Project Report

1. INTRODUCTION

Chronic Kidney disease means your kidneys are damaged and can't filter blood the way they should.The disease is called "CHRONIC" because they damage to your kidneys happens slowly over a long period of time.This damage can cause wastes to build up in your body.CKD can also cause other health problems.

The kidneys main job is to filter extra water and wastes out of your blood to make urine.To keep your body working properly the kidneys balance the salts and minerals such as calcium,phosphorous,sodium and potassium that circulate in the blood.Your kidneys also make homones that help control blood pressure,make red blood cells,and keep your bones strong.

Kidney disease often can get worse over time and may lead to kidney failure.If your kidneys fail,you will need dialysis or a kidney transplant to protect your kidneys.The sooner you know you have kidney disease,the sooner you can make changes to protect your kidneys.

PROJECT OVERVIEW:

The main goal of treatment is to prevent progression CKD to complete kidney failure. The best way to do this is to diagnose CKD early and control the underlying cause. The symptoms, evaluation, and management of CKD will be reviewed here. To detect the detection of disease at the earliest stage of spread of disease. To provide correct accuracy of spread of disease. To prevent spread of disease at the early stage.

PURPOSE:

- To detect the disease spread in the early stage.
- To prevent the loss of life and to prevent the kidney failure.
- To estimate the accuracy of chronic kidney disease.
- To save the time and to detect in an easier way.
- Your test results can be used to determine how damaged your kidneys are known as the stage of ckd.
- To prevent side effects of ckd such as breathing.

LITERATURE SURVEY OF CHRONIC KIDNEY DISEASE ANALYSIS:

STATISTICAL AND DATA MINING ASPECTSON KIDNEY STONES: A SYSTEMATIC REVIEW AND META ANALYSIS:

This project is about a systematic review and meta-analysis using classification algorithms studies detected good accuracy with C4.5, classification tree and Random forest(93%) followed by Support Vector Machines(SVM)(91.98%).Logistic and NNge has also shown good accuracy results also shown good accuracy Results with zero relative absolute error and 100% correctly classified results. Machine Learning approaches may provide better results in the treatment of kidney stones. Data mining offers a more quantative approach to quality control with ,user friendly for clinicians in reading the reports and reduce the errors. A meta-analysis combines results of a number of studies that deal with a set of related research hypotheses. A meta-analysis may be conducted on a several clinical trials of a medical treatment which refer tostatistical methods combining evidence. In the present experimentation, we had analyzed a setoff parameters related to kidney stone formation collected from patients in kaviti, and Andhra Pradesh, India.

DETECTION OF CHRONIC KIDNEY DISEASE USING RANDOM FOREST MACHINE LEARNING ALGORITHM:

In this paper they have used random forest machine learning algorithm to detect the chronic kidney disease they have compared the performance of six classifiers in the detection of chronic kidney disease analysis. The experimental results of the proposed method have demonstrated the RF has produced superior detection performance in terms of classification accuracy. AUC and MCC respectively for our considered dataset. It was also observed that few classifiers have yielded poor classification accuracy as compared to RF like SMO and RBF.

A NOVEL DETECTION FOR KIDNEY DISEASE USING IMPROVED SUPPORT VECTOR MACHINE:

This paper is about a novel detection for kidney disease using improved support vector machine. In this work, kidney disease detection system was developed using classification algorithms(KNN, Naive Bayes, SVM, ISVM) through MATLAB data mining tool to detect effective and better accurate results regarding whether the patient is suffering from kidney disease or not. As the kidney disease patients are increasing world-wide each year and huge amounts of data is available for research, where different data mining techniques are used in the diagnosis of kidney disease. Different attributes are used for detection of kidney disease.

DATA MINING CLASSIFICATION ALGORITHMS FOR KIDNEY DISEASE DETECTION:

In this paper data mining classification algorithm for kidney disease detection naive Bayes, svm, Ann, anfis they have used kidney function test (KFT) dataset. The algorithm which has the higher accuracy with the minimum execution time has chosen as the best algorithm machine learning tool is resulting in high Classification accuracy rate. The gap identified in the classifiers show different accuracy rate. Data mining is an approach which dispense an intermixture of technique to identify a block of data or decision making knowledge in the database and eradicating these data in such a way that they can be put to use in decision support, forecasting and estimation.

REFERENCES:

- Amirgaliyevi Y, Shamiluulu S,Serek A(2018) Analysis of Chronic Kidney Disease Dataset by Applying Machine Learning Methods. 2018 IEEE 12th International Conference on Application of Information and Communication Technologies(AICT).

● Devika R, Sai Vaishnavi A, Subramaniya Swamy V(2019) Comparative Study of Classifier for Chronic Kidney Disease Detection using Naive Bayes, KNN and Random Forest,2019 3rd International Conference on Computing Methodologies and Communication(ICCMC).

Problem statement Definition:

1.Kidney disease also increases your risk of having heart and blood Vessel disease.These problems may happen slowly over a long time.

2.Adults have CKD,and millions of Others are at increased risk so,early detection can help preventthe progression of kidney disease to Kidney failure.

3.Chronic Kidney disease includes conditions that damage your kidneys decrease their ability to keep you healthy by filtering wastes from your blood.

4.To effectively predict a disease,information such as narration about symptoms felt by the patients. Details of consultation with medical practitioners.

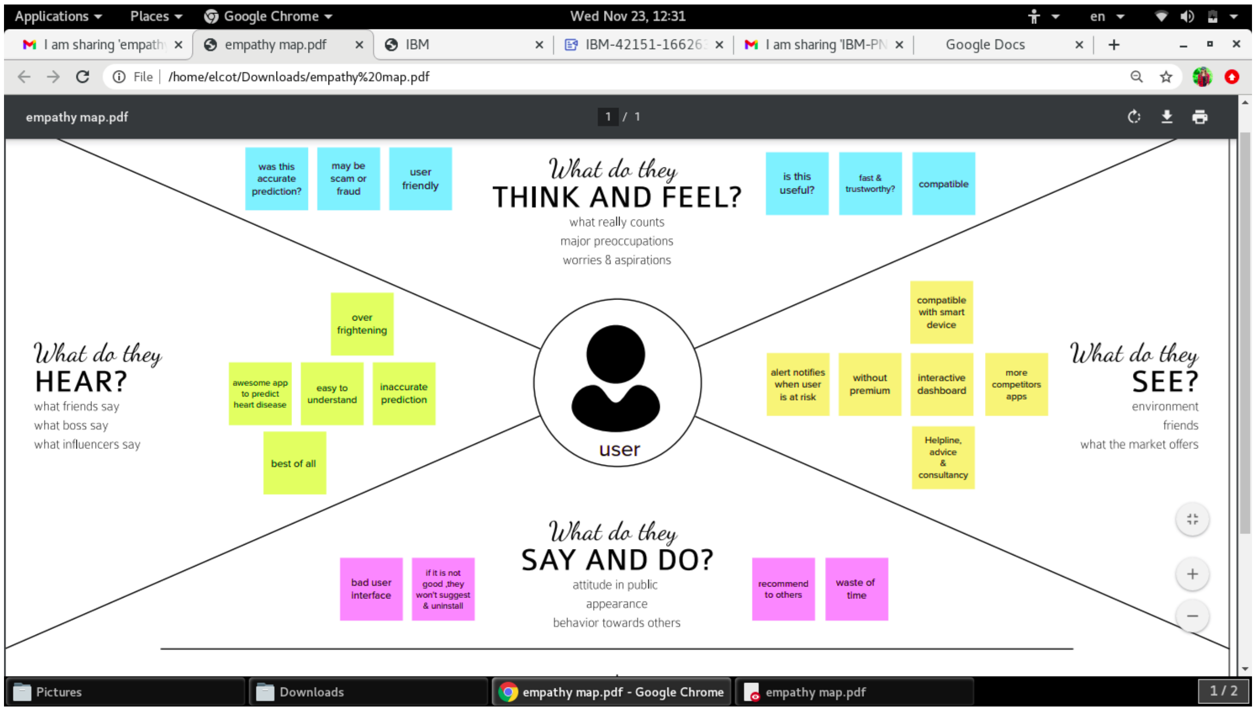
The USER needs away to identify:-

-whether he/she is affected by Chronic disease by analysing symptoms
-whether he/she need to or need not to consult a doctor
-notify when he/she is in risk
-improve diagnosis & quality of care
-keep uptodate medical records by analysing for predicting diseases
-advice for Chronic disease prevention.

IDEATION AND PROPOSED SOLUTION:

Empathy Map:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



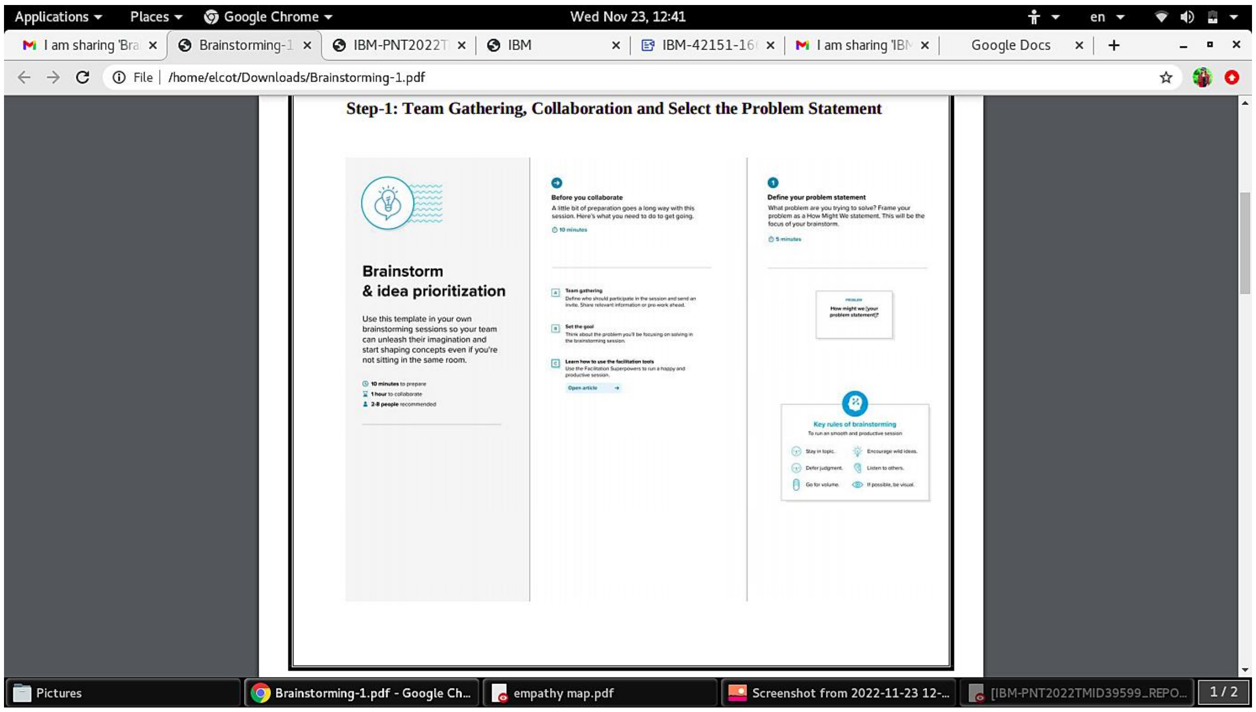
Ideation and Brainstorming:

Ideation:

Ideation is the process of forming ideas from conception to implementation, most often in a business setting. Ideation is expressed via graphical, written, or verbal methods, and arises from past or present knowledge, influences, opinions, experiences, and personal convictions.

Brainstorming:

Brainstorming is a group activity where everyone comes together to discuss strategies for growth and improvement. You can exchange ideas, share important information and use these meetings as informal catch-up sessions with your co-workers.


Step-1: Team Gathering, Collaboration and Select the Problem Statement

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A. **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B. **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C. **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

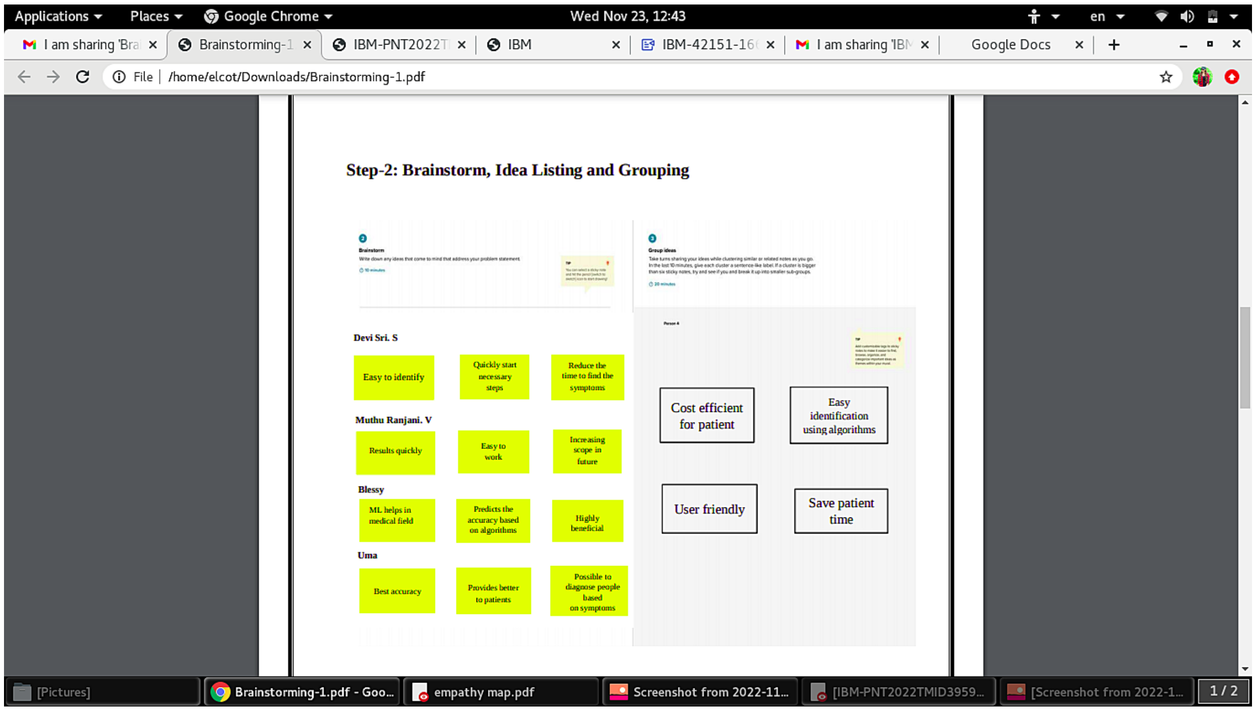
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.


Step-2: Brainstorm, Idea Listing and Grouping

2 **Brainstorm**
Write down any ideas that come to mind that address your problem statement.
10 minutes

3 **Group ideas**
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes

Devi Sri. S
- Easy to identify
- Quickly start necessary steps
- Reduce the time to find the symptoms

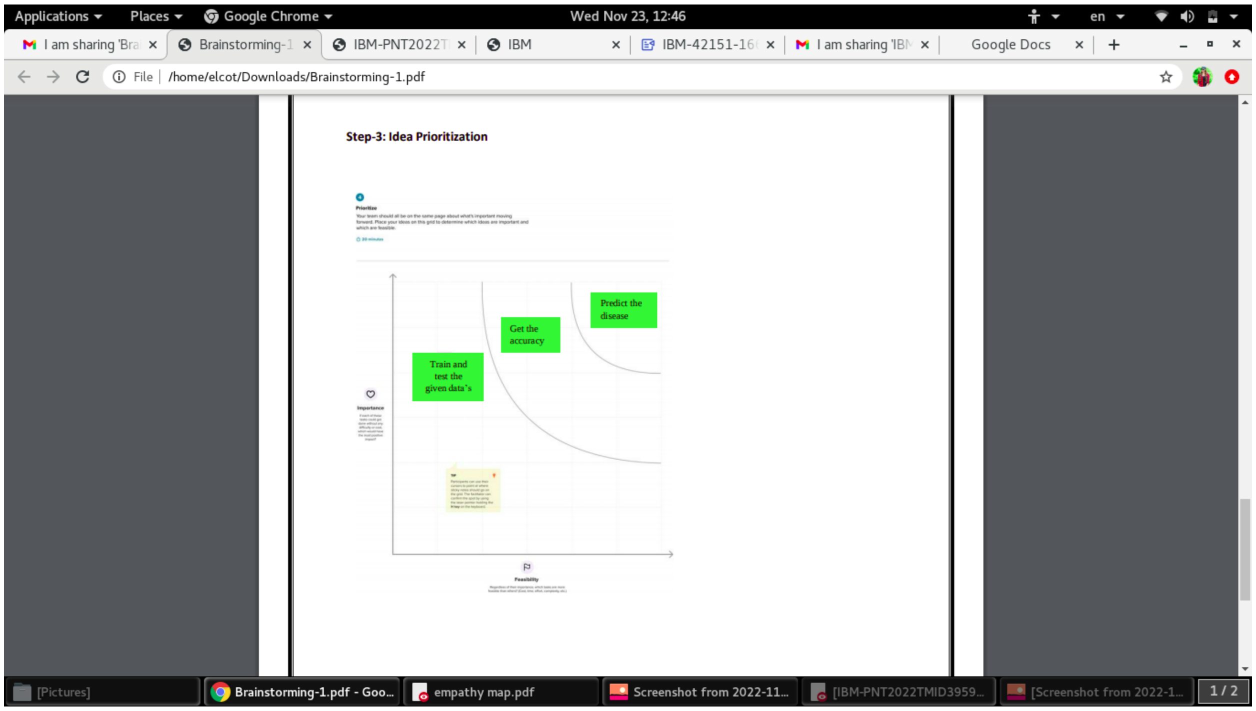
Muthu Ranjani. V
- Results quickly
- Easy to work
- Increasing scope in future

Blessy
- ML helps in medical field
- Predicts the accuracy based on algorithms
- Highly beneficial

Uma
- Best accuracy
- Provides better to patients
- Possible to diagnose people based on symptoms

Person 4

- Cost efficient for patient
- Easy identification using algorithms
- User friendly
- Save patient time



Proposed Solution:

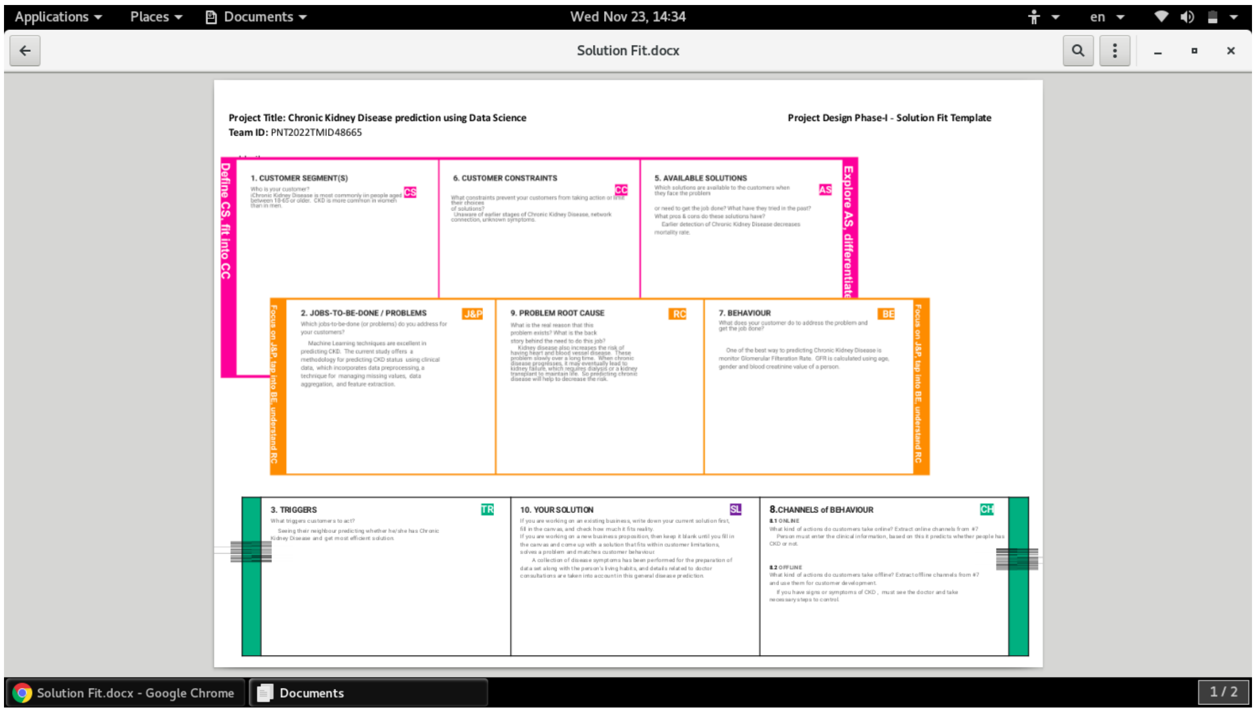
Proposed Solution means the combination of software, hardware, other products or equipment, and any and all services (including any installation, implementation, training, maintenance and support services) necessary to implement the solution described by Vendor in its Proposal.

S.no	Parameter	Description

1	Problem Statement (Problem to be solved)	CKD is a progressive disease often resulting in irreversible kidney failure, known as end-stage renal disease (ESRD), at which point dialysis or a kidney transplant is required to survive. So, early detection of Chronic Disease will help to reduce mortality rate.
2	Idea / Solution description	Machine learning (ML) techniques are excellent in predicting CKD. The current study offers a methodology for predicting CKD status using clinical data, which incorporates data preprocessing, a technique for managing missing values, data aggregation, and feature extraction.
3	Novelty / Uniqueness	In the proposed solution, we use some specified algorithms such as Decision tree,Knearest neighbour is suitable for accurate prediction. we can predict the chronic Kidney Disease with more than 97% accuracy.
4	Social Impact / Customer Satisfaction	This will aid in the achievement of improved outcomes as well as the accuracy and efficiency with which healthcare practitioners can anticipate kidney issues. This will enhance the dependability of the framework as well as the framework’s presentation. The hope is that it would encourage people to seek early treatment for chronic renal disease and to make improvements in their lives.
5	Business Model (Revenue Model)	When the patients use this model, they don’t have to spend more amount of money for initial stages of diagnosis. This model will identify and predict chronic disease earlier so more number of clients will approach us and it makes more profit in both sides.
6	Scalability of the Solution	Training the model with more number of attributes will increase the efficiency.

Problem Solution Fit:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.



Project Title: Chronic Kidney Disease prediction using Data Science
Team ID: PNT2022TMID48665

Project Design Phase-I - Solution Fit Template

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
Who is your customer?
iChronic Kidney Disease is most commonly iin people aged between 18-65 or older. CKD is more common in women than in men.

6. CUSTOMER CONSTRAINTS CC
What constraints prevent your customers from taking action or limit their choices
of solutions?
Unaware of earlier stages of Chronic Kidney Disease, network connection, unknown symptoms.

5. AVAILABLE SOLUTIONS AS
Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past?
What pros & cons do these solutions have?
Earlier detection of Chronic Kidney Disease decreases mortality rate.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P
Which jobs-to-be-done (or problems) do you address for your customers?
Machine Learning techniques are excellent in predicting CKD. The current study offers a methodology for predicting CKD status using clinical data, which incorporates data preprocessing, a technique for managing missing values, data aggregation, and feature extraction.

9. PROBLEM ROOT CAUSE RC
What is the real reason that this problem exists? What is the back story behind the need to do this job?
Kidney disease also increases the risk of having heart and blood vessel disease. These problem slowly over a long time. When chronic disease progresses, it may eventually lead to kidney failure, which requires dialysis or a kidney transplant to maintain life. So predicting chronic disease will help to decrease the risk.

7. BEHAVIOUR BE
What does your customer do to address the problem and get the job done?
One of the best way to predicting Chronic Kidney Disease is monitor Glomerular Filteration Rate. GFR is calculated using age, gender and blood creatinine value of a person.

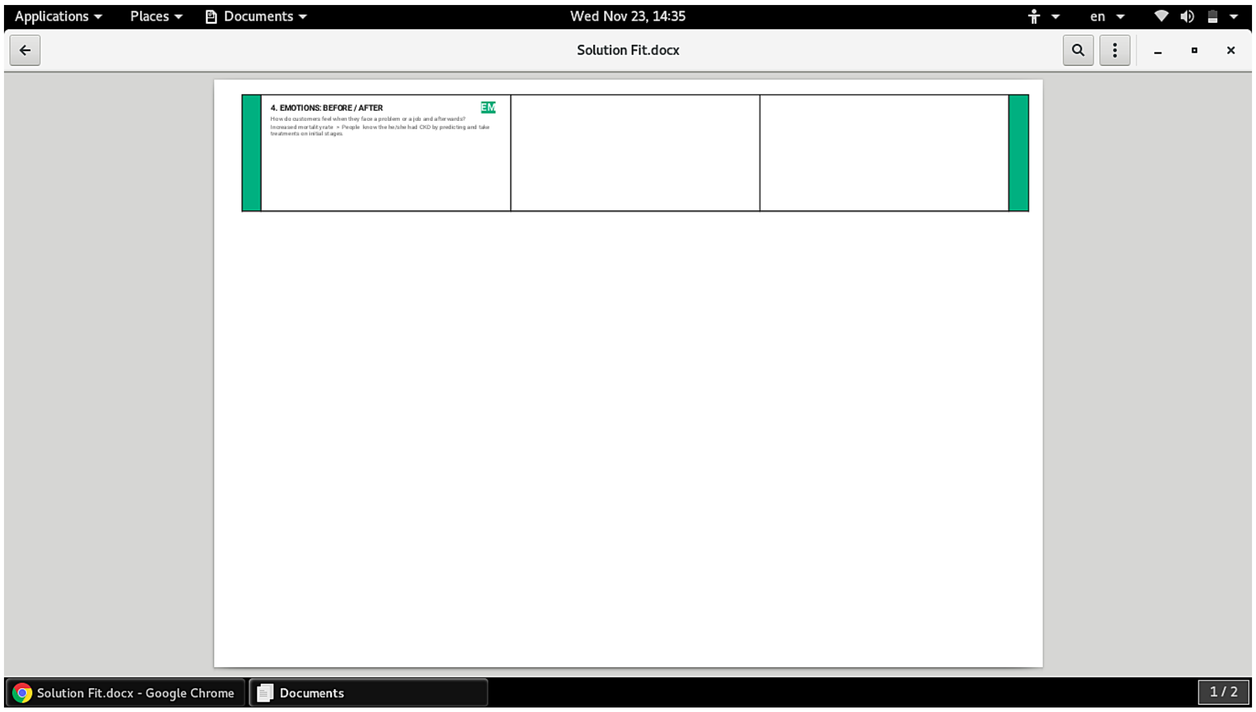
Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR
What triggers customers to act?
Seeing their neighbour predicting whether he/she has Chronic Kidney Disease and get most efficient solution.

10. YOUR SOLUTION SL
If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.
A collection of disease symptoms has been performed for the preparation of data set along with the person's living habits, and details related to doctor consultations are taken into account in this general disease prediction.

8.CHANNELS of BEHAVIOUR CH
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7
Person must enter the clinical information, based on this it predicts whether people has CKD or not.

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.
If you have signs or symptoms of CKD , must see the doctor and take necessary steps to control.



4. EMOTIONS: BEFORE / AFTER EM
How do customers feel when they face a problem or a job and afterwards?
Increased mortality rate > People know the he/she had CKD by predicting and take treatments on initial stages.

REQUIREMENT ANALYSIS:

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. In software engineering, it is sometimes referred to loosely by names such as requirements gathering or requirements capturing.

TYPES OF REQUIREMENTS:

1. Functional Requirements.
2. Non-functional Requirements.

Functional Requirements:

S.no	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	*website opens *Explains Chronic Kidney Disease, causes and treatments. *new user asks to sign up, or already if the user exists asks to login
FR-2	User Confirmation and verificaton	The user had to enter the email id or mobile number,password, and her/his name. The OTP will be sent to registered email id or phone number.
FR-3	Collection of dataset	Collect the dataset based on CKD and process the data.
FR-4	Training the model	By using the processed data, the data will be trained again again to get greater accuracy.
FR-5	Testing the model	Model will be tested using 20% of data.
FR-6	Prediction	After completing the above process, the result is predicted.(whether he/she has CKD or not).

S.no	Non-Functional Requirement	Description
NFR-1	Usability	By Predicting CKD in earlier stages helps to decrease mortality rate. So, detecting CKD in earlier stages by using machine learning model using the attributes of clinical tests is going to very useful for the users.
NFR-2	Security	The reports are maintained confidentially to the customer.
NFR-3	Reliability	The model will predict the CKD more accurate and keep the customer data more secure. As a result more number of customers will approaches us.
NFR-4	Performance	By training the model again and again to get 95% of accuracy. So the performance of model is good.
NFR-5	Availability	Website and trained will be available to predict at any time.
NFR-6	Scalability	This model can be explained to include more attributes for more accurate detection. Training the model with even more attributes will increase the efficiency.

PROJECT DESIGN:

Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information.

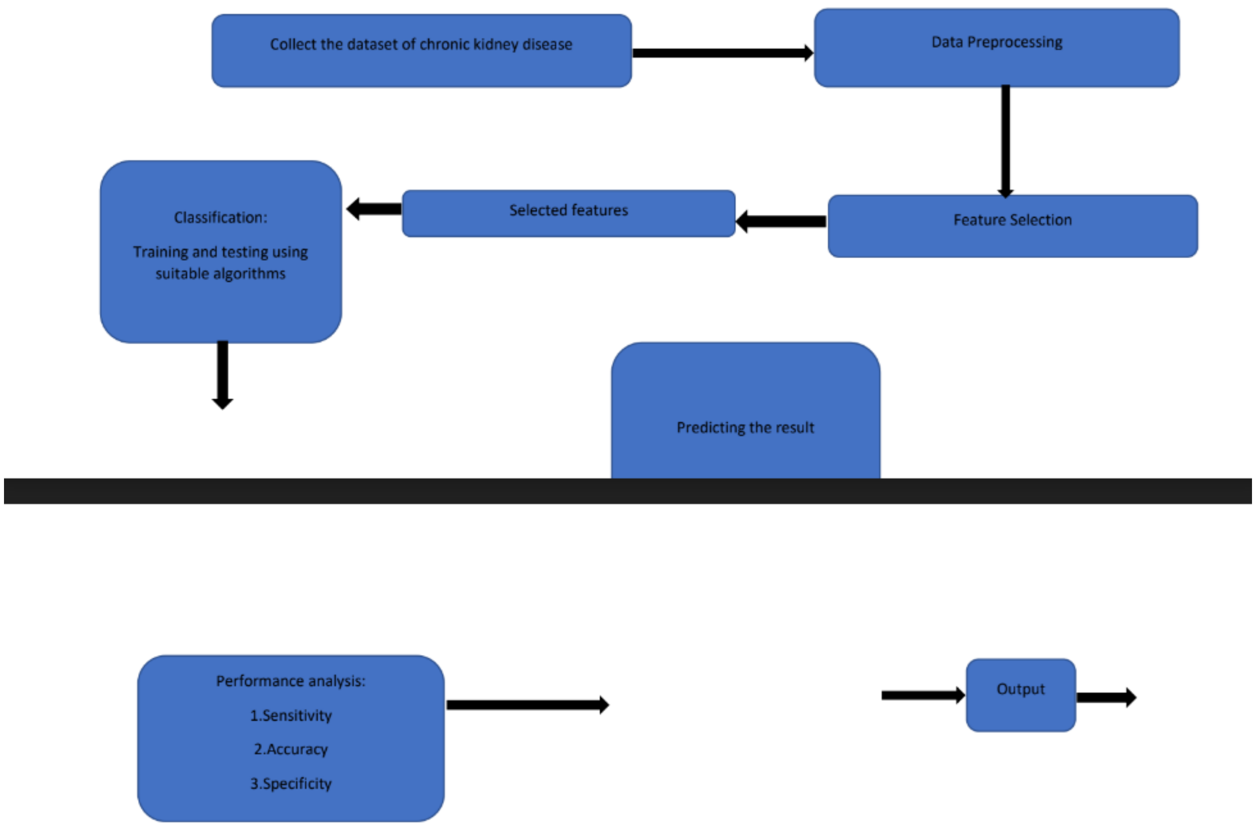
TOPICS IN PROJECT DESIGN:

- DATA FLOW DIAGRAMS.
- SOLUTION AND TECHNICAL ARCHITECTURE.
- USER STORIES.

DATA FLOW DIAGRAMS:

A data flow diagram (DFD) maps out the flow of information for any process or system.It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

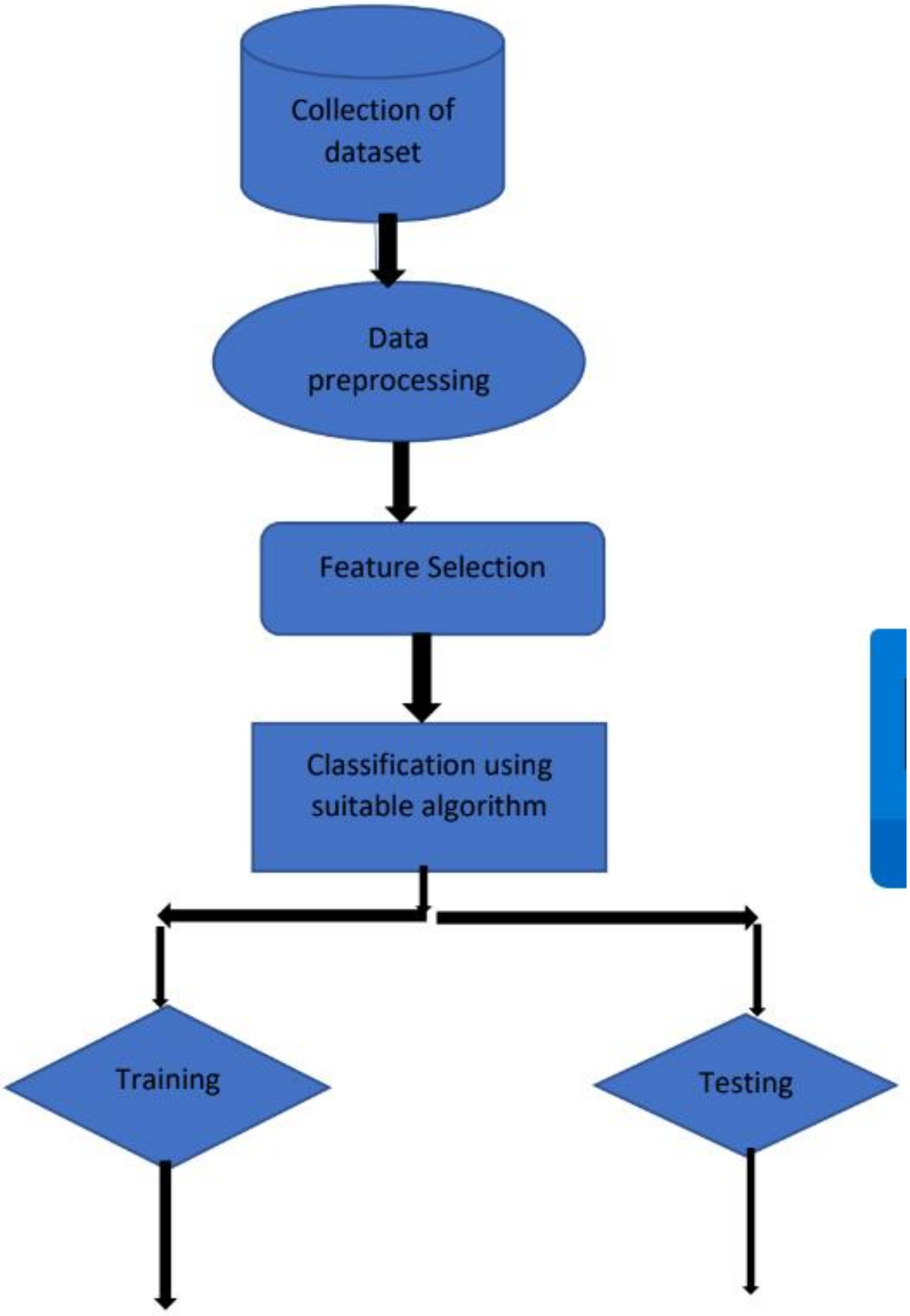
Data Flow Diagrams:

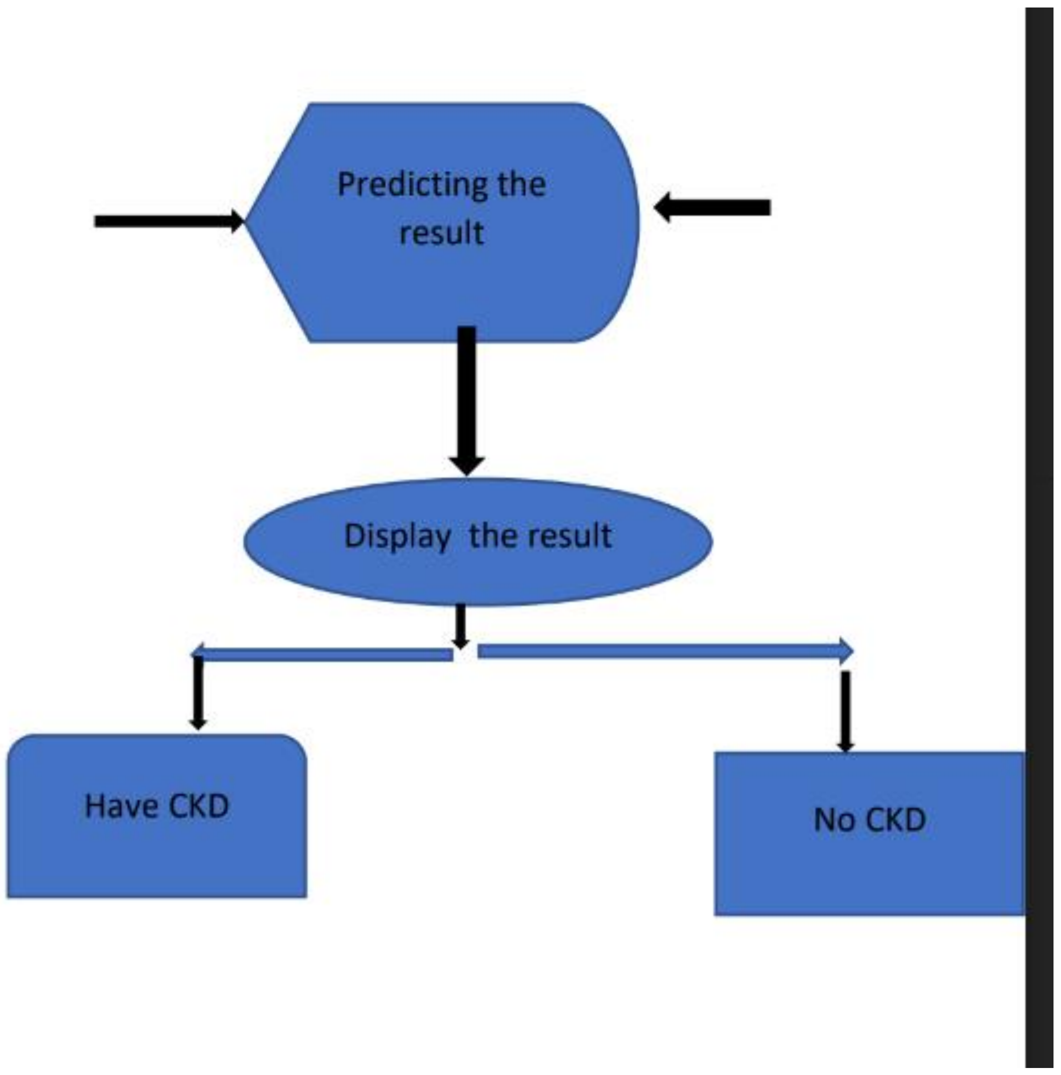


SOLUTION AND TECHNICAL ARCHITECTURE:

SOLUTION ARCHITECTURE:

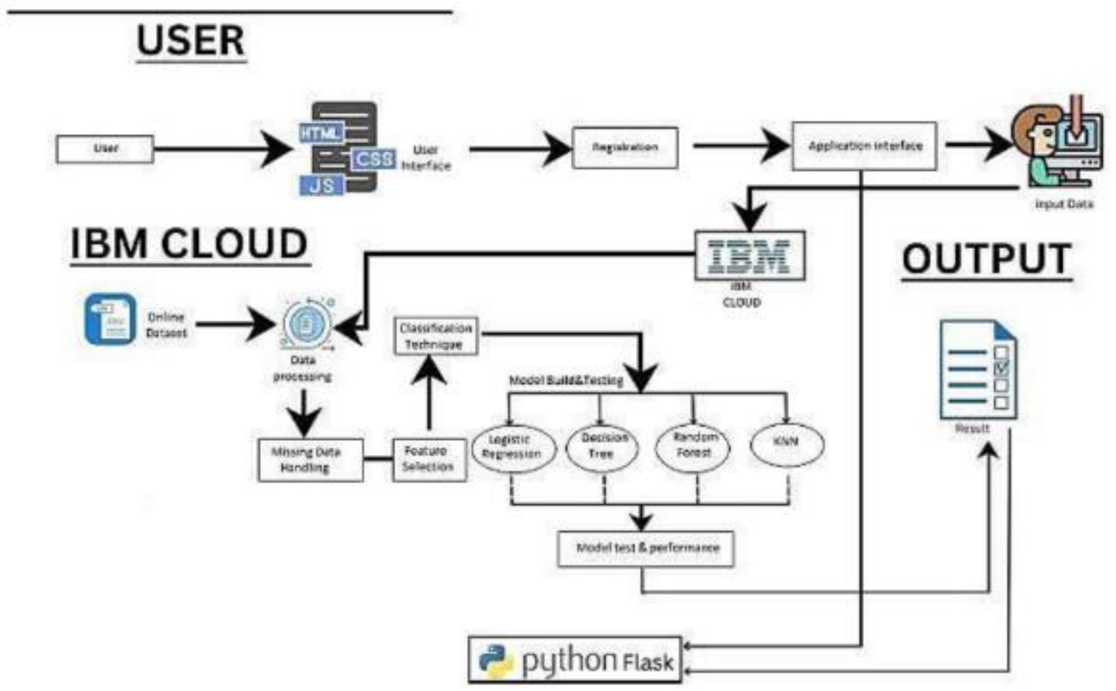
CKD is a condition in which the kidneys are damaged and cannot filter blood as well as they should. Because of this, excess fluid and waste from blood remain in the body and may cause other health problems, such as heart disease and stroke. So early detection of chronic Kidney disease is helpful to decrease mortality rate.


Collection of dataset
Data preprocessing
Feature Selection
Classification using suitable algorithm
Training
Testing



TECHNICAL ARCHITECTURE:

Technical architecture—which is also often referred to as application architecture, IT architecture, business architecture, etc., refers to creating a structured software solution that will meet the business needs and expectations while providing a strong technical plan for the growth of the software application through its lifetime.



USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my mail id.	I can access my account and view the initial page of the web.	High	Sprint-1
	Confirmation	USN-2	As a user, I will receive confirmation email once,I have registered for the website.	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user, I can login into the site by clicking onto the login link.	I can successfully login to the page and my details will be shown.	Low	Sprint-2
	Dashboard	USN-4	As a user, I can access my dashboard.	I can modify the details in the dashboard.	Medium	Sprint-1
	Homepage	USN-5	As a user, I can view the entered details and required information.	Based on used requirements, contents are categorized.	High	Sprint-3
Customer Care Executive	Help	USN-6	As a user, I can contact the site owner if the user facing any problems.	Reporting issues option will be provided.	High	Sprint-3
Administrator	Verification	USN-7	Administrator has a unique email id and password in order to access and organize the registered users.	Admin sign in option will be provided.	High	Sprint-2

PROJECT PLANNING AND SCHEDULING:
PROJECT PLANNING:
Project planning is a discipline addressing how to complete a project in a certain timeframe, usually with defined stages and designated resources. One view of

project planning divides the activity into these steps: setting measurable objectives. identifying deliverables.scheduling.
SCHEDULING:

Scheduling is the process of arranging, controlling and optimizing work and workloads in a production process or manufacturing process. Scheduling is used to allocate plant and machinery resources, plan human resources, plan production processes and purchase materials.

SPRINT PLANNING:

Sprint Planning is an event that defines what can be delivered in the upcoming Sprint and how their work can be achieved. It kicks off the Sprint.

Each Sprint has a specific duration.

- Sprint 1-Data Collection.
- Sprint 2- Model Building.
- Sprint 3- Training and Testing.
- Sprint 4- Implementation of the Application.

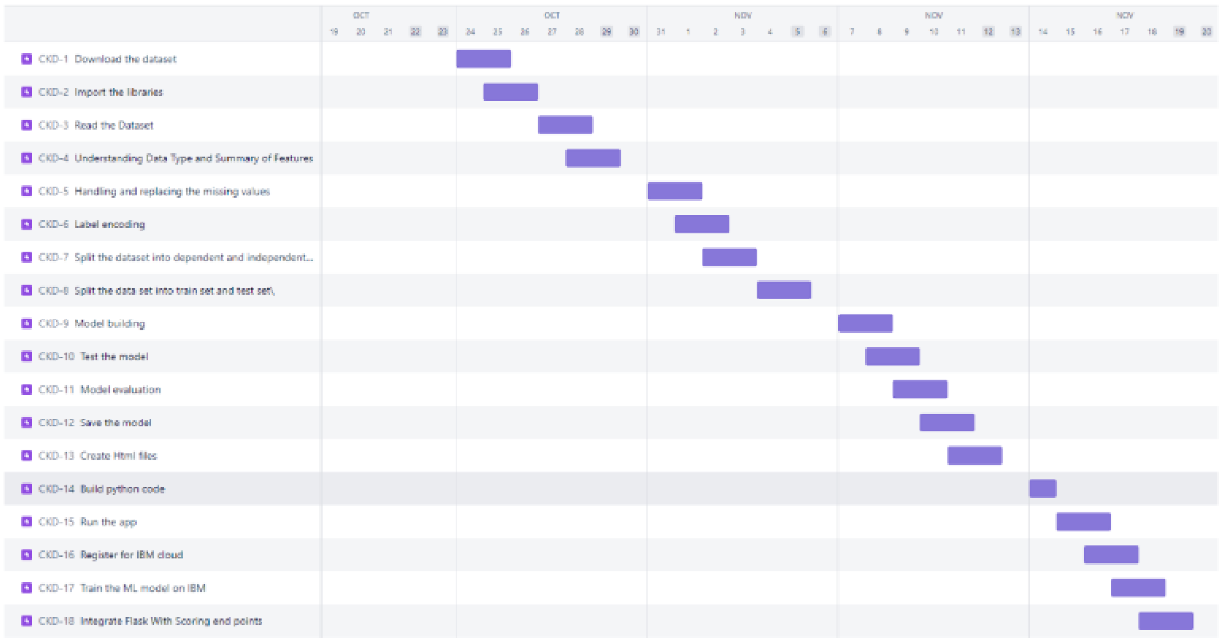
ESTIMATION:

Estimation is a process to detect the time and the cost that a project requires to be finished appropriately.

SPRINT DELIVERY SCHEDULE:

Sprint	Sprint start date	Sprint end date	Sprint release date
Sprint-1	24 Oct 2022	29 Oct 2022	29 Oct 2022
Sprint-2	31 Oct 2022	05 Nov 2022	05 Nov 2022
Sprint-3	07 Nov 2022	12 Nov 2022	12 Nov 2022
Sprint-4	14 Nov 2022	19 Nov 2022	19 Nov 2022

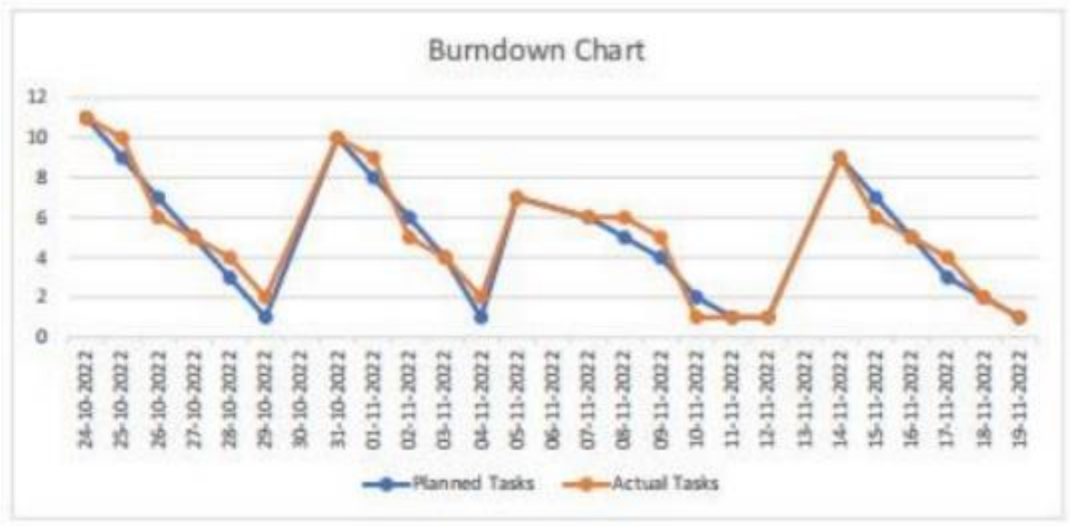
REPORTS FROM JIRA:



CKD-1 Download the dataset
CKD-2 Import the libraries
CKD-3 Read the Dataset
CKD-4 Understanding Data Type and Summary of Features
CKD-5 Handling and replacing the missing values
CKD-6 Label encoding
CKD-7 Split the dataset into dependent and independent...
CKD-8 Split the data set into train set and test set\
CKD-9 Model building
CKD-10 Test the model
CKD-11 Model evaluation
CKD-12 Save the model
CKD-13 Create Html files
CKD-14 Build python code
CKD-15 Run the app
CKD-16 Register for IBM cloud
CKD-17 Train the ML model on IBM
CKD-18 Integrate Flask With Scoring end points

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum.



CODING AND SOLUTIONING:

Coding or programming is the key activity and an engineering methodology through which the system visualized by the end user in terms of requirements is brought to life.

Solutioning is the process of acquiring the solution to the given problem is called as Solutioning.

FEATURE 1:

The main feature used in the coding is the machine learning concept. This machine learning is mainly used for detection of accuracy of disease.

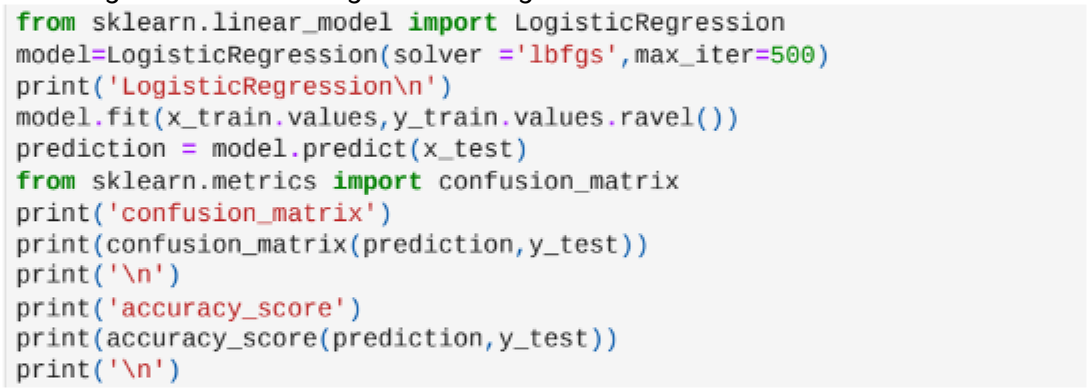
Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behavior.

FEATURE 2:

The second main feature used in coding is the Regression model.

Logistic Regression model is used for detection of the chronic kidney disease.

Logistic regression aims to solve classification problems. It does this by detecting categorical outcomes, unlike linear regression that detects a continuous outcome. In the simplest case there are two outcomes, which is called binomial, an example of which is detecting if a tumor is malignant or benign.



```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression(solver ='lbfgs',max_iter=500)
print('LogisticRegression\n')
model.fit(x_train.values,y_train.values.ravel())
prediction = model.predict(x_test)
from sklearn.metrics import confusion_matrix
print('confusion_matrix')
print(confusion_matrix(prediction,y_test))
print('\n')
print('accuracy_score')
print(accuracy_score(prediction,y_test))
print('\n')
```

TESTING:

Testing is the practice of making objective judgments regarding the extent to which the. system(device) meets, exceeds or fails to meet stated objectives.

TEST CASES:

A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly.

TEST SCENARIOS:

- Verify whether is able to enter data to detection of accuracy of disease.
- Verify whether the user is getting the correct accuracy of disease.
- Verify whether the environment is user friendly or not.

USER ACCEPTANCE TESTING:

User acceptance testing (UAT), also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by its intended audience.

It involves testing the entire software and detecting the errors, measuring the level of security, environment friendly software etc.

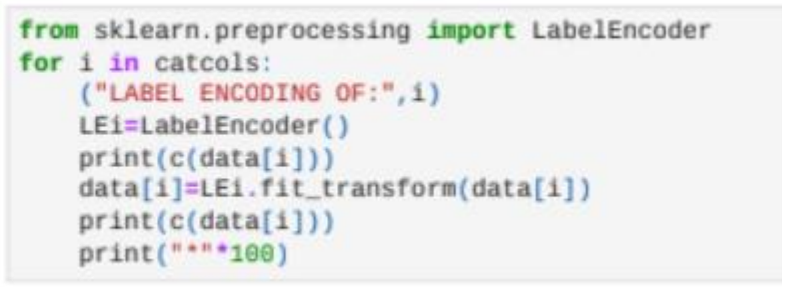
RESULTS:

 Performance Metrics:

Performance metrics are defined as figures and data representative of an organization's actions, abilities, and overall quality.

 Label Encoding:

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form.



```
from sklearn.preprocessing import LabelEncoder
for i in catcols:
    ("LABEL ENCODING OF:",i)
    LEi=LabelEncoder()
    print(c(data[i]))
    data[i]=LEi.fit_transform(data[i])
    print(c(data[i]))
    print("*"*100)
```

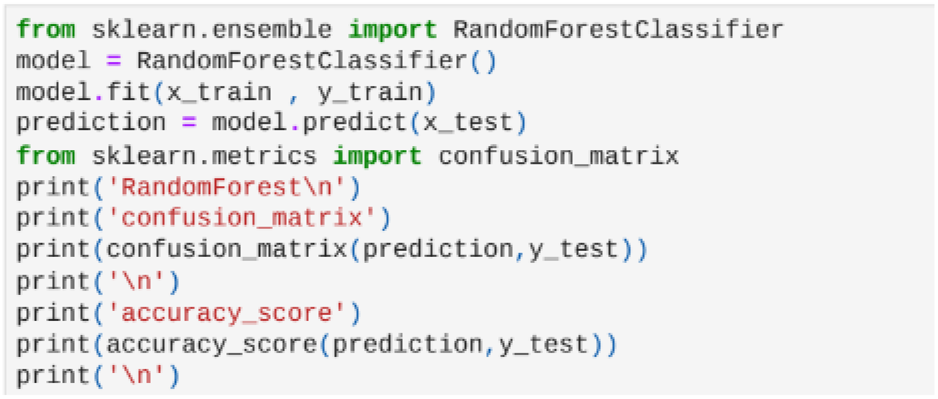
 Independent and Dependent Variables:

A dependent variable is a variable whose value depends on another variable, whereas An Independent variable is a variable whose value never depends on another variable.

 Build The Model:

Random Forest Classifier:

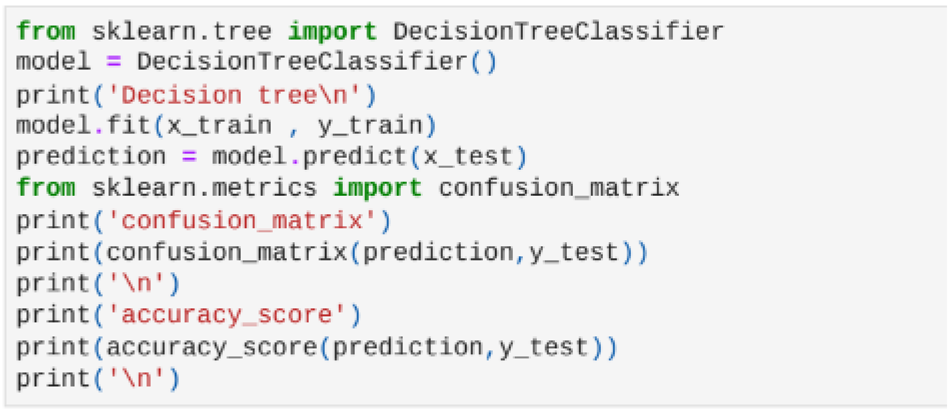
Random forest is a flexible,easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms,due to its simplicity and diversity (it can be used for both classification and regression tasks).



```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(x_train , y_train)
prediction = model.predict(x_test)
from sklearn.metrics import confusion_matrix
print('RandomForest\n')
print('confusion_matrix')
print(confusion_matrix(prediction,y_test))
print('\n')
print('accuracy_score')
print(accuracy_score(prediction,y_test))
print('\n')
```

Decision Tree Classifier:

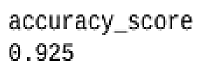
Decision trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter.The tree can be explained by two entities,namely decision nodes and leaves.



```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
print('Decision tree\n')
model.fit(x_train , y_train)
prediction = model.predict(x_test)
from sklearn.metrics import confusion_matrix
print('confusion_matrix')
print(confusion_matrix(prediction,y_test))
print('\n')
print('accuracy_score')
print(accuracy_score(prediction,y_test))
print('\n')
```

 Accuracy Score Of The Model:

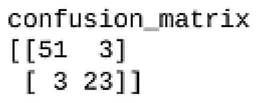
It is the process of detecting the accuracy score of the model.



```
accuracy_score
0.925
```

 Confusion Matrix of our Model:

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes.



```
confusion_matrix
[[51  3]
 [ 3 23]]
```

ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

- 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 effectivelymassist physicians in achieving this goal.
- To prevent the kidney failure.
- To prevent loss of life.
- Using this we an able to detect at the early stage and the patient can take efficient treatment according to the spread.

DISADVANTAGES:

- Sometimes the accuracy may vary.
- The result may vary if the accuracy value is wrong or vary.These are some of the advantages of disadvantages of chronic kidney disease analysis using machine learning.

CONCLUSION:

Chronic renal failure represents a critical period in the evolution of chronic renal disease and is associated with complications and comorbidities that begin early in the course of the disease. These conditions are initially subclinical but progress relentlessly and may eventually become symptomatic and irreversible. Early in the course of chronic renal failure, these conditions are amenable to interventions with relatively simple treatments that have the potential to prevent adverse outcomes.6 Globally, CKD is most commonly attributed to diabetes and/or hypertension, but other causes such as glomerulonephritis, infection, and environmental exposures (such as air pollution, herbal remedies, and pesticides) are common in Asia, sub-Saharan Africa, and many developing countries.4 Genetic risk factors may also contribute to CKD risk. For example, sickle cell trait and the presence of 2 APOL1 risk alleles, both common in people of African ancestry but not European ancestry, may double the risk of CKD.4,7–10.

FUTURE SCOPE:

- In future, it can be used in hospitals.
- It can be used at medical fields for easy detection of kidney disease at the early stage.
- It can be used as a web service for detection using online.
- It can be used to develop an application which may be helpful for detection of disease at the easiest way.
- For old age people it is very helpful to detect the disease.
- Easy way of detection of kidney disease.

APPENDIX:

A document that describes the design of a software component, product, or system.

SOURCE CODE:

```
In [1]: import pandas as pd
        import numpy as np
        from collections import Counter as c
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.metrics import accuracy_score, confusion_matrix
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import LabelEncoder
        from sklearn.linear_model import LogisticRegression
        import pickle
```

```
In [2]: data = pd.read_csv(r"C:\Users\ELCOT\Desktop\chronickidneydisease.csv")
```

```
In [3]: data
```

```
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
...   ...   ...   ...    ...  ...  ...     ...       ...         ...         ...  ...  ...   ...  ...  ...  ...  ...   ...  ...  ...             ...
395   395  55.0  80.0  1.020  0.0  0.0  normal    normal  notpresent  notpresent  ...   47  6700  4.9   no   no  no   good   no   no          notckd
396   396  42.0  70.0  1.025  0.0  0.0  normal    normal  notpresent  notpresent  ...   54  7800  6.2   no   no  no   good   no   no          notckd
397   397  12.0  80.0  1.020  0.0  0.0  normal    normal  notpresent  notpresent  ...   49  6600  5.4   no   no  no   good   no   no          notckd
398   398  17.0  60.0  1.025  0.0  0.0  normal    normal  notpresent  notpresent  ...   51  7200  5.9   no   no  no   good   no   no          notckd
399   399  58.0  80.0  1.025  0.0  0.0  normal    normal  notpresent  notpresent  ...   53  6800  6.1   no   no  no   good   no   no          notckd

400 rows × 26 columns
```

```
In [4]: #Understanding data
```

```
In [5]: data.drop(["id"],axis=1,inplace=True)
        data.columns=['age','blood_pressure','specific_gravity','albumin','sugar','red_blood_cells','pus_cell','pus_cell_clu
                     'blood glucose random','blood_urea','serum_creatinine','sodium','potassium','hemoglobin','packed_cell_v
                      'white_blood_cell_count','red_blood_cell_count','hypertension','diabetesmellitus','coronary_artery_dis
                      'appetite','pedal_edema','anemia','class']
```

```
In [6]: data
```

```
Out[6]:
                                                                                                       blood
        age  blood_pressure  specific_gravity  albumin  sugar  red_blood_cells  pus_cell  pus_cell_clumps  bacteria  glucose  ...  packed_cell_volu
                                                                                                      random
```

Out[6]:

	age	blood_pressure	specific_gravity	albumin	sugar	red_blood_cells	pus_cell	pus_cell_clumps	bacteria	blood glucose random	...	packed_cell_volu
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	...	
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	...	
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	...	
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	...	
4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	140.0	...	
396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	75.0	...	
397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	100.0	...	
398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	114.0	...	
399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	131.0	...	

400 rows × 25 columns

In [7]:

```
catcols=set(data.dtypes[data.dtypes=='O'].index.values)
print(catcols)
```

```
{'hypertension', 'packed_cell_volume', 'diabetesmellitus', 'white_blood_cell_count', 'coronary_artery_disease', 'bac
teria', 'pus_cell', 'red_blood_cells', 'pus_cell_clumps', 'red_blood_cell_count', 'anemia', 'appetite', 'class', 'pe
dal_edema'}
```

In [8]:

```
contcols=set(data.dtypes[data.dtypes!='O'].index.values)
print(contcols)
```

```
{'hemoglobin', 'blood_pressure', 'sodium', 'specific_gravity', 'age', 'potassium', 'albumin', 'serum_creatinine', 's
ugar', 'blood glucose random', 'blood_urea'}
```

In [9]:

```
catcols.remove('red_blood_cell_count')
catcols.remove('packed_cell_volume')
catcols.remove('white_blood_cell_count')
catcols.add('specific_gravity')
catcols.add('albumin')
catcols.add('sugar')
print(catcols)
```

```
{'hypertension', 'diabetesmellitus', 'coronary_artery_disease', 'bacteria', 'pus_cell', 'sugar', 'specific_gravity',
'red_blood_cells', 'pus_cell_clumps', 'anemia', 'appetite', 'albumin', 'class', 'pedal_edema'}
```

In [10]:

```
contcols.remove('specific_gravity')
contcols.remove('albumin')
contcols.remove('sugar')
contcols.add('red_blood_cell_count')
contcols.add('packed_cell_volume')
contcols.add('white_blood_cell_count')
print(contcols)
```

```
{'hemoglobin', 'blood_pressure', 'packed_cell_volume', 'white_blood_cell_count', 'sodium', 'age', 'red_blood_cell_co
unt', 'potassium', 'serum_creatinine', 'blood glucose random', 'blood_urea'}
```

```
In [11]: #Get unique values from categorical columns
```

```
In [12]: for col in catcols:
             print(f"{col} has {data[col].unique()} values \n")
```

```
hypertension has ['yes' 'no' nan] values

diabetesmellitus has ['yes' 'no' ' yes' '\tno' '\tyes' nan] values

coronary_artery_disease has ['no' 'yes' '\tno' nan] values

bacteria has ['notpresent' 'present' nan] values

pus_cell has ['normal' 'abnormal' nan] values

sugar has [ 0.  3.  4.  1. nan  2.  5.] values

specific_gravity has [1.02  1.01  1.005 1.015   nan 1.025] values

red_blood_cells has [nan 'normal' 'abnormal'] values

pus_cell_clumps has ['notpresent' 'present' nan] values

anemia has ['no' 'yes' nan] values

appetite has ['good' 'poor' nan] values

albumin has [ 1.  4.  2.  3.  0. nan  5.] values

class has ['ckd' 'ckd\t' 'notckd'] values

pedal_edema has ['no' 'yes' nan] values
```

```
In [13]: data['class']=data['class'].replace("ckd\t","ckd")
         data['coronary_artery_disease']=data.coronary_artery_disease.replace('\tno','no')
         data['diabetesmellitus']=data.diabetesmellitus.replace(to_replace={'\tno':'no','\tyes':'yes',' yes':'yes'})
```

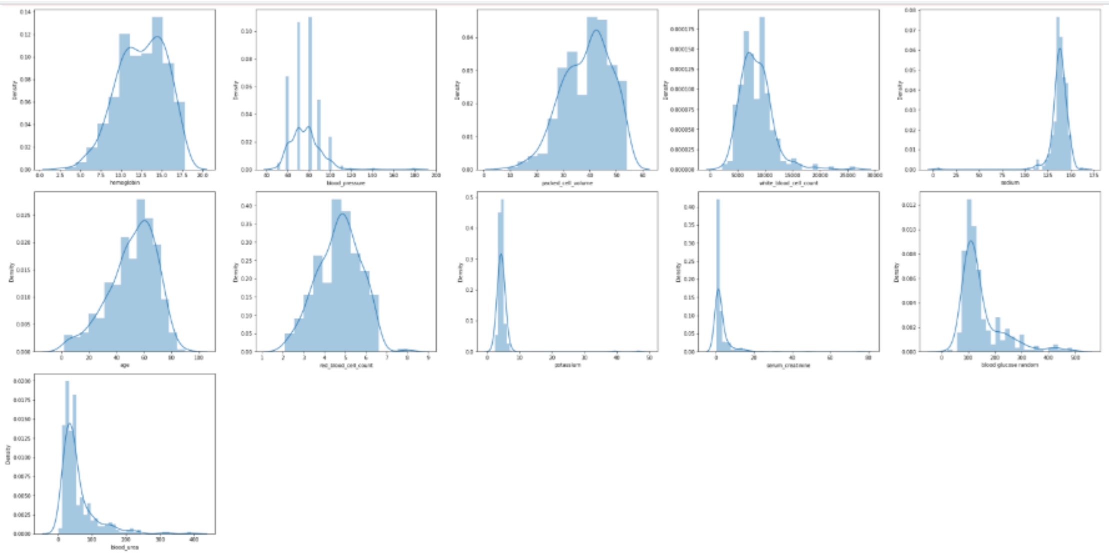
```
In [14]: data.packed_cell_volume = pd.to_numeric(data.packed_cell_volume, errors='coerce')
         data.white_blood_cell_count = pd.to_numeric(data.white_blood_cell_count, errors='coerce')
         data.red_blood_cell_count = pd.to_numeric(data.red_blood_cell_count, errors='coerce')
```

```
In [15]: # To handle skewness in the data
```

```
In [16]: plt.figure(figsize = (30,15))
         plotnumber =1
         for column in contcols:
             if plotnumber <= 14:
                 ax = plt.subplot(3, 5, plotnumber)
                 sns.distplot(data[column])
                 plt.xlabel(column)

             plotnumber += 1

         plt.tight_layout()
         plt.show()
```



In [17]:
```
#some data's still possess skewness because of the null values in it
```

In [18]:
```
data.isna().sum()
```

Out[18]:
```
age                          9
blood_pressure              12
specific_gravity            47
albumin                     46
sugar                       49
red_blood_cells            152
pus_cell                    65
pus_cell_clumps              4
bacteria                     4
blood glucose random        44
blood_urea                  19
serum_creatinine            17
sodium                      87
potassium                   88
hemoglobin                  52
packed_cell_volume          71
white_blood_cell_count     106
red_blood_cell_count       131
hypertension                 2
diabetesmellitus             2
coronary_artery_disease      2
appetite                     1
pedal_edema                  1
anemia                       1
class                        0
dtype: int64
```

In [19]:
```python
data['blood glucose random'].fillna(data['blood glucose random'].mean(),inplace=True)
data['blood_pressure'].fillna(data['blood_pressure'].mean(),inplace=True)
data['blood_urea'].fillna(data['blood_urea'].mean(),inplace=True)
data['hemoglobin'].fillna(data['hemoglobin'].mean(),inplace=True)
data['packed_cell_volume'].fillna(data['packed_cell_volume'].mean(),inplace=True)
data['potassium'].fillna(data['potassium'].mean(),inplace=True)
data['red_blood_cell_count'].fillna(data['red_blood_cell_count'].mean(),inplace=True)
data['serum_creatinine'].fillna(data['serum_creatinine'].mean(),inplace=True)
data['sodium'].fillna(data['sodium'].mean(),inplace=True)
data['white_blood_cell_count'].fillna(data['white_blood_cell_count'].mean(),inplace=True)
```

In [20]:
```python
data['age'].fillna(data['age'].mode()[0],inplace=True)
data['hypertension'].fillna(data['hypertension'].mode()[0],inplace=True)
data['pus_cell_clumps'].fillna(data['pus_cell_clumps'].mode()[0],inplace=True)
data['appetite'].fillna(data['appetite'].mode()[0],inplace=True)
data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)
data['pus_cell'].fillna(data['pus_cell'].mode()[0],inplace=True)
data['red_blood_cells'].fillna(data['red_blood_cells'].mode()[0],inplace=True)
data['coronary_artery_disease'].fillna(data['coronary_artery_disease'].mode()[0],inplace=True)
data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)
data['anemia'].fillna(data['anemia'].mode()[0],inplace=True)
data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)
data['diabetesmellitus'].fillna(data['diabetesmellitus'].mode()[0],inplace=True)
data['pedal_edema'].fillna(data['pedal_edema'].mode()[0],inplace=True)
data['specific_gravity'].fillna(data['specific_gravity'].mode()[0],inplace=True)
```

In [21]:
```python
#label encoder
```

In [22]:
```python
from sklearn.preprocessing import LabelEncoder
for i in catcols:
    ("LABEL ENCODING OF:",i)
    LEi=LabelEncoder()
    print(c(data[i]))
    data[i]=LEi.fit_transform(data[i])
    print(c(data[i]))
    print("*"*100)
```

```
Counter({'no': 253, 'yes': 147})
Counter({0: 253, 1: 147})
****************************************************************************************************
Counter({'no': 263, 'yes': 137})
Counter({0: 263, 1: 137})
****************************************************************************************************
Counter({'no': 366, 'yes': 34})
Counter({0: 366, 1: 34})
****************************************************************************************************
Counter({'notpresent': 378, 'present': 22})
Counter({0: 378, 1: 22})
****************************************************************************************************
Counter({'normal': 324, 'abnormal': 76})
Counter({1: 324, 0: 76})
****************************************************************************************************
Counter({0.0: 339, 2.0: 18, 3.0: 14, 4.0: 13, 1.0: 13, 5.0: 3})
Counter({0: 339, 2: 18, 3: 14, 4: 13, 1: 13, 5: 3})
****************************************************************************************************
Counter({1.02: 153, 1.01: 84, 1.025: 81, 1.015: 75, 1.005: 7})
Counter({3: 153, 1: 84, 4: 81, 2: 75, 0: 7})
****************************************************************************************************
```

```
Counter({1.02: 153, 1.01: 84, 1.025: 81, 1.015: 75, 1.005: 7})
Counter({3: 153, 1: 84, 4: 81, 2: 75, 0: 7})
****************************************************************************************************
Counter({'normal': 353, 'abnormal': 47})
Counter({1: 353, 0: 47})
****************************************************************************************************
Counter({'notpresent': 358, 'present': 42})
Counter({0: 358, 1: 42})
****************************************************************************************************
Counter({'no': 340, 'yes': 60})
Counter({0: 340, 1: 60})
****************************************************************************************************
Counter({'good': 318, 'poor': 82})
Counter({0: 318, 1: 82})
****************************************************************************************************
Counter({0.0: 245, 1.0: 44, 2.0: 43, 3.0: 43, 4.0: 24, 5.0: 1})
Counter({0: 245, 1: 44, 2: 43, 3: 43, 4: 24, 5: 1})
****************************************************************************************************
Counter({'ckd': 250, 'notckd': 150})
Counter({0: 250, 1: 150})
****************************************************************************************************
Counter({'no': 324, 'yes': 76})
Counter({0: 324, 1: 76})
****************************************************************************************************
```

In [23]:
```
#DATA SPLIT INTO TRAIN SET AND TEST SET
```

In [24]:
```
selcols=['age','blood_urea','blood glucose random','coronary_artery_disease','anemia','pus_cell','red_blood_cells',
         'diabetesmellitus','pedal_edema']

x=pd.DataFrame(data,columns=selcols)
y=pd.DataFrame(data,columns=['class'])
print(x.shape)
print(y.shape)
```

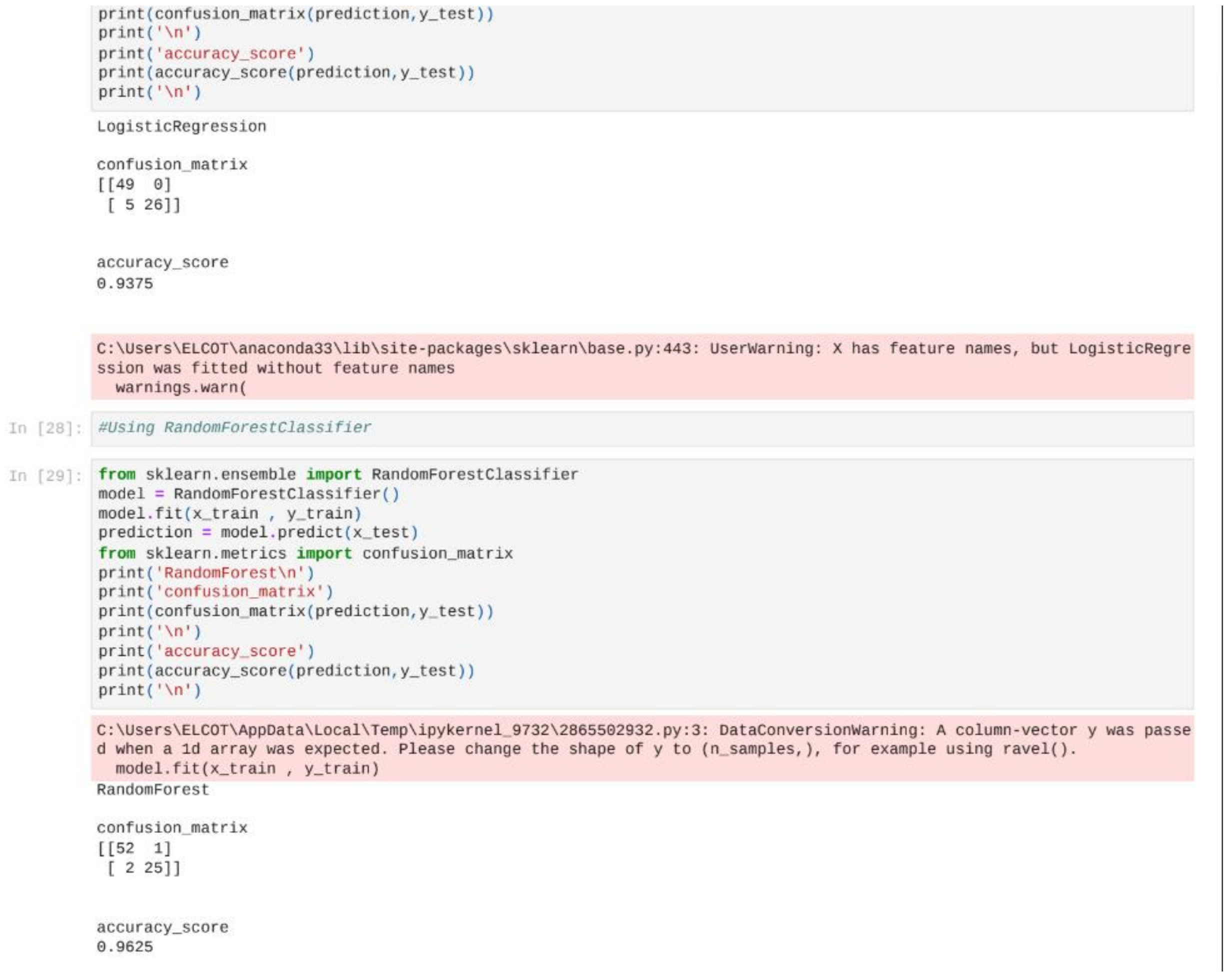
```
(400, 9)
(400, 1)
```

In [25]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

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

In [26]:
```
#MODEL BUILDING using LogisticRegression
```

In [27]:
```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression(solver ='lbfgs',max_iter=500)
print('LogisticRegression\n')
model.fit(x_train.values,y_train.values.ravel())
prediction = model.predict(x_test)
from sklearn.metrics import confusion_matrix
print('confusion_matrix')
```



```
print(confusion_matrix(prediction,y_test))
print('\n')
print('accuracy_score')
print(accuracy_score(prediction,y_test))
print('\n')
```

```
LogisticRegression

confusion_matrix
[[49  0]
 [ 5 26]]


accuracy_score
0.9375


C:\Users\ELCOT\anaconda33\lib\site-packages\sklearn\base.py:443: UserWarning: X has feature names, but LogisticRegre
ssion was fitted without feature names
  warnings.warn(
```

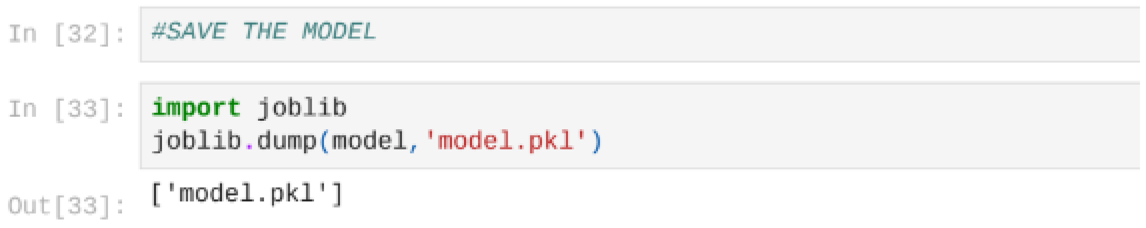
```
In [28]: #Using RandomForestClassifier
```

```
In [29]: from sklearn.ensemble import RandomForestClassifier
         model = RandomForestClassifier()
         model.fit(x_train , y_train)
         prediction = model.predict(x_test)
         from sklearn.metrics import confusion_matrix
         print('RandomForest\n')
         print('confusion_matrix')
         print(confusion_matrix(prediction,y_test))
         print('\n')
         print('accuracy_score')
         print(accuracy_score(prediction,y_test))
         print('\n')
```

```
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_9732\2865502932.py:3: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(x_train , y_train)
RandomForest

confusion_matrix
[[52  1]
 [ 2 25]]


accuracy_score
0.9625
```



```
In [32]: #SAVE THE MODEL

In [33]: import joblib
         joblib.dump(model,'model.pkl')

Out[33]: ['model.pkl']
```

Github Project Link:
https://github.com/IBM-EPBL/IBM-Project-42151-1660652735