**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

In [3]:

**import** os**,** types

**import** pandas **as** pd

**from** botocore.client **import** Config

**import** ibm\_boto3

**def** \_\_iter\_\_(self): **return** 0

*# @hidden\_cell*

*# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.*

*# You might want to remove those credentials before you share the notebook.*

cos\_client **=** ibm\_boto3**.**client(service\_name**=**'s3',

ibm\_api\_key\_id**=**'6Hn-g1WK1ZmzhRizR5NnrRtXWsoHSNBcRMeWAWiTbzDh',

ibm\_auth\_endpoint**=**"https://iam.cloud.ibm.com/oidc/token",

config**=**Config(signature\_version**=**'oauth'),

endpoint\_url**=**'https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket **=** 'chronicmodel-donotdelete-pr-e7m5bqwsliiphj'

object\_key **=** 'kidney\_disease.csv'

body **=** cos\_client**.**get\_object(Bucket**=**bucket,Key**=**object\_key)['Body']

*# add missing \_\_iter\_\_ method, so pandas accepts body as file-like object*

**if** **not** hasattr(body, "\_\_iter\_\_"): body**.**\_\_iter\_\_ **=** types**.**MethodType( \_\_iter\_\_, body )

data **=** pd**.**read\_csv(body)

data**.**head()

Out[3]:

|  | **id** | **age** | **bp** | **sg** | **al** | **su** | **rbc** | **pc** | **pcc** | **ba** | **...** | **pcv** | **wc** | **rc** | **htn** | **dm** | **cad** | **appet** | **pe** | **ane** | **classification** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | 48.0 | 80.0 | 1.020 | 1.0 | 0.0 | NaN | normal | notpresent | notpresent | ... | 44 | 7800 | 5.2 | yes | yes | no | good | no | no | ckd |
| **1** | 1 | 7.0 | 50.0 | 1.020 | 4.0 | 0.0 | NaN | normal | notpresent | notpresent | ... | 38 | 6000 | NaN | no | no | no | good | no | no | ckd |
| **2** | 2 | 62.0 | 80.0 | 1.010 | 2.0 | 3.0 | normal | normal | notpresent | notpresent | ... | 31 | 7500 | NaN | no | yes | no | poor | no | yes | ckd |
| **3** | 3 | 48.0 | 70.0 | 1.005 | 4.0 | 0.0 | normal | abnormal | present | notpresent | ... | 32 | 6700 | 3.9 | yes | no | no | poor | yes | yes | ckd |
| **4** | 4 | 51.0 | 80.0 | 1.010 | 2.0 | 0.0 | normal | normal | notpresent | notpresent | ... | 35 | 7300 | 4.6 | no | no | no | good | no | no | ckd |

5 rows × 26 columns

In [4]:

data**.**tail()

Out[4]:

|  | **id** | **age** | **bp** | **sg** | **al** | **su** | **rbc** | **pc** | **pcc** | **ba** | **...** | **pcv** | **wc** | **rc** | **htn** | **dm** | **cad** | **appet** | **pe** | **ane** | **classification** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **395** | 395 | 55.0 | 80.0 | 1.020 | 0.0 | 0.0 | normal | normal | notpresent | notpresent | ... | 47 | 6700 | 4.9 | no | no | no | good | no | no | notckd |
| **396** | 396 | 42.0 | 70.0 | 1.025 | 0.0 | 0.0 | normal | normal | notpresent | notpresent | ... | 54 | 7800 | 6.2 | no | no | no | good | no | no | notckd |
| **397** | 397 | 12.0 | 80.0 | 1.020 | 0.0 | 0.0 | normal | normal | notpresent | notpresent | ... | 49 | 6600 | 5.4 | no | no | no | good | no | no | notckd |
| **398** | 398 | 17.0 | 60.0 | 1.025 | 0.0 | 0.0 | normal | normal | notpresent | notpresent | ... | 51 | 7200 | 5.9 | no | no | no | good | no | no | notckd |
| **399** | 399 | 58.0 | 80.0 | 1.025 | 0.0 | 0.0 | normal | normal | notpresent | notpresent | ... | 53 | 6800 | 6.1 | no | no | no | good | no | no | notckd |

