

```

In [1]: import pandas as pd

In [2]: import numpy as np

In [3]: from collections import Counter as c

In [4]: import matplotlib.pyplot as plt

In [5]: import seaborn as sns

In [6]: import missingno as msn

In [7]: from sklearn.metrics import accuracy_score, confusion_matrix

In [8]: from sklearn.model_selection import train_test_split

In [9]: from sklearn.preprocessing import LabelEncoder

In [10]: from sklearn.linear_model import LogisticRegression

In [11]: import pickle

In [12]: data=pd.read_csv("F:\SEM 7\ISH Project\kidney_disease.csv")

In [13]: data.head()

```

```

Out[13]:
   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 [14]: data.tail()

Out[14]:
   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 [15]: data.head(10)

Out[15]:
   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  5  60.0  90.0  1.015  3.0  0.0   NaN   NaN  notpresent  notpresent ... 39  7800  4.4  yes  yes  no  good  yes  no      ckd
6  6  68.0  70.0  1.010  0.0  0.0   NaN  normal  notpresent  notpresent ... 36   NaN   NaN  no  no  no  good  no  no      ckd
7  7  24.0   NaN  1.015  2.0  4.0  normal  abnormal  notpresent  notpresent ... 44  6900  5  no  yes  no  good  yes  no      ckd
8  8  52.0  100.0  1.015  3.0  0.0  normal  abnormal  present  notpresent ... 33  9600  4  yes  yes  no  good  no  yes      ckd
9  9  53.0  90.0  1.020  2.0  0.0  abnormal  abnormal  present  notpresent ... 29  12100  3.7  yes  yes  no  poor  no  yes      ckd

10 rows × 26 columns

```

```

In [16]: data.drop(["id"],axis=1,inplace=True)

```

```

In [17]: data.head(10)

```

```

Out[17]:
   age  bp    sg  al  su  rbc    pc    pcc    ba  bgr ... pcv  wc  rc  htn  dm  cad  appet  pe  ane  classification
0  48.0  80.0  1.020  1.0  0.0   NaN  normal  notpresent  notpresent  121.0 ... 44  7800  5.2  yes  yes  no  good  no  no      ckd
1   7.0  50.0  1.020  4.0  0.0   NaN  normal  notpresent  notpresent   NaN ... 38  6000   NaN  no  no  no  good  no  no      ckd
2  62.0  80.0  1.010  2.0  3.0  normal  normal  notpresent  notpresent  423.0 ... 31  7500   NaN  no  yes  no  poor  no  yes      ckd
3  48.0  70.0  1.005  4.0  0.0  normal  abnormal  present  notpresent  117.0 ... 32  6700  3.9  yes  no  no  poor  yes  yes      ckd
4  51.0  80.0  1.010  2.0  0.0  normal  normal  notpresent  notpresent  106.0 ... 35  7300  4.6  no  no  no  good  no  no      ckd
5  60.0  90.0  1.015  3.0  0.0   NaN   NaN  notpresent  notpresent  74.0 ... 39  7800  4.4  yes  yes  no  good  yes  no      ckd
6  68.0  70.0  1.010  0.0  0.0   NaN  normal  notpresent  notpresent  100.0 ... 36   NaN   NaN  no  no  no  good  no  no      ckd
7  24.0   NaN  1.015  2.0  4.0  normal  abnormal  notpresent  notpresent  410.0 ... 44  6900  5  no  yes  no  good  yes  no      ckd
8  52.0  100.0  1.015  3.0  0.0  normal  abnormal  present  notpresent  138.0 ... 33  9600  4  yes  yes  no  good  no  yes      ckd
9  53.0  90.0  1.020  2.0  0.0  abnormal  abnormal  present  notpresent  70.0 ... 29  12100  3.7  yes  yes  no  poor  no  yes      ckd

10 rows × 25 columns

```

```

In [18]: data.columns

Out[18]: Index(['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgn', 'bu',
               'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
               'appet', 'pe', 'ane', 'classification'],
              dtype='object')

In [19]: data.columns=['age','blood_pressure','specific_gravity','albumin','sugar','red_blood_cells','pus_cell','pus_cell_clumps','bacteria','blood']

In [20]: data.columns

Out[20]: 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')

In [21]: data.columns

Out[21]: 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')

In [22]: 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 [23]: data['class'].unique()

Out[23]: array(['ckd', 'ckd\t', 'notckd'], dtype=object)

In [24]: data['class']=data['class'].replace("ckd\t","ckd")
data['class'].unique()

Out[24]: array(['ckd', 'notckd'], dtype=object)

In [25]: catcols=set(data.dtypes[data.dtypes=='O'].index.values)
print(catcols)

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

In [26]: for i in catcols:
print("Columns: "+i)
print(c(data[i]))
print('*'*120+'\n')

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

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

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

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, '17': 1, '14': 1, '18': 1, '15': 1, '21': 1, '20': 1, '\t43': 1, '9': 1})
*****

Columns: red_blood_cell_count
Counter({nan: 130, '5.2': 18, '4.5': 16, '4.9': 14, '4.7': 11, '3.9': 10, '5': 10, '4.8': 10, '4.6': 9, '3.4': 9, '3.7': 8, '6.1': 8, '5.5': 8, '5.9': 8, '3.8': 7, '5.4': 7, '5.8': 7, '5.3': 7, '4': 6, '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': 4, '6.3': 4, '3.5': 3, '3.3': 3, '3': 3, '2.6': 2, '2.8': 2, '2.5': 2, '3.1': 2, '2.1': 2, '2.9': 2, '2.7': 2, '2.3': 1, '8': 1, '2.4': 1, '\t?': 1})
*****

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

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

Columns: hypertension

```


[illegible][illegible][illegible]

```
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, 150.0: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1, nan: 1})
```

[illegible]

```
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))
```

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, 9:5: 4, 5:2: 4, 12:4: 4, 12:5: 4, 15:2: 4, 9:1: 4, 16:1: 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, 11:0: 3, 9:4: 3, 8:1: 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, 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, 15:3: 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, 9:1: 1, 9:2: 1, 12:9: 1, 9:0: 1, 14:1: 1, 14:2: 1, 14:3: 1, 14:4: 1, 14:5: 1, 14:6: 1, 14:7: 1, 14:8: 1, 14:9: 1, 15:0: 1, 15:1: 1, 15:2: 1, 15:3: 1, 15:4: 1, 15:5: 1, 15:6: 1, 15:7: 1, 15:8: 1, 15:9: 1, 16:0: 1, 16:1: 1, 16:2: 1, 16:3: 1, 16:4: 1, 16:5: 1, 16:6: 1, 16:7: 1, 16:8: 1, 16:9: 1, 17:0: 1, 17:1: 1, 17:2: 1, 17:3: 1, 17:4: 1, 17:5: 1, 17:6: 1, 17:7: 1, 17:8: 1, 17:9: 1, 18:0: 1, 18:1: 1, 18:2: 1, 18:3: 1, 18:4: 1, 18:5: 1, 18:6: 1, 18:7: 1, 18:8: 1, 18:9: 1, 19:0: 1, 19:1: 1, 19:2: 1, 19:3: 1, 19:4: 1, 19:5: 1, 19:6: 1, 19:7: 1, 19:8: 1, 19:9: 1, 20:0: 1, 20:1: 1, 20:2: 1, 20:3: 1, 20:4: 1, 20:5: 1, 20:6: 1, 20:7: 1, 20:8: 1, 20:9: 1, 21:0: 1, 21:1: 1, 21:2: 1, 21:3: 1, 21:4: 1, 21:5: 1, 21:6: 1, 21:7: 1, 21:8: 1, 21:9: 1, 22:0: 1, 22:1: 1, 22:2: 1, 22:3: 1, 22:4: 1, 22:5: 1, 22:6: 1, 22:7: 1, 22:8: 1, 22:9: 1, 23:0: 1, 23:1: 1, 23:2: 1, 23:3: 1, 23:4: 1, 23:5: 1, 23:6: 1, 23:7: 1, 23:8: 1, 23:9: 1, 24:0: 1, 24:1: 1, 24:2: 1, 24:3: 1, 24:4: 1, 24:5: 1, 24:6: 1, 24:7: 1, 24:8: 1, 24:9: 1, 25:0: 1, 25:1: 1, 25:2: 1, 25:3: 1, 25:4: 1, 25:5: 1, 25:6: 1, 25:7: 1, 25:8: 1, 25:9: 1, 26:0: 1, 26:1: 1, 26:2: 1, 26:3: 1, 26:4: 1, 26:5: 1, 26:6: 1, 26:7: 1, 26:8: 1, 26:9: 1, 27:0: 1, 27:1: 1, 27:2: 1, 27:3: 1, 27:4: 1, 27:5: 1, 27:6: 1, 27:7: 1, 27:8: 1, 27:9: 1, 28:0: 1, 28:1: 1, 28:2: 1, 28:3: 1, 28:4: 1, 28:5: 1, 28:6: 1, 28:7: 1, 28:8: 1, 28:9: 1, 29:0: 1, 29:1: 1, 29:2: 1, 29:3: 1, 29:4: 1, 29:5: 1, 29:6: 1, 29:7: 1, 29:8: 1, 29:9: 1, 30:0: 1, 30:1: 1, 30:2: 1, 30:3: 1, 30:4: 1, 30:5: 1, 30:6: 1, 30:7: 1, 30:8: 1, 30:9: 1, 31:0: 1, 31:1: 1, 31:2: 1, 31:3: 1, 31:4: 1, 31:5: 1, 31:6: 1, 31:7: 1, 31:8: 1, 31:9: 1, 32:0: 1, 32:1: 1, 32:2: 1, 32:3: 1, 32:4: 1, 32:5: 1, 32:6: 1, 32:7: 1, 32:8: 1, 32:9: 1, 33:0: 1, 33:1: 1, 33:2: 1, 33:3: 1, 33:4: 1, 33:5: 1, 33:6: 1, 33:7: 1, 33:8: 1, 33:9: 1, 34:0: 1, 34:1: 1, 34:2: 1, 34:3: 1, 34:4: 1, 34:5: 1, 34:6: 1, 34:7: 1, 34:8: 1, 34:9: 1, 35:0: 1, 35:1: 1, 35:2: 1, 35:3: 1, 35:4: 1, 35:5: 1, 35:6: 1, 35:7: 1, 35:8: 1, 35:9: 1, 36:0: 1, 36:1: 1, 36:2: 1, 36:3: 1, 36:4: 1, 36:5: 1, 36:6: 1, 36:7: 1, 36:8: 1, 36:9: 1, 37:0: 1, 37:1: 1, 37:2: 1, 37:3: 1, 37:4: 1, 37:5: 1, 37:6: 1, 37:7: 1, 37:8: 1, 37:9: 1, 38:0: 1, 38:1: 1, 38:2: 1, 38:3: 1, 38:4: 1, 38:5: 1, 38:6: 1, 38:7: 1, 38:8: 1, 38:9: 1, 39:0: 1, 39:1: 1, 39:2: 1, 39:3: 1, 39:4: 1, 39:5: 1, 39:6: 1, 39:7: 1, 39:8: 1, 39:9: 1, 40:0: 1, 40:1: 1, 40:2: 1, 40:3: 1, 40:4: 1, 40:5: 1, 40:6: 1, 40:7: 1, 40:8: 1, 40:9: 1, 41:0: 1, 41:1: 1, 41:2: 1, 41:3: 1, 41:4: 1, 41:5: 1, 41:6: 1, 41:7: 1, 41:8: 1, 41:9: 1, 42:0: 1, 42:1: 1, 42:2: 1, 42:3: 1, 42:4: 1, 42:5: 1, 42:6: 1, 42:7: 1, 42:8: 1, 42:9: 1, 43:0: 1, 43:1: 1, 43:2: 1, 43:3: 1, 43:4: 1, 43:5: 1, 43:6: 1, 43:7: 1, 43:8: 1, 43:9: 1, 44:0: 1, 44:1: 1, 44:2: 1, 44:3: 1, 44:4: 1, 44:5: 1, 44:6: 1, 44:7: 1, 44:8: 1, 44:9: 1, 45:0: 1, 45:1: 1, 45:2: 1, 45:3: 1, 45:4: 1, 45:5: 1, 45:6: 1, 45:7: 1, 45:8: 1, 45:9: 1, 46:0: 1, 46:1: 1, 46:2: 1, 46:3: 1, 46:4: 1, 46:5: 1, 46:6: 1, 46:7: 1, 46:8: 1, 46:9: 1, 47:0: 1, 47:1: 1, 47:2: 1, 47:3: 1, 47:4: 1, 47:5: 1, 47:6: 1, 47:7: 1, 47:8: 1, 47:9: 1, 48:0: 1, 48:1: 1, 48:2: 1, 48:3: 1, 48:4: 1, 48:5: 1, 48:6: 1, 48:7: 1, 48:8: 1, 48:9: 1, 49:0: 1, 49:1: 1, 49:2: 1, 49:3: 1, 49:4: 1, 49:5: 1, 49:6: 1, 49:7: 1, 49:8: 1, 49:9: 1, 50:0: 1, 50:1: 1, 50:2: 1, 50:3: 1, 50:4:

[illegible][illegible]

```

30: controls.remove('specific_gravity')
   controls.remove('albumin')
   controls.remove('sugar')
   print(controls)

{'serum_creatinine', 'blood_glucose_random', 'potassium', 'sodium', 'blood_pressure', 'blood_urea', 'hemoglobin', 'age'}

```

```
[51]: controls.add('red_blood_cell_count')
      controls.add('packed_cell_volume')
      controls.add('white_blood_cell_count')
      print(controls)

{'serum_creatinine', 'blood_glucose_random', 'red_blood_cell_count', 'potassium', 'packed_cell_volume', 'white_blood_cell_count', 'sodium', 'blood_pressure', 'blood_urea', 'hemoglobin', 'age'}
```

```
In [32]: catcols.add('specific_gravity')
catcols.add('albumin')
catcols.add('sugar')
print(catcols)

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

```
In [33]: data['coronary_artery_disease']=data.coronary_artery_disease.replace({'tno','no'})
c(data['coronary_artery_disease'])
```

```
Out[33]: Counter({'no': 364, 'yes': 34, nan: 2})
```

```
In [34]: data['diabetesmellitus']=data.diabetesmellitus.replace(to_replace={'\tno','no','\tyes','yes', ' yes','yes', ' no':'no'})
c(data['diabetesmellitus'])
```

```
Out[34]: Counter({'yes': 197, 'no': 261, nan: 2})
```

```
In [35]: data.isnull().any()
```

```
Out[35]: 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                0
dtype: int64
```

```
In [37]: 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 [38]: 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 [39]: 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 [40]: from sklearn.preprocessing import LabelEncoder
```

```
In [41]: for i in catcols:
print("LABEL ENCODING OF:",i)
LE=LabelEncoder()
print(c(data[i]))
data[i]=LE.fit_transform(data[i])
print(c(data[i]))
print("====100")
```

```
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: sugar
Counter({0: 359, 1: 18, 3: 14, 4: 13, 5: 3})
Counter({0: 359, 1: 18, 3: 14, 4: 13, 5: 3})
=====
LABEL ENCODING OF: anemia
Counter({'no': 340, 'yes': 60})
Counter({0: 340, 1: 60})
=====
LABEL ENCODING OF: appetite
Counter({'good': 318, 'poor': 82})
Counter({0: 318, 1: 82})
=====
LABEL ENCODING OF: pus_cell_clumps
Counter({'notpresent': 358, 'present': 42})
Counter({0: 358, 1: 42})
=====
LABEL ENCODING OF: class
Counter({'cid': 250, 'notcid': 150})
Counter({0: 250, 1: 150})
=====
LABEL ENCODING OF: bacteria
Counter({'notpresent': 378, 'present': 22})
Counter({0: 378, 1: 22})
=====
LABEL ENCODING OF: hypertension
Counter({'no': 253, 'yes': 147})
Counter({0: 253, 1: 147})
=====
LABEL ENCODING OF: pus_cell
Counter({'normal': 324, 'abnormal': 76})
Counter({1: 324, 0: 76})
=====
LABEL ENCODING OF: pedal_edema
Counter({'no': 324, 'yes': 76})
Counter({0: 324, 1: 76})
=====
LABEL ENCODING OF: red_blood_cells
Counter({'normal': 353, 'abnormal': 47})
Counter({1: 353, 0: 47})
=====
LABEL ENCODING OF: coronary_artery_disease
Counter({'no': 366, 'yes': 34})
Counter({0: 366, 1: 34})
=====
```

```

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

In [42]: selcols=['red_blood_cells','pus_cell','blood_glucose_random','blood_urea','pedal_edema','anemia','diabetesmellitus','coronary_artery_disease']

In [43]: x=pd.DataFrame(data,columns=selcols)
y=pd.DataFrame(data,columns=['class'])
print(x.shape)
print(y.shape)

(400, 8)
(400, 1)

In [44]: 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 [45]: from sklearn.ensemble import RandomForestClassifier
lgr = RandomForestClassifier()

In [46]: #from sklearn.linear_model import LogisticRegression

In [47]: #lgr=LogisticRegression(solver='lbfgs', max_iter=1000)
lgr.fit(x_train.values, y_train.values.ravel())

Out[47]: RandomForestClassifier()

In [47]: #lgr=LogisticRegression(solver='lbfgs', max_iter=1000)
lgr.fit(x_train.values, y_train.values.ravel())

Out[47]: RandomForestClassifier()

In [48]: y_pred=lgr.predict(x_test.values)

In [49]: y_pred=lgr.predict([[90,157,0,1,1,0,1,1]])
print(y_pred)
c(y_pred)

[0]
Out[49]: Counter({0: 1})

In [50]: accuracy_score(y_test,y_pred)

Out[50]: 0.95

In [51]: conf_mat=confusion_matrix(y_test,y_pred)
conf_mat

Out[51]: array([[52,  2],
       [ 2, 24]], dtype=int64)

In [52]: pickle.dump(lgr,open('CKD.pkl','wb'))

```