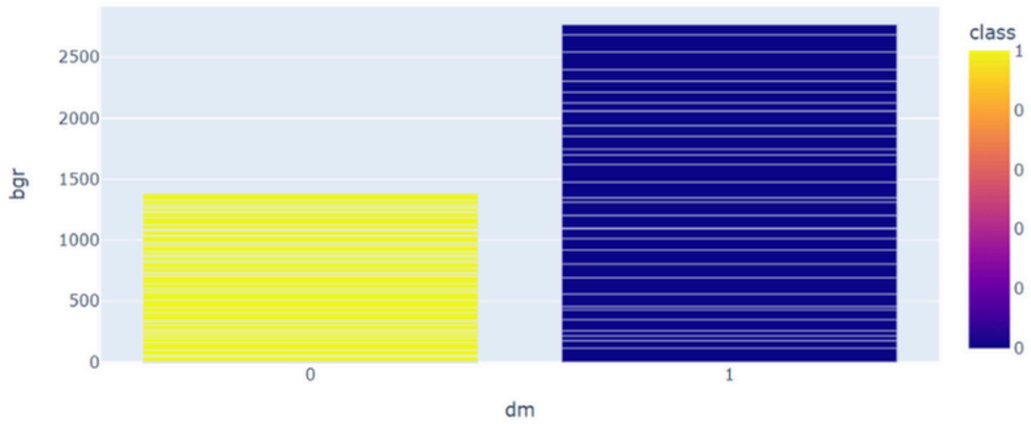




dm=f(bgr)

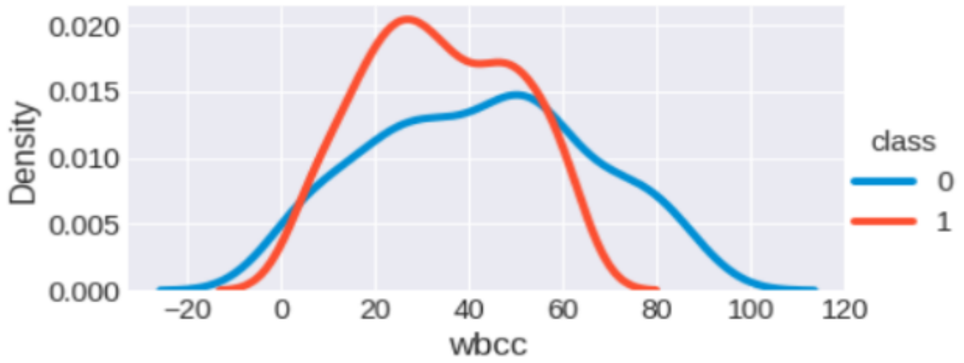
ckd

notckd

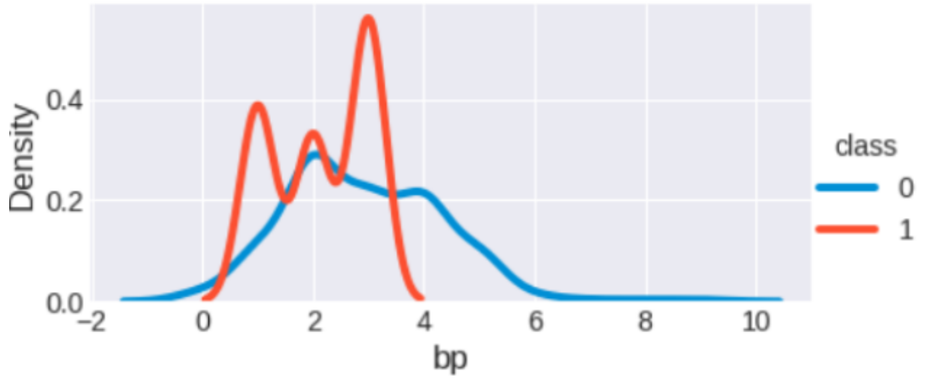


Variation of each feature VS Target

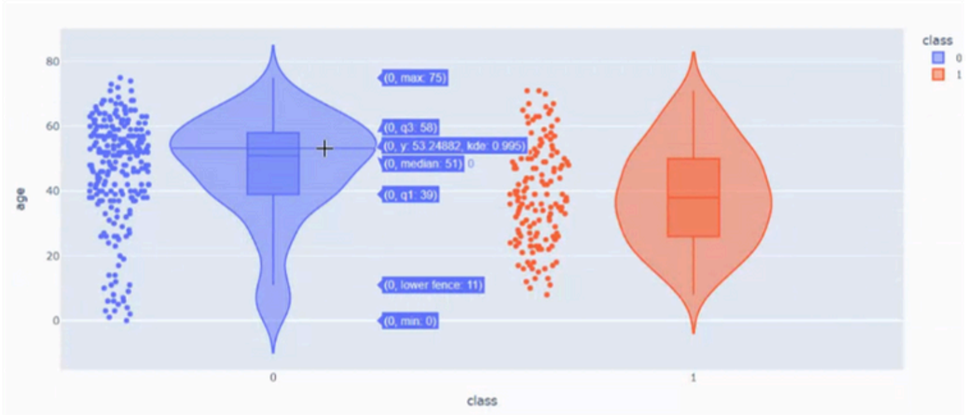
White blood cell count :



Blood pressure



Age



ckd

not ckd

2-Encoding of categorical data

No hierarchy

Most categorical features only have 2 values

No order

age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wbcc	rbcc	htn	dm	cad	appet	pe	ane	class
48.0	80.0	1.020	1	0	NaN	normal	notpresent	notpresent	121.0	...	44.0	7800.0	5.2	yes	yes	no	good	no	no	ckd
7.0	50.0	1.020	4	0	NaN	normal	notpresent	notpresent	NaN	...	38.0	6000.0	NaN	no	no	no	good	no	no	ckd
62.0	80.0	1.010	2	3	normal	normal	notpresent	notpresent	423.0	...	31.0	7500.0	NaN	no	yes	no	poor	no	yes	ckd
48.0	70.0	1.005	4	0	normal	abnormal	present	notpresent	117.0	...	32.0	6700.0	3.9	yes	no	no	poor	yes	yes	ckd
51.0	80.0	1.010	2	0	normal	normal	notpresent	notpresent	106.0	...	35.0	7300.0	4.6	no	no	no	good	no	no	ckd

Label Encoding

age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wbcc	rbcc	htn	dm	cad	appet	pe	ane	class
40.0	3.0	3.0	1.0	0.0	NaN	1.0	0.0	0.0	48.0	...	31.0	35.0	30.0	1.0	1.0	0.0	0.0	0.0	0.0	0
5.0	0.0	3.0	4.0	0.0	NaN	1.0	0.0	0.0	NaN	...	25.0	19.0	NaN	0.0	0.0	0.0	0.0	0.0	0.0	0
54.0	3.0	1.0	2.0	3.0	1.0	1.0	0.0	0.0	140.0	...	18.0	33.0	NaN	0.0	1.0	0.0	1.0	0.0	1.0	0
40.0	2.0	0.0	4.0	0.0	1.0	0.0	1.0	0.0	44.0	...	19.0	25.0	17.0	1.0	0.0	0.0	1.0	1.0	1.0	0
43.0	3.0	1.0	2.0	0.0	1.0	1.0	0.0	0.0	33.0	...	22.0	31.0	24.0	0.0	0.0	0.0	0.0	0.0	0.0	0

3. Feature Scaling

DIFFERENCE IN DATA RANGE → **SUPERIORITY**

Feature Scaling

When do we need feature scaling

- K-nearest neighbors (KNN)
- K-Means
- PCA

Min-Max Scaler

```
Statistics=0.208, p=0.000
Sample does not look Gaussian (reject H0)
/usr/local/lib/python3.8/dist-packages/scipy/

p-value may not be accurate for N > 5000.
```

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

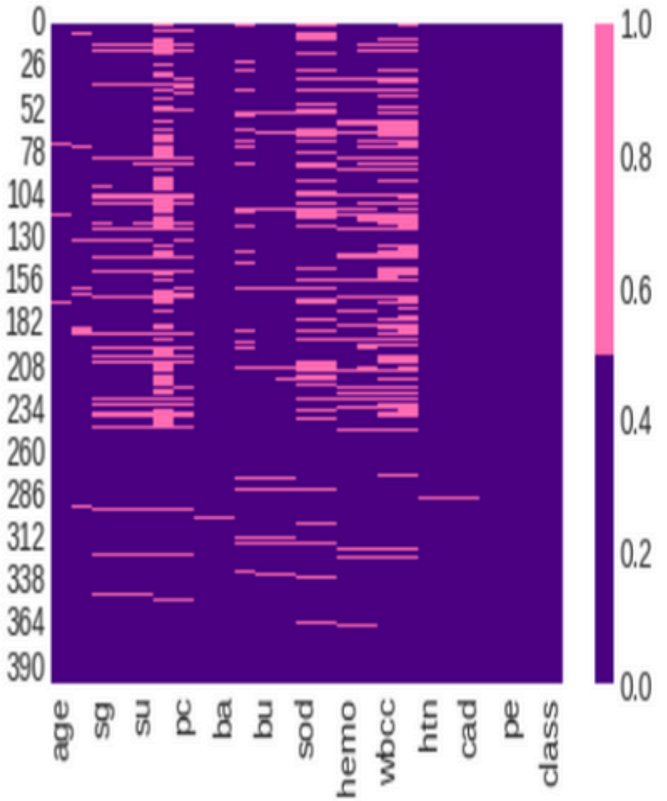
This Scaler shrinks the data in the range of -1 to 1 if there are negative values.

Non-Gaussian distribution

Sensitive to outliers

4. Handling missing values

Percentage of missing values per column




```
print(df.isnull().sum()*100/len(df))

age       0.00
bp        0.00
sg       11.75
al       11.50
su       12.25
rbc      38.00
pc       16.25
pcc       1.00
ba        1.00
bgr       0.00
bu        0.00
sc        0.00
sod       0.00
pot       0.00
hemo      0.00
pcv       0.00
wbcc      0.00
rbcc      0.00
htn       0.50
dm        0.50
cad       0.50
appet     0.25
pe        0.25
ane       0.25
class     0.00
dtype: float64
```

- We have no columns with more than 80% missing values, removing columns is not an appropria solution.
- We have a small dataset, so deleting rows with missing values is a loss of information.

IMPUTATION OF MISSING DATA

Quantitative data	Qualitative data
Mean	Mode
Median	KNN Imputer
KNNImputer	Logistic regression preceded by Simple Imputer
Iterative Imputer with ExtraTreesRegressor as an estimator	Decision Tree preceded by Simple Imputer
Linear regression preceded by Simple Imputer	