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 [7]:

data**.**columns**=**['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']

In [8]:

data**.**info()

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

In [9]:

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

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

Out[9]:

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

In [10]:

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

print(catcols)

{'pus\_cell', 'coronary\_artery\_disease', 'class', 'packed\_cell\_volume', 'white\_blood\_cell\_count', 'red\_blood\_cell\_count', 'bacteria', 'anemia', 'hypertension', 'pedal\_edema', 'appetite', 'diabetesmellitus', 'red\_blood\_cells', 'pus\_cell\_clumps'}

In [11]:

**for** i **in** catcols:

print("columns :",i)

print(c(data[i]))

print('\*'**\***120**+**'\n')

columns : pus\_cell

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

columns : coronary\_artery\_disease

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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 : 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, '7900': 5, '8600': 5, '5600': 5, '10200': 5, '5000': 5, '8100': 5, '9500': 5, '6000': 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, '12400': 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, '12500': 1, '11900': 1, '12700': 1, '13600': 1, '14900': 1, '16300': 1, '\t8400': 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, '4100': 1, '11500': 1, '10800': 1, '9900': 1, '5200': 1, '5900': 1, '9700': 1, '5100': 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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 : bacteria

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

columns : anemia

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

columns : hypertension

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

columns : pedal\_edema

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

columns : appetite

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

columns : diabetesmellitus

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

columns : red\_blood\_cells

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

columns : pus\_cell\_clumps

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In [12]:

catcols**.**remove('red\_blood\_cell\_count')

catcols**.**remove('packed\_cell\_volume')

catcols**.**remove('white\_blood\_cell\_count')

print(catcols)

{'pus\_cell', 'coronary\_artery\_disease', 'class', 'bacteria', 'anemia', 'hypertension', 'pedal\_edema', 'appetite', 'diabetesmellitus', 'red\_blood\_cells', 'pus\_cell\_clumps'}

In [13]:

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

print(contcols)

{'blood\_pressure', 'age', 'hemoglobin', 'blood\_urea', 'specific\_gravity', 'sodium', 'blood glucose random', 'sugar', 'potassium', 'serum\_creatinine', 'albumin'}

In [14]:

**for** i **in** contcols:

print("continous columns :",i)

print(c(data[i]))

print('\*'**\***120**+**'\n')

continous columns : blood\_pressure

Counter({80.0: 116, 70.0: 112, 60.0: 71, 90.0: 53, 100.0: 25, 50.0: 5, 110.0: 3, nan: 1, nan: 1, 140.0: 1, 180.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 120.0: 1, nan: 1, nan: 1, nan: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : age

Counter({60.0: 19, 65.0: 17, 48.0: 12, 50.0: 12, 55.0: 12, 47.0: 11, 62.0: 10, 45.0: 10, 54.0: 10, 59.0: 10, 56.0: 10, 61.0: 9, 70.0: 9, 46.0: 9, 34.0: 9, 68.0: 8, 73.0: 8, 64.0: 8, 71.0: 8, 57.0: 8, 63.0: 7, 72.0: 7, 67.0: 7, 30.0: 7, 42.0: 6, 69.0: 6, 35.0: 6, 44.0: 6, 43.0: 6, 33.0: 6, 51.0: 5, 52.0: 5, 53.0: 5, 75.0: 5, 76.0: 5, 58.0: 5, 41.0: 5, 66.0: 5, 24.0: 4, 40.0: 4, 39.0: 4, 80.0: 4, 23.0: 4, 74.0: 3, 38.0: 3, 17.0: 3, 8.0: 3, 32.0: 3, 37.0: 3, 25.0: 3, 29.0: 3, 21.0: 2, 15.0: 2, 5.0: 2, 12.0: 2, 49.0: 2, 19.0: 2, 36.0: 2, 20.0: 2, 28.0: 2, 7.0: 1, nan: 1, 82.0: 1, 11.0: 1, 26.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, 81.0: 1, 14.0: 1, 27.0: 1, nan: 1, 83.0: 1, 4.0: 1, 3.0: 1, 6.0: 1, nan: 1, 90.0: 1, 78.0: 1, nan: 1, 2.0: 1, nan: 1, 22.0: 1, 79.0: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : hemoglobin

Counter({15.0: 16, 10.9: 8, 9.8: 7, 11.1: 7, 13.0: 7, 13.6: 7, 11.3: 6, 10.3: 6, 12.0: 6, 13.9: 6, 15.4: 5, 11.2: 5, 10.8: 5, 9.7: 5, 12.6: 5, 7.9: 5, 10.0: 5, 14.0: 5, 14.3: 5, 14.8: 5, 12.2: 4, 12.4: 4, 12.5: 4, 15.2: 4, 9.1: 4, 11.9: 4, 13.5: 4, 16.1: 4, 14.1: 4, 13.2: 4, 13.8: 4, 13.7: 4, 13.4: 4, 17.0: 4, 15.5: 4, 15.8: 4, 9.6: 3, 11.6: 3, 9.5: 3, 9.4: 3, 12.7: 3, 9.9: 3, 10.1: 3, 8.6: 3, 11.0: 3, 15.6: 3, 8.1: 3, 8.3: 3, 10.4: 3, 11.8: 3, 11.4: 3, 11.5: 3, 15.9: 3, 14.5: 3, 16.2: 3, 14.4: 3, 14.2: 3, 16.3: 3, 16.5: 3, 15.7: 3, 16.4: 3, 14.9: 3, 15.3: 3, 17.8: 3, 12.1: 2, 9.3: 2, 10.2: 2, 10.5: 2, 6.0: 2, 11.7: 2, 8.0: 2, 12.3: 2, 8.7: 2, 13.1: 2, 8.8: 2, 13.3: 2, 14.6: 2, 16.9: 2, 16.0: 2, 14.7: 2, 16.6: 2, 16.7: 2, 16.8: 2, 15.1: 2, 17.1: 2, 17.2: 2, 17.4: 2, 5.6: 1, 7.6: 1, 7.7: 1, nan: 1, nan: 1, 12.9: 1, nan: 1, nan: 1, nan: 1, nan: 1, 6.6: 1, nan: 1, nan: 1, 7.5: 1, nan: 1, nan: 1, 4.8: 1, nan: 1, nan: 1, 7.1: 1, nan: 1, nan: 1, nan: 1, 9.2: 1, nan: 1, 6.2: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 8.2: 1, nan: 1, nan: 1, 6.1: 1, nan: 1, nan: 1, nan: 1, nan: 1, 8.4: 1, nan: 1, 9.0: 1, nan: 1, nan: 1, 10.6: 1, nan: 1, nan: 1, nan: 1, 10.7: 1, nan: 1, 5.5: 1, nan: 1, 5.8: 1, 6.8: 1, 8.5: 1, 7.3: 1, nan: 1, nan: 1, 12.8: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 6.3: 1, nan: 1, 3.1: 1, nan: 1, 17.3: 1, nan: 1, nan: 1, nan: 1, nan: 1, 17.7: 1, 17.5: 1, nan: 1, 17.6: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : blood\_urea

Counter({46.0: 15, 25.0: 13, 19.0: 11, 40.0: 10, 18.0: 9, 50.0: 9, 15.0: 9, 48.0: 9, 26.0: 8, 27.0: 8, 32.0: 8, 49.0: 8, 36.0: 7, 28.0: 7, 20.0: 7, 17.0: 7, 38.0: 7, 16.0: 7, 30.0: 7, 44.0: 7, 31.0: 6, 45.0: 6, 39.0: 6, 29.0: 6, 24.0: 6, 37.0: 6, 22.0: 6, 23.0: 6, 53.0: 5, 55.0: 5, 33.0: 5, 66.0: 5, 35.0: 5, 42.0: 5, 47.0: 4, 51.0: 4, 34.0: 4, 68.0: 4, 41.0: 4, 60.0: 3, 107.0: 3, 80.0: 3, 96.0: 3, 52.0: 3, 106.0: 3, 125.0: 3, 56.0: 2, 54.0: 2, 72.0: 2, 86.0: 2, 90.0: 2, 87.0: 2, 155.0: 2, 153.0: 2, 77.0: 2, 89.0: 2, 111.0: 2, 73.0: 2, 98.0: 2, 82.0: 2, 132.0: 2, 58.0: 2, 10.0: 2, 162.0: 1, 148.0: 1, 180.0: 1, 163.0: 1, nan: 1, 75.0: 1, 65.0: 1, 103.0: 1, 70.0: 1, 202.0: 1, 114.0: 1, nan: 1, nan: 1, 164.0: 1, 142.0: 1, 391.0: 1, nan: 1, nan: 1, 92.0: 1, 139.0: 1, 85.0: 1, 186.0: 1, 217.0: 1, 88.0: 1, 118.0: 1, 50.1: 1, 71.0: 1, nan: 1, 21.0: 1, 219.0: 1, 166.0: 1, 208.0: 1, 176.0: 1, nan: 1, 145.0: 1, 165.0: 1, 322.0: 1, 235.0: 1, 76.0: 1, nan: 1, nan: 1, 113.0: 1, 1.5: 1, 146.0: 1, 133.0: 1, 137.0: 1, 67.0: 1, 115.0: 1, 223.0: 1, 98.6: 1, 158.0: 1, 94.0: 1, 74.0: 1, nan: 1, 150.0: 1, nan: 1, 61.0: 1, 57.0: 1, nan: 1, 95.0: 1, 191.0: 1, nan: 1, 93.0: 1, 241.0: 1, 64.0: 1, 79.0: 1, 215.0: 1, 309.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : specific\_gravity

Counter({1.02: 106, 1.01: 84, 1.025: 81, 1.015: 75, 1.005: 7, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : sodium

Counter({135.0: 40, 140.0: 25, 141.0: 22, 139.0: 21, 142.0: 20, 138.0: 20, 137.0: 19, 136.0: 17, 150.0: 17, 147.0: 13, 145.0: 11, 132.0: 10, 146.0: 10, 131.0: 9, 144.0: 9, 133.0: 8, 130.0: 7, 134.0: 6, 143.0: 4, 127.0: 3, 124.0: 3, 114.0: 2, 125.0: 2, 128.0: 2, 122.0: 2, 113.0: 2, 120.0: 2, nan: 1, nan: 1, nan: 1, 111.0: 1, nan: 1, 104.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, 4.5: 1, nan: 1, 129.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 163.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 126.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 115.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous 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, 124.0: 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, 233.0: 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: 1, nan: 1, 307.0: 1, 220.0: 1, 447.0: 1, 309.0: 1, 22.0: 1, 261.0: 1, 215.0: 1, 234.0: 1, 352.0: 1, nan: 1, nan: 1, 239.0: 1, nan: 1, nan: 1, 184.0: 1, nan: 1, 252.0: 1, 230.0: 1, 341.0: 1, nan: 1, 255.0: 1, nan: 1, 238.0: 1, 248.0: 1, 241.0: 1, 269.0: 1, nan: 1, nan: 1, 201.0: 1, 203.0: 1, 463.0: 1, 176.0: 1, nan: 1, nan: 1, 116.0: 1, nan: 1, nan: 1, nan: 1, 134.0: 1, 87.0: 1, nan: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : sugar

Counter({0.0: 290, 2.0: 18, 3.0: 14, 4.0: 13, 1.0: 13, 5.0: 3, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : potassium

Counter({5.0: 30, 3.5: 30, 4.9: 27, 4.7: 17, 4.8: 16, 4.0: 14, 4.2: 14, 4.1: 14, 3.8: 14, 3.9: 14, 4.4: 14, 4.5: 13, 3.7: 12, 4.3: 12, 3.6: 8, 4.6: 7, 3.4: 5, 5.2: 5, 5.7: 4, 5.3: 4, 3.2: 3, 5.5: 3, 2.9: 3, 5.4: 3, 6.3: 3, 3.3: 3, 2.5: 2, 5.8: 2, 5.9: 2, 5.6: 2, 3.0: 2, 6.5: 2, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 6.4: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 6.6: 1, 39.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 7.6: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 47.0: 1, nan: 1, 5.1: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 2.8: 1, nan: 1, nan: 1, nan: 1, nan: 1, 2.7: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : serum\_creatinine

Counter({1.2: 40, 1.1: 24, 1.0: 23, 0.5: 23, 0.7: 22, 0.9: 22, 0.6: 18, 0.8: 17, 2.2: 10, 1.5: 9, 1.7: 9, 1.3: 8, 1.6: 8, 1.8: 7, 1.4: 7, 2.5: 7, 2.8: 7, 1.9: 6, 2.7: 5, 2.1: 5, 2.0: 5, 3.2: 5, 3.3: 5, 3.9: 4, 7.3: 4, 4.0: 3, 2.4: 3, 3.4: 3, 2.9: 3, 5.3: 3, 2.3: 3, 7.2: 2, 4.6: 2, 4.1: 2, 5.2: 2, 6.3: 2, 3.0: 2, 6.1: 2, 6.7: 2, 5.6: 2, 6.5: 2, 4.4: 2, 6.0: 2, 3.8: 1, 24.0: 1, 9.6: 1, 76.0: 1, 7.7: 1, nan: 1, 10.8: 1, 5.9: 1, 3.25: 1, nan: 1, 9.7: 1, 6.4: 1, 32.0: 1, nan: 1, nan: 1, 8.5: 1, 15.0: 1, 3.6: 1, 10.2: 1, 11.5: 1, nan: 1, 12.2: 1, 9.2: 1, 13.8: 1, 16.9: 1, 7.1: 1, 18.0: 1, 13.0: 1, 48.1: 1, 14.2: 1, 16.4: 1, nan: 1, nan: 1, 2.6: 1, 7.5: 1, 4.3: 1, 18.1: 1, 11.8: 1, 9.3: 1, 6.8: 1, 13.5: 1, nan: 1, 12.8: 1, 11.9: 1, nan: 1, nan: 1, nan: 1, 12.0: 1, nan: 1, 13.4: 1, 15.2: 1, 13.3: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 0.4: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continous columns : albumin

Counter({0.0: 199, 1.0: 44, 2.0: 43, 3.0: 43, 4.0: 24, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, 5.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In [15]:

contcols**.**remove('specific\_gravity')

contcols**.**remove('albumin')

contcols**.**remove('sugar')

print(contcols)

{'blood\_pressure', 'age', 'hemoglobin', 'blood\_urea', 'sodium', 'blood glucose random', 'potassium', 'serum\_creatinine'}

In [16]:

contcols**.**add('red\_blood\_cell\_count')

contcols**.**add('packed\_cell\_volume')

contcols**.**add('white\_blood\_cell\_count')

print(contcols)

{'blood\_pressure', 'age', 'hemoglobin', 'blood\_urea', 'red\_blood\_cell\_count', 'packed\_cell\_volume', 'white\_blood\_cell\_count', 'sodium', 'blood glucose random', 'potassium', 'serum\_creatinine'}

In [17]:

catcols**.**add('specific\_gravity')

catcols**.**add('albumin')

catcols**.**add('sugar')

print(catcols)

{'pus\_cell', 'coronary\_artery\_disease', 'class', 'bacteria', 'specific\_gravity', 'albumin', 'anemia', 'sugar', 'hypertension', 'pedal\_edema', 'appetite', 'diabetesmellitus', 'red\_blood\_cells', 'pus\_cell\_clumps'}

In [18]:

data['coronary\_artery\_disease']**.**unique()

Out[18]:

array(['no', 'yes', '\tno', nan], dtype=object)

In [19]:

data['diabetesmellitus']**.**unique()

Out[19]:

array(['yes', 'no', ' yes', '\tno', '\tyes', nan], dtype=object)

In [20]:

data['coronary\_artery\_disease']**=**data**.**coronary\_artery\_disease**.**replace('\tno','no')

c(data['coronary\_artery\_disease'])

Out[20]:

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

In [21]:

data['diabetesmellitus']**=**data**.**diabetesmellitus**.**replace(to\_replace**=**{'\tno':'no','\tyes':'yes',' yes':'yes'})

c(data['diabetesmellitus'])

Out[21]:

Counter({'yes': 137, 'no': 261, nan: 2})

In [22]:

data**.**isnull()**.**sum()

Out[22]:

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 [23]:

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')

In [24]:

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 [25]:

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 [26]:

**from** sklearn.preprocessing **import** LabelEncoder

**for** i **in** catcols:

("LABEL ENCODING OF:",i)

LEi**=**LabelEncoder()

print(c(data[i]))

data[i]**=**LEi**.**fit\_transform(data[i])

print(c(data[i]))

print("\*"**\***100)

Counter({'normal': 324, 'abnormal': 76})

Counter({1: 324, 0: 76})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'no': 366, 'yes': 34})

Counter({0: 366, 1: 34})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

Counter({0: 250, 1: 150})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'notpresent': 378, 'present': 22})

Counter({0: 378, 1: 22})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'no': 340, 'yes': 60})

Counter({0: 340, 1: 60})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'no': 253, 'yes': 147})

Counter({0: 253, 1: 147})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'no': 324, 'yes': 76})

Counter({0: 324, 1: 76})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'good': 318, 'poor': 82})

Counter({0: 318, 1: 82})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'no': 263, 'yes': 137})

Counter({0: 263, 1: 137})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'normal': 353, 'abnormal': 47})

Counter({1: 353, 0: 47})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Counter({'notpresent': 358, 'present': 42})

Counter({0: 358, 1: 42})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In [27]:

selcols**=**['age','blood\_urea','blood glucose random','coronary\_artery\_disease','anemia','pus\_cell','red\_blood\_cells',

'diabetesmellitus','pedal\_edema']

x**=**pd**.**DataFrame(data,columns**=**selcols)

y**=**pd**.**DataFrame(data,columns**=**['class'])

print(x**.**shape)

print(y**.**shape)

(400, 9)

(400, 1)

In [28]:

**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, 9)

(320, 1)

(80, 9)

(80, 1)

In [48]:

**from** sklearn.ensemble **import** RandomForestClassifier

model **=** RandomForestClassifier()

model**.**fit(x\_train , y\_train)

prediction **=** model**.**predict(x\_test)

print(prediction)

**from** sklearn.metrics **import** confusion\_matrix

print('RandomForest\n')

print('confusion\_matrix')

print(confusion\_matrix(prediction,y\_test))

print('\n')

print('accuracy\_score')

print(accuracy\_score(prediction,y\_test))

print('\n')

[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 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1

0 0 0 0 1 0]

RandomForest

confusion\_matrix

[[52 1]

[ 2 25]]

accuracy\_score

0.9625

In [49]:

y\_pred**=**model**.**predict(x\_test)

y\_pred1**=**model**.**predict([[40,10,140,0,0,1,1,0,0],[62,53,423,0,1,1,1,1,0],[61.0,28.0,100,0,0,1,1,1,0],

[48,36.0,121.000000,0,0,1,1,1,0],[40,1,1,140,10,0,0,0,0]])

print(y\_pred1)

c(y\_pred)

*#print(x\_test.values)*

*#print(lgr)*

[1 0 0 0 0]

Out[49]:

Counter({0: 53, 1: 27})

In [50]:

accuracy\_score(y\_test,y\_pred)

Out[50]:

0.9625

In [52]:

print(x\_test)

age blood\_urea blood glucose random coronary\_artery\_disease anemia \

94 65.0 66.0 93.000000 0 0

32 61.0 39.0 159.000000 0 0

225 60.0 95.0 490.000000 0 0

157 62.0 42.0 122.000000 0 0

356 34.0 38.0 87.000000 0 0

.. ... ... ... ... ...

188 8.0 66.0 80.000000 0 0

223 71.0 30.0 303.000000 0 0

65 44.0 20.0 148.036517 0 0

262 55.0 18.0 118.000000 0 0

186 8.0 46.0 148.036517 0 0

pus\_cell red\_blood\_cells diabetesmellitus pedal\_edema

94 1 1 1 0

32 1 1 1 0

225 1 0 1 0

157 0 1 1 0

356 1 1 0 0

.. ... ... ... ...

