Project Report 2022

EARLY DETECTION OF CHRONIC KIDNEY DISEASE

Team ID : PNT2022TMID20576

Team Leader : Anandh A (49621915002)

Team Meamber 1 : Hariharasudhan P (49621915005)

Team Meamber 2 : Divyashri S (49621915020)

Team Meamber 3 : Ishwarya R (49621915006)

Team Member 4 : Ram Kishore V (49621915403)

Date: 19-11-2022

Table of Content

1. **INTRODUCTION**
    1. Project Overview
    2. Purpose
2. **LITERATURE SURVEY**
    1. Existing problem
    2. References
    3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    1. Empathy Map Canvas
    2. Ideation & Brainstorming
    3. Proposed Solution
    4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    1. Functional requirement
    2. Non-Functional requirements
5. **PROJECT DESIGN**
    1. Data Flow Diagrams
    2. Solution & Technical Architecture
    3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    1. Sprint Planning & Estimation
    2. Sprint Delivery Schedule
    3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
8. **TESTING**
    1. Test Cases
    2. User Acceptance Testing
9. **RESULTS**
        1. Performance Metrics
1. **ADVANTAGES & DISADVANTAGES**
2. **CONCLUSION**
3. **FUTURE SCOPE**
4. **APPENDIX**

Source Code

GitHub & Project Demo Link

CHAPTER 1

INTRODUCTION

Machine learning is simply making healthcare smarter. There are several obstacles impeding faster integration of machine learning in healthcare today. One of the biggest challenges is the ability to obtain patient data sets which have the necessary size and quality of samples needed to train state-of-the-art machine learning models. Since patient data is protected by strict privacy and security rules, the data is not easy to collect, share and distribute. Furthermore, there are challenges with the format and quality of data which usually require significant effort to clean and prepare for machine learning-analyses.

1.1 PROJECT OVERVIEW

Chronic Kidney Disease (CKD) is a major medical problem and can be cured if treated in the early stages. Usually, people are not aware that medical tests we take for different purposes could contain valuable information concerning kidney diseases. Consequently, attributes of various medical tests are investigated to distinguish which attributes may contain helpful information about the disease. The information says that it helps us to measure the severity of the problem and we make use of such information to build a machine learning model that predicts Chronic Kidney Disease.

1.2 PURPOSE

Testing asymptomatic individuals for CKD is justified because earlier identification may enable the implementation of therapeutic measures and prevent unwarranted exposure to nephritic substances, both of which may slow the progression of CKD to end-stage renal disease.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

In the existing system they had used machine learning as classifiers in order to predict the chronic kidney disease. In the prediction they had considered around 25attributes in the dataset and finds out which is better in terms of accuracy. In their case the accuracy achieved was 98.46% and this accuracy rate cannot be achieved in case of larger dataset.

2.1.1 EXISITING SYSTEM

In the existing system, they had developed a model to predict CKD disease in patients. The machine learning classifiers such as C5.0, logistic regression, Linear support vector machine,K- Nearest Neighbourhood random tree were used for training the model.

2.1.1.1 C5.0

C5.0 is a type of decision tree because it creates the decision tree from the input. The tree has the number of branches. It utilizes the tree structure to model the relationship between features and potential outcomes. At each node of the tree the attribute of the dataset is chosen. It can handle nominal and numeric features. C5.0 is the extended version of the C4.5 classification algorithm and uses information entropy concept. Entropy is used for finding the impurity of features. Information-entropy is produced based on the calculation of parent and

child entropy values. This process is iterative and works until there is no the further split. This algorithm gave accuracy of about 92.68%.

2.1.1.2 Logistic regression

Logistic regression is also a type of supervised learning algorithm. It is a statistical model. It is divided the target attribute into two-classes: successor not success. For success, it returns 1 whereas it returns 0 for not succeeding. It is used in various field of machine learning application in social sciences and medical arena, for example, for spam detection, diabetes detection, cancer detection, etc. Through this technique, we only concern about the probability of the outcome variable.This algorithm gave accuracy of about 51.22%.

