Code

Text

import pandas as pd

import numpy as np

from collections import Counter as c import matplotlib.pyplot as plt

import seaborn as sns

import missingno as msno

from sklearn.metrics import accuracy\_score,confusion\_matrix from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder

from sklearn.linear\_model import LogisticRegression import pickle

import csv

from google.colab import files uploaded=files.upload()

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Choose Files

Saving kidney\_disease.csv to kidney\_disease.csv

pd.read\_csv("kidney\_disease.csv",encoding="latin")

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **id** | **age** | **bp** | **sg** | **al** | **su** | **rbc** | **pc** | **pcc** | **ba ... pcv** | **wc** | **rc** | **htn** | **dm** |
| **0** | 0 | 48.0 | 80.0 | 1.020 | 1.0 | 0.0 | NaN | normal | notpresent | notpresent ... 44 | 7800 | 5.2 | yes | yes |
| **1** | 1 | 7.0 | 50.0 | 1.020 | 4.0 | 0.0 | NaN | normal | notpresent | notpresent ... 38 | 6000 | NaN | no | no |
| **2** | 2 | 62.0 | 80.0 | 1.010 | 2.0 | 3.0 | normal | normal | notpresent | notpresent ... 31 | 7500 | NaN | no | yes |
| **3** | 3 | 48.0 | 70.0 | 1.005 | 4.0 | 0.0 | normal | abnormal | present | notpresent ... 32 | 6700 | 3.9 | yes | no |
| **4** | 4 | 51.0 | 80.0 | 1.010 | 2.0 | 0.0 | normal | normal | notpresent | notpresent ... 35 | 7300 | 4.6 | no | no |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... ... ... | ... | ... | ... | .. |
| **395** | 395 | 55.0 | 80.0 | 1.020 | 0.0 | 0.0 | normal | normal | notpresent | notpresent ... 47 | 6700 | 4.9 | no | no |
| **396** | 396 | 42.0 | 70.0 | 1.025 | 0.0 | 0.0 | normal | normal | notpresent | notpresent ... 54 | 7800 | 6.2 | no | no |
| **397** | 397 | 12.0 | 80.0 | 1.020 | 0.0 | 0.0 | normal | normal | notpresent | notpresent ... 49 | 6600 | 5.4 | no | no |
| **398** | 398 | 17.0 | 60.0 | 1.025 | 0.0 | 0.0 | normal | normal | notpresent | notpresent ... 51 | 7200 | 5.9 | no | no |
| **399** | 399 | 58.0 | 80.0 | 1.025 | 0.0 | 0.0 | normal | normal | notpresent | notpresent ... 53 | 6800 | 6.1 | no | no |

400 rows × 26 columns

data=pd.read\_csv("kidney\_disease.csv") data.head()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **id** | **age** | **bp** | **sg** | **al** | **su** | **rbc** | **pc** | **pcc** | **ba ... pcv** | **wc** | **rc** | **htn** | **dm c** |
| **0** 0 | 48.0 | 80.0 | 1.020 | 1.0 | 0.0 | NaN | normal | notpresent | notpresent ... 44 | 7800 | 5.2 | yes | yes |
| **1** 1 | 7.0 | 50.0 | 1.020 | 4.0 | 0.0 | NaN | normal | notpresent | notpresent ... 38 | 6000 | NaN | no | no |
| **2** 2 | 62.0 | 80.0 | 1.010 | 2.0 | 3.0 | normal | normal | notpresent | notpresent ... 31 | 7500 | NaN | no | yes |
| **3** 3 | 48.0 | 70.0 | 1.005 | 4.0 | 0.0 | normal | abnormal | present | notpresent ... 32 | 6700 | 3.9 | yes | no |
| **4** 4 | 51.0 | 80.0 | 1.010 | 2.0 | 0.0 | normal | normal | notpresent | notpresent ... 35 | 7300 | 4.6 | no | no |

5 rows × 26 columns

data.columns

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=['id','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']

data.columns

Index(['id', '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')

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 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 diabetesmellitus | 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 class | 400 | non-null |  | object |

dtypes: float64(11), int64(1), object(14) memory usage: 81.4+ KB

data.isnull().any()

id False

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 False

dtype: bool

from IPython.utils.path import target\_outdated

data['id'].fillna(data['id'].mean(),inplace=True)

data['age'].fillna(data['age'].mean(),inplace=True)

data['blood\_pressure'].fillna(data['blood\_pressure'].mean(),inplace=True)

data['specific\_gravity'].fillna(data['specific\_gravity'].mode()[0],inplace=True) data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)

data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)

data['red\_blood\_cells'].fillna(data['red\_blood\_cells'].mode()[0],inplace=True) data['pus\_cell'].fillna(data['pus\_cell'].mode()[0],inplace=True)

data['pus\_cell\_clumps'].fillna(data['pus\_cell\_clumps'].mode()[0],inplace=True) data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)

data['blood glucose random'].fillna(data['blood glucose random'].mean(),inplace=True) data['blood\_urea'].fillna(data['blood\_urea'].mean(),inplace=True)

data['serum\_creatinine'].fillna(data['serum\_creatinine'].mean(),inplace=True) data['sodium'].fillna(data['sodium'].mean(),inplace=True)

data['potassium'].fillna(data['potassium'].mean(),inplace=True)

data['hemoglobin'].fillna(data['hemoglobin'].mean(),inplace=True)

data['packed\_cell\_volume'].fillna(data['packed\_cell\_volume'].mode()[0],inplace=False)

data['white\_blood\_cell\_count'].fillna(data['white\_blood\_cell\_count'].mode()[0],inplace=False) data['red\_blood\_cell\_count'].fillna(data['red\_blood\_cell\_count'].mode()[0],inplace=False)

data['hypertension'].fillna(data['hypertension'].mode()[0],inplace=False)

data['diabetesmellitus'].fillna(data['diabetesmellitus'].mode()[0],inplace=True)

data['coronary\_artery\_disease'].fillna(data['coronary\_artery\_disease'].mode()[0],inplace=True) data['appetite'].fillna(data['appetite'].mode()[0],inplace=True)

data['pedal\_edema'].fillna(data['pedal\_edema'].mode()[0],inplace=False) data['anemia'].fillna(data['anemia'].mode()[0],inplace=True)

data['class'].fillna(data['class'].mode()[0],inplace=True)

catcols=set(data.dtypes[data.dtypes=='0'].index.values) print(catcols)

set()

for i in catcols:

print("columns: ",i)

print(c(data[i]))

print('\*'\*120+'\n')

contcols=set(data.dtypes[data.dtypes!='0'].index.values) print(contcols)

{'hemoglobin', 'albumin', 'anemia', 'class', 'pus\_cell', 'red\_blood\_cell\_count', 'serum\_creatinine', 'id', 'bacteria'

for i in contcols:

print("Continous Columns: ",i) print(c(data[i]))

print('8'\*120+'\n')

Continous Columns: hemoglobin



Counter({12.526436781609195: 52, 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, 1

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: albumin

Counter({0.0: 245, 1.0: 44, 2.0: 43, 3.0: 43, 4.0: 24, 5.0: 1})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: anemia

Counter({'no': 340, 'yes': 60})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: class

Counter({'ckd': 248, 'notckd': 150, 'ckd\t': 2})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: pus\_cell

Counter({'normal': 324, 'abnormal': 76})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: red\_blood\_cell\_count

Counter({'5.2': 148, '4.5': 16, '4.9': 14, '4.7': 11, '3.9': 10, '4.8': 10, '4.6': 9, '3.4': 9, '3.7': 8, '5.0': 8,

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: serum\_creatinine

Counter({1.2: 40, 1.1: 24, 1.0: 23, 0.5: 23, 0.7: 22, 0.9: 22, 0.6: 18, 0.8: 17, 3.072454308093995: 17, 2.2: 10, 1.

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: id

Counter({0: 1, 1: 1, 2: 1, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 1, 12: 1, 13: 1, 14: 1, 15: 1, 16:

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: bacteria

Counter({'notpresent': 378, 'present': 22})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: packed\_cell\_volume

Counter({'41': 91, '52': 21, '44': 19, '48': 19, '40': 16, '43': 14, '45': 13, '42': 13, '32': 12, '36': 12, '33':

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: pus\_cell\_clumps

Counter({'notpresent': 358, 'present': 42})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: hypertension Counter({'no': 253, 'yes': 147})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: diabetesmellitus

Counter({'no': 260, 'yes': 134, '\tno': 3, '\tyes': 2, ' yes': 1})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

Continous Columns: red\_blood\_cells

Counter({'normal': 353, 'abnormal': 47})

8888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888

catcols.remove('red\_blood\_cell\_count') catcols.remove('packed\_cell\_volume')

catcols.remove('white\_blood\_cell\_count') print(catcols)

catcols=['anemia','pedal\_edama','appetite','bacteria','class','coronary\_artery\_disease','diabetesmellitus','hypertension','

from sklearn.preprocessing import LabelEncoder for i in catcols:

print("LABEL ENCODING OF: ",i)

LEi=LabelEncoder() print(c(data[i]))

data[i]=LEi.fit\_transform(data[i]) print(c(data[i]))

print('\*'\*100)

contcols=set(data.dtypes[data.dtypes!='0'].index.values) print(contcols)

for i in contcols:

print("Continous Columns: ",i) print(c(data[i]))

print('\*'\*120+'\n')

contcols.remove('specific\_gravity') contcols.remove('albumin')

contcols.remove('sugar') print(contcols)

contcols.add('red\_blood\_cell\_count') contcols.add('packed\_cell\_volume')

contcols.add('white\_blood\_cell\_count') print(contcols)

contcols.add('specific\_gravity') contcols.add('albumin')

contcols.add('sugar') print(contcols)

data['coronary\_artery\_disease']=data.coronary\_artery\_disease.replace('\tno','no') c(data['coronary\_artery\_disease'])

data['diabetesemellitus']=data.diabetesemellitus.replace(to\_replace={'tno':'no','tyes':'yes'}) c(data['diabetesemellitus'])

data.describe()

sns.displot(data.age)

import matplotlib.pyplot as plt fig=plt.figure(figsize =(5,5))

plt.scatter(data['age'],data['blood\_pressure'],color='blue') plt.xlabel('age')

plt.ylabel('blood\_pressure')

plt.title("age vs blood Scatter plot")

from sklearn.utils import column\_or\_1d

plt.figure(figsize=(20,15),facecolor='white') plotnumber=1

for i in contcols:

if plotnumber<=11:

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

plt.scatter(data['age'],data[column]) plt.xlabel(column,figsize=20)

f,ax=plt.subplot(figsize=(18,10))

sns.heatmap(data.corr(),annot=True,fmt="2f",ax=ax,linewidth=0.5,linecolor="orange") plt.xticks(rotation=45)

plt.yticks(rotation=45) plt.show()

sns.countplot(data['class'])

from sklrearn.preprocessing import StandardScaler sc=StandardScaler()

x\_bal=sc.fit\_transform(x)

setcols['red\_blood\_cells','pus\_cell','blood glucose random','blood\_urea','pedal\_edema','anemia','diabetesmellitus','coronar x=pd.DataFrame(data,columns=setcols)

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

print(y.shape)

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)

import tensorflow

from tensorflow.keras.models import Sequential from tensorflow.layers import Dense

classification=Sequential()

classification.add(Dense(30,activation='relu')) classification.add(Dense(128,activation='relu'))

classification.add(Dense(64,activation='relu')) classification.add(Dense(32,activation='relu')) classification.add(Dense(1,activation='sigmoid'))

classification.compile(optimizer='adam',loss='binary\_crossentropy',metrics=['accuracy']) classification.fit(x\_train,y\_train,batch\_size=10,validation\_split=0.2,epochs=100)

from sklearn.ensemble import RandomForestClassifier

rfc=RandomForestClassifier(n\_estimator=10,criterion='entropy') rfc.fit(x\_train,y\_train)

y\_predict=rfc.predict(x\_test)

y\_predict\_train=rfc.predict(x\_train)

from sklearn.metrics import accuracy\_score,classification\_report y\_predict=lgr.predict(x\_test)

from sklearn.tree import DecisionTreeClassifier

dtc=DecisionTreeClassifier(max\_depth=4,splitter='best',criterion='entropy') dtc.fit(x\_train,y\_train)

y\_predict=dtc.predict(x\_test) y\_predict

y\_predict\_train=dtc.predict(x\_train)

from sklearn.linear\_model import LogisticRegerssion lgr=LogisticRegression()

lgr.fit(x\_train,y\_train)

y\_pred=lgr.predict([[1,1,121.00000,36,0,0,0,1,0]]) print(y\_pred)

(y\_pred)

y\_pred=dtc.predict([[1,1,121.00000,36.0,0,0,01,0]]) print(y\_pred)

(y\_pred)

y\_pred=rfc.predict([[1,1,121.00000,36.0,0,0,0,1,0]]) print(y\_pred)

(y\_pred)

classification.save("ckd.h5")

y\_pred=classification.predict(x\_test) y\_pred

y\_pred=(y\_pred>0.5) y\_pred

def predict\_exit(sample\_value):

test=classification.predict([[1,1,121.00000,36.0,0,0,1,0]]) if test==1:

print('Prediction: High chance of CKD!') else:

print('Prediction: Low chance of CKD.')

 

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.