PROJECT REPORT

EARLY DETECTION OF CHRONIC KIDNEY DISEASE USING ML

TEAM ID: PNT2022TMID05509

1.INTRODUCTION

1.1 PROJECT OVERVIEW

The objective of the present project is to employ machine learning algorithms in an attempt to develop a prediction model for progression to detect the Chronic Kidney disease in earlier stage.By using the wrapper method, a feature reduction analysis has been performed to find the attributes that detect this disease with high accuracy. By considering the parameters like albumin, specific gravity, diabetes mellitus, hemoglobin, and hypertension as features, we can predict the CKD at earlier stage As the result of our project, following program objectives can be outlined as the fundamentals for research and practical work in the field of Nephrology,

i. Deep research about the chronic kidney disease.

ii. Conduct an enquiry about the causes of chronic kidney disease.

iii. Study the research observations of reported ways of treating the CKD.

iv. Select the most accurate machine learning method and carry out new research to trace the disease earlier.

v. Create awareness among the ordinary people and to produce a user friendly model to assess their conditions at ease of their comfort.

vi. State the limitations of the current program of CKD and to produce new ideas to make it better.

vii. To implement solutions for the limitations stated.

viii. Introduce research findings to the nephrology and medical researchers to update their treatment techniques and the overall process of finding CKD. ix. Outline the directions for future enhancement.

1.2 PURPOSE

Two bean-shaped organs, named kidney, are two important parts in human body. Kidney removes waste from blood by filtering. If this filtering system is hampered, protein can seep to urine and waste elements can remain in blood. And gradually, kidney loses its ability to filter. This failure of kidney is called Chronic Kidney Disease (CKD), also known as Chronic Renal Disease.

Whole body is affected by kidney failure. Generally people suffer with this disease with their age, but recently from 5 years children and youth are also suffering from CKD disease. There are some symptoms which shows kidneys are beginning to fail like muscle cramps, nausea and vomiting, appetite losses, swelling in your feet and ankles, too much urine or not enough urine, trouble catching your breath, trouble sleeping, fever and vomiting. Risk factors of CKD are diabetes, smoking, lack of sleeping, hypertension, improper diet, etc. Among them diabetes is the more dangerous factor. At the last stage, the patient must take dialysis or do kidney transplantation. One of the best ways to reduce this death rate is early treatment. Therefore, early prediction and proper treatments can possibly stop, or slow the progression of this chronic disease.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEMS

The COVID-19 pandemic has highlighted the need for healthcare system resilience. Dialysis, CKD, organ transplantation and diabetes are the four comorbidities associated with the highest mortality risk from COVID-19, while CKD is the most prevalent risk factor for severe COVID-19. This heightened risk is evident from CKD stage 1 and increases through the subsequent stages of this disease, with the highest risk in patients on KRT.33 In addition, preliminary data have shown that about 20–30% of patients hospitalised with COVID-19 develop kidney failure, which has led to a surge in the requirement for dialysis. At the same time, regular dialysis services have been interrupted as hospitals redirect resources to providing care for patients with COVID-19.34 Lack of access to dialysis has long been a reality in LMICs, but its impact has now been intensely highlighted even in high-income countries, owing to critical shortages of dialysis equipment and staff during the COVID-19 pandemic.21 Earlier diagnosis of CKD through early detection programmes, combined with timely management using evidence-based strategies, would reduce the number of patients at risk of developing a complication of COVID-19, as well as reducing the number of patients with CKD progressing to dialysis, decreasing demand on services in times of unprecedented pressure on healthcare systems.

2.2 REFERENCES

1. Kidney Disease: Improving Global Outcomes (KDIGO) CKD Work Group. KDIGO 2012 clinical practice guideline for the evaluation and management of chronic kidney disease. Kidney Int Suppl 2013;3(1):1-150

2. Szczech LA, Stewart RC, Su HL, et al. Primary care detection of chronic kidney disease in adults with type-2 diabetes: the ADD-CKD Study (awareness, detection and drug therapy in type 2 diabetes and chronic kidney disease). PLoS One 2014;9(11):e110535

3. Elshahat S, Cockwell P, Maxwell AP, Griffin M, O'Brien T, O'Neill C. The impact of chronic kidney disease on developed countries from a health economics perspective: A systematic scoping review. PLoS One 2020;15(3):e0230512

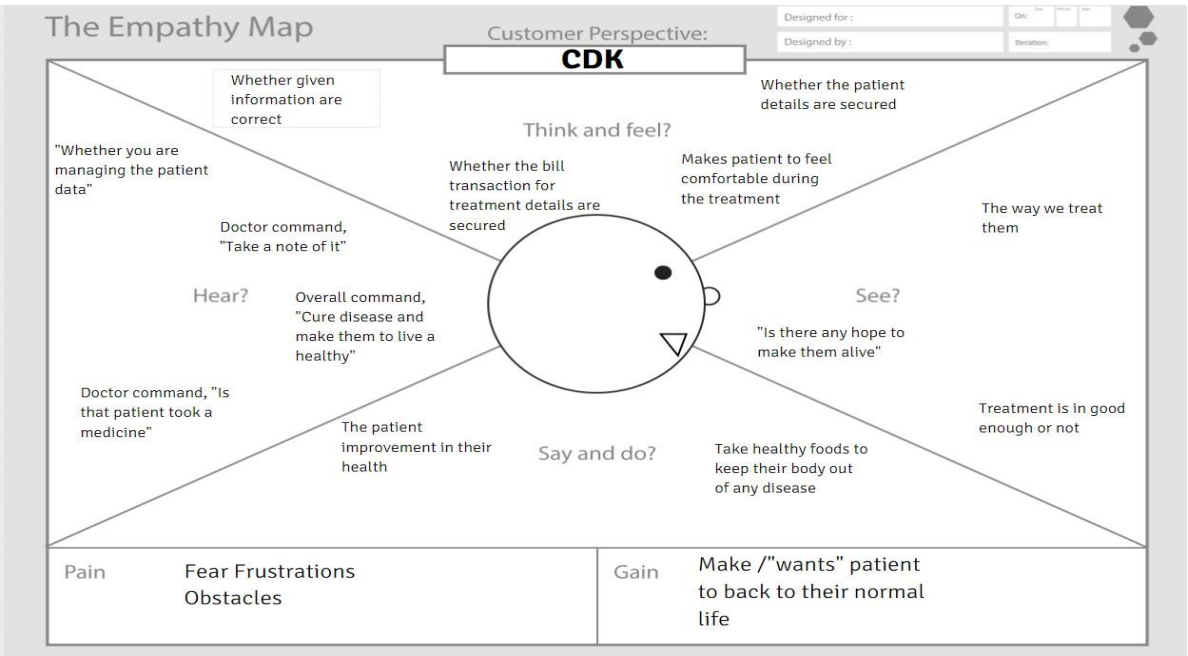
4. Jager KJ, Kovesdy C, Langham R, Rosenberg M, Jha V, Zoccali C. A single number for advocacy and communication-worldwide more than 850 million individuals have kidney diseases. Nephrol Dial Transplant 2019;34(11):1803-1805.

2.3 PROBLEM STATEMENT DEFINITION

To detect the chronic kidney disease early by given the dataset of affected patients to train the model and predict for the real time data using python machine learning technology.

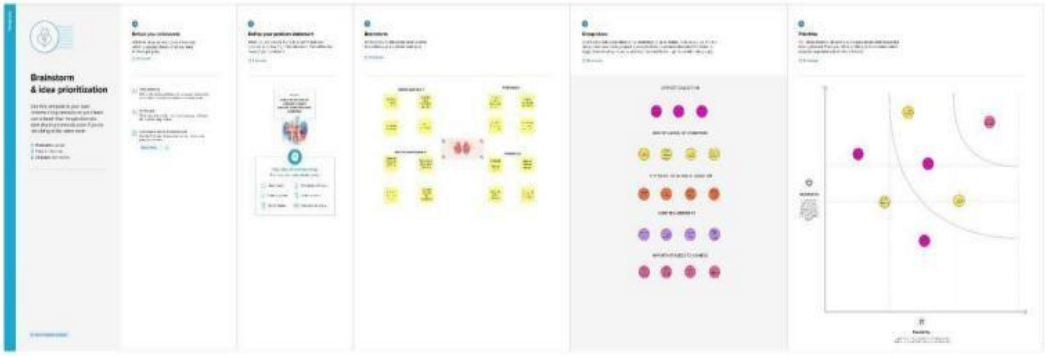
3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTROMING

Brainstorm&Idea Prioritization :



3.3 PROPOSED SOLUTION

1.	Problem Statement (Problem to be solved)	Patients who suffer from chronic kidney diseases need a way to control its progression to an advanced state with early detection and appropriate treatment. Machine learning has advanced to the point that it is now possible to look through patient medical information and identify chronic kidney disease in its early stages.
2.	Idea / Solution description	Since certain data are missing, the initial step is to perform pre-processing by cleaning the dataset, along with scaling and normalisation of values. The next step is to use dimensionality reduction to identify the key features in the dataset and to remove any irrelevant ones. To accomplish early detection of chronic kidney disease utilising the indicated key traits, a decision tree model must be fitted.
3.	Novelty / Uniqueness	• An indicator of how well the kidneys is working is the amount of a waste product called creatinine in the blood. By examining this data, early kidney disease can be identified by detecting deviations from the norm. • In the case of healthcare management products, it is especially important to have a UI that is very user-friendly and open to everyone.
4.	Social Impact / Customer Satisfaction	The primary goal of this application is early prediction, and appropriate treatments may be able to prevent or delay the disease's progression to an advanced state.
5.	Business Model (Revenue Model)	• The suggested strategy has the potential to generate income from

		direct patients as payment for the development of immediate outcomes. • It can also collaborate with the healthcare sector to generate revenue from patients who come in for kidney disease diagnosis.
6.	Scalability of the Solution	• The dimensionality reduction process can be adjusted to produce precise predictions with an increase in the features taken into account. • The accuracy of many models can be compared in order to determine which is best. • It can be used for a variety of illnesses in addition to chronic disorders.

3.4 PROBLEM SOLUTION FIT

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

1. All the data must be in the same format as a structured data. 2. The data collected will be vectorized and sent across to the classifier.

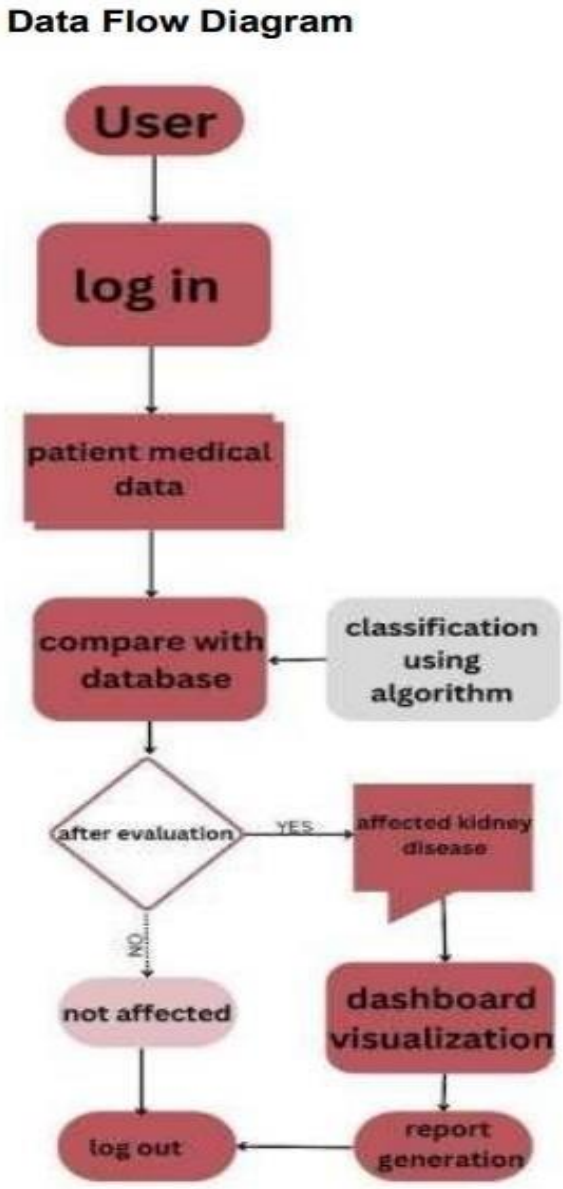
4.2 NON- FUNCTIONAL REQUIREMENTS

Non functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

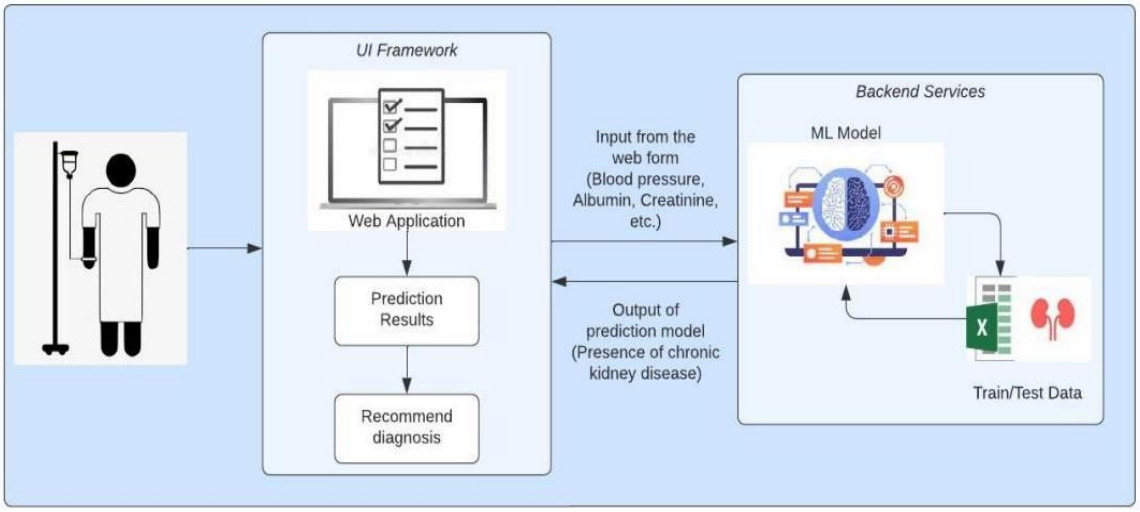
- Product Requirements
- Organizational Requirements
- User Requirements
- Basic Operational Requirements

5. PROJECT DESIGN

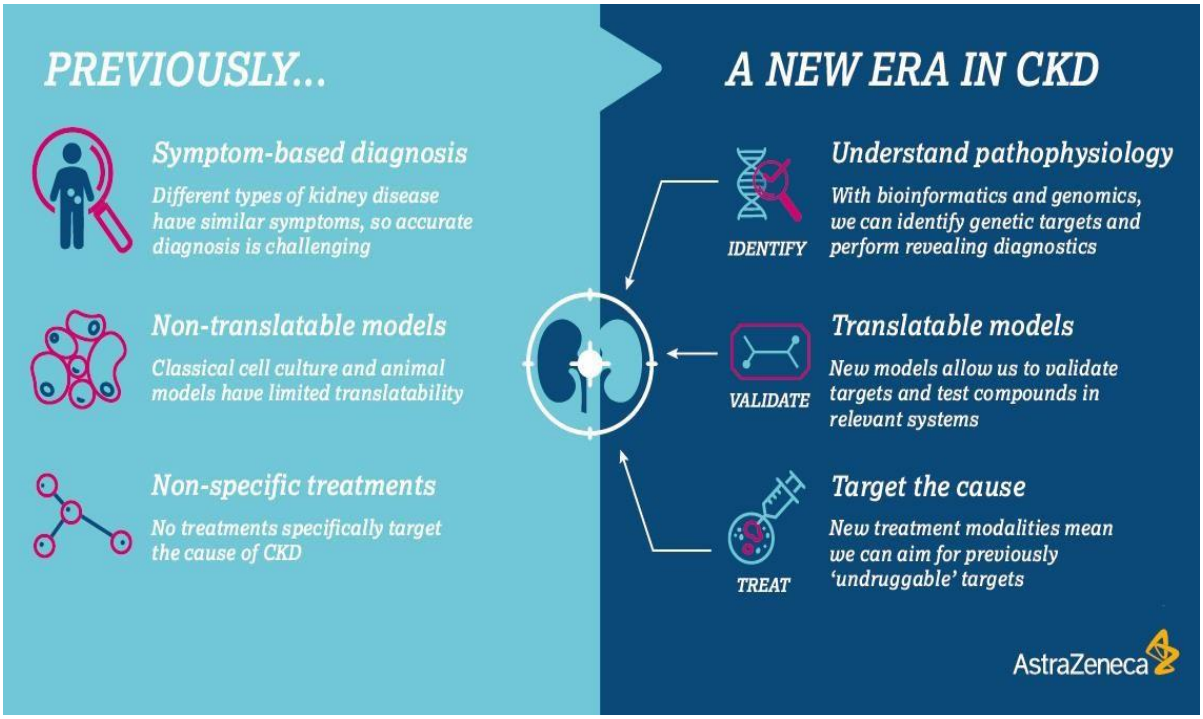
5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES



6. PROJECT PLANNING & SCHEDULING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Homepage	USN-1	As a user, I can see the homepage of the application	4	Medium	JEYA SRINIVASAN, JAYA SURYA
Sprint-2	Dashboard	USN-2	As a user, I must enter the required parameters required to make the prediction	7	High	JEYA SRINIVASAN, JAYA SURYA
Sprint-3	Result	USN-3	As a user, I can view the report generated by the tool (Prediction result - Positive/Negative)	8	High	JEYA SRINIVASAN, JAYA SURYA
Sprint-2		USN-4	As an administrator, I should identify the most significant factors that lead to CKD based on the present trend and come up with the input parameter that should be given by the user for CKD prediction	5	High	JAGAN, HARIHARAN
Sprint-3	Prediction	USN-5	As an administrator, I must use the most suitable ML model for detection of CKD	4	High	JAGAN, HARIHARAN
Sprint-4		USN-6	As an administrator, I must ensure that the web application is live and is accessible on any device with internet connectivity	7	High	JAGAN, HARIHARAN

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

7. CODING & SOLUTIONING

IMPORT THE LIBRARIES

In [7]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

READ THE DATASET

In [8]:
```
df = pd.read_csv("chronic_kidney_disease.csv")
```

In [9]:
```
print("The dataset shape is {}".format(df.shape))
```

```
The dataset shape is (400, 26)
```

In [10]:
```
df.head()
```

Out[10]:

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

5 rows × 26 columns

In [11]:
```
df.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',
```

CLEANING THE DATASET

In [16]:
```
# cleaning 'PCV'
df['pcv']=df['pcv'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\t43','43').replace('\t?','Nan'))

# cleaning "WC"
df['wc']=df['wc'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\t?','Nan').replace('\t6200','6200').replace('\t8400','8400'))

# cleaning "RC"
df['rc']=df['rc'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\t?','Nan'))

# cleaning "dm"
df['dm']=df['dm'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\tno','no').replace('\tyes','yes').replace(' yes','yes'))

# cleaning "CAD"
df['cad']=df['cad'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\tno','no'))

# cleaning "Classification"
df['classification']=df['classification'].apply(lambda x:x if type(x)==type(3.5) else x.replace('ckd\t','ckd'))
```

In [17]:
```
mistyped=[['pcv','rc','wc']]
for i in mistyped:
    df[i]=df[i].astype('float')
```

In [18]:
```
cat_cols=list(df.select_dtypes('object'))
cat_cols
```

Out[18]:
```
['rbc',
 'pc',
 'pcc',
 'ba',
 'htn',
 'dm',
 'cad',
 'appet',
 'pe',
 'ane',
 'classification']
```

HANDLING THE MISSING VALUES

In [20]:
```
df.isnull().sum().sort_values(ascending=False)
```

Out[20]:
```
rbc               152
rc                131
wc                106
pot                88
sod                87
pcv                71
pc                 65
hemo               52
su                 49
sg                 47
al                 46
bgr                44
bu                 19
sc                 17
bp                 12
age                 9
ba                  4
pcc                 4
htn                 2
dm                  2
cad                 2
ane                 1
appet               1
pe                  1
id                  0
classification      0
dtype: int64
```

In [21]:
```
for col in num_cols:
    df[col]=df[col].fillna(df[col].median())
```

In [22]:
```
df['rbc'].fillna('normal',inplace=True)
df['pc'].fillna('normal',inplace=True)
df['pcc'].fillna('notpresent',inplace=True)
df['ba'].fillna('notpresent',inplace=True)
df['htn'].fillna('no',inplace=True)
df['dm'].fillna('no',inplace=True)
df['cad'].fillna('no',inplace=True)
df['appet'].fillna('good',inplace=True)
df['pe'].fillna('no',inplace=True)
df['ane'].fillna('no',inplace=True)
```

In [23]:
```
df.isna().sum().sort_values(ascending=False)
```

Out[23]:
```
id                0
age               0
ane               0
pe                0
appet             0
cad               0
dm                0
htn               0
rc                0
wc                0
pcv               0
hemo              0
pot               0
sod               0
sc                0
bu                0
bgr               0
ba                0
pcc               0
pc                0
rbc               0
su                0
al                0
sg                0
bp                0
```

8. TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	3	1	1	18
Fixed	14	4	2	2	12
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	23	7	3	3	30

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	13	0	0	13
Client Application	21	0	0	21
Security	2	0	0	2
Exception Reporting	1	0	0	1
Final Report Output	8	0	0	8
Version Control	1	0	0	1

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve:-

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide operational reliability of the system.

9. RESULTS

For the purposes of this project we have used five popular algorithms: Logistic regression and Neural network, Decision Tree, Gaussian NB, KNN. All the algorithms are based on supervised learning. We are determining the best method considering 4 factors namely Specificity, Sensitivity, Log Loss, Accuracy. When plotted on a graph for all the algorithms it was found that Logistic Regression was the best method to use to find Chronic Kidney Disease.

Accuracy

Accuracy is the number of correctly predicted data points out of all the data points. More formally, it is defined as the number of true positives and true negatives divided by the number of true positives, true negatives, false positives, and false negatives.