2.1.1.3 CHAID

Chi-square automatic interaction detection (CHAID) is a type of decision tree technique. For each categorical predictor, all possible cross-tabulation is created in the CHAID model and its process works until the best outcome is attained. The target or dependent variable becomes a root node in the tree, the target variable is split into two or more parts as per the categories in target variable and child of the root node are created using the statistical method and variable relationship. Such a process will be till leaf nodes of the tree. F test is used for the continuous dependent variable and the Chi-square test is used for the categorical dependent variable. This algorithm gave accuracy of about 92.68%.

2.1.1.4 Linear Support Vector Machine(LSVM)

Linear support vector machine is the modern particularly fast machine learning algorithm for solving multi class classification problem for the large dataset based on a simple iterative approach. It is created the SVM model in linear CPU time of the dataset. LSVM can be used for the high dimensional dataset is the sparse and dense format. SupportVector Machine is a supervised classifier algorithm. It is used kernel trick for solving the classification problem. Based on these transformations, ideal edge is found between the possible outputs. SVM is used for the non-linear kernel, such as RBF. For the linear kernel, LSVM is an appropriate choice. LSVM classifier is sufficient for all linear problems. This algorithm gave accuracy of about 95.12%.

2.1.1.5 K- Nearest Neighbours (KNN)

KNN is a simple type of supervised algorithm. It can be used for both classification and regression problems. However, it is largely used for classification problems. KNN does not use a particular training stage and use all the data for training so that it is a lazy learning algorithm and also it does not consider anything about the underlying data, so that is a non-parametric learning algorithm. KNN stores the whole dataset because it has no model so that there is no learning required. When the new data enters for predicting the outcomes, it compares k- nearest neighbours so that selection of K’s value is very important. The distance is calculated between two already label data. The distance helps to find the nearest neighbour of the new data. This algorithm gave accuracy of about 53.17%.

2.1.1.6 Random tree

The random tree is a type of supervised classifiers. It produce slots of distinct learners. The stochastic process is used to form the tree. It is a type of ensemble learning technique for classification. It works the same as decision tree, but a random subset of attributes uses for each split. This algorithm uses for both classification problems and regression problems. A group of random trees is known as a forest. The random trees classifier takes the input feature set and classifies input for every tree in the forest. The output of the random tree selects from the majority of votes. In the tree, every leaf node holds a linear model. The bagging training algorithm is used to train the model. This algorithm gave accuracy of about87.80%. When compared between the above algorithmsLSVM gave the highest accuracy of about 95.12% and KNN gave the least accuracy of about 53.17%.

2.2 REFERENCES

1. **Q.-L. Z hang**,et.al "Prevalence of chronic kidney disease in population-based studies: Systematic review," BMC Public Health, vol. 8, no. 1, p. 117, Dec. 2008.8

2. **W. M. McClellan**,et.al ''Albumin- urea and racial disparities in the risk for ESRD,'' J. Amer. Soc. Nephrol., vol. 22, no. 9, pp. 1721–1728, Aug. 2011.

3. **M. K. Haroun**,et.al “Risk factors for chronic kidney disease: A prospective study of 23,534 men and women in Washington County, Maryland,” J. Amer. Soc. Nephrol., vol. 14, no. 11, pp. 2934–2941, Nov. 2003.

4. **W. D. Souza**,et.al “Incidence of chronic kidney disease hospitalisations and mortality in Espírito Santo between 1996 to 2017,” Wisit Cheungpasitporn, Univ. Mississippi Medical Center, Rochester, MN, USA, Tech. Rep., 2019, doi: 10.1371/journal.pone.0224889.

5. **W. Mula-Abed**,et.al “Estimated glomerular filtration rate (eGFR): A serum creatinine-based test for the detection of chronic kidney disease and its impact on clinical practice,” Oman Med. J., vol. 27, no. 4, pp. 339–340, 2012.

6. **A. S. Levey**,et.al “Proteinuria as a surrogate outcome in CKD: Report of a scientific workshop sponsored by the nationalkidney founda- tion and the US food and drug administration,” Amer. J. Kidney Diseases, vol. 54, no. 2, pp. 205–226, Aug. 2009.

7. **S. Gerogianni**,et.al “Concerns of patients on dialysis: A research study,” Health Sci. J., vol. 8, no. 4, pp. 423–437, 2014.

8. **K.R. A. Padmanaban**,et.al “Applyingmachine learning techniques for predicting the risk of chronic kidney disease,” Indian J. Sci. Technol., vol.

9, no. 29, Aug. 2016.

9. **W.Gunarathne**,et.al"Performance evaluation on machine learning classification techniques for disease classification,forecasting through data analytics for chronic kidney disease (CKD)," in Proc. IEEE 17th Int. Conf. Bioinf. Bioeng. (BIBE), Oct. 2017, pp. 291–296.

10. **S. Drall**,et.al "Chronic kidney disease prediction using machine learning: A new approach," Int. J. Manage., Technol. Eng., vol. 8, pp. 278–287, May 2018.

11. **M. Almasoud**,et.al "Detection of chronic kidney disease using machine learning algorithms with least number of predictors," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 8, pp. 89–96, 2019.

12. **S.Shankar**,et.al"Analysis and prediction of chronic kidney disease," Int. Res. J. Eng. Technol., vol. 7, no. 5, May 2020, pp. 4536–4541.

13. **S. Vijayarani1**,et.al"Kidney disease prediction using SVM and ANN algorithms," Int. J. Comput. Bus. Res., vol. 6, no. 2, pp. 1–12, Mar. 2015.

14. **J.Xiao**,et.al "Comparison and development of machine learning tools in the prediction of chronic kidney disease progression," J. Transl. Med., vol. 17, p. 119, Dec. 2019.

15. **M.S. Gharibdousti**,et.al "Prediction of chronic kidney disease using

data mining techniques," in Proc. Ind. Syst. Eng. Conf., K.Coperich, E.Cudney, H. Nembhard, Eds., 2017, pp. 2135–2140.

16. **E. M. Karabulut**,et.al "A comparative study on the effect of feature selection on classification accuracy," Procedia Technol., vol. 1, pp. 323–327, Jan. 2012.

2.3 PROBLEM STATEMENT

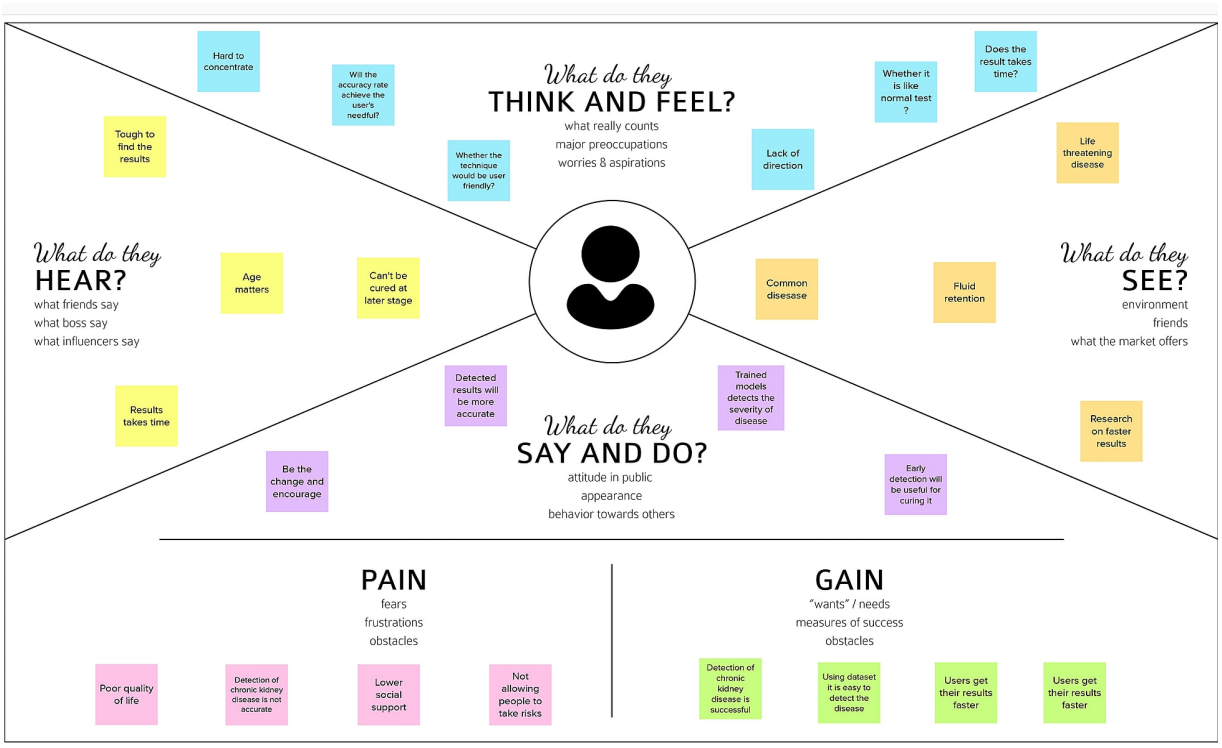
Kidney disease is most frequently brought on by diabetes. However, obesity and heart disease can also contribute to the harm that results in renal failure. Long-term functional decline can also be brought on by problems with the urinary system and inflammation in various kidney regions. The major issue while designing the proposed system are there are limited number of training samples available and the accuracy of CKD classification is not high. There is no feature selection involved which might affect accuracy.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

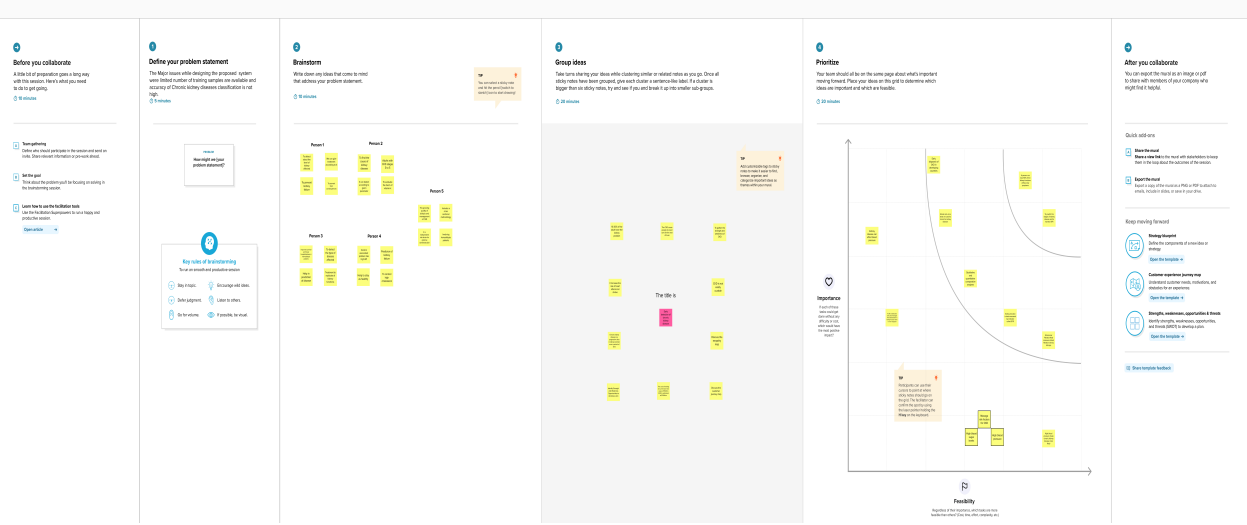
3.1 EMPATHY MAP CANVAS

An empathy map is a collaborative tool that allows teams to develop a better understanding of their customers. An empathy map, like a user persona, can represent a group of users, such as a consumer segment. Dave Gray invented the empathy map, which has grown in popularity among the agile community.



3.2 IDEATION AND BRAINSTORMING

Brainstorming is a problem-solving strategy used in groups that involves the spontaneous production of innovative ideas and solutions. This strategy necessitates a lengthy, free-flowing debate in which each member of the group is encouraged to think aloud and propose as many ideas as possible based on their diverse knowledge.



3.3 PROPOSED SOLUTION

The problem solution is categorised and certain parameters such as problem statement,Solution description,Novelty,Social impact,Business model,Scalability of the Solution.

The patient needs a way to detect chronic kidney disease at early stage accurately and cost effectively , so that the chances of curing the disease is high. • Therefore, machine learning techniques are of • great importance in the early detection of

CKD. These techniques are supportive of • experts and doctors in early diagnosis to avoid developing kidney failure. • So, the objective of this research is to provide an effective model to predict the CKD by least number of predictors.

Creating a machine learning model that uses the attributes of medical tests taken for different purposes to detect chronic kidney disease at early stage. • Evaluation is done on a patient's dataset containing 24 features like RBC count, blood pressure level, blood sugar level etc. • Deep Neural Network's accuracy can be achieved by increasing the number of hidden layers in the model.