Simple Imputation

QUANTITATIVE

MEAN

age
15
Na
5
14

age
15
11
5
14

MEDIAN

age
15
Na
5
14

age
15
14
5
14

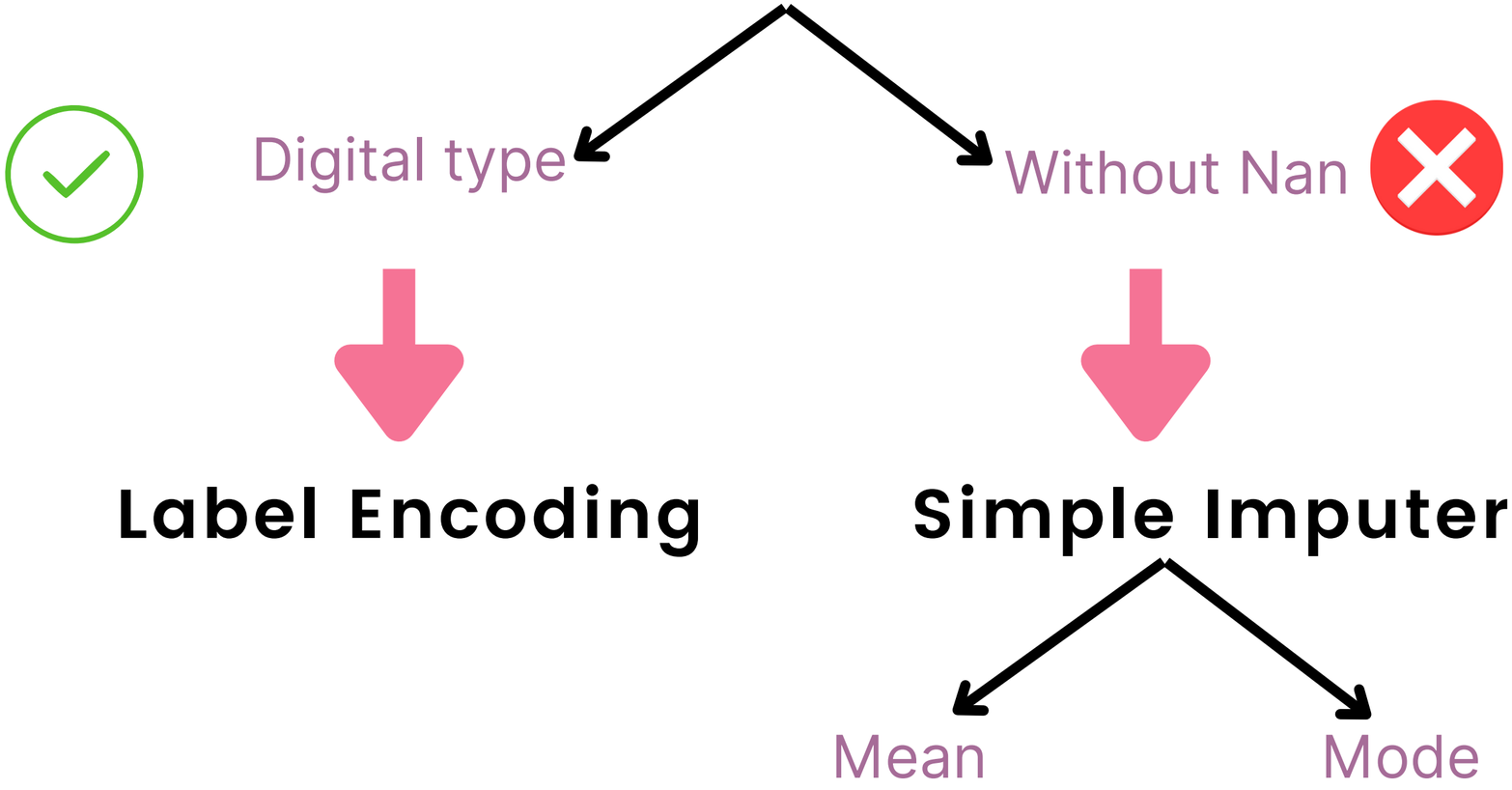
QUALITATIVE

MODE

class
ckd
Na
ckd
notckd

class
ckd
ckd
ckd
notckd

Imputation constraints with modeling algorithm



Principle Imputation with Algorithms modeling

1. Create a column f3_imp copy of f3
2. Imputation Simple(mean) missing values of f3
3. Create the model with X_train and Y_train
4. Predict NA Values

feature to be imputed (Target)

copy of f3

	f1	f2	f3	f4	f5	f3_imp
	41	A	50	211	0.9	50
	30	B	NA	100	0.7	NA
	NA	B	80	150	0.3	80
	7	A	NA	200	0.8	NA
	48	B	70	100	0.5	70

Know the NA indexes

X_train

feature to be imputed

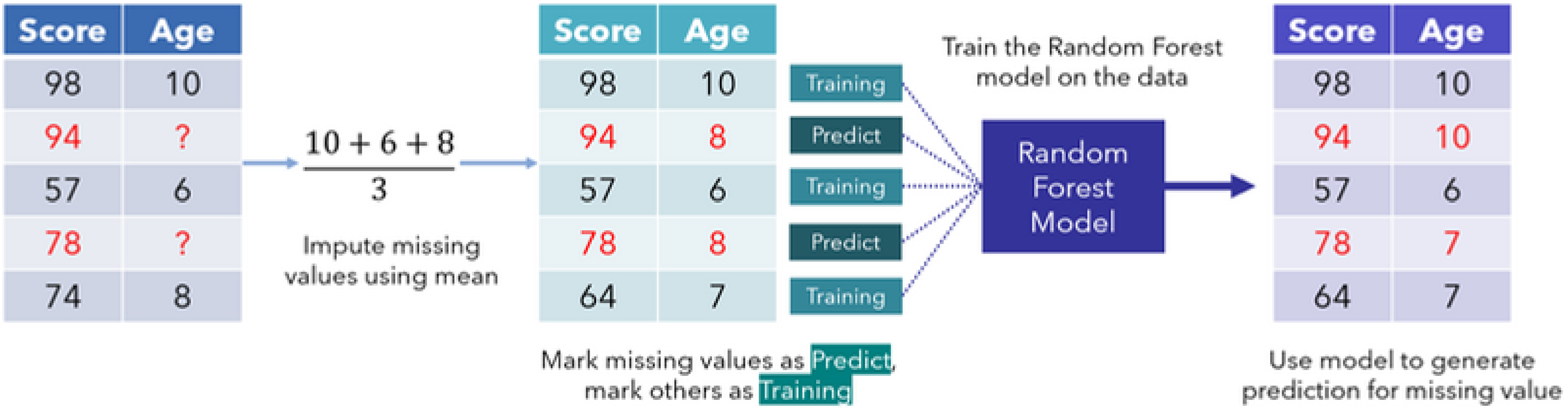
	f1	f2	f3	f4	f5	f3_imp
	41	A	50	211	0.9	50
	30	B	66.66	100	0.7	60
	NA	B	80	150	0.3	80
	7	A	66.66	200	0.8	66
	48	B	70	100	0.5	70

Y_train

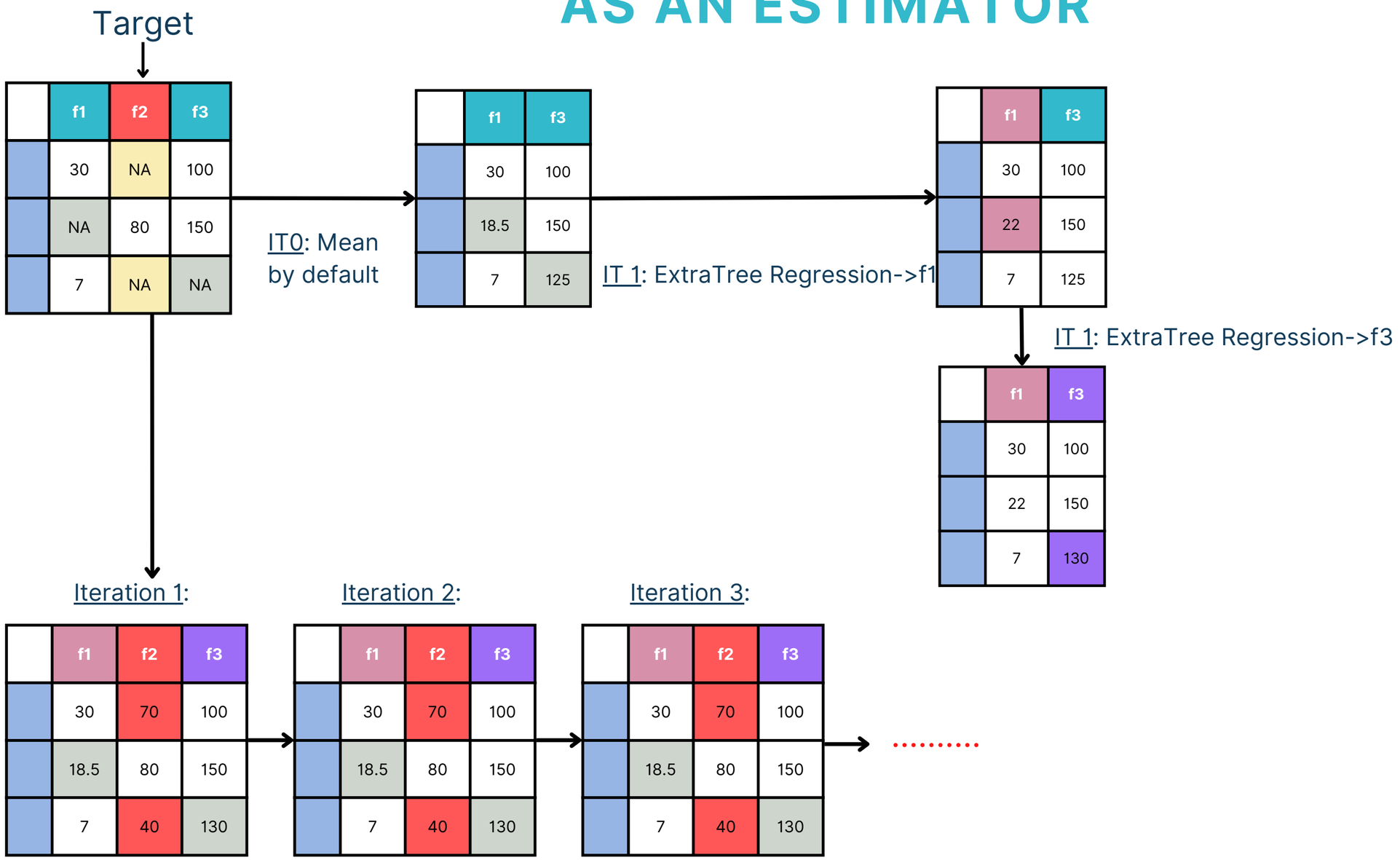
X_test

ITERATIVE IMPUTER WITH EXTRA TREES REGRESSOR AS AN ESTIMATOR

EXTRA TREES REGRESSOR = **Mean** + **Random forest**



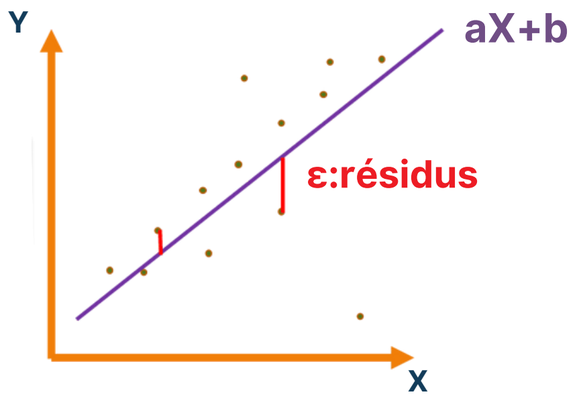
ITERATIVE IMPUTER WITH EXTRA TREES REGRESSOR AS AN ESTIMATOR



LINEAIRE REGRESSION

Simple Regression

Y=aX+b+ ε où f(X)= aX+b



Estimation of a and b by the least squares method

Multiple Regression

Y=aX+ε où f(X)= aX

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix}, X = \begin{pmatrix} 1 & x_1 \\ & \vdots \\ 1 & x_p \end{pmatrix}, A = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \text{ et } \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_p \end{pmatrix}$$

Estimate of A by the least squares method

Benefits

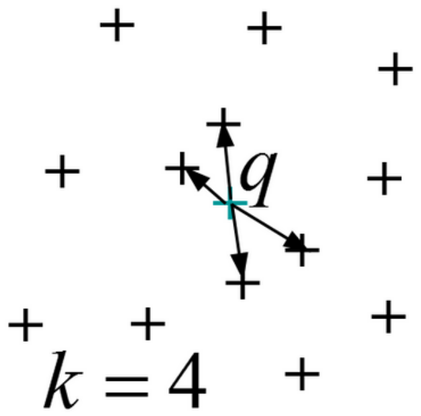
- The algorithm is simple and easy

Disadvantages

- sensitive to outliers

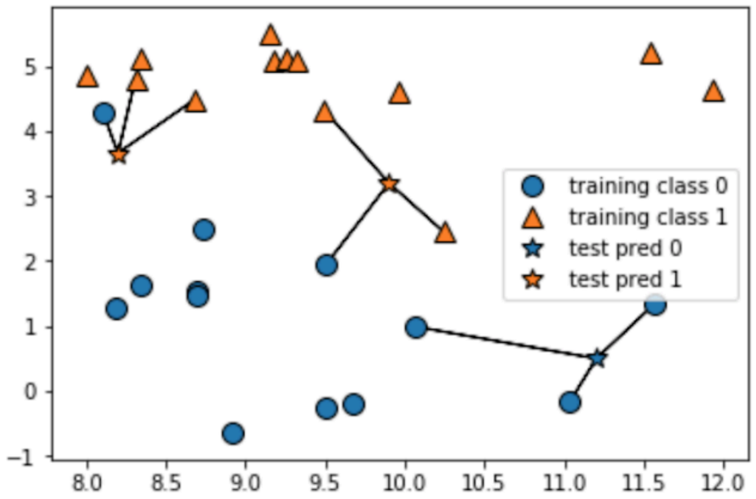
KNN IMPUTER

QUANTITATIVE



Imputation by the average of its non-zero K neighbors

QUALITATIVE



Imputation by the most frequency value of its k non-zero neighbors

Benefits

- The algorithm is simple and easy
- No assumptions about the data (linear, affine, etc.)
- The algorithm is versatile
- It can be used for:
  -classification
  -regression.

