

```

import pandas as pd #used for data manipulation

import numpy as np #used for numerical analysis

from collections import Counter as c # return counts of number of classes

import matplotlib.pyplot as plt #used for data visualization

import seaborn as sns #data visualization library

import missingno as msno #finding missing values

from sklearn.metrics import accuracy_score, confusion_matrix#model performance

from sklearn.model_selection import train_test_split #splics data in random train and test array

from sklearn.preprocessing import LabelEncoder #encoding the levels of categorical features

from sklearn.linear_model import LogisticRegression #classification ML algorithm

import pickle #python object hierarchy is converted into a byte stream,

data = pd.read_csv(r"/content/kidney_disease.csv") #loading the csv data

```

```
data.head() #return you the first 5 rows values
```

	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 × 26 columns

```
data.columns #return all the column names
```

```

Index(['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', 'classification'],
      dtype='object')

```

```
data.columns# rename column names to make it more user-friendly
```

```

Index(['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', 'classification'],
      dtype='object')

```

```
data.shape
```

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

```

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', 'haemoglobin', 'packed_cell_volume', 'white_blood_cell_count', 'red_blood_cell_count',
               'hypertension', 'diabetes_mellitus', 'coronary_artery_disease', 'appetite', 'peda_edema',

```

```

        'aanemia', 'class']
data.columns

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', 'haemoglobin', 'packed_cell_volume',
       'white_blood_cell_count', 'red_blood_cell_count', 'hypertension',
       'diabetes_mellitus', 'coronary_artery_disease', 'appetite',
       'peda_edema', 'aanemia', 'class'],
      dtype='object')

```

```
data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 25 columns):
#   Column                Non-Null Count  Dtype
---  -
0   age                    391 non-null   float64
1   blood_pressure         388 non-null   float64
2   specific_gravity       353 non-null   float64
3   albumin                354 non-null   float64
4   sugar                  351 non-null   float64
5   red_blood_cells        248 non-null   object
6   pus_cell               335 non-null   object
7   pus_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  haemoglobin            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  diabetes_mellitus      398 non-null   object
20  coronary_artery_disease 398 non-null   object
21  appetite               399 non-null   object
22  peda_edema             399 non-null   object
23  aanemia                399 non-null   object
24  class                  400 non-null   object
dtypes: float64(11), object(14)
memory usage: 78.2+ KB

```

```
#extracting numeric and categorical data
```

```

num_cols = [col for col in data.columns if data[col].dtype != 'object']
cat_cols = [col for col in data.columns if data[col].dtype == 'object']

```

```
num_cols
```

```

['age',
 'blood_pressure',
 'specific_gravity',
 'albumin',
 'sugar',
 'blood_glucose_random',
 'blood_urea',
 'serum_creatinine',
 'sodium',
 'potassium',
 'haemoglobin']

```

```
#####check unique values in the categorical data
```

```

for col in cat_cols:
    print(f"{col} has {data[col].unique()} values\n")

    red_blood_cells has [nan 'normal' 'abnormal'] values

    pus_cell has ['normal' 'abnormal' nan] values

    pus_cell_clumps has ['notpresent' 'present' nan] values

    bacteria has ['notpresent' 'present' nan] values