188 1 1 0 0

223 1 1 1 0

65 1 1 0 0

262 1 1 0 0

186 1 1 0 1

[80 rows x 9 columns]

**Deploy our Model in IBM cloud**

In [35]:

*#!pip install ibm\_watson\_machine\_learning*

In [53]:

**from** ibm\_watson\_machine\_learning **import** APIClient

wml\_credentials **=** {

"url": "https://us-south.ml.cloud.ibm.com",

"apikey": "EjnR5QWRh\_9zPFHorolJcaYJCPzfYS3xGZeFJlhbtkTS"

}

client **=** APIClient(wml\_credentials)

client

Out[53]:

In [54]:

**def** space\_Name(client, space\_name):

space **=** client**.**spaces**.**get\_details()

*#print(space)*

**return**(next(item **for** item **in** space['resources'] **if** item['entity']['name'] **==** space\_name)['metadata']['id'])

In [55]:

space\_uid **=** space\_Name(client, 'models-ckd')

print("space uid = " **+**space\_uid)

space uid = c3957aa3-dbdd-4199-b24d-d08c0cd9e99f

In [56]:

client**.**set**.**default\_space(space\_uid)

Out[56]:

'SUCCESS'

In [57]:

client**.**software\_specifications**.**list()

----------------------------- ------------------------------------ ----

NAME ASSET\_ID TYPE

default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base

kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base

pytorch-onnx\_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base

scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base

spark-mllib\_3.0-scala\_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base

pytorch-onnx\_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base

ai-function\_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base

shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base

tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base

pytorch\_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base

tensorflow\_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base

runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base

scikit-learn\_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base

default\_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base

pytorch-onnx\_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base

kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base

pytorch-onnx\_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base

tensorflow\_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base

spark-mllib\_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base

tensorflow\_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base

runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base

do\_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base

autoai-ts\_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base

tensorflow\_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base

kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base

pytorch\_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base

spark-mllib\_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base

pytorch-onnx\_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base

spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base

spark-mllib\_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base

xgboost\_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base

pytorch-onnx\_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base

default\_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base

autoai-ts\_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base

autoai-obm\_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base

pmml-3.0\_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base

spark-mllib\_2.4-r\_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base

xgboost\_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base

pytorch-onnx\_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base

autoai-ts\_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base

spark-mllib\_2.4-scala\_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base

spark-mllib\_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base

autoai-obm\_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base

spss-modeler\_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base

cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base

autoai-kb\_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base

pytorch-onnx\_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base

spark-mllib\_2.3-r\_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base

tensorflow\_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base

spss-modeler\_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base

----------------------------- ------------------------------------ ----

Note: Only first 50 records were displayed. To display more use 'limit' parameter.

In [58]:

software\_uid**=**client**.**software\_specifications**.**get\_uid\_by\_name("runtime-22.1-py3.9")

software\_uid

Out[58]:

'12b83a17-24d8-5082-900f-0ab31fbfd3cb'

In [59]:

model\_details **=** client**.**repository**.**store\_model(model**=**model,meta\_props**=**{

client**.**repository**.**ModelMetaNames**.**NAME:"chronic\_kidney\_disease",

client**.**repository**.**ModelMetaNames**.**TYPE:"scikit-learn\_1.0",

client**.**repository**.**ModelMetaNames**.**SOFTWARE\_SPEC\_UID:software\_uid }

)

model\_id **=** client**.**repository**.**get\_model\_uid(model\_details)

This method is deprecated, please use get\_model\_id()

In [60]:

model\_id

Out[60]:

'bde9d6eb-cb46-4bc7-bcef-80f7d856c968'

In [61]:

x\_train**.**columns

Out[61]:

Index(['age', 'blood\_urea', 'blood glucose random', 'coronary\_artery\_disease',

'anemia', 'pus\_cell', 'red\_blood\_cells', 'diabetesmellitus',

'pedal\_edema'],

dtype='object')