Disadvantages

- The algorithm becomes much slower as the number of training examples increases.
- The choice of the method of calculating the distance as well as the number of neighbors K may not be obvious

Simple Imputation

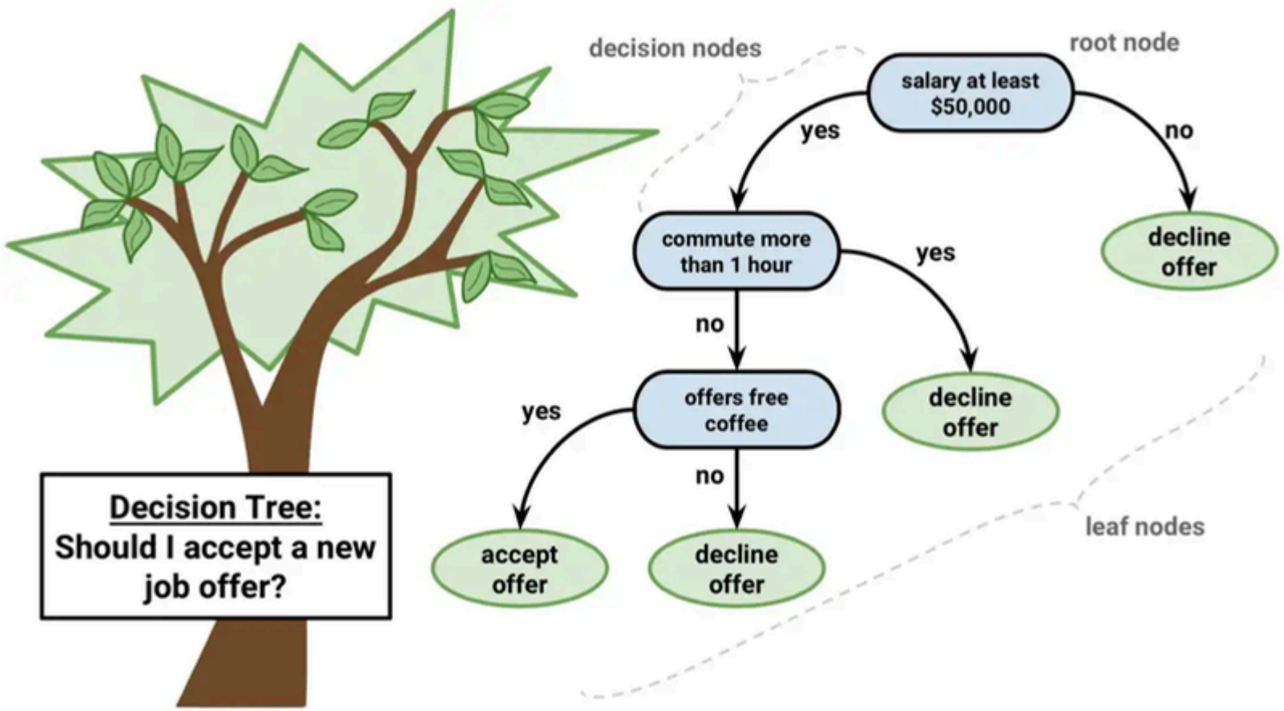
TRAIN_SET : ROWS WITHOUT MISSING VALUES

TEST_SET : ROWS WITHOUT MISSING VALUES

DECISION TREE

Resgression Tree

Classification Tree



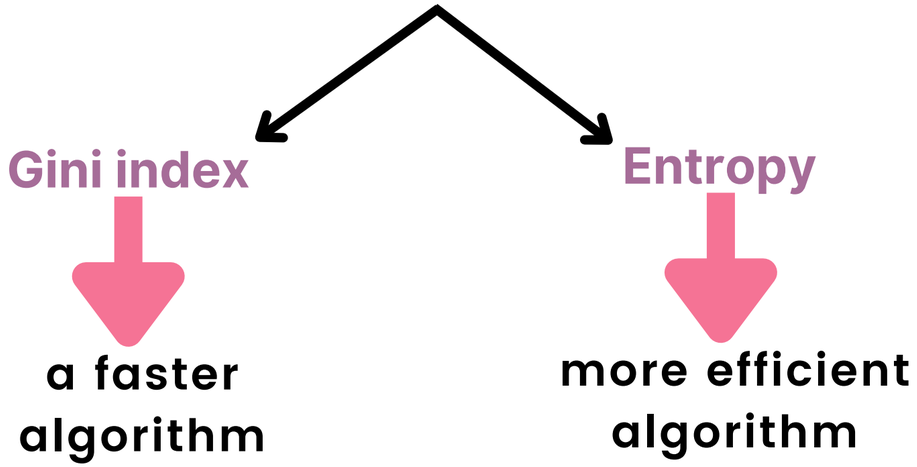
DECISION NODE : FEATURE

BRANCHE : TEST RESULT

LEAF : CLASS LABEL

ASM

Attribute Selection Method



leaves are pure

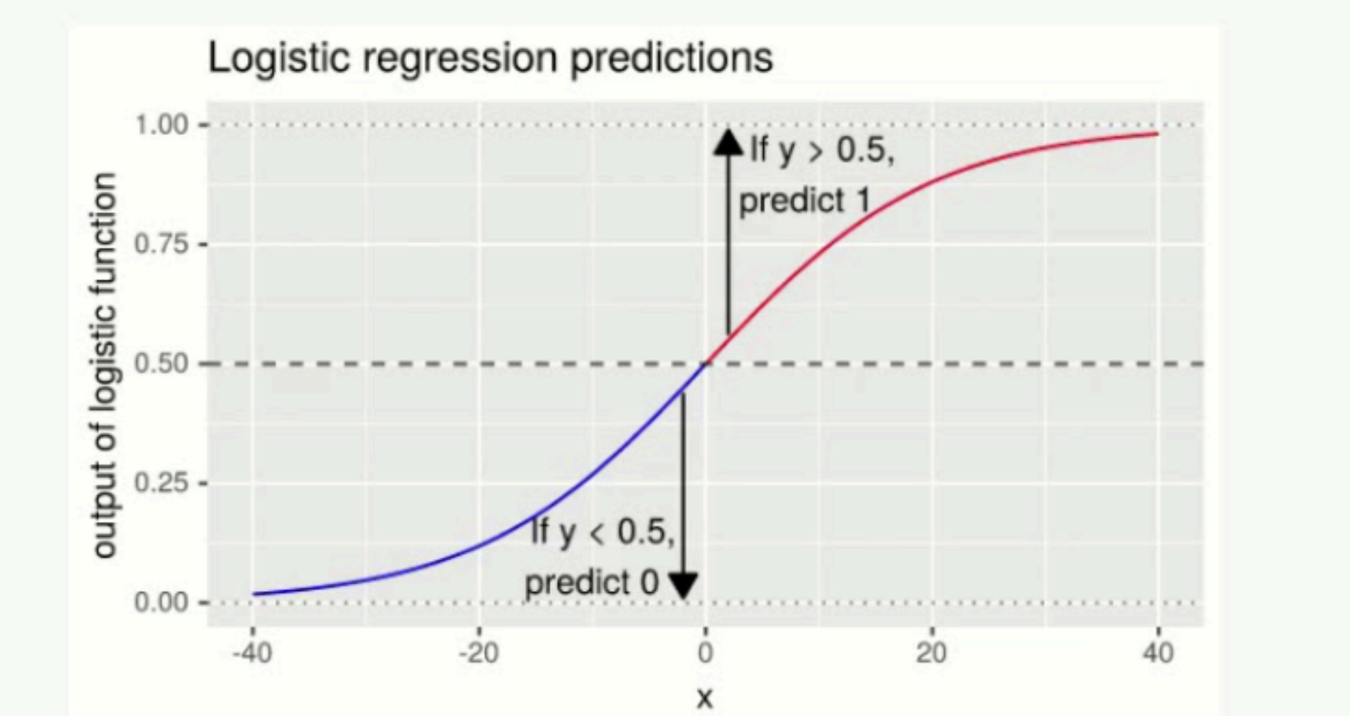
- sparse database (several zeros)
- encoding:
  dummy encoding or one-hot encoding

- Ne nécessite pas le scaling
- Accepte les variables catégoriques
- Facilité d'interprétation
  (par arbre ou avec des conditions)

Simple Imputation → LOGISTIC REGRESSION

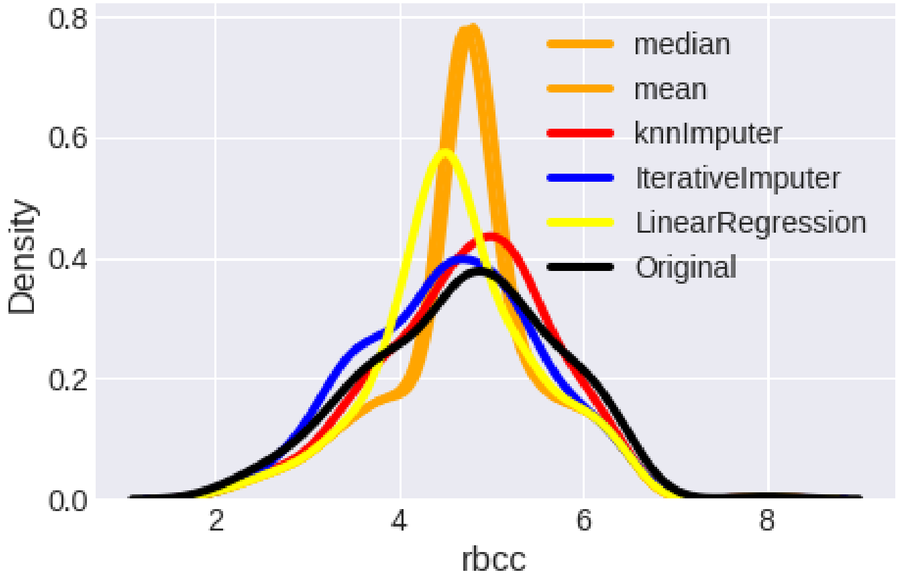
Supervised machine learning model

Soft clustering: probability of belonging to a cluster

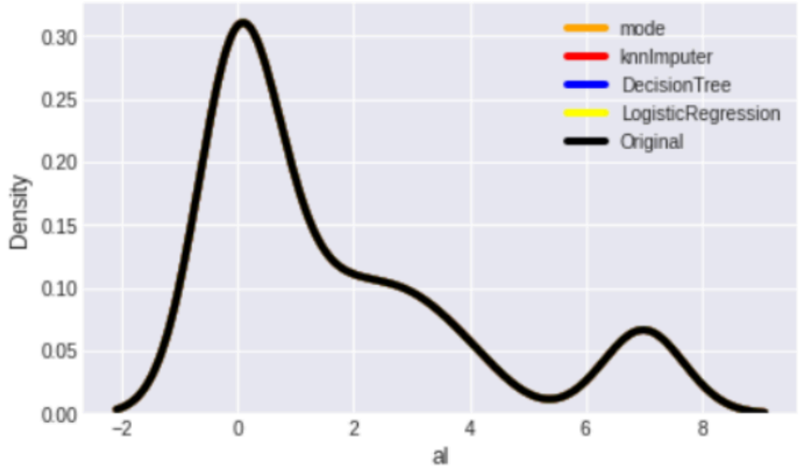


$$0 < P(Y(\text{outcome}) | X(\text{features})) < 1$$

age (Numeric feature)



al(Categorical feature)



Iterative Imputer

mode ✔
KNN Imputer
Logistic regression
Decision Tree

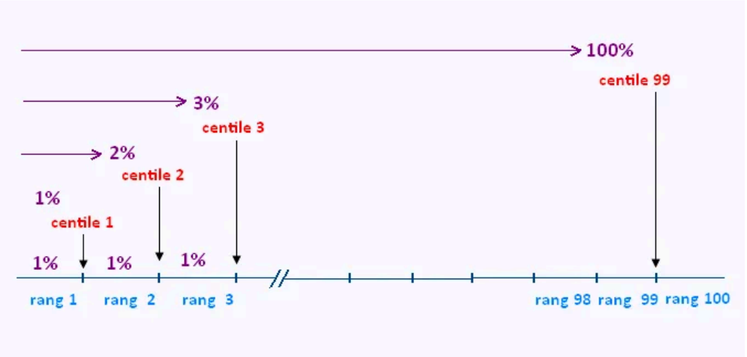
5. Treatment of outliers

PERCENTILE-BASED METHOD

STEP1 :

upper = np.percentile(data[feature], 100 - p)

lower = np.percentile(data[feature], p)



