SPRINT 1 - TEAM ID PNT2022TMID12917

Early Detection of Chronic Kidney Disease using Machine Learning

DATE: 5th November 2022

PROCESSING THE DATASET

1. Importing the necessary libraries

In [1]:

```
import pandas as pd
import numpy as mp
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 [2]:

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

In [3]:

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

Out[3]:

	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 [4]:

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

Out[4]:

	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 [5]:

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

In [6]:

```
data.columns
```

Out[6]:

```
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 [9]:

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

In [10]:

```
data.columns
```

Out[10]:

```
Index(['age', 'blood_pressure', 'specific_gravity', 'albumin', 'sugar',
       'red_blood_cells', 'pus_cell', 'pus_cel_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 [11]:

```
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_cel_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 [12]:

```
data['class'].unique()
```

Out[12]:

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

Rectifying the target column

In [13]:

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

Out[13]:

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

Categorical Columns

In [14]:

```
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_bl
ood_cell_count', 'anemia', 'pus_cel_clumps', 'diabetesmellitus', 'coronary_a
rtery_disease'}
```

In [15]:

```
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': 1
4, '45': 13, '42': 13, '32': 12, '36': 12, '33': 12, '28': 12, '50': 12, '3
7': 11, '34': 11, '35': 9, '29': 9, '30': 9, '46': 9, '31': 8, '39': 7, '2
4': 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 : pus_cel_clumps
Counter({'notpresent': 354, 'present': 42, nan: 4})
****************************************************************************
****************************************

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})
****************************************************************************
****************************************
```

4. Understanding Data Type And Summary Of Features

Removing the Columns which are not Categorical

In [16]:

```
catcols.remove('red_blood_cell_count')
catcols.remove('packed_cell_volume')
catcols.remove('white_blood_cell_count')
print(catcols)
```

```
{'class', 'red_blood_cells', 'pus_cell', 'appetite', 'bacteria', 'pedal_edem
a', 'hypertension', 'anemia', 'pus_cel_clumps', 'diabetesmellitus', 'coronar
y_artery_disease'}
```

Numerical Columns

In [17]:

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

```
{'blood_urea', 'age', 'potassium', 'hemoglobin', 'sodium', 'sugar', 'albumi
n', 'blood glucose random', 'specific_gravity', 'serum_creatinine', 'blood_p
ressure'}
```

In [18]:

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

```
Continuous Columns : blood glucose random
Counter({99.0: 10, 100.0: 9, 93.0: 9, 107.0: 8, 117.0: 6, 140.0: 6, 92.0:
6, 109.0: 6, 131.0: 6, 130.0: 6, 70.0: 5, 114.0: 5, 95.0: 5, 123.0: 5, 12
4.0: 5, 102.0: 5, 132.0: 5, 104.0: 5, 125.0: 5, 122.0: 5, 121.0: 4, 106.0:
4, 76.0: 4, 91.0: 4, 129.0: 4, 133.0: 4, 94.0: 4, 88.0: 4, 118.0: 4, 139.
0: 4, 111.0: 4, 113.0: 4, 120.0: 4, 119.0: 4, 74.0: 3, 108.0: 3, 171.0: 3,
137.0: 3, 79.0: 3, 150.0: 3, 112.0: 3, 127.0: 3, 219.0: 3, 172.0: 3, 89.0:
3, 128.0: 3, 214.0: 3, 105.0: 3, 78.0: 3, 103.0: 3, 82.0: 3, 97.0: 3, 81.
0: 3, 138.0: 2, 490.0: 2, 208.0: 2, 98.0: 2, 204.0: 2, 207.0: 2, 144.0: 2,
253.0: 2, 141.0: 2, 86.0: 2, 360.0: 2, 163.0: 2, 158.0: 2, 165.0: 2, 169.
0: 2, 210.0: 2, 101.0: 2, 153.0: 2, 213.0: 2, 424.0: 2, 303.0: 2, 192.0:
2, 80.0: 2, 110.0: 2, 96.0: 2, 85.0: 2, 83.0: 2, 75.0: 2, nan: 1, 423.0:
1, 410.0: 1, 380.0: 1, 157.0: 1, 263.0: 1, 173.0: 1, nan: 1, nan: 1, nan:
1, 156.0: 1, 264.0: 1, nan: 1, 159.0: 1, 270.0: 1, nan: 1, nan: 1, nan: 1,
162.0: 1, nan: 1, 246.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, 182.0: 1, 146.
0: 1, nan: 1, 425.0: 1, 250.0: 1, nan: 1, nan: 1, nan: 1, 415.0: 1, 251.0:
1, 280.0: 1, 295.0: 1, 298.0: 1, 226.0: 1, 143.0: 1, 115.0: 1, 297.0: 1, 2
33.0: 1, 294.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, 323.0: 1, nan: 1, nan:
1, 90.0: 1, 308.0: 1, 224.0: 1, nan: 1, 268.0: 1, nan: 1, 256.0: 1, nan:
1, 84.0: 1, nan: 1, 288.0: 1, 273.0: 1, 242.0: 1, 148.0: 1, nan: 1, 160.0:
```

