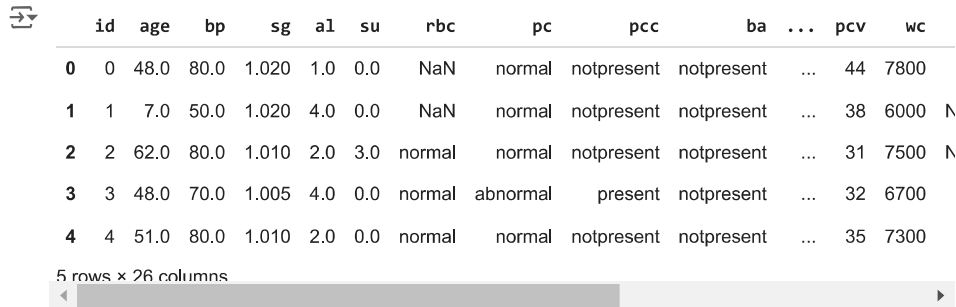


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

# For Filtering the warnings
import warnings
warnings.filterwarnings('ignore')
```

```
data = pd.read_csv('kidney_disease.csv')
```

```
data.head()
```

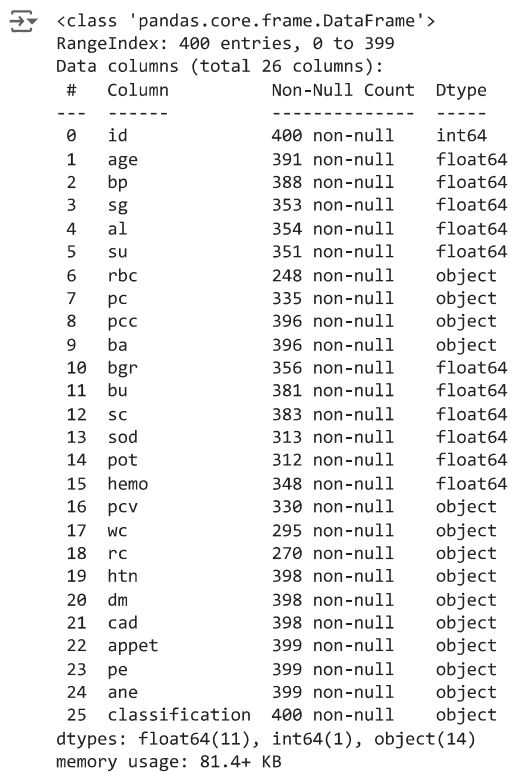


	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal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

5 rows x 26 columns

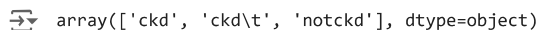
```
ckd=chronic kidney disease
```

```
data.info()
```



```
<class 'pandas.core.frame.DataFrame'>
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   bp                    388 non-null   float64
3   sg                    353 non-null   float64
4   al                    354 non-null   float64
5   su                    351 non-null   float64
6   rbc                   248 non-null   object
7   pc                    335 non-null   object
8   pcc                   396 non-null   object
9   ba                    396 non-null   object
10  bgr                   356 non-null   float64
11  bu                    381 non-null   float64
12  sc                    383 non-null   float64
13  sod                   313 non-null   float64
14  pot                   312 non-null   float64
15  hemo                  348 non-null   float64
16  pcv                   330 non-null   object
17  wc                    295 non-null   object
18  rc                    270 non-null   object
19  htn                   398 non-null   object
20  dm                    398 non-null   object
21  cad                   398 non-null   object
22  appet                 399 non-null   object
23  pe                    399 non-null   object
24  ane                   399 non-null   object
25  classification        400 non-null   object
dtypes: float64(11), int64(1), object(14)
memory usage: 81.4+ KB
```

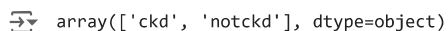
```
data.classification.unique()
```



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

```
data.classification=data.classification.replace("ckd\t","ckd")
```


```
data.classification.unique()
```



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

```
data.drop('id', axis = 1, inplace = True)
```

```
data.head()
```




	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wc
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	44	7800
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	38	6000
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	31	7500
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32	6700
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	35	7300

5 rows x 25 columns

```
data['classification'] = data['classification'].replace(['ckd','notckd'], [1,0])
```