STEP2 :

We compare the outliers to the lower and upper

- obs value < lower → obs value = lower
- obs value > upper → obs value = upper

```
#Identifying the outliers by the percentile based method
def pct_method(data, level ,feature):
    # Upper and lower limits by percentiles
    upper = np.percentile(data[feature], 100 - level)
    lower = np.percentile(data[feature], level)
    #Imputing the outliers with the maximum and the minimum non-outlier values
    data[feature].loc[data[feature] > upper]  = upper
    data[feature].loc[data[feature] < lower ]  = lower
    # Returning the updated dataframe
    return data
```

INTERQUARTILE RANGE METHOD:

STEP1 :

$$IQR = Q3 - Q1$$

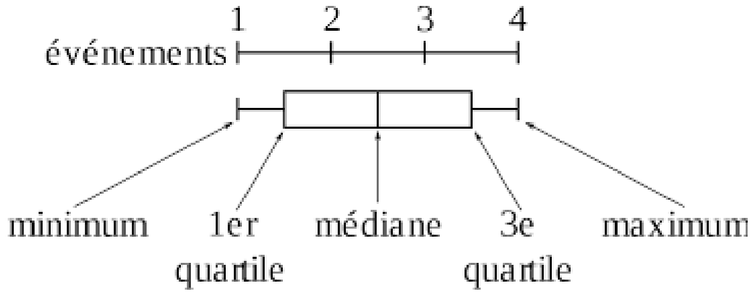
Q3 = np.percentile(data, 75)

Q1 = np.percentile(data, 25)

STEP2 :

$$IQRU = Q3 + 1.5 * IQR$$

$$IQRL = Q1 - 1.5 * IQR$$



STEP3 :

We compare the outliers to the IQRU and IQRL

- obs value < IQRL → obs value = IQRL
- obs value > IQRU → obs value = IQRU

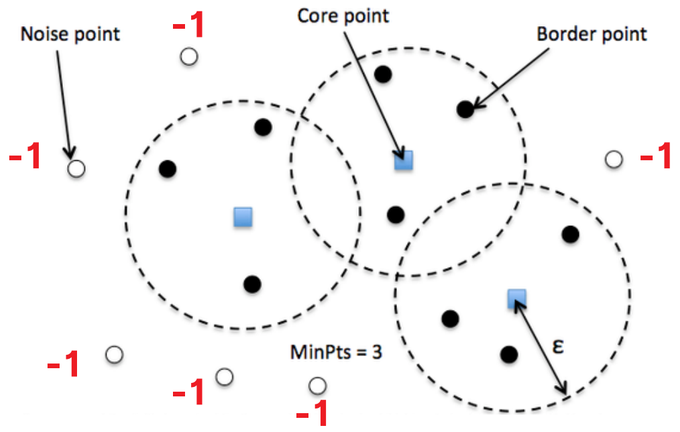
STANDARD DEVIATION METHOD (RULE OF THUMB):

$$std = \sigma(data[feature])$$

$$upper\_3std = E(data[feature]) + 3 * std$$

$$lower\_3std = E(data[feature]) - 3 * std$$

DBSCAN

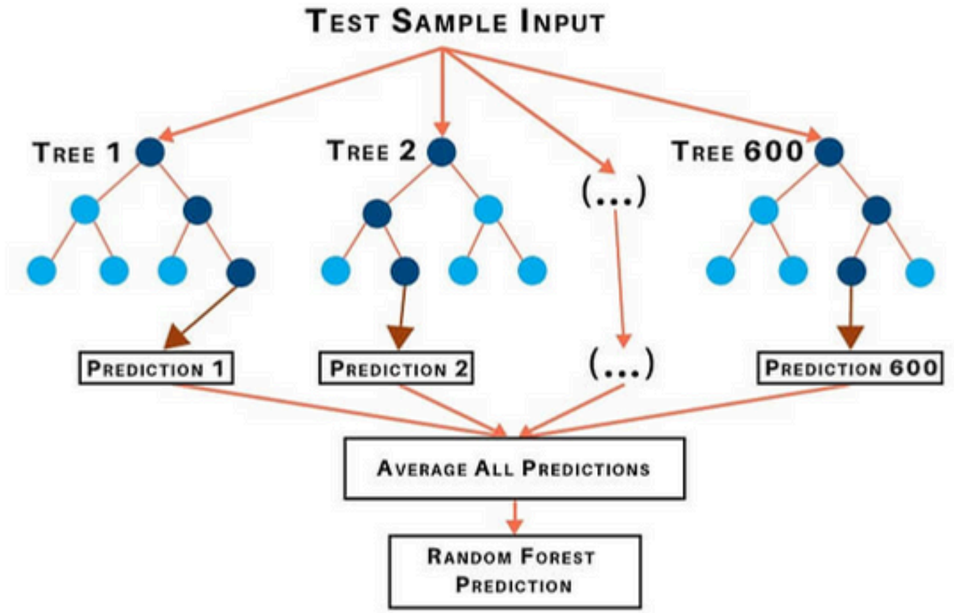


-1 { X_TEST
Y_TEST

↓

{ X_TRAIN
Y_TRAIN

RANDOM FOREST REGRESSOR

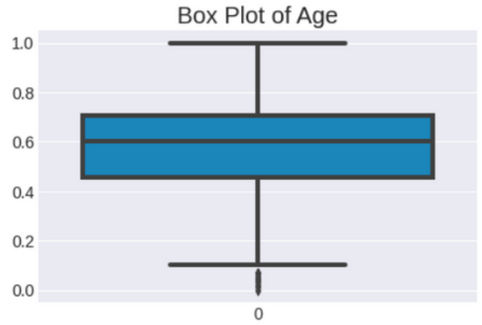


Hyperparameters

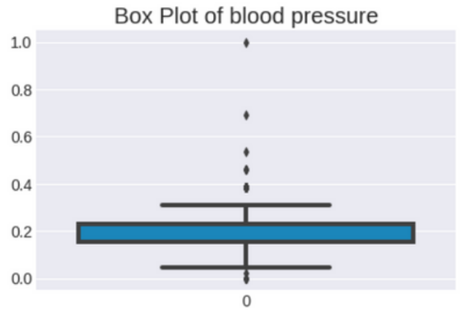
```
#Imputing the outliers using random forest regression
rf = RandomForestRegressor(n_estimators = 300, max_features = 'sqrt', max_depth = 5, random_state = 18).fit(X_train, y_train)
```

- n_estimators : Number of trees in the forest that the algo must build
- max_features : maximum number of features provided to each tree in a random forest.
- max_depth : the longest path between the root node and the leaf node.
- random_state : root(300)=18

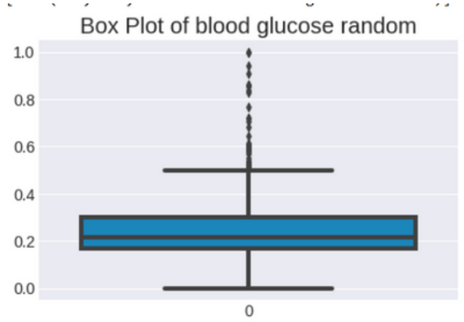
CHOICE OF METHOD: IMPUTATION OF OUTLIERS



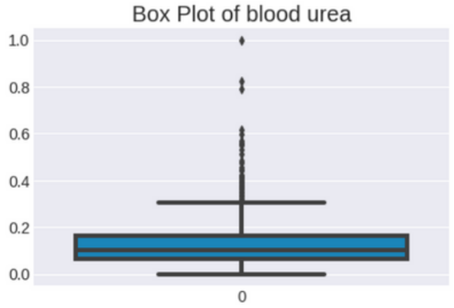

```
Kurtosis pct method:  0.03762099104839578
Kurtosis iqr method:  0.10186159977847797
Kurtosis std method:  0.10186159977847797
Kurtosis RandomForest Regression:  0.09872
```



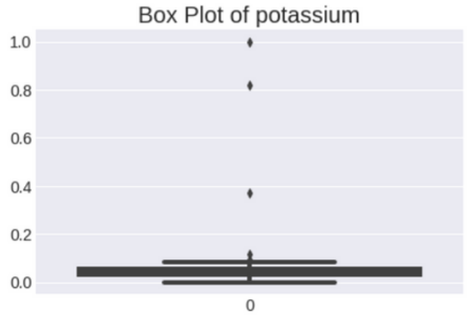

```
Kurtosis pct method:  -0.17596192263970556
Kurtosis iqr method:  8.576376266047816
Kurtosis std method:  0.1433101711443654
Kurtosis RandomForest Regression:  1.14822366
```




```
Kurtosis pct method:  3.370270588399983
Kurtosis iqr method:  4.2503517359464364
Kurtosis std method:  2.1455266473200116
Kurtosis RandomForest Regression:  4.26206
```

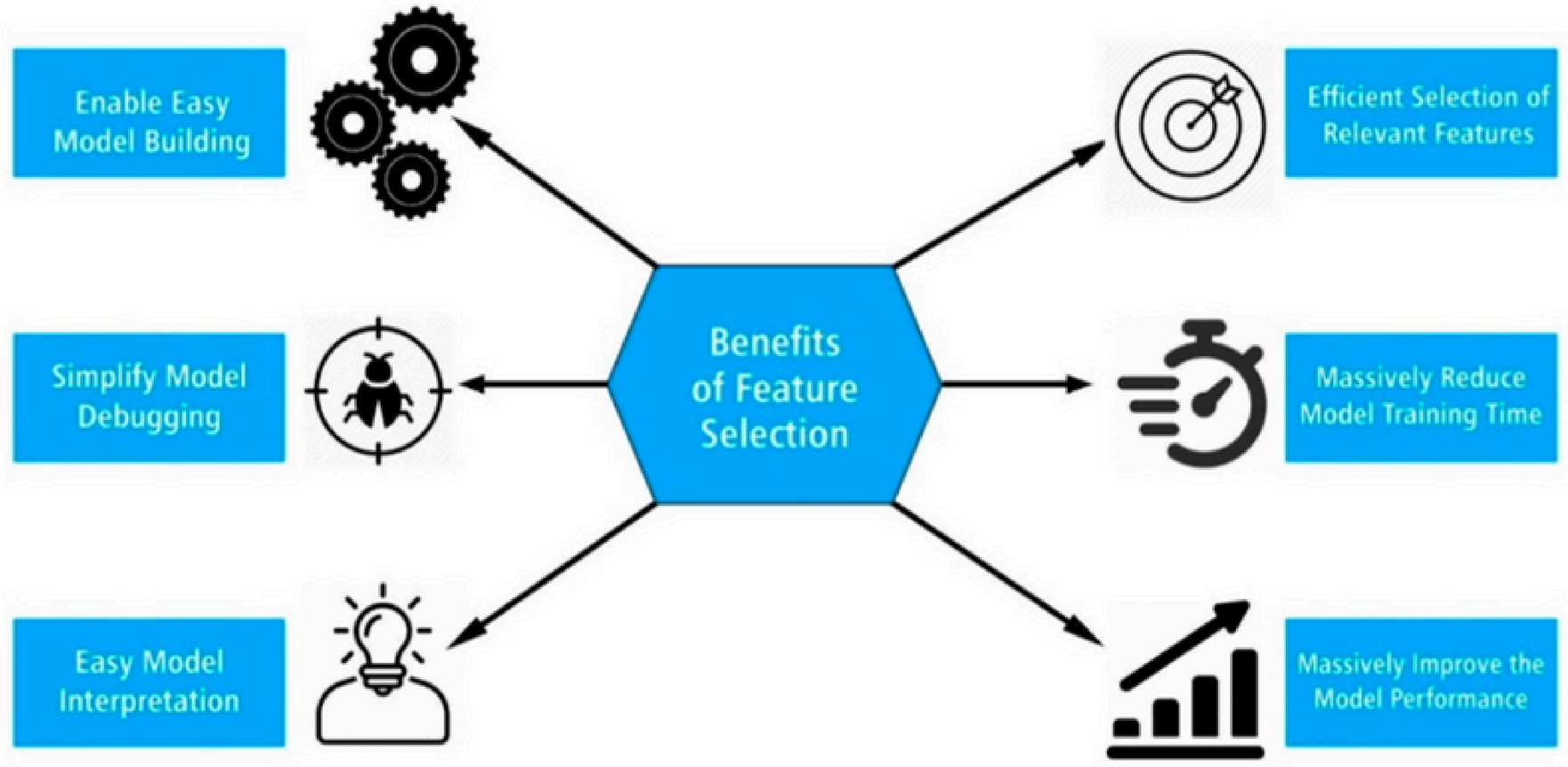



```
Kurtosis pct method:  3.979793061349391
Kurtosis iqr method:  9.483157258056195
Kurtosis std method:  2.9918470064997185
Kurtosis RandomForest Regression:  6.4059
```

