

# SPRINT 2 - TEAM ID PNT2022TMID12917

**DATE: 12th November 2022**

**Topic: Early Detection of Chronic Kidney Disease using Machine Learning**

## MODEL BUILDING

### 1. Importing the necessary libraries

In [64]:

```
import pandas as pd
import numpy as np
from collections import Counter as c
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
import pickle
```

### 2. Loading the Dataset

In [90]:

```
data=pd.read_csv(r"C:\Users\archa\Desktop\IBM Datasets\chronickidneydisease.csv")
```

In [91]:

```
data.head(5)
```

Out[91]:

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6

5 rows × 26 columns

In [92]:

```
data.tail(5)
```

Out[92]:

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47	6700	4.9
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54	7800	6.2
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49	6600	5.4
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51	7200	5.9
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53	6800	6.1

5 rows × 26 columns

In [93]:

```
data.drop(["id"],axis=1,inplace=True)
```

In [94]:

```
data.columns
```

Out[94]:

```
Index(['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
      'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
      'appet', 'pe', 'ane', 'classification'],
      dtype='object')
```

In [95]:

```
data.columns=['age', 'blood_pressure', 'specific_gravity', 'albumin', 'sugar', 'red_blood_cell',  
             'serum_creatinine', 'sodium', 'potassium', 'hemoglobin', 'packed_cell_volume', 'white_blood_cell_count',  
             'appetite', 'pedal_edema', 'anemia', 'class']
```

In [96]:

```
data.columns
```

Out[96]:

```
Index(['age', 'blood_pressure', 'specific_gravity', 'albumin', 'sugar',  
      'red_blood_cells', 'pus_cell', 'pus_cell_clumps', 'bacteria',  
      'blood glucose random', 'blood_urea', 'serum_creatinine', 'sodium',  
      'potassium', 'hemoglobin', 'packed_cell_volume',  
      'white_blood_cell_count', 'red_blood_cell_count', 'hypertension',  
      'diabetesmellitus', 'coronary_artery_disease', 'appetite',  
      'pedal_edema', 'anemia', 'class'],  
      dtype='object')
```

### 3. Understanding Data Type And Summary Of Features

In [97]:

data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 25 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   age                                    391 non-null    float64
1   blood_pressure                        388 non-null    float64
2   specific_gravity                      353 non-null    float64
3   albumin                              354 non-null    float64
4   sugar                                351 non-null    float64
5   red_blood_cells                       248 non-null    object
6   pus_cell                              335 non-null    object
7   pus_cell_clumps                       396 non-null    object
8   bacteria                              396 non-null    object
9   blood_glucose_random                  356 non-null    float64
10  blood_urea                            381 non-null    float64
11  serum_creatinine                      383 non-null    float64
12  sodium                                313 non-null    float64
13  potassium                             312 non-null    float64
14  hemoglobin                            348 non-null    float64
15  packed_cell_volume                    330 non-null    object
16  white_blood_cell_count                 295 non-null    object
17  red_blood_cell_count                   270 non-null    object
18  hypertension                           398 non-null    object
19  diabetesmellitus                      398 non-null    object
20  coronary_artery_disease                398 non-null    object
21  appetite                              399 non-null    object
22  pedal_edema                           399 non-null    object
23  anemia                                399 non-null    object
24  class                                  400 non-null    object
dtypes: float64(11), object(14)
memory usage: 78.2+ KB
```

## Target Column

In [98]:

data['class'].unique()

Out[98]:

array(['ckd', 'ckd\t', 'notckd'], dtype=object)

## Rectifying the target column

In [99]:

```
data['class']=data['class'].replace("ckd\t","ckd")
data['class'].unique()
```

Out[99]:

array(['ckd', 'notckd'], dtype=object)

## Categorical Columns

In [100]:

```
catcols=set(data.dtypes[data.dtypes=='O'].index.values)
print(catcols)
```

```
{'class', 'packed_cell_volume', 'red_blood_cells', 'pus_cell', 'appetite',
'bacteria', 'pedal_edema', 'hypertension', 'red_blood_cell_count', 'white_blood_cell_count', 'anemia', 'diabetesmellitus', 'coronary_artery_disease', 'pus_cell_clumps'}
```

In [101]:

