

● **IMPORT THE LIBRARIES**

```
In [ ]: from google.colab import files
files.upload()
```

[illegible]

0.0,70.0,1.015,1.0,0.0,normal,normal,notpresent,notpresent,92.0,29.0,1.8,133.0,3.9,10.3,32,,yes,no,no,good,no,no,cckd\m37,
72.0,80.0,,,notpresent,notpresent,137.0,65.0,3.4,141.0,4.7,9.7,28,6900,2.5,yes,yes,no,poor,no,yes,cckd\t\n38,69.0,80.0,
1.02,3.0,0.0,abnormal,normal,notpresent,notpresent,,103.0,4.4,132.0,5.9,12.5,,,yes,no,no,good,no,no,cckd\m39,82.0,80.0,1.0
1.2,0.2,0.0,normal,notpresent,notpresent,140.0,70.0,3.4,136.0,4.2,13.0,40,9800,4.2,yes,yes,no,good,no,no,cckd\m40,46.0,90.0,
1.01,2.0,0.0,normal,abnormal,notpresent,notpresent,99.0,80.0,2.1,,,11.1,32,9100,4.1,yes,no,\tno,good,no,no,cckd\m41,45.0,7.
0.0,1.01,0.0,0.0,normal,notpresent,notpresent,,20.0,0.7,,,,,no,no,no,good,yes,no,cckd\m42,47.0,100.0,1.01,0.0,0.0,norma
l,notpresent,notpresent,204.0,29.0,1.0,139.0,4.2,9.7,33,9200,4.5,yes,no,no,good,yes,yes,cckd\m43,35.0,80.0,1.01,1.0,1.0,0.0,abno
rmal,,notpresent,notpresent,79.0,202.0,10.8,134.0,3.4,7.9,24,7900,3.1,no,yes,no,good,no,no,cckd\m44,54.0,80.0,1.01,3.0,0.0,0.
abnormal,abnormal,notpresent,notpresent,207.0,77.0,6.3,134.0,4.8,9.7,28,,,yes,yes,no,poor,yes,no,cckd\m45,54.0,80.0,1.02,3.
0,0.0,,abnormal,notpresent,notpresent,208.0,89.0,5.9,130.0,4.9,9.3,,,,yes,yes,no,poor,yes,no,cckd\m46,48.0,70.0,1.015,0.0,
0.0,normal,notpresent,notpresent,124.0,24.0,1.2,142.0,4.2,12.4,37,6400,4.7,no,yes,no,good,no,no,cckd\m47,11.0,80.0,1.01,3.
0,0.0,normal,notpresent,notpresent,,17.0,0.8,,,15.0,45,8600,,no,no,no,good,no,no,cckd\m48,73.0,70.0,1.005,0.0,0.0,normal,n
ormal,notpresent,notpresent,70.0,32.0,0.9,125.0,4.0,10.0,29,18900,3.5,yes,yes,no,good,yes,no,cckd\m49,60.0,70.0,1.01,2.0,0.
0,normal,abnormal,present,notpresent,144.0,72.0,3.0,,,9.7,29,21600,3.5,yes,yes,no,poor,yes,cckd\m50,53.0,60.0,,,notpr
esent,notpresent,91.0,114.0,3.25,142.0,4.3,8.6,28,11000,3.8,yes,yes,no,poor,yes,yes,cckd\m51,54.0,100.0,1.015,3.0,0.0,0.
nmal,present,notpresent,162.0,66.0,1.6,136.0,4.4,10.3,33,,,yes,yes,no,poor,yes,no,cckd\m52,53.0,90.0,1.015,0.0,0.0,norma
l,present,notpresent,,38.0,2.2,,,10.9,34,4300,3.7,no,no,no,poor,no,yes,cckd\m53,62.0,80.0,1.015,0.0,5.0,,,notpresent,notpres
ent,246.0,24.0,1.0,,,13.6,40,8500,4.7,yes,yes,no,good,no,no,cckd\m54,63.0,80.0,1.01,2.0,2.0,normal,,notpresent,notpresen
t,,,3.4,136.0,4.2,13.0,40,9800,4.2,yes,yes,yes,good,no,no,cckd\m55,35.0,80.0,1.005,3.0,0.0,abnormal,normal,notpresent,notpre
sent,,,9.5,28,,,no,no,no,good,yes,no,cckd\m56,76.0,70.0,1.015,3.0,4.0,normal,abnormal,present,notpresent,,164.0,9.7,131.
0,4.4,10.2,30,11300,3.4,yes,yes,yes,poor,yes,no,cckd\m57,76.0,90.0,,,normal,notpresent,notpresent,93.0,155.0,7.3,132.0,4.
9,,,yes,yes,yes,poor,no,no,cckd\m58,73.0,80.0,1.02,2.0,0.0,abnormal,abnormal,notpresent,notpresent,253.0,142.0,4.6,138.0,
5.8,10.5,33,7200,4.3,yes,yes,yes,good,no,no,cckd\m59,59.0,100.0,,,,,notpresent,notpresent,,96.0,6.4,,,6.6,,,yes,yes,no,go
od,no,yes,cckd\m60,67.0,90.0,1.02,1.0,0.0,,abnormal,present,notpresent,141.0,66.0,3.2,138.0,6.6,,,yes,no,no,good,no,no,ck
d\m61,67.0,80.0,1.01,1.0,3.0,normal,abnormal,notpresent,notpresent,182.0,391.0,32.0,163.0,39.0,,,yes,no,no,good,yes,no,ck
d\m62,15.0,60.0,1.02,3.0,0.0,normal,notpresent,notpresent,86.0,15.0,0.6,138.0,4.0,11.0,33,7700,3.8,yes,yes,no,good,no,ye
cckd\m63,46.0,70.0,1.015,1.0,0.0,abnormal,normal,notpresent,notpresent,150.0,111.0,6.1,131.0,3.7,7.5,27,,,no,no,no,good,
yes,cckd\m64,55.0,80.0,1.01,0.0,0.0,normal,notpresent,notpresent,146.0,,,,,9.8,,,no,no,\tno,good,no,no,cckd\m65,44.0,90.0,
1.01,1.0,0.0,normal,notpresent,notpresent,,20.0,1.1,,,15.0,48,,,no,\tno,nogood,no,yes,cckd\m66,67.0,70.0,1.02,2.0,0.0,abno
rmal,normal,notpresent,notpresent,150.0,55.0,1.6,131.0,4.8,\t?,,,yes,yes,no,good,yes,cckd\m67,45.0,80.0,1.02,3.0,0.0,0.

[illegible]

0,44,0,1.2,142,0,4,9,14,5,44,9400,6,4,no,no,no,good,no,no,notckd\n279,24,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,140,0,23,0,0,6,140,0,4,7,16,3,48,5800,5,6,no,no,no,good,no,no,notckd\n280,47,0,80,0,,,,,notpresent,notpresent,93,0,33,0,0,9,144,0,4,5,13,3,52,8100,5,2,no,n o,no,good,no,no,notckd\n281,55,0,80,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,130,0,50,0,1,2,147,0,5,0,15,5,41,9100,6,0,no,no,no,good, no,no,notckd\n282,20,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,123,0,44,0,1,0,135,0,3,8,14,6,44,5500,4,8,no,no,no,good,no,no,notckd\n283,60,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,,,,,16,4,43,10800,5,7,no,no,no,good,no,no,notckd\n284,33,0,80,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,100,0,37,0,1,2,142,0,4,0,16,9,52,6700,6,0,no,no,no,good,no,no,notckd\n285,66,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,94,0,19,0,0,7,135,0,3,9,16,0,41,5300,5,9,no,no,no,good,no,no,notckd\n286,71,0,70,0,1.02,0,0,0,0,norm al,norm al,n otpresent,notpresent,81,0,18,0,0,8,145,0,5,0,14,7,44,9800,6,0,no,no,no,good,no,no,notckd\n287,39,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent, notpresent,124,0,22,0,0,6,137,0,3,8,13,4,43,,no,no,no,good,no,no,notckd\n288,56,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,70,0,46,0,1,2,135,0,4,9,15,9,50,11000,5,1,,,,,good,no,no,notckd\n289,42,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,93,0,32,0,0,9,143,0,4,7,16,6,43,7100,5,3,no,no,no,good,no,no,notckd\n290,54,0,70,0,1.02,0,0,0,0,,,,,76,0,28,0,0,6,146,0,3,5,14,8,52,8400,5,9,no,no,no,good,no,no,notckd\n291,47,0,80,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,124,0,44,0,1,0,140,0,4,9,14,9,41,7000,5,7,no,no,no,good,no,no,notckd\n292,30,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,89,0,42,0,0,5,139,0,5,0,16,7,52,10200,5,0,no,no,no,good,no,no,notckd\n293,50,0,,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,92,0,19,0,1,2,150,0,4,8,14,9,48,4700,5,4,no,no,no,good,no,no,notckd\n294,75,0,60,0,1.02,0,0,0,0,norm al,norm al,norm al,notpresent,notpresent,110,0,50,0,0,7,135,0,5,0,14,3,40,8300,5,8,no,no,no,,,,,notckd\n295,44,0,70,0,,,,,,notpresent,notpresent,106,0,25,0,0,9,150,0,3,6,15,0,50,9600,6,5,no,no,no,good,no,no,notckd\n296,41,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,125,0,38,0,0,6,140,0,5,0,16,8,41,6300,5,9,no,no,no,good,no,no,notckd\n297,53,0,60,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,116,0,26,0,1,0,146,0,4,9,15,8,45,7700,5,2,,,,,good,no,no,notckd\n298,34,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,91,0,49,0,1,2,135,0,4,5,13,5,48,8600,4,9,no,no,no,good,no,no,notckd\n299,73,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,127,0,48,0,0,5,150,0,3,5,15,1,52,11000,4,7,no,n o,no,good,no,no,notckd\n300,45,0,60,0,1.02,0,0,0,0,norm al,,,,,114,0,26,0,7,141,0,4,2,15,0,43,9200,5,8,no,no,no,good,no,no,notckd\n301,4,0,60,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,96,0,33,0,0,9,147,0,4,5,16,9,41,7200,5,0,no,no,no,good,no,no,notckd\n302,29,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,127,0,44,0,1,2,145,0,5,0,14,8,48,,no,no,no,good,no,no,notckd\n303,55,0,70,0,1.02,0,0,0,norm al,norm al,notpresent,notpresent,107,0,26,0,1,1,,17,0,50,6700,6,1,no,no,no,good,no,no,notckd\n304,33,0,80,0,1.025,0,0,0,0,norm al,norm al,notpres ent,notpresent,128,0,38,0,0,6,135,0,3,9,13,1,45,6200,4,5,no,no,no,good,no,no,notckd\n305,41,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpre sent,122,0,25,0,0,16,8,41,6300,5,2,no,no,no,good,no,no,notckd\n306,52,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,128,0,30,0,1,2,140,0,4,5,15,2,52,4300,5,7,no,no,no,good,no,no,notckd\n307,47,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,137,0,17,0,0,5,150,0,3,5,13,6,44,7900,4,5,no,no,no,good,no,no,notckd\n308,43,0,80,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,81,0,46,0,0,6,135,0,4,9,13,9,48,6900,4,9,no,no,no,good,no,no,notckd\n309,51,0,60,0,1.02,0,0,0,0,,,,,notpresent,notpresent,129,0,25,0,1,2,139,0,5,0,17,2,40,8100,5,9,no, no,good,no,no,notckd\n310,46,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,102,0,27,0,0,7,142,0,4,9,13,2,44,11000,5,4,no,no,no,good, no,no,notckd\n311,56,0,60,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,132,0,18,0,1,1,147,0,4,7,13,7,45,7500,5,6,no,no,no,good,no,no,notckd\n312,80,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,,,,,135,0,4,1,15,3,48,6300,6,1,no,no,no,good,no,no,notckd\n313,55,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,104,0,28,0,0,9,142,0,4,8,17,3,52,8200,4,8,no,no,no,good,no,no,notckd\n314,39,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,131,0,46,0,0,6,145,0,5,0,15,6,41,9400,4,7,no,no,no,good,no,no,notckd\n315,44,0,70,0,1.025,0,0,0,0,norm al,1,norm al,notpresent,notpresent,,,,,13,8,48,7800,4,4,no,no,no,good,no,no,notckd\n316,35,0,,1.02,0,0,0,0,norm al,norm al,,,,,99,0,30,0,0,5,135,0,4,

9,15,4,48,5000,5,2,no,no,no,good,no,no,notckd\n317,58,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,102,0,48,0,1,2,139,0,4,3,15,0,40,8100,4,9,no,no,no,good,no,no,notckd\n318,61,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,120,0,29,0,0,7,137,0,3,5,17,4,52,7000,5,3,no, no,no,good,no,no,notckd\n319,30,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,138,0,15,0,1,1,135,0,4,4,,,,,no,no,good,no,no,notckd\n320,57,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,105,0,49,0,1,2,150,0,4,7,15,7,44,10400,6,2,no,no,no,good,no,no,notckd\n321,65,0,60,0,1.02,0,0,0,0,norm al,norm al,norm al,notpresent,notpresent,109,0,39,0,1,0,144,0,3,5,13,9,48,9600,4,8,no,no,no,good,no,no,notckd\n322,70,0,60,0,,,,,,notpresent,notpresent,120,0,40,0,0,5,140,0,4,6,16,0,43,4500,4,9,no,no,no,good,no,no,notckd\n323,43,0,80,0,1.025,0,0,0,0,norm al,norm al,n otpresent,notpresent,130,0,30,0,1,1,143,0,5,0,15,9,45,7800,4,5,no,no,no,good,no,no,notckd\n324,40,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,n tpresent,119,0,15,0,0,7,150,0,4,9,,,,,no,no,no,good,no,no,notckd\n325,58,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,100,0,50,0,1,2,140,0,3,5,14,0,50,6700,6,5,no,no,no,good,no,no,notckd\n326,47,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,109,0,25,0,1,1,141,0,4,7,15,8,41,8300,5,2,no,no,no,good,no,no,notckd\n327,30,0,60,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,120,0,31,0,0,8,150,0,4,6,13,4,4,10700,5,8,no,no,no,good,no,no,notckd\n328,28,0,70,0,1.02,0,0,0,0,norm al,norm al,,,,,131,0,29,0,0,6,145,0,4,9,,45,8600,6,5,no,no,no,good,no,no,notckd\n329,33,0,60,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,80,0,25,0,0,9,146,0,3,5,14,1,48,7800,5,1,no,no,no,good,no,no,notckd\n330,43,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,114,0,32,0,1,1,135,0,3,9,,42,,no,no,no,good,no,no,notckd\n331,59,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,130,0,39,0,0,7,147,0,4,7,13,5,46,6700,4,5,no,no,no,good,no,no,notckd\n332,34,0,70,0,1.025,0,0,0,0,norma l,norm al,notpresent,notpresent,,33,0,1,0,150,0,5,0,15,3,44,10500,6,1,no,no,no,good,no,no,notckd\n333,23,0,80,0,1.02,0,0,0,0,norm al,norm al,notpr esent,notpresent,99,0,46,0,1,2,142,0,4,0,17,7,46,4300,5,5,no,no,no,good,no,no,notckd\n334,24,0,80,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpre sent,125,0,,136,0,3,5,15,4,43,5600,4,5,no,no,no,good,no,no,notckd\n335,60,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,134,0,45,0,0,5,139,0,4,8,14,2,48,10700,5,6,no,no,no,good,no,no,notckd\n336,25,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,119,0,0,27,0,0,5,,,15,2,40,9200,5,2,no,no,no,good,no,no,notckd\n337,44,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,92,0,40,0,0,9,141,0,4,9,14,0,52,750 0,6,2,no,no,no,good,no,no,notckd\n338,62,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,132,0,34,0,0,8,147,0,3,5,17,8,44,4700,4,5,no,n o,no,good,no,no,notckd\n339,25,0,70,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,88,0,42,0,0,5,136,0,3,5,13,3,48,7000,4,9,no,no,no,good,n o,no,notckd\n340,32,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,100,0,29,0,1,1,142,0,4,5,14,3,43,6700,5,9,no,no,no,good,no,no,notc kd\n341,63,0,70,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,130,0,37,0,0,9,150,0,5,0,13,4,41,7300,4,7,no,no,no,good,no,no,notckd\n342,4 4,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,95,0,46,0,0,5,138,0,4,2,15,0,50,7700,6,3,no,no,no,good,no,no,notckd\n343,37,0,60,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,111,0,35,0,0,8,135,0,4,1,16,2,50,5500,5,7,no,no,no,good,no,no,notckd\n344,04,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,106,0,27,0,7,150,0,3,3,14,4,42,8100,4,7,no,no,no,good,no,no,notckd\n345,22,0,60,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,97,0,18,0,1,2,138,0,4,3,13,5,42,7900,6,4,no,no,no,good,no,no,notckd\n346,33,0,60,0,,,,,norm al,norm al,notpresent,notpre sent,130,0,41,0,0,9,141,0,4,4,15,5,52,4300,5,8,no,no,no,good,no,no,notckd\n347,43,0,60,0,1.025,0,0,0,0,norm al,norm al,notpresent,notpresent,108,0,25,0,0,1,0,144,0,5,0,17,8,43,7200,5,5,no,no,no,good,no,no,notckd\n348,38,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,99,0,19,0,0,5,147,0,3,5,13,6,44,7300,6,4,no,no,no,good,no,no,notckd\n349,35,0,70,0,1.025,0,0,0,0,,,,,notpresent,notpresent,82,0,36,0,1,1,150,0,3,5,14,5,52,9400,6,1,no,no,no,good,no,no,notckd\n350,65,0,70,0,1.025,0,0,0,0,,,,,notpresent,notpresent,85,0,20,0,1,0,142,0,4,8,16,1,43,9600,4,5,no,no,no,good, no,no,notckd\n351,29,0,80,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,83,0,49,0,0,9,139,0,3,3,17,5,40,9900,4,7,no,no,no,good,no,no,notc kd\n352,37,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,109,0,47,0,1,1,141,0,4,9,15,0,48,7000,5,2,no,no,no,good,no,no,notckd\n353,3 9,0,60,0,1.02,0,0,0,0,norm al,norm al,notpresent,notpresent,86,0,37,0,0,6,150,0,5,0,13,6,51,5800,4,5,no,no,no,good,no,no,notckd\n354,32,0,60,0,1,

● READ THE DATASET

```
In [ ]: df = pd.read_csv("kidney_disease2.csv")

In [ ]: df.head()

Out[ ]:
  id  age  bp    sg  al  su  rbc    pc    pcc    ba  bgr  bu  sc  sod  pot  hemo  pcv  wc  rc  htn  dm  cad  appet  pe  ane  classifi
0   0   40  80.0  1.020  1.0  0.0   NaN  normal  notpresent  notpresent  121.0  36.0  1.2   NaN   NaN   15.4  44  7800  5.2  yes  yes  no  good  no  no
1   1   7.0  50.0  1.020  4.0  0.0   NaN  normal  notpresent  notpresent   NaN  18.0  0.8   NaN   NaN   11.3  38  6000  NaN  no  no  no  good  no  no
2   2  62.0  80.0  1.010  2.0  3.0  normal  normal  notpresent  notpresent  423.0  53.0  1.8   NaN   NaN   9.6  31  7500  NaN  no  yes  no  poor  yes  yes
3   3   48.0  70.0  1.005  4.0  0.0  normal  abnormal  present  notpresent  117.0  56.0  3.8  111.0  2.5  11.2  32  6700  3.9  yes  no  no  poor  yes  yes
4   4  51.0  80.0  1.010  2.0  0.0  normal  normal  notpresent  notpresent  106.0  26.0  1.4   NaN   NaN   11.6  35  7300  4.6  no  no  no  good  no  no

#making columns meaningful
df.columns = ['id','Age','Blood_Pressure','Specific_Gravity','Albumin','Sugar','Red_Blood_Cells','Pus_Cells','Puss_Cell_Clumps','Bacteria',
              'Blood_Glucose_Random','Blood_Urea','Serum_Creatinine','Sodium','Potassium','Haemoglobin','Packed_Cell_Volume',
              'White_Blood_Cell_Count','Red_Blood_Cell_Count','Hypertension','Diabetes_Mellitus','Coronary_Artery_Disease',
              'Appetite','Peda_Edema','Aanemia', 'clas_s']

In [ ]: df.head()
```

```
Out[ ]:
```

	id	Age	Blood_Pressure	Specific_Gravity	Albumin	Sugar	Red_Blood_Cells	Pus_Cells	Puss_Cell_Clumps	Bacteria	Blood_Gulcose_Random	Blood_Urea	Serum_Creati
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	36.0	
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	18.0	
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	53.0	
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	56.0	
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	26.0	

Understanding Data Type And Summary Of Features

```
In [ ]:
```

```
df.info()

RangeIndex: 400 entries, 0 to 399
Data columns (total 26 columns):
#   Column              Non-Null Count  Dtype
---  -
0   id                   400 non-null   int64
1   Age                  391 non-null   float64
2   Blood_Pressure       388 non-null   float64
3   Specific_Gravity     353 non-null   float64
4   Albumin              354 non-null   float64
5   Sugar                351 non-null   float64
6   Red_Blood_Cells      248 non-null   object
7   Pus_Cells            335 non-null   object
8   Puss_Cell_Clumps     396 non-null   object
9   Bacteria              396 non-null   object
10  Blood_Gulcose_Random  356 non-null   float64
11  Blood_Urea            381 non-null   float64
12  Serum_Creatinine     383 non-null   float64
13  Sodium                313 non-null   float64
14  Potassium             312 non-null   float64
15  Haemoglobin           348 non-null   float64
16  Packed_Cell_Volume    330 non-null   object
17  White_Blood_Cell_Count 295 non-null   object
18  Red_Blood_Cell_Count  270 non-null   object
19  Hypertension           398 non-null   object
20  Diabetes_Mellitus     398 non-null   object
21  Coronary_Artery_Disease 398 non-null   object
22  Appetite               399 non-null   object
23  Peda_Edema            399 non-null   object
24  Aanemia                399 non-null   object
25  clas_s                 400 non-null   object
dtypes: float64(11), int64(1), object(14)
memory usage: 81.4+ KB
```

```
In [ ]:
```

```
df.describe()
```

```
Out[ ]:
```

	id	Age	Blood_Pressure	Specific_Gravity	Albumin	Sugar	Blood_Gulcose_Random	Blood_Urea	Serum_Creatinine
count	400.000000	391.000000	388.000000	353.000000	354.000000	351.000000	356.000000	381.000000	383.000000
mean	199.500000	51.483376	76.469072	1.017408	1.016949	0.450142	148.036517	57.425722	3.072454
std	115.614301	17.169714	13.683637	0.005717	1.352679	1.099191	79.281714	50.503006	5.741126
min	0.000000	2.000000	50.000000	1.005000	0.000000	0.000000	22.000000	1.500000	0.400000
25%	99.750000	42.000000	70.000000	1.010000	0.000000	0.000000	99.000000	27.000000	0.900000
50%	199.500000	55.000000	80.000000	1.020000	0.000000	0.000000	121.000000	42.000000	1.300000
75%	299.250000	64.500000	80.000000	1.020000	2.000000	0.000000	163.000000	66.000000	2.800000
max	399.000000	90.000000	180.000000	1.025000	5.000000	5.000000	490.000000	391.000000	76.000000

SOME VISUALISATIONS OVER DATA

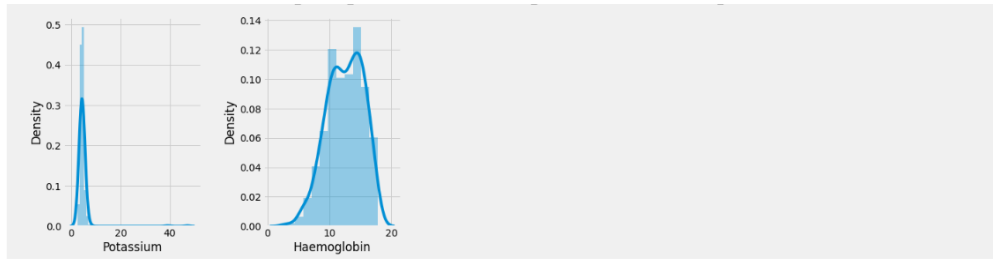
```
In [ ]:
```

```
number_columns = [col for col in df.columns if df[col].dtype != 'object']
```

```
In [ ]:
```

```
plt.figure(figsize = (20, 15))
plotnumber = 1

for column in number_columns:
    if plotnumber <= 14:
        ax = plt.subplot(3, 5, plotnumber)
        sns.distplot(df[column])
        plt.xlabel(column)
        plotnumber += 1
plt.tight_layout()
plt.show()
```



```
In [ ]: # Dropping the index ID from the dataset/Becasue we already have default indexing
df.drop('id', axis=1, inplace=True)
```

```
In [ ]: df
```

```
Out[ ]:
```

	Age	Blood_Pressure	Specific_Gravity	Albumin	Sugar	Red_Blood_Cells	Pus_Cells	Puss_Cell_Clumps	Bacteria	Blood_Gulcose_Random	Blood_Urea	Serum_Creatini
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	36.0	
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	18.0	
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	53.0	
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	56.0	
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	26.0	

395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	140.0	49.0	
396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	75.0	31.0	
397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	100.0	26.0	
398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	114.0	50.0	
399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	131.0	18.0	

400 rows x 25 columns



```
In [ ]: df.dtypes
```

```
Out[ ]:
```

Age	float64
Blood_Pressure	float64
Specific_Gravity	float64
Albumin	float64
Sugar	float64
Red_Blood_Cells	object
Pus_Cells	object
Puss_Cell_Clumps	object
Bacteria	object
Blood_Gulcose_Random	float64
Blood_Urea	float64
Serum_Creatinine	float64
Sodium	float64
Potassium	float64
Haemoglobin	float64
Packed_Cell_Volume	object
White_Blood_Cell_Count	object
Red_Blood_Cell_Count	object
Hypertension	object
Diabetes_Mellitus	object
Coronary_Artery_Disease	object
Appetite	object
Peda_Edema	object
Aanemia	object
clas_s	object
dtype:	object

```
In [ ]: #Converting the categorical to 0 and 1
```

```
In [ ]: df['Red_Blood_Cells'].value_counts()
```

```
Out[ ]:
```

normal	281
abnormal	47
Name: Red_Blood_Cells, dtype: int64	

```
In [ ]: df['Red_Blood_Cells'] = df['Red_Blood_Cells'].replace(to_replace = {'normal' : 0, 'abnormal' : 1})
```

```
In [ ]: df['Pus_Cells'].value_counts()
```

```
Out[ ]:
```

normal	259
abnormal	76
Name: Pus_Cells, dtype: int64	

```
In [ ]: df['Pus_Cells'] = df['Pus_Cells'].replace(to_replace = {'normal' : 0, 'abnormal' : 1})
```

```
In [ ]: df['Puss_Cell_Clumps'].value_counts()
```

```
Out[ ]: notpresent    354
        present       42
        Name: Puss_Cell_Clumps, dtype: int64
```

```
In [ ]: df['Puss_Cell_Clumps'] = df['Puss_Cell_Clumps'].replace(to_replace = {'notpresent':0,'present':1})
```

```
In [ ]: df['Hypertension'].value_counts()
```

```
Out[ ]: no      251
        yes     147
        Name: Hypertension, dtype: int64
```

```
In [ ]: df['Hypertension'] = df['Hypertension'].replace(to_replace = {'yes' : 1, 'no' : 0})
```

```
In [ ]: df['Diabetes_Mellitus'].value_counts()
```

```
Out[ ]: no      258
        yes     134
        \tno      3
        \tyes      2
        yes        1
        Name: Diabetes_Mellitus, dtype: int64
```

```
In [ ]: df['Coronary_Artery_Disease'].value_counts()
```

```
Out[ ]: no      362
        yes      34
        \tno      2
        Name: Coronary_Artery_Disease, dtype: int64
```

```
In [ ]: df['Coronary_Artery_Disease'] = df['Coronary_Artery_Disease'].replace(to_replace = {'\tno':'no'})
```

```
In [ ]: df['Appetite'].unique()
```

```
Out[ ]: array(['good', 'poor', nan], dtype=object)
```

```
In [ ]: df['Appetite'] = df['Appetite'].replace(to_replace={'good':1,'poor':0,'no':np.nan})
```

```
In [ ]: df['Peda_Edema'].value_counts()
```

```
Out[ ]: no      323
        yes      76
        Name: Peda_Edema, dtype: int64
```

```
In [ ]: df['Peda_Edema'] = df['Peda_Edema'].replace(to_replace = {'yes' : 1, 'no' : 0})
```

```
In [ ]: df['Aanemia'].value_counts()
```

```
Out[ ]: no      339
        yes      60
        Name: Aanemia, dtype: int64
```

```
In [ ]: df['Aanemia'] = df['Aanemia'].replace(to_replace = {'yes' : 1, 'no' : 0})
```

```
In [ ]: #TARGET COLUMN
        df['clas_s'].value_counts()
```

```
Out[ ]: ckd      248
        notckd   150
        ckd\t      2
        Name: clas_s, dtype: int64
```

```
In [ ]: df['clas_s'] = df['clas_s'].replace(to_replace={'ckd\t':'ckd'})
```

```
In [ ]: df["clas_s"] = [1 if i == "ckd" else 0 for i in df["clas_s"]]
```

```
In [ ]: # Gonna remove the bacteria column , nothing is present in the dataset
        df.drop ('Bacteria', axis=1, inplace = True)
```

```
In [ ]: df.info() #Converting PCV,WBCC,RBCC ( Objects to numeric)
```

```

RangeIndex: 400 entries, 0 to 399
Data columns (total 24 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   float64
 6   Pus_Cells             335 non-null   float64
 7   Puss_Cell_Clumps      396 non-null   float64
 8   Blood_Gulcose_Random  356 non-null   float64
 9   Blood_Urea            381 non-null   float64
10   Serum_Creatinine      383 non-null   float64
11   Sodium                313 non-null   float64
12   Potassium             312 non-null   float64
13   Haemoglobin           348 non-null   float64
14   Packed_Cell_Volume    330 non-null   object  
15   White_Blood_Cell_Count 295 non-null   object  
16   Red_Blood_Cell_Count  270 non-null   object  
17   Hypertension          398 non-null   float64
18   Diabetes_Mellitus     398 non-null   object  
19   Coronary_Artery_Disease 398 non-null   object  
20   Appetite              399 non-null   float64
21   Peda_Edema            399 non-null   float64
22   Aanemia               399 non-null   float64
23   clas_s                400 non-null   int64  
dtypes: float64(18), int64(1), object(5)
memory usage: 75.1+ KB