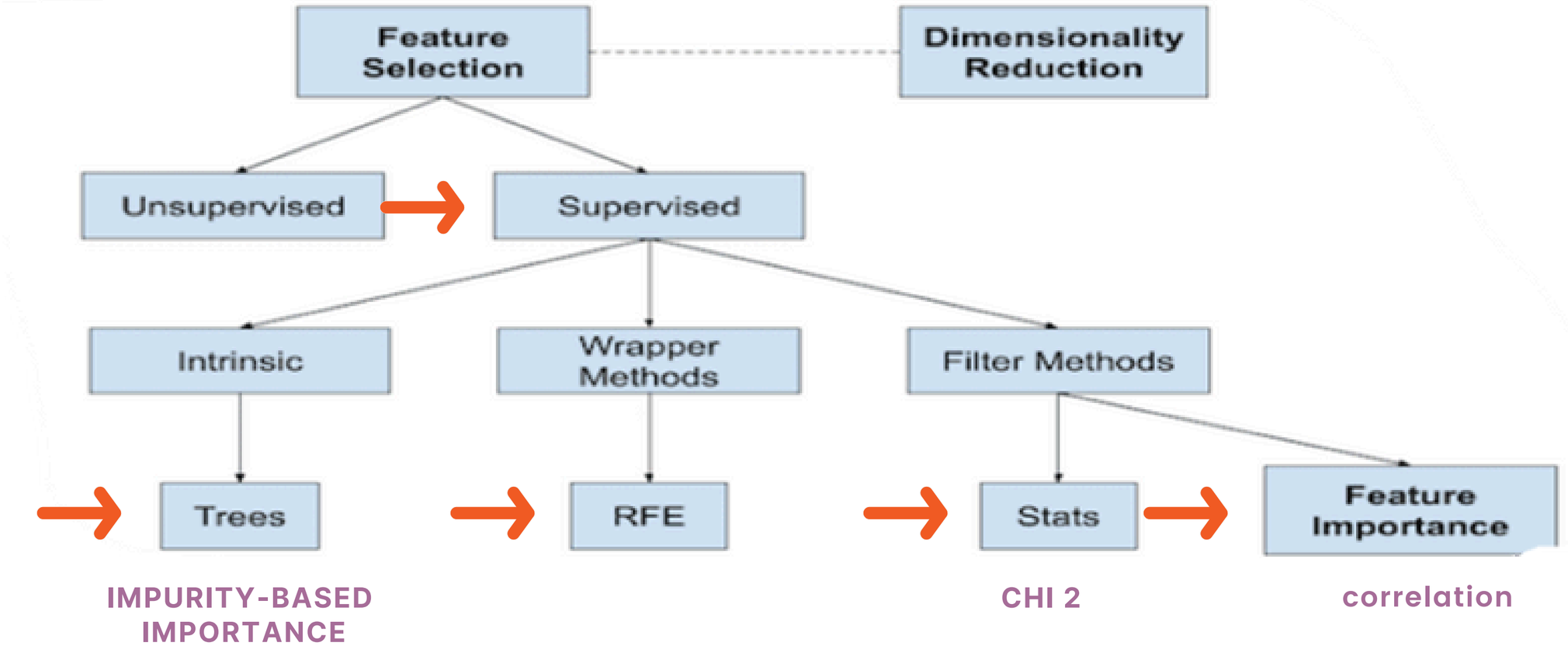



```
Kurtosis pct method:  0.5314832994517125
Kurtosis iqr method:  160.54525445670052
Kurtosis std method:  37.82689962547356
Kurtosis RandomForest Regression:  1.07491
```

6. Feature Selection



This impacts the performance of the model

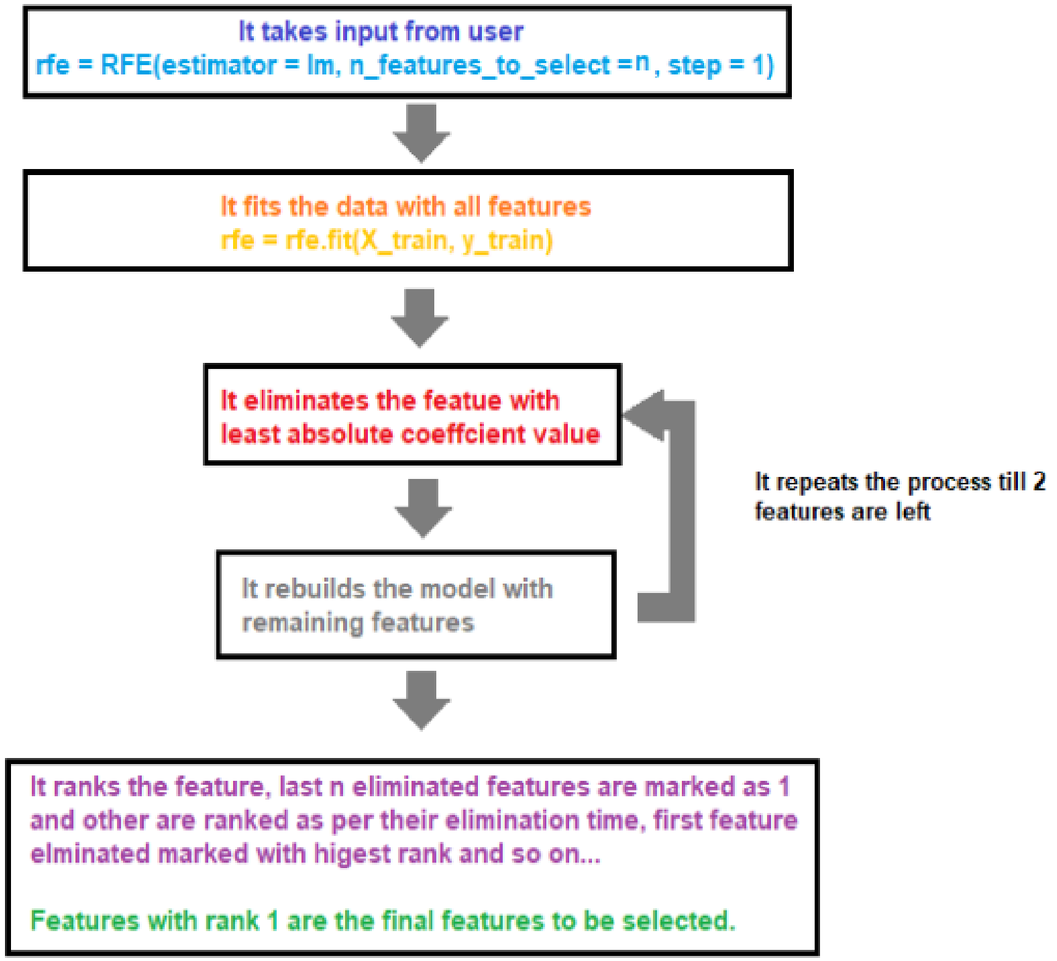

Feature Selection
Dimensionality Reduction
Unsupervised
Supervised
Intrinsic
Wrapper Methods
Filter Methods
Trees
RFE
Stats
Feature Importance
IMPURITY-BASED IMPORTANCE
CHI 2
correlation

WRAPPER METHODS

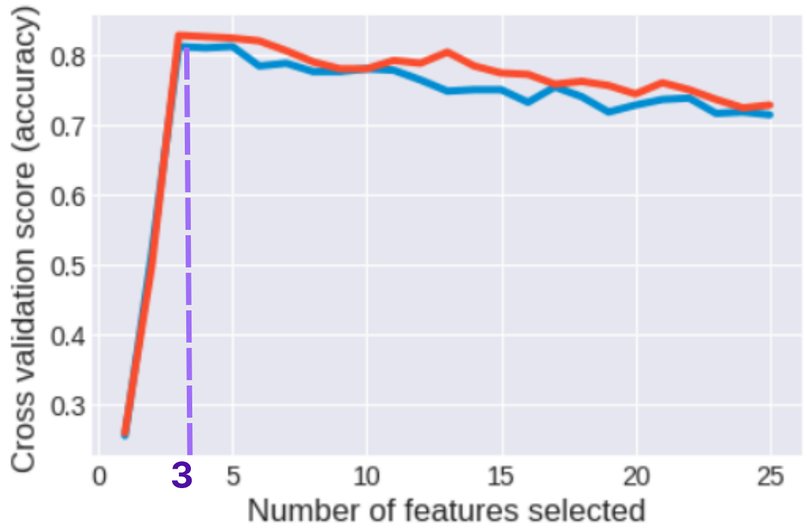
1) Recursive Feature Elimination

- RFE is a feature selection method that uses a model to evaluate the importance of each feature, and recursively eliminates the least important features until the desired number of features is obtained.

- By recursively eliminating less important features, RFE can help improve the performance of a predictive model and reduce model complexity.



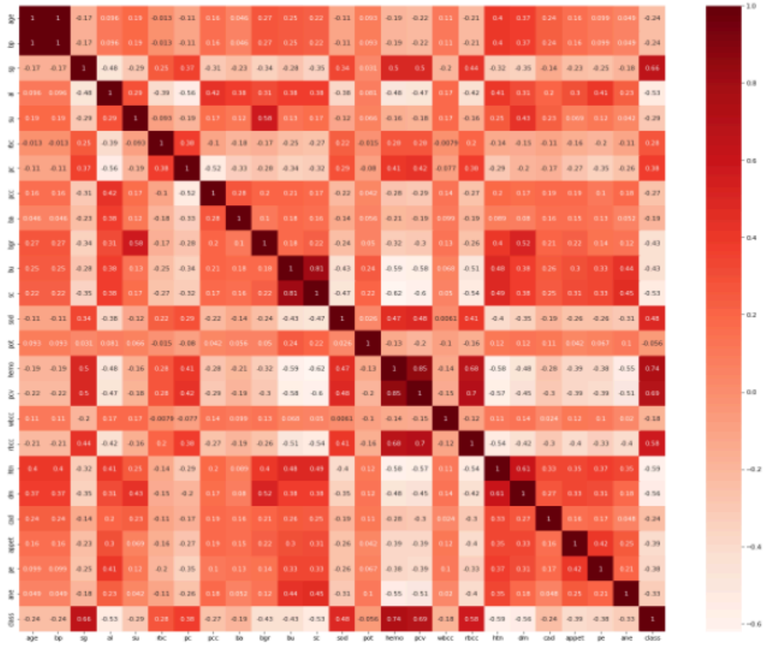
RFECV (Recursive Feature Elimination with Cross-Validation) performs recursive feature elimination with a cross-validation loop to extract optimal features.



The optimal number of entities selected by cross-validation is 3.

FILTER METHODS

1) Correlation method by choosing the correlation threshold :



```
print(str((cor_selector1(X, y,0.3))), 'selected features')
```

```
Index(['sg', 'al', 'su', 'rbc', 'bgr', 'bu', 'sod', 'hemo', 'pcv', 'rbcc',
       'htn', 'dm', 'appet'],
      dtype='object') selected features
```

By manually choosing the threshold of 0.3, the number of features selected by the algorithm is 13.

2) Correlation method by choosing the number of features to keep

```
cor_feature = cor_selector2(X,y,10)
print(str((cor_feature)), 'selected features')
```

```
['rbc', 'bu', 'bgr', 'sod', 'dm', 'al', 'htn', 'rbcc', 'pcv', 'hemo'] selected features
```

3) Chi-square Methods : For categorical data

$$X^2 = \frac{(Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency}$$

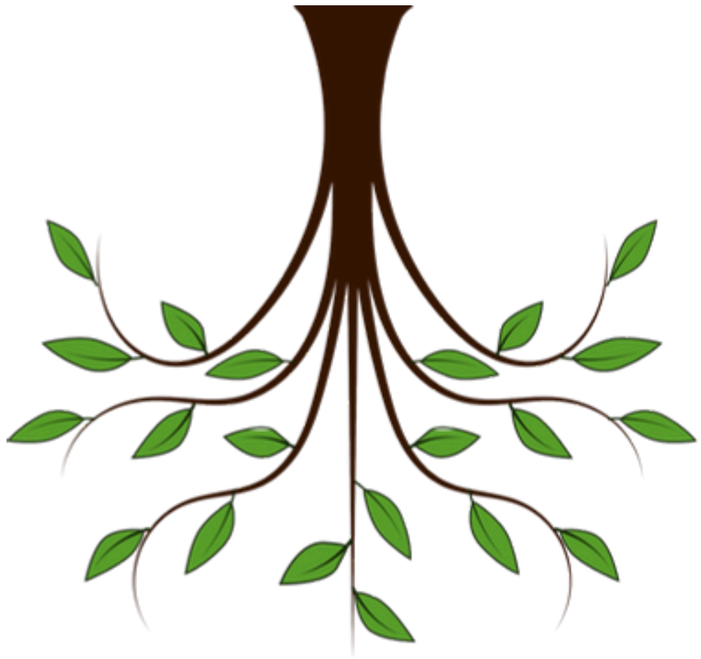
- Observed frequency = class observations
- Expected frequency =expected class observations if there was no relationship between the feature and the target.