Accuracy = $(TP + TN)/(TP + TN + FP + FN)$

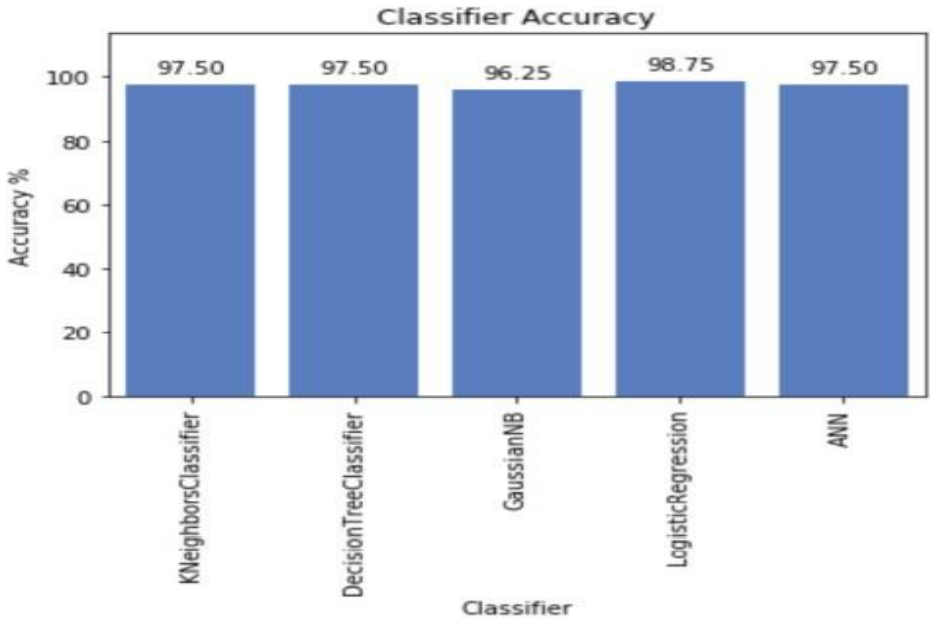


Figure 9.1: Accuracy comparison

9.1 PERFORMANCE METRICES

PREFORMANCE METRICES

Here we will be evaluating the model built. We will be using the test set for evaluation. The test set is given to the model for prediction and prediction values are stored in another variable called y_pred. The score of the model is calculated and its performance is
estimated.

Learning algorithm	Test Data	Training Data	Training Time
Linear Regression	0.84	0.89	6 minutes
Random Forest Algorithm	0.85	0.93	12 minutes

10. ADVANTAGES & DISADVANTAGES

techniques

Technique	Advantages	Disadvantages
Dual-energy X-ray Absorptiometry (DEXA/DXA)	• High accuracy and sensitivity • Gold standard for bone mineral density • Quick and painless for the patient	• Exposure to radiation • Expensive to use • Limited access/availability • Very obese may exceed the weight limit of the compartment
Computed (axial) tomography (CT or CAT scan)	• High accuracy and sensitivity • Gold standard for brain scans, organ, and bone assessment • Patient time: 5-10 minutes	• Exposure to radiation • Expensive to use • Limited access/availability
Magnetic Resonance Imaging (MRI)	• High accuracy and sensitivity • Gold standard for imaging any part of the body	• Expensive to use • Limited access/availability • Patients discomfort (claustrophobia) • Patient time: 15-90 minutes
Muscle biopsy	• High sensitivity for assessing small tissue samples • Gold standard for diagnosis of muscle tissue disease • Can assess muscle quality	• Patient discomfort and apprehension • Safety issues/ infection • Requires local anaesthetic • Patient time: 30 minutes plus recovery (10 days before stitches removed)
Body mass index (BMI)= Weight(kg)/ height(m)^2	• Easy to perform/calculate in large population groups • Simple, and standardized • No skill required • Patient time: 1-5 minutes	• Cannot differentiate between muscle and fat • Does not consider fat repositioning with age (peripheral to central)

11. CONCLUSION

This system presented the best prediction algorithm to predict CKD at an early stage. The dataset shows input parameters collected from the CKD patients and the models are trained and validated for the given input parameters. K-Nearest-Neighbors Classifier, Decision Tree Classifier, GaussianNB, Logical Regression and Artificial Neural Network learning models are constructed to carry out the diagnosis of CKD. The performance of the models is evaluated based on a variety of comparison metrics are being used, namely Accuracy, Specificity, Sensitivity and Log Loss. The results of the research showed that Logical Regression model better predicts CKD in comparison to the other models taking all the metrics under consideration. This system would help detect the chances of a person having CKD further on in his life which would be really helpful and cost-effective people. This model could be integrated with normal blood report generation, which could automatically flag out if there is a person at risk. Patients would not have to go to a doctor unless they are flagged by the algorithms. This would make it cheaper and easier for the modern busy person.

12. FUTURE SCOPE

- This would help detect the chances of a person having CKD further on in his life which would be really helpful and cost-effective people.

- This model could be integrated with normal blood report generation, which could automatically flag out if there is a person at risk.

- Patients would not have to go to a doctor unless they are flagged by the algorithms. This would make it cheaper and easier for the modern busy person.