```

● Handling The Missing Values

```
In [ ]: df.isnull().sum().sort_values(ascending=False)
```

```

Out[ ]: Red_Blood_Cells      152
Red_Blood_Cell_Count      131
White_Blood_Cell_Count    106
Potassium                  88
Sodium                     87
Packed_Cell_Volume        71
Pus_Cells                  65
Haemoglobin                52
Sugar                      49
Specific_Gravity           47
Albumin                    46
Blood_Gulcose_Random       44
Blood_Urea                 19
Serum_Creatinine           17
Blood_Pressure             12
Age                         9
Puss_Cell_Clumps           4
Hypertension               2
Diabetes_Mellitus          2
Coronary_Artery_Disease    2
Appetite                   1
Peda_Edema                 1
Aanemia                    1

Aanemia                    1
clas_s                     0
dtype: int64

```

```
In [ ]: cat_features = ['Red_Blood_Cells', 'Pus_Cells', 'Puss_Cell_Clumps',
                        'Hypertension', 'Diabetes_Mellitus', 'Coronary_Artery_Disease',
                        'Appetite', 'Peda_Edema', 'Aanemia']
```

```
In [ ]: for feature in cat_features:
        df[feature] = df[feature].fillna(df[feature].mode())
```

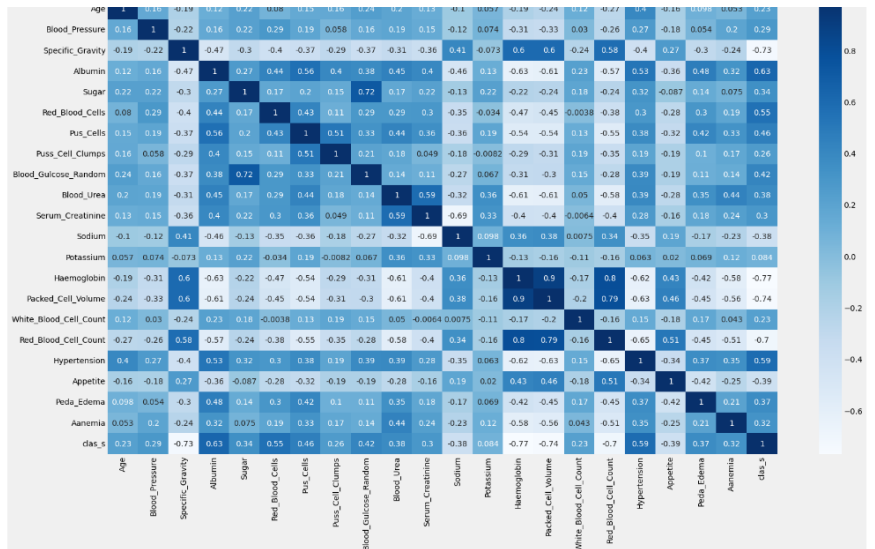
```
In [ ]: num_features = ['Age', 'Blood_Pressure', 'Specific_Gravity', 'Albumin', 'Sugar',
                        'Blood_Gulcose_Random', 'Blood_Urea', 'Serum_Creatinine', 'Sodium', 'Potassium', 'Haemoglobin',
                        'White_Blood_Cell_Count', 'Red_Blood_Cell_Count']
```

```
In [ ]: for feature in num_features:
        df[feature] = df[feature].fillna(df[feature].mode())
```

● Checking the total number of null values

```
In [ ]: df.columns.isnull().sum()
```

```
Out[ ]: 0
```



Splitting The Dataset Into Dependent And Independent Variable

```
In [ ]: # Independent and Dependent Feature:
X = df.iloc[:, :-1]
y = df.iloc[:, -1]

In [ ]: X.head()

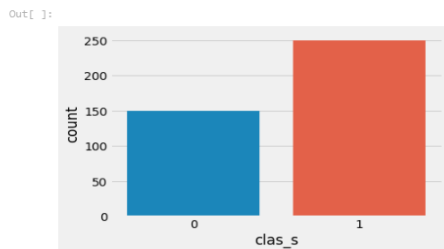
Out[ ]:
Age  Blood_Pressure  Specific_Gravity  Albumin  Sugar  Red_Blood_Cells  Pus_Cells  Pus_Cell_Clumps  Blood_Glucose_Random  Blood_Urea  Serum_Creatinine  Sodium  Potassium
0    40              3                3         1      0              0          0              0              48             23              8         19
1      5              0                3         4      0              2          0              0              146            5              4         34
2     54              3                1         2      3              0          0              0              140            40             14         34
3     40              2                0         4      0              0          1              1              44            43             32          2
4     43              3                1         2      0              0          0              0              33            13             10         34

In [ ]: y.head()

Out[ ]:
0    1
1    1
2    1
3    1
4    1
Name: clas_s, dtype: int64
```

THERE ARE NO FEATURES ABOVE 0.95 MULTI-COLLINEARITY SO NO DROPPING

```
In [ ]: # Target feature:
# Show the counts of observations using bars.
sns.countplot(df['clas_s'])
```



LABEL ENCODING

```
In [ ]: from sklearn.preprocessing import LabelEncoder

lab_enc = LabelEncoder()
for i in df:
    df[i] = lab_enc.fit_transform(df[i])
```