```
data.head()
```



	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wc
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	44	7800
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	38	6000
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	31	7500
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32	6700
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	35	7300

5 rows x 25 columns

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



```


age          9
bp           12
sg           47
al           46
su           49
rbc          152
pc           65
pcc           4
ba            4
bgr          44
bu           19
sc           17
sod           87
pot           88
hemo         52
pcv          70
wc          105
rc           130
htn           2
dm            2
cad           2
appet         1
pe            1
ane           1
classification 0
dtype: int64

```

```

df = data.dropna(axis = 0)
print(f"Before dropping all NaN values: {data.shape}")
print(f"After dropping all NaN values: {df.shape}")

```



```

Before dropping all NaN values: (400, 25)
After dropping all NaN values: (158, 25)

```

```
df.head()
```



	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32
9	53.0	90.0	1.020	2.0	0.0	abnormal	abnormal	present	notpresent	70.0	...	29
11	63.0	70.0	1.010	3.0	0.0	abnormal	abnormal	present	notpresent	380.0	...	32
14	68.0	80.0	1.010	3.0	2.0	normal	abnormal	present	present	157.0	...	16
20	61.0	80.0	1.015	2.0	0.0	abnormal	abnormal	notpresent	notpresent	173.0	...	24

5 rows x 25 columns

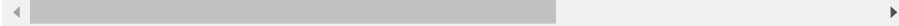


```
df.index = range(0, len(df), 1)
df.head()
```



	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv
0	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32
1	53.0	90.0	1.020	2.0	0.0	abnormal	abnormal	present	notpresent	70.0	...	29
2	63.0	70.0	1.010	3.0	0.0	abnormal	abnormal	present	notpresent	380.0	...	32
3	68.0	80.0	1.010	3.0	2.0	normal	abnormal	present	present	157.0	...	16
4	61.0	80.0	1.015	2.0	0.0	abnormal	abnormal	notpresent	notpresent	173.0	...	24

5 rows x 25 columns



```
for i in df['wc']:
    print(i)
```



```

>800
6600
7400
9500
6700
7800
6600
7200
6800

```

```
df['wc']=df['wc'].replace(["\t6200","\t8400"],[6200,8400])
```

```

for i in df['wc']:
    print(i)

```

```

↩ 4300
10700
7500
4700
7000
6700
7300
7700
5500
8100
7900
7200
7300
9900
7000
5800
6800
6300
7400
11000
5200
6000
5800
5400
10300
9300
10500
6700
9400
10300
9300
6500
10500
9200
8000
9700
9100
6400
5400
6500
6000
5100
11000
8000
5700
6200
9500
7200
6300
5800
6600
7400
9500
6700
7800
6600
7200
6800

```

```
df.info()
```

```

↩ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   age             158 non-null    float64

```

```
1  bp          158 non-null  float64
2  sg          158 non-null  float64
3  al          158 non-null  float64
4  su          158 non-null  float64
5  rbc         158 non-null  object
6  pc          158 non-null  object
7  pcc         158 non-null  object
8  ba          158 non-null  object
9  bgr         158 non-null  float64
10 bu          158 non-null  float64
11 sc          158 non-null  float64
12 sod         158 non-null  float64
13 pot         158 non-null  float64
14 hemo        158 non-null  float64
15 pcv         158 non-null  object
16 wc          158 non-null  int64
17 rc          158 non-null  object
18 htn         158 non-null  object
19 dm          158 non-null  object
20 cad         158 non-null  object
21 appet       158 non-null  object
22 pe          158 non-null  object
23 ane         158 non-null  object
24 classification 158 non-null  int64
dtypes: float64(11), int64(2), object(12)
memory usage: 31.0+ KB
```

```
df['pcv']=df['pcv'].astype(int)
df['wc']=df['wc'].astype(int)
df['rc']=df['rc'].astype(float)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 25 columns):
#   Column                Non-Null Count  Dtype
---  -
0   age                   158 non-null   float64
1   bp                   158 non-null   float64
2   sg                   158 non-null   float64
3   al                   158 non-null   float64
4   su                   158 non-null   float64
5   rbc                  158 non-null   object
6   pc                   158 non-null   object
7   pcc                  158 non-null   object
8   ba                   158 non-null   object
9   bgr                  158 non-null   float64
10  bu                   158 non-null   float64
11  sc                   158 non-null   float64
12  sod                  158 non-null   float64
13  pot                  158 non-null   float64
14  hemo                 158 non-null   float64
15  pcv                  158 non-null   int32
16  wc                   158 non-null   int32
17  rc                   158 non-null   float64
18  htn                  158 non-null   object
19  dm                   158 non-null   object
20  cad                  158 non-null   object
21  appet               158 non-null   object
22  pe                   158 non-null   object
23  ane                  158 non-null   object
24  classification      158 non-null   int64
dtypes: float64(12), int32(2), int64(1), object(10)
memory usage: 29.8+ KB
```

```
object_dtypes = df.select_dtypes(include = 'object')
object_dtypes.head()
```

	rbc	pc	pcc	ba	htn	dm	cad	appet	pe	ane
0	normal	abnormal	present	notpresent	yes	no	no	poor	yes	yes
1	abnormal	abnormal	present	notpresent	yes	yes	no	poor	no	yes
2	abnormal	abnormal	present	notpresent	yes	yes	no	poor	yes	no
3	normal	abnormal	present	present	yes	yes	yes	poor	yes	no
4	abnormal	abnormal	notpresent	notpresent	yes	yes	yes	poor	yes	yes

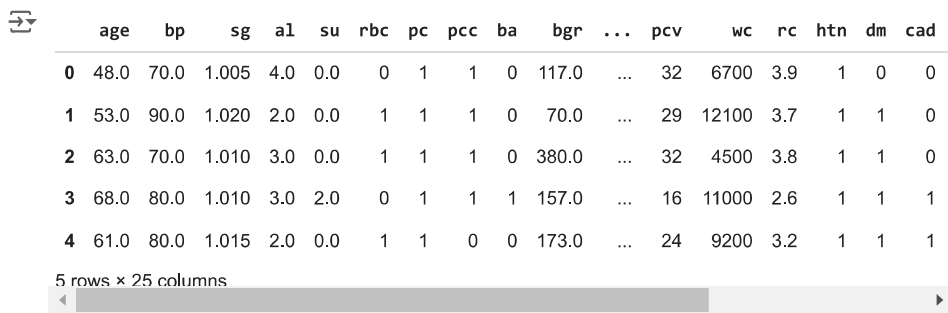
```

dictionary = {
    "rbc": {
        "abnormal":1,
        "normal": 0,
    },
    "pc":{
        "abnormal":1,
        "normal": 0,
    },
    "pcc":{
        "present":1,
        "notpresent":0,
    },
    "ba":{
        "notpresent":0,
        "present": 1,
    },
    "htn":{
        "yes":1,
        "no": 0,
    },
    "dm":{
        "yes":1,
        "no":0,
    },
    "cad":{
        "yes":1,
        "no": 0,
    },
    "appet":{
        "good":1,
        "poor": 0,
    },
    "pe":{
        "yes":1,
        "no":0,
    },
    "ane":{
        "yes":1,
        "no":0,
    }
}

```

```
df=df.replace(dictionary)
```

```
df.head()
```



	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wc	rc	htn	dm	cad
0	48.0	70.0	1.005	4.0	0.0	0	1	1	0	117.0	...	32	6700	3.9	1	0	0
1	53.0	90.0	1.020	2.0	0.0	1	1	1	0	70.0	...	29	12100	3.7	1	1	0
2	63.0	70.0	1.010	3.0	0.0	1	1	1	0	380.0	...	32	4500	3.8	1	1	0
3	68.0	80.0	1.010	3.0	2.0	0	1	1	1	157.0	...	16	11000	2.6	1	1	1
4	61.0	80.0	1.015	2.0	0.0	1	1	0	0	173.0	...	24	9200	3.2	1	1	1

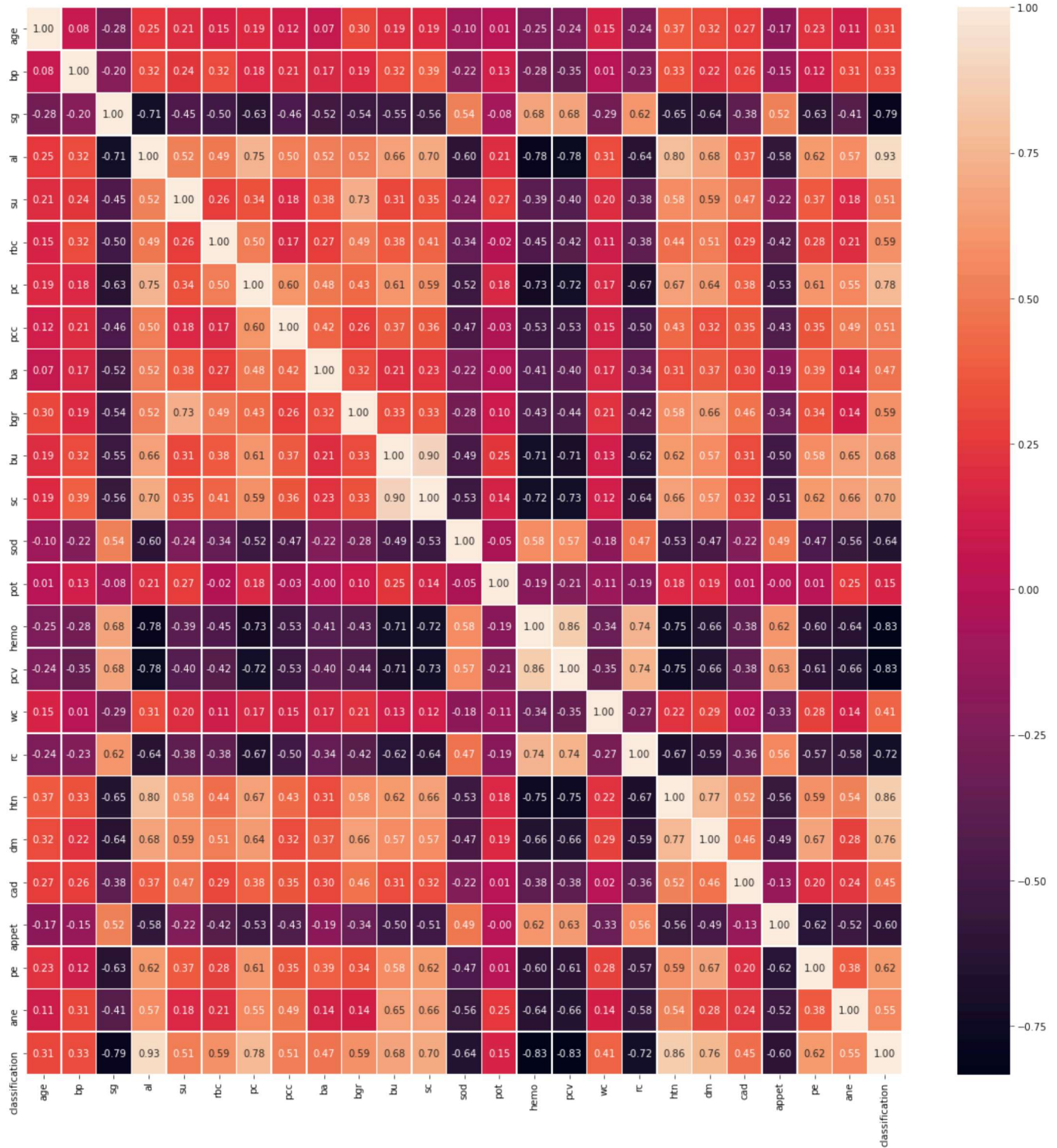
5 rows x 25 columns

