import numpy as np   
import pandas as pd  
import matplotlib.pyplot as plt

[ ]

df=pd.read\_csv(r"/content/kidney\_disease.csv")

[ ]

df.head()

[ ]

df.describe()

[ ]

df.rename(columns = {  
'age' : 'age',  
'bp' : 'blood pressure',  
'sg' : 'specific gravity',  
'al' : 'albumin',  
'su' : 'sugar',  
'rbc' : 'red blood cells',  
'pc' : 'pus cell',  
'pcc' : 'pus cell clumps',  
'ba' : 'bacteria',  
'bgr' : 'blood glucose random',  
'bu' : 'blood urea',  
'sc' : 'serum creatinine',  
'sod' : 'sodium',  
'pot' : 'potassium',  
'hemo' : 'hemoglobin',  
'pcv' : 'packed cell volume',  
'wc' : 'white blood cell count',  
'rc' : 'red blood cell count',  
'htn' : 'hypertension',  
'dm' : 'diabetes mellitus',  
'cad' : 'coronary artery disease',  
'appet' : 'appetite',  
'pe' : 'pedal edema',  
'ane' : 'anemia',  
'class' : 'class'}, inplace = True)

[ ]

df.head()

[ ]

df.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 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 cell 335 non-null object

8 pus cell clumps 396 non-null object

9 bacteria 396 non-null object

10 blood glucose 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 hemoglobin 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 pedal edema 399 non-null object

24 anemia 399 non-null object

25 classification 400 non-null object

dtypes: float64(11), int64(1), object(14)

memory usage: 81.4+ KB

[ ]

df.isnull()

[ ]

df.isnull().sum()

id 0

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

diabetes mellitus 2

coronary artery disease 2

appetite 1

pedal edema 1

anemia 1

classification 0

dtype: int64

[ ]

df.shape[0]

400

[ ]

df.isnull().sum()/df.shape[0]

id 0.0000

age 0.0225

blood pressure 0.0300

specific gravity 0.1175

albumin 0.1150

sugar 0.1225

red blood cells 0.3800

pus cell 0.1625

pus cell clumps 0.0100

bacteria 0.0100

blood glucose random 0.1100

blood urea 0.0475

serum creatinine 0.0425

sodium 0.2175

potassium 0.2200

hemoglobin 0.1300

packed cell volume 0.1750

white blood cell count 0.2625

red blood cell count 0.3250

hypertension 0.0050

diabetes mellitus 0.0050

coronary artery disease 0.0050

appetite 0.0025

pedal edema 0.0025

anemia 0.0025

classification 0.0000

dtype: float64

[ ]

((df.isnull().sum()/df.shape[0])\*100).sort\_values(ascending=False)

red blood cells 38.00

red blood cell count 32.50

white blood cell count 26.25

potassium 22.00

sodium 21.75

packed cell volume 17.50

pus cell 16.25

hemoglobin 13.00

sugar 12.25

specific gravity 11.75

albumin 11.50

blood glucose random 11.00

blood urea 4.75

serum creatinine 4.25

blood pressure 3.00

age 2.25

bacteria 1.00

pus cell clumps 1.00

hypertension 0.50

diabetes mellitus 0.50

coronary artery disease 0.50

anemia 0.25

appetite 0.25

pedal edema 0.25

id 0.00

classification 0.00

dtype: float64

[ ]

df.drop(["id"], axis=1 , inplace=True)

[ ]

df.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 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 diabetes mellitus 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 classification 400 non-null object

dtypes: float64(11), object(14)

memory usage: 78.2+ KB

[ ]

df['red blood cell count']=pd.to\_numeric(df['red blood cell count'],errors='coerce')

[ ]

df.describe()

[ ]

df.describe(include="all").T

[ ]

print(df.apply(lambda col: col.unique()))

age [48.0, 7.0, 62.0, 51.0, 60.0, 68.0, 24.0, 52.0...

blood pressure [80.0, 50.0, 70.0, 90.0, nan, 100.0, 60.0, 110...

specific gravity [1.02, 1.01, 1.005, 1.015, nan, 1.025]

albumin [1.0, 4.0, 2.0, 3.0, 0.0, nan, 5.0]

sugar [0.0, 3.0, 4.0, 1.0, nan, 2.0, 5.0]

red blood cells [nan, normal, abnormal]

pus cell [normal, abnormal, nan]

pus cell clumps [notpresent, present, nan]

bacteria [notpresent, present, nan]

blood glucose random [121.0, nan, 423.0, 117.0, 106.0, 74.0, 100.0,...

blood urea [36.0, 18.0, 53.0, 56.0, 26.0, 25.0, 54.0, 31....

serum creatinine [1.2, 0.8, 1.8, 3.8, 1.4, 1.1, 24.0, 1.9, 7.2,...

sodium [nan, 111.0, 142.0, 104.0, 114.0, 131.0, 138.0...

potassium [nan, 2.5, 3.2, 4.0, 3.7, 4.2, 5.8, 3.4, 6.4, ...

hemoglobin [15.4, 11.3, 9.6, 11.2, 11.6, 12.2, 12.4, 10.8...

packed cell volume [44, 38, 31, 32, 35, 39, 36, 33, 29, 28, nan, ...

white blood cell count [7800, 6000, 7500, 6700, 7300, nan, 6900, 9600...

red blood cell count [5.2, nan, 3.9, 4.6, 4.4, 5.0, 4.0, 3.7, 3.8, ...

hypertension [yes, no, nan]

diabetes mellitus [yes, no, yes, \tno, \tyes, nan]

coronary artery disease [no, yes, \tno, nan]

appetite [good, poor, nan]

pedal edema [no, yes, nan]

anemia [no, yes, nan]

classification [ckd, ckd\t, notckd]

dtype: object

[ ]

df['packed cell volume'] = df['packed cell volume'].replace(to\_replace = {'\t?': np.nan,'\t43':'43'})  
df['diabetes mellitus'] = df['diabetes mellitus'].replace(to\_replace = {'\tno':'no','\tyes':'yes',' yes':'yes'})  
df['coronary artery disease'] = df['coronary artery disease'].replace(to\_replace = '\tno', value='no')  
df['classification'] =df['classification'].replace(to\_replace = 'ckd\t', value = 'ckd')  
df['packed cell volume'] = pd.to\_numeric(df['packed cell volume'], errors='coerce')

[ ]

print(df.apply(lambda col: col.unique()))

age [48.0, 7.0, 62.0, 51.0, 60.0, 68.0, 24.0, 52.0...

blood pressure [80.0, 50.0, 70.0, 90.0, nan, 100.0, 60.0, 110...

specific gravity [1.02, 1.01, 1.005, 1.015, nan, 1.025]

albumin [1.0, 4.0, 2.0, 3.0, 0.0, nan, 5.0]

sugar [0.0, 3.0, 4.0, 1.0, nan, 2.0, 5.0]

red blood cells [nan, normal, abnormal]

pus cell [normal, abnormal, nan]

pus cell clumps [notpresent, present, nan]

bacteria [notpresent, present, nan]

blood glucose random [121.0, nan, 423.0, 117.0, 106.0, 74.0, 100.0,...

blood urea [36.0, 18.0, 53.0, 56.0, 26.0, 25.0, 54.0, 31....

serum creatinine [1.2, 0.8, 1.8, 3.8, 1.4, 1.1, 24.0, 1.9, 7.2,...

sodium [nan, 111.0, 142.0, 104.0, 114.0, 131.0, 138.0...

potassium [nan, 2.5, 3.2, 4.0, 3.7, 4.2, 5.8, 3.4, 6.4, ...

hemoglobin [15.4, 11.3, 9.6, 11.2, 11.6, 12.2, 12.4, 10.8...

packed cell volume [44.0, 38.0, 31.0, 32.0, 35.0, 39.0, 36.0, 33....

white blood cell count [7800, 6000, 7500, 6700, 7300, nan, 6900, 9600...

red blood cell count [5.2, nan, 3.9, 4.6, 4.4, 5.0, 4.0, 3.7, 3.8, ...

hypertension [yes, no, nan]

diabetes mellitus [yes, no, nan]

coronary artery disease [no, yes, nan]

appetite [good, poor, nan]

pedal edema [no, yes, nan]

anemia [no, yes, nan]

classification [ckd, notckd]

dtype: object

[ ]

df\_num=[feature for feature in df.columns if df[feature].dtypes=='float64']  
print('total numerical column :',len(df\_num))  
print(df\_num)

total numerical column : 13

['age', 'blood pressure', 'specific gravity', 'albumin', 'sugar', 'blood glucose random', 'blood urea', 'serum creatinine', 'sodium', 'potassium', 'hemoglobin', 'packed cell volume', 'red blood cell count']

[ ]

df\_cat=[feature for feature in df.columns if df[feature].dtypes=='O']  
print('total categorical column :',len(df\_cat))  
print(df\_cat)

total categorical column : 12

['red blood cells', 'pus cell', 'pus cell clumps', 'bacteria', 'white blood cell count', 'hypertension', 'diabetes mellitus', 'coronary artery disease', 'appetite', 'pedal edema', 'anemia', 'classification']

[ ]

df[df\_num].describe(include="all").T

[ ]

df[df\_cat].describe(include="all").T

[ ]

import seaborn as sns  
import plotly.express as px

[ ]

sns.set(rc={"figure.figsize":(15, 8)})  
for i in df.columns:  
    sns.countplot(x=df[i],data=df)  
    plt.xlabel(i)  
    plt.ylabel("Count")  
    plt.title("target Class")  
    plt.show()  
    print("\n")

[ ]

corr\_df = df.corr()  
f,ax=plt.subplots(figsize=(15,15))  
sns.heatmap(corr\_df,annot=True,fmt=".2f",ax=ax,linewidths=0.5,linecolor="yellow")  
print(df.corr())

[ ]

for i in df.columns:  
    if df[i].isna().sum() > 0 :  
        if df[i].dtype == 'float64':  
            df[i].fillna(df[i].median(), inplace=True)  
        else:  
            df[i].fillna(df[i].mode()[0], inplace=True)

[ ]

import warnings  
  
warnings.filterwarnings('ignore')

[ ]

def violin(col):   
    fig = px.violin(df, y=col, x="classification", color="classification", box=True, points="all", hover\_data=df.columns)  
    return fig.show()  
def kde\_plot(feature):  
    grid = sns.FacetGrid(df, hue="classification", aspect = 2)  
    grid.map(sns.kdeplot, feature)  
    grid.add\_legend()

[ ]

fig = px.scatter(df, x="red blood cell count", y="hemoglobin", color="classification")  
fig.show()

[ ]

kde\_plot('red blood cell count')

[ ]

for i in df.columns:

    fig = px.scatter(df, x=i, y="blood pressure", color="classification")

    fig.show()





[ ]

for i in df.columns:  
    fig = px.scatter(df, x=i, y= "serum creatinine", color="classification")  
    fig.show()

[ ]

  violin('blood pressure')

[ ]

for i in df.columns:  
    fig = px.scatter(df, x=i, y="hemoglobin", color="classification")  
    fig.show()

[ ]

violin('age')



[ ]

sns.pairplot(df, hue="classification",corner=False)



[Colab paid products](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links)- [Cancel contracts here](https://colab.research.google.com/cancel-subscription)