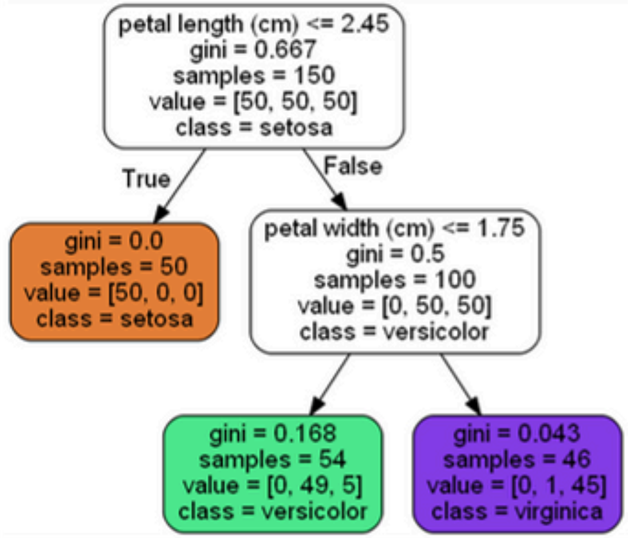
```
print(str(len(chi2_selector(X, y,num_feats))), 'selected features')
```

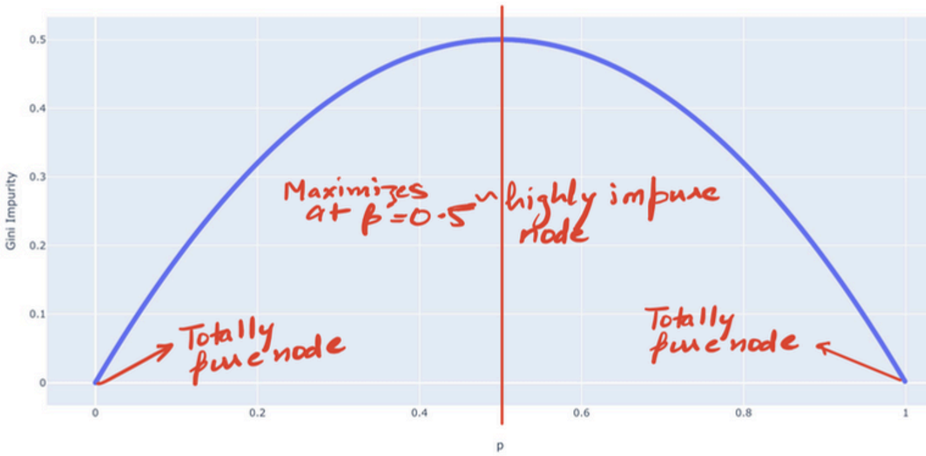
```
10 selected features
```

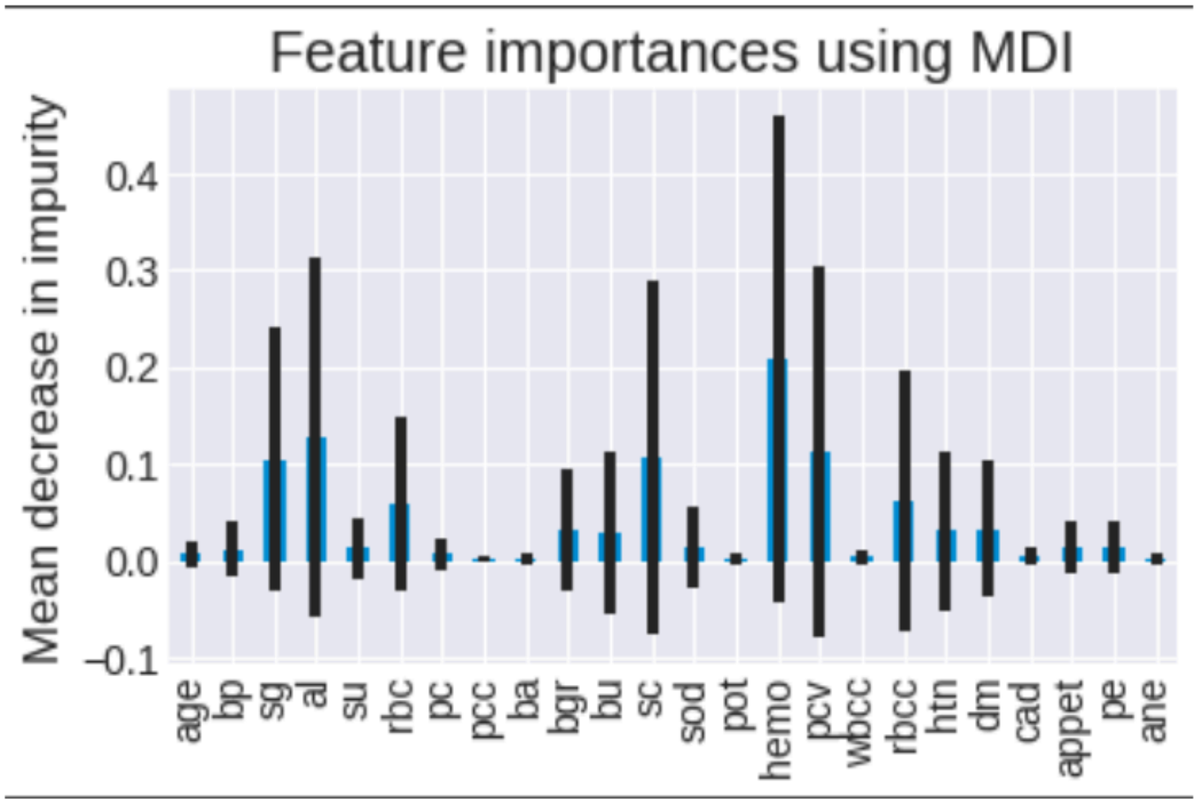
FEATURE IMPORTANCE WITH DECISION TREES :

This method combines the qualities of the Filter and Wrapper method to create the best subset.









We select features that have an importance greater than 0.028.

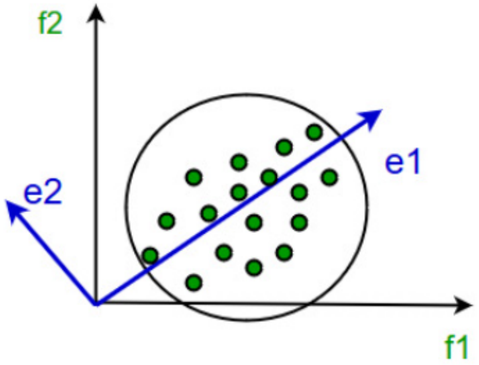
COMPARISON BETWEEN METHODS:

We perform KNN classification with selected features from each method.

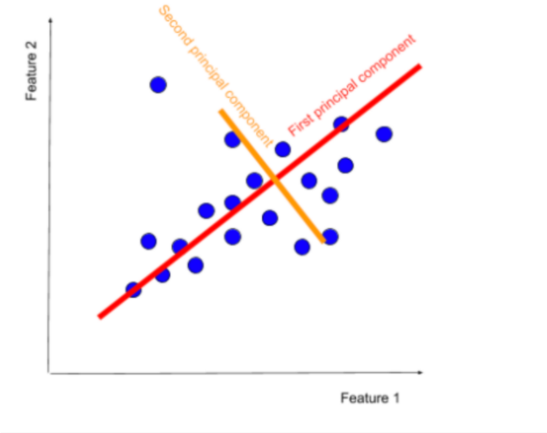
```
cor_selector1,  Test accuracy = 0.86
cor_selector2,  Test accuracy = 0.90
rfe_selector,   Test accuracy = 0.95 ✔
chi2_selector,  Test accuracy = 0.81
impurity-based importance,      Test accuracy = 0.90
```

the technical RFE has the highest precision

7. Feature Reduction



Principle component analysis(PCA) :

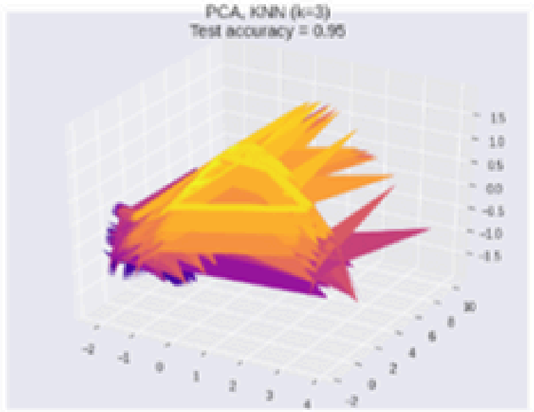


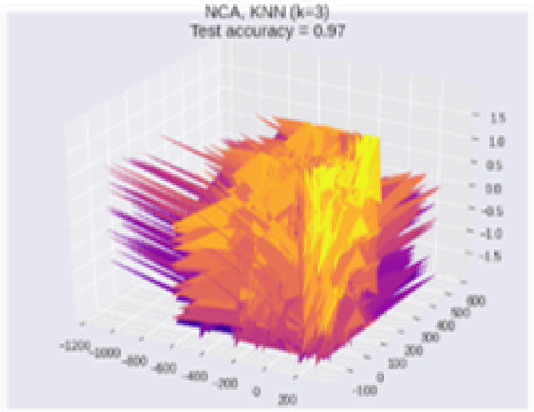

```
pca = make_pipeline(StandardScaler(),
                    PCA(n_components=2,
                        random_state=random_state))
```

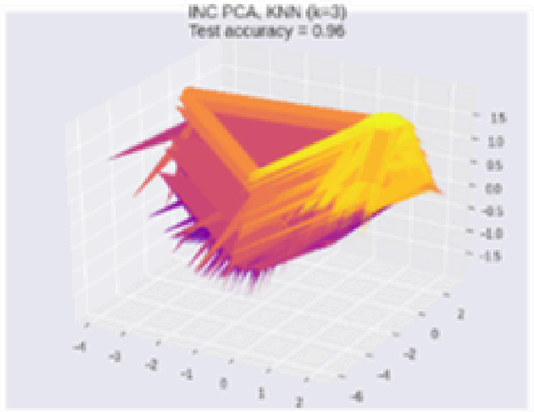
METHODS APPLIED

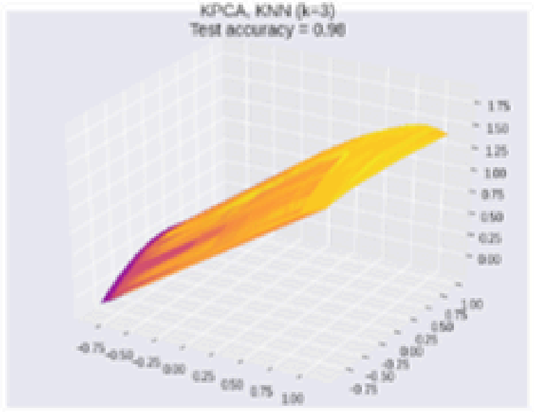
- Analyse des composants du principe incrémental (IPCA)
- Kernel PCA (KPCA)
- Sparse PCA (KPCA)
- Singular Value Decomposition (SVD)
- Gaussian random projection (GRP)
- Sparse random projection (SRP)
- Linear Discriminant Analysis(LDA)
- Neighborhood component analysis (NCA)