```

for i in catcols:
    print("Columns :",i)
    print(c(data[i]))
    print('*'*120+'\n')

```

Columns : class

Counter({'ckd': 250, 'notckd': 150})

```

*****
*****

```

Columns : packed\_cell\_volume

```

Counter({nan: 70, '52': 21, '41': 21, '44': 19, '48': 19, '40': 16, '43': 14, '45': 13, '42': 13, '32': 12, '36': 12, '33': 12, '28': 12, '50': 12, '37': 11, '34': 11, '35': 9, '29': 9, '30': 9, '46': 9, '31': 8, '39': 7, '24': 7, '26': 6, '38': 5, '47': 4, '49': 4, '53': 4, '51': 4, '54': 4, '27': 3, '22': 3, '25': 3, '23': 2, '19': 2, '16': 1, '\t?': 1, '14': 1, '18': 1, '17': 1, '15': 1, '21': 1, '20': 1, '\t43': 1, '9': 1})

```

```

*****
*****

```

Columns : red\_blood\_cells

Counter({'normal': 201, nan: 152, 'abnormal': 47})

```

*****
*****

```

Columns : pus\_cell

Counter({'normal': 259, 'abnormal': 76, nan: 65})

```

*****
*****

```

Columns : appetite

Counter({'good': 317, 'poor': 82, nan: 1})

```

*****
*****

```

Columns : bacteria

Counter({'notpresent': 374, 'present': 22, nan: 4})

```

*****
*****

```

Columns : pedal\_edema

Counter({'no': 323, 'yes': 76, nan: 1})

```

*****
*****

```

Columns : hypertension

Counter({'no': 251, 'yes': 147, nan: 2})

```

*****
*****

```

Columns : red\_blood\_cell\_count

```

Counter({nan: 130, '5.2': 18, '4.5': 16, '4.9': 14, '4.7': 11, '3.9': 10, '4.8': 10, '4.6': 9, '3.4': 9, '3.7': 8, '5.0': 8, '6.1': 8, '5.5': 8, '5.9': 8, '3.8': 7, '5.4': 7, '5.8': 7, '5.3': 7, '4.3': 6, '4.2': 6, '5.6': 6, '4.4': 5, '3.2': 5, '4.1': 5, '6.2': 5, '5.1': 5, '6.4': 5, '5.7': 5, '6.5': 5, '3.6': 4, '6.0': 4, '6.3': 4, '4.0': 3, '4': 3, '3.5': 3, '3.3': 3, '5': 2, '2.6': 2, '2.8': 2, '2.5': 2, '3.1': 2, '2.1': 2, '2.9': 2, '2.7': 2, '3.0': 2, '2.3': 1, '8.0': 1, '3': 1, '2.4': 1, '\t?': 1})

```

```

*****

```

\*\*\*\*\*

Columns : white\_blood\_cell\_count

```
Counter({nan: 105, '9800': 11, '6700': 10, '9600': 9, '9200': 9, '7200': 9,
'6900': 8, '11000': 8, '5800': 8, '7800': 7, '9100': 7, '9400': 7, '7000':
7, '4300': 6, '6300': 6, '10700': 6, '10500': 6, '7500': 5, '8300': 5, '790
0': 5, '8600': 5, '5600': 5, '10200': 5, '5000': 5, '8100': 5, '9500': 5, '6
000': 4, '6200': 4, '10300': 4, '7700': 4, '5500': 4, '10400': 4, '6800': 4,
'6500': 4, '4700': 4, '7300': 3, '4500': 3, '8400': 3, '6400': 3, '4200': 3,
'7400': 3, '8000': 3, '5400': 3, '3800': 2, '11400': 2, '5300': 2, '8500':
2, '14600': 2, '7100': 2, '13200': 2, '9000': 2, '8200': 2, '15200': 2, '124
00': 2, '12800': 2, '8800': 2, '5700': 2, '9300': 2, '6600': 2, '12100': 1,
'12200': 1, '18900': 1, '21600': 1, '11300': 1, '\t6200': 1, '11800': 1, '12
500': 1, '11900': 1, '12700': 1, '13600': 1, '14900': 1, '16300': 1, '\t840
0': 1, '10900': 1, '2200': 1, '11200': 1, '19100': 1, '\t?': 1, '12300': 1,
'16700': 1, '2600': 1, '26400': 1, '4900': 1, '12000': 1, '15700': 1, '410
0': 1, '11500': 1, '10800': 1, '9900': 1, '5200': 1, '5900': 1, '9700': 1,
'5100': 1})
```

\*\*\*\*\*

Columns : anemia