```

import seaborn as sns
plt.figure(figsize = (20,20))
sns.heatmap(df.corr(), annot = True, fmt=".2f",linewidths=0.5)

```

<AxesSubplot:>



df.corr()



	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	1
age	1.000000	0.079712	-0.277303	0.253380	0.207711	0.147971	0.188907	0.124032	0.068353	0.301915	...	-0.235116	0.15311
bp	0.079712	1.000000	-0.198897	0.322507	0.243828	0.316670	0.179834	0.206507	0.174555	0.190113	...	-0.353504	0.0082
sg	-0.277303	-0.198897	1.000000	-0.712331	-0.448477	-0.500494	-0.630323	-0.460050	-0.516392	-0.544781	...	0.678472	-0.2889
al	0.253380	0.322507	-0.712331	1.000000	0.521448	0.489941	0.752956	0.503341	0.516104	0.518123	...	-0.775528	0.3145
su	0.207711	0.243828	-0.448477	0.521448	1.000000	0.256568	0.335901	0.177327	0.381929	0.730050	...	-0.404821	0.2010
rbc	0.147971	0.316670	-0.500494	0.489941	0.256568	1.000000	0.498959	0.168592	0.273177	0.493857	...	-0.422537	0.1088
pc	0.188907	0.179834	-0.630323	0.752956	0.335901	0.498959	1.000000	0.600092	0.481227	0.430646	...	-0.718042	0.1699
pcc	0.124032	0.206507	-0.460050	0.503341	0.177327	0.168592	0.600092	1.000000	0.415033	0.257768	...	-0.534564	0.1467
ba	0.068353	0.174555	-0.516392	0.516104	0.381929	0.273177	0.481227	0.415033	1.000000	0.318095	...	-0.397500	0.1700
bgr	0.301915	0.190113	-0.544781	0.518123	0.730050	0.493857	0.430646	0.257768	0.318095	1.000000	...	-0.443818	0.2120
bu	0.190636	0.316287	-0.545319	0.661940	0.312259	0.378478	0.613318	0.366726	0.205351	0.326496	...	-0.706582	0.1289
sc	0.189721	0.386551	-0.563122	0.702889	0.347196	0.410408	0.588517	0.361965	0.229238	0.331284	...	-0.726187	0.1239
sod	-0.102933	-0.224710	0.539285	-0.599334	-0.242491	-0.344916	-0.520324	-0.473954	-0.221374	-0.284968	...	0.570045	-0.1762
pot	0.006866	0.127801	-0.075057	0.209492	0.271954	-0.019319	0.176150	-0.030297	-0.000279	0.102226	...	-0.213488	-0.1075
hemo	-0.245645	-0.282365	0.682086	-0.784745	-0.385511	-0.452566	-0.733140	-0.531182	-0.410353	-0.434158	...	0.856775	-0.3374
pcv	-0.235116	-0.353504	0.678472	-0.775528	-0.404821	-0.422537	-0.718042	-0.534564	-0.397500	-0.443818	...	1.000000	-0.3496
wc	0.153132	0.008274	-0.288930	0.314574	0.201000	0.108857	0.169936	0.146742	0.170071	0.212093	...	-0.349607	1.0000
rc	-0.242235	-0.228533	0.619092	-0.640099	-0.377726	-0.379378	-0.667113	-0.499401	-0.343299	-0.418085	...	0.739019	-0.2723
htn	0.372348	0.334951	-0.648168	0.796876	0.577286	0.442400	0.666767	0.432876	0.314961	0.579407	...	-0.752043	0.2239
dm	0.323957	0.218096	-0.639391	0.678582	0.591010	0.511777	0.636288	0.321900	0.367477	0.663012	...	-0.655039	0.2870
cad	0.269868	0.257709	-0.379305	0.374755	0.466658	0.293269	0.384223	0.352255	0.297063	0.459164	...	-0.375627	0.0212
appet	-0.170259	-0.145047	0.523944	-0.578080	-0.220547	-0.418639	-0.528435	-0.432515	-0.187815	-0.338924	...	0.629102	-0.3287
pe	0.232327	0.117878	-0.633622	0.622268	0.374128	0.282868	0.606234	0.350171	0.393819	0.336141	...	-0.606829	0.2826
ane	0.105809	0.311097	-0.413252	0.569529	0.179811	0.209797	0.545380	0.485941	0.141344	0.139854	...	-0.655724	0.1392
classification	0.305119	0.326567	-0.790102	0.925816	0.510615	0.586391	0.775388	0.509915	0.468845	0.591217	...	-0.827983	0.4075

25 rows × 25 columns

```
X = df.drop(['classification', 'sg', 'appet', 'rc', 'pcv', 'hemo', 'sod'], axis = 1)
y = df['classification']
```

```
X.columns
```

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

```
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 = 42)
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
model = RandomForestClassifier(n_estimators = 20)
model.fit(X_train, y_train)
```

```
RandomForestClassifier(n_estimators=20)
```

```
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
confusion_matrix(y_test, model.predict(X_test))
```