**Early Prediction For Chronic Kidney Disease Detection: A Progressive Approach To Health Management**

**Project Submitted by:**

**Team Leader:**

K.M.Nivethini (Register number:1420128010)

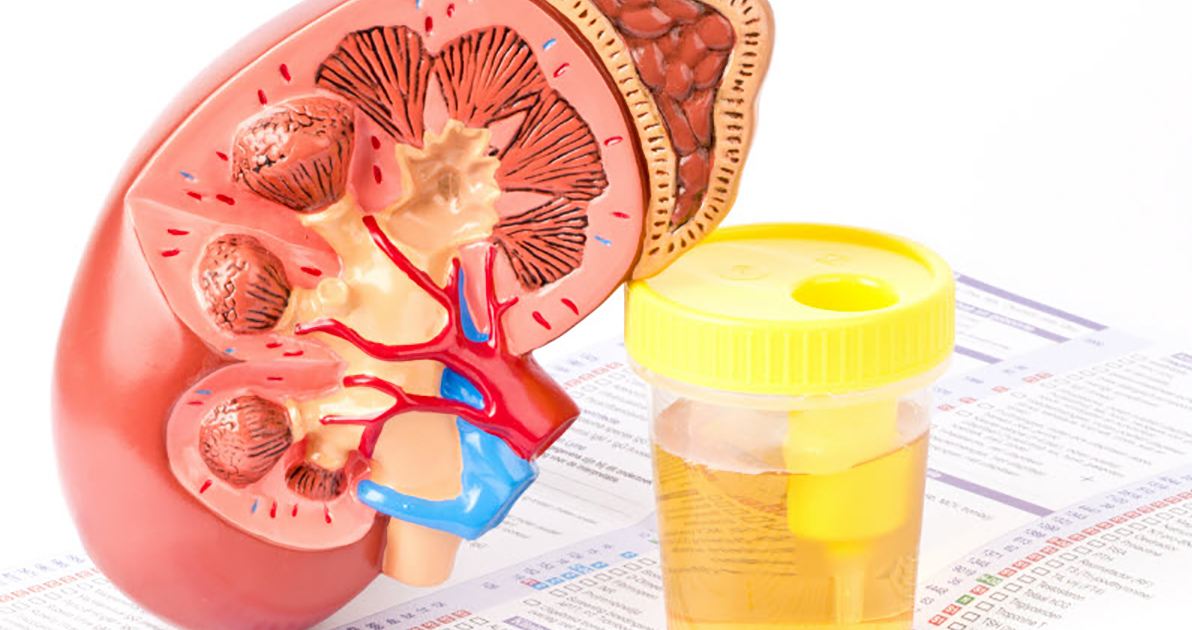
**Team members:**

1. M.Velmurugan (Register number: 1420128028)
2. M.M.Ananthakrishnan (Register number: 1420128019)
3. G.R.Balaji (Register number:1420128020)

**ABSTRACT:**

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 todays 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 pretrained model to predict the Chronic Kidney Disease which can help in treatments of peoples who are suffer from this disease.

****

**Introduction:**

The present era, especially the last two decades, can be named the era of big data where digital data is turning out to be very crucial more and more in various fields such as science, healthcare, technology, and society. Huge data volumes have been produced and generated from multiple sensor networks and mobile applications in almost all fields, including healthcare in specific, and this multitude of data volumes is what we call big data [1]. Wide variety of data sources such as streaming machines, high-end output instruments, visualizing, and knowledge extraction across these vast and diverse types of data pose a significant challenge when sufficient cutting-edge technologies and tools are not used. One of the most eminent technological challenges facing big data analytics lays in exploring ways that are adequate to obtain useful and relevant information for different user categories in an effective manner.

Nowadays, the different forms and types of data sources in healthcare are being gathered in both clinical and nonclinical environments, where the most crucial data in healthcare analytics is the digital copy of a patient’s medical history. On that account, the process of designing and making up a distributed data system to handle big data is challenged by three main issues. The first challenge is that it is difficult to collect data from distributed locations because of the diverse and large data volume. The second challenge is that storage is the chief issue for heterogeneous and enormous datasets as big data system requires to store while allowing performance guarantee. The third challenge is more connected to big data analytics, specifically to enormous mining datasets in real time, and this includes visualization, prediction, and optimization [2].

Considering the difficulty imposed by these challenges, they require an up-to-date and advanced processing paradigm provided that the present data management systems do not provide adequate efficiency in handling the heterogeneous nature of data or the real-time aspect. Traditional database management systems cannot support the continuous increase in huge data size. To address these issues related to enormous and heterogeneous data storage, the research community has proposed a number of research works, such as Apache Spark, Apache Hadoop [3], Apache Kafka [4], and Apache Storm [5], to solve healthcare problems [6–8].

Chronic kidney disease (CKD) has received a lot of interest due to its high death rate. Chronic diseases have become a major hazard to emerging countries, according to the World Health Organization (WHO) [9]. CKD is a kidney illness that can be treated in its early stages, but it eventually leads to renal failure if not treated early. In 2016, chronic kidney disease claimed the lives of 753 million individuals globally, accounting for 336 million male deaths and 417 million female deaths [10]. Chronic renal disease can be prevented from progressing to kidney failure if diagnosed and treated early. Diagnosing chronic kidney disease early is the best method to treat it, while delaying treatment until it is too late may lead to renal failure, which necessitates dialysis or kidney transplantation to live normally. Therefore, global strategies for early detection and treatment of people with CKD are required. To mine hidden patterns from data for effective decision-making and to help doctors in making more accurate diagnoses, a computer-aided diagnosis system based on artificial intelligence strategies is needed for clinical information. Artificial intelligence techniques (machine learning and deep learning) have been used in the health field, namely, in disease prediction and diagnosis.

Chronic kidney disease (CKD) is a condition that affects the kidney’s ability to function. In general, CKD is separated into phases, with renal failures occurring when the kidneys are no longer able to complete their roles of blood purification and mineral balance in the body [11]. According to the current estimates, CKD is more common in adults over 65 years old (38%) than in people aged 45–64 years (12%) and people aged 18–44 years (6%). Women have a rather higher rate of CKD (14%) than males [12].

Machine learning is an exciting field that focuses on studying huge amounts of data with multiple variables. Machine learning has basically developed from studying the theory of pattern recognition and computational learning in artificial intelligence; it presupposes computational methods, algorithms, and analysis techniques. From the perspective of Medical Sciences, machine learning undertakes to aid health specialists and doctors in carrying out scintillate and flawless diagnoses, choosing the best-fit medicines for patients, determining patients at high risk, and, most importantly, improving patients’ physical condition with minimal cost.

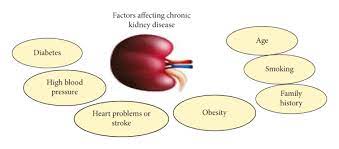
Machine learning (ML) has demonstrated remarkable performance across a range of applications, such as speech recognition [13], computer vision [14], medical diagnostics [15], and engineering [16].

Being a constituent of the ML process, feature selection (FS) is a crucial preprocessing step that determines the most relevant attributes within a dataset. Removing unimportant and unnecessary attributes can result in less complicated and more accurate models. In this paper, two feature selection methods based on Apache Spark are used, namely, Relief-F [17] and chi-squared [18] feature selection method. Some of the research works have used ML techniques to predict CKD. For example, Charleonnan [19] et al. used four ML algorithms, K-nearest neighbors (KNN), support vector machine (SVM), logistic regression (LR), and decision tree (DT), to predict CKD. Other research works used hybrid ML algorithms that are integrated between feature selection methods and ML to predict CKD. Feature selection methods have been used to reduce the number of features and select the optimal subsets of features from the dataset. For example [20], authors used chi-square, correlation-based feature selection (CFS), and Lasso feature selection to select the essential features from the database. They applied artificial neural network (ANN), C5.0, LR, SVM, KNN, and RF to both full features and the selected features.

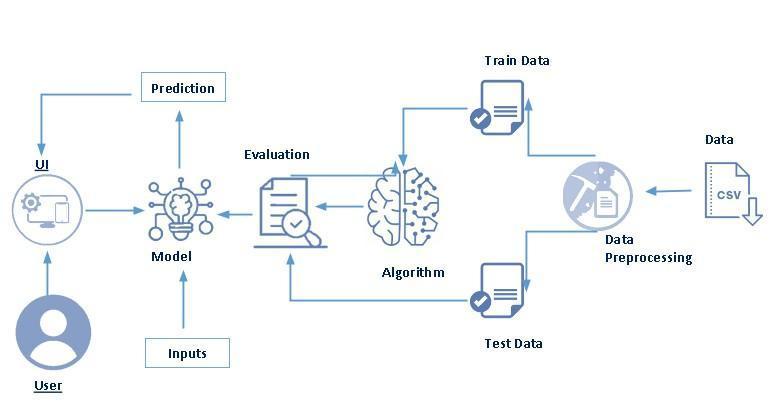
Recently, researchers have been using big data platforms such as Apache Spark [21] which is a large-scale data processing engine with a unified analytics engine. Spark is 100 times quicker than Hadoop in running workloads on large-scale clusters. It includes Java, Scala, *Python*, and *R* high-level APIs, as well as an efficient engine that supports broad execution graphs. It also includes a number of higher-level tools such as Spark SQL for SQL and structured data processing, MLlib, GraphX, and Structured Streaming.

Spark’s machine learning (ML) [21] library is called MLlib. Its purpose is to make scalable and simple machine learning a reality. It provides, at a high level, tools such as classification, regression, clustering, and collaborative filtering as examples of machine learning algorithms. It also provides feature extraction, transformation, dimensionality reduction, and selection as examples of featurization.

The previous studies of CKD prediction have not used big data platforms to solve this problem. The goal of this work is to predict CKD using hybrid ML techniques based on Apache Spark to predict CKD. Our contribution can be summarized as follows: Developing hybrid ML techniques based on Apache Spark to predict CKD Applying feature selection algorithms to select the important features from the dataset Applying optimization techniques, including grid search with cross-validation to optimize ML algorithms to enhance performance Applying different ML classification algorithms to both full features and the selected features Applying ensemble learning such as Gradient-Boosted Trees based on Apache Spark to predict CKD.

****

**Technical architecture:**



**Project flow:**

* User interacts with the UI to enter the input.
* Entered input is analysed 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
  + Social or Business Impact.
* Data Collection & Preparation
  + Collect the dataset
  + Data Preparation
* Exploratory Data Analysis
  + Descriptive statistical
  + Visual Analysis
* Model Building
  + 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

**Test for CKD**

* Chronic kidney disease is when a disease or condition makes it hard for the kidneys to work, causing the damage to the kidneys to get worse over time. This can occur when the kidneys are affected by another disease or condition.
* Studies[6](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9874070/#bb0030),[8](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9874070/" \l "bb0040) show that the number of people with CKD who are admitted to hospitals is going up by 6.23 percent every year, even though the global death rate has stayed the same.[8](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9874070/#bb0040) There are just a few diagnostic tests available to check the status of CKD, including: (i) estimated glomerular filtration rate (eGFR) (ii) a urine test; (iii) a blood pressure reading; (iv) tests for CKD.
* The eGFR value provides information on how well your kidneys cleanse the blood. If your eGFR number is higher than 90, it means that your kidneys are working well. If the value of your eGFR is less than 60, this indicates that you have CKD.[8](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9874070/#bb0040)
* In order to evaluate kidney function, the physician also requests a urine sample. Urine is produced by the kidneys. If your urine contains blood and protein,[24](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9874070/" \l "bb0120) it is an indication that one or both of your kidneys are not functioning normally.
* The doctor takes your blood pressure because the range of your blood pressure reveals how well your heart is pumping blood. If the patient’s eGFR value falls below 15, this means they have reached the end stage of kidney disease. There are just 2two treatments that are now available for renal failure: (i) dialysis and (ii) kidney transplantation. The patient’s life expectancy after dialysis is contingent on a number of characteristics, including age, gender, the frequency and length of dialysis treatments, the patient’s level of physical mobility, and their mental state.[10](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9874070/#bb0050) Kidney transplantation is the only option left for the doctor to consider if dialysis cannot be performed successfully. Nevertheless, the price is exorbitantly high.[15](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9874070/#bb0075)
* When determining the extent of the damage to your kidneys, it is not uncommon for additional tests to be performed. These may include an ultrasound scan, a magnetic resonance imaging scan, or a computed tomography scan. Their purpose is to look at the kidneys and see if there are any blockages. A needle is used to take a small piece of kidney tissue, and the cells are looked at under a microscope to look for signs of kidney disease. This is done in order to diagnose kidney conditions.
* The field of medicine is an extremely important area for the application of intellectually sophisticated systems.[21](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9874070/#bb0105) Then, data mining could be a big part of finding hidden information in the huge amount of patient medical and treatment data. This is information that doctors often get from their patients to learn more about their symptoms and make more accurate treatment plans.

### Milestone – 1:

### Define Problem / Problem Understanding:

In This milestone, we will see the problem understanding.

### Specify The Business Problem

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 todays 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 pretrained model to predict the Chronic Kidney Disease which can help in treatments of peoples who are suffer from this disease.

### Business Requirements

The business requirements for a machine learning model to predict chronic kidney disease include the ability to accurately predict the ckd based on given information, Minimise 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.

### Literature Survey

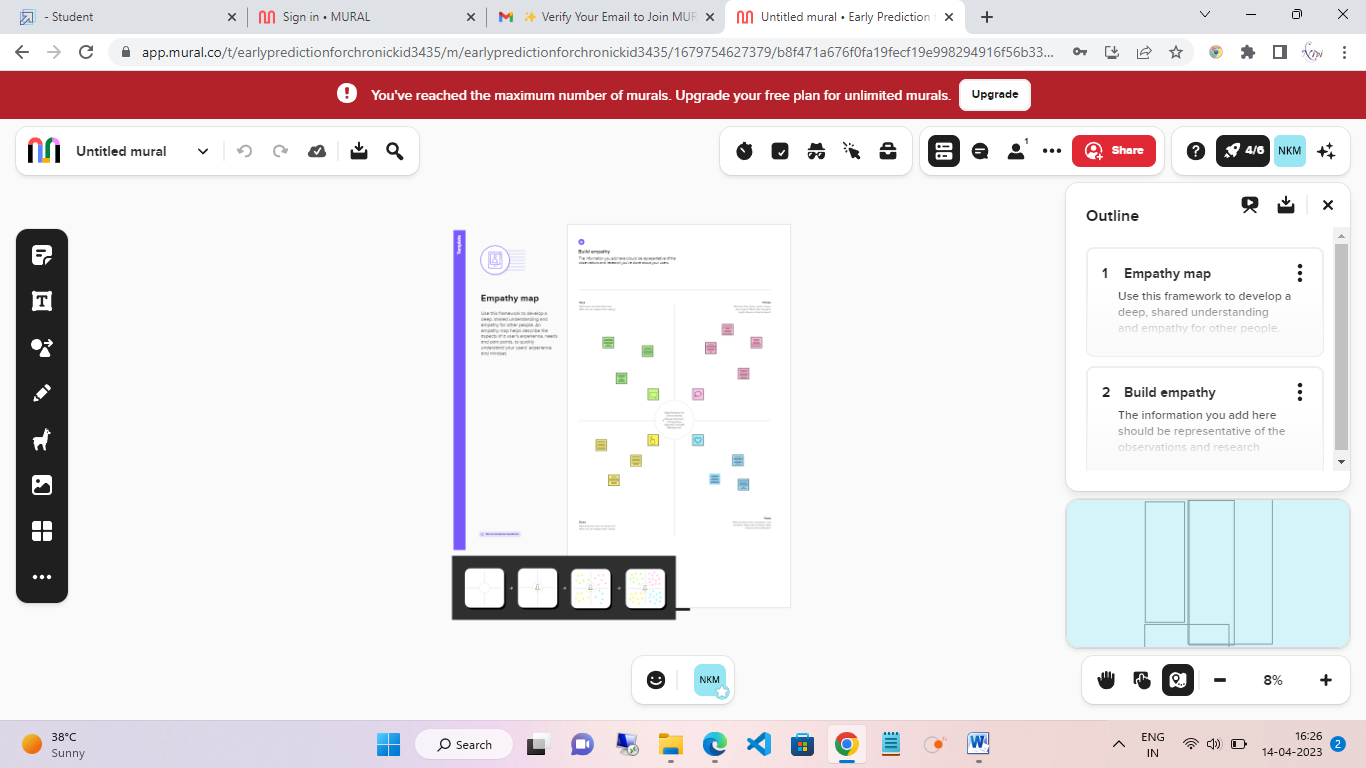
Chronic kidney disease (CKD) is a significant public health issue, affecting an estimated 14% of the global population. The disease is characterized by a gradual loss of kidney function over time, leading to a range of serious health complications, including end-stage renal disease (ESRD) requiring dialysis 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.

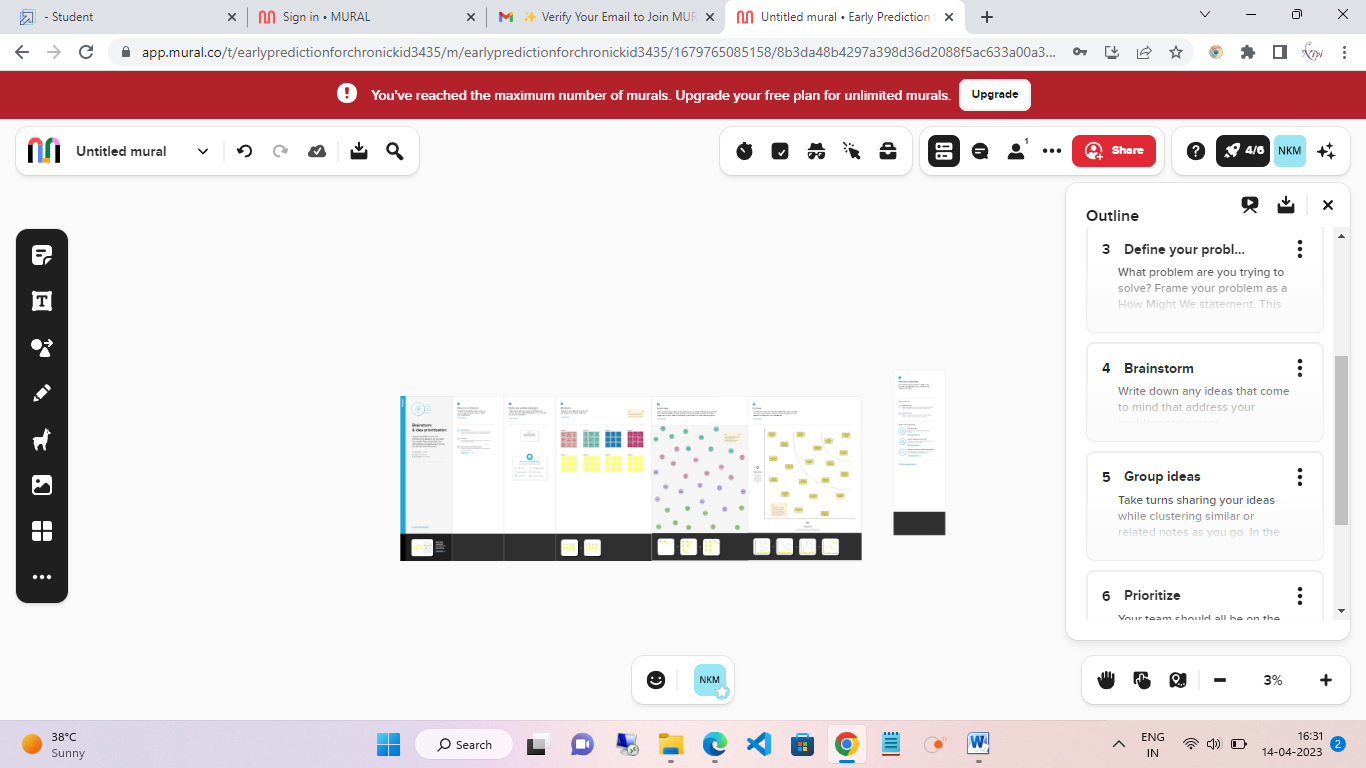
**Social Or Business Impact**

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.

**Empathy map:**



**Brainstorm map:**



**Results:**

### Cohort characteristics

The dataset contained a total of 748 subjects with the follow-up duration of 6.3 ± 2.3 years. The baseline characteristics are summarized in Table [1](https://www.nature.com/articles/s41598-022-12316-z#Tab1). Most patients were in stage 2 (24.5%) or 3 (47.1%) CKD at baseline. ESKD was observed in 70 patients (9.4%), all of whom subsequently received KRT, including hemodialysis in 49 patients, peritoneal dialysis in 17 and kidney transplantation in 4.

### Model performance

Details of the five imputed sets are provided in the supplemental parameter. There was no significant difference between the imputed sets and the original dataset in each variable where missing data were replaced by imputed values. The hyper parameter settings for each classifier are displayed in Table [2](https://www.nature.com/articles/s41598-022-12316-z#Tab2). The best overall performance, as measured by the AUC score, was achieved by the random forest algorithm (0.81, see Table [3](https://www.nature.com/articles/s41598-022-12316-z#Tab3)). Nonetheless, this score and its 95% confidence interval had overlap with those of the other three models, including the logistic regression, naïve Bayes, and the KFRE (Fig. [1](https://www.nature.com/articles/s41598-022-12316-z#Fig1)). Interestingly, the KFRE model that was based on 3 simple variables, demonstrated not only a comparable AUC score but also the highest accuracy, specificity, and precision. At the default threshold, however, the KFRE was one of the least sensitive models (47%).

**Advantages of CKD prediction:**

The early detection of CKD allows patients to receive timely treatment, slowing the disease's progression. Due to its rapid recognition performance and accuracy, machine learning models can effectively assist physicians in achieving this goal.

Disease Prediction using Machine Learning is the system that is **used to predict the diseases from the symptoms which are given by the patients or any user**. The system processes the symptoms provided by the user as input and gives the output as the probability of the disease.

**Applications of CKD prediction:**

We validate the results generated by the algorithm “bayesian classifier” and “KNN algorithm”.

**TABLE I: Attributes and Description of CKD dataset.**

|  |  |
| --- | --- |
| Attribute | Description |
| age | Age |
| bp | Blood Pressure |
| sg | Specific Gravity |
| al | Albumin |
| rbc | Red Blood Cells |
| pc | Pus Cell |
| pcc | Pus Cell Clumps |
| ba | Bacteria |
| bgr | Blood Glucose Random |
| bu | Blood Urea |
| sc | Serum Creatinine |
| sod | Sodium |
| pot | Potassium |
| hemo | Hemoglobin |
| pcv | Packed Cell Volume |
| wc | White Blood Cell Count |
| rc | Red Blood Cell Count |
| htn | Hypertension |
| dm | Diabetes Mellitus |
| cad | Coronary Artery Disease |
| appet | Appetite |
| pe | Pedal Edema |
| ane | Anemia |
| class | Class |
| su | Sugar |

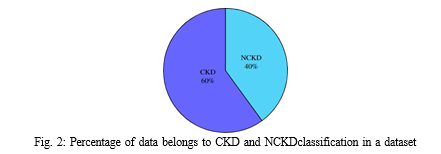
**TABLEII: Value range and Measurements of Attributes**

|  |  |
| --- | --- |
| Attribute | Measurement and Value Range |
| age | Age in Years |
| bp | bp in mm/Hg |
| sg | 1.005,1.010,1.015,1.020,1.025 |
| al | 0,1,2,3,4,5 |
| rbc | normal, abnormal |
| pc | normal, abnormal |
| pcc | Present, notpresent |
| ba | Present, notpresent |
| bgr | mgs/dl |
| bu | mgs/dl |
| sc | mgs/dl |
| sod | mEq/L |
| pot | mEq/L |
| hemo | gms |
| pcv | numerical values |
| wc | cells/cumm |
| rc | millions/cmm |
| htn | yes, no |
| dm | yes, no |
| cad | yes, no |
| appet | good, poor |
| pe | yes, no |
| ane | yes, no |
| class | ckd, notckd |
| su | 0,1,2,3,4,5 |

Chronic Kidney Disease (CKD) data set for analyzing experiment is obtained from UCI machine learning repository and clinical data set, Among 700 data sets 420 are diagnosed as having CDK and 280 as NCKD(no CKD),data set includes 25 attributes and 2 outcomes i.e., ’CKD’ and ‘ NCKD’.

**TABLE III: Confusion matrix terminology**

|  |  |  |
| --- | --- | --- |
| -- | Class/Positive 1 | Class/Negative0 |
| Class1/Positive(1) | True Positive | False Positive |
| Class2/Negative(0) | False Negative | True Negative |



### Milestone – 2:

### Data collection and preparation:

### ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.

### Collect The Dataset

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset. We should download the dataset from the kaggle.  
As the dataset is downloaded. Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques.

**Importing the libraries**

#importing required lib

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings('ignore')

#checking for available styles

plt.style.available

['Solarize\_Light2', '\_classic\_test\_patch', '\_mpl-gallery', '\_mpl-gallery-nogrid', 'bmh', 'classic', 'dark\_background', 'fast', 'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn-v0\_8', 'seaborn-v0\_8-bright', 'seaborn-v0\_8-colorblind', 'seaborn-v0\_8-dark', 'seaborn-v0\_8-dark-palette', 'seaborn-v0\_8-darkgrid', 'seaborn-v0\_8-deep', 'seaborn-v0\_8-muted', 'seaborn-v0\_8-notebook', 'seaborn-v0\_8-paper', 'seaborn-v0\_8-pastel', 'seaborn-v0\_8-poster', 'seaborn-v0\_8-talk', 'seaborn-v0\_8-ticks', 'seaborn-v0\_8-white', 'seaborn-v0\_8-whitegrid', 'tableau-colorblind10']

#Applying styles to notebook

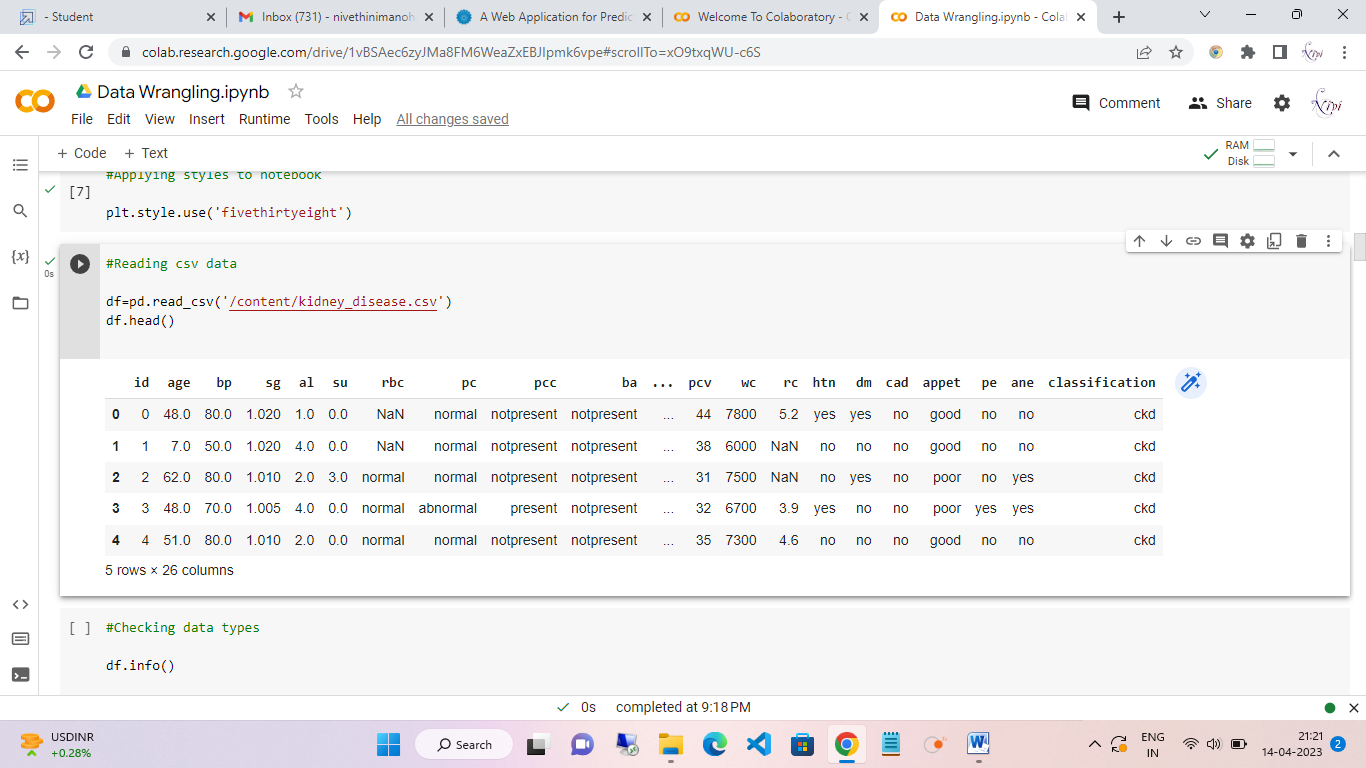
plt.style.use('fivethirtyeight')

**Reading the datasets:**

#Reading csv data

df=pd.read\_csv('/content/kidney\_disease.csv')

df.head()



**Data preparation:**

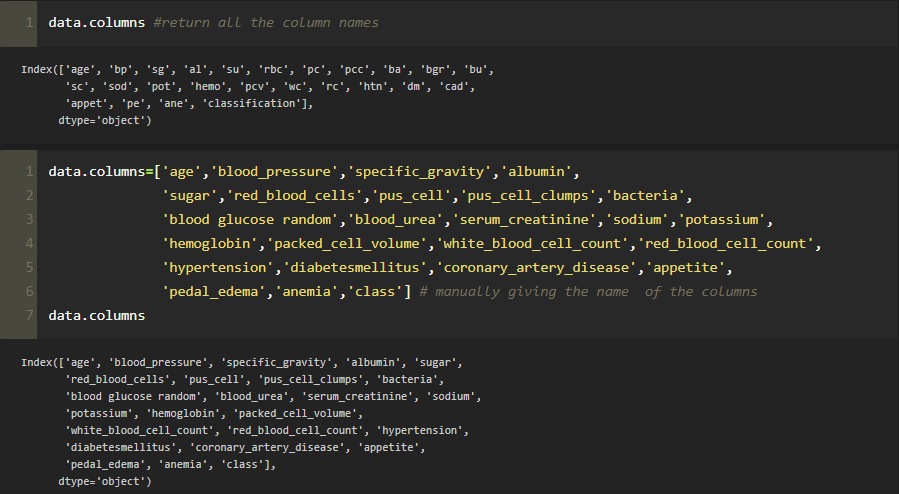
As we have understood how the data is, let's pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

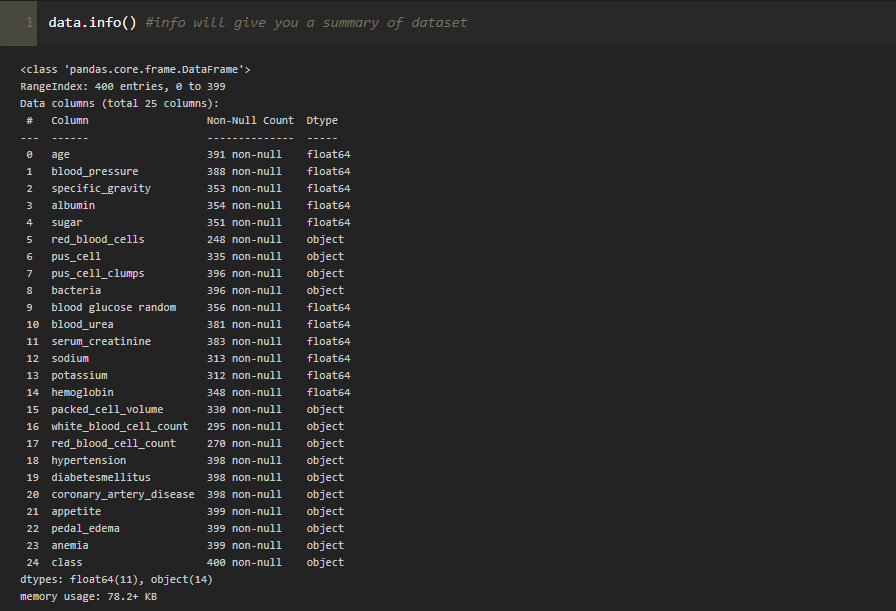
* Rename the columns
* Handling missing values
* Handling categorical data
* Handling Numerical data

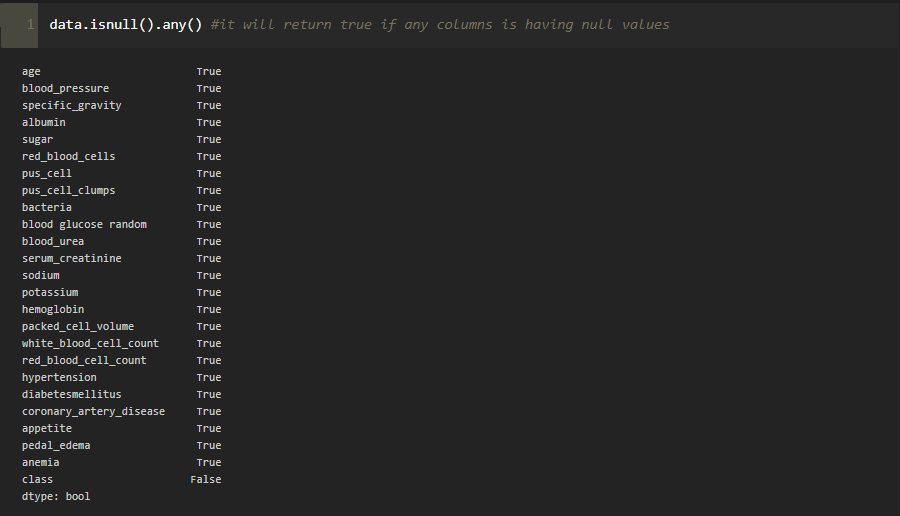
Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

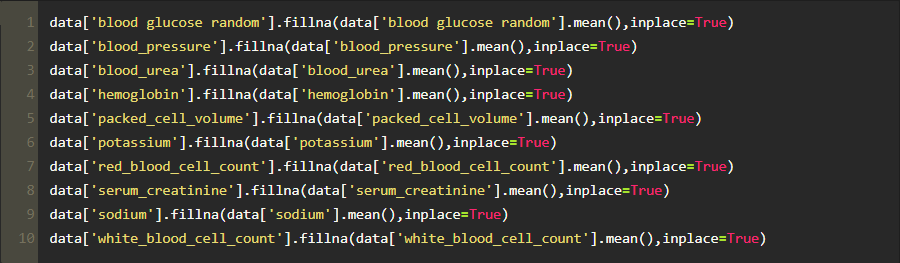
**Rename the columns**

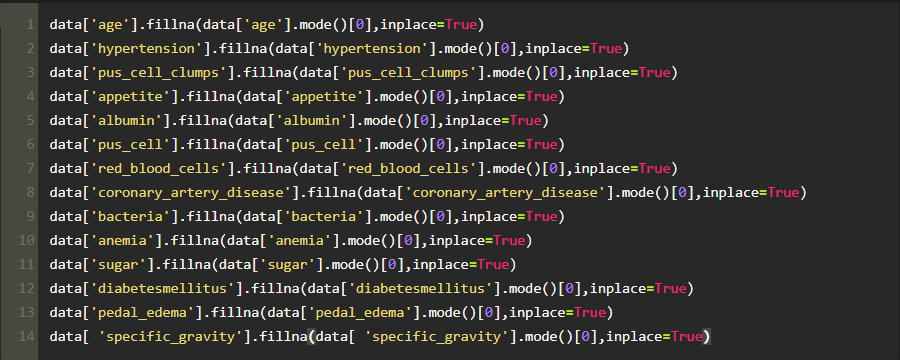
****

**Handling missing values**

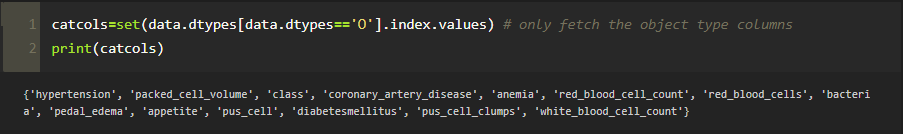
****

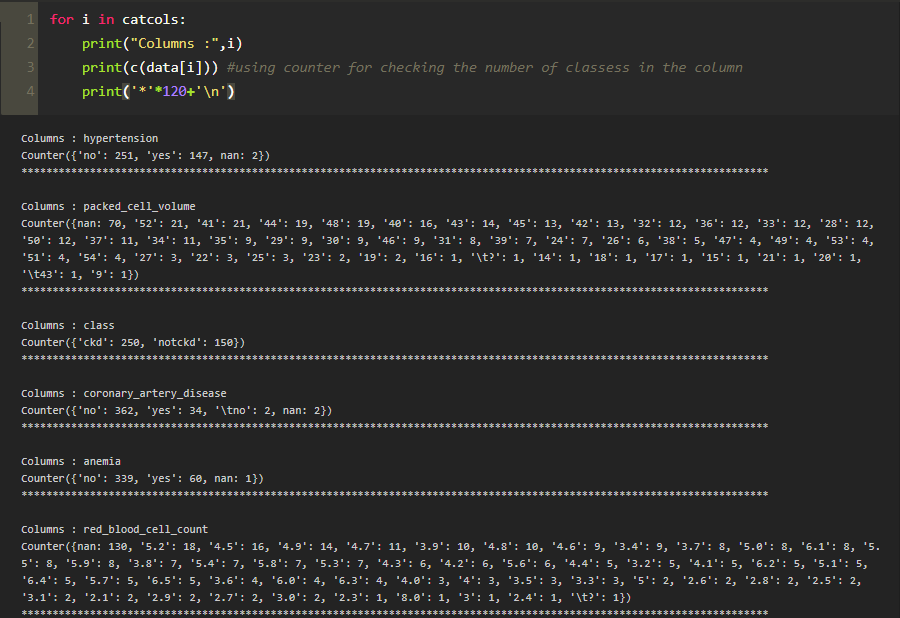
****

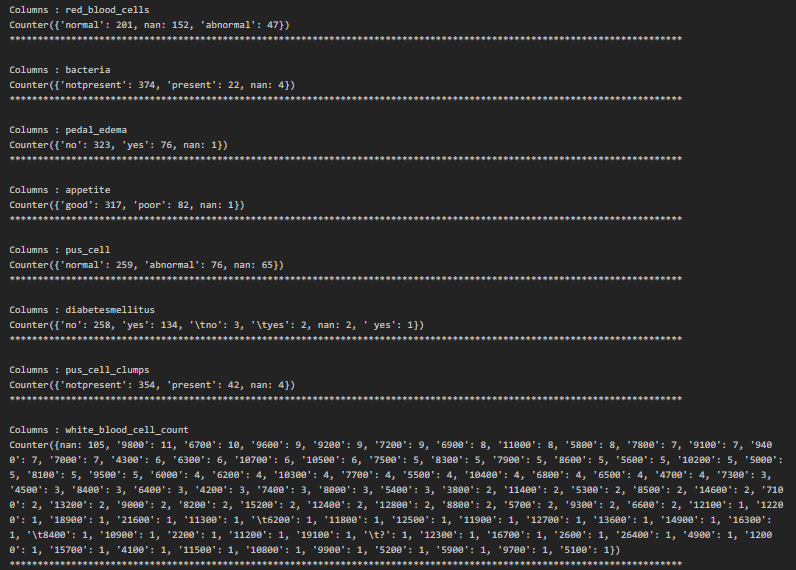


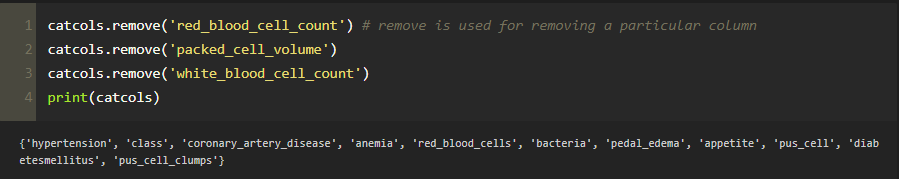


**Handling categorical data**

****



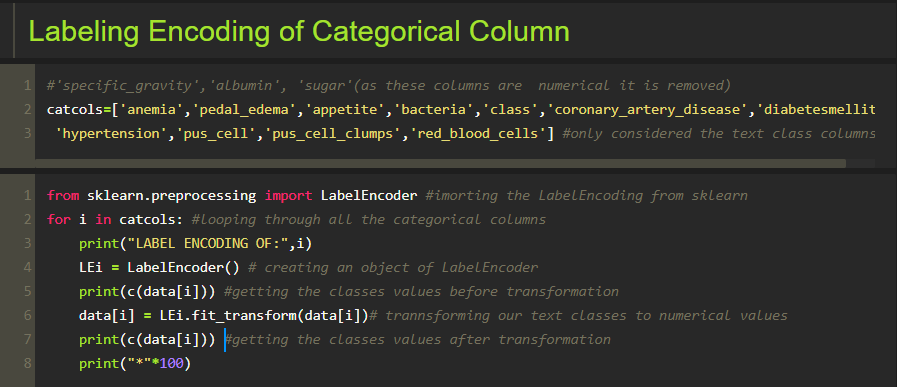


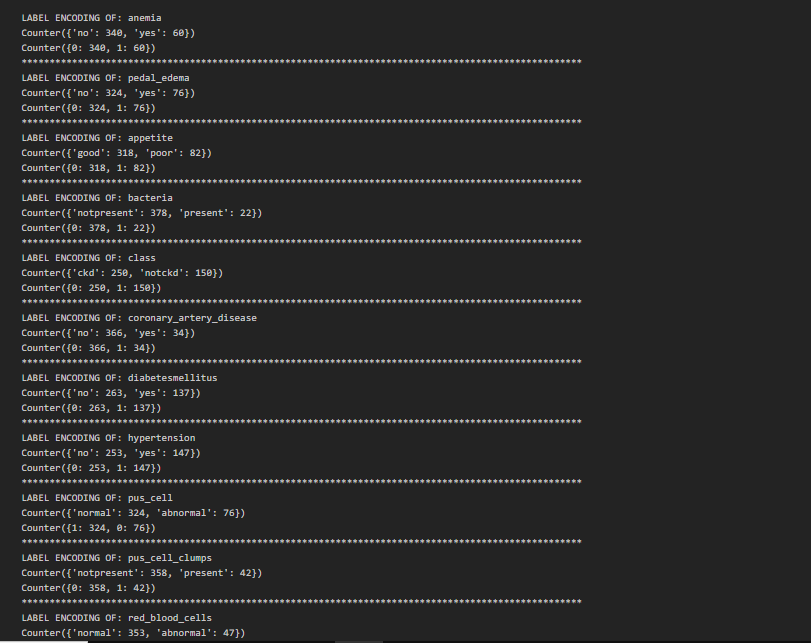


### Label Encoding For Categorical Columns

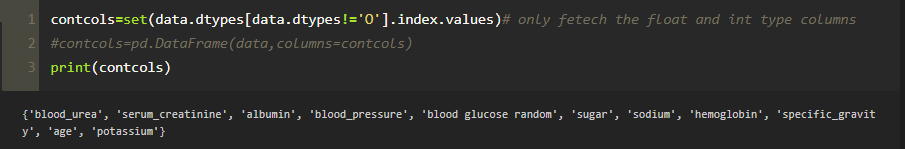
Typically, any structured dataset includes multiple columns with combination of numerical as well as categorical variables. A machine can only understand the numbers. It cannot understand the text. That’s essentially the case with Machine Learning algorithm too. We need to convert each text category to numbers in order for the machine to process those using mathematical equations.

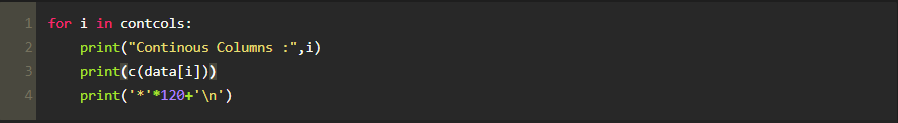
**Label Encoding** is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.

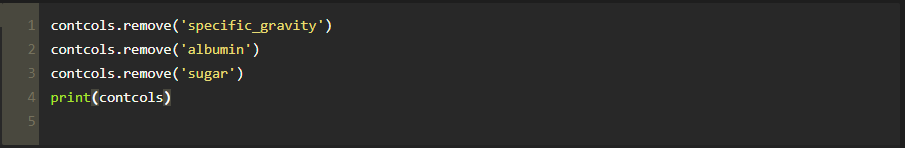


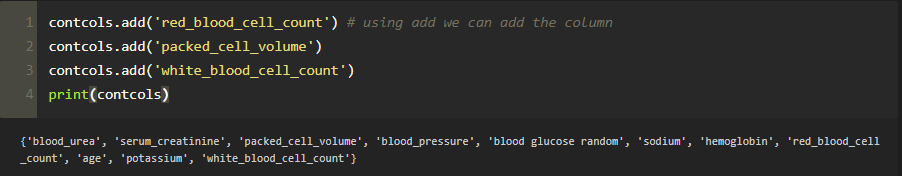


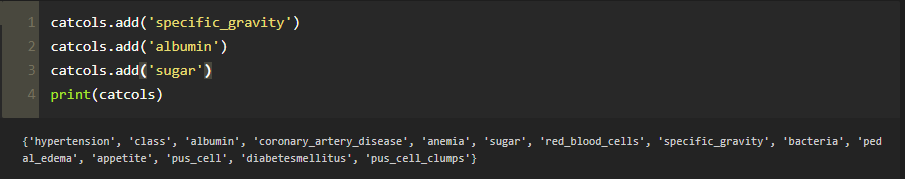
**Handling Numerical columns**

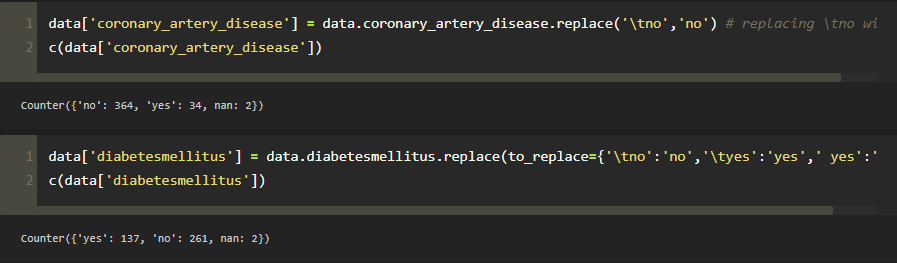
****









****

### Milestone – 3:

### Exploratory Data Analysis:

### Descriptive statistical analysis

### Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

#descriptive analysis

df.describe(include='all')

### 

### Visual Analysis

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

### Univariate Analysis

In simple words, univariate analysis is understanding the data with a single feature. Here we have displayed two different graphs such as distplot and countplot.

The Seaborn package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature.

#Univariate analysis - Extracting info from a single column

#Checking data distribution

plt.subplot(121)

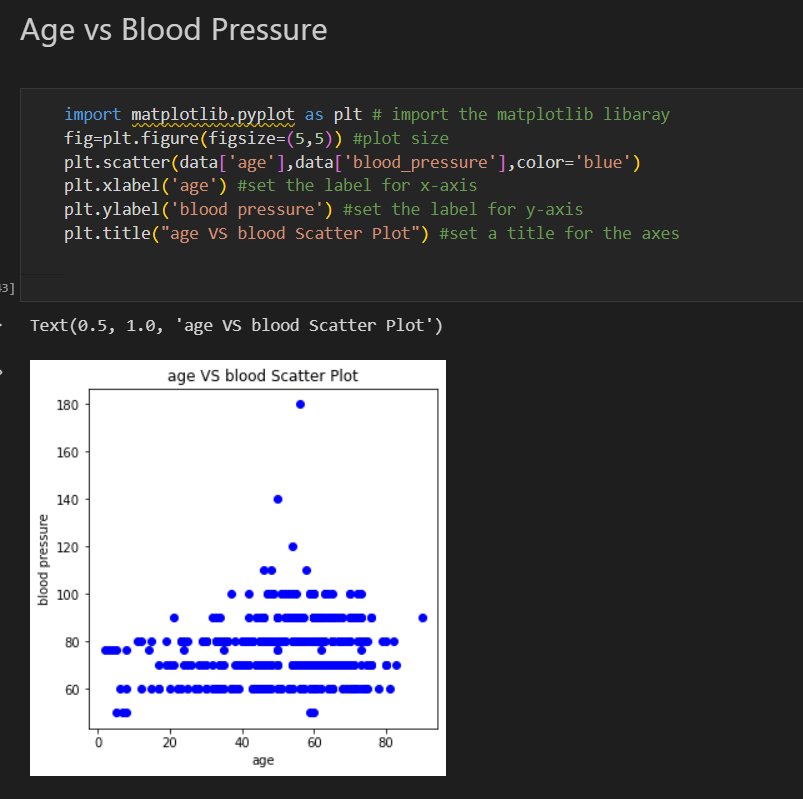
sns.distplot(df['age'])

plt.subplot(122)

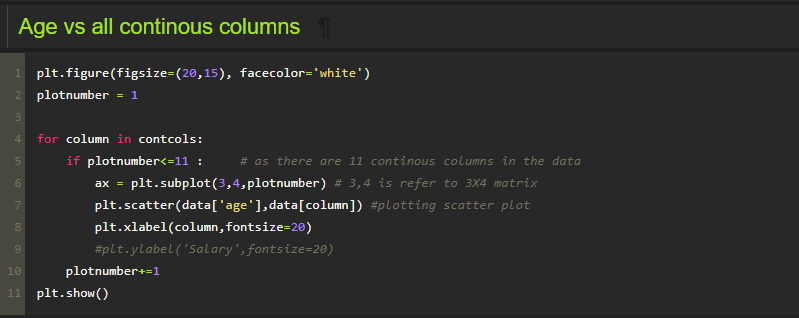
sns.distplot(df['bp'])

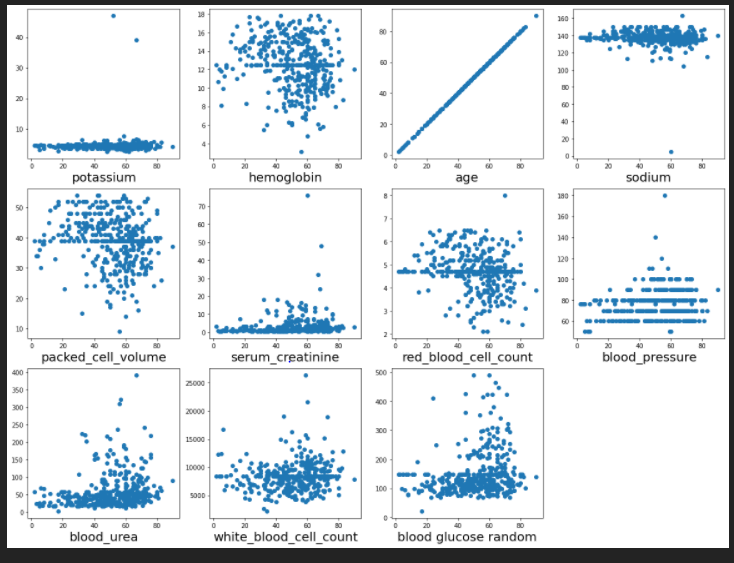
### 

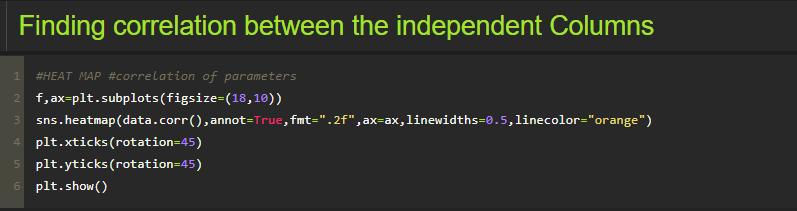
**Bivariate analysis**

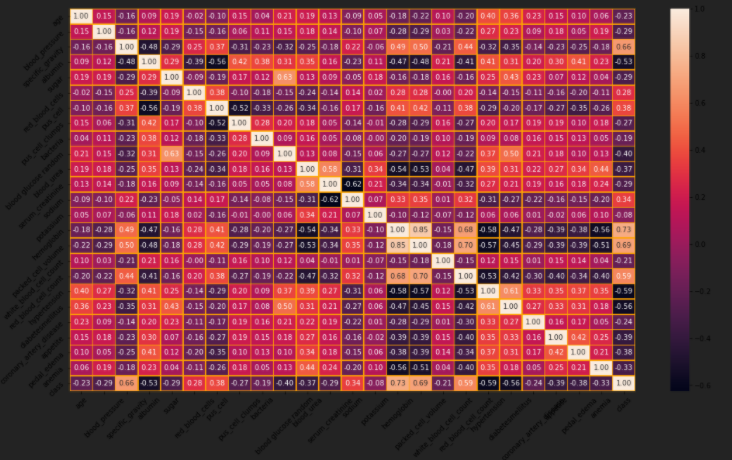
****

**Multivariate analysis**

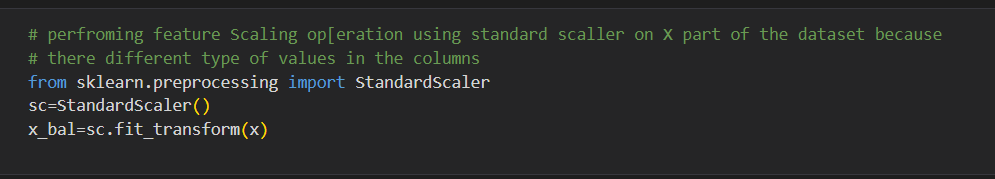
****

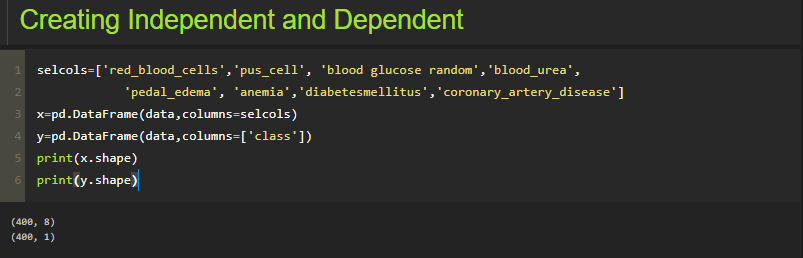
****

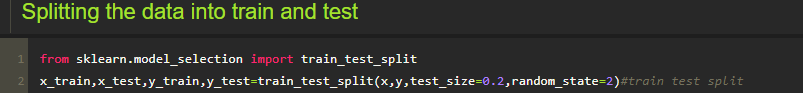
****

****

****

****

****

****

### Milestone – 4:

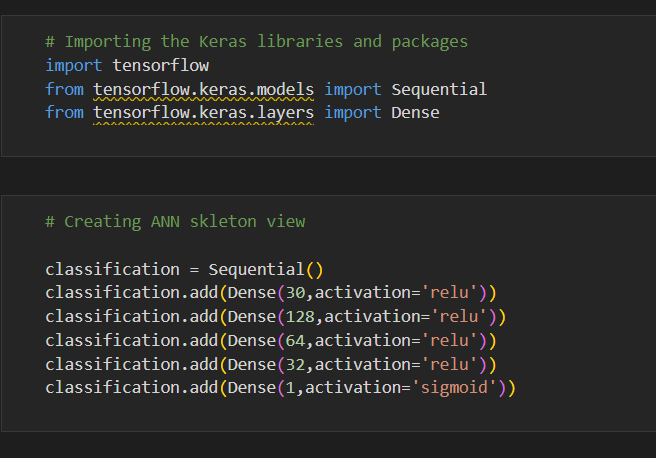
### Data modelling:

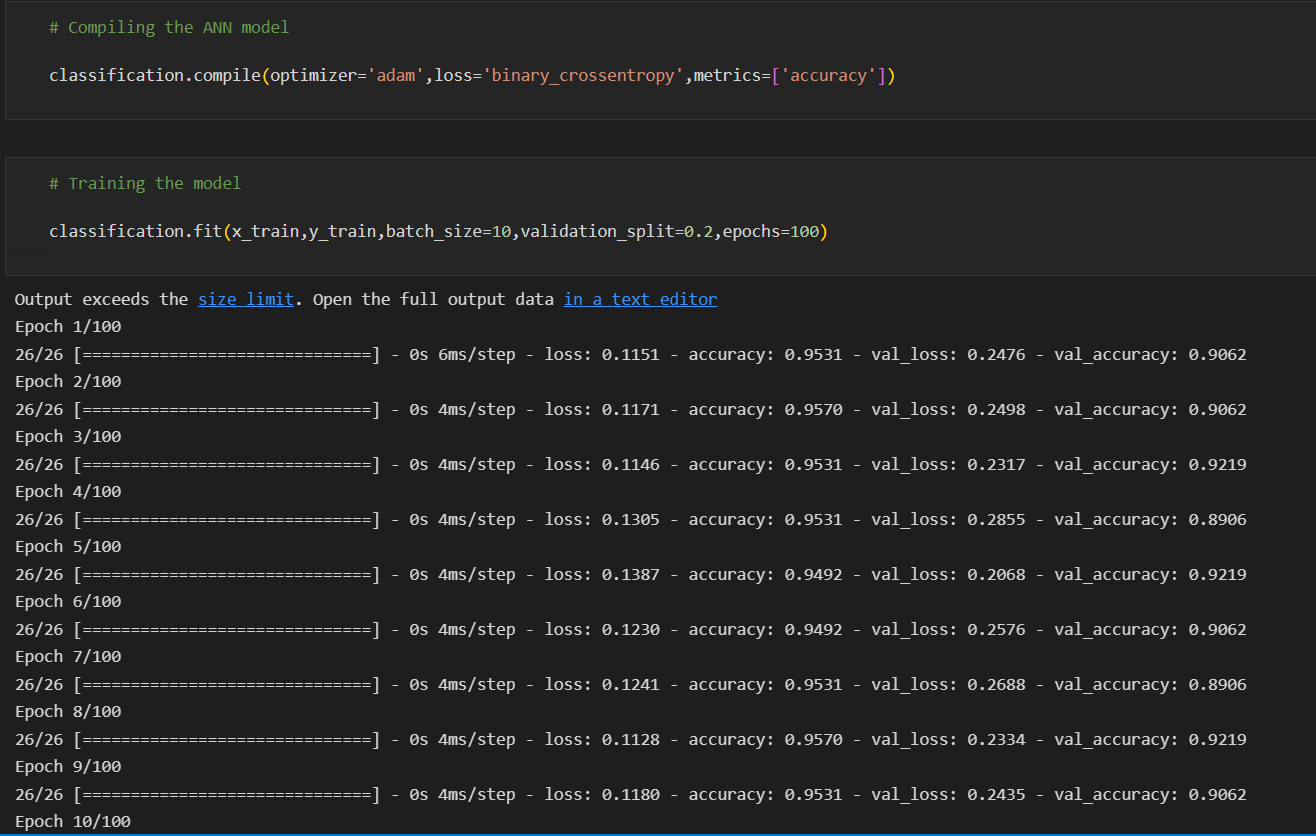
### Training the Model In Multiple Algorithms

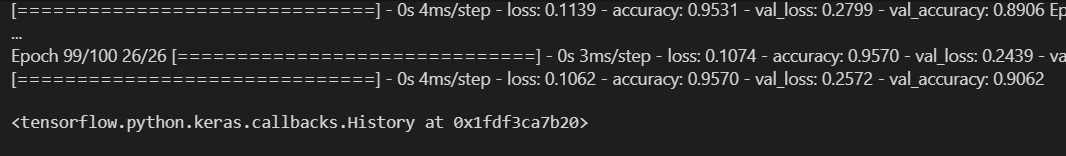
Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying four classification algorithms. The best model is saved based on its performance.

### ANN Model

Building and training an Artificial Neural Network (ANN) using the Keras library with TensorFlow as the backend. The ANN is initialised as an instance of the Sequential class, which is a linear stack of layers. Then, the input layer and two hidden layers are added to the model using the Dense class, where the number of units and activation function are specified. The output layer is also added using the Dense class with a sigmoid activation function. The model is then compiled with the Adam optimizer, binary cross-entropy loss function, and accuracy metric. Finally, the model is fit to the training data with a batch size of 100, 20% validation split, and 100 epochs

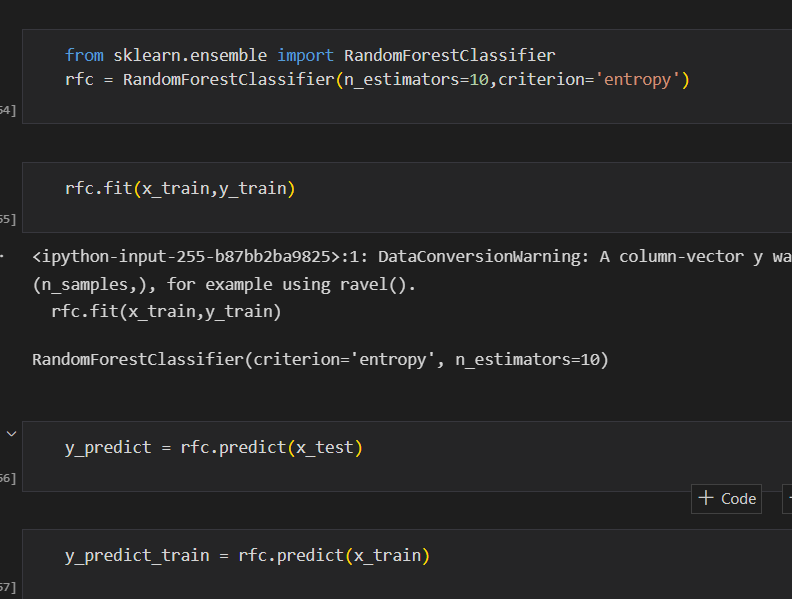




****

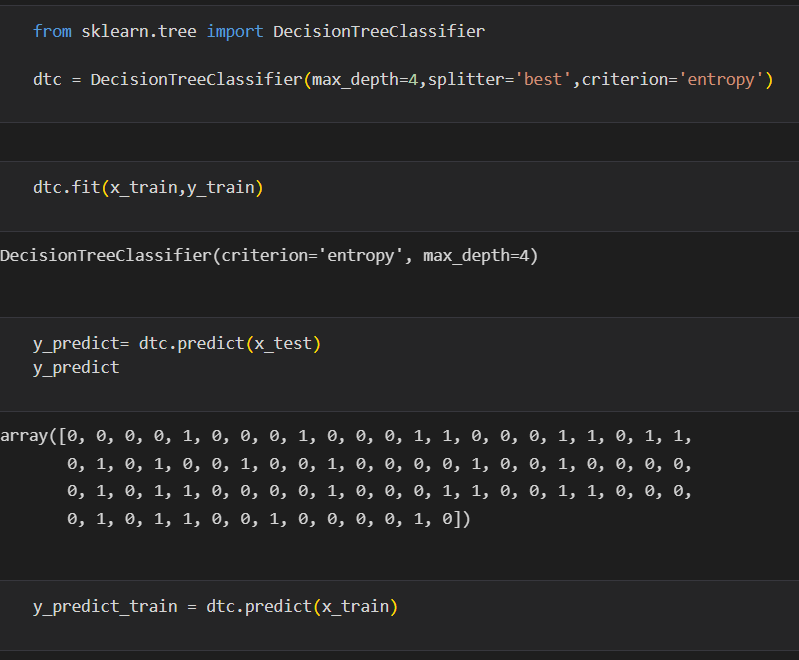
### Random Forest Model

A function named random Forest is created and train and test data are passed as the parameters. Inside the function, Random Forest Classifier algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with. predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

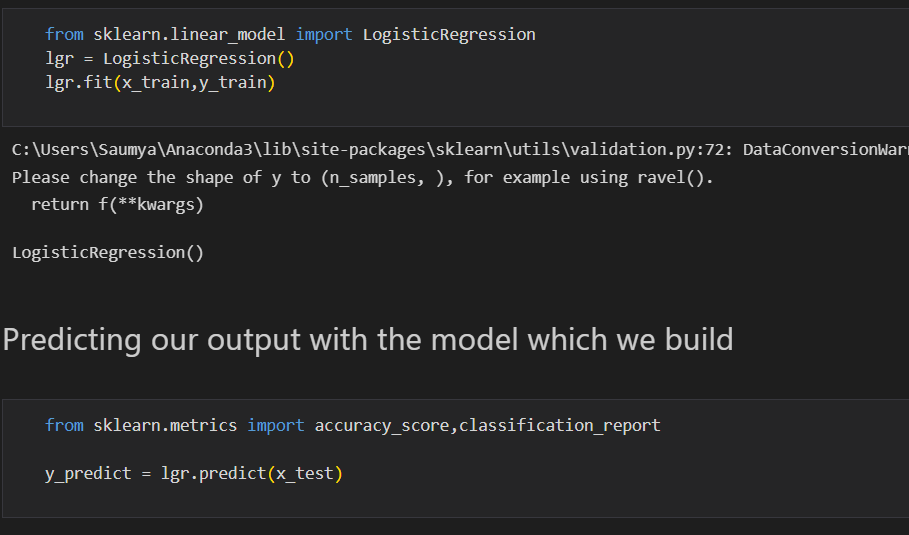
****

### Decision Tree Model

A function named decision Tree is created and train and test data are passed as the parameters. Inside the function, Decision Tree Classifier algorithm is initialised and training data is passed to the model with the .fit() function. Test data is predicted with. predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

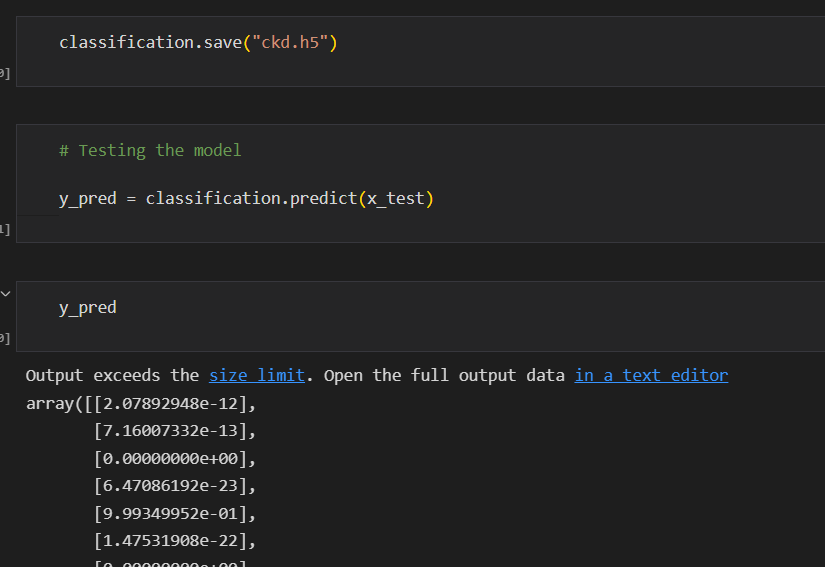
****

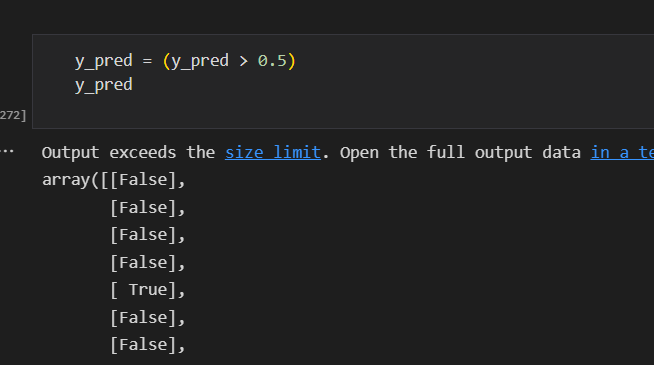
### Logistic regression

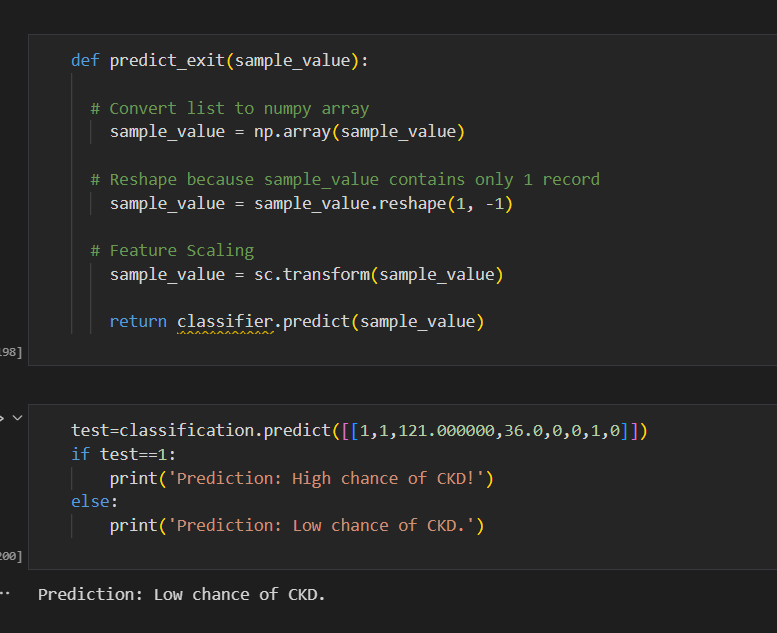
****

### Testing the model

### C:\Users\Ram Ramana\Downloads\NM project\37.png

****

****

****

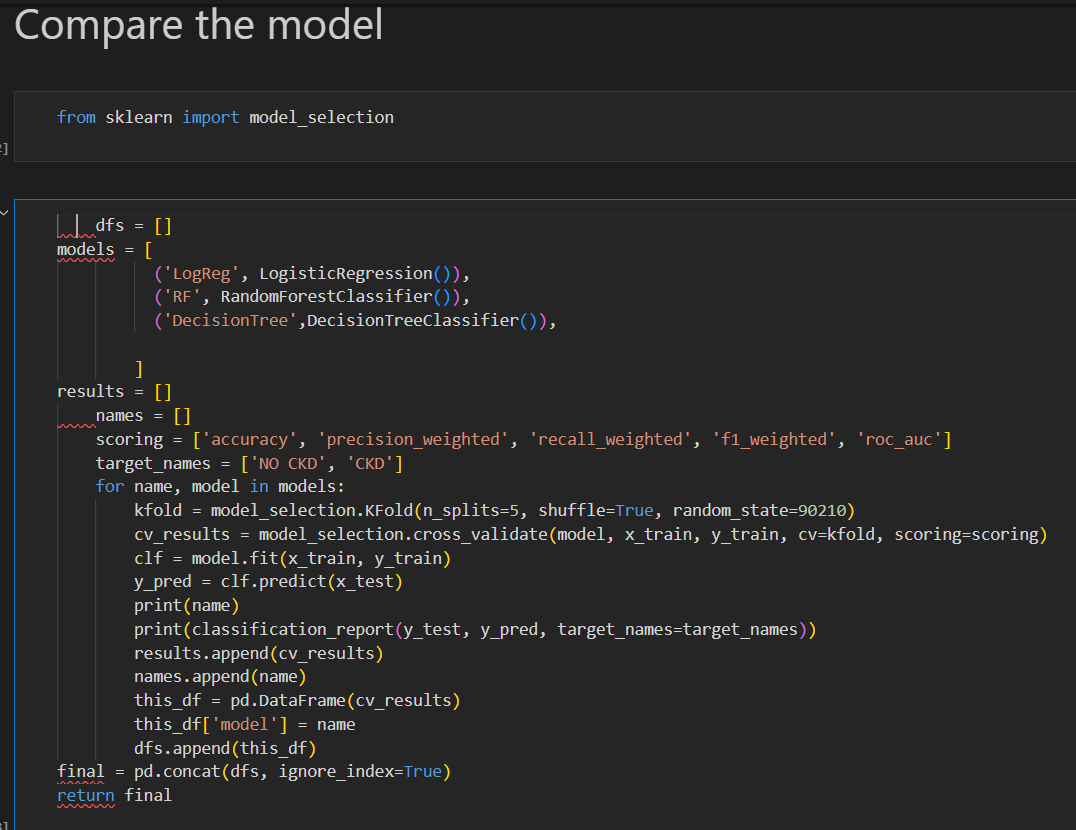
### Milestone – 5:

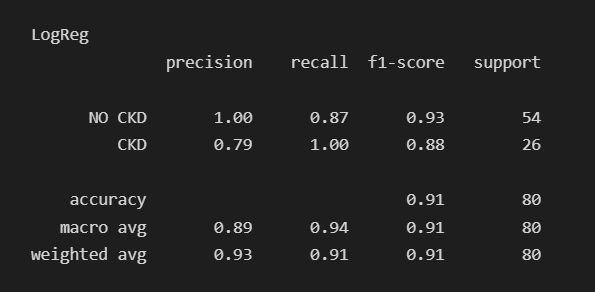
### Performance testing and evaluating the results:

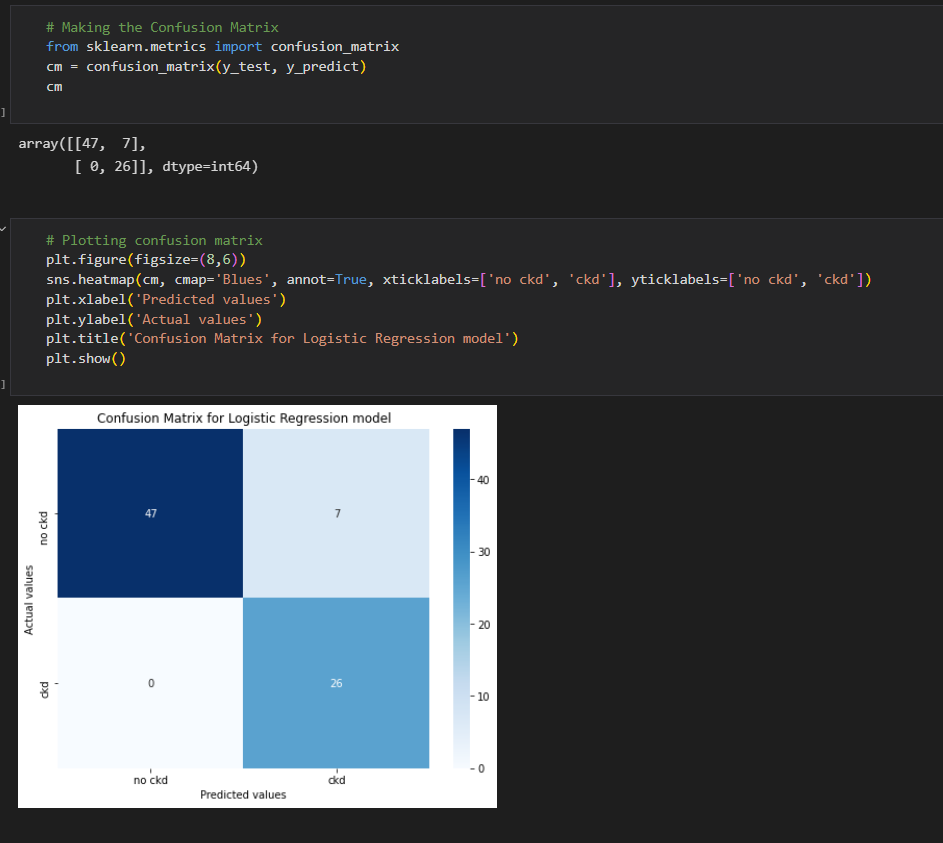
In this milestone, we will see the performance testing

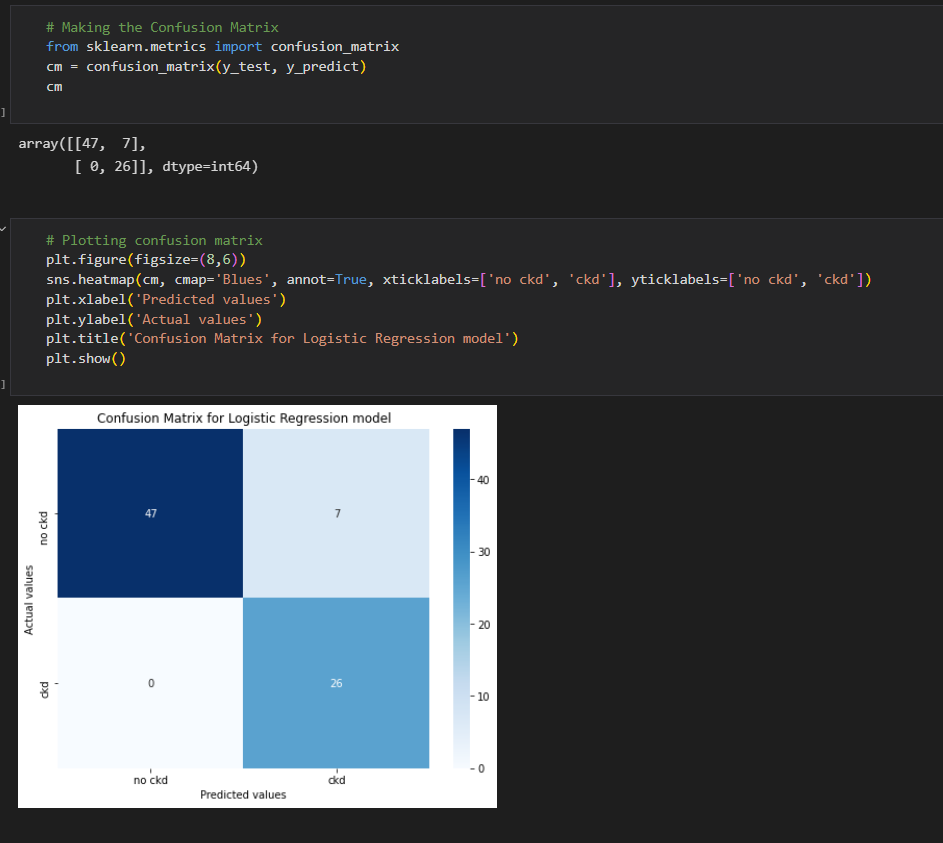
### Testing Model With Multiple Evaluation Metrics

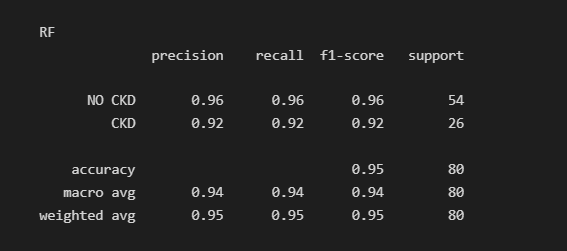
Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for classification tasks including accuracy, precision, recall, support and F1-score.

****

****

****

****

****

****

****

### Evaluating the results

### 

### 

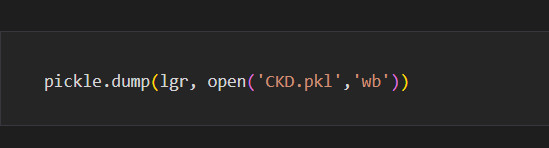
### Milestone – 6:

### Model deployment:

In this milestone, we will see the model deployment.

### Save The Best Model

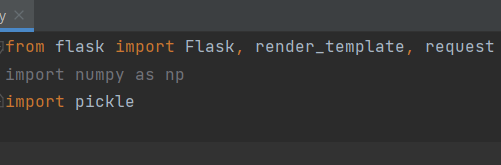
Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and to be able to use it in the future.

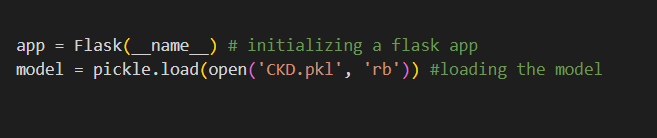
****

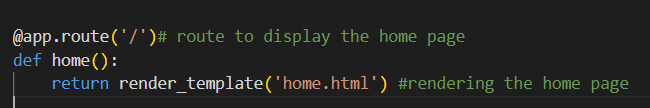
### Integrate With Web Framework

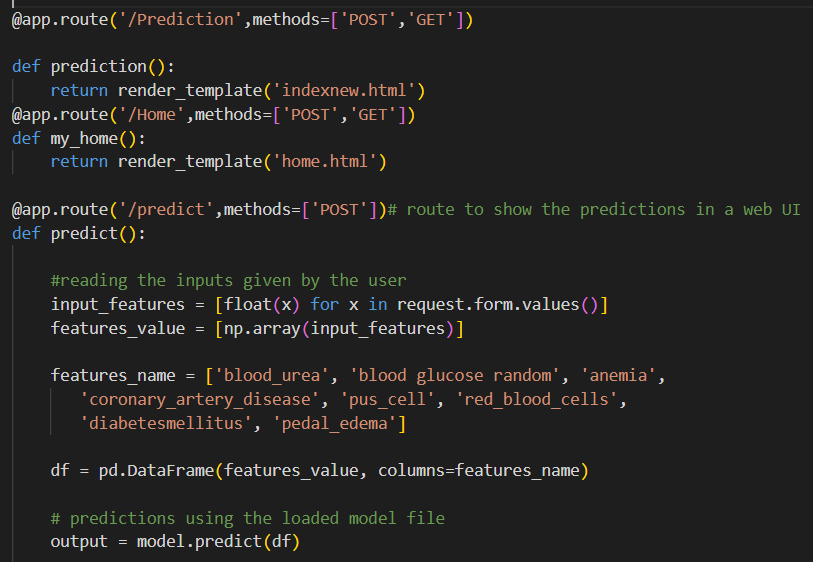
In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

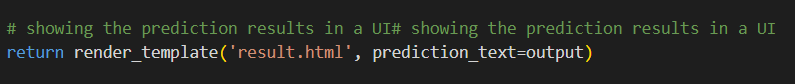
**Build python code**

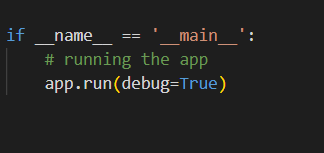
****

****

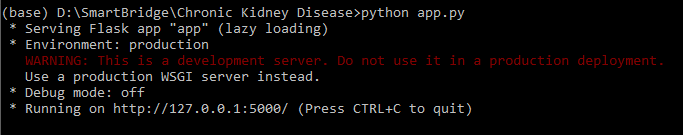
****

****

****

****

**Run the web application**

****

### Source code

### Importing necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model\_selection import GridSearchCV, train\_test\_split, cross\_val\_score

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import r2\_score

import scipy.stats as stats

import seaborn as sns

%matplotlib inline

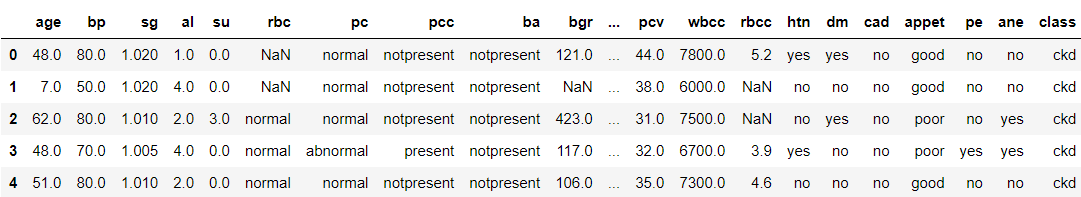
**Now, let’s read our dataset**

df = pd.read\_csv('kidney.csv')

data = df

data.head()

**Output:**

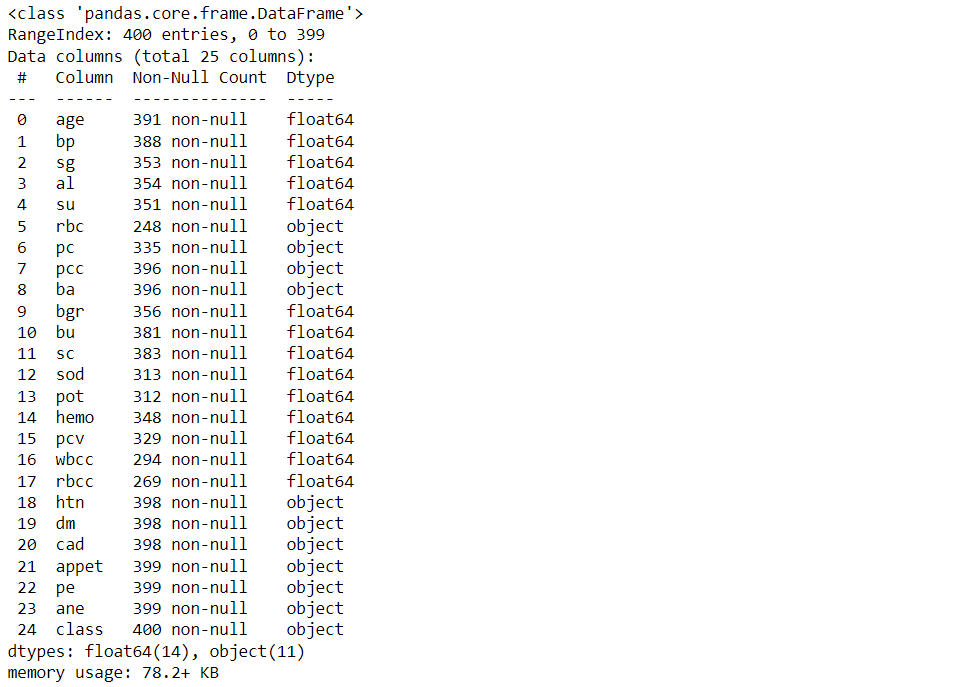


data.shape

(400, 25)

## Data Pre-processing

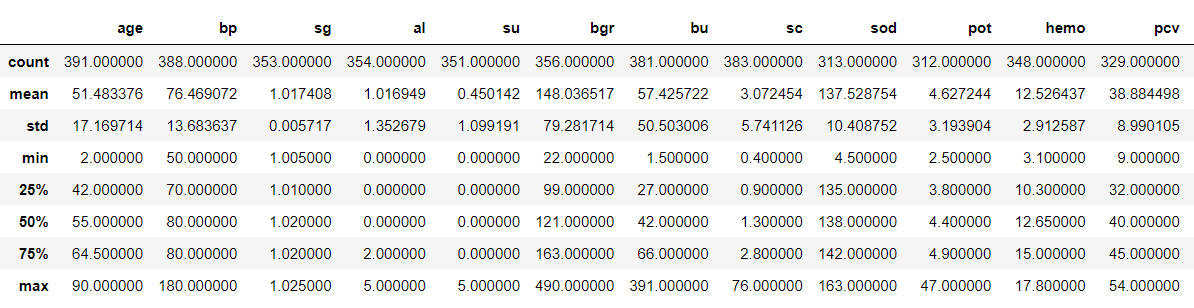
df.info()



**Now, let’s see our data statistically**

df.describe()

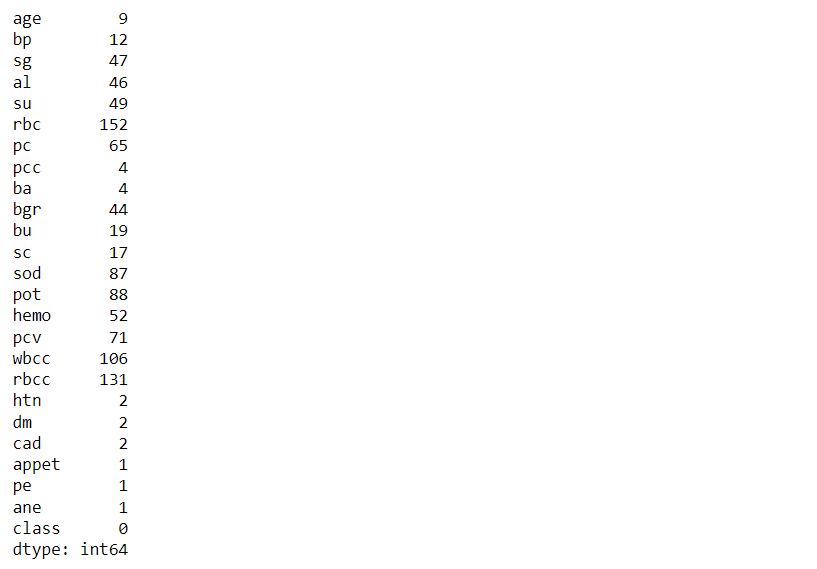
**Output:**



**Now, let’s see the total count of null values that every feature holds**

df.isna().sum()

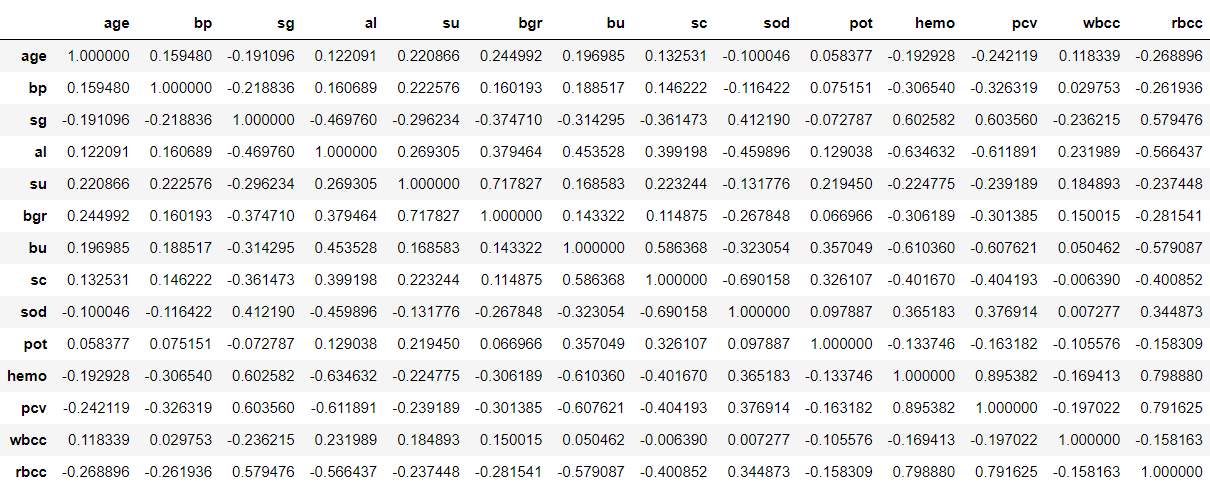
**Output:**



### Correlation matrix & Matrix Visualisation

df.corr()

**Output:**



**Let’s find out how many of each class are:**

df['class'].value\_counts()

**Output:**

ckd 250

notckd 150

Name: class, dtype: int64

**Representation of Target variable in Percentage**

countNoDisease = len(df[df['class'] == 0])

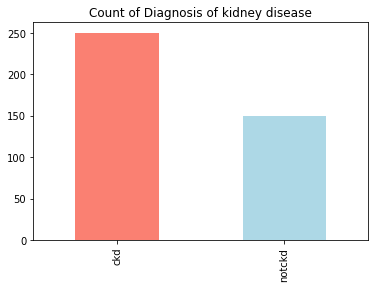
countHaveDisease = len(df[df['class'] == 1])

print("Percentage of Patients Haven't Heart Disease: {:.2f}%".format((countNoDisease / (len(df['class']))\*100)))

print("Percentage of Patients Have Heart Disease: {:.2f}%".format((countHaveDisease / (len(df['class']))\*100)))

**Understanding the balancing of the data visually**

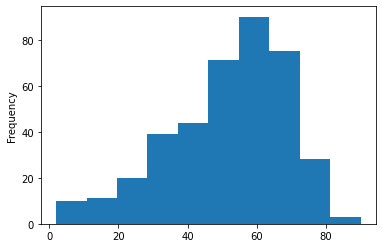
df['class'].value\_counts().plot(kind='bar',color=['salmon','lightblue'],title="Count of Diagnosis of kidney disease")



**Here we will be checking the distribution of the age column**

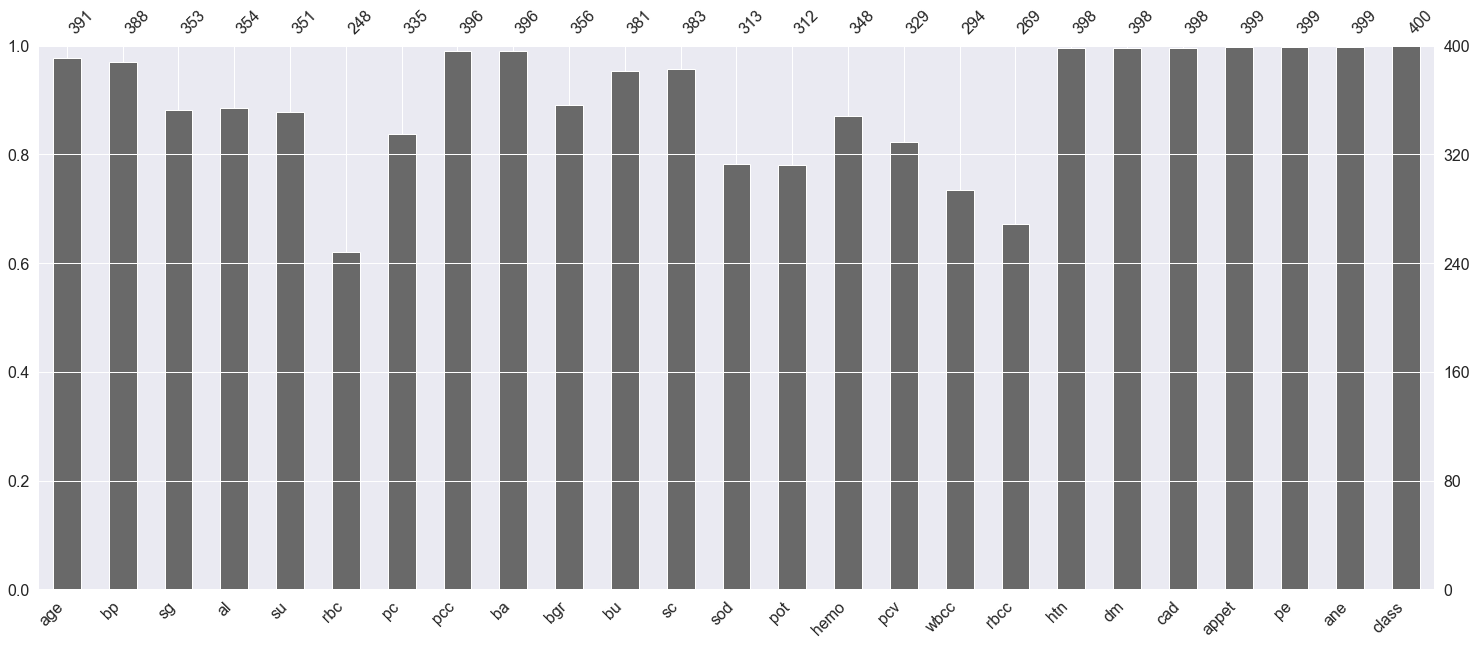
df['age'].plot(kind='hist')

**Output:**



**Here we are plotting the graph to see the null values in the dataset.**

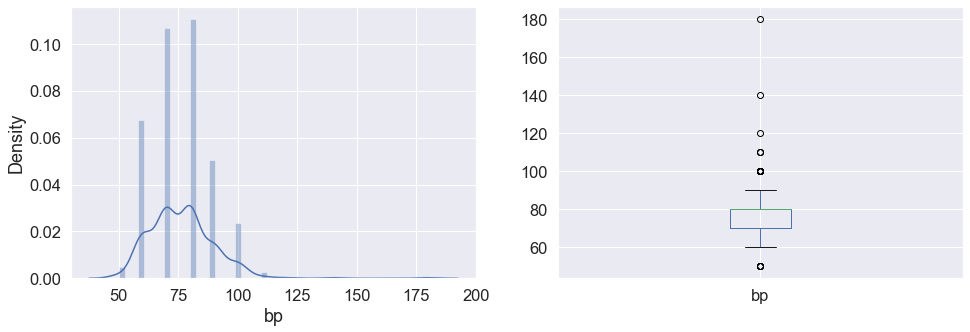
p = msno.bar(data)

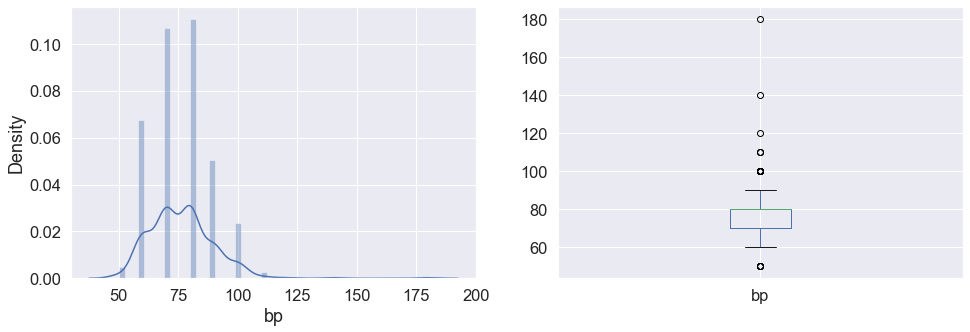


plt.subplot(121), sns.distplot(data['bp'])

plt.subplot(122), data['bp'].plot.box(figsize=(16,5))

plt.show()





**Now we will convert the categorical values(object) to categorical values(int)**

data['class'] = data['class'].map({'ckd':1,'notckd':0})

data['htn'] = data['htn'].map({'yes':1,'no':0})

data['dm'] = data['dm'].map({'yes':1,'no':0})

data['cad'] = data['cad'].map({'yes':1,'no':0})

data['appet'] = data['appet'].map({'good':1,'poor':0})

data['ane'] = data['ane'].map({'yes':1,'no':0})

data['pe'] = data['pe'].map({'yes':1,'no':0})

data['ba'] = data['ba'].map({'present':1,'notpresent':0})

data['pcc'] = data['pcc'].map({'present':1,'notpresent':0})

data['pc'] = data['pc'].map({'abnormal':1,'normal':0})

data['rbc'] = data['rbc'].map({'abnormal':1,'normal':0})

data['class'].value\_counts()

**Output:**

1 250

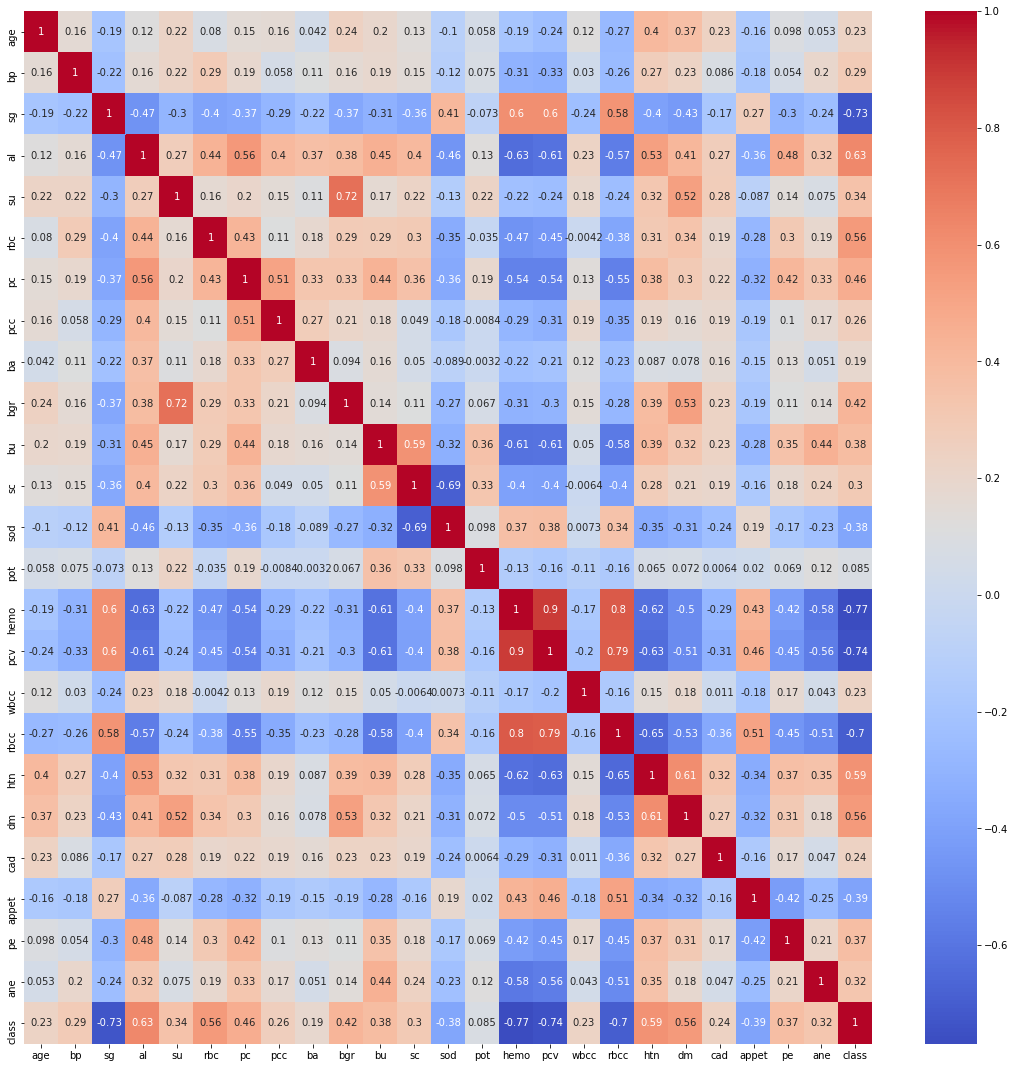
0 150

Name: class, dtype: int64

**Finding the Correlation between the plots**

plt.figure(figsize = (19,19))

sns.heatmap(data.corr(), annot = True, cmap = 'coolwarm') # looking for strong correlations with "class" row



## Exploratory data analysis (EDA)

Let’s see the shape of the dataset again after analysis

data.shape

(400, 25)

**Now let’s see the final columns present in our dataset.**

data.columns

**Output:**

Index(['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',

'sc', 'sod', 'pot', 'hemo', 'pcv', 'wbcc', 'rbcc', 'htn', 'dm', 'cad',

'appet', 'pe', 'ane', 'class'],

dtype='object')

**Now dropping the null values.**

data.shape[0], data.dropna().shape[0]

**Results:**

Here from the above output, **we can see that there are 158 null values in the dataset.** Now here we are left with two choices that we could either drop all the null values or keep them when we will drop that NA values so we should understand that our dataset is not that large and if we drop those null values then **it would be even smaller in that case if we provide very fewer data to our machine learning model then the performance would be very less also we yet don’t know that these null values are related to some other features in the dataset.**

**So for this time I’ll keep these values and see how the model will perform in this dataset.**

Also when we are working on some healthcare project where we will be predicting whether the person is suffering from that disease or not then one thing we should **keep in my mind is that the model evaluation should have the least false positive errors.**

data.dropna(inplace=True)

data.shape

**Output:**

(158, 25)

## Model Building

1. **Logistic regression**

from sklearn.linear\_model import LogisticRegression

logreg = LogisticRegression()

X = data.iloc[:,:-1]

y = data['class']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, stratify = y, shuffle = True)

logreg.fit(X\_train,y\_train)

**Output:**

LogisticRegression()

**Training score**

logreg.score(X\_train,y\_train)

**Output:**

1.0

**Testing accuracy**

logreg.score(X\_test,y\_test)

**Output:**

0.975

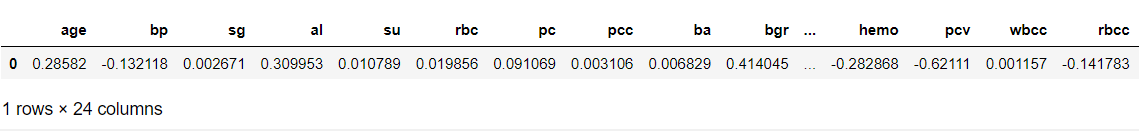
Train Accuracy: 1.0

Test Accuracy: 0.975

### The cell below shows the coefficients for each variable.

(example on reading the coefficients from a Logistic Regression: a one-unit increase in age makes an individual about e^0.14 time as likely to have CKD, while a one-unit increase in blood pressure makes an individual about e^-0.07 times as likely to have CKD.

pd.DataFrame(logreg.coef\_, columns=X.columns)



### Confusion Matrix

sns.set(font\_scale=1.5)

def plot\_conf\_mat(y\_test,y\_preds):

"""

This function will be heloing in plotting the confusion matrix by using seaborn

"""

fig,ax=plt.subplots(figsize=(3,3))

ax=sns.heatmap(confusion\_matrix(y\_test,y\_preds),annot=True,cbar=False)

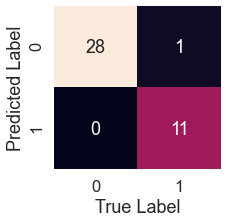
plt.xlabel("True Label")

plt.ylabel("Predicted Label")

log\_pred = logreg.predict(X\_test)

plot\_conf\_mat(y\_test, log\_pred)

**Output:**



tn, fp, fn, tp = confusion\_matrix(y\_test, test\_pred).ravel()

print(f'True Neg: {tn}')

print(f'False Pos: {fp}')

print(f'False Neg: {fn}')

print(f'True Pos: {tp}')

**Output:**



### K-Nearest Neighbors Classifier

It is a good practice to first balance the class well before using the KNN, as we know that in the case of unbalanced classes KNN doesn’t perform well.

df["class"].value\_counts()

**Output:**

0 115

1 43

Name: class, dtype: int64

**Now let’s CONCATENATE our class variables together.**

balanced\_df = pd.concat([df[df["class"] == 0], df[df["class"] == 1].sample(n = 115, replace = True)], axis = 0)

balanced\_df.reset\_index(drop=True, inplace=True)

balanced\_df["class"].value\_counts()

**Output:**

1 115

0 115

Name: class, dtype: int64

**Now let’s scale down the data**

ss = StandardScaler()

ss.fit(X\_train)

X\_train = ss.transform(X\_train)

X\_test = ss.transform(X\_test)

**Output:**

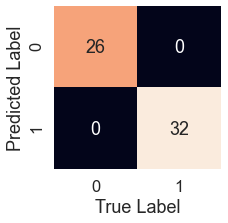
1.0

**Confusion matrix for KNN model**

knn\_pred = model.predict(X\_test)

plot\_conf\_mat(y\_test, knn\_pred)

**Output:**



tn, fp, fn, tp = confusion\_matrix(y\_test, preds).ravel()

print(f'True Neg: {tn}')

print(f'False Pos: {fp}')

print(f'False Neg: {fn}')

print(f'True Pos: {tp}')

**Output:**



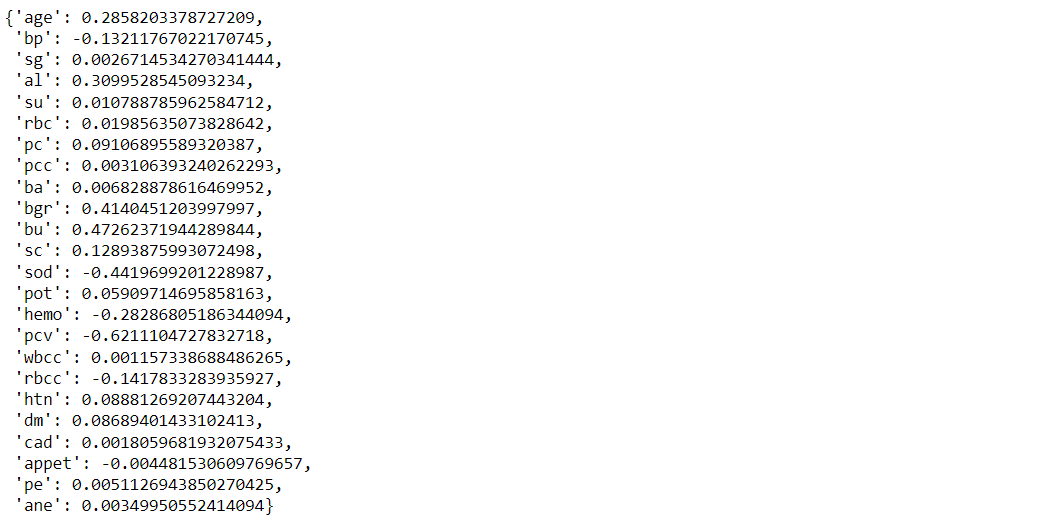
### Feature Importance

feature\_dict=dict(zip(df.columns,list(logreg.coef\_[0])))

feature\_dict

**Here we will get the coefficient from the features which will tell the weightage of each feature.**

**Output:**

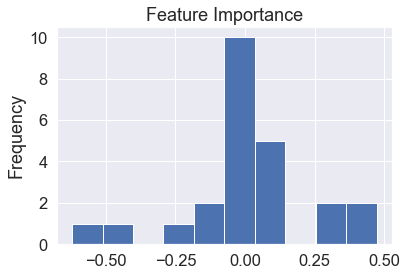


**Visualize feature importance**

feature\_df=pd.DataFrame(feature\_dict,index=[0])

feature\_df.T.plot(kind="hist",legend=False,title="Feature Importance")

**Output:**

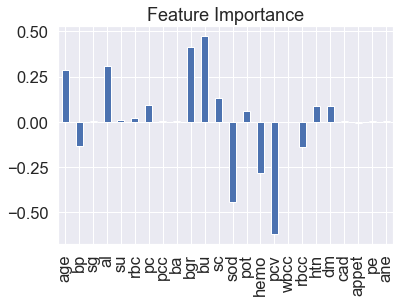


**Visualize feature importance – Transpose**

feature\_df=pd.DataFrame(feature\_dict,index=[0])

feature\_df.T.plot(kind="bar",legend=False,title="Feature Importance")

**Output:**



## Saving the model

import pickle

**# Now with the help of pickle model we will be saving the trained model**

saved\_model = pickle.dumps(logreg)

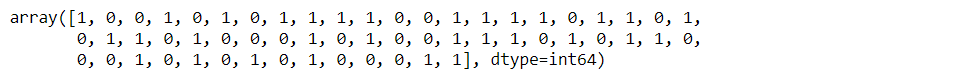
**# Load the pickled model**

logreg\_from\_pickle = pickle.loads(saved\_model)

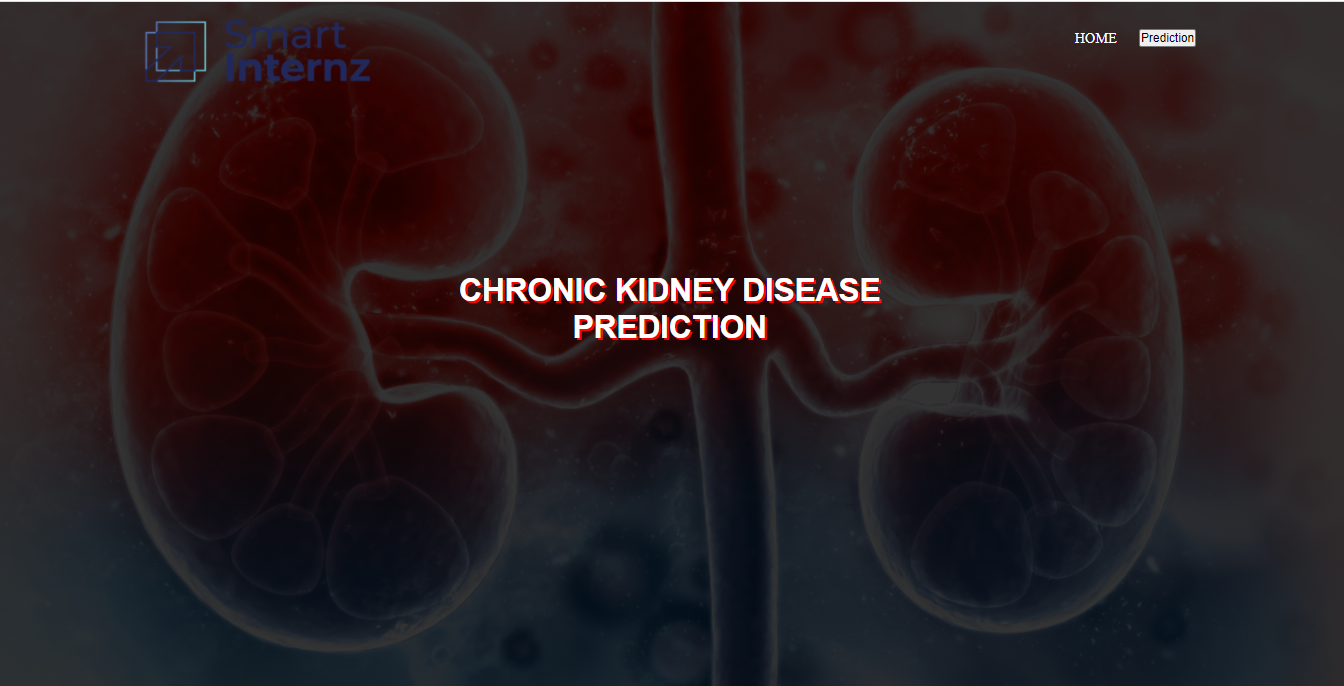
**# Now here we will load the model**

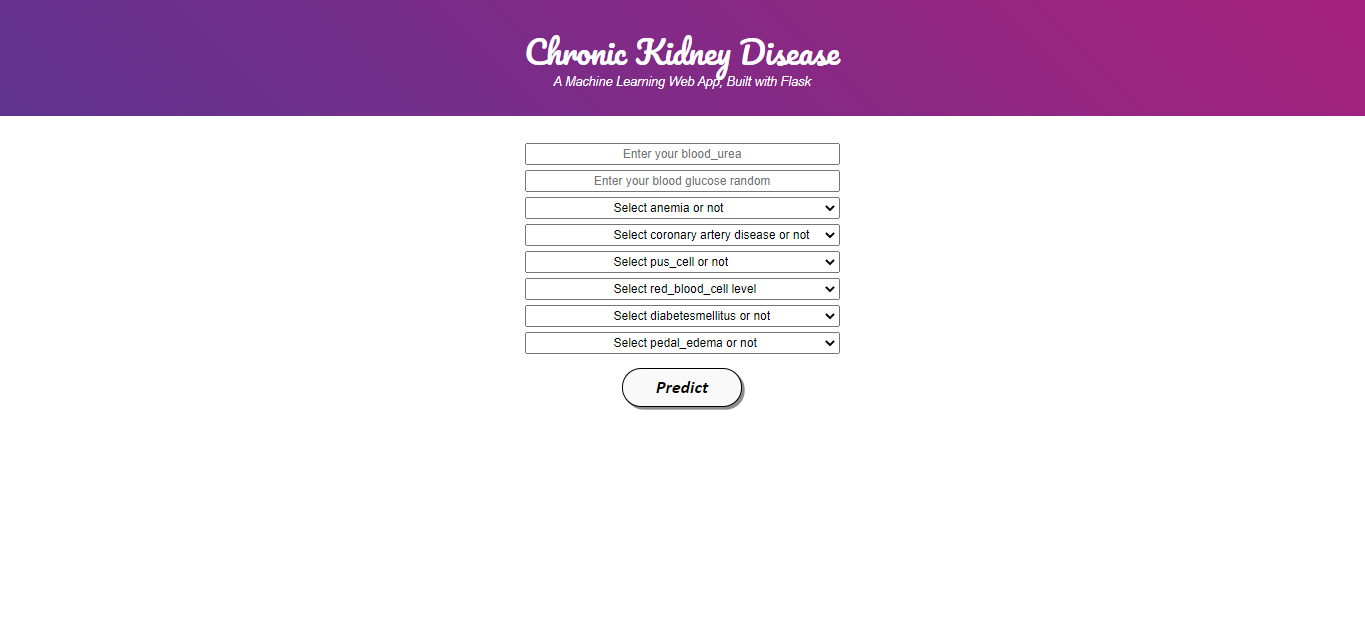
logreg\_from\_pickle.predict(X\_test)

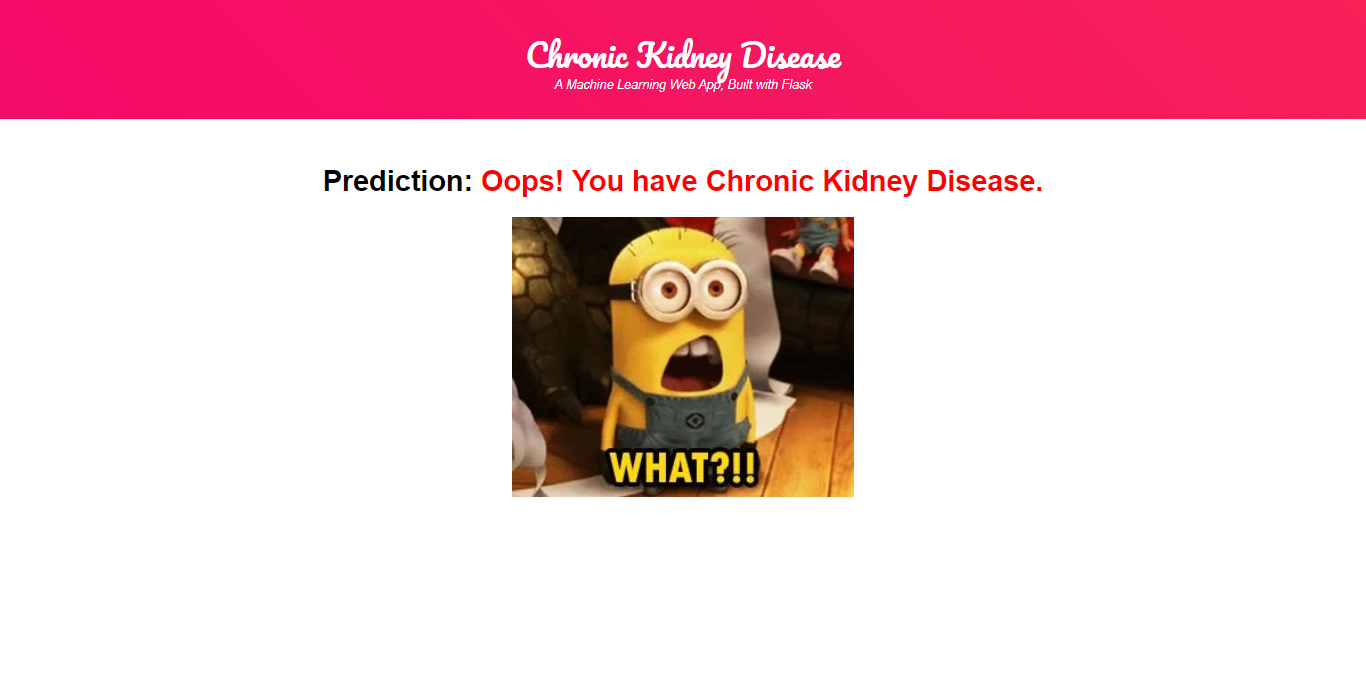
**Output:**

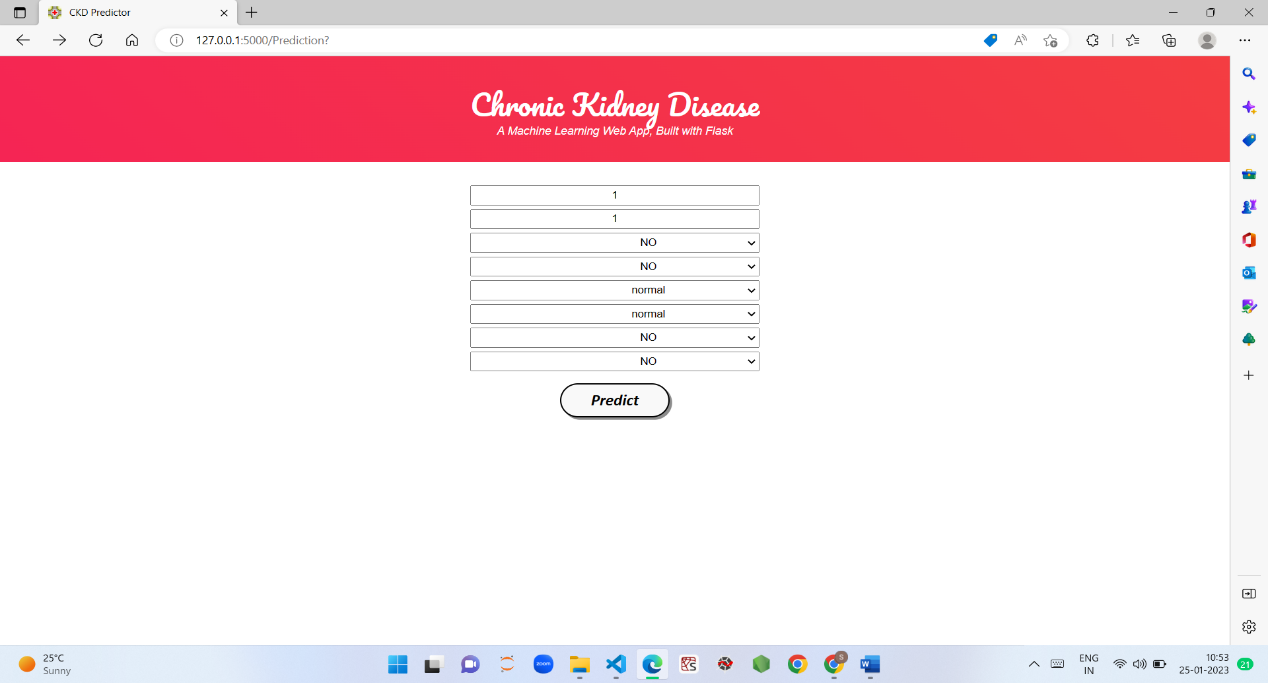


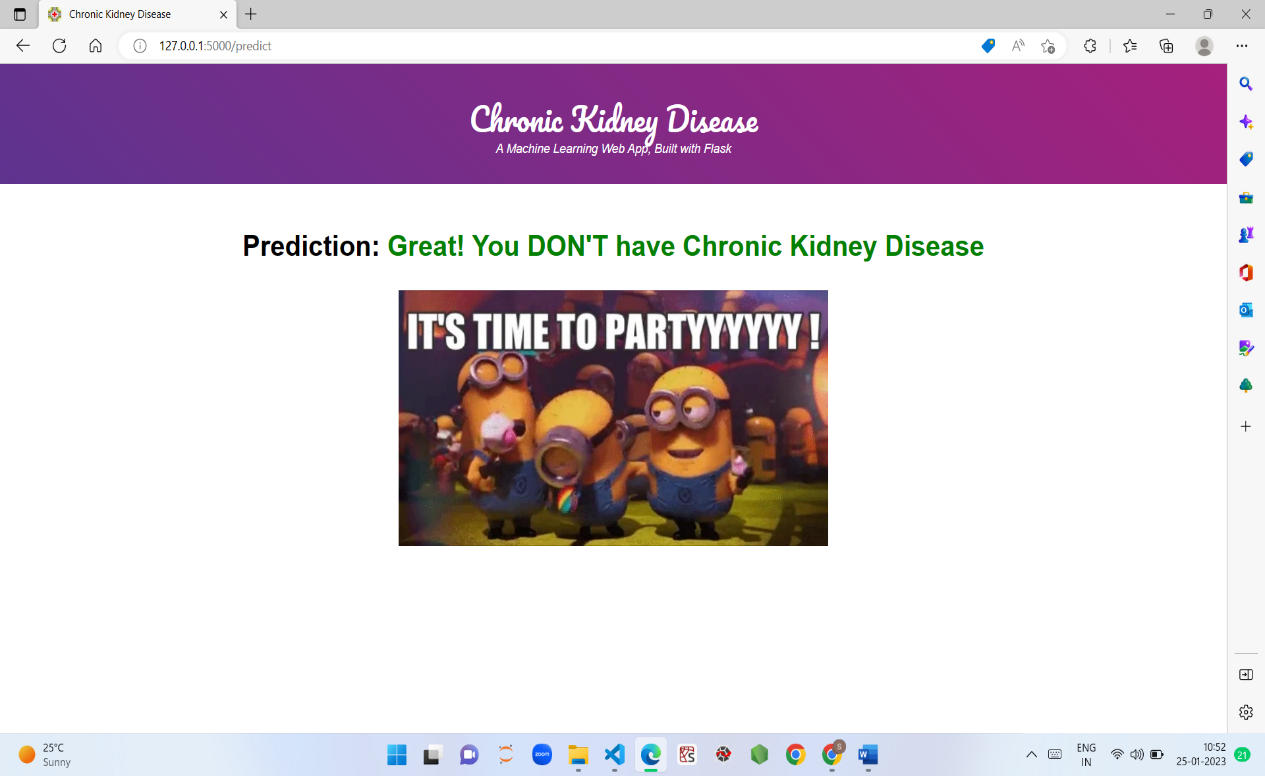
**Screen layouts**

****

****

****

****

****

**Conclusions:**

This study explored how a learning model can be used to classify the possibility of a CKD diagnosis.

Consequently, the indirect agreement with the objectives of the project, the CRISP-DM methodology was adapted to the context of the problem so that the different logically organized stages were taken; data collection, pre-processing, learning, evaluation, and selection, which allowed the construction of a model capable of classifying the possibility of a diagnosis of CKD with an accuracy of 93%. As evidenced in the results, the decision forests algorithm has obtained quite optimal results, where predictions of 93% have been obtained. Data preparation is a fundamental step in the process and the absolute precision of the model is directly dependent on this phase. Thanks to the models, we can see how changing the characteristics affects the search for the target value with a simple change of column selection or improvements in the data. The innovation of this work results from the design adjusted to the environment of the health system in Iraq and the pathology of the CKD in our country, with a methodology adapted to the case study and a production architecture proposal for the model with Microsoft Azure tools of form that allows satisfying in the future the scalability of the solution. Furthermore, this methodology could apply to clinical data of other diseases and pathologies inaccurate medical diagnoses. The development of this project allowed the author to acquire more excellent knowledge through both practical and theoretical work about current techniques for the development of machine learning.

This study has limitations, so there is a room for future research. The study did not have a significant data sample due to the restrictions of medical data and its legal effects in Iraq. Continuing with the expansion of the database (increasing the number of examples for each variable) would reduce the limited generalization error for the model and, at the same time, allow the severity of the disease to be detected. This model can be refined with increasing data size and quality. It also opens the space for a variety of studies from other disciplines, such as economic studies around the impact of obtaining a diagnosis in less time to treat the disease in early stages, reducing costs in the health system. In addition, a variety of sociological and clinical studies on the consequences of early CKD management brings about the quality of life of patients and their families. Although the validity of this research is internal, since the document corpus is private and cannot be published for other works, it will help interested professionals with machine learning to carry out their studies in the classification area.

This study presented a number of different machine learning algorithms with the intention of making a CKD diagnosis at an earlier stage. The models that are constructed using CKD patients are then trained and validated using the input parameters that were discussed earlier. Studies have been done on the associations between different factors so that the number of features can be cut down and redundant information eliminated. When applying a filter feature selection approach to the remaining attributes, it was discovered that hemoglobin, albumin, and specific gravity had the biggest impact when it comes to predicting CKD. This was the case after the method was used. This work presented a number of different machine learning algorithms with the intention of making a CKD diagnosis at an earlier stage. The original CKD dataset has been preprocessed first to validate the machine learning-based detection models. After that, the PCA has been performed to identify the most dominant features, thereby detecting CKD. The models that are constructed using CKD patients are then trained and validated using the input parameters that were discussed earlier. The accuracy of such algorithms was the primary criterion that was utilized in evaluating their overall performance.