```
Counter({'no': 339, 'yes': 60, nan: 1})
```

\*\*\*\*\*

Columns : diabetesmellitus

```
Counter({'no': 258, 'yes': 134, '\tno': 3, '\tyes': 2, nan: 2, ' yes': 1})
```

\*\*\*\*\*

Columns : coronary\_artery\_disease

```
Counter({'no': 362, 'yes': 34, '\tno': 2, nan: 2})
```

\*\*\*\*\*

Columns : pus\_cell\_clumps

```
Counter({'notpresent': 354, 'present': 42, nan: 4})
```

\*\*\*\*\*

## 4. Understanding Data Type And Summary Of Features

### Removing the Columns which are not Categorical





In [105]:

```
contcols.remove('specific_gravity')
contcols.remove('albumin')
contcols.remove('sugar')
print(contcols)
```

```
{'blood_urea', 'age', 'potassium', 'hemoglobin', 'sodium', 'blood glucose random', 'serum_creatinine', 'blood_pressure'}
```

## 5. Understanding Data Type And Summary Of Features

### Adding the Columns which are continuous

In [106]:

```
contcols.add('red_blood_cell_count')
contcols.add('packed_cell_volume')
contcols.add('white_blood_cell_count')
print(contcols)
```

```
{'blood_urea', 'packed_cell_volume', 'age', 'potassium', 'hemoglobin', 'sodium', 'blood glucose random', 'red_blood_cell_count', 'white_blood_cell_count', 'serum_creatinine', 'blood_pressure'}
```

### Adding the Columns which are categorical

In [107]:

```
catcols.add('specific_gravity')
catcols.add('albumin')
catcols.add('sugar')
print(catcols)
```

```
{'class', 'red_blood_cells', 'pus_cell', 'appetite', 'bacteria', 'albumin', 'pedal_edema', 'sugar', 'hypertension', 'anemia', 'specific_gravity', 'diabetes mellitus', 'coronary_artery_disease', 'pus_cell_clumps'}
```

### Rectification of categorical columns

In [108]:

```
data['coronary_artery_disease']=data.coronary_artery_disease.replace('\tno','no')
c(data['coronary_artery_disease'])
```

Out[108]:

```
Counter({'no': 364, 'yes': 34, nan: 2})
```

In [109]:

```
data['diabetesmellitus']=data.diabetesmellitus.replace(to_replace={'\tno':'no','\tyes':'yes'},value=1, inplace=True)
```

Out[109]:

```
Counter({'yes': 136, 'no': 261, ' yes': 1, nan: 2})
```

## 6. Handling the Missing Values

In [110]:

```
data.isnull().any()
```

Out[110]:

age	True
blood_pressure	True
specific_gravity	True
albumin	True
sugar	True
red_blood_cells	True
pus_cell	True
pus_cell_clumps	True
bacteria	True
blood_glucose_random	True
blood_urea	True
serum_creatinine	True
sodium	True
potassium	True
hemoglobin	True
packed_cell_volume	True
white_blood_cell_count	True
red_blood_cell_count	True
hypertension	True
diabetesmellitus	True
coronary_artery_disease	True
appetite	True
pedal_edema	True
anemia	True
class	False
dtype: bool	

In [111]:

```
data.isnull().sum()
```

Out[111]:

```
age                9
blood_pressure     12
specific_gravity   47
albumin            46
sugar              49
red_blood_cells    152
pus_cell           65
pus_cell_clumps    4
bacteria           4
blood_glucose_random 44
blood_urea         19
serum_creatinine   17
sodium             87
potassium          88
hemoglobin         52
packed_cell_volume 70
white_blood_cell_count 105
red_blood_cell_count 130
hypertension       2
diabetesmellitus   2
coronary_artery_disease 2
appetite           1
pedal_edema        1
anemia             1
class              0
dtype: int64
```

In [112]:

```
data.packed_cell_volume = pd.to_numeric(data.packed_cell_volume, errors='coerce')
data.white_blood_cell_count = pd.to_numeric(data.white_blood_cell_count, errors='coerce')
data.red_blood_cell_count = pd.to_numeric(data.red_blood_cell_count, errors='coerce')
```

## 7. Replacing the Missing (Null) Values

In [113]:

```
data['blood_glucose_random'].fillna(data['blood glucose random'].mean(),inplace=True)
data['blood_pressure'].fillna(data['blood_pressure'].mean(),inplace=True)
data['blood_urea'].fillna(data['blood_urea'].mean(),inplace=True)
data['hemoglobin'].fillna(data['hemoglobin'].mean(),inplace=True)
data['packed_cell_volume'].fillna(data['packed_cell_volume'].mean(),inplace=True)
data['potassium'].fillna(data['potassium'].mean(),inplace=True)
data['red_blood_cell_count'].fillna(data['red_blood_cell_count'].mean(),inplace=True)
data['serum_creatinine'].fillna(data['serum_creatinine'].mean(),inplace=True)
data['sodium'].fillna(data['sodium'].mean(),inplace=True)
data['white_blood_cell_count'].fillna(data['white_blood_cell_count'].mean(),inplace=True)
```

In [114]:

```

data['age'].fillna(data['age'].mode()[0],inplace=True)
data['hypertension'].fillna(data['hypertension'].mode()[0],inplace=True)
data['pus_cell_clumps'].fillna(data['pus_cell_clumps'].mode()[0],inplace=True)
data['appetite'].fillna(data['appetite'].mode()[0],inplace=True)
data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)
data['pus_cell'].fillna(data['pus_cell'].mode()[0],inplace=True)
data['red_blood_cells'].fillna(data['red_blood_cells'].mode()[0],inplace=True)
data['coronary_artery_disease'].fillna(data['coronary_artery_disease'].mode()[0],inplace=True)
data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)
data['anemia'].fillna(data['anemia'].mode()[0],inplace=True)
data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)
data['diabetesmellitus'].fillna(data['diabetesmellitus'].mode()[0],inplace=True)
data['pedal_edema'].fillna(data['pedal_edema'].mode()[0],inplace=True)
data['specific_gravity'].fillna(data['specific_gravity'].mode()[0],inplace=True)

```

In [115]:

```
data.isnull().any()
```

Out[115]:

```

age                False
blood_pressure     False
specific_gravity   False
albumin            False
sugar              False
red_blood_cells    False
pus_cell           False
pus_cell_clumps    False
bacteria           False
blood glucose random  False
blood_urea         False
serum_creatinine   False
sodium             False
potassium          False
hemoglobin         False
packed_cell_volume False
white_blood_cell_count False
red_blood_cell_count False
hypertension       False
diabetesmellitus   False
coronary_artery_disease False
appetite           False
pedal_edema        False
anemia             False
class              False
dtype: bool

```

## 8. Label Encoding

In [116]:

```
for i in catcols:
    print("Label Encoding of: ",i)
    LEi = LabelEncoder()
    print(c(data[i]))
    data[i]=LEi.fit_transform(data[i])
    print(c(data[i]))
    print(""*100)
```

```
Label Encoding of:  class
Counter({'ckd': 250, 'notckd': 150})
Counter({0: 250, 1: 150})
*****
*****