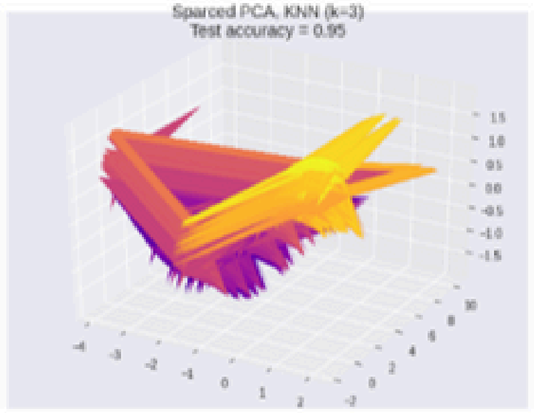
COMPARISON BETWEEN METHODS

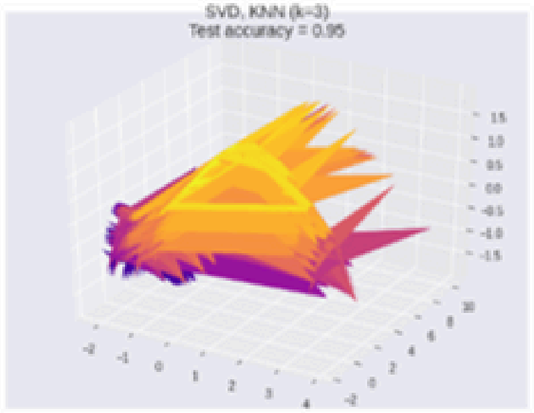


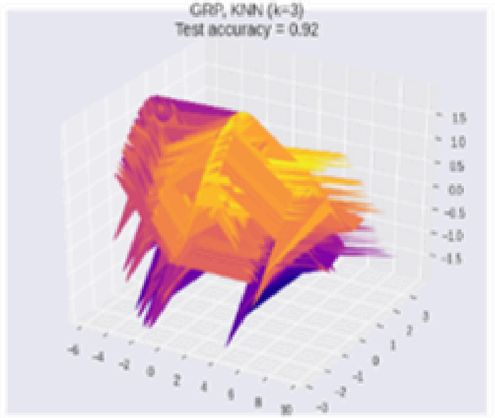


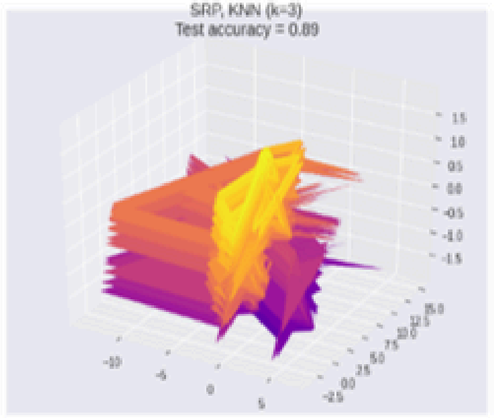


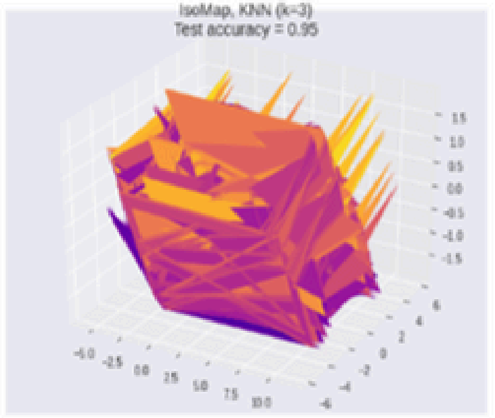


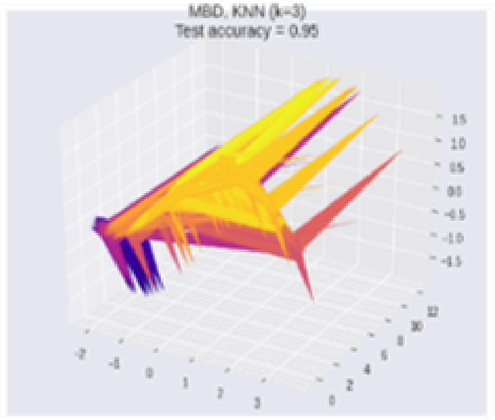


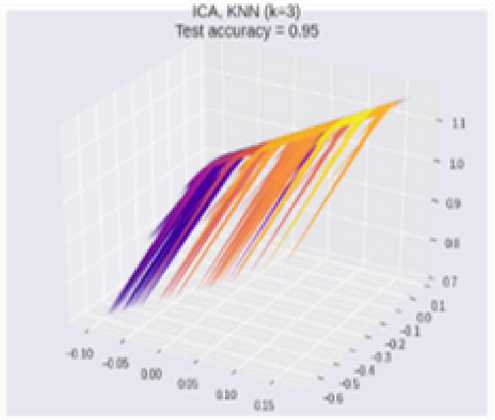


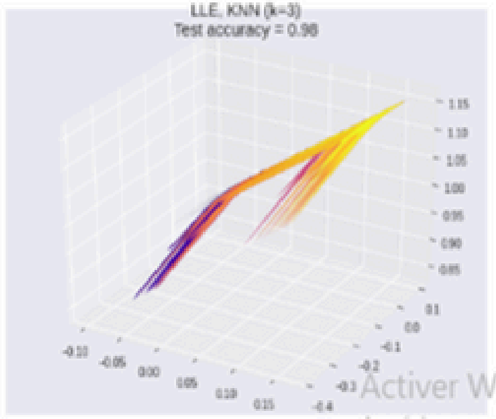












NCA, KPCA have the highest accuracy, we choose to continue with Kernel PCA.

MODELING AND EVALUATION

04

ARTICLE 1: Boosted Classifier and Features Selection for Enhancing Chronic Kidney Disease Diagnose

we used the 3 classifiers:

K-nearest neighbors

Naive bayes

Support Vector machine

Principe :

it is a probabilistic model which is based on the famous Bayes theorem.
we assume that all the variables are independent, then the presence of each variable does not affect the others, this is why it is called naive.

Types :

Multinomiale NB

used in the classification of documents (politics, technology, etc.). the features used by the classifier are for example the frequency of words present in the document.

Bernoulli NB

like the multinomial but The parameters we use to predict the class variable take the values yes or no(exp: whether or not a word appears in the text.)

Gaussien NB

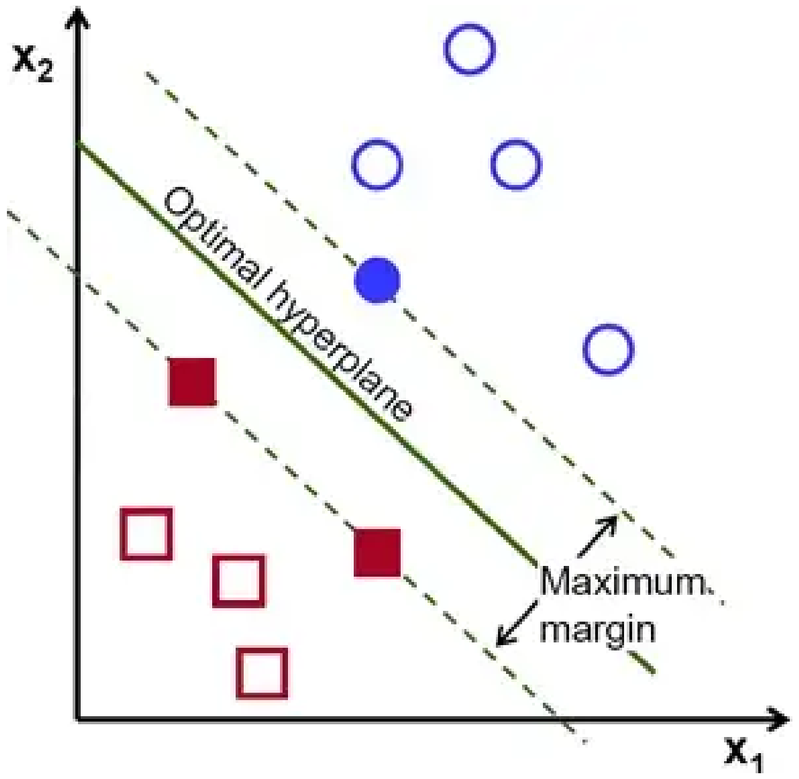
the variables take non-discrete continuous values which follow the normal law.

Advantage : **simple to implement and very quick.**

Disadvantage: **the problem of p=0 (eg: there are words in the test data which do not exist in the training data)**

Principe :

it is a supervised model which aims to find the optimal hyperplane which distinctly classifies the data in an N-dim space (N: number of variables).



The optimal hyperplane: to separate between two different classes with the maximum margin which gives reinforcement to classify future data points.

Advantage :

-more productive with high dimensional space
-when there is a very clear margin it can separate classes well

Disadvantage:

As the support vector works by placing the data above and below the hyperplane, there is no probabilistic clarification for classification.

Method 1: BASE

the basic method is modeling without feature selection

Split between x (features) et y (target) :

```
[ ]  # split into x and y dataset
     X=df.drop('class',axis=1)
     y=df['class']
```

Evaluation with cross validation (K=10):

For all algorithms (KNN, SVM, NB) we used cross validation with K which is equal to 10 to evaluate the models

	labels	Accuracy	Precision	Recall	F1
0	KNN	0.910221	0.946667	0.9425	0.926279
1	Naive Bayes	0.796011	1.000000	0.9000	0.884693
2	SVM	0.949661	0.966667	0.9675	0.956863

=>SVM is the best model with the highest F1-score

Method 2: CFS

the CFS method is modeling with feature selection

Split entre x (features) et y (target) :

```
[ ]  # split into x and y dataset
     # Applying cfs
     cor_feature = cor_selector2(X,y,10)
     x=df[cor_feature]
     y=df['class']
```

Evaluation with cross validation (K=10):

For all algorithms (KNN, SVM, NB) we used cross validation with K which is equal to 10 to evaluate the models

	labels	Accuracy	Precision	Recall	F1
0	KNN	0.948057	0.920000	0.9500	0.930511
1	Naive Bayes	0.714952	0.986667	0.8425	0.827472
2	SVM	0.913911	0.946667	0.9450	0.927796

=>KNN is the best model with the highest F1-score

Method 3: CFS + ADABOOST

la méthode de CFS avec adaboost c'est le modeling avec feature selection plus Adaboost

Split between x (features) et y (target) :

```
[ ]  # split into x and y dataset
     # Applying cfs
     cor_feature = cor_selector2(X,y,10)
     x=df[cor_feature]
     y=df['class']
```

Evaluation with cross validation (K=10):

For all algorithms (W-KNN, SVM, NB) we used the Adaboost classifier for all models and cross validation with K which is equal to 10 to evaluate them.

	labels	Accuracy	Precision	Recall	F1
0	AdaBoost with SVM	0.955998	0.926667	0.9550	0.938125
1	AdaBoost with weighted KNN	NaN	NaN	NaN	NaN
2	AdaBoost with Naive Bayes	0.476907	0.993333	0.5875	0.644184

=>SVM is the best model with the highest F1-score

Comparison

We will compare the F1-Score for each model in each method: without feature selection, with feature selection and finally with feature selection and using Adaboost.
We chose to work with the F1-Score because our dataset is unbalanced with an unequal distribution

KNN :

F1-score (Base)=0,92
F1-Score(CFS)= 0,93.

Naive Bayes :

F1-score (Base)=0,884
F1-Score(CFS)=0.827
F1-score(CFS+Adaboost) = 0.644.

=>F1 score decreased with adaboost and CFS drastically, which is an indicator that we over-adjusted.
(overfitting: when the model matches the training data exactly and the algorithm cannot work accurately against invisible data).

SVM :

F1-score (Base)=0.95
F1-Score(CFS)=0.927
F1-score(CFS+Adaboost)=0.94.