```

```

packed_cell_volume has ['44' '38' '31' '32' '35' '39' '36' '33' '29' '28' nan '16' '24' '37' '30'
'34' '40' '45' '27' '48' '\t?' '52' '14' '22' '18' '42' '17' '46' '23'
'19' '25' '41' '26' '15' '21' '43' '20' '\t43' '47' '9' '49' '50' '53'
'51' '54'] values

white_blood_cell_count has ['7800' '6000' '7500' '6700' '7300' nan '6900' '9600' '12100' '4500'
'12200' '11000' '3800' '11400' '5300' '9200' '6200' '8300' '8400' '10300'
'9800' '9100' '7900' '6400' '8600' '18900' '21600' '4300' '8500' '11300'
'7200' '7700' '14600' '6300' '\t6200' '7100' '11800' '9400' '5500' '5800'
'13200' '12500' '5600' '7000' '11900' '10400' '10700' '12700' '6800'
'6500' '13600' '10200' '9000' '14900' '8200' '15200' '5000' '16300'
'12400' '\t8400' '10500' '4200' '4700' '10900' '8100' '9500' '2200'
'12800' '11200' '19100' '\t?' '12300' '16700' '2600' '26400' '8800'
'7400' '4900' '8000' '12000' '15700' '4100' '5700' '11500' '5400' '10800'
'9900' '5200' '5900' '9300' '9700' '5100' '6600'] values

red_blood_cell_count has ['5.2' nan '3.9' '4.6' '4.4' '5' '4.0' '3.7' '3.8' '3.4' '2.6' '2.8' '4.3'
'3.2' '3.6' '4' '4.1' '4.9' '2.5' '4.2' '4.5' '3.1' '4.7' '3.5' '6.0'
'5.0' '2.1' '5.6' '2.3' '2.9' '2.7' '8.0' '3.3' '3.0' '3' '2.4' '4.8'
'\t?' '5.4' '6.1' '6.2' '6.3' '5.1' '5.8' '5.5' '5.3' '6.4' '5.7' '5.9'
'6.5'] values

hypertension has ['yes' 'no' nan] values

diabetes_mellitus has ['yes' 'no' ' yes' '\tno' '\tyes' nan] values

coronary_artery_disease has ['no' 'yes' '\tno' nan] values

appetite has ['good' 'poor' nan] values

peda_edema has ['no' 'yes' nan] values

aanemia has ['no' 'yes' nan] values

class has ['ckd' 'ckd\t' 'notckd'] values

#####to handel the skewness in the data
def handel_outlier(col):
    data[col] =np.log1p(data[col])

handel_outlier('blood_urea')
handel_outlier('sodium')
handel_outlier('potassium')
handel_outlier('serum_creatinine')
handel_outlier('sugar')

##that explins why soe data still skeness although we to process on it , becaus it stil has null data
data.isna().sum()

```

```

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
haemoglobin        52
packed_cell_volume 70
white_blood_cell_count 105
red_blood_cell_count 130
hypertension       2
diabetes_mellitus  2
coronary_artery_disease 2
appetite           1
peda_edema         1
aanemia            1
class              0
dtype: int64

```

```
#filling null values, we will use two methods, random sampling for higher null values and
# mean/mode sampling for lower null values
```

```
def random_value_imputation(feature):
    random_sample = data[feature].dropna().sample(data[feature].isna().sum())
    random_sample.index = data[data[feature].isnull()].index
    data.loc[data[feature].isnull(),feature] =random_sample
```

```
def impute_mode(feature):
    mode = data[feature].mode()[0]
    data[feature] =data[feature].fillna(mode)
```

```
###filling num columns null values uysing rando sampling method
```

```
for col in num_cols:
    random_value_imputation(col)
```

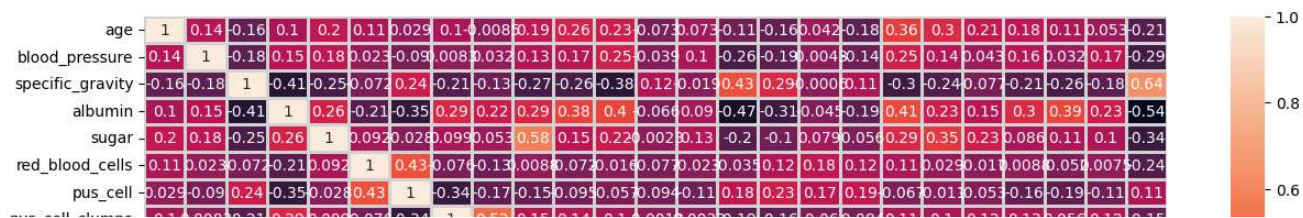
```
data[num_cols].isnull().sum()
```

```
age                0
blood_pressure     0
specific_gravity   0
albumin            0
sugar              0
blood_glucose_random  0
blood_urea         0
serum_creatinine   0
sodium             0
potassium          0
haemoglobin        0
dtype: int64
```

```
# heatmap of data
```

```
plt.figure(figsize = (15, 8))
```

```
sns.heatmap(data.corr(), annot = True, linewidths = 2, linecolor = 'lightgrey')
plt.show()
```



```
# looking at categorical columns
```