Label Encoding of:  red_blood_cells
Counter({'normal': 353, 'abnormal': 47})
Counter({1: 353, 0: 47})
*****
*****

Label Encoding of:  pus_cell
Counter({'normal': 324, 'abnormal': 76})
Counter({1: 324, 0: 76})
*****
*****

Label Encoding of:  appetite
Counter({'good': 318, 'poor': 82})
Counter({0: 318, 1: 82})
*****
*****

Label Encoding of:  bacteria
Counter({'notpresent': 378, 'present': 22})
Counter({0: 378, 1: 22})
*****
*****

Label Encoding of:  albumin
Counter({0.0: 245, 1.0: 44, 2.0: 43, 3.0: 43, 4.0: 24, 5.0: 1})
Counter({0: 245, 1: 44, 2: 43, 3: 43, 4: 24, 5: 1})
*****
*****

Label Encoding of:  pedal_edema
Counter({'no': 324, 'yes': 76})
Counter({0: 324, 1: 76})
*****
*****

Label Encoding of:  sugar
Counter({0.0: 339, 2.0: 18, 3.0: 14, 4.0: 13, 1.0: 13, 5.0: 3})
Counter({0: 339, 2: 18, 3: 14, 4: 13, 1: 13, 5: 3})
*****
*****

Label Encoding of:  hypertension
Counter({'no': 253, 'yes': 147})
Counter({0: 253, 1: 147})
*****
*****

Label Encoding of:  anemia
Counter({'no': 340, 'yes': 60})
Counter({0: 340, 1: 60})
*****
*****
```

```

Label Encoding of:  specific_gravity
Counter({1.02: 153, 1.01: 84, 1.025: 81, 1.015: 75, 1.005: 7})
Counter({3: 153, 1: 84, 4: 81, 2: 75, 0: 7})
*****
*****

Label Encoding of:  diabetesmellitus
Counter({'no': 263, 'yes': 136, ' yes': 1})
Counter({1: 263, 2: 136, 0: 1})
*****
*****

Label Encoding of:  coronary_artery_disease
Counter({'no': 366, 'yes': 34})
Counter({0: 366, 1: 34})
*****
*****

Label Encoding of:  pus_cell_clumps
Counter({'notpresent': 358, 'present': 42})
Counter({0: 358, 1: 42})
*****
*****

```

## 9.Splitting The Dataset Into Dependent And Independent Variable

In [132]:

```

selcols=['red_blood_cells','pus_cell','blood glucose random','blood_urea','pedal_edema','an
x=pd.DataFrame(data,columns=selcols)
y=pd.DataFrame(data,columns=['class'])
print(x.shape)
print(y.shape)

```

```

(400, 8)
(400, 1)

```

## 10.Splitting The Dataset Into Train Set And Test Set

In [133]:

```

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)

```

```

(320, 8)
(320, 1)
(80, 8)
(80, 1)

```

## 11. Building a Machine Learning Model

In [146]:

```
from sklearn.linear_model import LogisticRegression
lgr=LogisticRegression()
lgr.fit(x_train, y_train)
```

G:\Anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
return f(*args, **kwargs)
```

Out[146]:

LogisticRegression()

## 12. Test the Model

In [151]:

```
y_pred=lgr.predict(x_test)
y_pred1=lgr.predict([[129,99,1,0,0,1,0,1]])
print(y_pred)
print(c(y_pred))
print(y_pred1)
```

```
[0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1
 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 0 1
 0 1 0 1 1 0]
Counter({0: 47, 1: 33})
[1]
```

## 13. Model Evaluation

In [152]:

```
accuracy_score(y_test,y_pred)
```

Out[152]:

0.9125

In [153]:

```
confusion_mat = confusion_matrix(y_test,y_pred)
confusion_mat
```

Out[153]:

```
array([[47,  7],
       [ 0, 26]], dtype=int64)
```

## 14. Saving the Model

In [154]:

```
pickle.dump(lgr,open('CKD.pkl','wb'))
```

In [ ]: