EARLY PREDICTION FOR CHRONIC KIDNEY DISEASE DETECTION :

A PROGRESSIVE APPROACH TO HEALTH MANAGEMENT

TEAM LEADER :

S.Devadharshini

TEAM MEMBERS :

G.Dheivaraj

A.Hariharasuthan

Introduction:

Kidney disease is a prevalent health problem that affects millions of people worldwide. It is a complex disease that requires careful monitoring and treatment to prevent its progression. Machine learning (ML) algorithms have the potential to improve the diagnosis and management of kidney disease by analyzing large amounts of patient data to identify patterns and predict disease progression. In this report, we

present an analysis of the current state of ML applications for kidney disease and suggest future directions for research.

Literature Review:

Several studies have demonstrated the potential of ML algorithms for kidney disease analysis. For instance, Al-Taie et al. (2020) developed a predictive model for end-stage renal disease (ESRD) using a combination of demographic, clinical, and laboratory data. The model achieved an accuracy of 85.7% in predicting ESRD within five years. In another study, Kalra et al. (2019) developed an ML-based tool to predict acute kidney injury (AKI) in critically ill patients. The tool achieved an area under the curve (AUC) of 0.81 in predicting AKI within 48 hours.

Other studies have focused on using ML algorithms to identify biomarkers for kidney disease. For example, Wang et al. (2020) used a deep learning algorithm to analyze kidney biopsy images and identified novel biomarkers for diabetic nephropathy. Similarly, Yang et al. (2019) used an ML-based approach to identify gene expression patterns that were associated with the progression of chronic kidney disease (CKD).

Discussion:

The studies reviewed here demonstrate the potential of ML algorithms for kidney disease analysis. ML can be used to identify patterns and predict disease progression, which can help clinicians in the diagnosis and management of kidney disease. ML can also be used to identify biomarkers for kidney disease, which can aid in the development of new treatments.

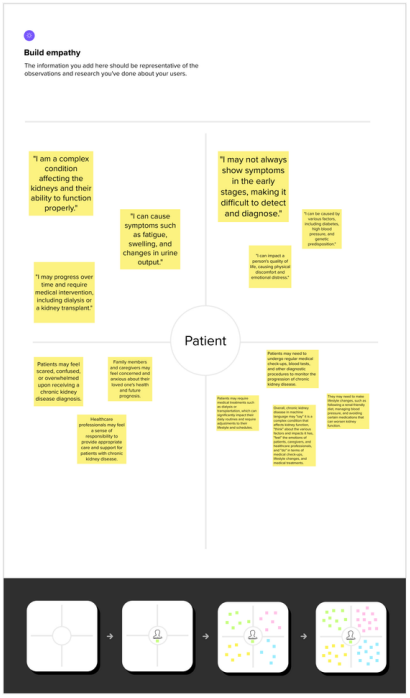
However, there are several challenges that need to be addressed before ML can be widely used in clinical practice. One challenge is the availability of high-quality data. ML algorithms require large amounts of high-quality data to train and validate models. Another challenge is the interpretability of ML models.
ML models are often seen as "black boxes," and it can be difficult to understand how they arrive at their predictions.

Conclusion:

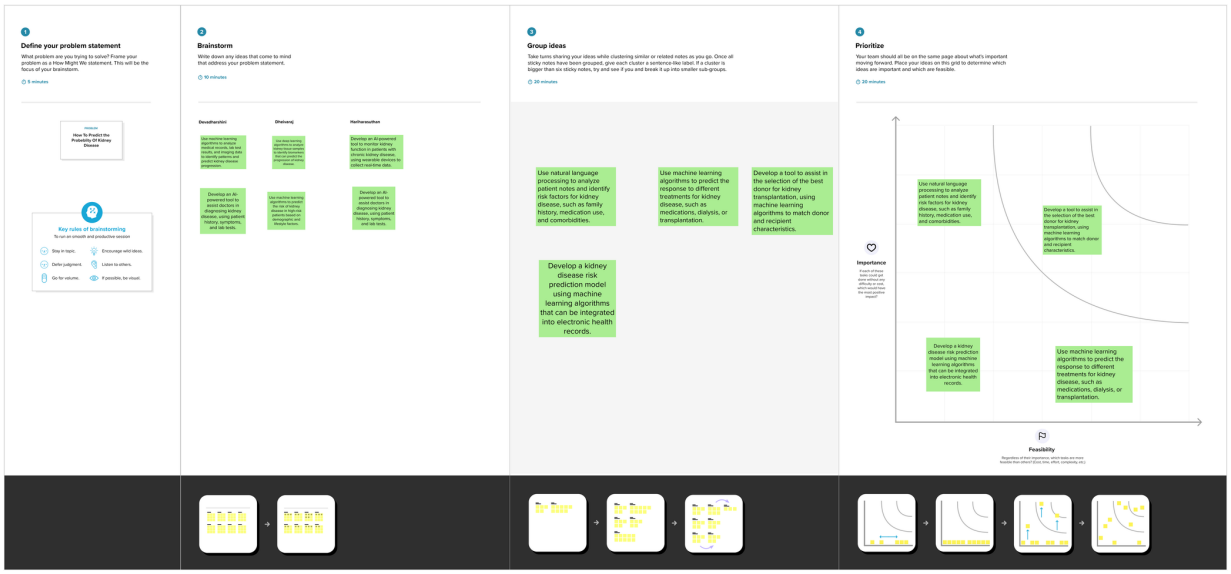
In conclusion, ML algorithms have the potential to improve the diagnosis and management of kidney disease. They can be used to identify patterns and predict disease progression, as well as to identify biomarkers for kidney disease. However, several challenges need to be addressed before ML can be

widely used in clinical practice. Further research is needed to develop more accurate and interpretable ML models for kidney disease analysis.

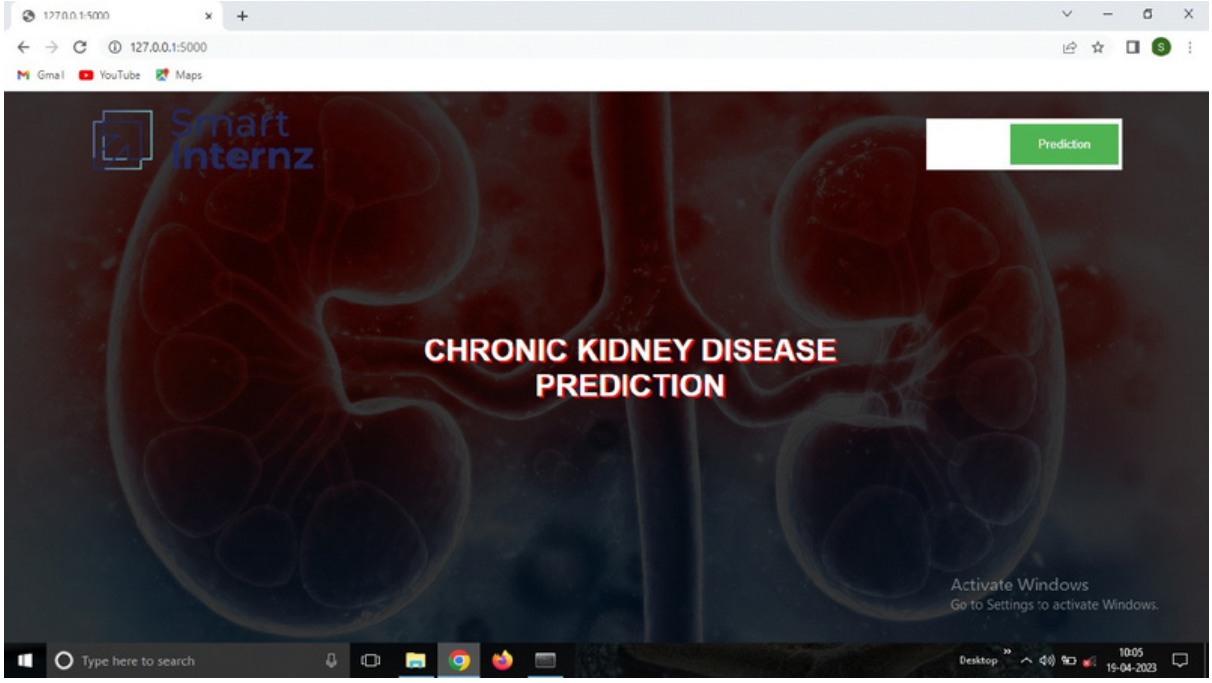
EMPATHY MAP

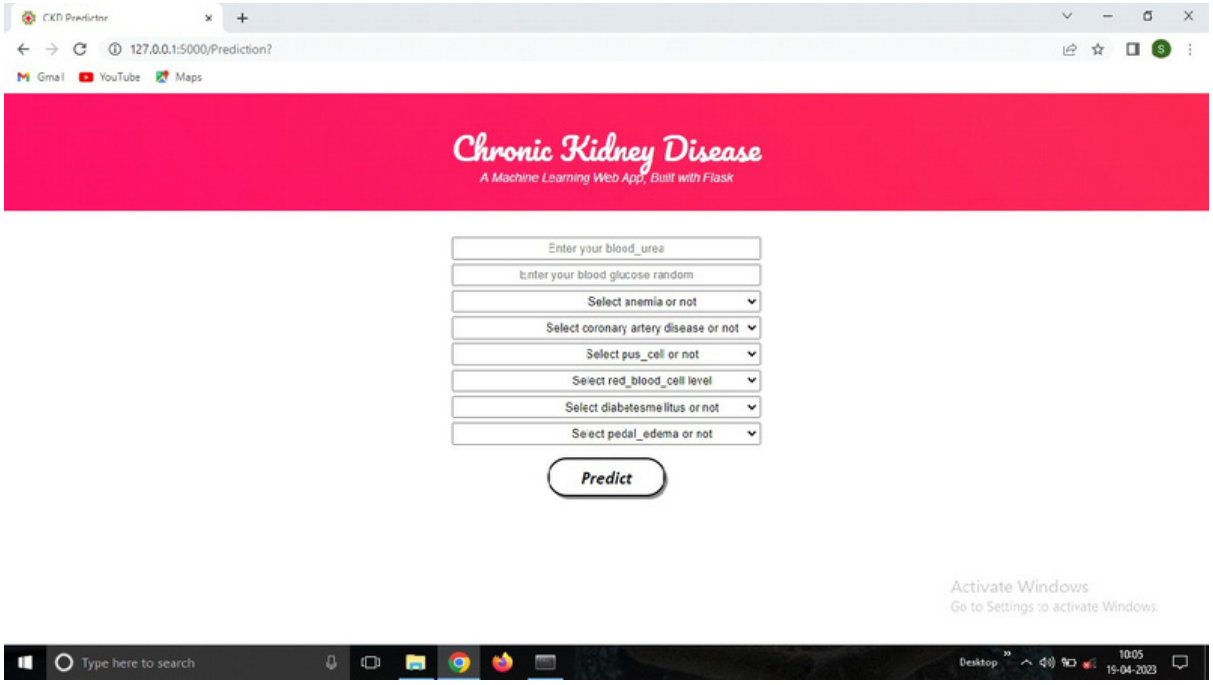

Build empathy
The information you add here should be representative of the observations and research you've done about your users.
"I am a complex condition affecting the kidneys and their ability to function properly."
"I can cause symptoms such as fatigue, swelling, and changes in urine output."
"I may progress over time and require medical intervention, including dialysis or a kidney transplant."
"I may not always show symptoms in the early stages, making it difficult to detect and diagnose."
"I can be caused by various factors, including diabetes, high blood pressure, and genetic predisposition."
"I can impact a person's quality of life, causing physical discomfort and emotional distress."
Patient
Patients may feel scared, confused, or overwhelmed upon receiving a chronic kidney disease diagnosis.
Family members and caregivers may feel concerned and anxious about their loved one's health and future prognosis.
Healthcare professionals may feel a sense of responsibility to provide appropriate care and support for patients with chronic kidney disease.
Patients may need to undergo regular medical check-ups, blood tests, and other diagnostic procedures to monitor the progression of chronic kidney disease.
Patients may require medical treatments such as dialysis or medications, which can significantly impact their daily routines and require adjustments to their lifestyle and schedules.
Overall, chronic kidney disease is a complex language the "has" it is a complex condition that affects the kidneys and "How" about the various factors and impacts it has on the patient, caregivers, and healthcare professionals, and "do" in terms of lifestyle adjustments, dietary changes, and medical treatments.
They may need to make lifestyle changes, such as adjusting their diet and diet, managing blood pressure, and avoiding certain medications that can worsen kidney function.

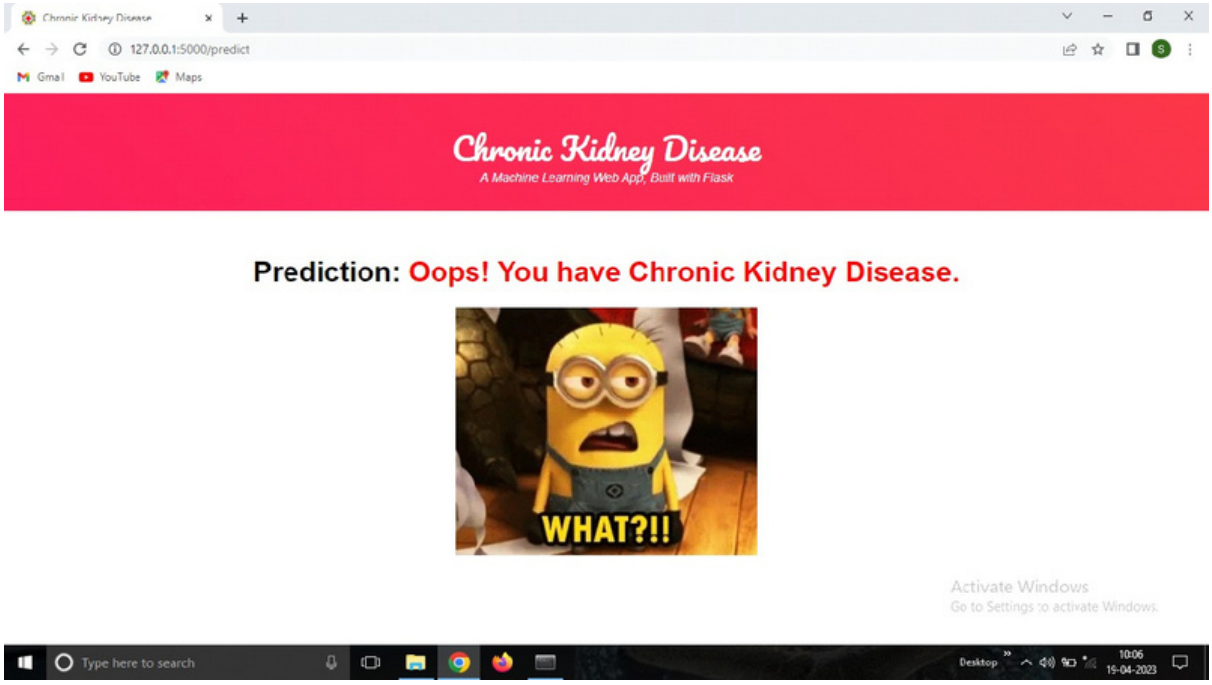
BRAIN STROM


1
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes
PROBLEM
How To Predict the Probability Of Kidney Disease
Key rules of brainstorming
To run an smooth and productive session
Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.
2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Devadharshini
Use machine learning algorithms to analyze medical records, lab test results, and other data to identify patterns and predict kidney disease progression.
Develop an AI-powered tool to assist doctors in diagnosing kidney disease using patient history, symptoms, and lab tests.
Dhalvaraj
Use deep learning algorithms to analyze kidney tissue samples and identify the stages that can provide the most accurate diagnosis of kidney disease.
Use machine learning algorithms to predict the risk of kidney disease in high-risk patients based on demographic and lifestyle factors.
Hariharasudhan
Develop an AI-powered tool to monitor kidney function in patients with chronic kidney disease. The tool would analyze data to collect real-time data.
Develop an AI-powered tool to assist doctors in diagnosing kidney disease, using patient history, symptoms, and lab tests.
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Use natural language processing to analyze patient notes and identify risk factors for kidney disease, such as family history, medication use, and comorbidities.
Use machine learning algorithms to predict the response to different treatments for kidney disease, such as medications, dialysis, or transplantation.
Develop a tool to assist in the selection of the best donor for kidney transplantation, using machine learning algorithms to match donor and recipient characteristics.
Develop a kidney disease risk prediction model using machine learning algorithms that can be integrated into electronic health records.
4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Importance
Feasibility
Use natural language processing to analyze patient notes and identify risk factors for kidney disease, such as family history, medication use, and comorbidities.
Develop a tool to assist in the selection of the best donor for kidney transplantation, using machine learning algorithms to match donor and recipient characteristics.
Develop a kidney disease risk prediction model using machine learning algorithms that can be integrated into electronic health records.
Use machine learning algorithms to predict the response to different treatments for kidney disease, such as medications, dialysis, or transplantation.

RESULT


127.0.0.1:5000
Gmail
YouTube
Maps
Smart Internz
Prediction
CHRONIC KIDNEY DISEASE PREDICTION
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:05
15-04-2023


CKD Predictor
127.0.0.1:5000/Prediction?
Gmail
YouTube
Maps
Chronic Kidney Disease
A Machine Learning Web App, Built with Flask
Enter your blood_urea
Enter your blood glucose random
Select anemia or not
Select coronary artery disease or not
Select pus_cell or not
Select red_blood_cell level
Select diabetesmellitus or not
Select pedal_edema or not
Predict
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:05
19-04-2023


Chronic Kidney Disease
127.0.0.1:5000/predict
Gmail
YouTube
Maps
Chronic Kidney Disease
A Machine Learning Web App, Built with Flask
Prediction: Oops! You have Chronic Kidney Disease.
WHAT?!!
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:06
19-04-2023

C O D E S

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail YouTube Maps

jupyter Chronic kidney disease analysis (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Importing Libaries

In [1]:
```
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
```

Loading the dataset

In [2]:
```
data=pd.read_csv("chronickidneydisease.csv") #Loading the csv data
```

Data Pre-Processing

In [3]:
```
data.head() #return you the first 5 rows values
```

Out[3]:

	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

Type here to search

Desktop 10:37 19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail YouTube Maps

jupyter Chronic kidney disease analysis (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Importing Libaries

In [1]:
```
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
```

Loading the dataset

In [2]:
```
data=pd.read_csv("chronickidneydisease.csv") #Loading the csv data
```

Data Pre-Processing

In [3]:
```
data.head() #return you the first 5 rows values
```

Out[3]:

	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

Type here to search

Desktop 10:37 19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail | YouTube | Maps

jupyter Chronic kidney disease analysis (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Importing Libaries

In [1]:
```
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
```

Loading the dataset

In [2]:
```
data=pd.read_csv("chronickidneydisease.csv") #Loading the csv data
```

Data Pre-Processing

In [3]:
```
data.head() #return you the first 5 rows values
```

Out[3]:

	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

Type here to search | Desktop | 10:37 19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail | YouTube | Maps

jupyter Chronic kidney disease analysis (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Data Pre-Processing

In [3]:
```
data.head() #return you the first 5 rows values
```

Out[3]:

	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

In [4]:
```
data.tail() #return you the last 5 rows values
```

Out[4]:

	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	9600	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

5 rows × 26 columns

In [5]:
```
data.head(10) # return the exact top 10 rows values
```

Out[5]:

	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 | Desktop | 10:37 19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail | YouTube | Maps

jupyter Chronic kidney disease analysis (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

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

In [5]:
```
data.head(10) # return the exact top 10 rows values
```

Out[5]:

	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

Dropping id column

In [6]:
```
data.drop(["id"],axis=1,inplace=True) # drop is used for dropping the column
```

Renaming the Column Names

In [7]:
```
data.columns #return all the column names
```

Out[7]:
```
Index(['age', 'bp',  sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
```


Type here to search | Desktop | 10:38 19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail YouTube Maps

jupyter Chronic kidney disease analysis (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Renaming the Column Names

In [7]:
```
data.columns #return all the column names
```

Out[7]:
```
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')
```

In [8]:
```
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'] # manually giving the name  of the columns
data.columns
```

Out[8]:
```
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')
```

Understanding the data type and its summary

In [9]:
```
data.info() #info will give you a summary of dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 25 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
```


Type here to search

Desktop 10:38 19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail YouTube Maps

jupyter Chronic kidney disease analysis (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Understanding the data type and its summary

In [9]:
```
data.info() #info will give you a summary of dataset
```

```
<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
 23  anemia                    399 non-null    object
 24  class                     400 non-null    object
dtypes: float64(11), object(14)
memory usage: 78.2+ KB
```

In [10]:
```
data.describe() # computes summary values for continous column data
```


Type here to search

Desktop 10:38 19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail YouTube Maps

jupyter Chronic kidney disease analysis (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

```
dtypes: float64(11), object(14)
memory usage: 78.2+ KB
```

In [10]:
```
data.describe() # computes summary values for continous column data
```

Out[10]:

	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

Observing Target, Categoricl and Numerical Columns

Target Column

In [11]:
```
data['class'].unique() # find the unique elements of an array
```

Out[11]:
```
array(['ckd', 'ckd\t', 'notckd'], dtype=object)
```

Rectifying the Target Column

In [12]:
```
data['class']=data['class'].replace("ckd\t","ckd") #replace is used for renaming
data['class'].unique()
```


Type here to search | Desktop | 10:38 19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail | YouTube | Maps

jupyter Chronic kidney disease analysis (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Out[13]:
```
(array([dtype('float64'), dtype('O')], dtype=object),
 array([11, 14], dtype=int64))
```

Categorical Columns

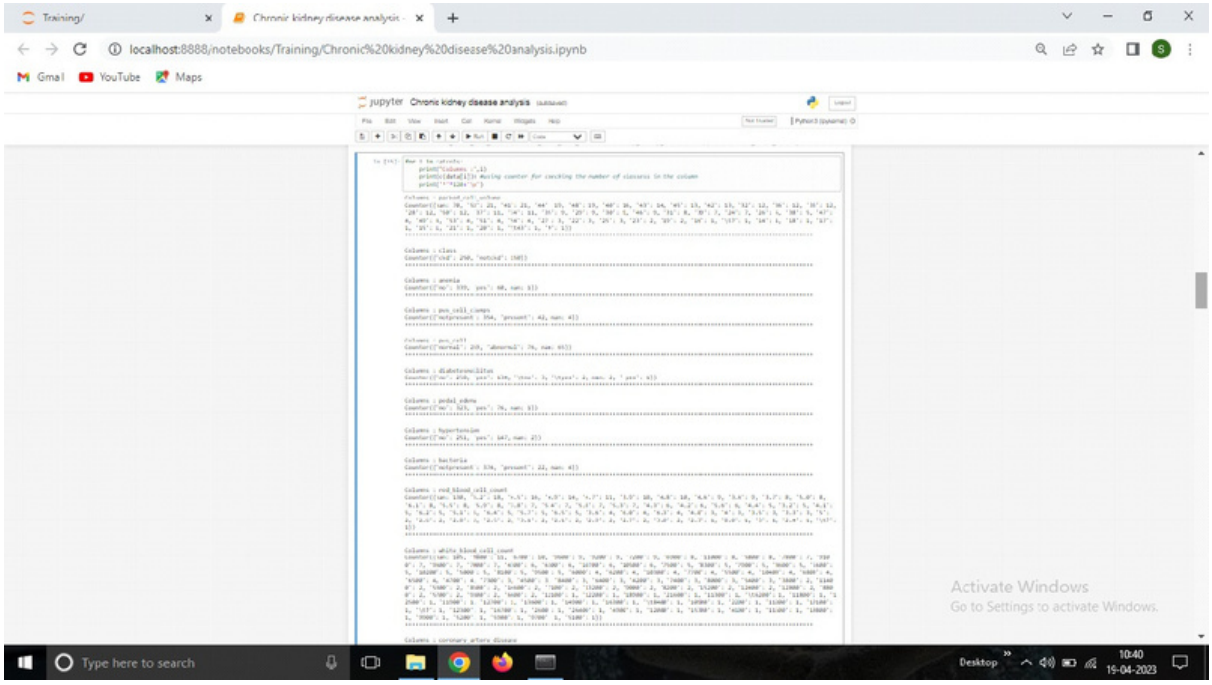
In [14]:
```
catcols=set(data.dtypes[data.dtypes=='O'].index.values) # only fetch the object type columns
print(catcols)
```

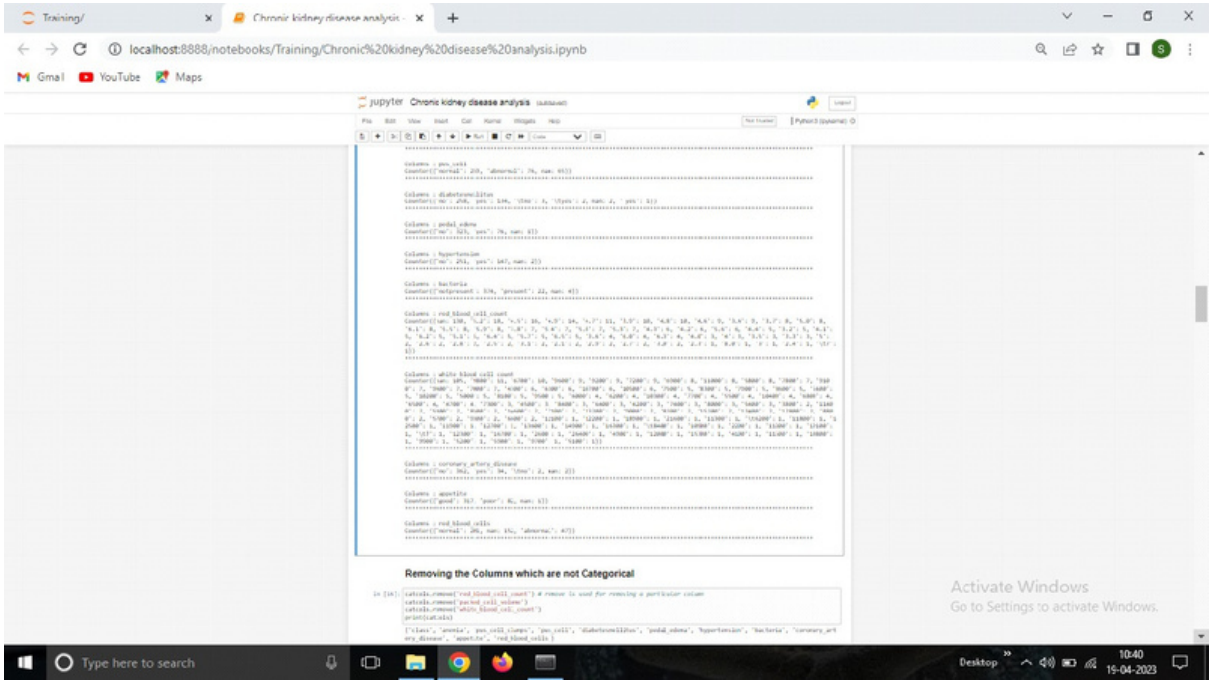
```
{'packed_cell_volume , 'class', 'anemia', 'pus_cell_clumps', 'pus_cell', 'diabetesmellitus', 'pedal_edema', 'hypertension', 'ba
cteria , 'red_blood_cell_count', 'white_blood_cell_count', 'coronary_artery_disease', 'appetite', 'red_blood_cells'}
```

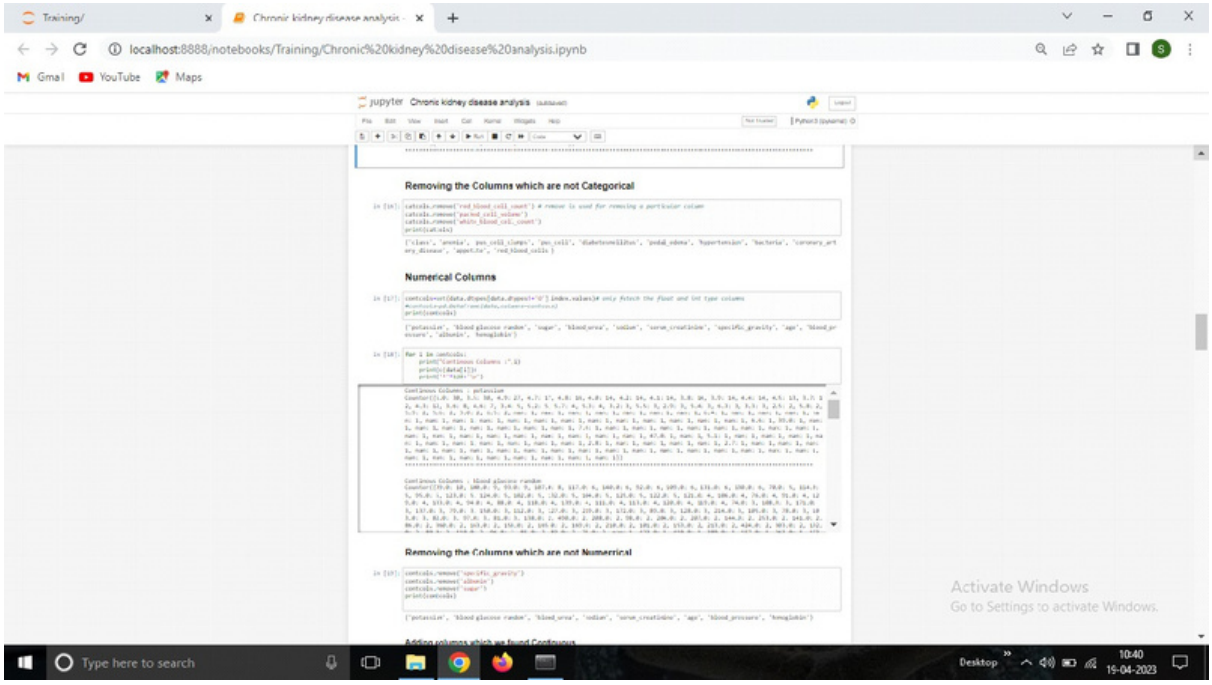
In [16]:
```
for i in catcols:
    print("Columns :",i)
    print(c(data[i])) #using counter for checking the number of classess in the column
    print('*'*120+'\n')
```

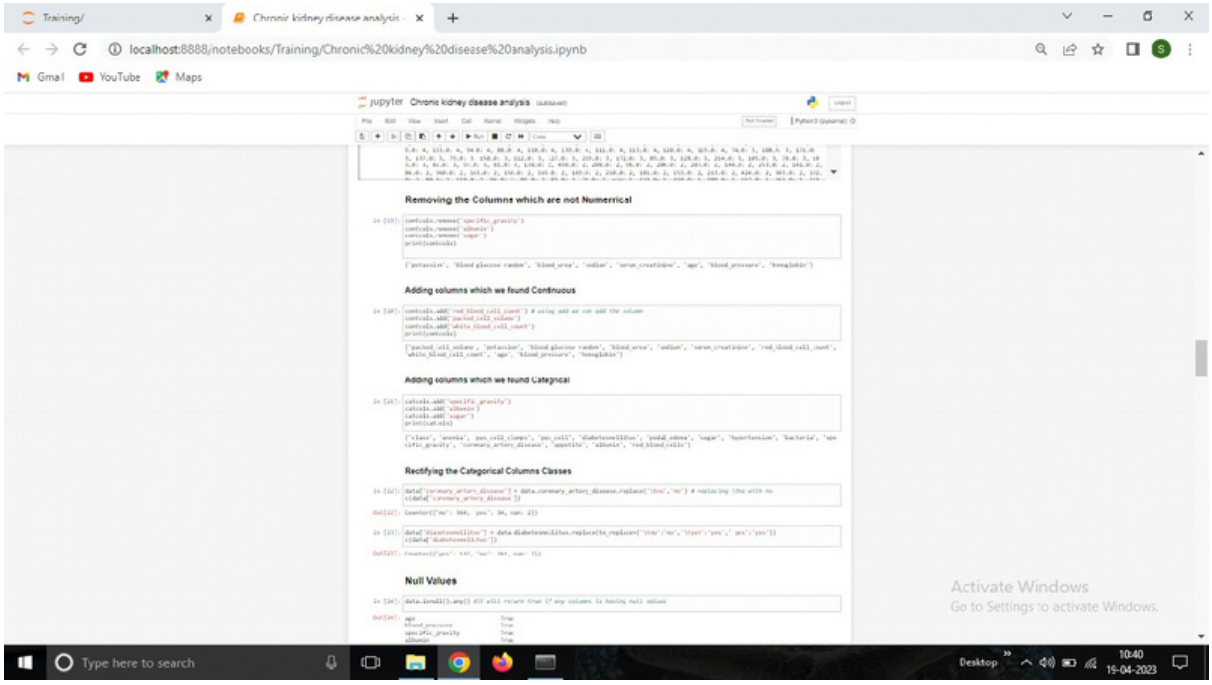
```
Columns : packed_cell_volume
Counter({nan: 70, '51': 21, '41': 21, '44': 19, '40': 19, '40': 16, '43': 14, '46': 13, '42': 13, '32': 13, '36': 11, '33': 11,
'28': 12, '50 : 12,  37': 11, '34': 11, '35': 9, '29': 9, '30': 9, '46': 9, '31': 8, '39': 7, '24': 7, '26': 6, '38': 5, '47':
4, '49 : 4, '53': 4, '51': 4, '54': 4, '27': 3, '22': 3, '25': 3, '23': 2, '19': 2, '16': 1, '\t?': 1, '14': 1, '18': 1, '17':
1, '15 : 1, '11': 1, '20': 1, '\t43': 1, '9': 1})
************************************************************************************************************************

Columns : class
Counter({'ckd : 250, 'notckd': 150})
************************************************************************************************************************

Columns : anemia
Counter({'no'  339,  yes': 60, nan: 1})
************************************************************************************************************************

Columns : pus_cell_clumps
Counter({'notprecent : 354, 'precent': 42, nan: 4})
************************************************************************************************************************

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

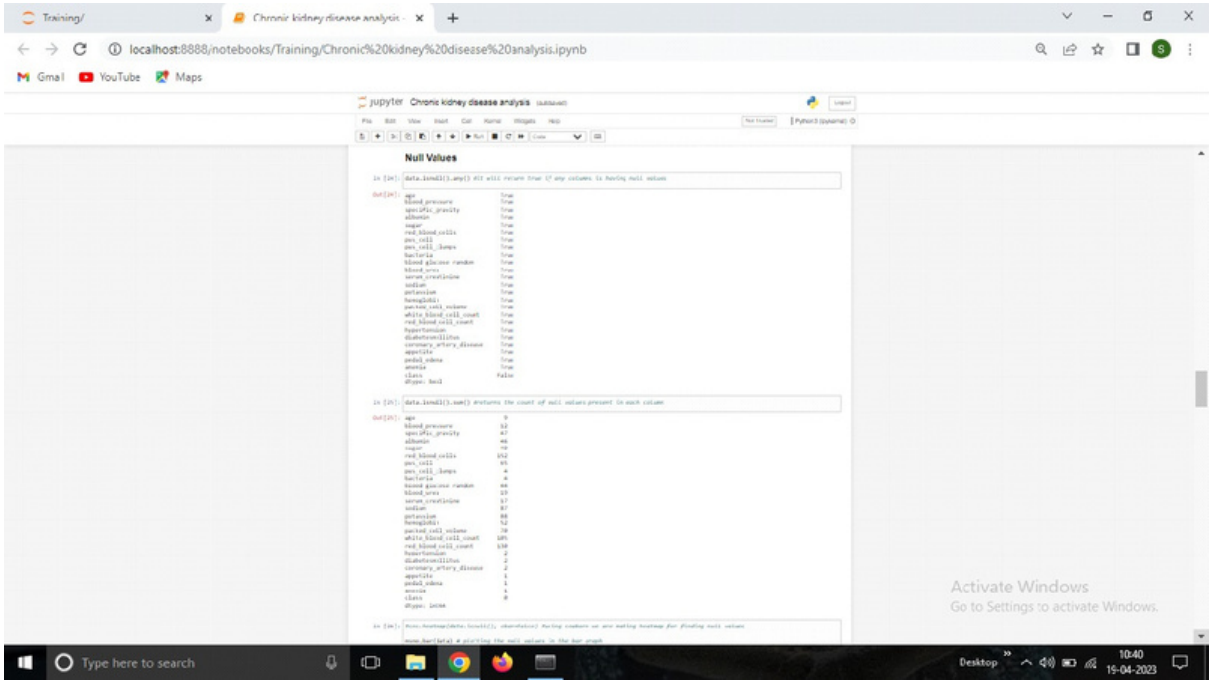

Type here to search | Desktop | 10:39 19-04-2023

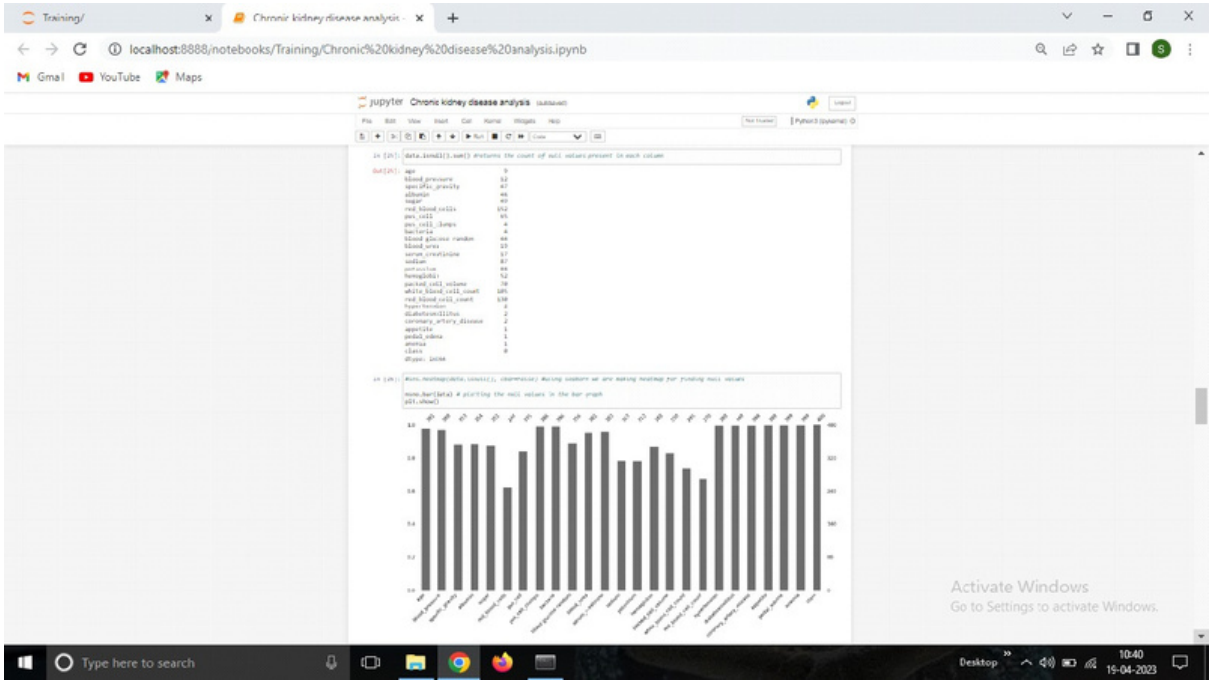

Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:40
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Removing the Columns which are not Categorical
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:40
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Removing the Columns which are not Categorical
Numerical Columns
Removing the Columns which are not Numerrical
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:40
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Removing the Columns which are not Numerrical
Adding columns which we found Continuous
Adding columns which we found Categrical
Rectifying the Categorical Columns Classes
Null Values
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:40
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Null Values
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:40
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:40
19-04-2023

Training/ | Chronic kidney disease analysis

localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb

Gmail | YouTube | Maps

jupyter Chronic kidney disease analysis

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Converting the column into numeric type

In [27]:
```
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')
```

Handling Continous/numerical columns Null values

In [28]:
```
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)
```

In [29]:
```
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)
```

In [30]:
```
data.isnull().sum().sum()
```

Out[30]: 0

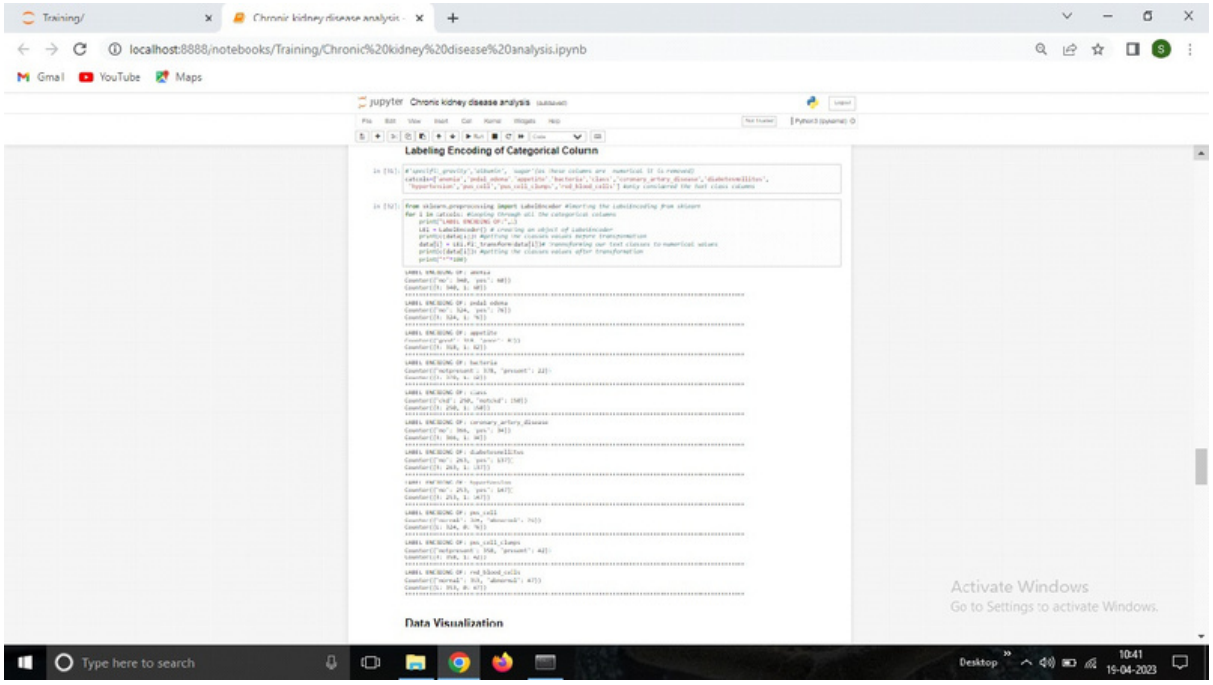
Labeling Encoding of Categorical Column

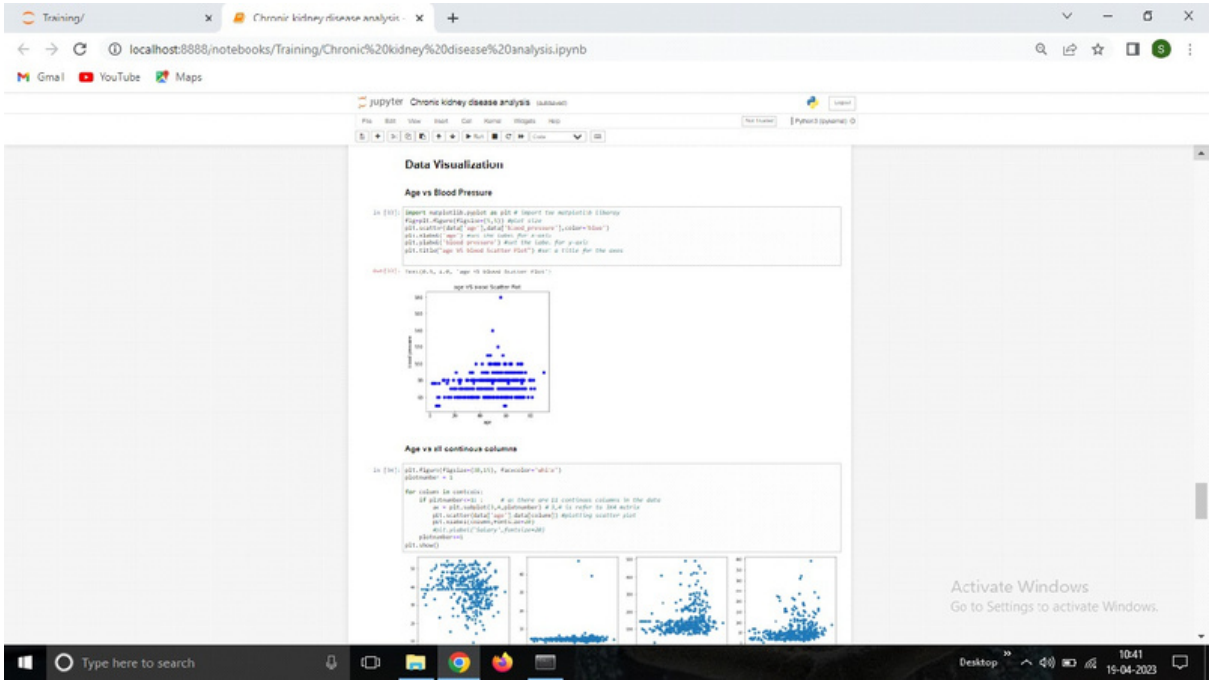
In [31]:
```
#'specific gravity','albumin', 'sugar'(as these columns are numerical it is removed)
catcols=['anemia','pedal edema','appetite','bacteria','class','coronary artery disease','diabetesmellitus',
 'hypertension','pus cell','pus cell clumps','red blood cells'] #only considered the bot class columns
```

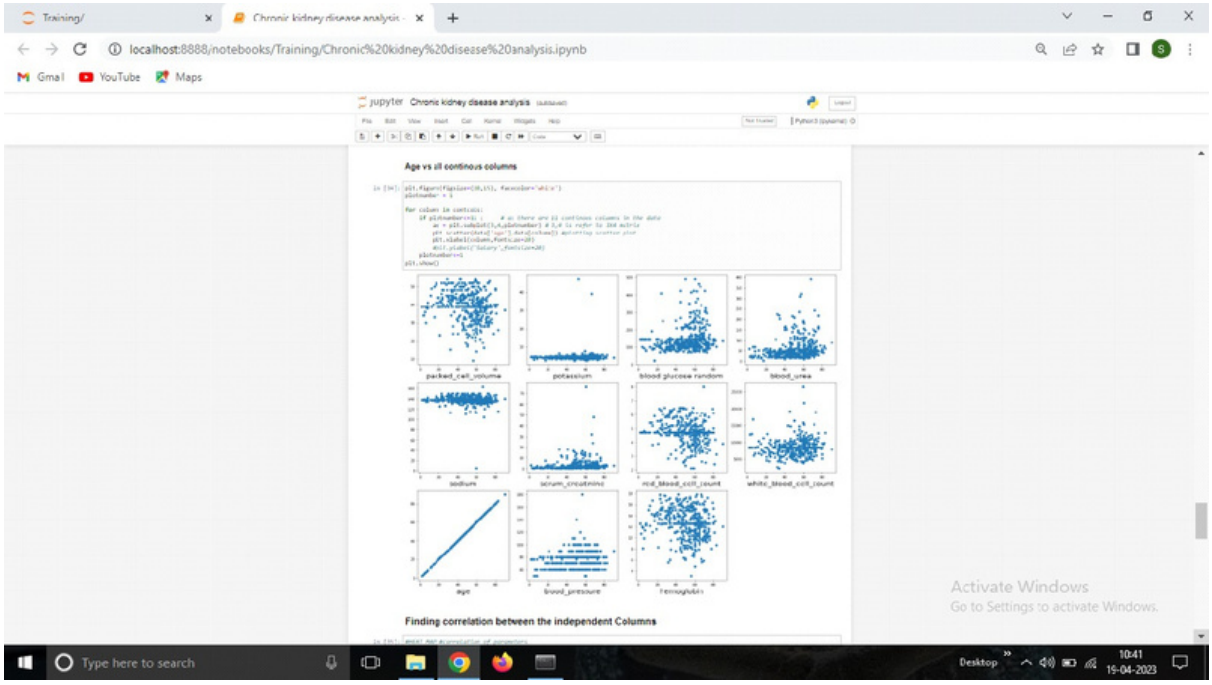
In [32]:
```
from sklearn.preprocessing import LabelEncoder #importing the LabelEncoding from sklearn
for i in catcols: #looping through all the categorical columns
    print("\nBEG ENCODING OF:",i)
    LE1 = LabelEncoder() # creating an object of LabelEncoder
    print(data[i]) #printing the classes values before transformation
    data[i] = LE1.fit_transform(data[i])# transforming our text classes to numerical values
    print(data[i]) #printing the classes values after transformation
    print("""""END)
```

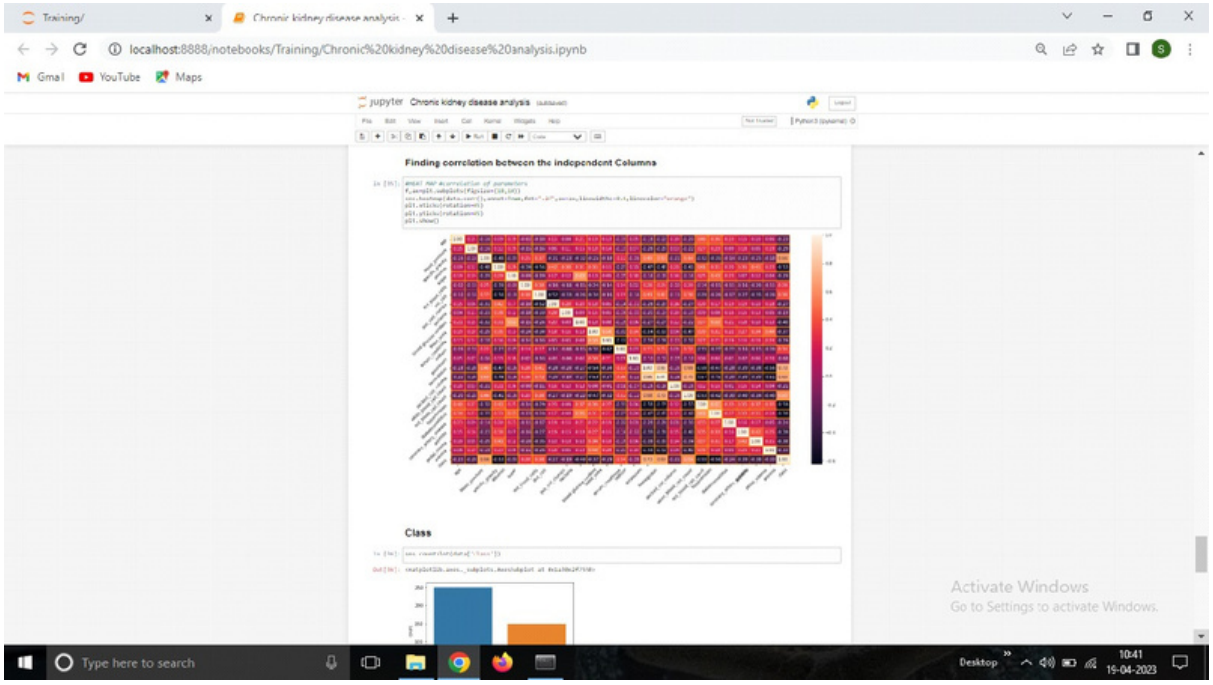
```
[anemia: ckd/notckd (0 - ckd)]
Counter({'no': 340, yes': 60})
Counter({0: 340, 1: 60})
```

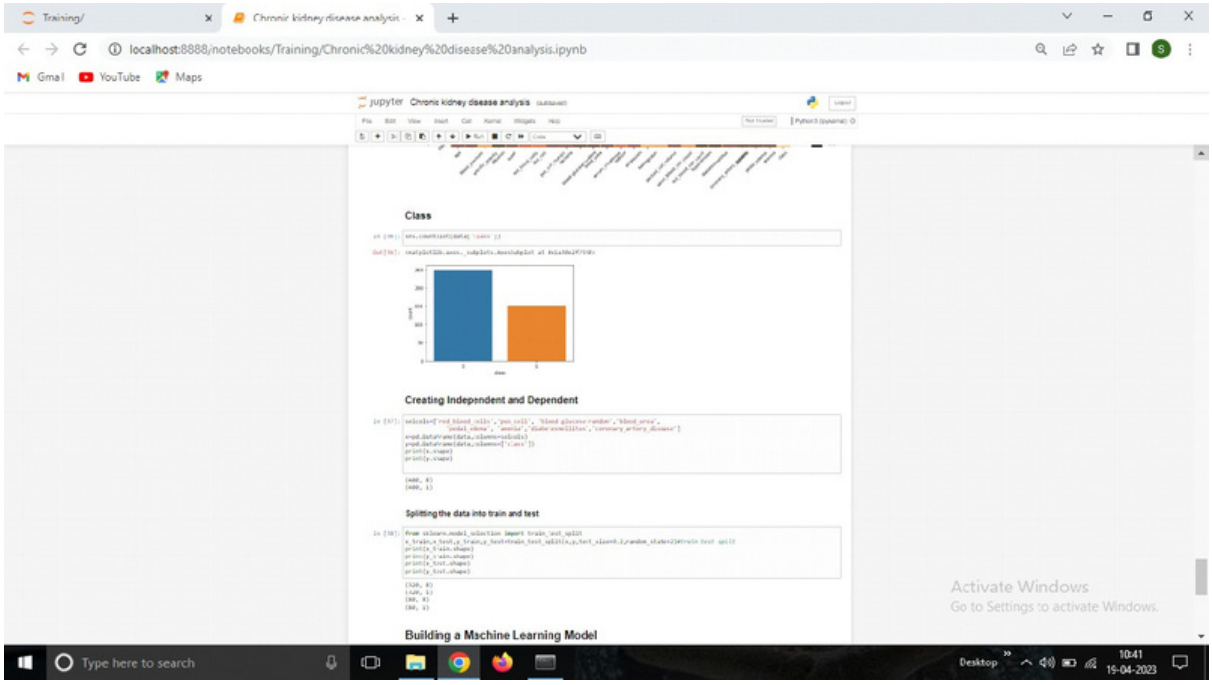

Type here to search | Desktop | 10:41 | 19-04-2023

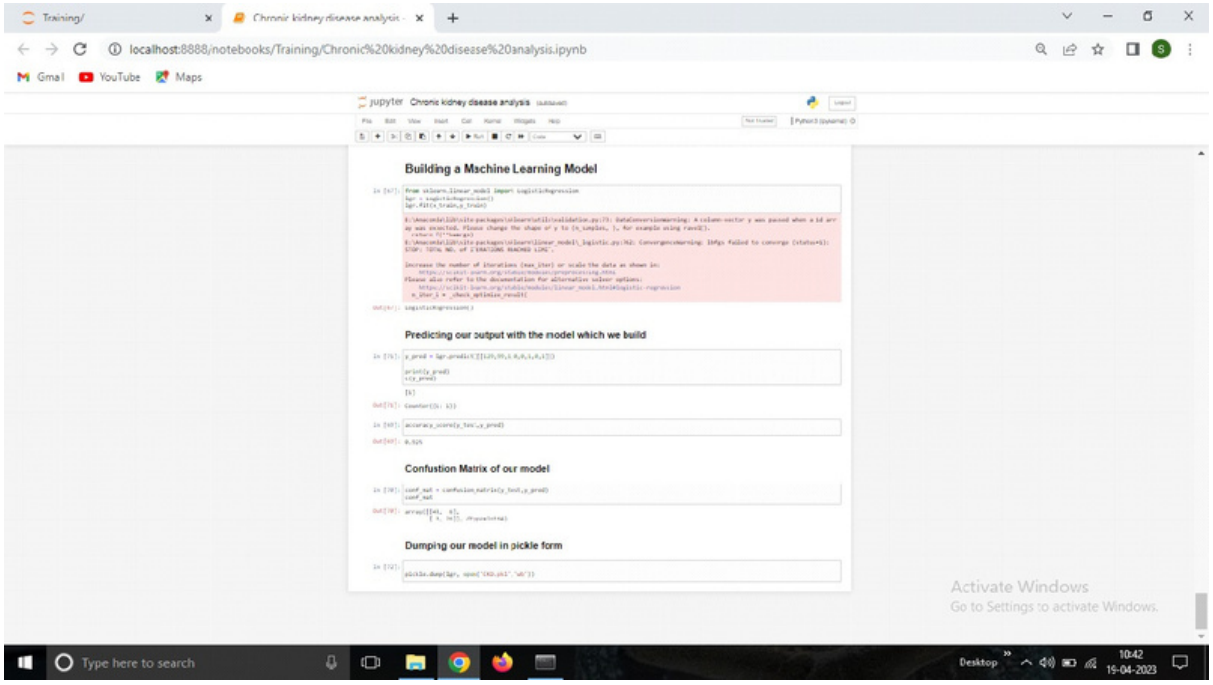

Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Labeling Encoding of Categorical Column
Data Visualization
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:41
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Data Visualization
Age vs Blood Pressure
Age vs all continous columns
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:41
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Age vs all continous columns
packed_cell_volume
potassium
blood_glucose_random
blood_urea
sodium
serum_creatinine
red_blood_cell_count
white_blood_cell_count
age
blood_pressure
hemoglobin
Finding correlation between the independent Columns
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:41
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Finding correlation between the independent Columns
Class
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:41
19-04-2023


Training/
Chronic kidney disease analysis
localhost:8888/notebooks/Training/Chronic%20kidney%20disease%20analysis.ipynb
Gmail
YouTube
Maps
jupyter Chronic kidney disease analysis
Class
Creating Independent and Dependent
Splitting the data into train and test
Building a Machine Learning Model
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
10:41
19-04-2023



SOURCE CODE

importing the necessary dependencies

```
import numpy as np
import pandas as pd
from flask import Flask, request,
render_template import pickle



app = Flask(__name__) # initializing a flask app
model = pickle.load(open('CKD.pkl', 'rb')) #loading the
model


@app.route('/')# route to display the home page
def home():
   return render_template('home.html') #rendering the
home page
@app.route('/Prediction',methods=['POST','GET'])
def prediction():
return render_template('indexnew.html')
@app.route('/Home',methods=['POST','GET'])
def my_home():
return render_template('home.html')
```

```
@app.route('/predict',methods=['POST'])# route to show the predictions in a web UI
def predict():

    #reading the inputs given by the user
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    features_name = ['blood_urea', 'blood glucose random', 'anemia',
    'coronary_artery_disease', 'pus_cell', 'red_blood_cells', 'diabetesmellitus', 'pedal_edema']

    df = pd.DataFrame(features_value, columns=features_name)

    # predictions using the loaded model
    file output = model.predict(df)

    # showing the prediction results in a UI# showing the prediction results in a UI
```

```
    return render_template('result.html',
prediction_text=output)


if __name__ == '__main__':
# running the app
app.run(debug=True)
```