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

```
plotnumber = 1
```

```
for column in cat_cols:
```

```
    if plotnumber <= 11:
```

```
        ax = plt.subplot(3, 4, plotnumber)
```

```
        sns.countplot(data[column], palette = 'rocket')
```

```
        plt.xlabel(column)
```

```
    plotnumber += 1
```

```
plt.tight_layout()
```

```
plt.show()
```



```
data[cat_cols].isnull().sum()
```

```
red_blood_cells      0
pus_cell             0
pus_cell_clumps      0
bacteria             0
packed_cell_volume   0
white_blood_cell_count 0
red_blood_cell_count 0
hypertension         0
diabetes_mellitus     0
coronary_artery_disease 0
appetite             0
peda_edema           0
aanemia              0
class                0
dtype: int64
```



```
# checking for null values
```

```
data.isna().sum().sort_values(ascending = False)
```

```
age                0
potassium          0
aanemia            0
peda_edema         0
appetite           0
coronary_artery_disease 0
diabetes_mellitus  0
hypertension       0
red_blood_cell_count 0
white_blood_cell_count 0
packed_cell_volume 0
haemoglobin        0
sodium             0
blood_pressure     0
serum_creatinine   0
blood_urea         0
blood_glucose_random 0
bacteria           0
pus_cell_clumps    0
pus_cell           0
red_blood_cells    0
sugar              0
albumin            0
specific_gravity   0
class              0
dtype: int64
```

```
# checking numerical features distribution
```

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

```
plotnumber = 1
```

```
for column in num_cols:
    if plotnumber <= 14:
        ax = plt.subplot(3, 5, plotnumber)
        sns.distplot(data[column])
        plt.xlabel(column)
```

```
    plotnumber += 1
```

```
plt.tight_layout()
```

```
plt.show()
```


For a guide to updating your code to use the new functions, please see

```
for col in cat_cols:
    print(f"{col} has {data[col].unique()} values\n")

    red_blood_cells has [2 1 0] values

    pus_cell has [1 0 2] values

    pus_cell_clumps has [0 1 2] values

    bacteria has [0 1 2] values

    packed_cell_volume has [32 26 19 20 23 27 24 21 17 16 44 4 12 25 18 22 28 33 15 36 1 40 2 10
6 30 5 34 11 7 13 29 14 3 9 31 8 0 35 43 37 38 41 39 42] values

    white_blood_cell_count has [72 56 70 62 68 92 64 88 18 43 19 10 39 13 49 84 57 77 78 4 90 83 73 59
80 33 35 42 79 12 67 71 27 58 0 66 15 86 51 54 25 22 52 65 16 5 7 23
63 60 26 3 82 28 76 29 46 31 21 1 6 41 44 9 75 87 36 24 11 34 2 20
32 37 38 81 69 45 74 17 30 40 53 14 50 8 91 48 55 85 89 47 61] values

    red_blood_cell_count has [34 49 19 27 25 31 21 17 18 14 5 7 24 12 16 20 22 30 4 23 26 11 28 15
42 32 1 38 2 8 6 48 13 10 9 3 29 0 36 43 44 45 33 40 37 35 46 39
41 47] values

    hypertension has [1 0 2] values

    diabetes_mellitus has [4 3 2 0 1 5] values

    coronary_artery_disease has [1 2 0 3] values

    appetite has [0 1 2] values

    peda_edema has [0 1 2] values

    aanemia has [0 1 2] values

    class has [0 1 2] values
```