● Splitting The Dataset Into Dependent And Independent Variable

```
In [ ]: # Independent and Dependent Feature:
X = df.iloc[:, :-1]
y = df.iloc[:, -1]

In [ ]: X.head()

Out[ ]:
Age  Blood_Pressure  Specific_Gravity  Albumin  Sugar  Red_Blood_Cells  Pus_Cells  Puss_Cell_Clumps  Blood_Gulcose_Random  Blood_Urea  Serum_Creatinine  Sodium  Potassium
0      40             3                3        1      0              0         0              0              48         23              6        19         8
1       5             0                3        4      0              2         0              0             146          5              4        34        23
2      54             3                1        2      3              0         0              0             140         40              14        34        40
3      40             2                0        4      0              0         1              1              44         43              32         2         0
4      43             3                1        2      0              0         0              0              33         13              10        34        40

In [ ]: y.head()

Out[ ]:
0      1
1      1
2      1
3      1
4      1
Name: class_s, dtype: int64
```

● CHECKING FEATURE IMPORTANCE

```
In [ ]: # Feature Importance:
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X,y)

plt.figure(figsize=(8,6))
ranked_features=pd.Series(model.feature_importances_,index=X.columns)
ranked_features.nlargest(24).plot(kind='barh')
plt.show()
```

Feature	Importance
Anemia	0.00
Albumin	0.00
Sugar	0.00
Pus_Cells	0.00
Puss_Cell_Clumps	0.00
Blood_Gulcose_Random	0.00
Blood_Urea	0.00
Potassium	0.00
Sodium	0.00
White_Blood_Cell_Count	0.00
Diabetes_Mellitus	0.00
Coronary_Artery_Disease	0.00
Appetite	0.00
Packed_Cell_Volume	0.00
Age	0.00
Hypertension	0.00
Blood_Pressure	0.00
Red_Blood_Cells	0.00
Peda_Edema	0.08
Red_Blood_Cell_Count	0.12
Haemoglobin	0.15
Specific_Gravity	0.25
Serum_Creatinine	0.58

```
In [ ]: ranked_features.nlargest(8).index

Out[ ]: Index(['Serum_Creatinine', 'Specific_Gravity', 'Haemoglobin',
              'Red_Blood_Cell_Count', 'Peda_Edema', 'Red_Blood_Cells',
              'Blood_Pressure', 'Hypertension'],
              dtype='object')

In [ ]: X = df[['Haemoglobin', 'Specific_Gravity', 'Red_Blood_Cell_Count', 'Albumin', 'Blood_Urea', 'Blood_Pressure', 'Blood_Gulcose_Random', 'Serum_Creatinine']]
X.head()

Out[ ]:
Haemoglobin  Specific_Gravity  Red_Blood_Cell_Count  Albumin  Blood_Urea  Blood_Pressure  Blood_Gulcose_Random  Serum_Creatinine
0           90                3                    30         1           23              3              48              8
1           49                3                    45         4           5              0             146              4
2           32                1                    45         2          40              3             140             14
3           48                0                    17         4          43              2              44             32
4           52                1                    24         2          13              3              33             10
```

● SPLITTING THE DATASET INTO TRAINSET AND SPLIT TEST

```
In [ ]: # Train Test Split:
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.3, random_state=33)
print(X_train.shape)
print(X_test.shape)

(288, 8)
(120, 8)
```

● BUILDING THE MODEL

```
In [ ]: # Importing Performance Metrics:
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
In [ ]: # DecisionTreeClassifier:
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model = model.fit(X_train,y_train)
```

● TESTING THE MODEL

```
In [ ]: # Predictions:
y_pred = model.predict(X_test)
y_pred = model.predict([[10, 20, 30, 40, 50, 60, 80, 80]])
print(y_pred)

[1]
```

● MODEL EVALUATION

```
In [ ]: y_pred = model.predict(X_test)
# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print('Confusion Matrix:')
print(confusion_matrix(y_test,y_pred))
print('Classification Report:')
print(classification_report(y_test,y_pred))

Accuracy: 0.9833333333333333
Confusion Matrix:
[[57  1]
 [ 1 61]]
Classification Report:
      precision    recall  f1-score   support


```

```
print(classification_report(y_test,y_pred))
```

```
Accuracy: 0.9833333333333333
Confusion Matrix:
[[57  1]
 [ 1 61]]
Classification Report:
      precision    recall  f1-score   support

      0      0.98      0.98      0.98        58
      1      0.98      0.98      0.98        62

   accuracy      0.98
  macro avg      0.98
weighted avg      0.98
```

● SAVING THE MODEL

```
In [ ]: #Saving the model
pickle.dump(model,open('_CKD_NEW2.pkl','wb'))
df.to_csv('processed_kidney_disease2.csv', index=False)
```

● LISTING THE FILES

```
In [ ]: ls

_CKD_NEW2.pkl      'processed_kidney_disease 2.csv'  sample_data/
kidney_disease2.csv processed_kidney_disease2.csv
```

● DOWNLOADING THE FILES

```
In [ ]: files.download('_CKD_NEW2.pkl') # Downloading the pickle file --> Python object hierarchy is converted into a byte stream
```