SPRINT 2

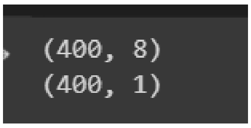
ML MODEL CREATION

Importing the packages

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import missingno as msng

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
```

1.splitting the dataset

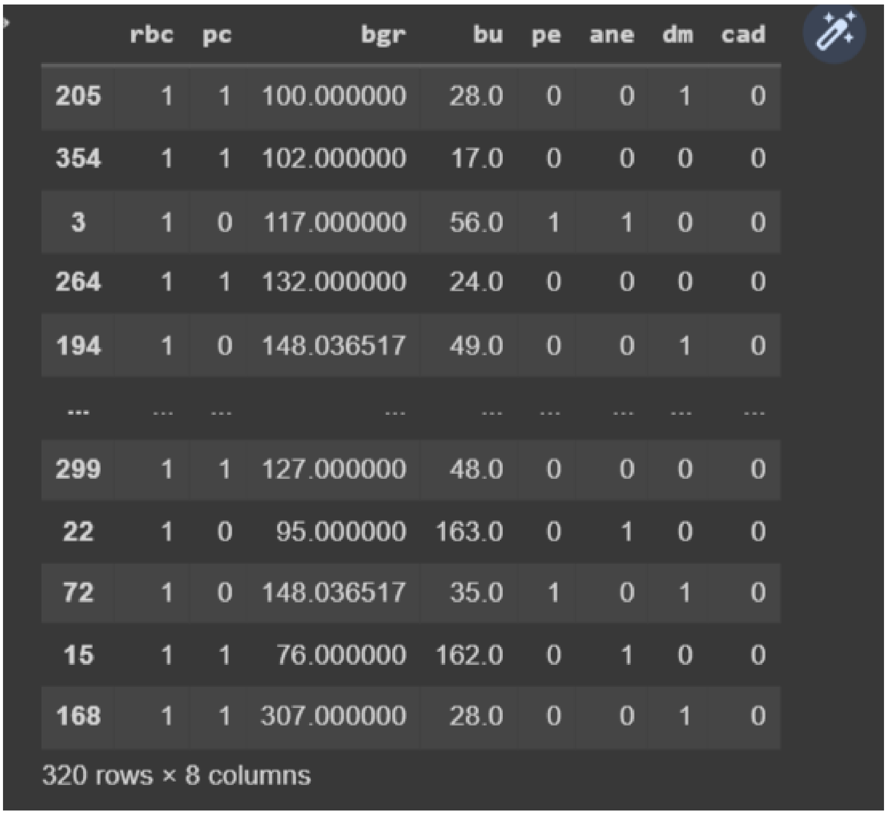
```
for i in catcols:

    print("LABEL ENCODING OF :",i)

    le=LabelEncoder()

    print(c(data[i]))

    data[i]=le.fit_transform(data[i])

    print(c(data[i]))

    print('*'*100)


data['rbc']=le.fit_transform(data['rbc'])

selcols=['rbc','pc','bgr','bu','pe','ane','dm','cad']

x=pd.DataFrame(data,columns=selcols)

y=pd.DataFrame(data,columns=['classification'])


print(x.shape)

print(y.shape)
```



```
(400, 8)
(400, 1)
```

xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=2)

xtrain



	rbc	pc	bgr	bu	pe	ane	dm	cad
205	1	1	100.000000	28.0	0	0	1	0
354	1	1	102.000000	17.0	0	0	0	0
3	1	0	117.000000	56.0	1	1	0	0
264	1	1	132.000000	24.0	0	0	0	0
194	1	0	148.036517	49.0	0	0	1	0
...	...	...	...	...	...	...	...	...
299	1	1	127.000000	48.0	0	0	0	0
22	1	0	95.000000	163.0	0	1	0	0
72	1	0	148.036517	35.0	1	0	1	0
15	1	1	76.000000	162.0	0	1	0	0
168	1	1	307.000000	28.0	0	0	1	0

320 rows × 8 columns

2. Model creation

lgr=LogisticRegression()

lgr.fit(xtrain.values,ytrain.values)

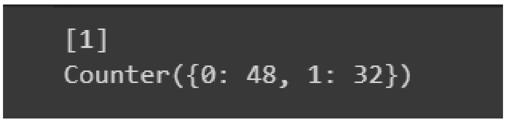
ypred=lgr.predict(xtest)

ypred1=lgr.predict([[129,99,1,0,0,1,0,1]])

print(ypred1)

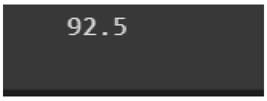
c(ypred)

3. Accuracy , Confustion Matrix , Classification Report



```
[1]
Counter({0: 48, 1: 32})
```

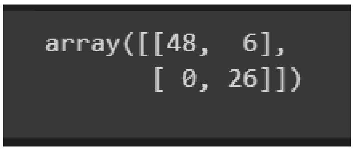
print(accuracy_score(ytest,ypred)*100)



```
92.5
```

confmat=confusion_matrix(ytest,ypred)

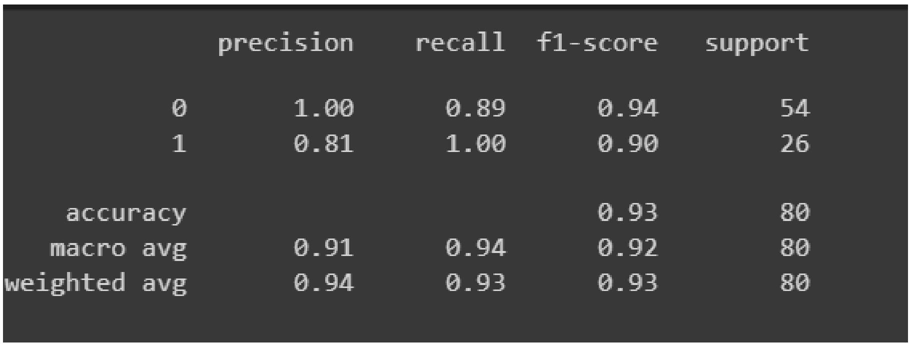
confmat



```
array([[48,  6],
       [ 0, 26]])
```

from sklearn.metrics import classification_report

print(classification_report(ytest, ypred))

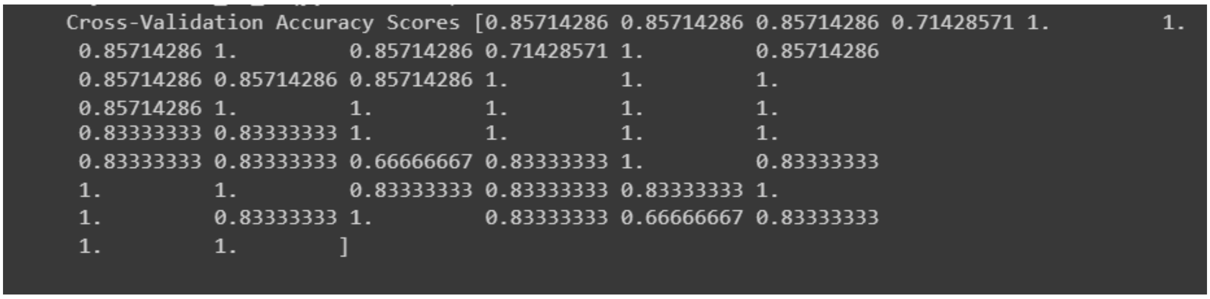


```
              precision    recall  f1-score   support

           0       1.00      0.89      0.94        54
           1       0.81      1.00      0.90        26

    accuracy                           0.93        80
   macro avg       0.91      0.94      0.92        80
weighted avg       0.94      0.93      0.93        80
```

from sklearn.model_selection import cross_val_score

scores = cross_val_score(lgr, xtrain, ytrain, cv=50)

print('Cross-Validation Accuracy Scores', scores)



```
Cross-Validation Accuracy Scores [0.85714286 0.85714286 0.85714286 0.71428571 1.         1.
 0.85714286 1.         0.85714286 0.71428571 1.         0.85714286
 0.85714286 0.85714286 0.85714286 1.         1.         1.
 0.85714286 1.         1.         1.         1.         1.
 0.83333333 0.83333333 1.         1.         1.         1.
 0.83333333 0.83333333 0.66666667 0.83333333 1.         0.83333333
 1.         1.         0.83333333 0.83333333 0.83333333 1.
 1.         0.83333333 1.         0.83333333 0.66666667 0.83333333
 1.         1.        ]
```

Model Saving in pickle

Import pickle

pickle.dump(lgr,open('CKD.pkl','wb'))