13. APPENDIX

SOURCE CODE

IMPORT THE LIBRARIES

In [7]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

READ THE DATASET

In [8]:
```
df = pd.read_csv("chronic_kidney_disease.csv")
```

In [9]:
```
print("The dataset shape is {}".format(df.shape))
```

```
The dataset shape is (400, 26)
```

In [10]:
```
df.head()
```

Out[10]:

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

5 rows × 26 columns

In [11]:
```
df.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',
```

CLEANING THE DATASET

In [16]:
```
# cleaning 'PCV'
df['pcv']=df['pcv'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\t43','43').replace('\t?','Nan'))

# cleaning "WC"
df['wc']=df['wc'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\t?','Nan').replace('\t6200','6200').replace('\t8400','8400'))

# cleaning "RC"
df['rc']=df['rc'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\t?','Nan'))

# cleaning "dm"
df['dm']=df['dm'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\tno','no').replace('\tyes','yes').replace(' yes','yes'))

# cleaning "CAD"
df['cad']=df['cad'].apply(lambda x:x if type(x)==type(3.5) else x.replace('\tno','no'))

# cleaning "Classification"
df['classification']=df['classification'].apply(lambda x:x if type(x)==type(3.5) else x.replace('ckd\t','ckd'))
```

In [17]:
```
mistyped=[['pcv','rc','wc']]
for i in mistyped:
    df[i]=df[i].astype('float')
```

In [18]:
```
cat_cols=list(df.select_dtypes('object'))
cat_cols
```

Out[18]:
```
['rbc',
 'pc',
 'pcc',
 'ba',
 'htn',
 'dm',
 'cad',
 'appet',
 'pe',
 'ane',
 'classification']
```

HANDLING THE MISSING VALUES

In [20]:
```
df.isnull().sum().sort_values(ascending=False)
```

Out[20]:
```
rbc               152
rc                131
wc                106
pot                88
sod                87
pcv                71
pc                 65
hemo               52
su                 49
sg                 47
al                 46
bgr                44
bu                 19
sc                 17
bp                 12
age                 9
ba                  4
pcc                 4
htn                 2
dm                  2
cad                 2
ane                 1
appet               1
pe                  1
id                  0
classification      0
dtype: int64
```

In [21]:
```
for col in num_cols:
    df[col]=df[col].fillna(df[col].median())
```

In [22]:
```
df['rbc'].fillna('normal',inplace=True)
df['pc'].fillna('normal',inplace=True)
df['pcc'].fillna('notpresent',inplace=True)
df['ba'].fillna('notpresent',inplace=True)
df['htn'].fillna('no',inplace=True)
df['dm'].fillna('no',inplace=True)
df['cad'].fillna('no',inplace=True)
df['appet'].fillna('good',inplace=True)
df['pe'].fillna('no',inplace=True)
df['ane'].fillna('no',inplace=True)
```

In [23]:
```
df.isna().sum().sort_values(ascending=False)
```

Out[23]:
```
id                0
age               0
ane               0
pe                0
appet             0
cad               0
dm                0
htn               0
rc                0
wc                0
pcv               0
hemo              0
pot               0
sod               0
sc                0
bu                0
bgr               0
ba                0
pcc               0
pc                0
rbc               0
su                0
al                0
sg                0
bp                0
```

FINAL OUTCOME:

PREDICTION-1

In [59]:
```
prediction = predict(67.4,7.2,0.99,4,17.0,1,160.6,87,22089,36)[0]
if prediction:
  print('Oops! You have Chronic Kidney Disease.')
else:
  print("Great! You don't have Chronic Kidney Disease.")
```

```
Great! You don't have Chronic Kidney Disease.
```

PREDICTION-2

In [60]:
```
prediction = predict(27.4,4.2,0.19,1,7.0,0,90.6,37,30949,26)[0]
if prediction:
  print('Oops! You have Chronic Kidney Disease.')
else:
  print("Great! You don't have Chronic Kidney Disease.")
```

```
Great! You don't have Chronic Kidney Disease.
```

PREDICTION-3

In [61]:
```
prediction = predict(17.4,2.2,0.89,0,12.0,0,50.6,87,949,19)[0]
if prediction:
  print('Oops! You have Chronic Kidney Disease.')
else:
  print("Great! You don't have Chronic Kidney Disease.")
```

```
Great! You don't have Chronic Kidney Disease.
```

GitHub & Project Demo Link:-

Our team Github link is as follows:

https://github.com/IBM-EPBL/IBM-Project-9160-1658984945