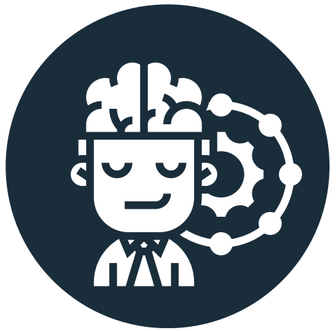
ARTICLE 2:Diagnosis of Chronic Kidney Disease Using Effective Classification Algorithms and Recursive Feature Elimination Techniques

we used the 4 classifiers:

Random Forest regressor

Decision Trees

K-nearest neighbors



Support Vector machine

Split between training and test data :

```
# split into x and y dataset
X=finalDf.drop('class',axis=1)
y=finalDf['class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
#Training : 300 patients
#Test : 100 patients
```

Hyperparameter Tuning:

For all algorithms (RF, DT, SVM) we estimated the hyperparameters using Gridsearch

```
[ ]  rf_random.best_params_
```

```
{'n_estimators': 313,
 'min_samples_split': 5,
 'min_samples_leaf': 6,
 'max_features': 'sqrt',
 'max_depth': 80}
```

```
grid_search_cv.best_estimator_
```

```
DecisionTreeClassifier(max_leaf_nodes=2, random_state=42)
```

```
{'C': 1000, 'gamma': 1, 'kernel': 'rbf'}
SVC(C=1000, gamma=1)
```

the training :

RANDOM FOREST REGRESSION

```
[ ]  #Train the model using the training sets
     rf_model.fit(X_train, y_train)
```

K-NEAREST NEIGHBORS

```
#Train the model using the training sets
KNN.fit(X_train, y_train)
```

DECISION TREES

```
#Train the model using the training sets
DT_model.fit(X_train, y_train)
```

SUPPORT VECTOR MACHINE

```
# fitting the model for grid search
grid.fit(X_train, y_train)
```

The evaluation:

RANDOM FOREST REGRESSION

Accuracy: 0.9702970297029703
Precision: 0.9705882352941176
Recall: 0.9428571428571428
F1: 0.9565217391304348

K-NEAREST NEIGHBORS

Accuracy: 0.9603960396039604
Precision: 0.9696969696969697
Recall: 0.9142857142857143
F1: 0.9411764705882354

DECISION TREES

Accuracy: 0.9801980198019802
Precision: 0.9714285714285714
Recall: 0.9714285714285714
F1: 0.9714285714285714

SUPPORT VECTOR MACHINE

Accuracy: 0.9702970297029703
Precision: 0.9705882352941176
Recall: 0.9428571428571428
F1: 0.9565217391304348

COMPARISON OF MODELS

we will compare the F1-Score for each model because our dataset is unbalanced with an unequal distribution.
The model with the highest F1-score is the model we will deploy.
=>here the decision trees model will be deployed.

	labels	Accuracy	Precision	Recall	F1
0	SVM	0.970297	0.970588	0.942857	0.956522
1	KNN	0.960396	0.969697	0.914286	0.941176
2	Decision Trees	0.980198	0.971429	0.971429	0.971429
3	Random Forest	0.970297	0.970588	0.942857	0.956522

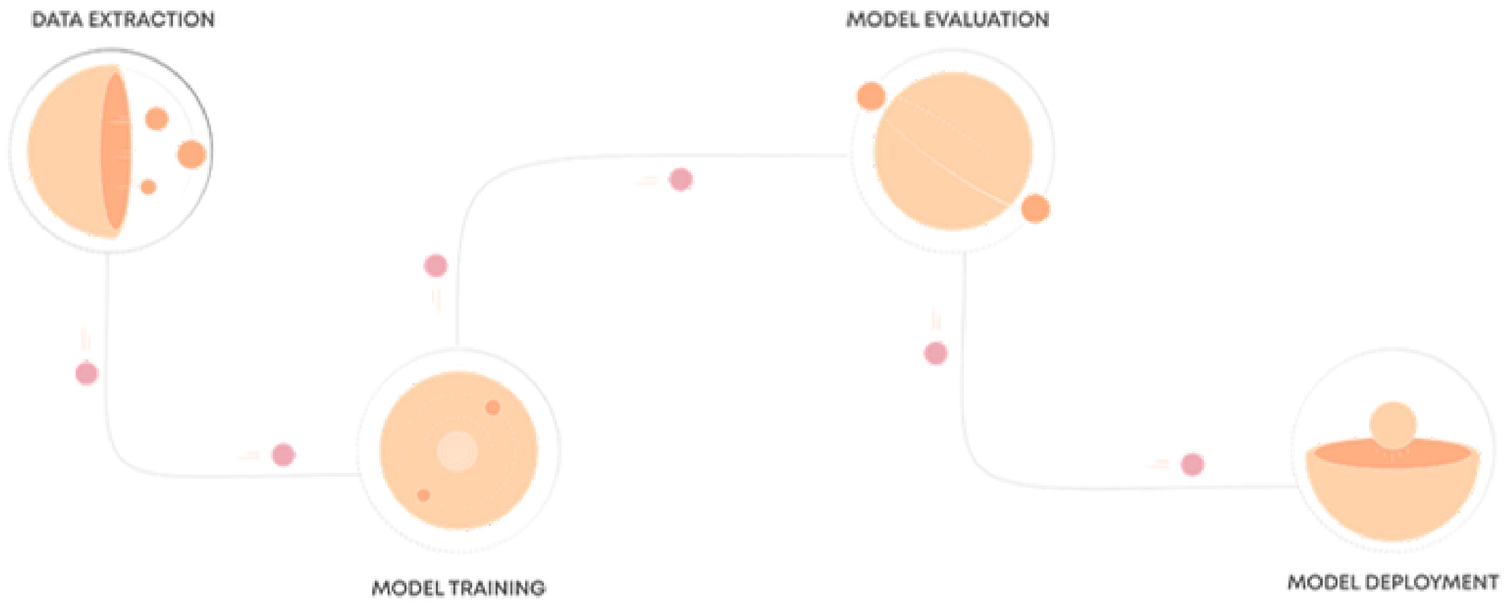
DEPOY**MENT**

05

Deployment :

- It’s good to implement a powerful machine learning model. But the latter will not be of much interest if it cannot be used in “an application”.
- Indeed, after training a model, it is not quite ready to be used. Additional code must be added so that it can actually be used properly.
- Django is a high-level Python framework, allowing rapid development of secure, maintainable websites.
- Django has an MVT "View Template" architecture, it has several similarities with the View Controller Model architecture.

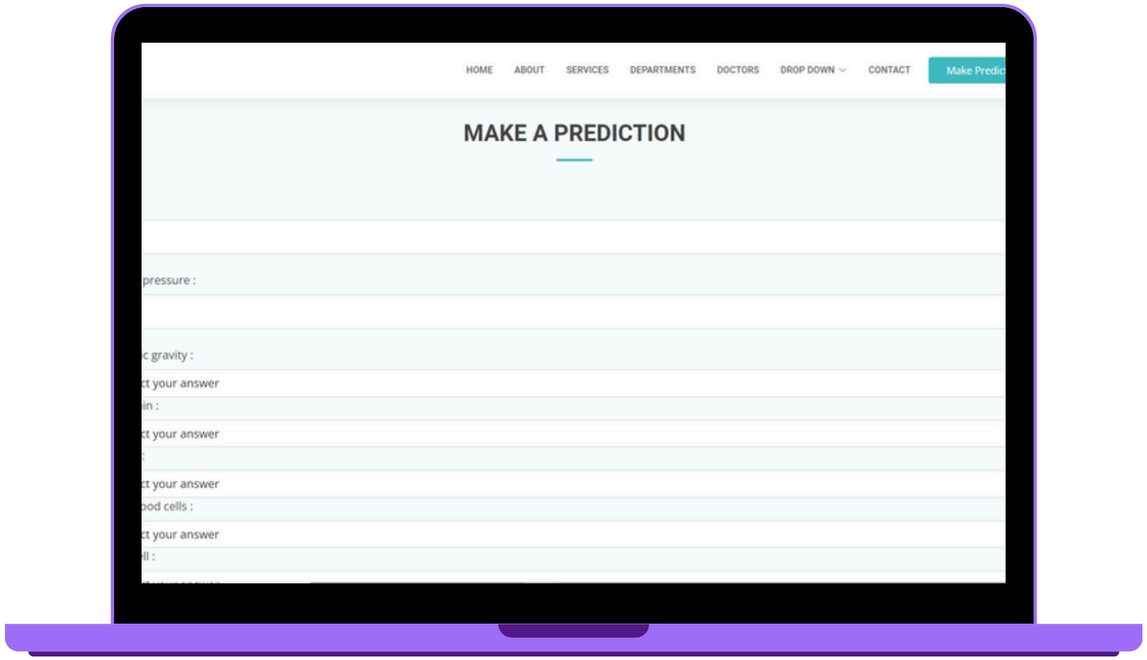
Pipeline :

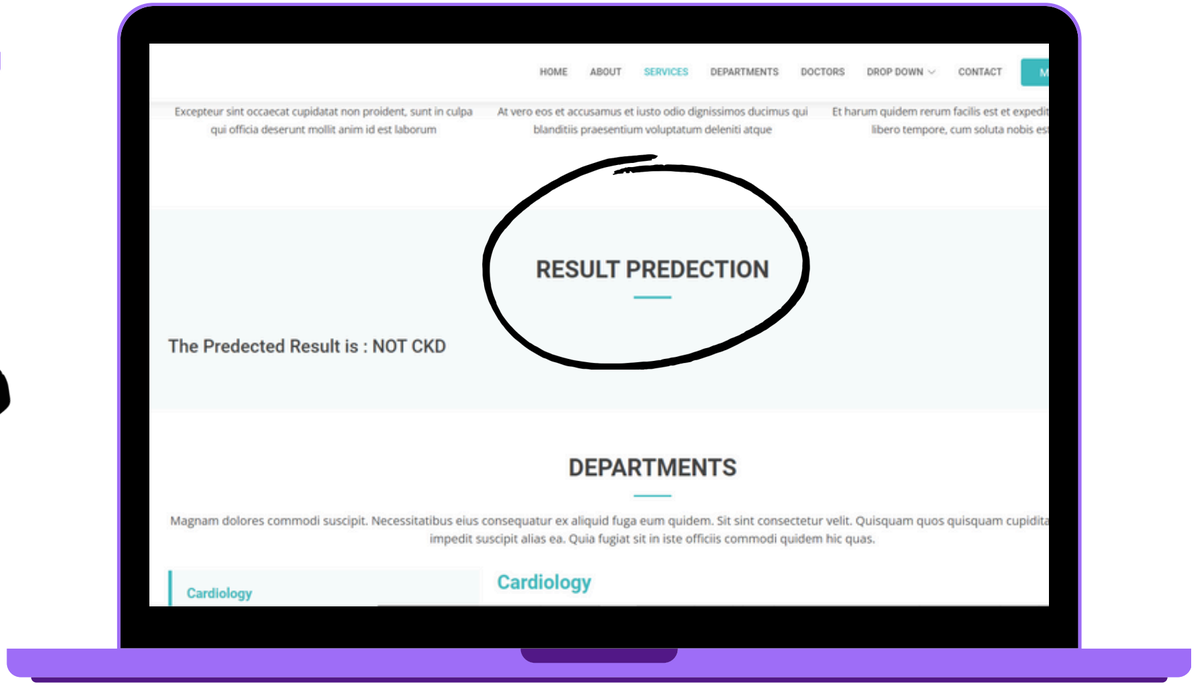


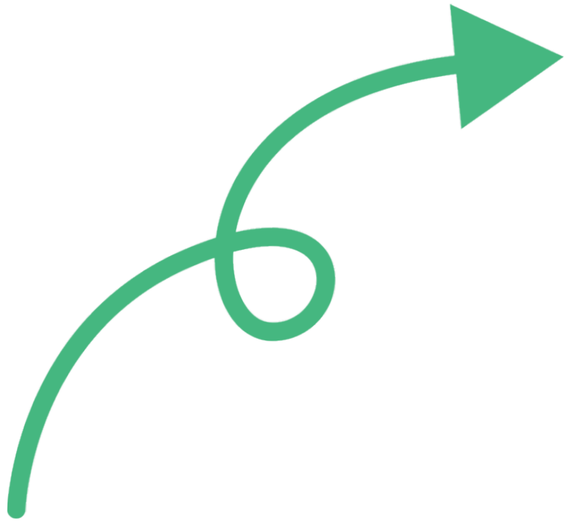
A pipeline is often used to refer to a series of processes connected in this way, where the output of each process becomes the input to the next process.

Feature scalling → **Feature selection** → **Modeling**

Deployment :


HOME
ABOUT
SERVICES
DEPARTMENTS
DOCTORS
DROP DOWN
CONTACT
MAKE A PREDICTION
pressure :
c gravity :
t your answer
in :
t your answer
t your answer
od cells :
t your answer


HOME
ABOUT
SERVICES
DEPARTMENTS
DOCTORS
DROP DOWN
CONTACT
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum
At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque
RESULT PREDECTION
The Predected Result is : NOT CKD
DEPARTMENTS
Magnam dolores commodi suscipit. Necessitatibus eius consequatur ex aliquid fuga eum quidem. Sit sint consectetur velit. Quisquam quos quisquam cupidita
impedit suscipit alias ea. Quia fugiat sit in iste officiis commodi quidem hic quas.
Cardiology
Cardiology



THANK YOU FOR YOUR ATTENTION