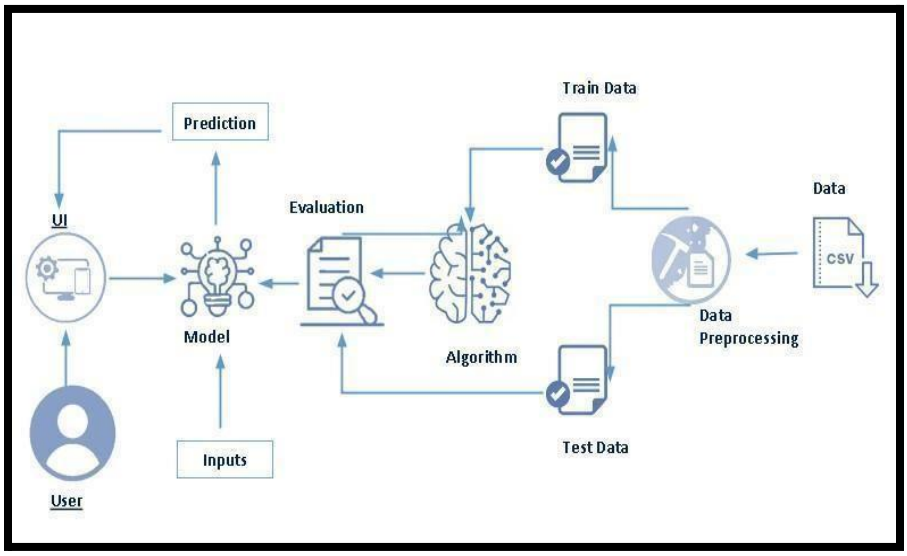
EARLY PREDICTION FOR CHRONIC KIDNEY DISEASE DETECTION: A PROGRESSIVE APPROACH TO HEALTH MANAGEMENT

OVERVIEW:

Chronic Kidney Disease (CKD) is a major medical problem and can be cured if treated in the early stages. Usually, people are not aware that medical tests we take for different purposes could contain valuable information concerning kidney diseases. Consequently, attributes of various medical tests are investigated to distinguish which attributes may contain helpful information about the disease. The information says that it helps us to measure the severity of the problem, the predicted survival of the patient after the illness, the pattern of the disease and work for curing the disease.

In world as we know most of the people are facing so many disease and as this can be cured if we treat people in early stages this project can use a model to predict the Chronic Kidney Disease which can help in treatments of peoples who are suffer from this disease.

Technical Architecture:



Project Flow:

User interacts with the UI to enter the input.

Entered input is by the model which is integrated.

Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

Define Problem / Problem Understanding Specify the business problem

- Business requirements
- Literature Survey
- Collect the dataset
- Data Preparation

Exploratory Data Analysis

Descriptive statistical

- Visual Analysis
- Training the model in multiple algorithms
- Testing the model

Performance Testing & Evaluate the results

Testing model with multiple evaluation metrics

- Evaluate the results

Model Deployment

Save the best model

- Integrate with Web Framework

Project Demonstration & Documentation

Record explanation Video for project end to end solution

- Project Documentation-Step by step project development procedure

Project Structure:

Create the Project folder which contains files as shown below

We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.

CKD.pkl is our saved model. Further we will use this model for flask integration.

Training folder contains a model training file.

Define Problem / Problem Understanding

Activity 1: Specify the business problem

Refer Project Description

Activity 2: Business requirements

The business requirements for a machine learning model to predict chronic kidney disease include the ability to accurately predict the based on given information, the number of false positives (predicting diseased) and false negatives (not diseased). Provide an explanation for the model's decision, to comply with regulations and improve transparency.

Activity 3: Literature Survey (Student Will Write)

Chronic kidney disease (CKD) is a significant public health issue, affecting an estimated 14% of the global population. The disease is characterized by a gradual loss of kidney function over time, leading to a range of serious health complications, including end-stage renal disease (ESRD) requiring dialysis or kidney transplant. Early detection and management of CKD is crucial to prevent progression to ESRD and improve patient outcomes.

There have been numerous studies in recent years aimed at developing accurate and efficient methods for predicting CKD progression. These studies have employed a variety of techniques, including machine learning, deep learning, and artificial neural networks.

Activity 4: Social or Business Impact.

On a social level, early detection and prediction of CKD can lead to improved patient outcomes and quality of life. By identifying individuals at risk for CKD, healthcare providers can intervene early and slow the progression of the disease through lifestyle changes, medication management, and other treatments. This can help prevent the need for dialysis or kidney transplantation, which can be costly and life-altering for patients. Additionally, early prediction can also help reduce the overall burden of CKD on the healthcare system by reducing the number of hospitalizations and emergency room visits.

Purpose:

There are limited data to describe academic achievement outcomes for children with mild to moderate pediatric chronic kidney disease (CKD). The objective of this study was to describe the prevalence of low academic achievement in patients with mild to moderate CKD.

Design, Setting, Participants, and Measurements:

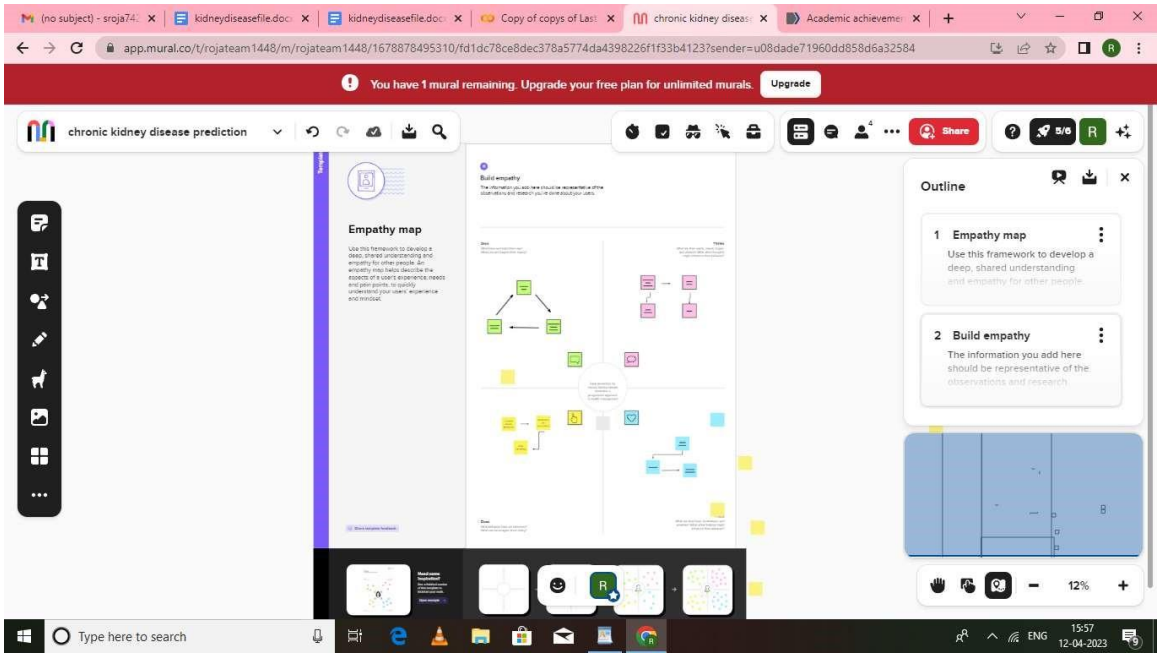
Wechsler Individual Achievement Test, Second Edition, Abbreviated (WIAT-II-A) data were collected at entry into the Chronic Kidney Disease in Children (CKD) study. Achievement in basic reading, spelling, mathematics, and total achievement was evaluated with a focus on the effects of CKD-related variables and school-based characteristics on academic achievement.

The physical sequence of pediatric chronic kidney disease (CKD) is well characterized. Data support the parallel presence of subtle deficits in even mild to moderate pediatric CKD.
Specifically, cognitive executive function, the ability to plan and make complex decisions, may be impaired in CKD . As with executive function, behavioral concerns are reported by parents of children with CKD. Although the specific mechanisms and behavioral dysfunction associated with CKD are not fully understood, there is evidence to suggest that decline in renal function, early age of onset, and increase the risk for and behavioral shortfalls even before the disease advances to the need for dialysis or transplantation .

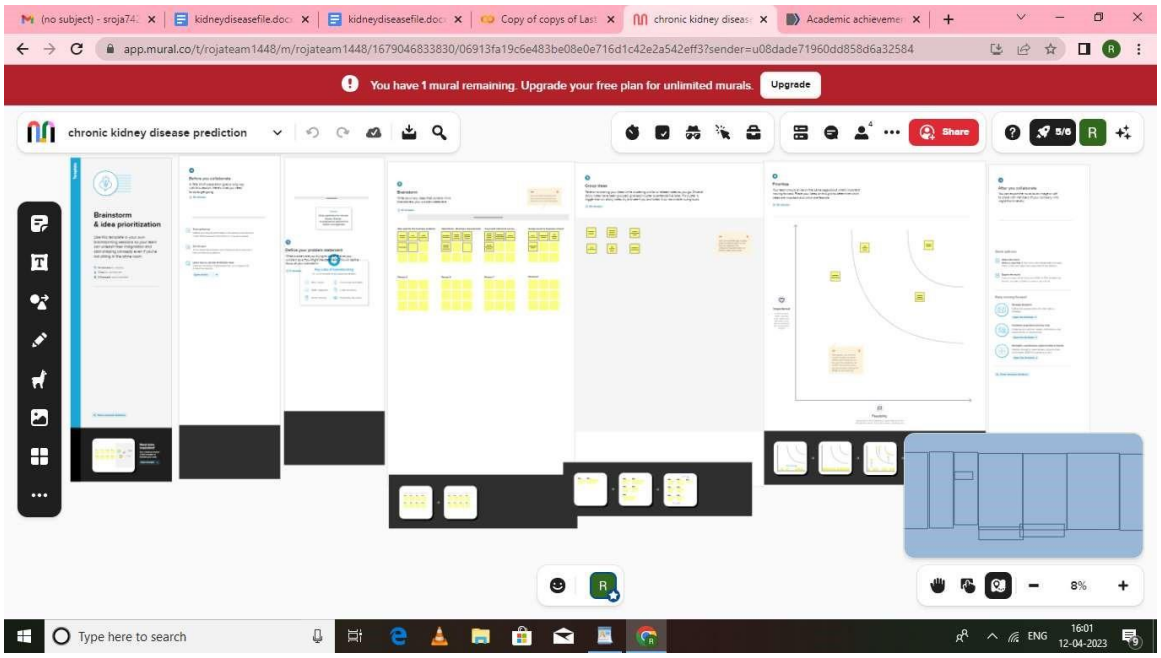
PROBLEM DEFINITION AND DESIGN THINKING

Machine learning has become an increasingly popular tool in recent years, given its ability to automatically detect patterns in data and make predictions about future events .This can be extremely useful for making decisions in a wide range of domain from financial trading to medical diagnoses

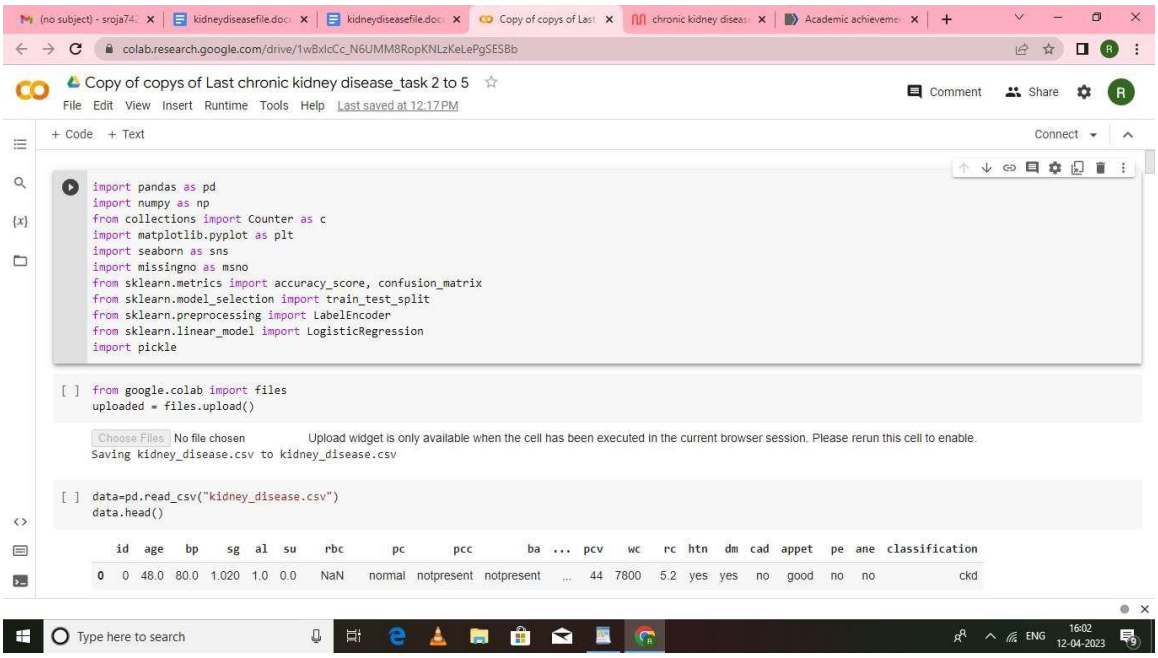
Empathy Map :

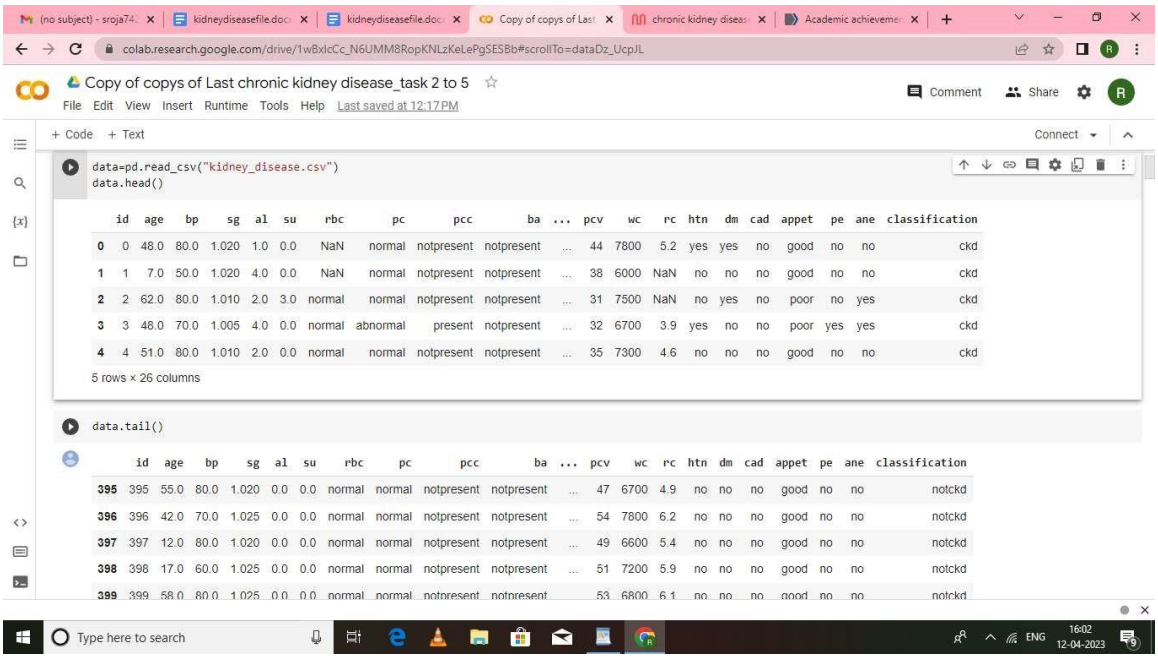


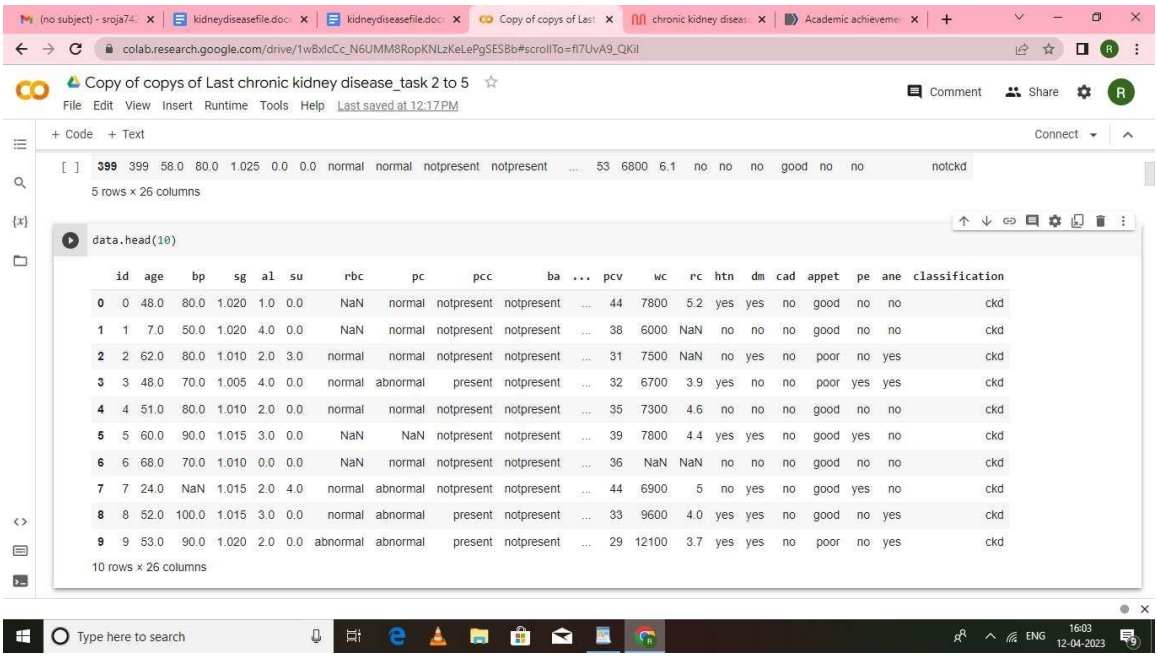
Ideation and Brainstorm Map :

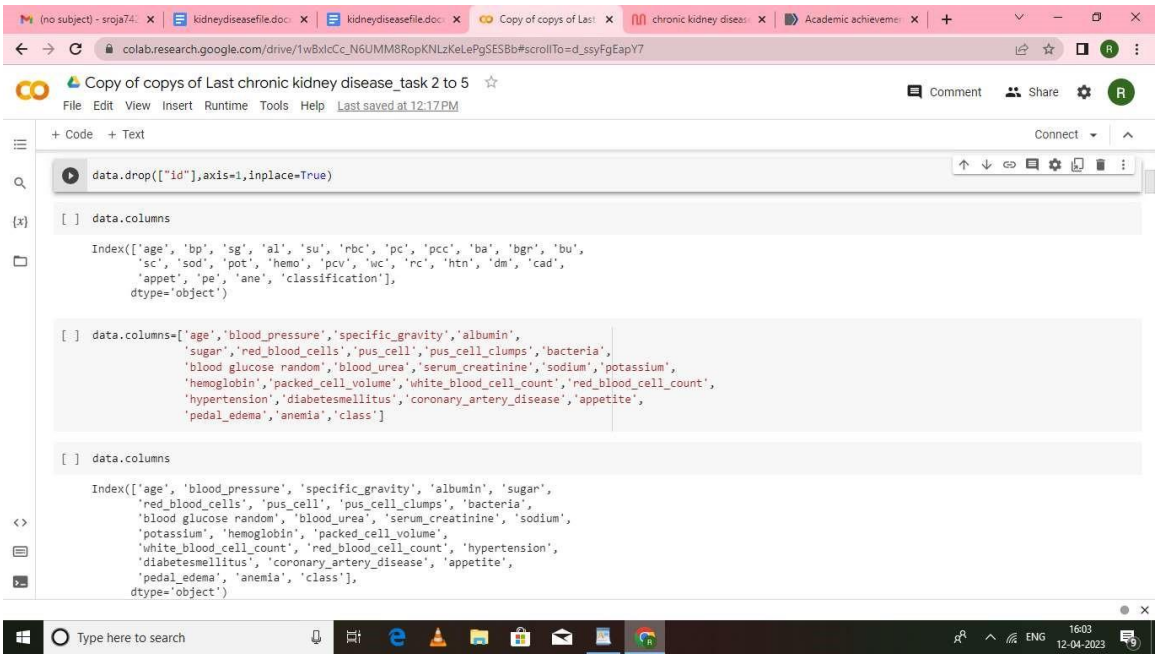

app.mural.co/t/rojateam1448/m/rojateam1448/1679046833830/06913fa19c6e483be08e0e716d1c42e2a542eff3?sender=u08dade71960dd858d6a32584
You have 1 mural remaining. Upgrade your free plan for unlimited murals.
Upgrade
chronic kidney disease prediction
Share
Brainstorm
& idea prioritization
Type here to search
ENG
16:01
12-04-2023

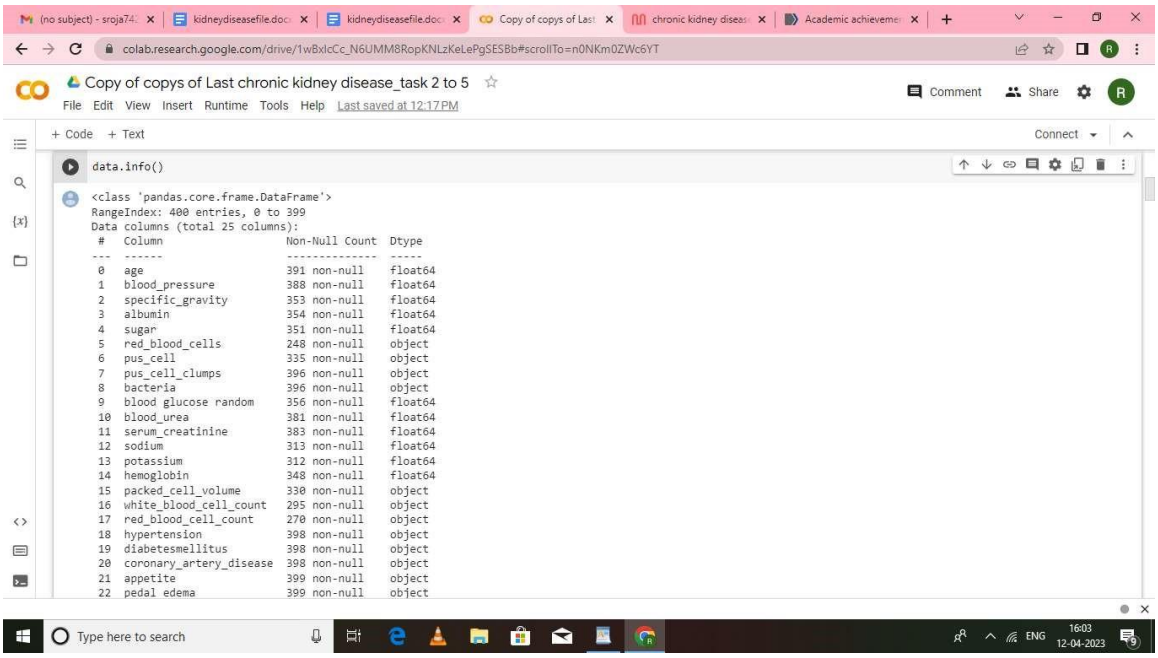
Result :

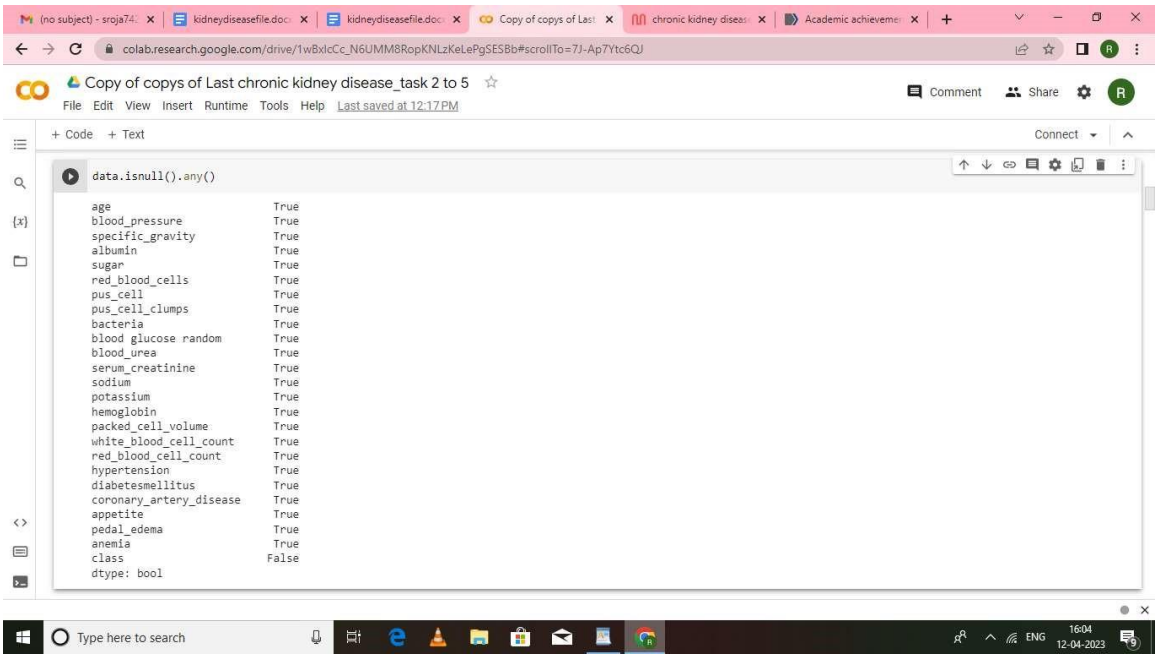

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
import pandas as pd
import numpy as np
from collections import Counter as c
import matplotlib.pyplot as plt
import seaborn as sns
import missingno 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
[ ] from google.colab import files
uploaded = files.upload()
Choose Files 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.
Saving kidney_disease.csv to kidney_disease.csv
[ ] data=pd.read_csv("kidney_disease.csv")
data.head()
id age bp sg al su rbc pc pcc ba ... pcv wc rc htn dm cad appet pe ane classification
0 0 48.0 80.0 1.020 1.0 0.0 NaN normal notpresent notpresent ... 44 7800 5.2 yes yes no good no no ckd
Type here to search
ENG
16:02
12-04-2023

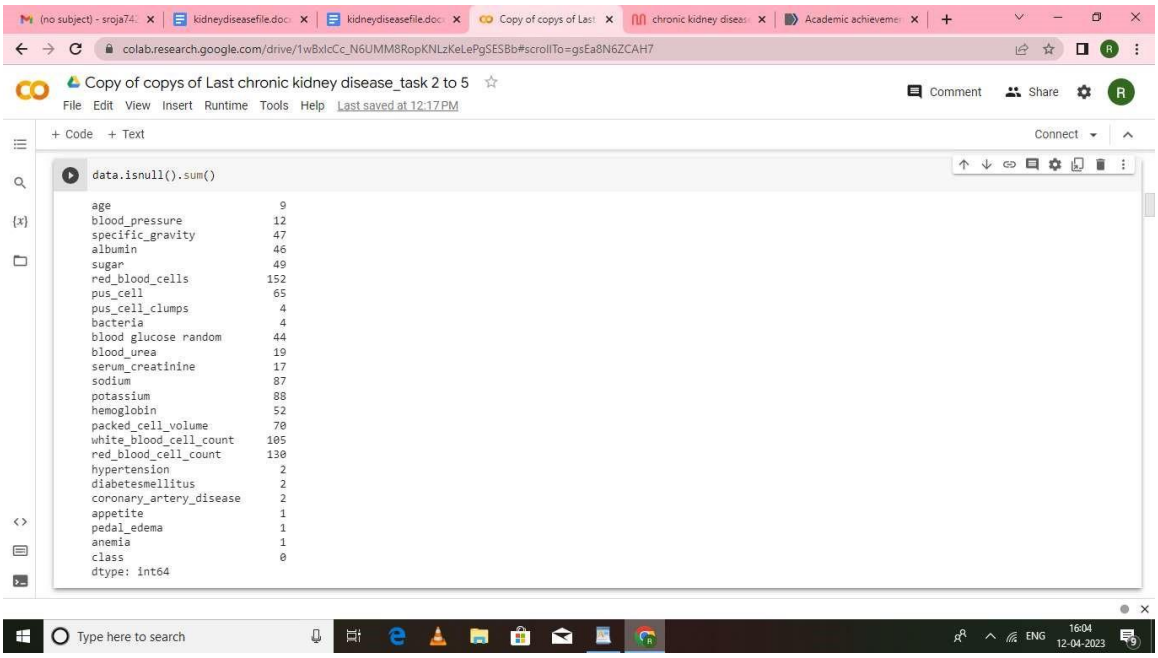

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data=pd.read_csv("kidney_disease.csv")
data.head()
id age bp sg al su rbc pc pcc ba ... pcv wc rc htn dm cad appet pe ane classification
0 0 48.0 80.0 1.020 1.0 0.0 NaN normal notpresent notpresent ... 44 7800 5.2 yes yes no good no no ckd
1 1 7.0 50.0 1.020 4.0 0.0 NaN normal notpresent notpresent ... 38 6000 NaN no no no good no no ckd
2 2 62.0 80.0 1.010 2.0 3.0 normal normal notpresent notpresent ... 31 7500 NaN no yes no poor no yes ckd
3 3 48.0 70.0 1.005 4.0 0.0 normal abnormal present notpresent ... 32 6700 3.9 yes no no poor yes yes ckd
4 4 51.0 80.0 1.010 2.0 0.0 normal normal notpresent notpresent ... 35 7300 4.6 no no no good no no ckd
5 rows × 26 columns
data.tail()
id age bp sg al su rbc pc pcc ba ... pcv wc rc htn dm cad appet pe ane classification
395 395 55.0 80.0 1.020 0.0 0.0 normal normal notpresent notpresent ... 47 6700 4.9 no no no good no no notckd
396 396 42.0 70.0 1.025 0.0 0.0 normal normal notpresent notpresent ... 54 7800 6.2 no no no good no no notckd
397 397 12.0 80.0 1.020 0.0 0.0 normal normal notpresent notpresent ... 49 6600 5.4 no no no good no no notckd
398 398 17.0 60.0 1.025 0.0 0.0 normal normal notpresent notpresent ... 51 7200 5.9 no no no good no no notckd
399 399 58.0 80.0 1.025 0.0 0.0 normal normal notpresent notpresent ... 53 6800 6.1 no no no good no no notckd
Type here to search
ENG
16:02
12-04-2023

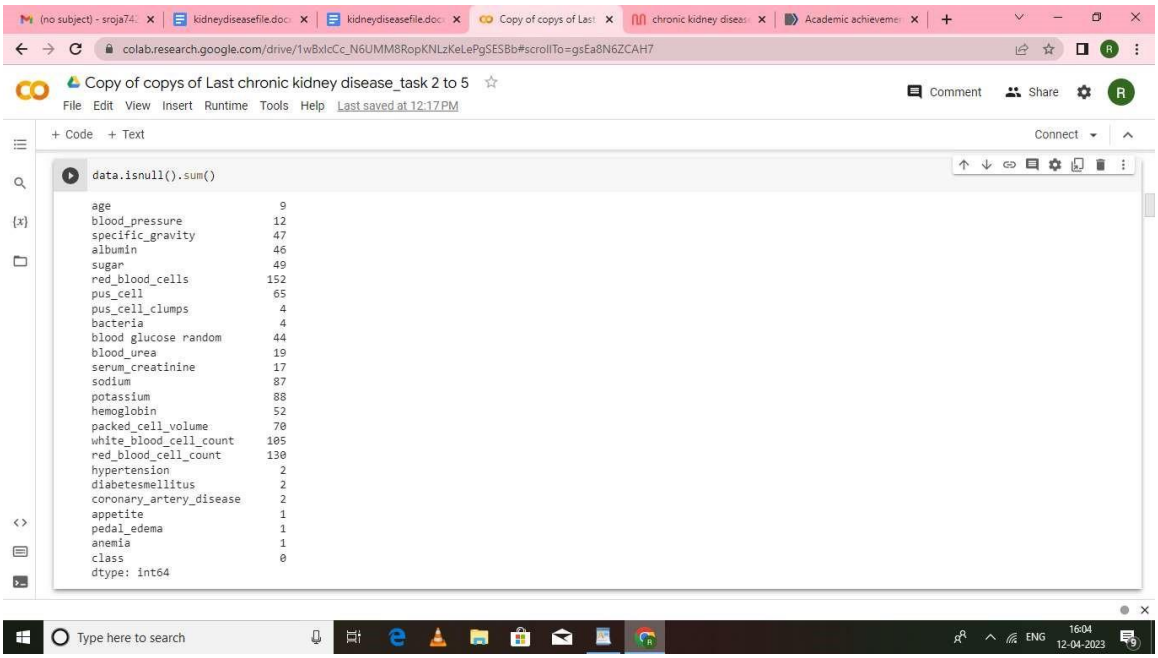

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
[ ] 399 399 58.0 80.0 1.025 0.0 0.0 normal normal notpresent notpresent ... 53 6800 6.1 no no no good no no notckd
5 rows × 26 columns
data.head(10)
id age bp sg al su rbc pc pcc ba ... pcv wc rc htn dm cad appet pe ane classification
0 0 48.0 80.0 1.020 1.0 0.0 NaN normal notpresent notpresent ... 44 7800 5.2 yes yes no good no no ckd
1 1 7.0 50.0 1.020 4.0 0.0 NaN normal notpresent notpresent ... 38 6000 NaN no no no good no no ckd
2 2 62.0 80.0 1.010 2.0 3.0 normal normal notpresent notpresent ... 31 7500 NaN no yes no poor no yes ckd
3 3 48.0 70.0 1.005 4.0 0.0 normal abnormal present notpresent ... 32 6700 3.9 yes no no poor yes yes ckd
4 4 51.0 80.0 1.010 2.0 0.0 normal normal notpresent notpresent ... 35 7300 4.6 no no no good no no ckd
5 5 60.0 90.0 1.015 3.0 0.0 NaN NaN notpresent notpresent ... 39 7800 4.4 yes yes no good yes no ckd
6 6 68.0 70.0 1.010 0.0 0.0 NaN normal notpresent notpresent ... 36 NaN NaN no no no good no no ckd
7 7 24.0 NaN 1.015 2.0 4.0 normal abnormal notpresent notpresent ... 44 6900 5 no yes no good yes no ckd
8 8 52.0 100.0 1.015 3.0 0.0 normal abnormal present notpresent ... 33 9600 4.0 yes yes no good no yes ckd
9 9 53.0 90.0 1.020 2.0 0.0 abnormal abnormal present notpresent ... 29 12100 3.7 yes yes no poor no yes ckd
10 rows × 26 columns
Type here to search
ENG
16:03
12-04-2023

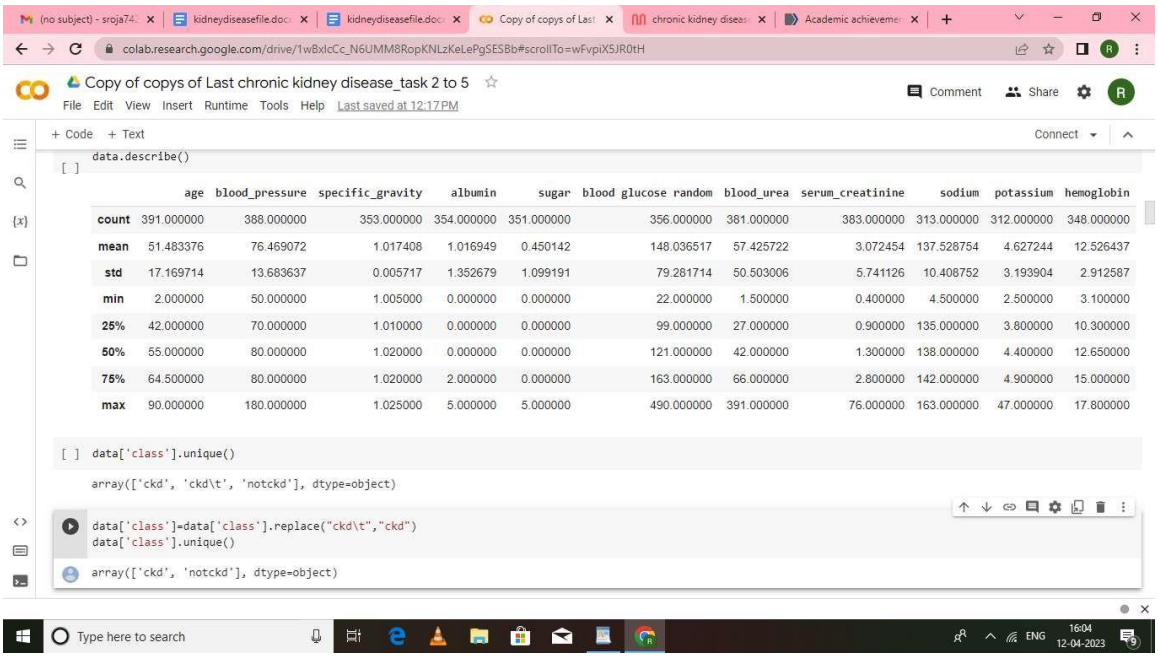

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data.drop(["id"],axis=1,inplace=True)
[ ] data.columns
Index(['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
'appet', 'pe', 'ane', 'classification'],
dtype='object')
[ ] data.columns=['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(['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')
Type here to search
ENG
16:03
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
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 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 diabetesmellitus 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
Type here to search
ENG
16:03
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data.isnull().any()
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
Type here to search
ENG
16:04
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data.isnull().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
hemoglobin 52
packed_cell_volume 70
white_blood_cell_count 105
red_blood_cell_count 130
hypertension 2
diabetesmellitus 2
coronary_artery_disease 2
appetite 1
pedal_edema 1
anemia 1
class 0
dtype: int64
Type here to search
ENG
16:04
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data.isnull().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
hemoglobin 52
packed_cell_volume 70
white_blood_cell_count 105
red_blood_cell_count 130
hypertension 2
diabetesmellitus 2
coronary_artery_disease 2
appetite 1
pedal_edema 1
anemia 1
class 0
dtype: int64
Type here to search
ENG
16:04
12-04-2023



Copy of copys of Last chronic kidney disease_task 2 to 5

File Edit View Insert Runtime Tools Help Last saved at 12:17 PM

Comment Share

+ Code + Text

Connect

```
data.describe()
```

	age	blood_pressure	specific_gravity	albumin	sugar	blood glucose random	blood_urea	serum_creatinine	sodium	potassium	hemoglobin
count	391.000000	388.000000	353.000000	354.000000	351.000000	356.000000	381.000000	383.000000	313.000000	312.000000	348.000000
mean	51.483376	76.469072	1.017408	1.016949	0.450142	148.036517	57.425722	3.072454	137.528754	4.627244	12.526437
std	17.169714	13.683637	0.005717	1.352679	1.099191	79.281714	50.503006	5.741126	10.408752	3.193904	2.912587
min	2.000000	50.000000	1.005000	0.000000	0.000000	22.000000	1.500000	0.400000	4.500000	2.500000	3.100000
25%	42.000000	70.000000	1.010000	0.000000	0.000000	99.000000	27.000000	0.900000	135.000000	3.800000	10.300000
50%	55.000000	80.000000	1.020000	0.000000	0.000000	121.000000	42.000000	1.300000	138.000000	4.400000	12.650000
75%	64.500000	80.000000	1.020000	2.000000	0.000000	163.000000	66.000000	2.800000	142.000000	4.900000	15.000000
max	90.000000	180.000000	1.025000	5.000000	5.000000	490.000000	391.000000	76.000000	163.000000	47.000000	17.800000

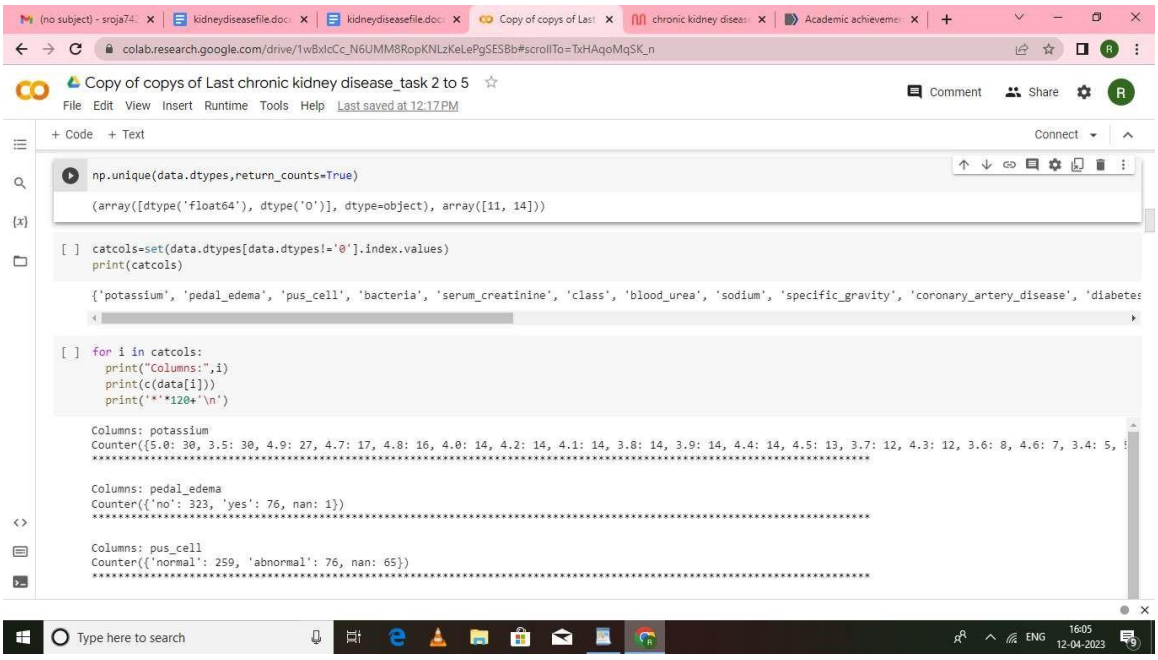
```
data['class'].unique()
```

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

```
data['class']=data['class'].replace("ckd\t","ckd")
data['class'].unique()
```

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

Type here to search ENG 16:04 12-04-2023



Copy of copys of Last chronic kidney disease_task 2 to 5

File Edit View Insert Runtime Tools Help Last saved at 12:17 PM

Comment Share

+ Code + Text

Connect

```
np.unique(data.dtypes,return_counts=True)
```

```
(array([dtype('float64'), dtype('O')], dtype=object), array([11, 14]))
```

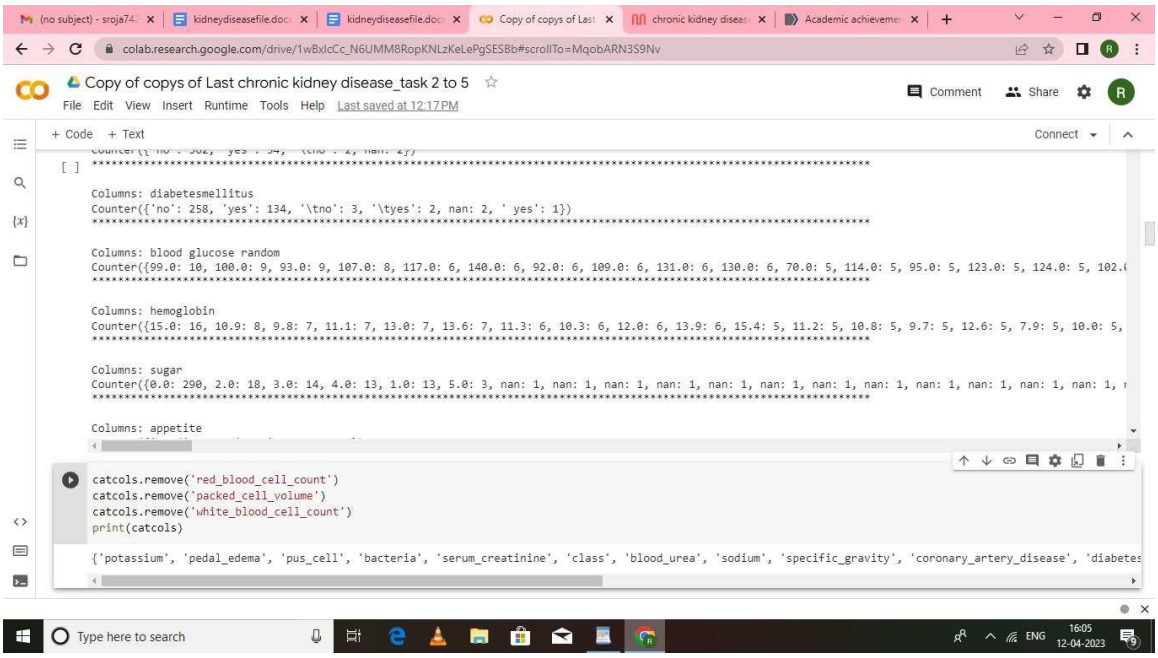
```
catcols=set(data.dtypes[data.dtypes!='0'].index.values)
print(catcols)
```

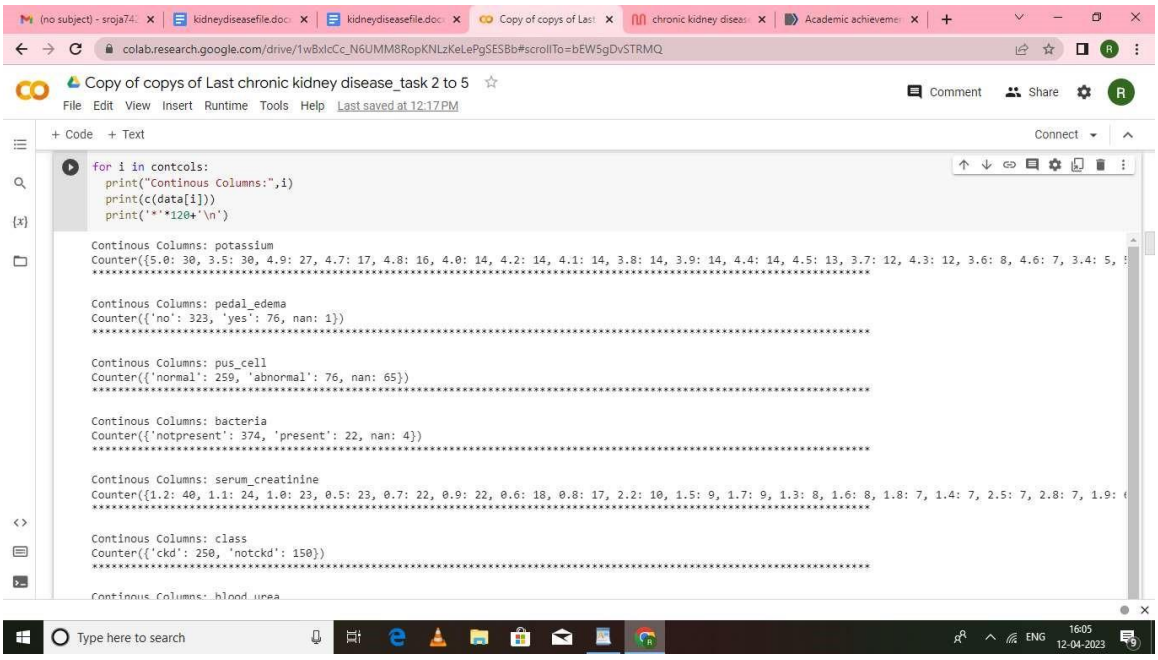
```
{'potassium', 'pedal_edema', 'pus_cell', 'bacteria', 'serum_creatinine', 'class', 'blood_urea', 'sodium', 'specific_gravity', 'coronary_artery_disease', 'diabetes
```

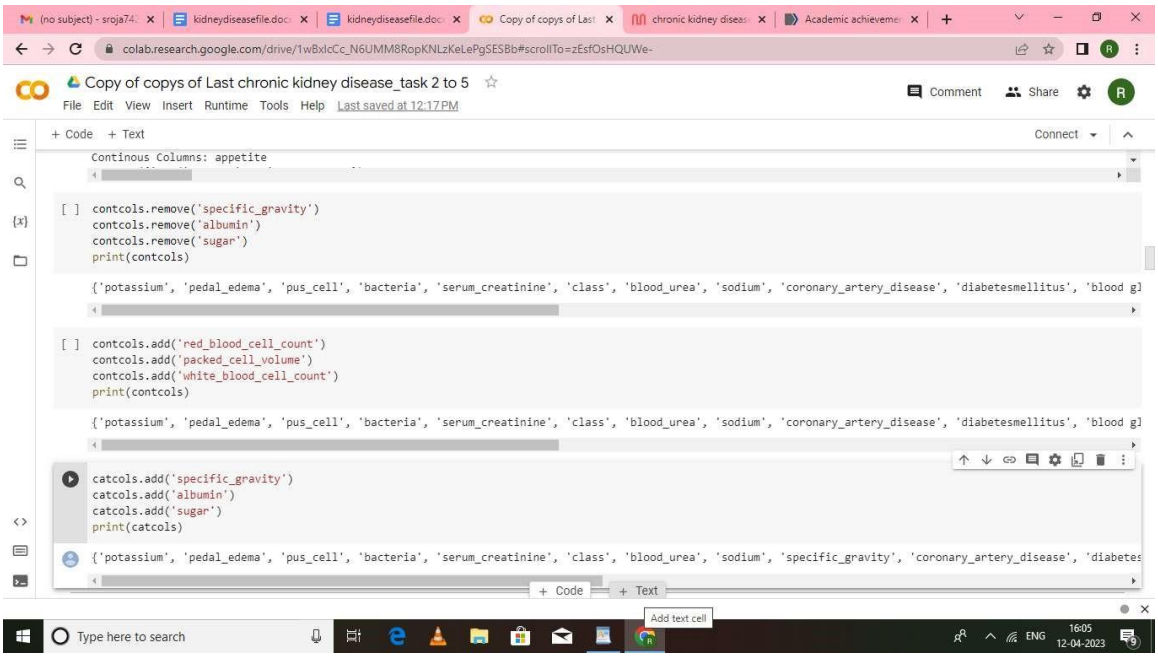
```
for i in catcols:
  print("Columns:",i)
  print(c(data[i]))
  print('*'*120+'\n')
```

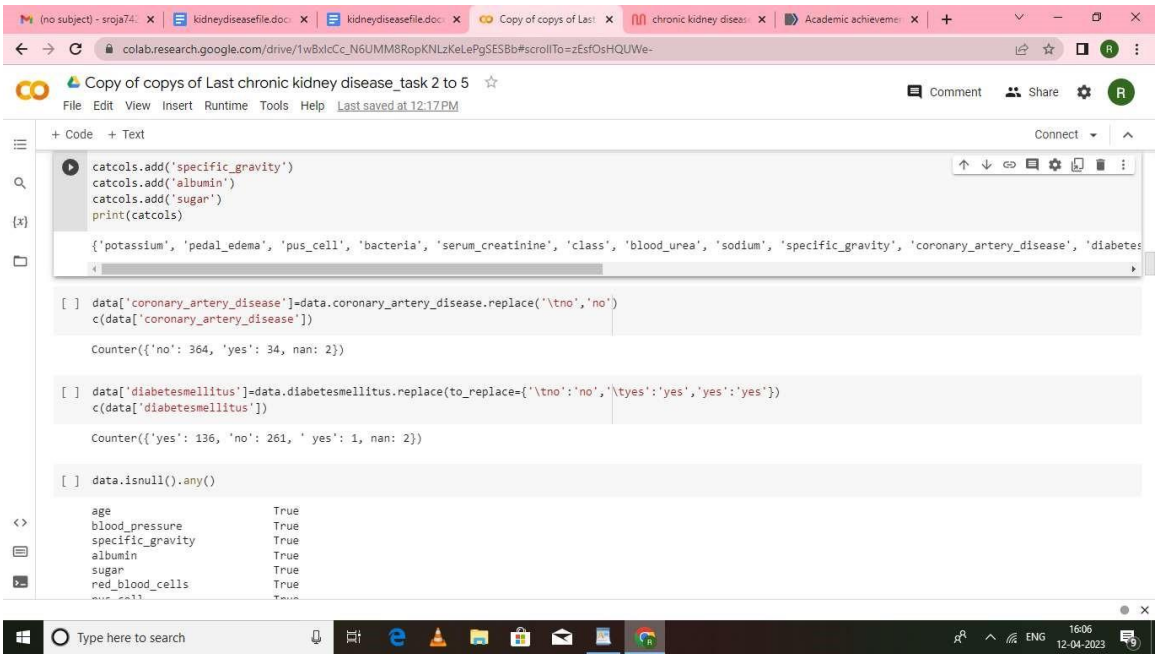
```
Columns: potassium
Counter({5.0: 30, 3.5: 30, 4.9: 27, 4.7: 17, 4.8: 16, 4.0: 14, 4.2: 14, 4.1: 14, 3.8: 14, 3.9: 14, 4.4: 14, 4.5: 13, 3.7: 12, 4.3: 12, 3.6: 8, 4.6: 7, 3.4: 5, 
************************************************************************************************************************

Columns: pedal_edema
Counter({'no': 323, 'yes': 76, nan: 1})
************************************************************************************************************************

Columns: pus_cell
Counter({'normal': 259, 'abnormal': 76, nan: 65})
************************************************************************************************************************
```

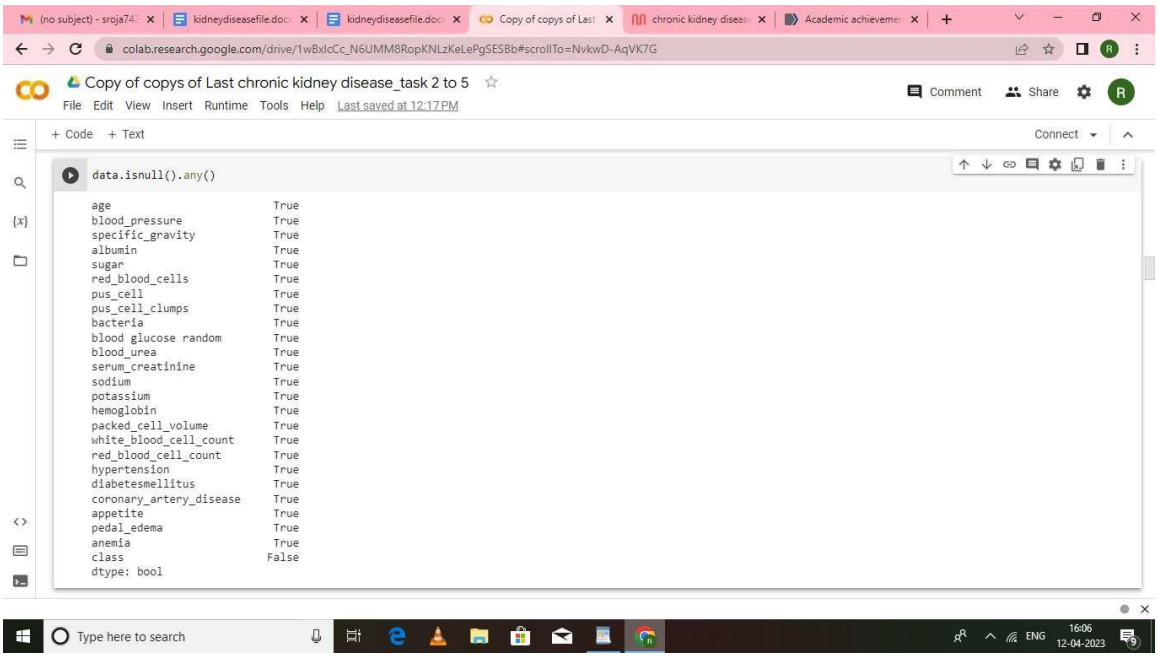
Type here to search ENG 16:05 12-04-2023

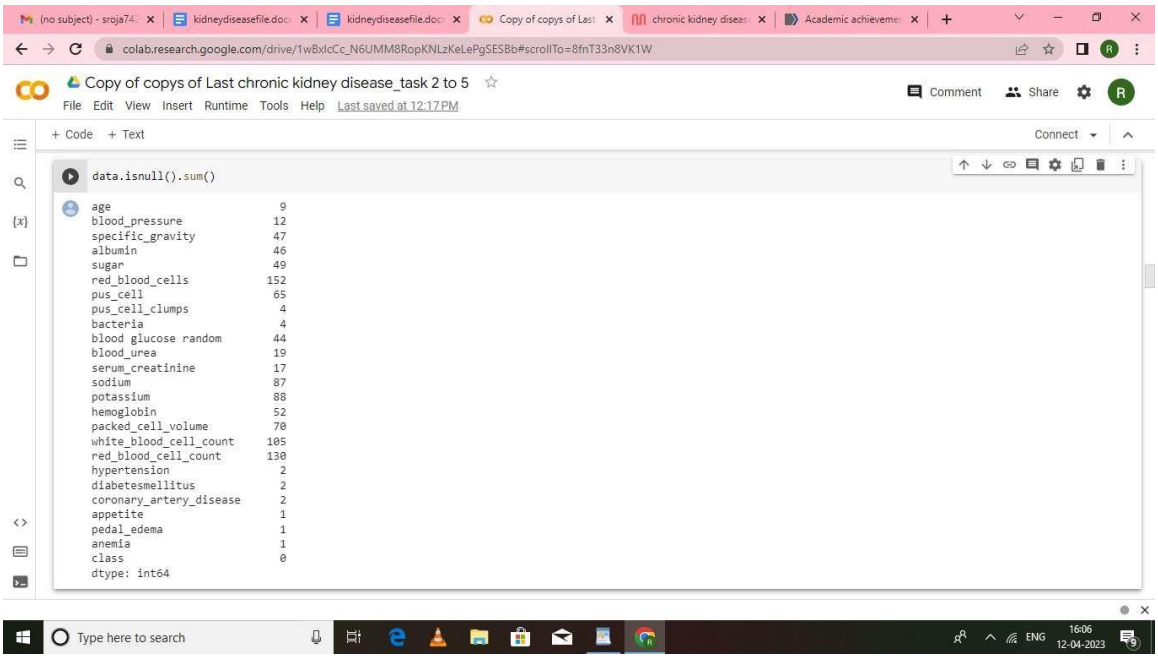

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
Columns: diabetesmellitus
Counter({'no': 258, 'yes': 134, '\tno': 3, '\tyes': 2, nan: 2, ' yes': 1})
Columns: blood glucose random
Columns: hemoglobin
Columns: sugar
Columns: appetite
catcols.remove('red_blood_cell_count')
catcols.remove('packed_cell_volume')
catcols.remove('white_blood_cell_count')
print(catcols)
{'potassium', 'pedal_edema', 'pus_cell', 'bacteria', 'serum_creatinine', 'class', 'blood_urea', 'sodium', 'specific_gravity', 'coronary_artery_disease', 'diabetes
Type here to search
ENG
16:05
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
for i in contcols:
print("Continous Columns:",i)
print(c(data[i]))
print('*'*120+'\n')
Continous Columns: potassium
Continous Columns: pedal_edema
Counter({'no': 323, 'yes': 76, nan: 1})
Continous Columns: pus_cell
Counter({'normal': 259, 'abnormal': 76, nan: 65})
Continous Columns: bacteria
Counter({'notpresent': 374, 'present': 22, nan: 4})
Continous Columns: serum_creatinine
Continous Columns: class
Counter({'ckd': 250, 'notckd': 150})
Type here to search
ENG
16:05
12-04-2023

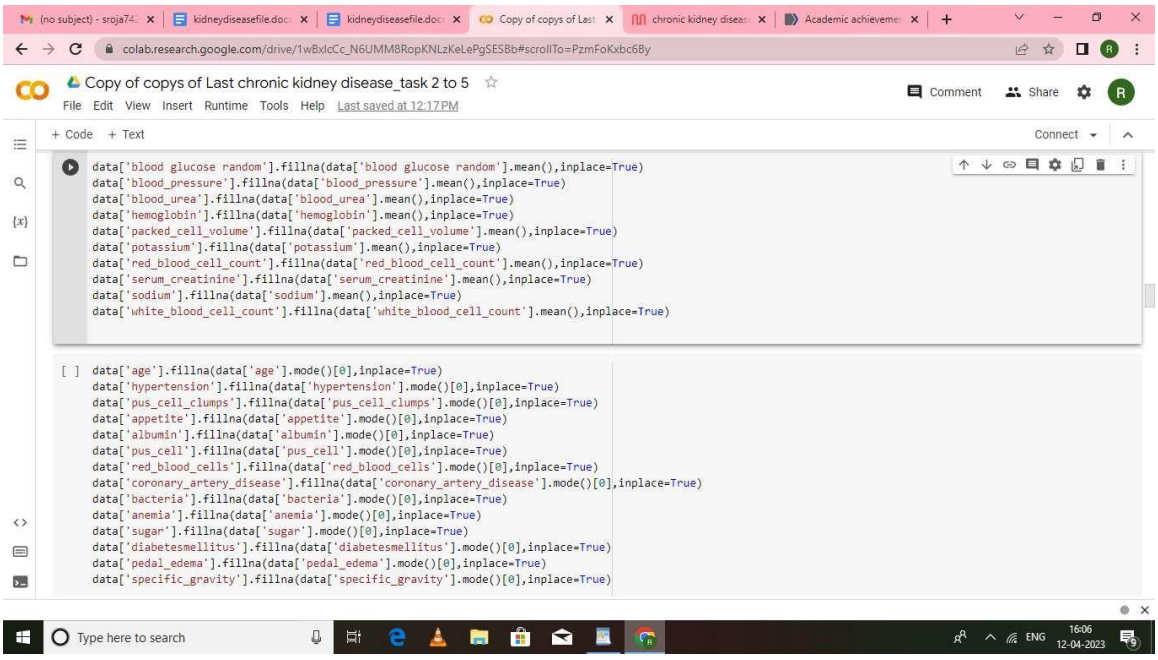

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
Continous Columns: appetite
[ ] contcols.remove('specific_gravity')
contcols.remove('albumin')
contcols.remove('sugar')
print(contcols)
{'potassium', 'pedal_edema', 'pus_cell', 'bacteria', 'serum_creatinine', 'class', 'blood_urea', 'sodium', 'coronary_artery_disease', 'diabetesmellitus', 'blood gl
[ ] contcols.add('red_blood_cell_count')
contcols.add('packed_cell_volume')
contcols.add('white_blood_cell_count')
print(contcols)
{'potassium', 'pedal_edema', 'pus_cell', 'bacteria', 'serum_creatinine', 'class', 'blood_urea', 'sodium', 'coronary_artery_disease', 'diabetesmellitus', 'blood gl
catcols.add('specific_gravity')
catcols.add('albumin')
catcols.add('sugar')
print(catcols)
{'potassium', 'pedal_edema', 'pus_cell', 'bacteria', 'serum_creatinine', 'class', 'blood_urea', 'sodium', 'specific_gravity', 'coronary_artery_disease', 'diabetes
+ Code + Text
Add text cell
Type here to search
ENG
16:05
12-04-2023

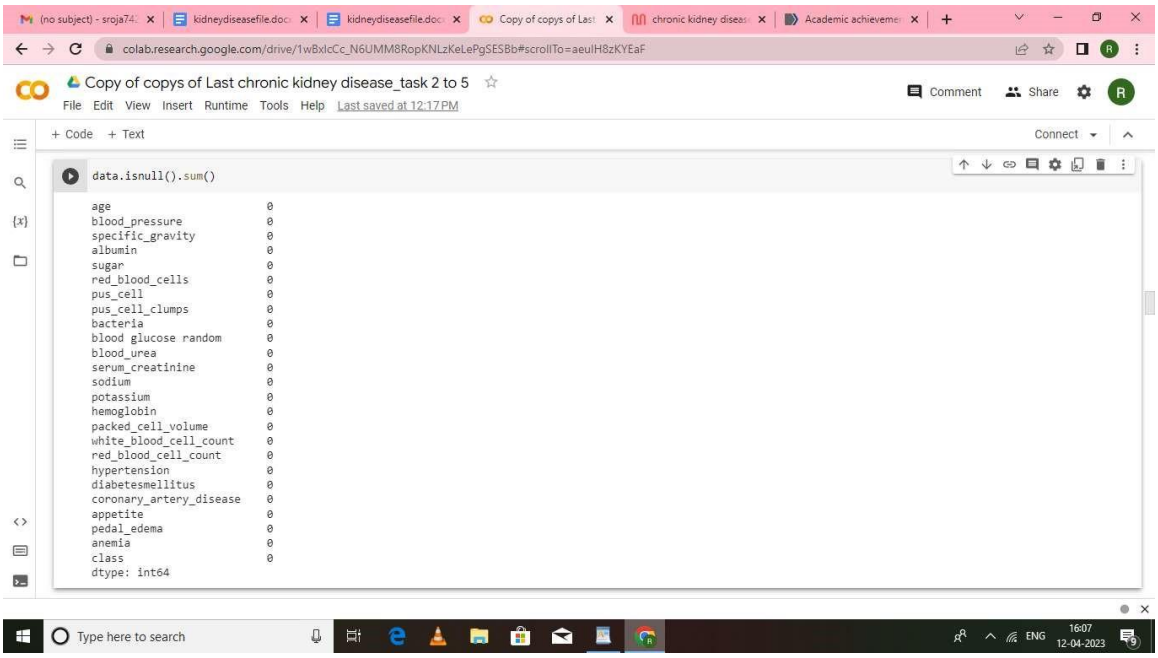

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
catcols.add('specific_gravity')
catcols.add('albumin')
catcols.add('sugar')
print(catcols)
{'potassium', 'pedal_edema', 'pus_cell', 'bacteria', 'serum_creatinine', 'class', 'blood_urea', 'sodium', 'specific_gravity', 'coronary_artery_disease', 'diabetes
[ ] data['coronary_artery_disease']=data.coronary_artery_disease.replace('\tno','no')
c(data['coronary_artery_disease'])
Counter({'no': 364, 'yes': 34, nan: 2})
[ ] data['diabetesmellitus']=data.diabetesmellitus.replace(to_replace={'\tno':'no','\tyes':'yes','yes':'yes'})
c(data['diabetesmellitus'])
Counter({'yes': 136, 'no': 261, ' yes': 1, nan: 2})
[ ] data.isnull().any()
age True
blood_pressure True
specific_gravity True
albumin True
sugar True
red_blood_cells True
Type here to search
ENG
16:06
12-04-2023

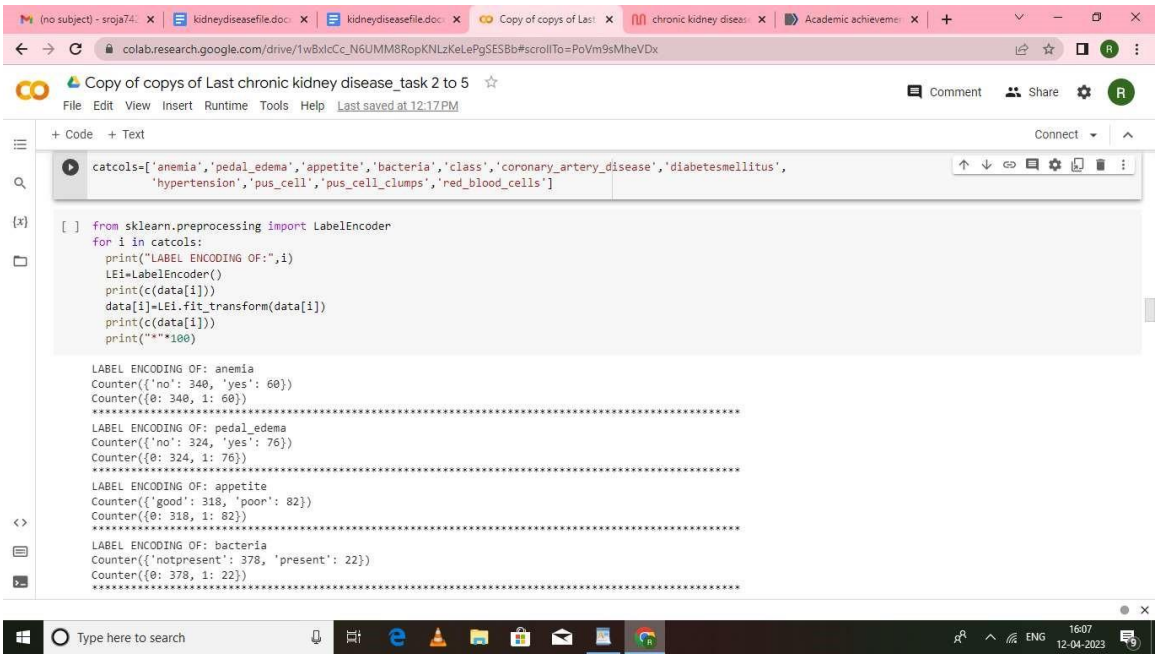

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data.isnull().any()
age True
blood_pressure True
specific_gravity True
albumin True
sugar True
red_blood_cells True
pus_cell True
pus_cell_clumps 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
Type here to search
ENG
16:06
12-04-2023

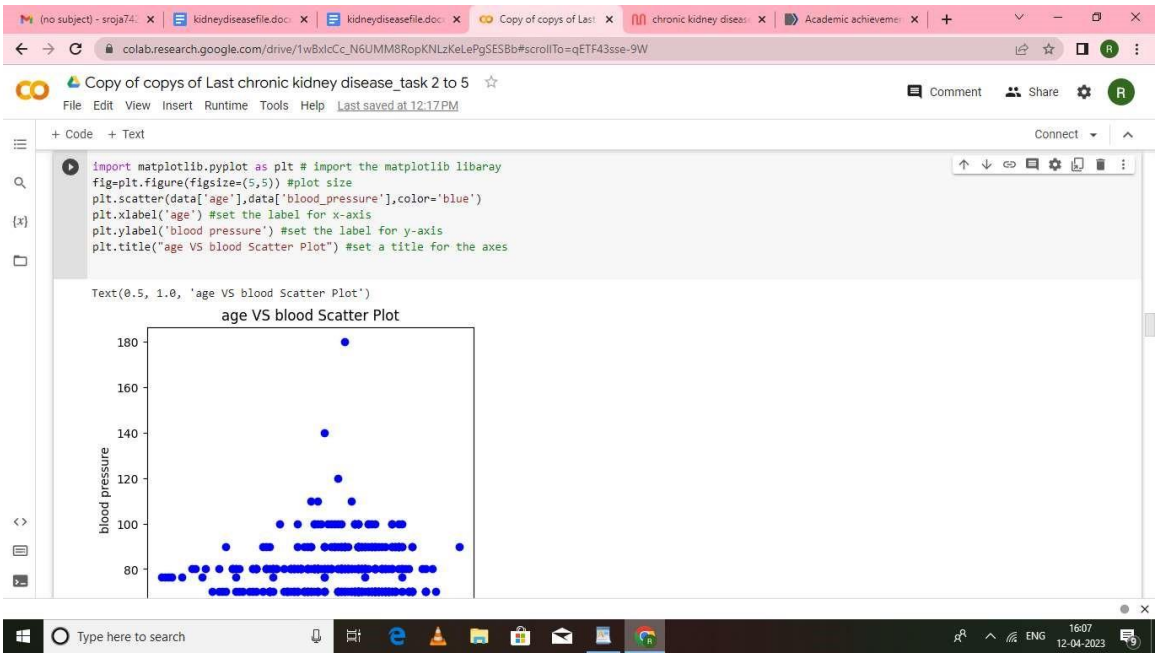

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data.isnull().sum()
age 9
blood_pressure 12
specific_gravity 47
albumin 46
sugar 49
red_blood_cells 152
pus_cell 65
pus_cell_clumps 4
bacteria 4
blood glucose random 44
blood_urea 19
serum_creatinine 17
sodium 87
potassium 88
hemoglobin 52
packed_cell_volume 70
white_blood_cell_count 105
red_blood_cell_count 130
hypertension 2
diabetesmellitus 2
coronary_artery_disease 2
appetite 1
pedal_edema 1
anemia 1
class 0
dtype: int64
Type here to search
ENG
16:06
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
msno.bar(data)
plt.show()
391 388 353 354 351 248 335 396 396 356 381 383 313 312 348 330 295 270 398 398 398 399 399 399 400
1.0
0.8
0.6
0.4
0.2
400
320
240
160
80
Type here to search
ENG
16:06
12-04-2023

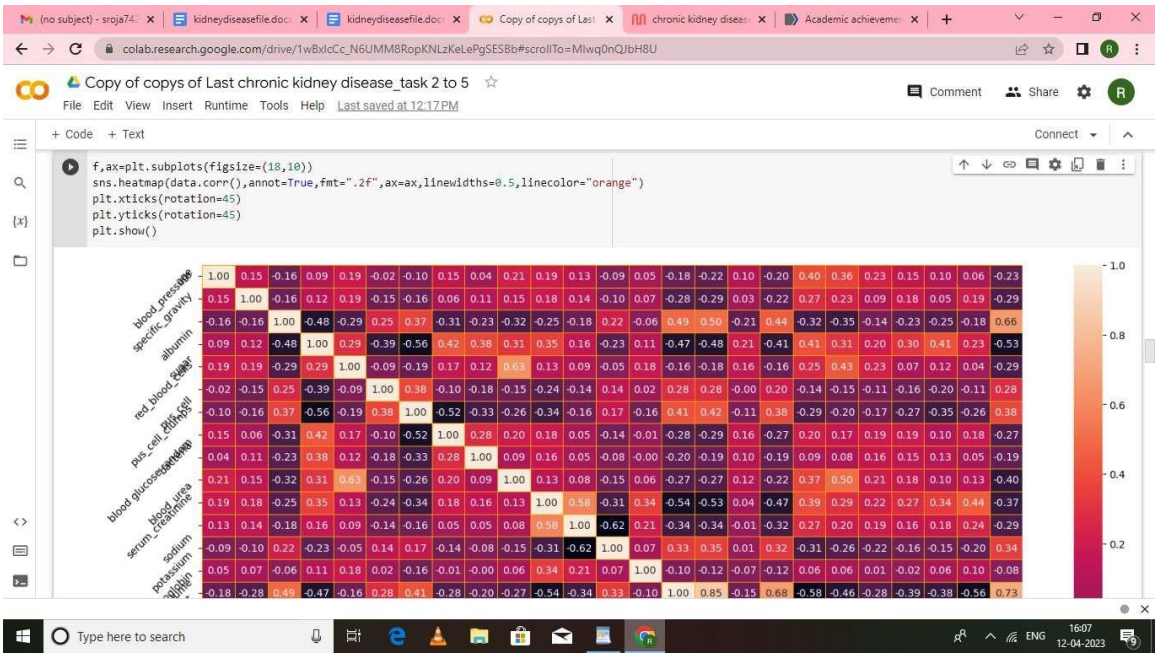

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data['blood glucose random'].fillna(data['blood glucose random'].mean(),inplace=True)
data['blood_pressure'].fillna(data['blood_pressure'].mean(),inplace=True)
data['blood_urea'].fillna(data['blood_urea'].mean(),inplace=True)
data['hemoglobin'].fillna(data['hemoglobin'].mean(),inplace=True)
data['packed_cell_volume'].fillna(data['packed_cell_volume'].mean(),inplace=True)
data['potassium'].fillna(data['potassium'].mean(),inplace=True)
data['red_blood_cell_count'].fillna(data['red_blood_cell_count'].mean(),inplace=True)
data['serum_creatinine'].fillna(data['serum_creatinine'].mean(),inplace=True)
data['sodium'].fillna(data['sodium'].mean(),inplace=True)
data['white_blood_cell_count'].fillna(data['white_blood_cell_count'].mean(),inplace=True)
[ ] data['age'].fillna(data['age'].mode()[0],inplace=True)
data['hypertension'].fillna(data['hypertension'].mode()[0],inplace=True)
data['pus_cell_clumps'].fillna(data['pus_cell_clumps'].mode()[0],inplace=True)
data['appetite'].fillna(data['appetite'].mode()[0],inplace=True)
data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)
data['pus_cell'].fillna(data['pus_cell'].mode()[0],inplace=True)
data['red_blood_cells'].fillna(data['red_blood_cells'].mode()[0],inplace=True)
data['coronary_artery_disease'].fillna(data['coronary_artery_disease'].mode()[0],inplace=True)
data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)
data['anemia'].fillna(data['anemia'].mode()[0],inplace=True)
data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)
data['diabetesmellitus'].fillna(data['diabetesmellitus'].mode()[0],inplace=True)
data['pedal_edema'].fillna(data['pedal_edema'].mode()[0],inplace=True)
data['specific_gravity'].fillna(data['specific_gravity'].mode()[0],inplace=True)
Type here to search
ENG
16:06
12-04-2023

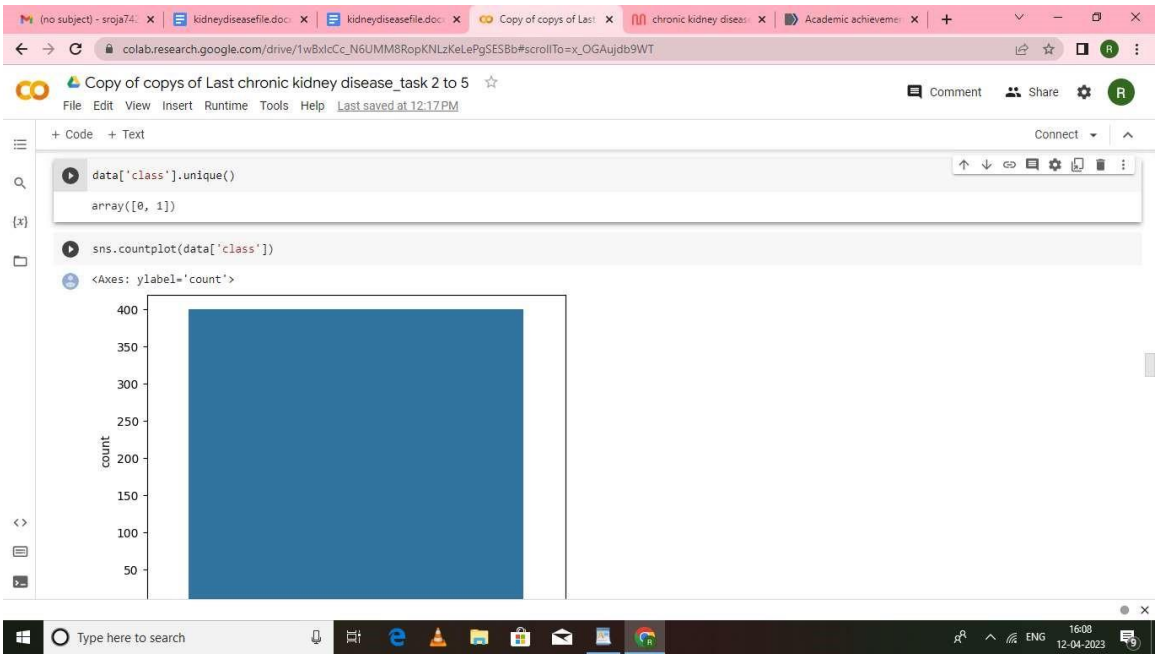

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data.isnull().sum()
age 0
blood_pressure 0
specific_gravity 0
albumin 0
sugar 0
red_blood_cells 0
pus_cell 0
pus_cell_clumps 0
bacteria 0
blood glucose random 0
blood_urea 0
serum_creatinine 0
sodium 0
potassium 0
hemoglobin 0
packed_cell_volume 0
white_blood_cell_count 0
red_blood_cell_count 0
hypertension 0
diabetesmellitus 0
coronary_artery_disease 0
appetite 0
pedal_edema 0
anemia 0
class 0
dtype: int64

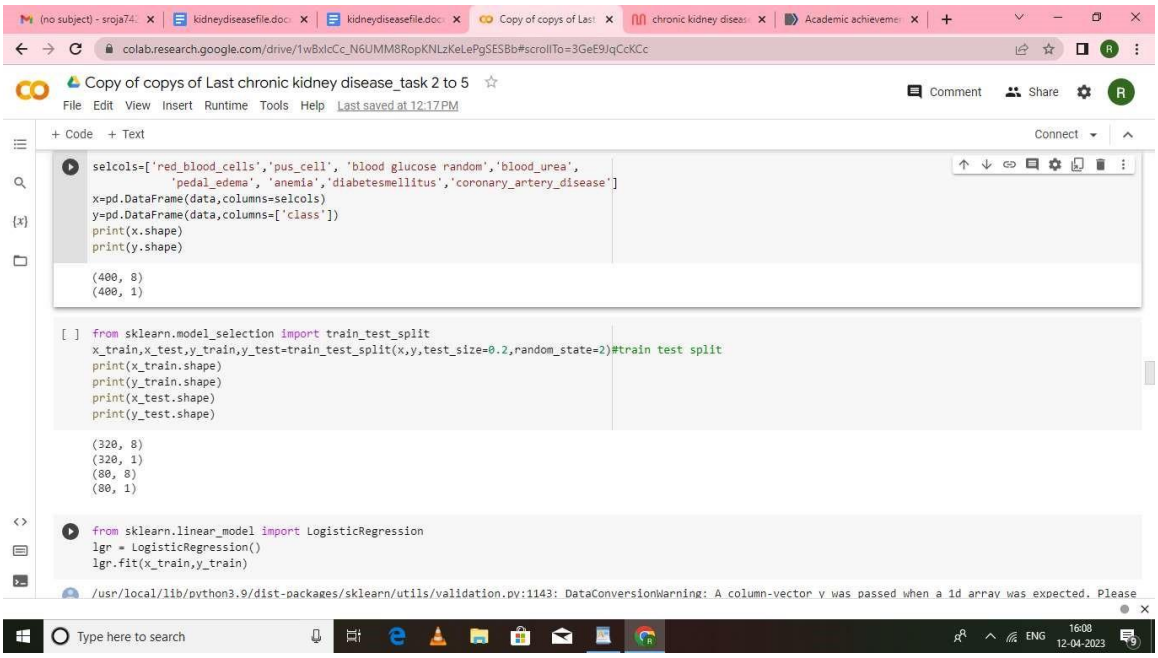

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
catcols=['anemia','pedal_edema','appetite','bacteria','class','coronary_artery_disease','diabetesmellitus',
'hypertension','pus_cell','pus_cell_clumps','red_blood_cells']
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)
LABEL ENCODING OF: anemia
Counter({'no': 340, 'yes': 60})
Counter({0: 340, 1: 60})
LABEL ENCODING OF: pedal_edema
Counter({'no': 324, 'yes': 76})
Counter({0: 324, 1: 76})
LABEL ENCODING OF: appetite
Counter({'good': 318, 'poor': 82})
Counter({0: 318, 1: 82})
LABEL ENCODING OF: bacteria
Counter({'notpresent': 378, 'present': 22})
Counter({0: 378, 1: 22})

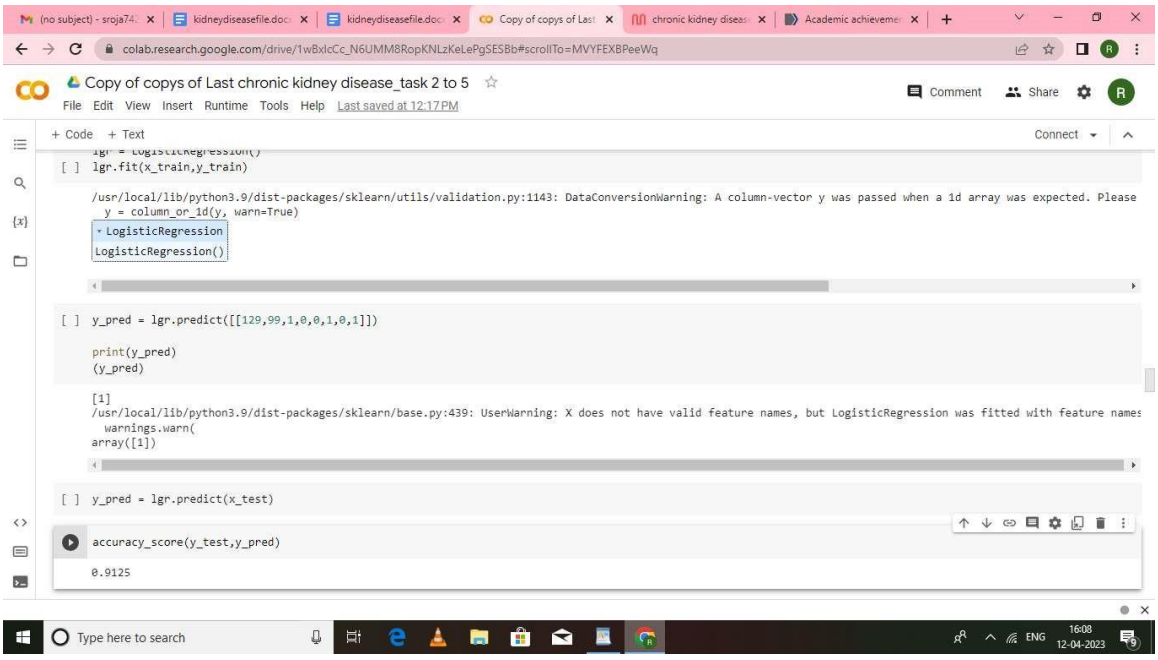

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
import matplotlib.pyplot as plt # import the matplotlib libaray
fig=plt.figure(figsize=(5,5)) #plot size
plt.scatter(data['age'],data['blood_pressure'],color='blue')
plt.xlabel('age') #set the label for x-axis
plt.ylabel('blood pressure') #set the label for y-axis
plt.title("age VS blood Scatter Plot") #set a title for the axes
Text(0.5, 1.0, 'age VS blood Scatter Plot')
age VS blood Scatter Plot
blood pressure
180
160
140
120
100
80
Type here to search
ENG
16:07
12-04-2023

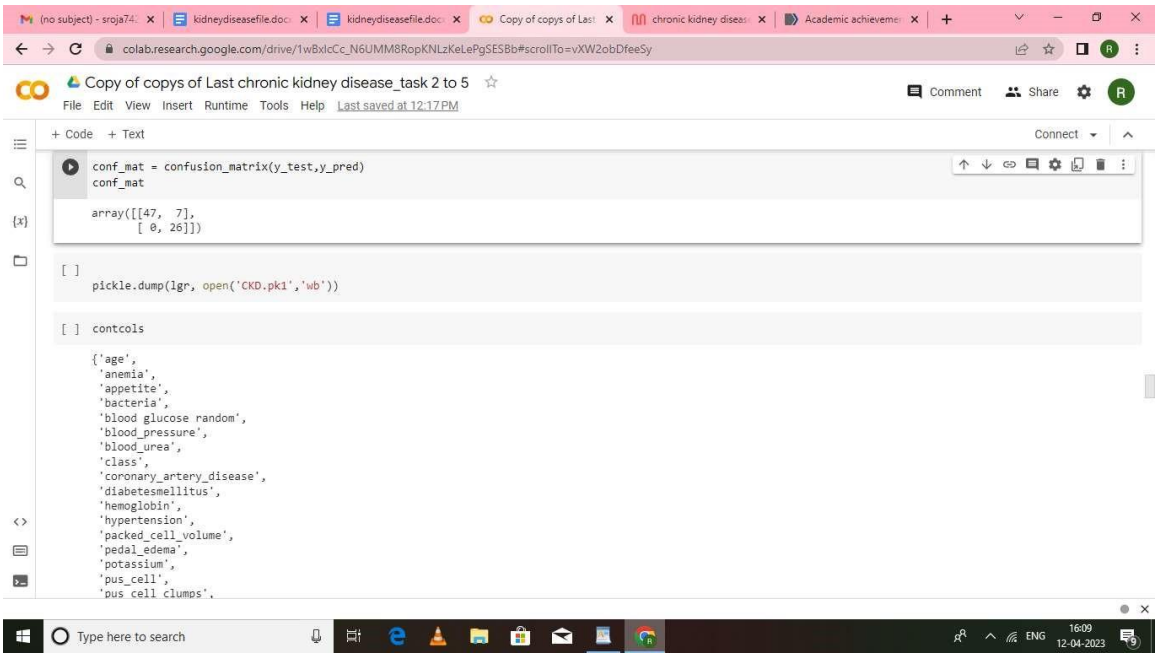

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
plt.figure(figsize=(20,15), facecolor='white')
plotnumber = 1
for column in contcols:
if plotnumber<=11 :
ax = plt.subplot(3,4,plotnumber)
plt.scatter(data['age'],data[column])
plt.xlabel(column,fontsize=20)
plotnumber+=1
plt.show()
potassium
pedal_edema
pus_cell
bacteria
Type here to search
ENG
16:07
12-04-2023

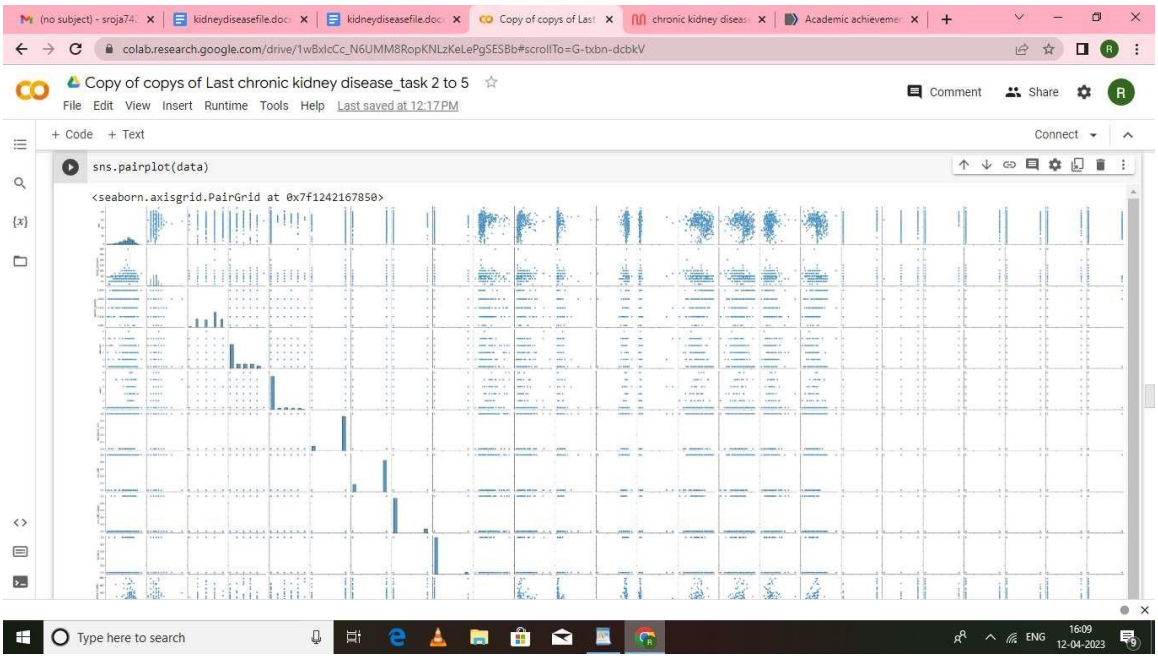

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
f,ax=plt.subplots(figsize=(18,10))
sns.heatmap(data.corr(),annot=True,fmt=".2f",ax=ax,linewidths=0.5,linecolor="orange")
plt.xticks(rotation=45)
plt.yticks(rotation=45)
plt.show()
blood_pressure
specific_gravity
albumin
red_blood_cells
pus_cell
pus_cell_clumps
blood_glucose_random
blood_urea
serum_creatinine
sodium
potassium
1.0
0.8
0.6
0.4
0.2
Type here to search
ENG
16:07
12-04-2023

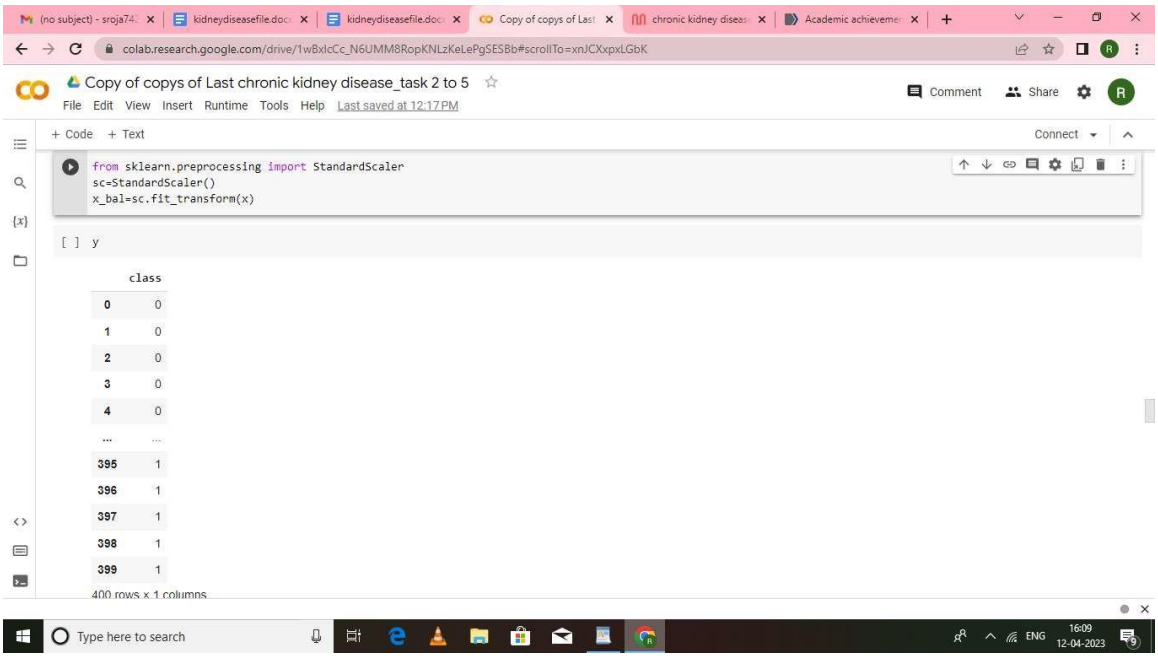

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
data['class'].unique()
array([0, 1])
sns.countplot(data['class'])
<Axes: ylabel='count'>
count
400
350
300
250
200
150
100
50
Type here to search
ENG
16:08
12-04-2023

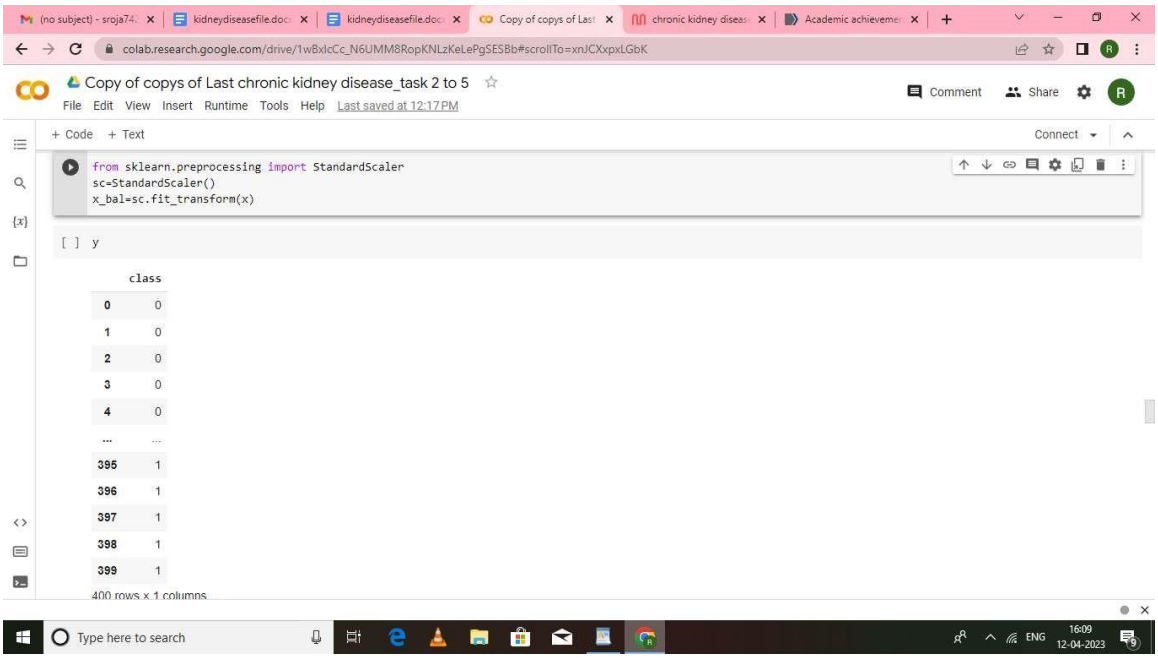

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
selcols=['red_blood_cells','pus_cell', 'blood glucose random','blood_urea',
'pedal_edema', 'anemia','diabetesmellitus','coronary_artery_disease']
x=pd.DataFrame(data,columns=selcols)
y=pd.DataFrame(data,columns=['class'])
print(x.shape)
print(y.shape)
(400, 8)
(400, 1)
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)#train test split
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
(320, 8)
(320, 1)
(80, 8)
(80, 1)
from sklearn.linear_model import LogisticRegression
lgr = LogisticRegression()
lgr.fit(x_train,y_train)
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
Type here to search
ENG
16:08
12-04-2023

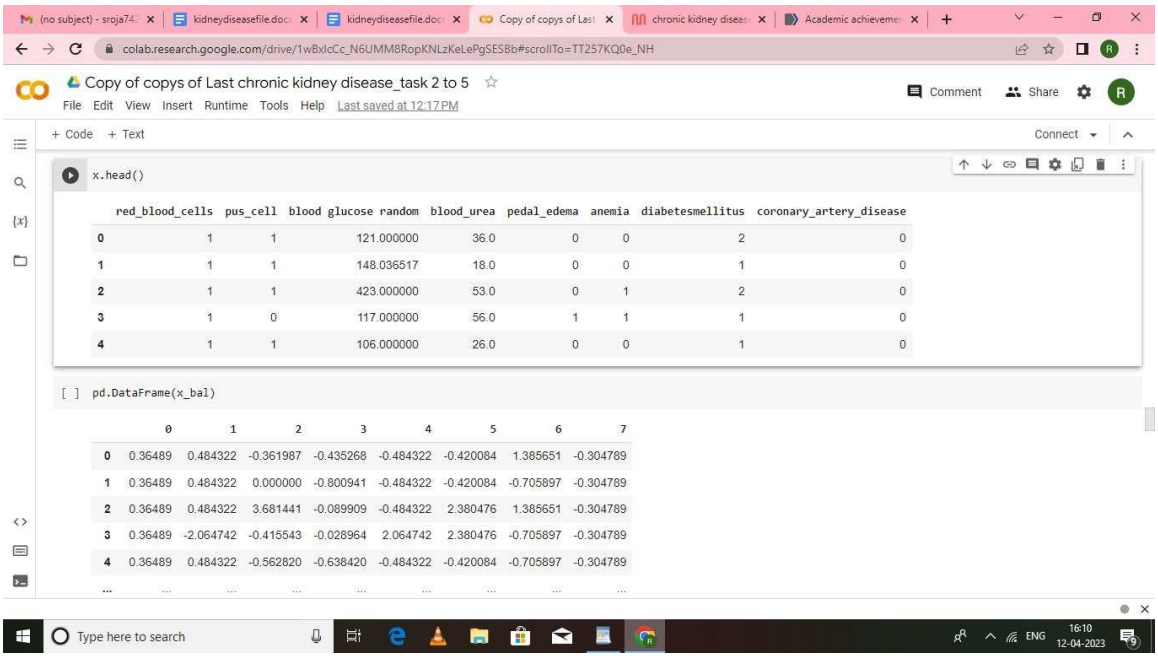

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
lgr.fit(x_train,y_train)
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
y = column_or_1d(y, warn=True)
LogisticRegression()
y_pred = lgr.predict([[129,99,1,0,0,1,0,1]])
print(y_pred)
(y_pred)
[1]
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
warnings.warn(
array([1])
y_pred = lgr.predict(x_test)
accuracy_score(y_test,y_pred)
0.9125
Type here to search
ENG
16:08
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
conf_mat = confusion_matrix(y_test,y_pred)
conf_mat
array([[47, 7],
[0, 26]])
pickle.dump(lgr, open('CKD.pk1','wb'))
contcols
{'age',
'anemia',
'appetite',
'bacteria',
'blood glucose random',
'blood_pressure',
'blood_urea',
'class',
'coronary_artery_disease',
'diabetesmellitus',
'hemoglobin',
'hypertension',
'packed_cell_volume',
'pedal_edema',
'potassium',
'pus_cell',
'pus cell clumps',
Type here to search
ENG
16:09
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
sns.pairplot(data)
<seaborn.axisgrid.PairGrid at 0x7f1242167850>
Type here to search
ENG
16:09
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_bal=sc.fit_transform(x)
[ ] y
class
0 0
1 0
2 0
3 0
4 0
... ...
395 1
396 1
397 1
398 1
399 1
400 rows x 1 columns
Type here to search
ENG
16:09
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_bal=sc.fit_transform(x)
[ ] y
class
0 0
1 0
2 0
3 0
4 0
... ...
395 1
396 1
397 1
398 1
399 1
400 rows x 1 columns
Type here to search
ENG
16:09
12-04-2023



Copy of copys of Last chronic kidney disease_task 2 to 5

File Edit View Insert Runtime Tools Help Last saved at 12:17PM

Comment Share

+ Code + Text

Connect

```
x.head()
```

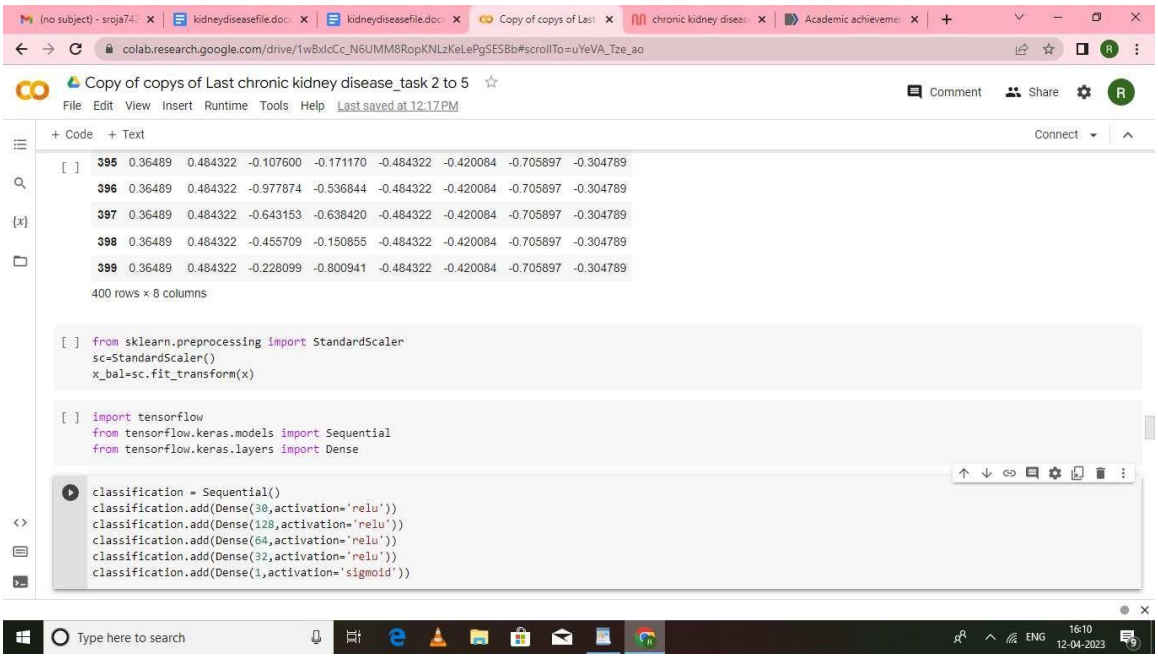
	red_blood_cells	pus_cell	blood glucose random	blood_urea	pedal_edema	anemia	diabetesmellitus	coronary_artery_disease
0	1	1	121.000000	36.0	0	0	2	0
1	1	1	148.036517	18.0	0	0	1	0
2	1	1	423.000000	53.0	0	1	2	0
3	1	0	117.000000	56.0	1	1	1	0
4	1	1	106.000000	26.0	0	0	1	0

```
[ ] pd.DataFrame(x_bal)
```

	0	1	2	3	4	5	6	7
0	0.36489	0.484322	-0.361987	-0.435268	-0.484322	-0.420084	1.385651	-0.304789
1	0.36489	0.484322	0.000000	-0.800941	-0.484322	-0.420084	-0.705897	-0.304789
2	0.36489	0.484322	3.681441	-0.089909	-0.484322	2.380476	1.385651	-0.304789
3	0.36489	-2.064742	-0.415543	-0.028964	2.064742	2.380476	-0.705897	-0.304789
4	0.36489	0.484322	-0.562820	-0.638420	-0.484322	-0.420084	-0.705897	-0.304789
...	...	...	...	...	...	...	...	...

Type here to search

ENG 16:10 12-04-2023



Copy of copys of Last chronic kidney disease_task 2 to 5

File Edit View Insert Runtime Tools Help Last saved at 12:17PM

Comment Share

+ Code + Text

Connect

[ ] 395	0.36489	0.484322	-0.107600	-0.171170	-0.484322	-0.420084	-0.705897	-0.304789
396	0.36489	0.484322	-0.977874	-0.536844	-0.484322	-0.420084	-0.705897	-0.304789
397	0.36489	0.484322	-0.643153	-0.638420	-0.484322	-0.420084	-0.705897	-0.304789
398	0.36489	0.484322	-0.455709	-0.150855	-0.484322	-0.420084	-0.705897	-0.304789
399	0.36489	0.484322	-0.228099	-0.800941	-0.484322	-0.420084	-0.705897	-0.304789

400 rows × 8 columns

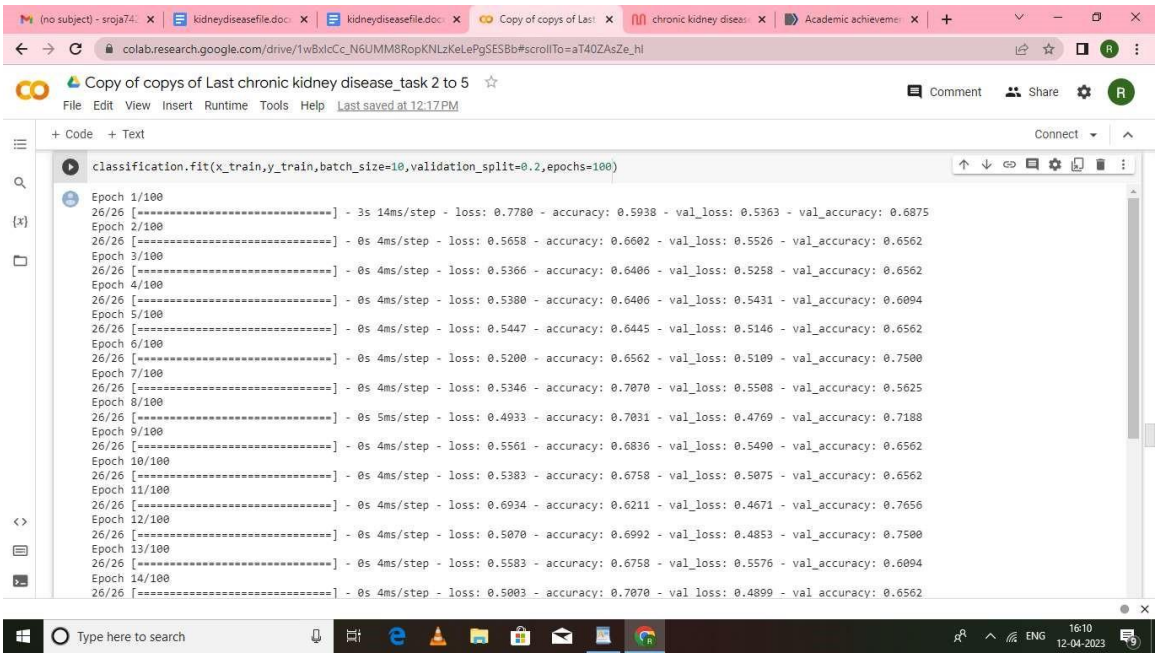
```
[ ] from sklearn.preprocessing import StandardScaler
    sc=StandardScaler()
    x_bal=sc.fit_transform(x)
```

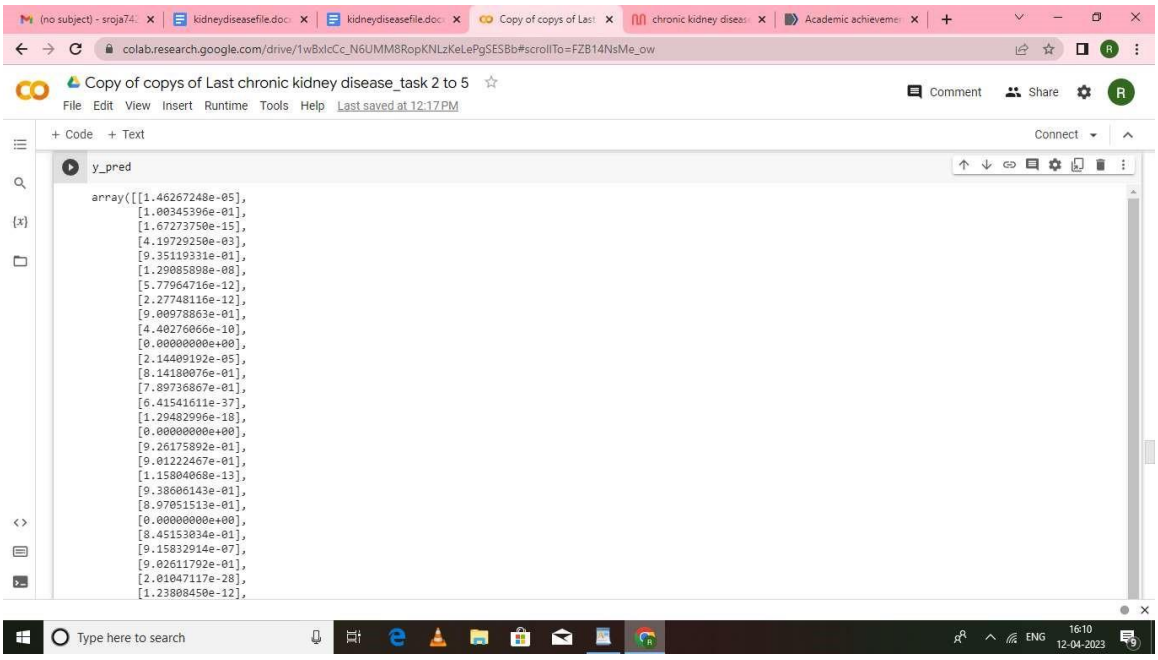
```
[ ] import tensorflow
    from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense
```

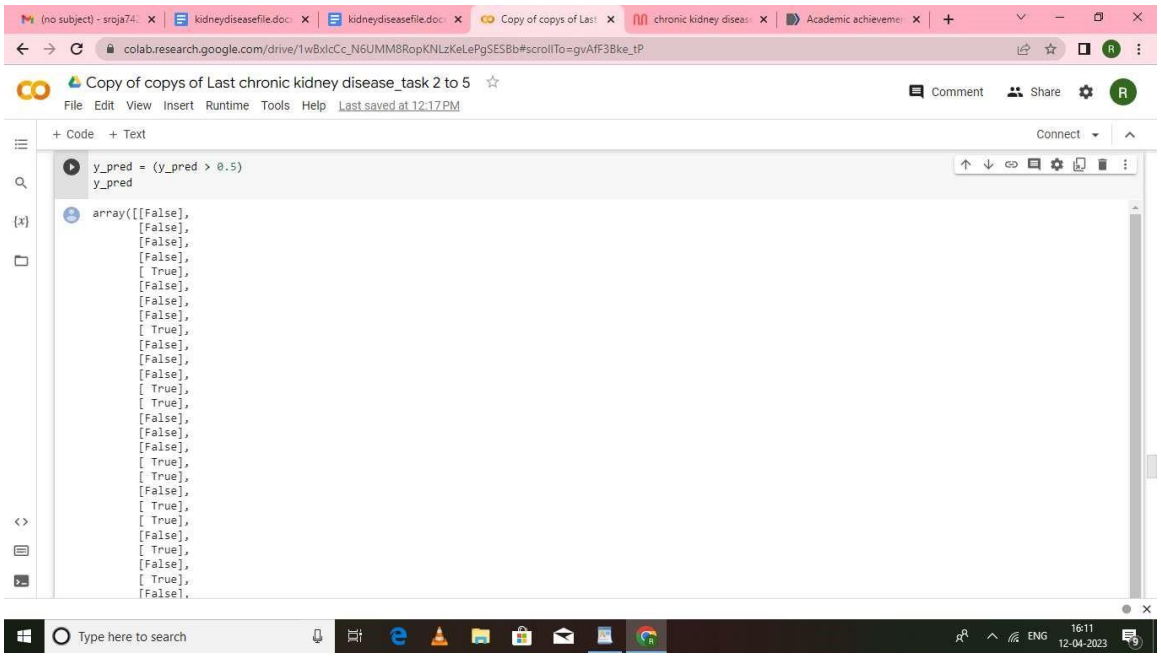
```
classification = Sequential()
classification.add(Dense(30,activation='relu'))
classification.add(Dense(128,activation='relu'))
classification.add(Dense(64,activation='relu'))
classification.add(Dense(32,activation='relu'))
classification.add(Dense(1,activation='sigmoid'))
```

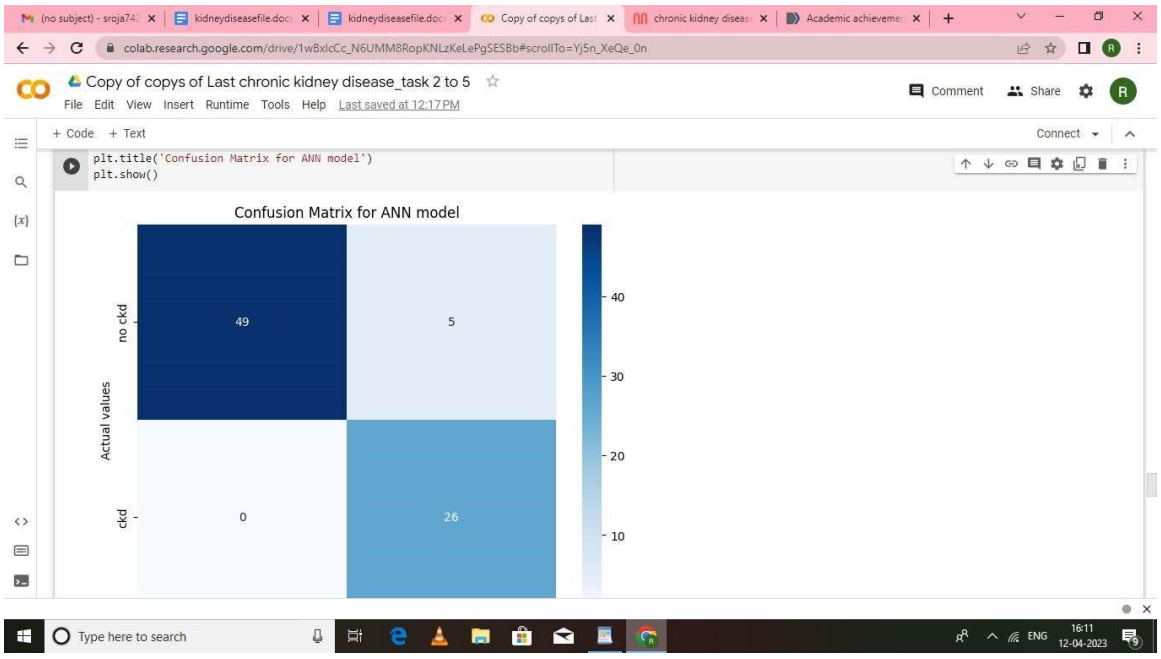
Type here to search

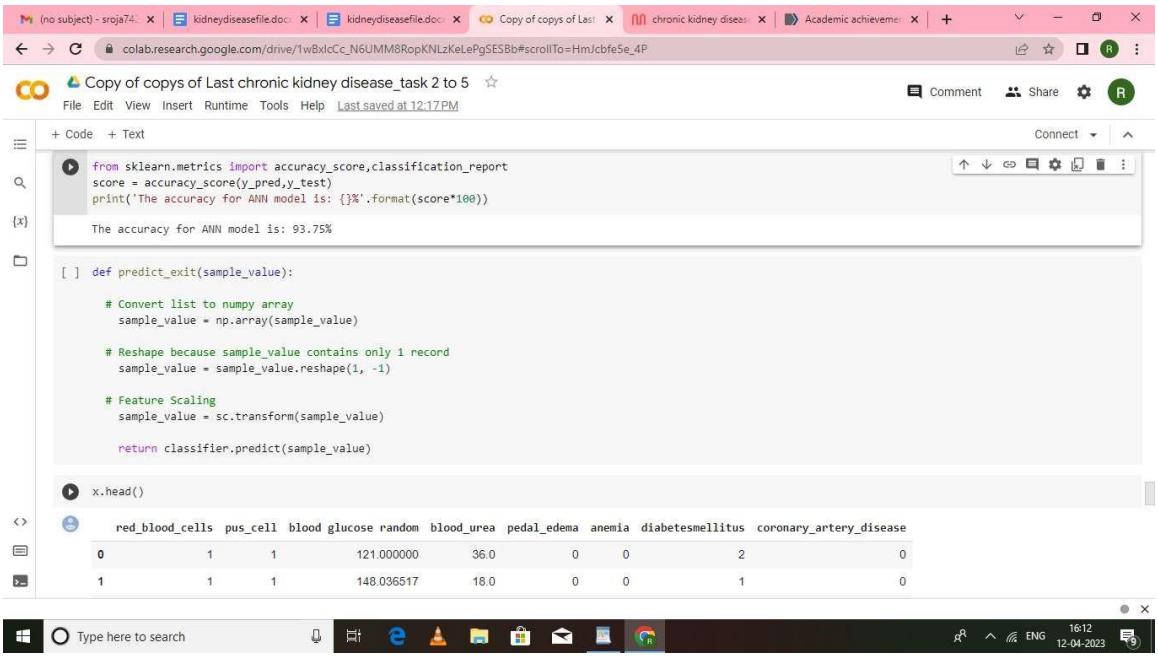
ENG 16:10 12-04-2023

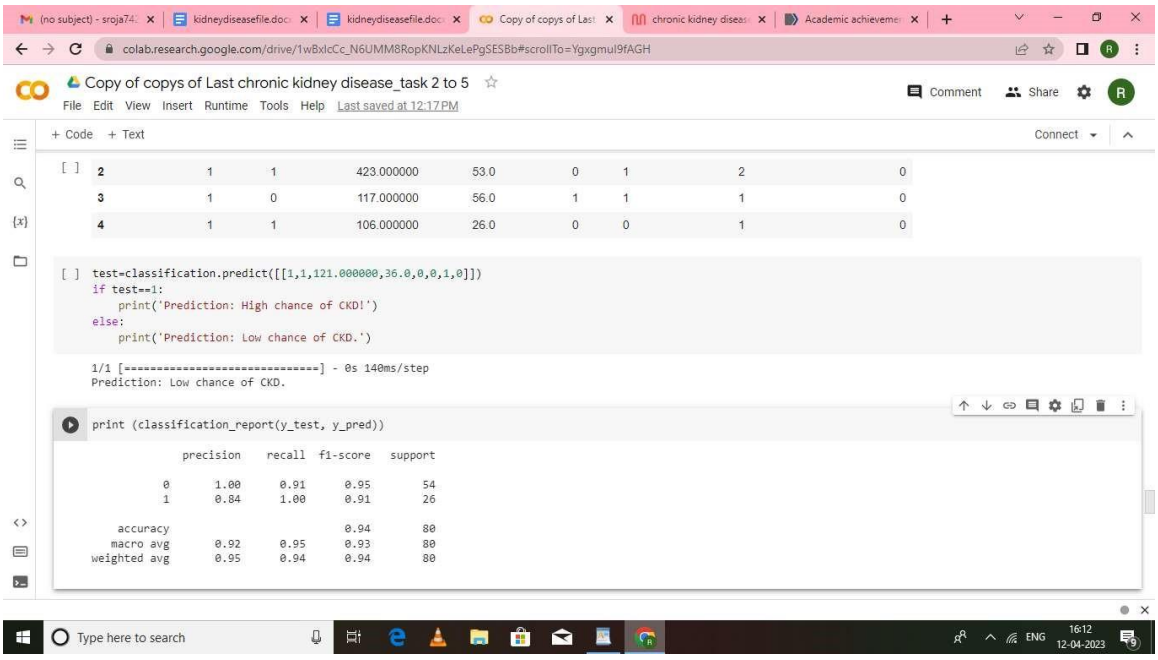

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
classification.fit(x_train,y_train,batch_size=10,validation_split=0.2,epochs=100)
Epoch 1/100
26/26 [==============================] - 3s 14ms/step - loss: 0.7780 - accuracy: 0.5938 - val_loss: 0.5363 - val_accuracy: 0.6875
Epoch 2/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5658 - accuracy: 0.6602 - val_loss: 0.5526 - val_accuracy: 0.6562
Epoch 3/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5366 - accuracy: 0.6406 - val_loss: 0.5258 - val_accuracy: 0.6562
Epoch 4/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5380 - accuracy: 0.6406 - val_loss: 0.5431 - val_accuracy: 0.6094
Epoch 5/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5447 - accuracy: 0.6445 - val_loss: 0.5146 - val_accuracy: 0.6562
Epoch 6/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5200 - accuracy: 0.6562 - val_loss: 0.5109 - val_accuracy: 0.7500
Epoch 7/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5346 - accuracy: 0.7070 - val_loss: 0.5508 - val_accuracy: 0.5625
Epoch 8/100
26/26 [==============================] - 0s 5ms/step - loss: 0.4933 - accuracy: 0.7031 - val_loss: 0.4769 - val_accuracy: 0.7188
Epoch 9/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5561 - accuracy: 0.6836 - val_loss: 0.5490 - val_accuracy: 0.6562
Epoch 10/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5383 - accuracy: 0.6758 - val_loss: 0.5075 - val_accuracy: 0.6562
Epoch 11/100
26/26 [==============================] - 0s 4ms/step - loss: 0.6934 - accuracy: 0.6211 - val_loss: 0.4671 - val_accuracy: 0.7656
Epoch 12/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5070 - accuracy: 0.6992 - val_loss: 0.4853 - val_accuracy: 0.7500
Epoch 13/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5583 - accuracy: 0.6758 - val_loss: 0.5576 - val_accuracy: 0.6094
Epoch 14/100
26/26 [==============================] - 0s 4ms/step - loss: 0.5003 - accuracy: 0.7070 - val_loss: 0.4899 - val_accuracy: 0.6562
Type here to search
ENG
16:10
12-04-2023

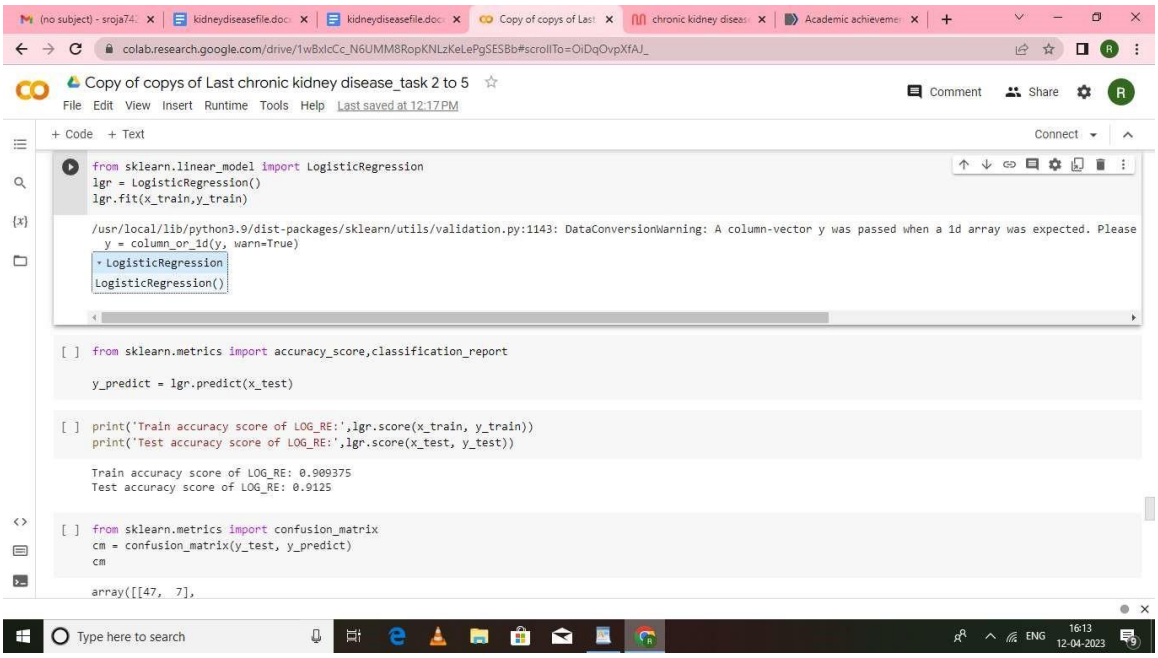

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
y_pred
array([[1.46267248e-05],
[1.00345396e-01],
[1.67273750e-15],
[4.19729250e-03],
[9.35119331e-01],
[1.29085898e-08],
[5.77964716e-12],
[2.27748116e-12],
[9.00978863e-01],
[4.40276066e-10],
[0.00000000e+00],
[2.14409192e-05],
[8.14180076e-01],
[7.89736867e-01],
[6.41541611e-37],
[1.29482996e-18],
[0.00000000e+00],
[9.26175892e-01],
[9.01222467e-01],
[1.15804068e-13],
[9.38606143e-01],
[8.97051513e-01],
[0.00000000e+00],
[8.45153034e-01],
[9.15832914e-07],
[9.02611792e-01],
[2.01047117e-28],
[1.23808450e-12],
Type here to search
ENG
16:10
12-04-2023

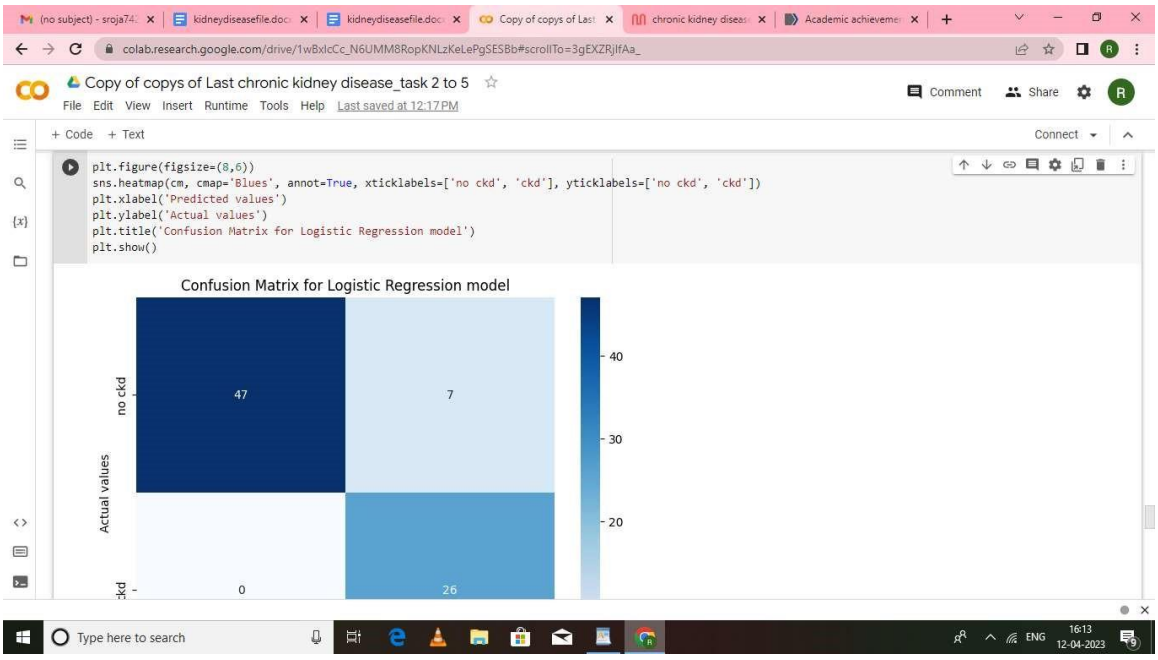

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
y_pred = (y_pred > 0.5)
y_pred
array([[False],
[False],
[False],
[False],
[True],
[False],
[False],
[False],
[True],
[False],
[False],
[False],
[True],
[True],
[False],
[False],
[False],
[True],
[True],
[False],
[True],
[True],
[False],
[True],
[False],
[True],
[False],
Type here to search
ENG
16:11
12-04-2023

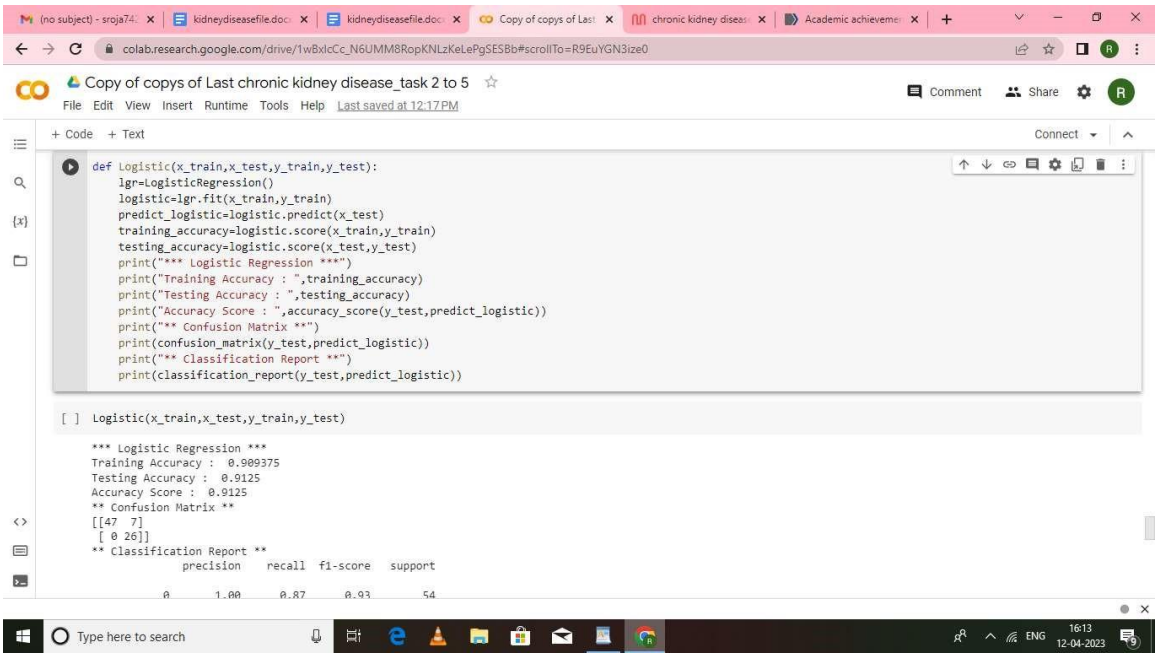

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
plt.title('Confusion Matrix for ANN model')
plt.show()
Confusion Matrix for ANN model
Actual values
no ckd
ckd
49
5
0
26
40
30
20
10
Type here to search
ENG
16:11
12-04-2023

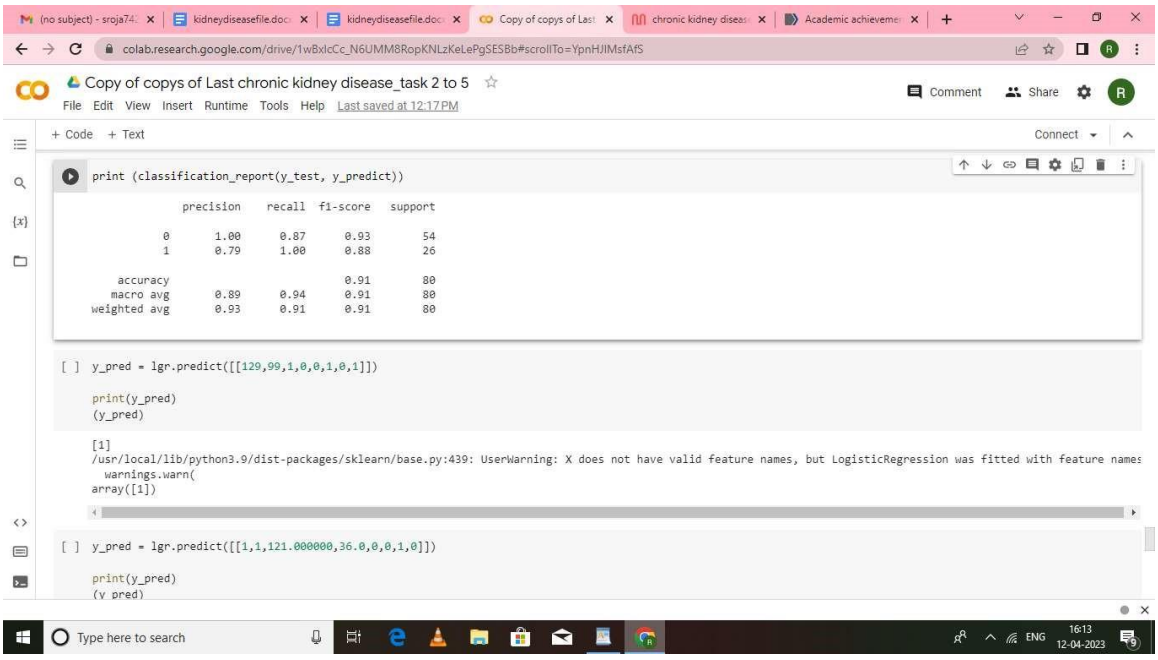

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
from sklearn.metrics import accuracy_score,classification_report
score = accuracy_score(y_pred,y_test)
print('The accuracy for ANN model is: {}%'.format(score*100))
The accuracy for ANN model is: 93.75%
def predict_exit(sample_value):
# Convert list to numpy array
sample_value = np.array(sample_value)
# Reshape because sample_value contains only 1 record
sample_value = sample_value.reshape(1, -1)
# Feature Scaling
sample_value = sc.transform(sample_value)
return classifier.predict(sample_value)
x.head()
red_blood_cells pus_cell blood glucose random blood_urea pedal_edema anemia diabetesmellitus coronary_artery_disease
0 1 1 121.000000 36.0 0 0 2 0
1 1 1 148.036517 18.0 0 0 1 0
Type here to search
ENG
16:12
12-04-2023

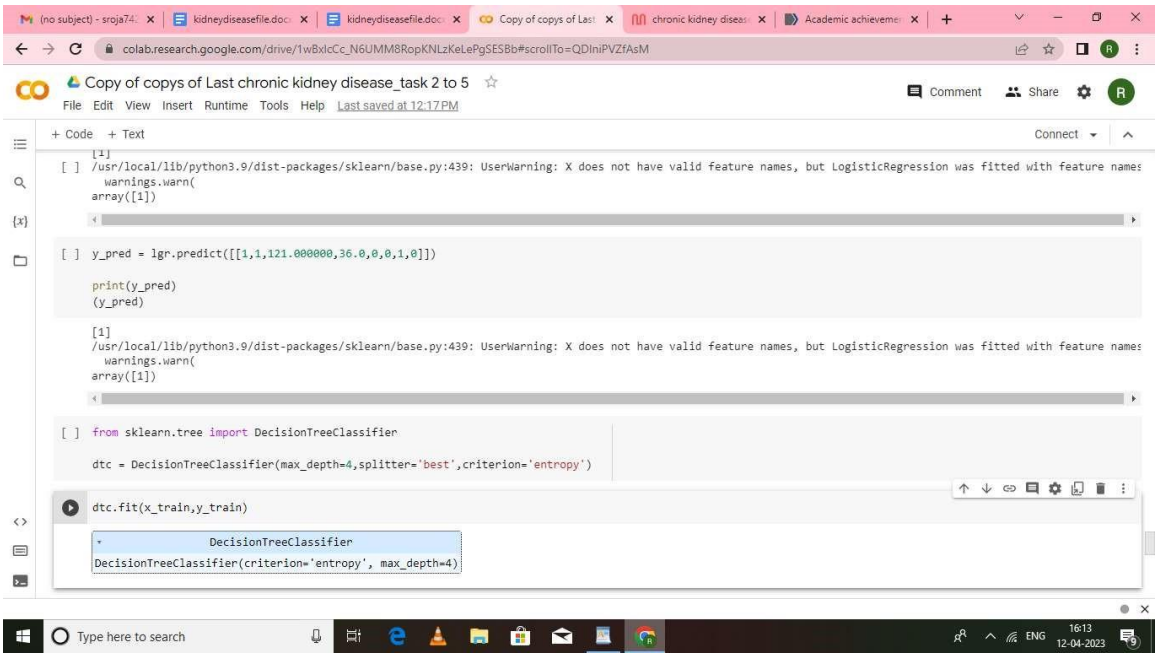

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
2 1 1 423.000000 53.0 0 1 2 0
3 1 0 117.000000 56.0 1 1 1 0
4 1 1 106.000000 26.0 0 0 1 0
test=classification.predict([[1,1,121.000000,36.0,0,0,1,0]])
if test==1:
print('Prediction: High chance of CKD!')
else:
print('Prediction: Low chance of CKD.')
1/1 [==============================] - 0s 140ms/step
Prediction: Low chance of CKD.
print (classification_report(y_test, y_pred))
precision recall f1-score support
0 1.00 0.91 0.95 54
1 0.84 1.00 0.91 26
accuracy 0.94 80
macro avg 0.92 0.95 0.93 80
weighted avg 0.95 0.94 0.94 80
Type here to search
ENG
16:12
12-04-2023

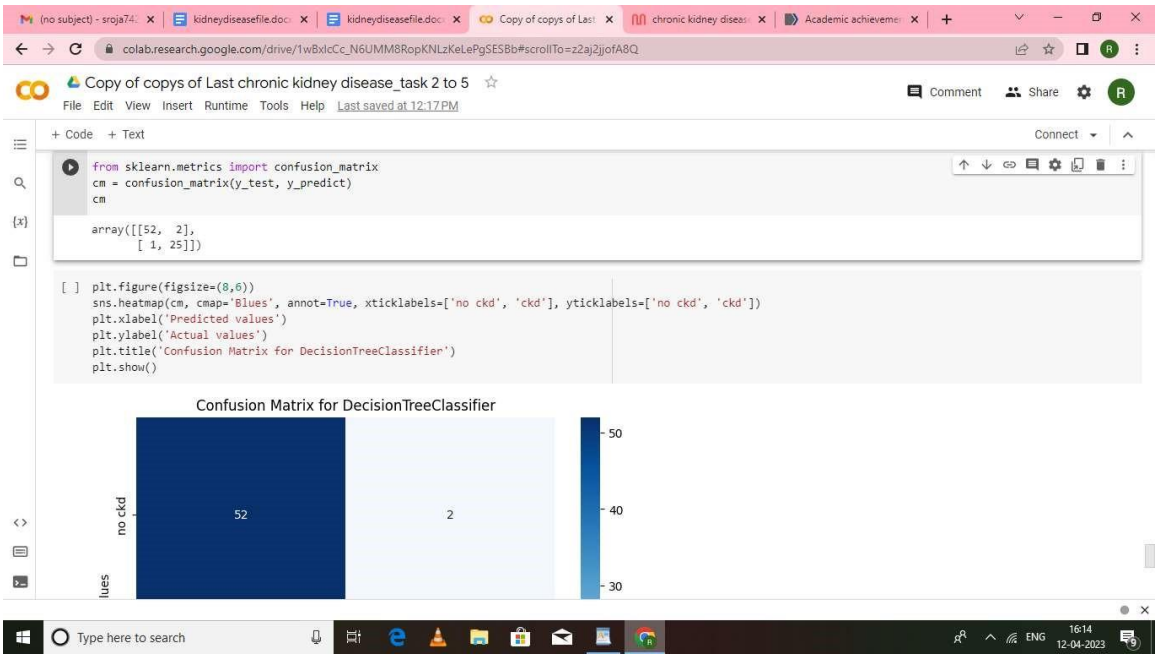

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
from sklearn.linear_model import LogisticRegression
lgr = LogisticRegression()
lgr.fit(x_train,y_train)
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
y = column_or_1d(y, warn=True)
LogisticRegression
LogisticRegression()
from sklearn.metrics import accuracy_score,classification_report
y_predict = lgr.predict(x_test)
print('Train accuracy score of LOG_RE:',lgr.score(x_train, y_train))
print('Test accuracy score of LOG_RE:',lgr.score(x_test, y_test))
Train accuracy score of LOG_RE: 0.909375
Test accuracy score of LOG_RE: 0.9125
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm
array([[47, 7],
Type here to search
ENG
16:13
12-04-2023

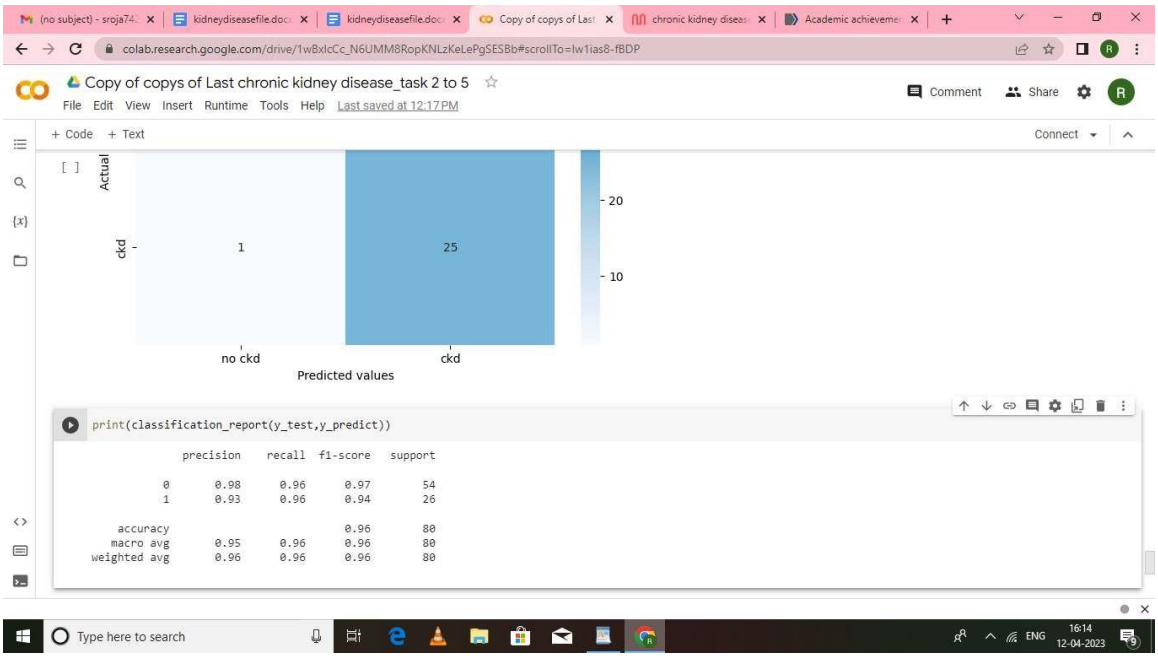

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
plt.figure(figsize=(8,6))
sns.heatmap(cm, cmap='Blues', annot=True, xticklabels=['no ckd', 'ckd'], yticklabels=['no ckd', 'ckd'])
plt.xlabel('Predicted values')
plt.ylabel('Actual values')
plt.title('Confusion Matrix for Logistic Regression model')
plt.show()
Confusion Matrix for Logistic Regression model
Actual values
no ckd
ckd
47
7
0
26
40
30
20
Type here to search
ENG
16:13
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
def Logistic(x_train,x_test,y_train,y_test):
lgr=LogisticRegression()
logistic=lgr.fit(x_train,y_train)
predict_logistic=logistic.predict(x_test)
training_accuracy=logistic.score(x_train,y_train)
testing_accuracy=logistic.score(x_test,y_test)
print("*** Logistic Regression ***")
print("Training Accuracy : ",training_accuracy)
print("Testing Accuracy : ",testing_accuracy)
print("Accuracy Score : ",accuracy_score(y_test,predict_logistic))
print("** Confusion Matrix **")
print(confusion_matrix(y_test,predict_logistic))
print("** Classification Report **")
print(classification_report(y_test,predict_logistic))
[ ] Logistic(x_train,x_test,y_train,y_test)
*** Logistic Regression ***
Training Accuracy : 0.909375
Testing Accuracy : 0.9125
Accuracy Score : 0.9125
** Confusion Matrix **
[[47 7]
[0 26]]
** Classification Report **
precision recall f1-score support
0 1.00 0.87 0.93 54
Type here to search
ENG
16:13
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
print (classification_report(y_test, y_predict))
precision recall f1-score support
0 1.00 0.87 0.93 54
1 0.79 1.00 0.88 26
accuracy 0.91 80
macro avg 0.89 0.94 0.91 80
weighted avg 0.93 0.91 0.91 80
[ ] y_pred = lgr.predict([[129,99,1,0,0,1,0,1]])
print(y_pred)
(y_pred)
[1]
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
warnings.warn(
array([1])
[ ] y_pred = lgr.predict([[1,1,121.000000,36.0,0,0,1,0]])
print(y_pred)
(y_pred)
Type here to search
ENG
16:13
12-04-2023

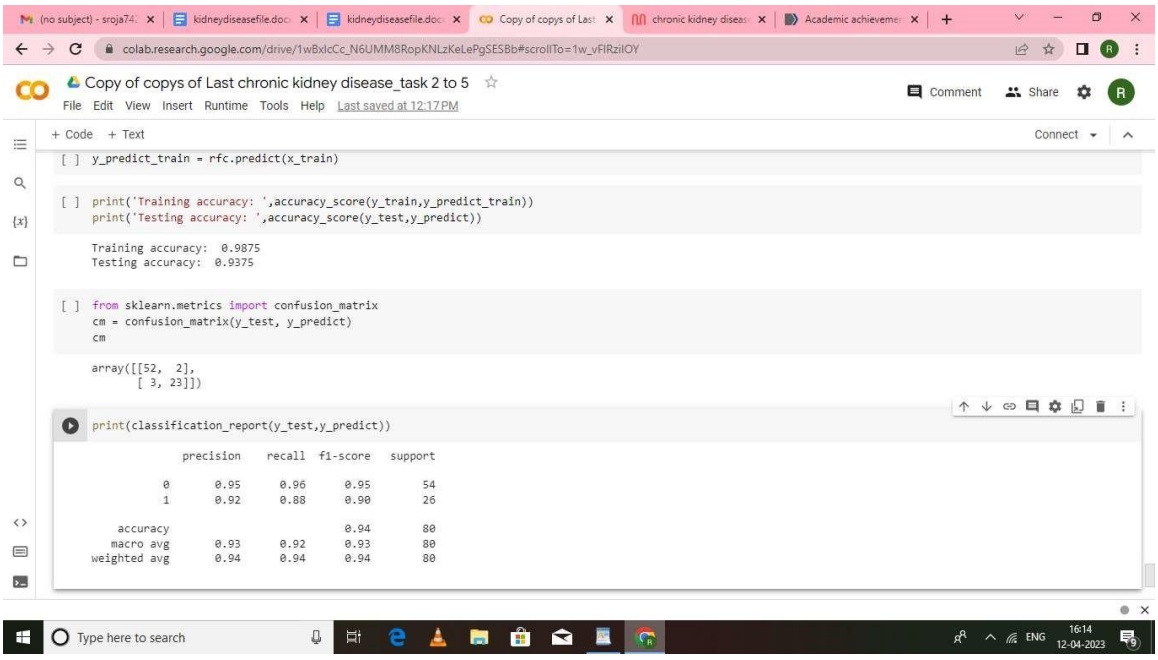

Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
[1]
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
warnings.warn(
array([1])
y_pred = lgr.predict([[1,1,121.000000,36.0,0,0,1,0]])
print(y_pred)
(y_pred)
[1]
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
warnings.warn(
array([1])
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(max_depth=4,splitter='best',criterion='entropy')
dtc.fit(x_train,y_train)
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=4)
Type here to search
ENG
16:13
12-04-2023


Copy of copys of Last chronic kidney disease_task 2 to 5
File Edit View Insert Runtime Tools Help Last saved at 12:17 PM
Comment
Share
+ Code + Text
Connect
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm
array([[52, 2],
[1, 25]])
plt.figure(figsize=(8,6))
sns.heatmap(cm, cmap='Blues', annot=True, xticklabels=['no ckd', 'ckd'], yticklabels=['no ckd', 'ckd'])
plt.xlabel('Predicted values')
plt.ylabel('Actual values')
plt.title('Confusion Matrix for DecisionTreeClassifier')
plt.show()
Confusion Matrix for DecisionTreeClassifier
no ckd
52
2
50
40
30
Type here to search
ENG
16:14
12-04-2023




```
print(classification_report(y_test,y_predict))
```

```
              precision    recall  f1-score   support

           0       0.98      0.96      0.97        54
           1       0.93      0.96      0.94        26

    accuracy                           0.96        80
   macro avg       0.95      0.96      0.96        80
weighted avg       0.96      0.96      0.96        80
```



```
y_predict_train = rfc.predict(x_train)
```

```
print('Training accuracy: ',accuracy_score(y_train,y_predict_train))
print('Testing accuracy: ',accuracy_score(y_test,y_predict))
```

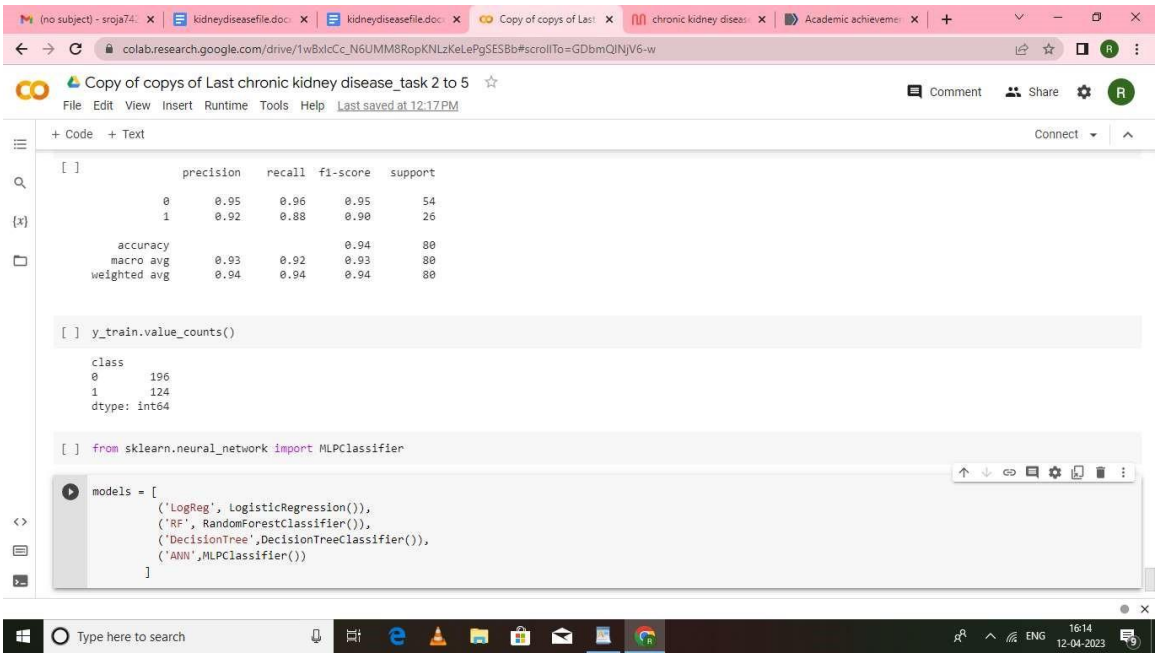
```
Training accuracy:  0.9875
Testing accuracy:  0.9375
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm
```

```
array([[52,  2],
       [ 3, 23]])
```

```
print(classification_report(y_test,y_predict))
```

```
              precision    recall  f1-score   support

           0       0.95      0.96      0.95        54
           1       0.92      0.88      0.90        26

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



ADVANTAGES AND DISADVANTAGES:

Advantages

- The number of patients on renal replacement therapy has doubled every decade since 1980, and prevalence of chronic kidney disease (CKD) in the early stages is also markedly increased.
- In addition, CKD is a significant risk factor for cardiovascular morbidity and mortality. The only effective approach to this problem is prevention and early detection of CKD. In recent years, screening studies have been carried out in several countries
- The findings have defined the scope of the problem and indicated which population groups are at risk of developing CKD. The most numerous are patients with hypertension and diabetes. Also, these studies have indicated that screening should include measurement of serum as well as urine albumin.
- Early detection of CKD allows proper management that could slow down CKD progression, prevent cardiovascular and enable timely initiation of dialysis. Screening for CKD could be best managed by partnership between primary care physicians and nephrologists. It is necessary to educate primary care physicians about CKD, its risk factors and associated co-morbidities.
- Although multiple benefits of screening for CKD are doubtless, the results obtained by screening should be interpreted with caution, bearing in mind that screening detects only markers of kidney disease but not the disease itself.

Disadvantages

- Having CKD increases the chances of having heart disease and stroke. Managing high blood pressure, blood sugar, and cholesterol levels—all factors that increase the risk for heart disease and stroke—is very important for people with CKD.

- This happens when your kidneys don't make enough erythropoietin (EPO), which affects their ability to make red blood cells.
- Bone weakness
- Fluid retention
- Gout.
- Heart disease.
- High blood pressure (hypertension).
- Metabolic acidosis.

Applications:

- The main treatments are: lifestyle changes – to help you stay as healthy as possible. Medicine – to control associated problems, such as high blood pressure and high cholesterol. Dialysis – treatment to replicate some of the kidney's functions, which may be necessary in advanced (stage 5) CKD.
- Lifestyle changes
- The following lifestyle measures are usually recommended for people with kidney disease:
- Stop smoking if you

  smoke eat a healthy,

  balanced diet

- Medicine
- There's no medicine specifically for CKD, but medicine can help control many of the problems that cause the condition and the complications that can happen as a result of it.
- You may need to take medicine to treat or prevent the different problems caused by CKD.
- High blood pressure
- Good control of blood pressure is vital to protect the kidneys.
- People with kidney disease should usually aim to get their blood pressure down to below 140/90mmHg, but you should aim to get it down to below 130/80mmHg if you also have diabetes

CONCLUSION :

Chronic renal failure represents a critical period in the evolution of chronic renal disease and is associated with complications and co morbidities that begin early in the course of the disease. These conditions are initially subclinical but progress relentlessly and may eventually become symptomatic and irreversible. Early in the course of chronic renal failure, these conditions are amenable to interventions with relatively simple treatments that have the potential to prevent adverse outcomes. Strategies for effective management of chronic renal disease. By acknowledging these facts, we have an excellent opportunity to change the paradigm of management of chronic renal failure and improve patient outcomes.

Additional educational resources

Integrated care by the primary care physician, renal team from an early stage is vital to reduce the overall morbidity and mortality associated with chronic renal disease. Practical points helpful at this stage of renal disease include

Patients receiving comprehensive care by the renal team have shown slower rates of decline in renal function, greater probability of starting dialysis with higher calcium control, a permanent access, and a greater likelihood of choosing peritoneal dialysisw12

Patients with progressive renal failure should be educated to save vessels of the non-dominant arm for future access; they should have a permanent vascular access (preferably fistula) created when the filtration rate falls below 25 ml/min or renal replacement treatment is anticipated within a year

FUTURE SCOPE :

Imaging: assessing number as a determinant of kidney disease and kidney fibrosis

Imaging techniques have the advantage of being non-invasive and, thus, may be safely repeated to evaluate changes, providing information for both kidneys and potentially combining functional with morphological information. The most interesting advances relate to estimation of number, overall kidney functions, fibrosis and new functional magnetic resonance imaging (MRI), as well as ultrasound techniques such as diffusion-weighted MRI (DWI or DW-MRI), blood oxygenation level–dependent MRI (BOLD-MRI), perfusion MRI, hyperpolarized (HP) carbon 13 MRI (13C MRI) and contrast-enhanced ultrasound (CEUS).

CFE-MRI uses filled with iron oxide future treatments for kidney disease

Novel therapeutic alternatives for ESRD include wearable artificial kidneys, stem cell–based therapy, and bioengineered and bio-artificial kidneys. Of note, one of the main objectives of these novel therapeutic approaches should be to maintain patients at home and to avoid Dialysis APPENDIX import pandas as Pd import from collections import Counter as c

APPENDIX

Source code

```
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 classess
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 #splits 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


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

data=pd.read_csv("kidney_disease.csv")
data.head()
```

```
Data preparation
data.tail()

data.head()


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

data.columns
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',
              'hemoglobin','packed_cell_volume','white_blood_cell_count','red_blood_cell_count',
              'hypertension','diabetesmellitus','coronary_artery_disease','appetite',
              'pedal_edema','anemia','class']
data.columns
data.info()
data.isnull().any()
data.isnull().sum()
data.describe()
data['class'].unique()
data['class']=data['class'].replace("ckd\t","ckd")
data['class'].unique()
data['class']=data['class'].replace("ckd\t","ckd")
data['class'].unique()
np.unique(data.dtypes,return_counts=True)

data['age'].fillna(data['age'].mode()[0],inplace=True)
data['hypertension'].fillna(data['hypertension'].mode()[0],inplace=True)
data['pus_cell_clumps'].fillna(data['pus_cell_clumps'].mode()[0],inplace=True)
data['appetite'].fillna(data['appetite'].mode()[0],inplace=True)
data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)
data['pus_cell'].fillna(data['pus_cell'].mode()[0],inplace=True)
data['red_blood_cells'].fillna(data['red_blood_cells'].mode()[0],inplace=True)
data['coronary_artery_disease'].fillna(data['coronary_artery_disease'].mode()[0],inplace=True)
data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)
data['anemia'].fillna(data['anemia'].mode()[0],inplace=True)
data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)
data['diabetesmellitus'].fillna(data['diabetesmellitus'].mode()[0],inplace=True)
data['pedal_edema'].fillna(data['pedal_edema'].mode()[0],inplace=True)
data['specific_gravity'].fillna(data['specific_gravity'].mode()[0],inplace=True)

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


for i in catcols:
  print("Columns:",i)
  print(c(data[i]))
  print('*'*120+'\n')

catcols.remove('red_blood_cell_count')
catcols.remove('packed_cell_volume')
catcols.remove('white_blood_cell_count')
print(catcols)
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)



catcols.add('specific_gravity')
catcols.add('albumin')
catcols.add('sugar')
print(catcols)
data['coronary_artery_disease']=data.coronary_artery_disease.replace('\tno','no')
c(data['coronary_artery_disease'])
data['diabetesmellitus']=data.diabetesmellitus.replace(to_replace={'\tno':'no','\tyes':'yes','yes':'yes
'})
c(data['diabetesmellitus'])
data.isnull().any()


data.isnull().sum()


msno.bar(data)
plt.show()



data.packed_cell_volume = pd.to_numeric(data.packed_cell_volume, errors='coerce')
data.white blood cell count = pd.to numeric(data.white blood cell count, errors='coerce')
data.red_blood_cell_count = pd.to_numeric(data.red_blood_cell_count, errors='coerce')



data['blood glucose random'].fillna(data['blood glucose random'].mean(),inplace=True)
data['blood pressure'].fillna(data['blood pressure'].mean(),inplace=True)
data['blood_urea'].fillna(data['blood_urea'].mean(),inplace=True)
data['hemoglobin'].fillna(data['hemoglobin'].mean(),inplace=True)
data['packed_cell_volume'].fillna(data['packed_cell_volume'].mean(),inplace=True)
data['potassium'].fillna(data['potassium'].mean(),inplace=True)
data['red_blood_cell_count'].fillna(data['red_blood_cell_count'].mean(),inplace=True)
data['serum_creatinine'].fillna(data['serum_creatinine'].mean(),inplace=True)
data['sodium'].fillna(data['sodium'].mean(),inplace=True)
data['white_blood_cell_count'].fillna(data['white_blood_cell_count'].mean(),inplace=True)

data['age'].fillna(data['age'].mode()[0],inplace=True)
data['hypertension'].fillna(data['hypertension'].mode()[0],inplace=True)
data['pus_cell_clumps'].fillna(data['pus_cell_clumps'].mode()[0],inplace=True)
data['appetite'].fillna(data['appetite'].mode()[0],inplace=True)
data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)
data['pus_cell'].fillna(data['pus_cell'].mode()[0],inplace=True)
data['red_blood_cells'].fillna(data['red_blood_cells'].mode()[0],inplace=True)
data['coronary_artery_disease'].fillna(data['coronary_artery_disease'].mode()[0],inplace=True)
data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)
data['anemia'].fillna(data['anemia'].mode()[0],inplace=True)
data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)
data['diabetesmellitus'].fillna(data['diabetesmellitus'].mode()[0],inplace=True)
data['pedal_edema'].fillna(data['pedal_edema'].mode()[0],inplace=True)
```

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

data.isnull().sum()
catcols=['anemia','pedal_edema','appetite','bacteria','class','coronary_artery_disease','diabetesmellit
us',
         'hypertension','pus_cell','pus_cell_clumps','red_blood_cells']

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

plotnumber

import matplotlib.pyplot as plt # import the matplotlib libaray
fig=plt.figure(figsize=(5,5)) #plot size
plt.scatter(data['age'],data['blood_pressure'],color='blue')
plt.xlabel('age') #set the label for x-axis
plt.ylabel('blood pressure') #set the label for y-axis
plt.title("age VS blood Scatter Plot") #set a title for the axes

for column in contcols:
    if plotnumber<=11 :
        ax = plt.subplot(3,4,plotnumber)
        plt.scatter(data['age'],data[column])
        plt.xlabel(column,fontsize=20)
    plotnumber+=1
plt.show()



f,ax=plt.subplots(figsize=(18,10))
sns.heatmap(data.corr(),annot=True,fmt=".2f",ax=ax,linewidths=0.5,linecolor="orange")
plt.xticks(rotation=45)
plt.yticks(rotation=45)
plt.show()
data['class'].unique()
f,ax=plt.subplots(figsize=(18,10))
sns.heatmap(data.corr(),annot=True,fmt=".2f",ax=ax,linewidths=0.5,linecolor="orange")
plt.xticks(rotation=45)
plt.yticks(rotation=45)
plt.show()
sns.countplot(data['class'])
selcols=['red_blood_cells','pus_cell', 'blood glucose random','blood_urea',
            'pedal_edema', 'anemia','diabetesmellitus','coronary_artery_disease']
x=pd.DataFrame(data,columns=selcols)
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)#train test split
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)


from sklearn.linear_model import LogisticRegression
lgr = LogisticRegression()
```

```
lgr.fit(x_train,y_train)


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

print(y_pred)
(y_pred)


y_pred = lgr.predict(x_test)
accuracy_score(y_test,y_pred)
conf_mat = confusion_matrix(y_test,y_pred)
conf_mat

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


contcols


sns.pairplot(data)
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_bal=sc.fit_transform(x)
y
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)


x.head()


pd.DataFrame(x_bal)
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_bal=sc.fit_transform(x)
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.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)
y_pred = classification.predict(x_test)
y_pred
y_pred = (y_pred > 0.5)
y_pred
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
plt.figure(figsize=(8,6))
sns.heatmap(cm, cmap='Blues', annot=True, xticklabels=['no ckd', 'ckd'], yticklabels=['no ckd', 'ckd'])
plt.xlabel('Predicted values')
plt.ylabel('Actual values')
plt.title('Confusion Matrix for ANN model')
plt.show()
```

```
from sklearn.metrics import accuracy_score,classification_report
score = accuracy_score(y_pred,y_test)
print('The accuracy for ANN model is: {}%'.format(score*100))


def predict_exit(sample_value):

  # Convert list to numpy array
    sample_value = np.array(sample_value)

  # Reshape because sample_value contains only 1 record
    sample_value = sample_value.reshape(1, -1)

  # Feature Scaling
    sample value = sc.transform(sample value)

    return classifier.predict(sample_value)

x.head()
test=classification.predict([[1,1,121.000000,36.0,0,0,1,0]])
if test==1:
    print('Prediction: High chance of CKD!')
else:
    print('Prediction: Low chance of CKD.')
print (classification report(y test, y pred))
from sklearn.linear_model import LogisticRegression
lgr = LogisticRegression()
lgr.fit(x_train,y_train)
from sklearn.metrics import accuracy_score,classification_report

y_predict = lgr.predict(x_test)
print('Train accuracy score of LOG_RE:',lgr.score(x_train, y_train))
print('Test accuracy score of LOG_RE:',lgr.score(x_test, y_test))
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm
plt.figure(figsize=(8,6))
sns.heatmap(cm, cmap='Blues', annot=True, xticklabels=['no ckd', 'ckd'], yticklabels=['no ckd', 'ckd'])
plt.xlabel('Predicted values')
plt.ylabel('Actual values')
plt.title('Confusion Matrix for Logistic Regression model')
plt.show()
def logistic(x_train,x_test,y_train,y_test):
  lgr.fit(x_train,y_train)
  y_predict = lgr.predict(x_test)
  print("")
def Logistic(x_train,x_test,y_train,y_test):
    lgr=LogisticRegression()
    logistic=lgr.fit(x train,y train)
    predict_logistic=logistic.predict(x_test)
    training_accuracy=logistic.score(x_train,y_train)
    testing_accuracy=logistic.score(x_test,y_test)
    print("*** Logistic Regression ***")
    print("Training Accuracy : ",training_accuracy)
    print("Testing Accuracy : ",testing_accuracy)
    print("Accuracy Score : ",accuracy_score(y_test,predict_logistic))
    print("** Confusion Matrix **")
    print(confusion_matrix(y_test,predict_logistic))
    print("** Classification Report **")
    print(classification_report(y_test,predict_logistic))
Logistic(x_train,x_test,y_train,y_test)
print (classification_report(y_test, y_predict))
```

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

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

print(y_pred)
(y_pred)

y_pred = lgr.predict([[1,1,121.000000,36.0,0,0,1,0]])

print(y_pred)
(y_pred)
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)


print('Testing accuracy = ', accuracy_score(y_test,y_predict))
print("Training accuracy= ",accuracy_score(y_train,y_predict_train))
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm
plt.figure(figsize=(8,6))
sns.heatmap(cm, cmap='Blues', annot=True, xticklabels=['no ckd', 'ckd'], yticklabels=[
'no ckd', 'ckd'])
plt.xlabel('Predicted values')
plt.ylabel('Actual values')
plt.title('Confusion Matrix for DecisionTreeClassifier')
plt.show()
print(classification_report(y_test,y_predict))


from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=10,criterion='entropy')
rfc.fit(x_train,y_train)
y_predict = rfc.predict(x_test)
y_predict_train = rfc.predict(x_train)
print('Training accuracy: ',accuracy_score(y_train,y_predict_train))
print('Testing accuracy: ',accuracy_score(y_test,y_predict))
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm

print(classification_report(y_test,y_predict))

y_train.value_counts()
from sklearn.neural_network import MLPClassifier
models = [
          ('LogReg', LogisticRegression()),
```

```
    ('RF', RandomForestClassifier()),
    ('DecisionTree',DecisionTreeClassifier()),
    ('ANN',MLPClassifier())
]
```