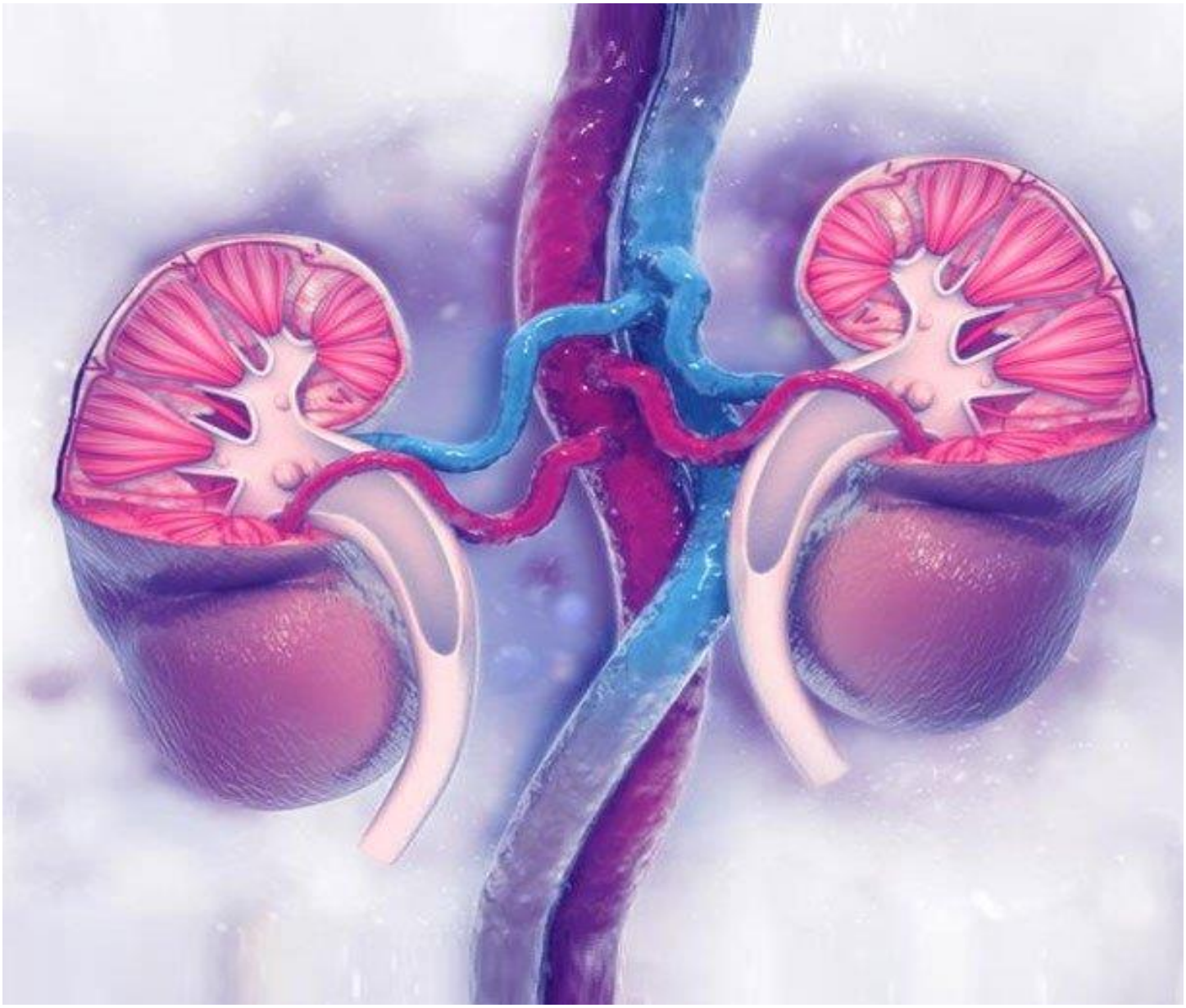


An Analytical Classification of Chronic Kidney Disease

AADERSHI MOHAN

4/2/22

Project on Data Analysis

ABSTRACT

Analytical Classification is the method of classifying the data in such a way that it got more meaning and can provide useful conclusions. In this project, we did data pre-processing which include data cleaning followed by the treatment of missing values, and then we worked on two classifiers namely, Decision Table and Random Forest classifiers. After using this classifier we have observed that the Decision table gives an accuracy of 99.69 while the Random Forest classifier gives the accuracy of 100% for the kidney dataset.

1. INTRODUCTION

Chronic Kidney Disease or Long-term Kidney Disease occurs after the loss of Kidney function which happens gradually over a long time i.e., months to years. This is why the symptoms of kidney disease are not seen at the initial stages. But after some time a few symptoms like a tired feeling, leg swelling, loss of appetite, vomiting, and confusion can be seen. Problems like high blood pressure, diabetes, glomerulonephritis, etc. can lead to kidney disease or it can also happen if there is any kidney history found in the family. Once a disease gets diagnosed the initial treatments advised are: to control and lower the blood sugar, blood pressure, and cholesterol. Nearly 1 in 3 people with diabetes and 1 in 5 people with high blood pressure have kidney disease, **[4]**

The diagnosis of the disease is done with the help of blood tests and increasing the glomerular filtration rate (eGFR). GFR rating helps to know how severe the damage is. Fig (a) is the GFR table to check the condition of one's kidney as per their GFR rating.

STAGES	RANGE (mL/min)	REMARKS
STAGE 1	90-100	Healthy kidneys or high GFR
STAGE 2	60-89	Kidney damage is a mild or mild decrease in GFR
STAGE 3	30-59	Moderate decrease in GFR
STAGE 4	15-29	Severe decrease in GFR
STAGE 5	<15	Kidney failure

Fig (a), GFR Rating is divided into 5 stages

If a person reaches stage 5, he requires a kidney transplant. Later in this report, we will study and analyze a kidney dataset of patients.

1.1. **Definitions**

Average- The average of any sequence can be calculated by the sum of all the observations divided by the total number of observations. **[2]**

Median- The median of any sequence is the average of two middle values if total observations are even after arranging them in ascending order in number otherwise the middle number is the median.

Quantile- The points that divide the continuous interval of a probability distribution into 25%, 50% (median), and 75% are called quantiles.

Outliers- These are the points or values which does not follow the trend and show abnormal properties.

Kappa Statistic- The measurement of how closely the instances are classified by the machine learning classifier suits the data labeled as ground truth, controlling the accuracy of a random classifier as measured by expected accuracy. **[8]**

TP Rate- This means true positives or correctly classified instances as a given class.

FP Rate- This means false positives or falsely classified instances as a given class.

Precision- Ratio of true positives to total instances classified as that given class.

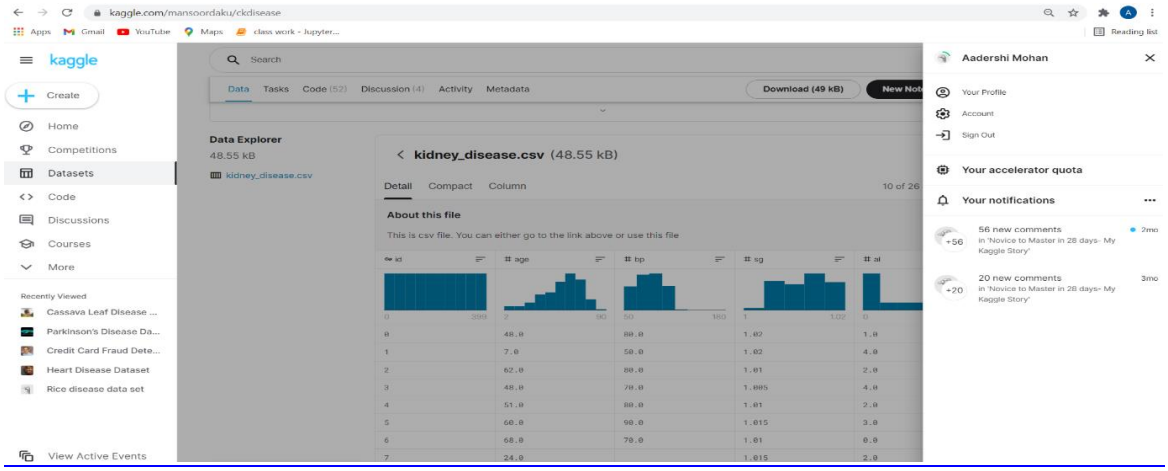
Recall- Proportion of instances that are truly divided by the actual total of that class.

F-Measure- This is a combined measurement for precession and recall calculated as

(2*Precession*Recall/ Precession+ Recall)

1.2. Data Availability

The dataset used in this report is taken from Kaggle. Chronic Kidney Disease dataset **[1]**.



1.3. Abbreviations

Bp- Bp stands for Blood Pressure measured in mm/Hg. This is the pressure that the blood generates when it pushes the arteries wall.

Sg- Sg stands for specific gravity and compares the density of urine to the density of the water.

Al- Al stands for albumin which is a protein present in the blood and if the kidney is healthy it does not pass from the blood into the urine.

Su- Su stands for sugar.

Rbc- Rbc stands for Red Blood Cells. Rbc in the report tells us the count of these cells present in the blood and they also contain hemoglobin.

PC- PC stands for Pus Cells. When kidney filtration units get damaged then the white blood cells (pus cells) are the signs of infection.

PCC- PCC stands for Pus Cell Clumps. These are the huge amount of white blood cells in the urine which lead it to appear cloudy.

Ba- Ba stands for Bacteria. Sometimes bacteria can cause kidney infections.

Bgr- Bgr stands for Blood Glucose Random measured in mg/dL. If the test shows 200mg/dL or more then there are chances that the person may be diabetic.

Bu- Bu stands for blood Urea and is measured in mg/dL. If blood urea is higher it can be an indication that the kidney is not functioning properly.

Sc- Sc stands for Serum Creatinine and is measured in mg/dL. Creatinine is a waste present in blood that reaches there from muscles. It informs us how well are the kidneys working.

Sod- Sod stands for Sodium and is measured in mEq/L. Too low or too high sodium can lead to kidney related problems.

Pot- Pot stands for Potassium and is measured in mEq/L. Higher potassium levels can impact heart rates.

Hemo- Hemo stands for hemoglobin and is measured in gms. Hemoglobin helps RBCs to carry oxygen from the lungs to the whole body. Low EPO levels can drop RBC count and lead to anemia.

PCV- PCV stands for Packed Cells Volume. It measures the proportion of the cells in 100 ml of the blood. Ex. PCV of 35% refers to 35 mL in 100 mL of blood.

WBC- WBC stands for White Blood Cells and is measured in cells/cmm. A Higher WBC count is the indicator of Ckd.

Htn- Htn stands for Hypertension. It is higher blood pressure caused by kidney disease.

Dm- Dm stands for Diabetes Mellitus. Kidney filtering units consist of many small blood vessels. These can become narrow if the sugar level increases over some time.

Cad- Cad stands for Coronary Artery Disease. The blood vessel that sends blood and oxygen to the heart is Cad.

Appet- Appet stands for Appetite.

Pe- Pe stands for Pedal Edema. The decline in albumin (protein) can cause fluid accumulation or edema.

Ane- Ane stands for Anemia. A low count of red blood cells is defined as anemia.

Ckd- Ckd stands for Chronic Kidney Disease or Long-term kidney disease.

WEKA - Waikato Environment for Knowledge Analysis is the graphic-user interface that consists of the tools for data cleaning (data pre-processing), classification, and a few more.

1.4. Software Used

Software used in this project includes Python**[10]**, and Microsoft Excel, in which we store our dataset, and make our dataset ready for analysis by preprocessing and cleaning the dataset, furthermore, we have also used the Jupyter notebook, where we read, study, clean, and analyze our data.

2. DATA PREPROCESSING

Data Preprocessing is the part before data analysis where we process data, read, study it, and make it ready for analysis.

2.1. **Retrieving Dataset**

To retrieve the dataset into the '*jupyter notebook'* **[7]** we have used the *'pandas'* **[3]** library as shown in Fig1,

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns

In [2]: kidney = pd.read_csv('kidney_disease_Original.csv')

In [3]: kidney

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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Fig 1: Importing dataset from CSV file to jupyter notebook.

2.2. **Dataset Info**

As the dataset has been already retrieved, now, we use the '*info'* command to know the basic information about the dataset.

```
In [9]: kidney.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 26 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              400 non-null    int64
 1   age             391 non-null    float64
 2   bp              388 non-null    float64
 3   sg              353 non-null    float64
 4   al              354 non-null    float64
 5   su              351 non-null    float64
 6   rbc             248 non-null    object
 7   pc              335 non-null    object
 8   pcc             396 non-null    object
 9   ba              396 non-null    object
 10  bgr             356 non-null    float64
 11  bu              381 non-null    float64
 12  sc              383 non-null    float64
 13  sod             313 non-null    float64
 14  pot             312 non-null    float64
 15  hemo            348 non-null    float64
 16  pcv             330 non-null    object
 17  wc              295 non-null    object
 18  rc              270 non-null    object
 19  htn             398 non-null    object
 20  dm              398 non-null    object
 21  cad             398 non-null    object
 22  appet           399 non-null    object
 23  pe              399 non-null    object
 24  ane             399 non-null    object
 25  classification  400 non-null    object
dtypes: float64(11), int64(1), object(14)
memory usage: 81.4+ KB
```

Fig 2: The information about Dtype and null values present in columns

Comparing Fig 1 and 2, we have observed that in Fig 2 the Dtype of *'pcv'*, *'wc'*, and *'rc'* is the '*object'* type but while observing Fig 1 we can see that these columns consist of numerical values. So, we need to check this.

2.3. **Datatype Error**

After looking at the CSV file **[9]** we have observed that,

P	Q	R	S
hemo	pcv	wc	rc
7.5	27		
9.8			
15	48		
	?		
10.9	37		
15.6	52	6900	6
15.2	44	8300	5.2

Fig 3: A part of numerical data which contains a string

In Fig 3, Column Q has a string value which is changing the datatype to the whole column to '*object*' datatype. To fix this we will remove this question mark from here.

Similarly in Column R,

Q	R	S	T
pcv	wc	rc	htn
14	6300		yes
29	6400	3.4	yes
			no
36	6200	4	no
34	7100	3.7	yes
30			yes
40	11800	5	yes
31	9400	3.8	yes
29	5500	3.7	yes

Fig 4: Column R has a string value

The value **6200** in column R is written as a string value which makes this column an '*object'* type. So, we need to convert it either into '*int*' type or '*float*' type.

2.4. **Spacing Error**

Spacing error occur when some elements got different alignment than other

T	U	V	W
htn	dm	cad	appet
no	no	no	good
yes	yes	no	poor
no	no	no	poor
yes	yes	no	poor
no	no	no	good
yes	yes	no	poor
yes	yes	no	good
yes	yes	yes	poor
yes	yes	no	poor

Fig 5: Spacing error in Column U

In fig 5, some cell values are wrongly spaced, which should be fixed or they will act as new variables.

Row Labels	Count of classification
ckd	59.79%
ckd	0.53%
notckd	39.68%
Grand Total	**100.00%**

Fig 6: The % classification of ckd and not ckd

For example:

In fig 6, **ckd** is appearing twice because of spacing error it's taking the same variable separately. Let's fix this by using the **TRIM** function.

Row Labels	% classification
ckd	62.50%
notckd	37.50%
Grand Total	**100.00%**

Fig 7: % classification of ckd and not ckd

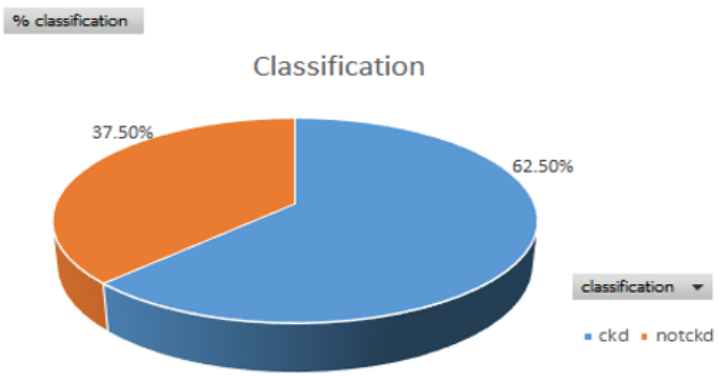


Fig 8: The % classification of kidney disease into ckd and not ckd

From fig 8, we observe that **62.50%** of patients have chronic kidney disease, and **37.50%** with non-chronic. This concludes that the ***symptoms of kidney disease are discoverable at higher stages***.

2.5. **Categorical and Numerical Datatype** – Checking if the errors have been removed

```
In [26]: def col_catego_numeri(kidney):
             catego_col = [col for col in kidney.columns if kidney[col].dtype== object]
             numeri_col = [col for col in kidney.columns if kidney[col].dtype!= object]
             return catego_col,numeri_col

In [27]: catego_col,numeri_col = col_catego_numeri(kidney)

In [28]: catego_col

Out[28]: ['rbc',
          'pc',
          'pcc',
          'ba',
          'htn',
          'dm',
          'cad',
          'appet',
          'pe',
          'ane',
          'classification']

In [29]: numeri_col

Out[29]: ['id',
          'age',
          'bp',
          'sg',
          'al',
          'su',
          'bgr',
          'bu',
          'sc',
          'sod',
          'pot',
          'hemo',
          'pcv',
          'wc',
          'rc']
```

Fig 9: Column names based on categorical and numerical data type

Comparing fig 1, 2, and 9 we can say that all errors have been removed. And all data types are now well defined as per categorical and non-categorical data types. So, we will proceed further.

2.6. **Dimensions of the dataset**

To find the exact number of rows and columns, we will use the **Shape** function.

```
In [4]: kidney.shape
Out[4]: (400, 26)
```

Fig 10: Dimensions of kidney dataset

Fig 10 shows, that this dataset has **400** rows and **26** columns.

2.7. Columns Names in Kidney Dataset

In fig 1, all column names are not visible, let's find them out.

```
In [11]: kidney.columns

Out[11]: Index(['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')
```

Fig 11: Names of all the columns present in dataset

Fig 11, display the names of **all columns**.

2.8. Description Table

Moving ahead let's find the numerical description of our dataset that provides us the basic statistics such as average, count, quantiles, min, and max.

```
In [17]: kidney.describe()
```

Out[17]:

	id	age	bp	sg	al	su	bgr	bu	sc	sod	pot	hemo
count	400.000000	391.000000	388.000000	353.000000	354.000000	351.000000	356.000000	381.000000	383.000000	313.000000	312.000000	348.000000
mean	199.500000	51.483376	76.469072	1.017408	1.016949	0.450142	148.036517	57.425722	3.072454	137.528754	4.627244	12.526437
std	115.614301	17.169714	13.683637	0.005717	1.352679	1.099191	79.281714	50.503006	5.741126	10.408752	3.193904	2.912587
min	0.000000	2.000000	50.000000	1.005000	0.000000	0.000000	22.000000	1.500000	0.400000	4.500000	2.500000	3.100000
25%	99.750000	42.000000	70.000000	1.010000	0.000000	0.000000	99.000000	27.000000	0.900000	135.000000	3.800000	10.300000
50%	199.500000	55.000000	80.000000	1.020000	0.000000	0.000000	121.000000	42.000000	1.300000	138.000000	4.400000	12.650000
75%	299.250000	64.500000	80.000000	1.020000	2.000000	0.000000	163.000000	66.000000	2.800000	142.000000	4.900000	15.000000
max	399.000000	90.000000	180.000000	1.025000	5.000000	5.000000	490.000000	391.000000	76.000000	163.000000	47.000000	17.800000

Fig 12: Description table

Here, 25%, 50% (median), and 75% are three quantiles.

3. DATA CLEANING

In this step, we will clean our dataset by fixing the null values which is the treatment of missing data. **[5]**

3.1. NULL Values

Treatment of null values is necessary before analyzing the dataset. So, let's find the null values present in our dataset. For that, we will use the 'IsNull' function and the 'sum' attribute to get the count of null values present in each column.

```
In [20]: kidney.isnull().sum()
Out[20]: id                   0
         age                  9
         bp                  12
         sg                  47
         al                  46
         su                  49
         rbc                152
         pc                  65
         pcc                  4
         ba                   4
         bgr                 44
         bu                  19
         sc                  17
         sod                 87
         pot                 88
         hemo                52
         pcv                 70
         wc                 105
         rc                 130
         htn                  2
         dm                   2
         cad                  2
         appet                1
         pe                   1
         ane                  1
         classification       0
         dtype: int64
```

Fig 13: Null values present in dataset

These numbers tell us about the empty cells present in our dataset. For more clarity let's plot a supportive graph.

Fig 14: Bar Graph of empty cells present in each column

There are **400** patients but for **152** patients their **RBC** report is not available which is **38%** of the total data. Now, filling in this much data can lead to complications. So we will drop this column.

Now, the **RC** column also has a higher amount of missing data. For this, we will check the correlation of **RC** with other factors to check how important the **RC** factor is in terms of calculating chronic kidney disease.

3.2. **Correlation between RC and Other Factors**

From fig 15, it is observed that **RC** is not related to any other factor either directly or inversely. So, we can neglect this factor while analyzing others.

Row Labels	CORREL
AGE	-0.268896285
AL	0.122090983
BGR	0.2449922
BP	0.159479693
BU	0.196984871
HEMO	-0.192928339
PCV	-0.24395094
POT	0.05837712
SC	0.132530865
SG	-0.191096381
SOD	-0.100045983
SU	0.220866325
WC	0.118340506
Grand Total	**0.256744635**

Fig 15: Correlation between RC and other factors

Now, because analyzing further we need to clean our data. So, for that, we will use cleaning methods.

3.3. Treatment of Missing Data

1. Delete all rows with empty cells.
2. Replacing the empty cells with Average values.
3. Replacing the empty cells with Median.

3.3.1. Deleting all rows with empty cells

As we have eliminated **RBC** and **RC** already. Now working on left data and proceeding by deleting the entire row containing empty cells. Then retrieving new data we have,

```
In [35]: kidney = pd.read_csv("Deleted.rows_kidney.csv")
```

```
In [38]: kidney.head(3)
```

Out[38]:

	id	age	bp	sg	al	su	pc	pcc	ba	bgr	...	hemo	pcv	wc	htn	dm	cad	appet	pe	ane	classification
0	3	48	70	1.005	4	0	abnormal	present	notpresent	117	...	11.2	32	6700	yes	no	no	poor	yes	yes	ckd
1	9	53	90	1.020	2	0	abnormal	present	notpresent	70	...	9.5	29	12100	yes	yes	no	poor	no	yes	ckd
2	11	63	70	1.010	3	0	abnormal	present	notpresent	380	...	10.8	32	4500	yes	yes	no	poor	yes	no	ckd

And the dimensions of this dataset are

```
In [37]: kidney.shape
Out[37]: (193, 24)
```

Now, we are left with **193** rows only.

To see how much deviation is taken place in the results, we will again check the results of the percentage of CKD and not CKD.

Row Labels	% classification
ckd	40.41%
not ckd	59.59%
Grand Total	**100.00%**

Fig 16: % Classification of ckd and not ckd

Originally CKD and not CKD were 62% and 38% resp. but now they are approx. 40% and 60% which is a huge error. So, this cannot be considered an appropriate analysis method.

3.3.2. Replacing the empty cells with an average value

Before proceeding with this method we need to check if there are outliers in this dataset.

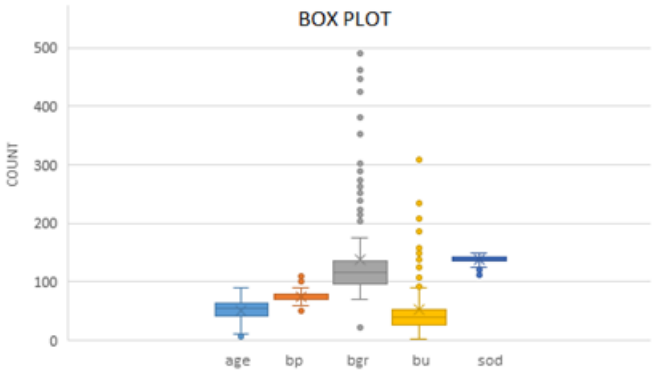


Fig 17: Box Plot of age, bp, bgr, bu, and sod

Here it's visible that there are so many outliers. Thus we cannot fill empty cells with the average value.

3.3.3. Replacing the empty cells with Median

Firstly, we will replace the empty cells of a numerical data type with its column median. And then will delete the entire row with empty cells in categorical data type.

After cleaning the data we calculate that the original dataset is required to **(327, 24)**.

Now, if we consider the dataset of **400** patients as a population and **327** as a sample dataset. For this let's find the margin of error.

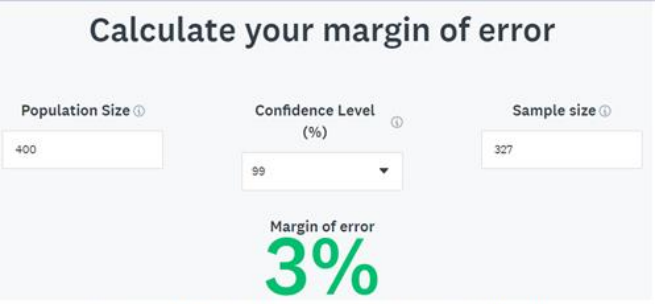


Fig 18: The Marginal Error after the data size is reduced.

In fig 18, the expected marginal error is **3%** which means that our result can contradict **±3%** of the original conclusion.

Now, we will again calculate the percentage distribution of CKD and not CKD patients.

Row Labels	Classification
ckd	59.20%
Not ckd	40.80%
Grand Total	**100.00%**

Fig 19: % classification of ckd and not ckd

Here, the percentage of CKD and not CKD is **59.2%** and **40.8%** which has an error of ∓ **2.8%**. This is nearly close to the original dataset.

Thus, we will proceed further with this dataset, which we have obtained after replacing the empty cells with the Median value for non-categorical data and further deleting the entire rows having any empty cells for categorical data, say Median_kidney dataset.

4. ANALYSIS

Importing Median_kidney dataset into jupyter file and studying the same.

Let's begin with retrieving this modified dataset.

```
data = pd.read_csv('Median_kidney.csv')
data
```

	id	age	bp	sg	al	su	pc	pcc	ba	bgr	...	hemo	pcv	wc	htn	dm	cad	appet	pe	ane	classification
0	0	48	80	1.020	1	0	normal	notpresent	notpresent	121	...	15.4	44	7800	yes	yes	no	good	no	no	ckd
1	1	7	50	1.020	4	0	normal	notpresent	notpresent	121	...	11.3	38	6000	no	no	no	good	no	no	ckd
2	2	62	80	1.010	2	3	normal	notpresent	notpresent	423	...	9.6	31	7500	no	yes	no	poor	no	yes	ckd
3	3	48	70	1.005	4	0	abnormal	present	notpresent	117	...	11.2	32	6700	yes	no	no	poor	yes	yes	ckd
4	4	51	80	1.010	2	0	normal	notpresent	notpresent	106	...	11.6	35	7300	no	no	no	good	no	no	ckd
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
322	395	55	80	1.020	0	0	normal	notpresent	notpresent	140	...	15.7	47	6700	no	no	no	good	no	no	notckd
323	396	42	70	1.025	0	0	normal	notpresent	notpresent	75	...	16.5	54	7800	no	no	no	good	no	no	notckd
324	397	12	80	1.020	0	0	normal	notpresent	notpresent	100	...	15.8	49	6600	no	no	no	good	no	no	notckd
325	398	17	60	1.025	0	0	normal	notpresent	notpresent	114	...	14.2	51	7200	no	no	no	good	no	no	notckd
326	399	58	80	1.025	0	0	normal	notpresent	notpresent	131	...	15.8	53	6800	no	no	no	good	no	no	notckd

327 rows × 24 columns

Fig 20: Importing Median kidney dataset into Jupyter Notebook from CSV

From fig 20, we can also note that the shape of this dataset is (327, 24).

Since the names of all the columns are not visible clearly. So using columns function to retrieve them.

```
data.columns
```

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

Fig 21: Names of all the columns present in this dataset.

Now, we'll use describe function to get some basic statistical information about this dataset.

```
data.describe()
```

	id	age	bp	sg	al	su	bgr	bu	sc	sod	pot	hemo	
count	327.000000	327.000000	327.000000	327.000000	327.000000	327.000000	327.000000	327.000000	327.000000	327.000000	327.000000	327.000000	327.00
mean	205.831804	50.428135	76.422018	1.017416	1.033639	0.422018	145.113150	54.024771	2.541590	138.009174	4.618349	12.725994	39.4
std	117.556536	17.193955	13.511386	0.005727	1.375319	1.065006	75.462639	48.928056	3.649109	6.569817	3.104435	2.667499	8.22
min	0.000000	2.000000	50.000000	1.005000	0.000000	0.000000	22.000000	1.500000	0.400000	104.000000	2.500000	3.100000	9.00
25%	101.500000	41.000000	70.000000	1.010000	0.000000	0.000000	102.000000	26.000000	0.900000	135.000000	4.000000	10.900000	34.00
50%	212.000000	54.000000	80.000000	1.020000	0.000000	0.000000	121.000000	40.000000	1.200000	138.000000	4.400000	13.000000	41.00
75%	309.000000	63.000000	80.000000	1.020000	2.000000	0.000000	148.000000	55.000000	2.350000	141.000000	4.800000	14.800000	45.00
max	399.000000	90.000000	180.000000	1.025000	5.000000	5.000000	490.000000	391.000000	32.000000	163.000000	47.000000	17.800000	54.00

Fig 22: The basic statistical explanation of the given dataset

4.1. Patients reported in various age groups.

Plotting a pivot table as shown Fig23, and chart in Fig 24, we get,

Age Group	Number of patients
1-10	9
11-20	12
21-30	25
31-40	35
41-50	69
51-60	75
61-70	69
71-80	31
81-90	2
Grand Total	**327**

Fig 23: Pivot table to see the number of patients in various age groups

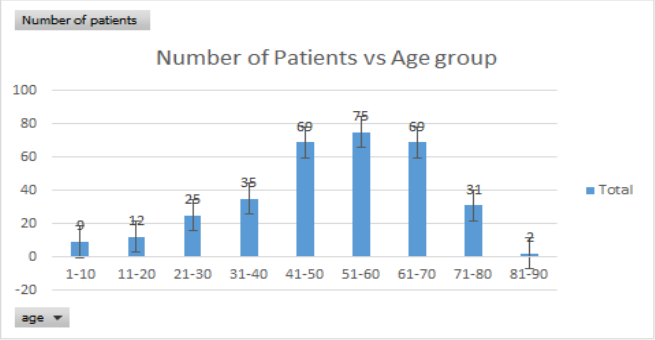


Fig 24: Number of patients present in various age group

In Fig 24, we can observe that number of patients are maximun between age group 51-60.

4.2. The number of patients with PCC count

PCC	Count
notpresent	286
present	41
Grand Total	**327**

Fig 25: Pivot table for PCC count

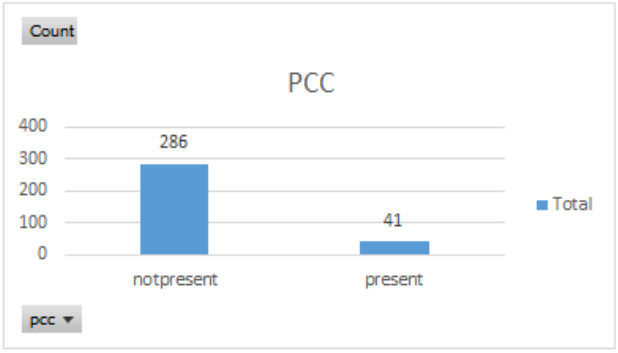


Fig 26: Graph representing PCC count

There are 41 patients which have PCC present.

4.3. Patients reported with bacteria

Bacteria	Count
notpresent	306
present	21
Grand Total	**327**

Fig 27 : Pivot table for Bacteria Count

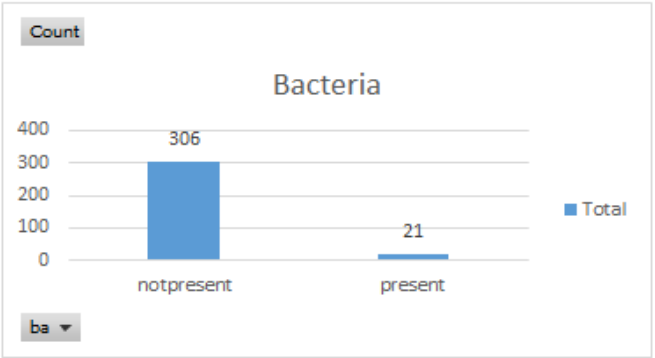


Fig 28: Graph representing Bacteria status

From Fig 28, there are 21 patients reported with bacteria.

4.4. Patients whose blood pressure is greater than 120

Creating a pivot table to check the blood pressure of patients

BP	Number of patients
<60	5
60-69	59
70-79	90
80-89	105
90-99	43
100-109	20
110-120	4
>120	1
Grand Total	**327**

Fig 29: Pivot table representing BP of patients

In the last row of the pivot table shown in Fig29, we observe that there is only 1 patient whose Blood pressure is greater than 120.

4.5. Number of patients who require kidney transplants

Row Labels	Count of sc
0-5	288
5-10	22
>10	17
Grand Total	**327**

Fig 30: Pivot table for sc count

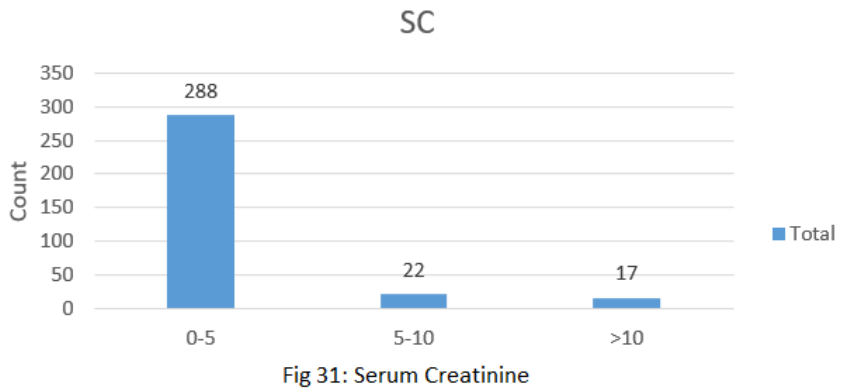


Fig 31: Serum Creatinine

In Fig31, above 10, more than 90% of kidney function has already vanished so the patients whose serum creatinine is more than 10 can't survive on drugs thus they need kidney transplants. Thus, 17 patients required kidney transplants.

4.6. Checking if serum creatinine is responsible for chronic kidney disease.

We know that the normal creatinine lies between 0.5 to 1.4 mg/dL. Filtering only not ckd patients and checking if their serum creatinine is still high.

Row Labels	Count of sc
notckd	133
0.4-0.6	19
0.6-0.8	29
0.8-1	25
1-1.2	60
Grand Total	133

Fig 32: sc of notckd patients

As we can see, all patients have their ckd less than 1.4. Thus, If serum creatinine of any person is high this means that they are suffering from chronic kidney disease.

4.7. Hypertension, diabetes mellitus, and anemia in non-ckd patients

Row Labels	Count of htn	Count of dm	Count of ane
notckd	133	133	133
no	133	133	133
Grand Total	133	133	133

Fig 33: Htn, dm, and ane in notckd patients

It is observed that not ckd patients do not have hypertension, diabetes mellitus, and anemia.

4.8. Average serum creatinine in various age groups

Age group	Average of sc
1-10	0.833333333
11-20	1.6
21-30	1.152
31-40	2.005714286
41-50	2.320289855
51-60	3.121333333
61-70	3.234782609
71-80	2.658064516
81-90	2.8
Grand Total	2.541590214

Fig 34: Average SC in various age groups

In the thirties, one should get their serum creatinine test done because the serum creatinine of the patients whose age is greater than 30 is high and this can lead to chronic kidney disease which can lead to severe kidney problems.

5. CLASSIFIERS

In machine learning, a classifier is an algorithm that self-assigns the data points to a particular range of categories or classes.

5.1. Decision Table

This is one of the simplest methods to classify things. Decision Table is also called a Boolean-based classifier because it takes responses only in Yes or No answers.

Example: In the kidney dataset, we select the factors responsible for not ckd

Id	Age	Htn	Dm	Cad	Pe	Ane	Classification
250	40	No	No	No	No	No	Not ckd
269	25	No	No	No	No	No	Not ckd
259	35	No	No	No	No	No	Not ckd

Here, we can see that, if all the answers are 'No' this means the disease is not Chronic.

Now, let's cross-check the result in WEKA.

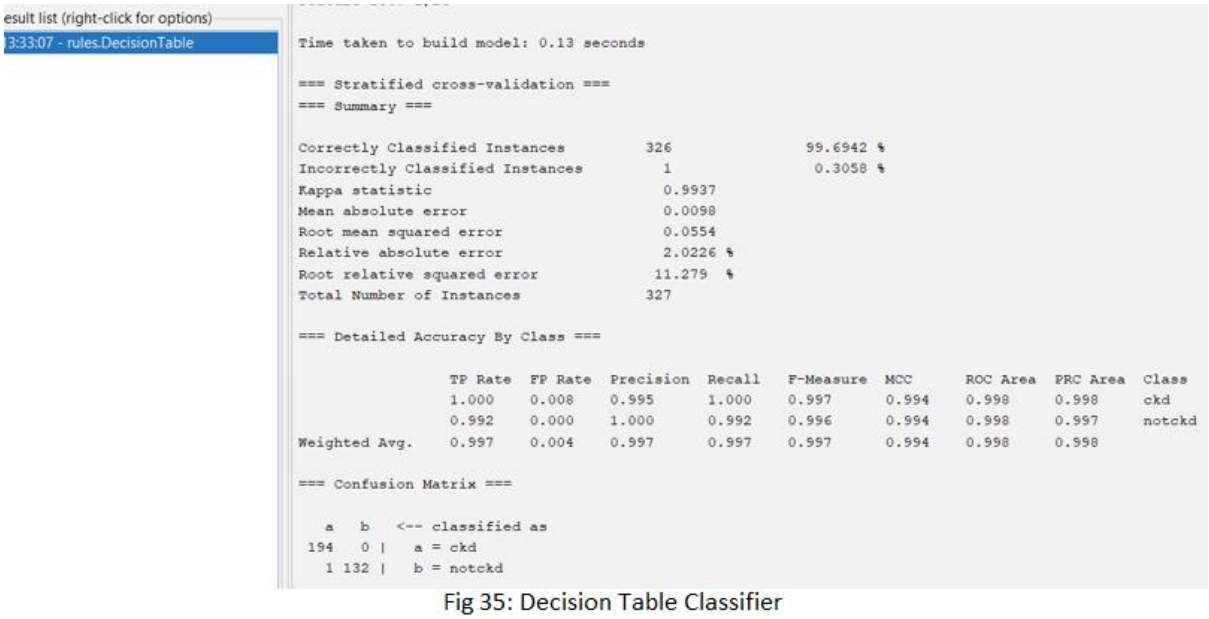


Fig 35: Decision Table Classifier

In fig 35, we see that there is one incorrectly classified instance which means there is an exception that does not follow the trend. That particular case satisfies all conditions to have a non-chronic disease but still has a chronic disease.

Like, if we see the report of patient Id 67, we observe that does not suffer from any of the factors but still has a chronic disease.

Id	Age	Htn	Dm	Cad	Pe	Ane	Classification
67	45	No	No	No	No	No	Ckd

5.2. Random Forest

Forest is generally defined as a group of trees put together. Similarly here as the name suggests Random Forest is a connection of several tree diagrams connected randomly.

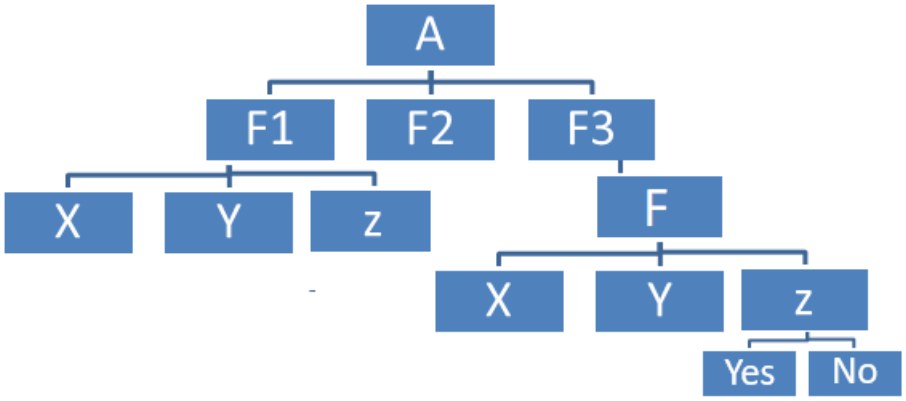


Fig 36: Sample diagram of Random Forest Classifier

Now, using WEKA we will apply this classifier to the kidney dataset.

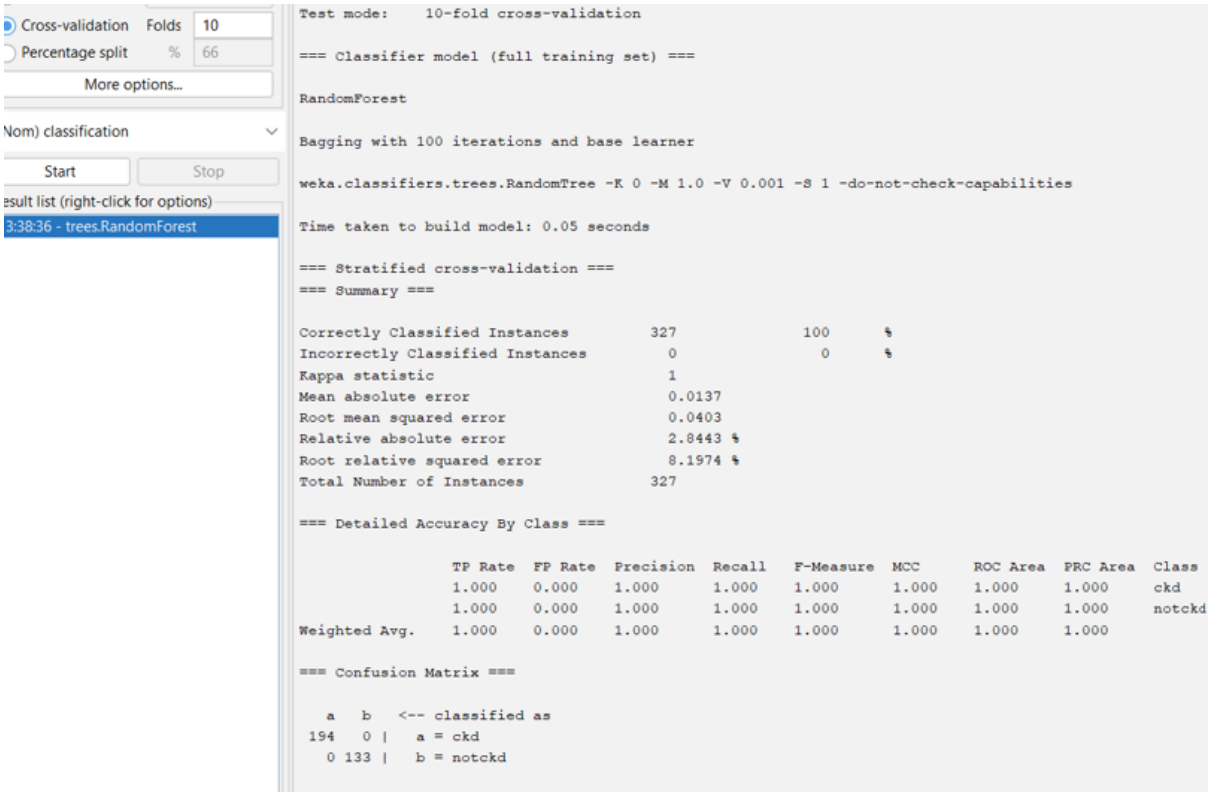


Fig 37: Random Forest Classifier

Great! In this classifier, all the instances are correctly classified which means Random Forest Classifier has shown 100% accuracy.

6. CONCLUSION

Kidney disease can be deadly if someone does not find it at its initial stage and can also lead to kidney transplant as on the severity of the disease. The main objective of this analysis is to make people aware of chronic kidney disease and its harmful effects and aware them so that they can take precautionary tests on time and reduce the risk of damaging their kidney functionality. It is advised that you should get your kidney-related tests done as you enter your thirties.

7. REFERENCES

[1] Chronic Kidney Disease Dataset, https://www.kaggle.com/mansoordaku/ckdisease

[2] Gupta, S.C., & Kapoor, V.K. (twelfth edition 2020). Fundamentals of Mathematical Statistics, Sultan Chand & Sons.

[3] Pandas documentation https://pandas.pydata.org/docs/user_guide/10min.html#viewing-data

[4] Centers for Disease Control and Prevention. *Chronic Kidney Disease in the United States, 2021*. Centers for Disease Control and Prevention, US Department of Health and Human Services; 2021.

[5] Google Data Analytics Professional Certificate https://www.coursera.org/account/accomplishments/specialization/certificate/TLBCU7URPLA3

[6] Python Code with Harry: Youtube https://youtu.be/gfDE2a7MKjA

[7] Jupyter Notebook: Youtube https://youtu.be/2WL-XTl2QYI

[8] Tazin, Nusrat & Sabab, Shahed & Chowdhury, Muhammed. (2016). Diagnosis of Chronic Kidney Disease using effective classification and feature selection technique. 1-6. 10.1109/MEDITEC.2016.7835365.

[9] Excel from Simplilearn: https://youtu.be/RkQl2wVpQAo

[10] Nptel Python for Data Science: https://youtu.be/N8RADjBmIws