data.describe()

	age	blood_pressure	specific_gravity	albumin	sugar	red_blood_cells	pus_cell	pus_cell_clumps	bacteria	blo
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	
mean	51.385000	76.27500	1.017550	1.002500	0.226718	1.262500	0.972500	0.125000	0.075000	
std	17.265631	13.61388	0.005752	1.338646	0.499318	0.655491	0.593823	0.360138	0.299331	
min	2.000000	50.00000	1.005000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	42.000000	70.00000	1.010000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	
50%	55.000000	80.00000	1.020000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	
75%	64.250000	80.00000	1.020000	2.000000	0.000000	2.000000	1.000000	0.000000	0.000000	
max	90.000000	180.00000	1.025000	5.000000	1.791759	2.000000	2.000000	2.000000	2.000000	

8 rows x 25 columns

```
!pip install nbconvert
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: nbconvert in /usr/local/lib/python3.9/dist-packages (6.5.4)
Requirement already satisfied: pygments>=2.4.1 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (2.14.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (2.1.2)
Requirement already satisfied: traitlets>=5.0 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (5.7.1)
Requirement already satisfied: bleach in /usr/local/lib/python3.9/dist-packages (from nbconvert) (6.0.0)
Requirement already satisfied: jupyter-core>=4.7 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (5.3.0)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (5.8.0)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (1.5.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (4.11.2)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (3.1.2)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.7.3)
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages (from nbconvert) (4.9.2)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (1.2.1)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.8.4)
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (from nbconvert) (23.0)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.4)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.7.1)
```



```
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.2.2)
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.9/dist-packages (from jupyter-core>=4.7->nbconvert) (3.2.0)
Requirement already satisfied: jupyter-client>=6.1.12 in /usr/local/lib/python3.9/dist-packages (from nbclient>=0.5.0->nbconvert) (6.1.1)
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.9/dist-packages (from nbformat>=5.1->nbconvert) (2.16.3)
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.9/dist-packages (from nbformat>=5.1->nbconvert) (4.3.3)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.9/dist-packages (from beautifulsoup4->nbconvert) (2.4)
Requirement already satisfied: webencodings in /usr/local/lib/python3.9/dist-packages (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.9/dist-packages (from bleach->nbconvert) (1.16.0)
Requirement already satisfied: pyparsing!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /usr/local/lib/python3.9/dist-packages (from jsonschema)
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.9/dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.9/dist-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert)
Requirement already satisfied: tornado>=4.1 in /usr/local/lib/python3.9/dist-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.9/dist-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert)
```

```
! jupyter nbconvert --to html kidney.ipynb
```

```
[NbConvertApp] Converting notebook kidney.ipynb to html
[NbConvertApp] Writing 1559074 bytes to kidney.html
```

```
!pip install flask-ngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting flask-ngrok
  Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from flask-ngrok) (2.27.1)
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.9/dist-packages (from flask-ngrok) (2.2.3)
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok) (2.1.2)
Requirement already satisfied: importlib-metadata>=3.6.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok) (6.3.0)
Requirement already satisfied: Werkzeug>=2.2.2 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok) (2.2.3)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok) (3.1.2)
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok) (8.1.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests->flask-ngrok) (2022.12.7)
Requirement already satisfied: charset-normalizer<=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests->flask-ngrok) (2.0.12)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests->flask-ngrok) (1.26.15)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests->flask-ngrok) (3.4)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from importlib-metadata>=3.6.0->Flask>=0.8->flask-ngrok)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from Jinja2>=3.0->Flask>=0.8->flask-ngrok) (2.1.1)
Installing collected packages: flask-ngrok
Successfully installed flask-ngrok-0.0.25
```

✓ 4s completed at 10:28 AM