Removing the Columns which are not Numerical

In [19]:

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

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

5. Understanding Data Type And Summary Of Features

Adding the Columns which are continuous

In [20]:

```
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', 'sodi
um', 'blood glucose random', 'red_blood_cell_count', 'white_blood_cell_coun
t', 'serum_creatinine', 'blood_pressure'}
```

Adding the Columns which are categorical

In [21]:

```
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', 'pus_c
el_clumps', 'diabetesmellitus', 'coronary_artery_disease'}
```

Rectification of categorical columns

In [22]:

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

Out[22]:

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

In [23]:

```
data['diabetesmellitus']=data.diabetesmellitus.replace(to_replace={'\tno':'no','\tyes':'yes
c(data['diabetesmellitus'])
```

Out[23]:

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

6. Handling the Missing Values

In [24]:

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

Out[24]:

```
age                          True
blood_pressure               True
specific_gravity             True
albumin                      True
sugar                        True
red_blood_cells              True
pus_cell                     True
pus_cel_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 [25]:

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

Out[25]:

```
age                             9
blood_pressure                 12
specific_gravity               47
albumin                        46
sugar                          49
red_blood_cells               152
pus_cell                       65
pus_cel_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 [26]:

```
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 [40]:

```
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 [43]:

```
data['age'].fillna(data['age'].mode()[0],inplace=True)
data['hypertension'].fillna(data['hypertension'].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=Tr
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 [47]:

```
data['pus_cel_clumps'].fillna(data['pus_cel_clumps'].mean(),inplace=True)
```

In [48]:

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

Out[48]:

```
age                       False
blood_pressure            False
specific_gravity          False
albumin                   False
sugar                     False
red_blood_cells           False
pus_cell                  False
pus_cel_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 [49]:

```
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({0: 250, 1: 150})
Counter({0: 250, 1: 150})
****************************************************************************************************
Label Encoding of:  red_blood_cells
Counter({1: 353, 0: 47})
Counter({1: 353, 0: 47})
****************************************************************************************************
Label Encoding of:  pus_cell
Counter({1: 324, 0: 76})
Counter({1: 324, 0: 76})
****************************************************************************************************
Label Encoding of:  appetite
Counter({0: 318, 1: 82})
Counter({0: 318, 1: 82})
****************************************************************************************************
Label Encoding of:  bacteria
Counter({0: 378, 1: 22})
Counter({0: 378, 1: 22})
****************************************************************************************************
Label Encoding of:  albumin
Counter({0: 245, 1: 44, 2: 43, 3: 43, 4: 24, 5: 1})
Counter({0: 245, 1: 44, 2: 43, 3: 43, 4: 24, 5: 1})
****************************************************************************************************
Label Encoding of:  pedal_edema
Counter({0: 324, 1: 76})
Counter({0: 324, 1: 76})
****************************************************************************************************
Label Encoding of:  sugar
Counter({0: 339, 2: 18, 3: 14, 4: 13, 1: 13, 5: 3})
Counter({0: 339, 2: 18, 3: 14, 4: 13, 1: 13, 5: 3})
****************************************************************************************************
Label Encoding of:  hypertension
Counter({0: 253, 1: 147})
Counter({0: 253, 1: 147})
****************************************************************************************************
Label Encoding of:  anemia
Counter({0: 340, 1: 60})
Counter({0: 340, 1: 60})
****************************************************************************************************
```

```
Label Encoding of:  specific_gravity
Counter({3: 153, 1: 84, 4: 81, 2: 75, 0: 7})
Counter({3: 153, 1: 84, 4: 81, 2: 75, 0: 7})
**************************************************************************
************************
Label Encoding of:  pus_cel_clumps
Counter({0: 354, 1: 42, 2: 4})
Counter({0: 354, 1: 42, 2: 4})
**************************************************************************
************************
Label Encoding of:  diabetesmellitus
Counter({1: 263, 2: 136, 0: 1})
Counter({1: 263, 2: 136, 0: 1})
**************************************************************************
************************
Label Encoding of:  coronary_artery_disease
Counter({0: 366, 1: 34})
Counter({0: 366, 1: 34})
**************************************************************************
************************
```

9.Splitting The Dataset Into Dependent And Independent Variable

In [50]:

```
selcols=['red_blood_cells','pus_cell','blood glucose random','blood_urea','pedal_enema','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 [53]:

```
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)
```

In [ ]: