# Chronic kidney Disease Analysis

Chronic Kidney Disease (CKD) is a major medical problem and can be cured if treated it 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

**Prerequisites:**

1. To develop this project, we need to install following software/packages:

**Anaconda Navigator:**

Anaconda Navigator is a free and open-source distribution of the Python and R

programming languages for data science and machine learning related applications. It can

be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform,

package management system. Anaconda comes with so very nice tools like JupyterLab,

Jupyter Notebook,

QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be

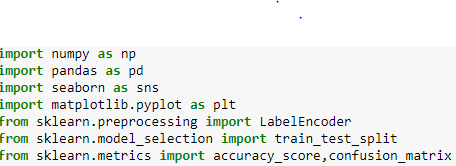
using Jupyter notebook and Visual studio code.

->To accomplish this, we must complete all the activities and tasks listed below

1. Data Collection

Download dataset/create dataset:

2.Data pre-processing:



**Pandas:** It is a python library mainly used for data manipulation.

**NumPy:** This python library is used for numerical analysis.

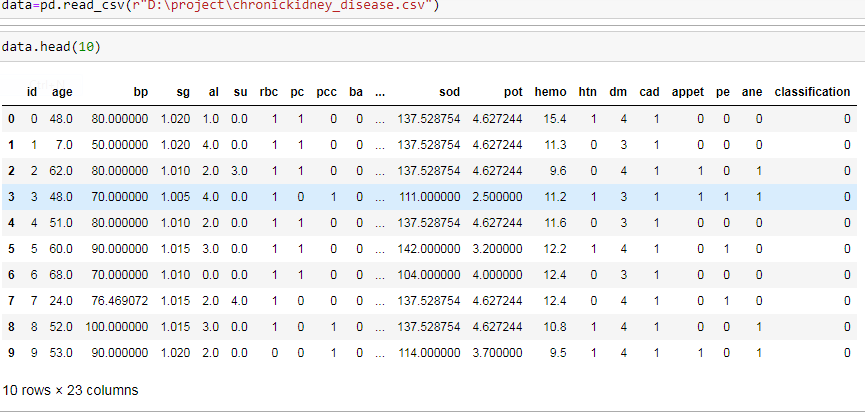
**Matplotlib and Seaborn:** Both are the data visualization library used for plotting graph which will help us for understanding the data.

**Accuracy Score:** used in classification type problem and for finding accuracy it is used.

**Train\_test\_split:** used for splitting data arrays into training data and for testing data.

**Sklearn i**s the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistence interface in Python.

Importing the dataset:

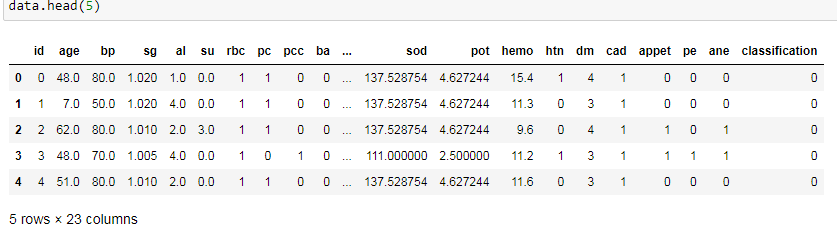


* You might have your data in .csv files, .excel files or .tsv files or something else. But the goal is the same in all cases. If you want to analyse that data using pandas, the first step will be to read it into a data structure that’s compatible with pandas.

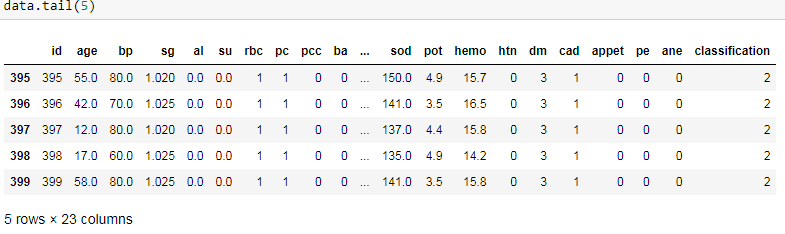
* Let’s load a .csv data file into pandas. There is a function for it, called **read\_csv().**We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).

Data visualization:

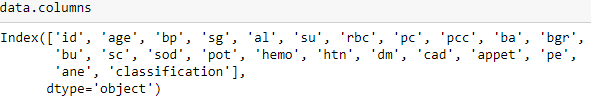
* To check first five rows of dataset, we have a function call **head( ).**



* To check last five rows of dataset, we have a function call **tail().**



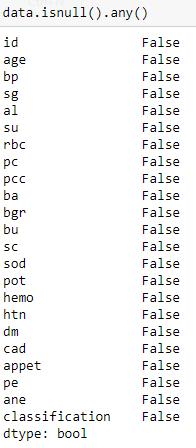
* **data.columns** will return you all the column names which are present in your data.



Taking care of Missing Data:

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. Obviously, you could remove the entire line of data but what if you are unknowingly removing crucial information? Of course we would not want to do that. One of the most common ideas to handle the problem is to take a mean of all the values for continuous and for categorical we make use of mode values and replace the missing data.

1. We will be using **isnull().any()** method to see which column has missing values.



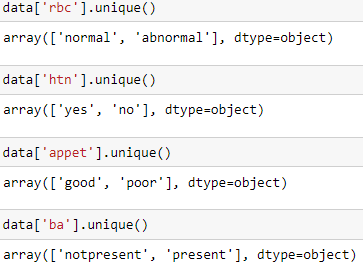
Label encoding

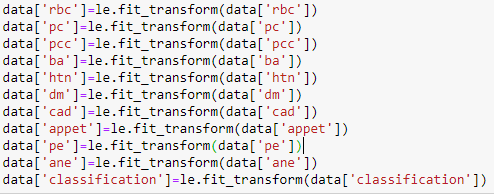
Typically, any structured dataset includes multiple columns with combination of numerical as well as categorical variables. A machine can only understand the numbers. It cannot understand the text. That’s essentially with machine learning algorithms too. We need to convert each text category to numbers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

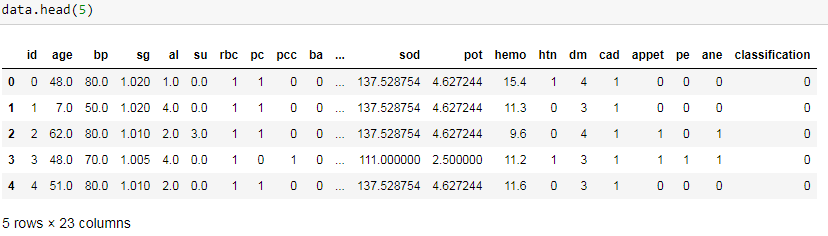
How should we handle categorical variables? There are Multiple way to handle, but will see one of it is LabelEncoding.

* **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.

As we have to convert only the text class category columns, we first select it then we will implement Label Encoding to it.







Feature scaling:

spliting data into train and test

The train-test split is a technique for evaluating the performance of a machine learning

algorithm.

**Train** **Dataset**: Used to fit the machine learning model.

**Test** **Dataset**: Used to evaluate the fit machine learning model.

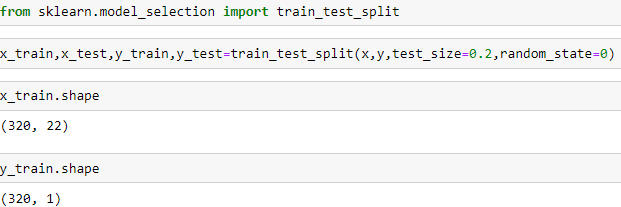
In general you can allocate 80% of the dataset to training set and the remaining 20% to test

set.

We will create 4 sets— X\_train (training part of the matrix of features), X\_test (test part of the matrix of features), Y\_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices), Y\_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices.

There are a few other parameters that we need to understand before we use the class:

1. **test\_size** — this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset
2. **train\_size** — you have to specify this parameter only if you’re not specifying the test\_size. This is the same as test\_size, but instead you tell the class what percent of the dataset you want to split as the training set.
3. **random\_state** — here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the Random\_state class, which will become the number generator. If you don’t pass anything, the Random\_state instance used by np.random will be used instead.
4. Now split our dataset into train set and test using train\_test\_split class from scikit learn library.



Model Building

**Training and testing the model:**

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:

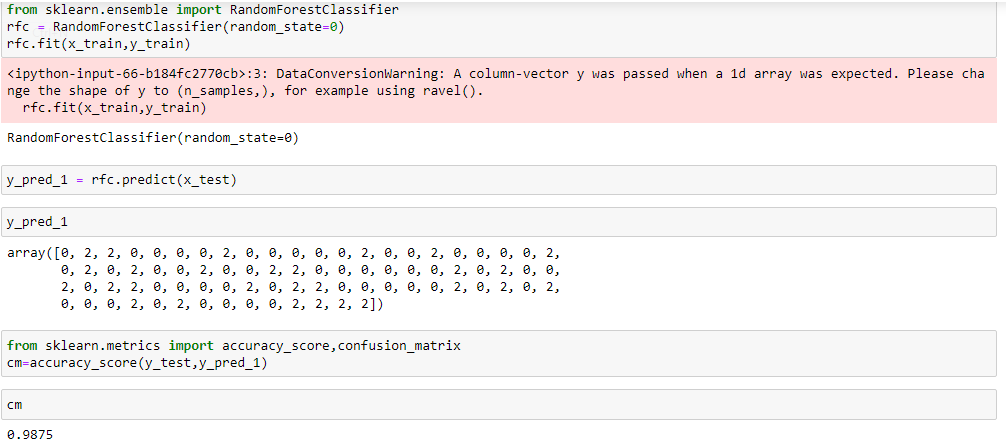
1.Random Forest Classifier

2.Support vector machine classifer

**Now we apply Random Forest Classifier to our dataset**

**Random Forest:**It is a supervised learning algorithm.It is an ensemble of decision trees usually trained with bagging method.The bagging method is combination of learning models that increases the overall result.

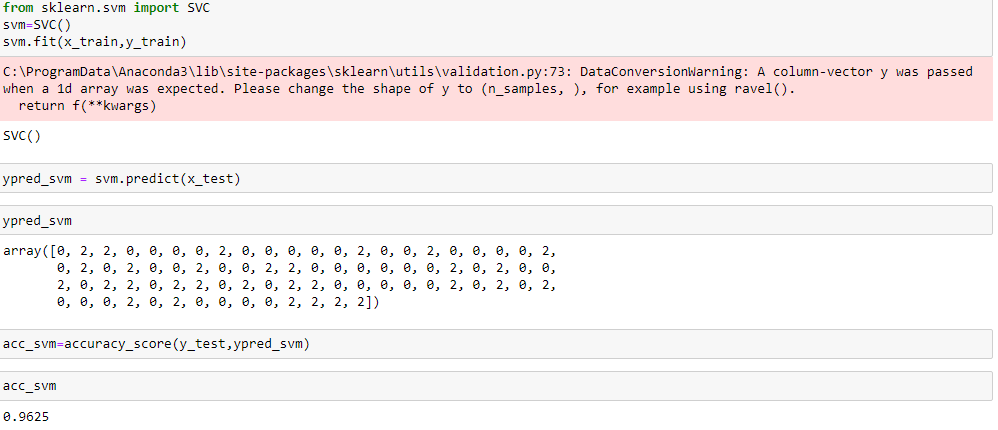
**# Using Random Forest Classifier**



**Now we will apply SVM to our dataset**

**SVM:**SVM's are a set of supervised learning methods used for classification, regression and outliers detection.The advantages of support vector machines are: Effective in high dimensional spaces.Still effective in cases where number of dimensions is greater than the number of samples.

**# Using SVM**



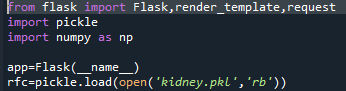
Application Building

Creating a HTML File, flask application.

1. Build python code
   1. Importing Libraries
   2. Routing to the html Page
   3. Showcasing prediction on UI
   4. Run The app in local browser

Task-1:Importing libraries

Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument Pickle library to load the model file.

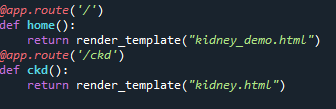


Task 2: Routing to the html Page

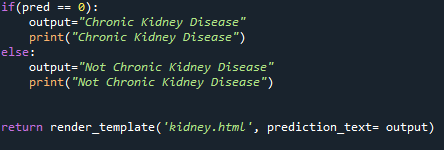
Here, declared constructor is used to route to the HTML page created earlier.

In the above example, ‘/’ URL is bound with home.html function. Hence, when the home page of the web server is opened in browser, the html page is rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Here, “kidney.html” is rendered when home button is clicked on the UI.





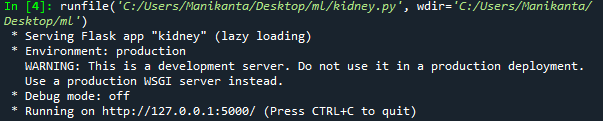


Task3: Main Function

This is used to run the application in a local host



Activity 3:Run the application



Output Screen:



