

EARLY CHRONIC KIDNEY DISEASE PREDICTION USING IBM WATSON STUDIO

INDEX

1. INTRODUCTION
 - 1.1 OVERVIEW
 - 1.2 PURPOSE
2. LITERATURE SURVEY
 - 2.1 Existing Problem
 - 2.2 Proposed solution
3. THEORITICAL ANALYSIS
 - 3.1 Block Diagram
 - 3.2 Hardware/Software Designing
4. EXPERIMENTAL INVESTIGATION
5. FLOWCHART
6. RESULT
7. ADVANTAGES AND DISADVANTAGES
8. APPLICATIONS
9. CONCLUSION
- 10.FUTURE SCOPE

1.INTRODUCTION

1.1 OVERVIEW

Chronic kidney disease (CKD) is a significant public health problem worldwide, especially for low and medium income countries. Chronic kidney disease (CKD) means that the kidney does not work as expected and cannot correctly filter blood. About 10% of the population worldwide suffers from (CKD), and millions die each year because they cannot get affordable treatment, with the number increasing in the elderly. According to the Global Burden Disease 2010 study conducted by the International Society of Nephrology, chronic kidney disease (CKD) has been raised as an important cause of mortality worldwide with the number of deaths increasing by 82.3% in the last two decades. Also, the number of patients reaching end-stage renal disease (ESRD) is increasing, which requires kidney transplantation or dialysis to save patients' lives. CKD, in its early stages, has no symptoms; testing may be the only way to find out if the patient has kidney disease. Early detection of CKD in its initial stages can help the patient get effective treatment and then prohibit the progression to ESRD. It is argued that every year, a person that has one of the CKD risk factors, such as a family history of kidney failure, hypertension, or diabetes, get checked. The sooner they know about having this disease, the sooner they can get treatment. To raise awareness and to encourage those who are most susceptible to the disease to perform the tests periodically, we hope that the disease can be detected with the least possible tests and at low cost. So, the objective of this research is to provide an effective model to predict the CKD by least number of predictors.

1.2 PURPOSE

Chronic kidney disease(CKD) or Chronic renal disease has become a major issue with a study growth rate. A person can only survive without kidneys from an average time of 18 days,which makes a huge demand for a kidney transplant and Dialysis.it is important to have efficient methods for early prediction of CKD prediction.This work proposes a workflow to predict CKD status based on Clinical data, incorporating data preprocessing, a missing value handling method with collaborative filtering and attribute selection. Out of the 11 machine learning methods considered, we used Random Forest Classifier are shown to result in the highest accuracy and minimal bias to the attributes.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

There is a mounting epidemic of CKD and end-stage renal disease (ESRD) in the US. As of 2002, between 4 million and 20 million Americans were affected with CKD, and about 300 000 were defined as having ESRD or requiring renal replacement therapy. It is estimated that by 2015 the number of patients with ESRD will be 712 000. The total number of expected patients receiving dialysis by 2010 will reach 560 000 resulting in an annual Medicare spending of \$28.3 billion by 2010. As of 2007, the total Medicare cost for CKD reached \$57.5 billion.

Patients with CKD are at risk for not only progression to ESRD but also increased cardiovascular morbidity and mortality. The key to preventing either of these two outcomes is recognition of the earliest stages of kidney disease and initiation of a targeted and aggressive management plan. The National Kidney Foundation provides evidence-based clinical practice guidelines for all stages of CKD and related complications, which include a recommendation for referral to a nephrologist if CKD is sufficiently advanced. The importance of a timely referral to a nephrologist is evident in multiple studies that have shown an association with late nephrology referral and poor outcomes when starting hemodialysis. Patients with unrecognized CKD may be referred by their provider at a later stage than a patient with recognized CKD.

Only if providers recognize that their patients have CKD will the appropriate targeted management be initiated. Several investigators have demonstrated considerable under-recognition by primary care practitioners. De Lusignan and colleagues demonstrated that less than 4% of patients with CKD had been coded as having renal disease. Studies conducted by manual chart review (bypassing the known *International Classification of Diseases* (ICD)-9 coding sensitivity issues) demonstrated that over three-quarters of patients with CKD were not recognized as having CKD.

A first step in creating a tool to prompt early recognition of CKD is to determine if the provider has recognized the patient's CKD. The tool could search for appropriate documentation of CKD in the patient's notes as a proxy for recognition.

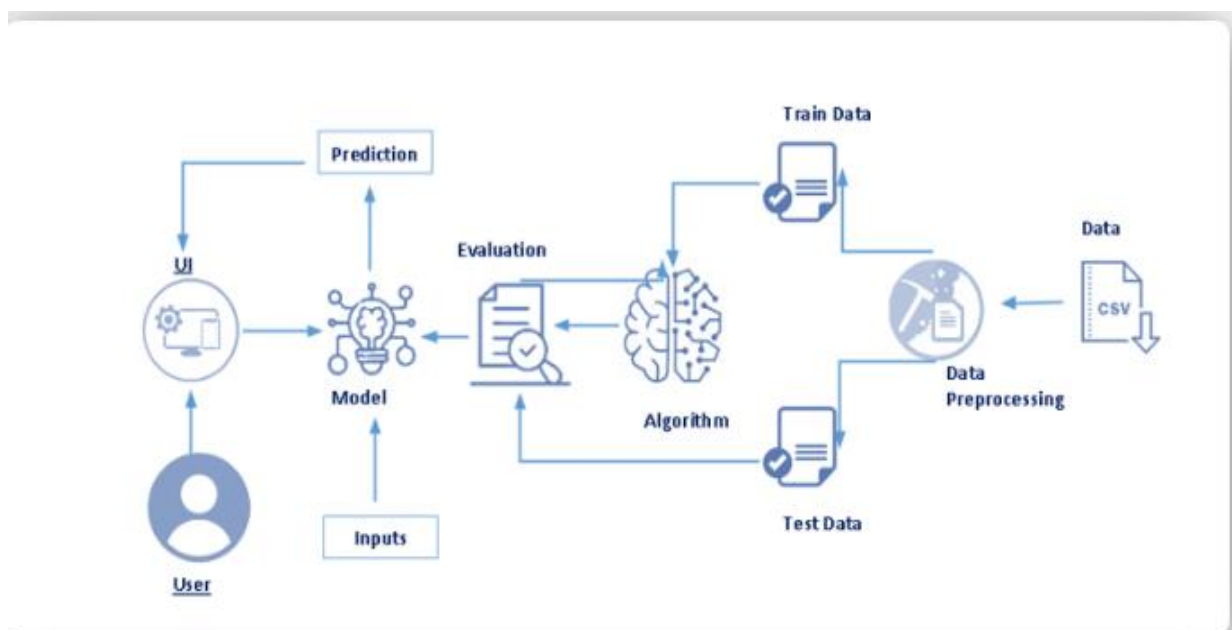
2.2 PROPOSED SOLUTION

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 and we make use of such information to build a machine learning model that predicts Chronic Kidney Disease . Out of the 11 machine learning methods considered, we used Random Forest Classifier are shown to result in the highest accuracy and minimal bias to the attributes Model is based on Random Forest Classifier , and it obtains the weight of each predictor and a bias. If the sum of the effects of all predictors exceeds a threshold, the category of the sample will be classified as ckd or notckd.

3. THEOROTICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE / SOFTWARE DESIGNING

SOFTWARE DESIGNING:

1. Jupyter Notebook Environment
2. Spyder
3. Machine Learning Algorithms
4. Python (pandas, numpy, matplotlib, seaborn, sklearn)
5. HTML

6. Flask

We developed this Chronic Kidney Disease prediction by using the Python language which is a interpreted and high level programming language and using the Machine Learning algorithms. for coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For creating a user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a micro frame work because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in the functions of the Flask and HTML.

HARDWARE REQUIRMENTS

- Processor : Intel Core i3
- Hard Disk Space : Min 100 GB
- Ram : 8 GB
- Display : 14.1 “Color Monitor(LCD, CRT or LED)

4. EXPERIMENTAL INVESTIGATIONS

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms or Regression algorithms.

Example: 1. Linear Regression.

2. Logistic Regression.

3. Random Forest Regression / Classification.

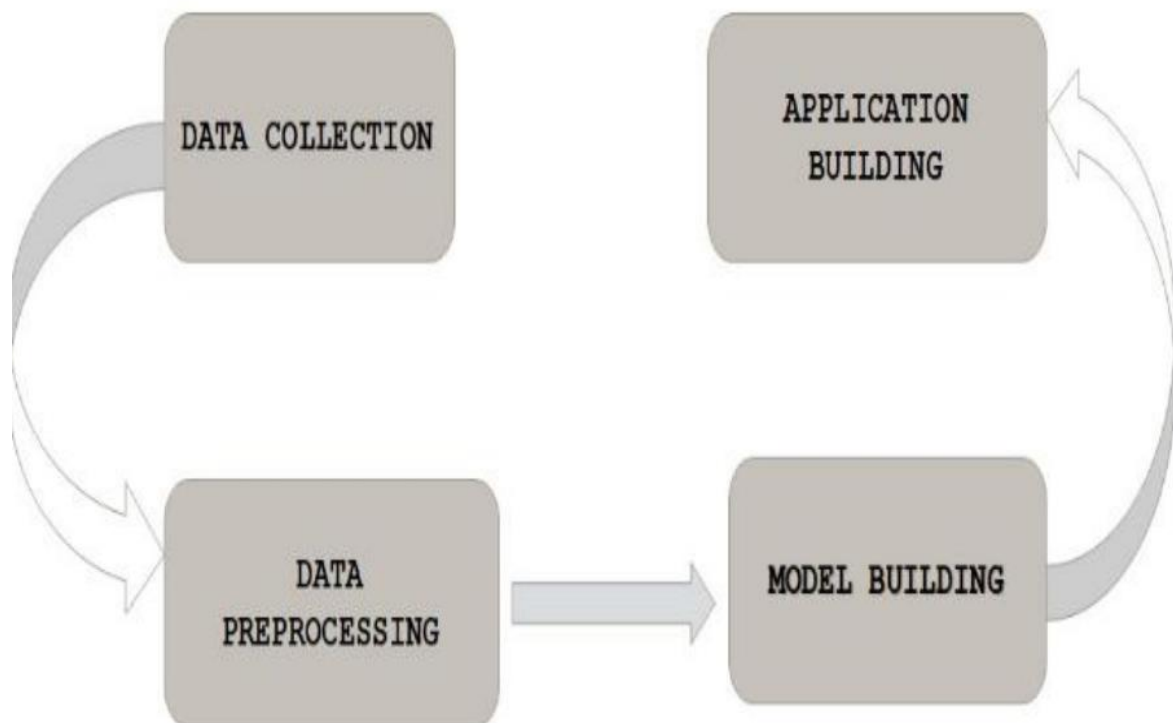
4. Decision Tree Regression / Classification.

Here we used Random Forest Classification, Random Forest grows multiple decision trees which are merged together for a more accurate prediction. The logic behind the Random Forest model is that multiple uncorrelated models (the individual decision trees) perform much better as a group than they do alone. When using Random Forest for classification, each tree gives a classification or a “vote.” The forest chooses the classification with the majority of the “votes.” When using Random Forest for regression, the forest picks the average of the outputs of all trees. The key here lies in the fact that there is low (or no)

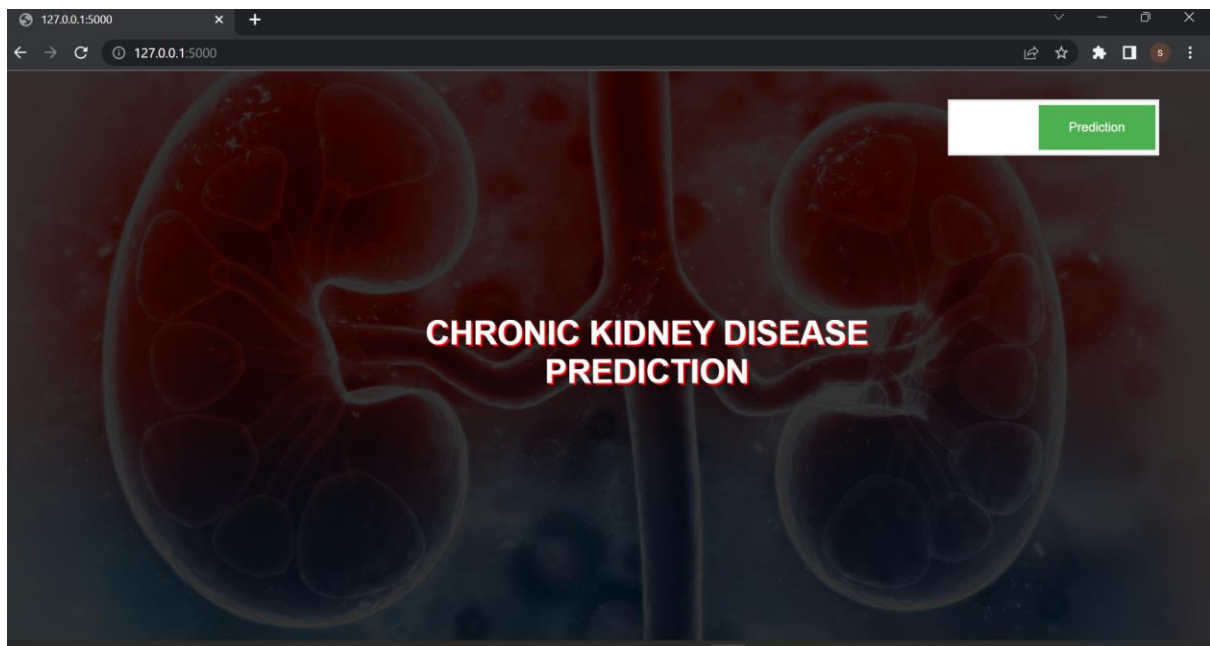
correlation between the individual models—that is, between the decision trees that make up the larger Random Forest model. While individual decision trees may produce errors, the majority of the group will be correct, thus moving the overall outcome in the right direction. Once the model is trained, it's ready to make predictions.

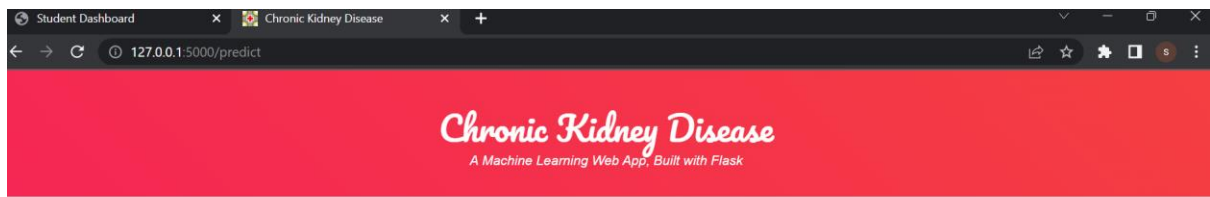
Finally, there is a need to check to see how well our model is performing on the test data. There are many evaluation techniques are there. For this, we evaluate the accuracy score produced by the model. Confusion Matrix for the model is been used.

5. FLOWCHART

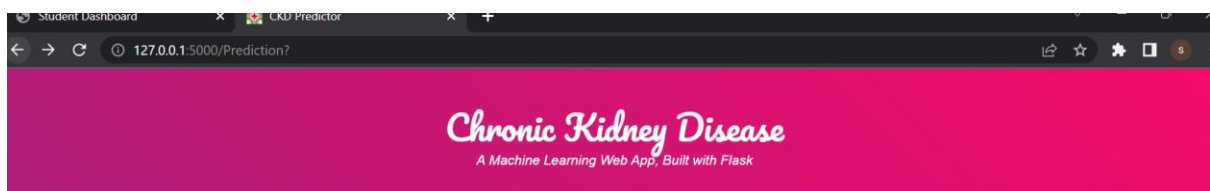


6.RESULT

A screenshot of a web browser window showing the input form for the "Chronic Kidney Disease" prediction application. The browser tabs include "Student Dashboard" and "CKD Predictor". The address bar shows "127.0.0.1:5000/Prediction?". The page has a pink header with the title "Chronic Kidney Disease" and the subtitle "A Machine Learning Web App, Built with Flask". Below the header, there is a form with eight input fields arranged vertically. The first three are text inputs containing the values "40", "18", and "140". The remaining five are dropdown menus with the following selected values: "NO", "YES", "normal", "normal", and "YES". At the bottom of the form is a rounded rectangular button labeled "Predict".

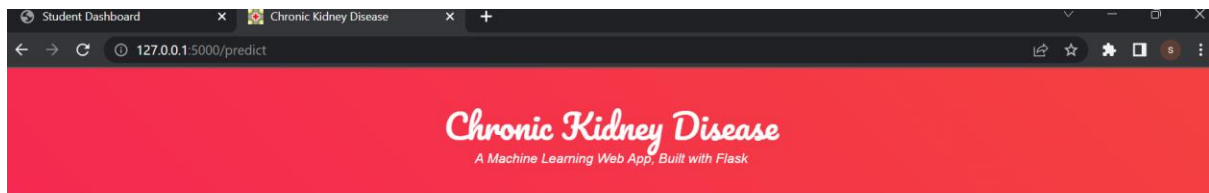


Prediction: Oops! You have Chronic Kidney Disease.



45
40
129
NO
NO
abnormal
normal
NO
NO

Predict



Prediction: Great! You DON'T have Chronic Kidney Disease



7. ADVANTAGES AND DISADVANTAGES

Advantages :

- Random Forest Classifier give the accurate result of the prediction up to 96% which is the algorithm we used for prediction
- Random Forest algorithm is less prone to overfitting than Decision Tree and other algorithms
- Random Forest algorithm outputs the importance of features which is a very useful

Disadvantages :

- Random Forest algorithm may change considerably by a small change in the data.
- Random Forest algorithm computations may go far more complex compared to other algorithms.

8. APPLICATIONS

- Deployment for the project can be in the form of an interactive Web Based Platform, where users can enter their details as per the model requirement and get the predictions as a result.

- This machine learning model is applicable in the field of medical science.
- In healthcare Random Forest can be used to analyze a patient's medical history to identify diseases. Pharmaceutical scientists use Random Forest to identify the correct combination of components in a medication or predict drug sensitivity

9.CONCLUSION

In this project, we have established the application to predict the outcome of whether chronic kidney disease or not based on the attributes of the applicant and ,several machine learning models like Random Forest Classifier algorithm can be used. Finally, this can be integrated to a web application.

10. FUTURE SCOPE

In further Random Forest Classifier algorithm can be applied on other data sets available for kidney disease prediction further investigate its accuracy. A rigorous analysis of other machine learning algorithms other than these six can also be done in future to investigate the power of machine learning algorithms for visa status prediction. In further study, we will try to conduct experiments on larger data sets or try to tune the model so as to achieve the state -of-art performance of the model and a great UI support system making it complete web application model.

11. BIBILOGRAPHY

- <https://randerson112358.medium.com/chronic-kidney-disease-prediction-detection-using-machine-learning-29cc7e3eba96>
- <https://github.com/smartinternz02/SI-GuidedProject-89764-1658129570>

APPENDIX

Python source code

Home Page - Select or create a notebook | chronickidney - Jupyter Notebook | +

localhost:8888/notebooks/chronickidney.ipynb

jupyter chronickidney Last Checkpoint: Last Tuesday at 1:32 PM (autosaved) Logout

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

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter as c
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.ensemble import RandomForestClassifier
import pickle
```

In [2]: `data=pd.read_csv("D:\chronickidney\dataset\chronickidneydisease.csv")`

In [3]: `data.head()`

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 x 26 columns

Home Page - Select or create a notebook | chronickidney - Jupyter Notebook | +

localhost:8888/notebooks/chronickidney.ipynb

jupyter chronickidney Last Checkpoint: Last Tuesday at 1:32 PM (autosaved) Logout

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

In [4]: `data.columns.values`

Out[4]: `array(['id', 'age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu', 'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad', 'appet', 'pe', 'ane', 'classification'], dtype=object)`

In [5]: `data.shape`

Out[5]: `(400, 26)`

In [6]: `data.drop(['id'],axis=1,inplace=True)`

In [7]: `data.head()`

Out[7]:

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	35	7300	4.6	no	no	no	good	no	no	ckd

5 rows x 25 columns

Home Page - Select or create a notebook | chronickidney - Jupyter Notebook | +

localhost:8888/notebooks/chronickidney.ipynb

jupyter chronickidney Last Checkpoint: Last Tuesday at 1:32 PM (autosaved) Logout

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

dtype='object')

```
In [11]: data.head()
```

Out[11]:

	age	bloodpressure	specific_gravity	albumin	sugar	red_blood_cells	pus_cell	pus_cell_clumps	bacteria	blood_glucose_random	...	packed_cell_volume
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	44
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	38
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	31
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	32
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	35

5 rows × 25 columns

```
In [12]: 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 bloodpressure 388 non-null float64
2 specific_gravity 353 non-null float64
3 albumin 354 non-null float64

Home Page - Select or create a notebook | chronickidney - Jupyter Notebook | +

localhost:8888/notebooks/chronickidney.ipynb

jupyter chronickidney Last Checkpoint: Last Tuesday at 1:32 PM (autosaved) Logout

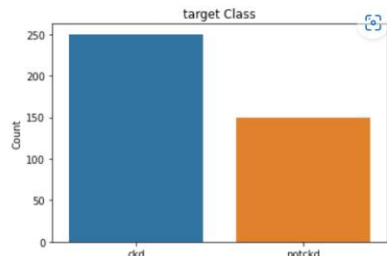
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [14]: data['class']=data['class'].replace("ckd\t", "ckd")  
data['class'].unique()
```

Out[14]: array(['ckd', 'notckd'], dtype=object)

```
In [15]: sns.countplot(x='class', data=data)  
plt.xlabel("class")  
plt.ylabel("Count")  
plt.title("target Class")
```

Out[15]: Text(0.5, 1.0, 'target Class')



class	Count
ckd	240
notckd	150

Home Page - Select or create a notebook | chronickidney - Jupyter Notebook | +

localhost:8888/notebooks/chronickidney.ipynb

Jupyter chronickidney Last Checkpoint: Last Tuesday at 1:32 PM (autosaved) Logout

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

```
rf = rf.fit(x_train,y_train)
```

C:\Users\SURYAMOL\AppData\Local\Temp\ipykernel_89872\2243325452.py:2: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```
rf = rf.fit(x_train,y_train)
```

In [244]: y_pred = rf.predict(x_test)

In [245]: print('Accuracy:', accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

Accuracy: 0.9625

```
[[52  2]
 [ 1 25]]
```

	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

Home Page - Select or create a notebook | chronickidney - Jupyter Notebook | +

localhost:8888/notebooks/chronickidney.ipynb

Jupyter chronickidney Last Checkpoint: Last Tuesday at 1:32 PM (autosaved) Logout

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

```
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')
```

In [31]: data.columns

Out[31]: Index(['age', 'bloodpressure', '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'],
dtypes='object')

In [32]: data['age'].fillna(data['age'].mode()[0],inplace=True)
data['bloodpressure'].fillna(data['bloodpressure'].mean(),inplace=True)
data['specific_gravity'].fillna(data['specific_gravity'].mode()[0],inplace=True)
data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)
data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)
data['red_blood_cells'].fillna(data['red_blood_cells'].mode()[0],inplace=True)
data['pus_cell'].fillna(data['pus_cell'].mode()[0],inplace=True)
data['pus_cell_clumps'].fillna(data['pus_cell_clumps'].mode()[0],inplace=True)
data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)
data['blood_glucose_random'].fillna(data['blood_glucose_random'].mean(),inplace=True)
data['blood_urea'].fillna(data['blood_urea'].mean(),inplace=True)
data['serum_creatinine'].fillna(data['serum_creatinine'].mean(),inplace=True)
data['sodium'].fillna(data['sodium'].mean(),inplace=True)
data['potassium'].fillna(data['potassium'].mean(),inplace=True)

Home Page - Select or create a notebook | chronickidney - Jupyter Notebook | +

localhost:8888/notebooks/chronickidney.ipynb

jupyter chronickidney Last Checkpoint: Last Tuesday at 1:32 PM (autosaved) Logout

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

anemia
class
dtype: int64

```
In [34]: for i in catcols :  
         le=LabelEncoder()  
         data[i]=le.fit_transform(data[i])
```

```
In [35]: data.head()
```

```
Out[35]:
```

	age	bloodpressure	specific_gravity	albumin	sugar	red_blood_cells	pus_cell	pus_cell_clumps	bacteria	blood_glucose_random	...	packed_cell_volume
0	48.0	80.0	3	1	0	1	1	0	0	121.000000	...	32
1	7.0	50.0	3	4	0	1	1	0	0	148.036517	...	25
2	62.0	80.0	1	2	3	1	1	0	0	423.000000	...	18
3	48.0	70.0	0	4	0	1	0	1	0	117.000000	...	19
4	51.0	80.0	1	2	0	1	1	0	0	106.000000	...	22

5 rows x 25 columns

```
In [36]: data.isnull().sum()
```

```
Out[36]: age 0
```

Home Page - Select or create a notebook | chronickidney - Jupyter Notebook | +

localhost:8888/notebooks/chronickidney.ipynb

jupyter chronickidney Last Checkpoint: Last Tuesday at 1:32 PM (autosaved) Logout

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

```
anemia  
diabetesmellitus  
coronary_artery_disease  
dtype: int64
```

```
In [41]: y.isnull().sum()
```

```
Out[41]: class 0  
dtype: int64
```

```
In [42]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)
```

```
In [43]: print(x_train.shape)  
         print(x_test.shape)  
         print(y_train.shape)  
         print(y_test.shape)
```

```
(320, 9)  
(80, 9)  
(320, 1)  
(80, 1)
```

```
In [241]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 400 entries, 0 to 399  
Data columns (total 25 columns):
```