In the proposed system we use a deep learning model which is called as Deep Neural Network which is suitable for accurate prediction. By using DNN, we can predict the chronic kidney disease with more than 95% of accuracy. In the DNN we have more hidden layers and hence its accuracy also high.
Since CKD is detected at early stages, there are high chances of curing the disease. This helps customer get right treatment at the right time.

This is a cost effective model, because when the patient uses this model they don't have to spend copious amount of money just for initial diagnosis. • This model will identify and detect the kidney disease earlier so more number of clients will approach us and it makes more profit in both sides.

This model can be expanded to include more attributes for more accurate detection. Training the model with even more attributes will increase the efficiency further

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit describes the problem that you have deal with your customer and that the solution you have realized for it actually solves the customer's problem.

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?

Signs and symptoms of chronic kidney disease develop over time if kidney damage progresses slowly. Depending on how severe it is, loss of kidney function can cause:

- Loss of appetite
- Fatigue and weakness
- Sleep problems
- Urinating more or less
- Swelling of feet and ankles
- Dry, itchy skin
- Chest pain, if fluid builds up around the lining of the heart

Signs and symptoms of kidney disease are often nonspecific. This means they can also be caused by other illnesses.

6. CUSTOMER — CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

- Lack of accuracy.
- Lack of awareness.
- By representing the data in a particular format there is a limitation of data.

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? i.e. pen and paper is an alternative to digital notetaking

- lifestyle changes – to help you stay as healthy as possible
- medicine – to control associated problems, such as high blood pressure and high cholesterol
- dialysis – treatment to replicate some of the kidney's functions, which may be necessary in advanced CKD

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers?
There could be more than one explore different sides.

The following jobs are to be done:

- Create an ML model that can predict the presence of chronic kidney disease
- Design an interactive, simple and Effective freely available UI for communicating with the patients.
- It provides an accurate information about the health of the patient's kidney.

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?
i.e. customers have to do it because of the change in regulations

- Overusing pain killers.
- Can't able to predict on early stage.
- Smoking habits and drinking habits lead to spread of chronic kidney disease.
- Due to the large intake of junk foods.

7. BEHAVIOUR — BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

- Follow up the prescription of the doctor.
- The customer should check and consult the doctor regularly.
- If the customer is fatigue or not feeling well immediately he or she should consult doctor and should take effective medication or take some rest.

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

Identify strong TR & EM

3. TRIGGERS — TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Patients are encouraged to get a kidney function test if they experience symptoms that point to potential renal issues. These signs and symptoms may include: unusual nausea and vomiting; blood in urine (hematuria) and painful urination (dysuria).

4. EMOTIONS: BEFORE / AFTER — EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before: Decreased mental sharpness, Results take time, Hypertension
After: Patients experience works on interacting with the trained model .Research on faster results.

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.

In the proposed system we use a deep learning model which is called as Deep Neural Network which is suitable for accurate prediction. By using DNN, we can predict the chronic kidney disease with more than 95% of accuracy. In the DNN we have more hidden layers and hence its accuracy also high.

8. CHANNELS of BEHAVIOUR — CH

ONLINE

What kind of actions do customers take online? Extract online channels from #7

- The test results are submitted based on the requirement of the software, the model process and give the result.
- The customer can view the results via the UI.

OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Taking test after receiving the result, If the test result is above threshold range the patient should

Extract online & offline CH of BE

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

The functional requirements of the proposed solution are categorized into different ways.

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Home Page (Login Page)	• Introduction page of the website. • Symptoms and steps to cure will be displayed. • If the user already exists asks to **login** or else redirects to **Sign Up.**
FR-2	User Sign Up Page	The user had to enter the username, phone number and password.
FR-3	User Verification	After getting the phone number the OTP will be sent via SMS and it will be verified.
FR-4	Dataset Collection	Collect the data set related to Chronic Kidney Disease and process the data.
FR-5	Training the Model	By using the processed data the model will be trained again and again by using back propagation techniques.
FR-6	Testing the Model	By using 20% of dataset the model will be tested.
FR-7	Prediction	By using the data collected from the tested model the result is predicted.

4.2 NON FUNCTIONAL REQUIREMENT

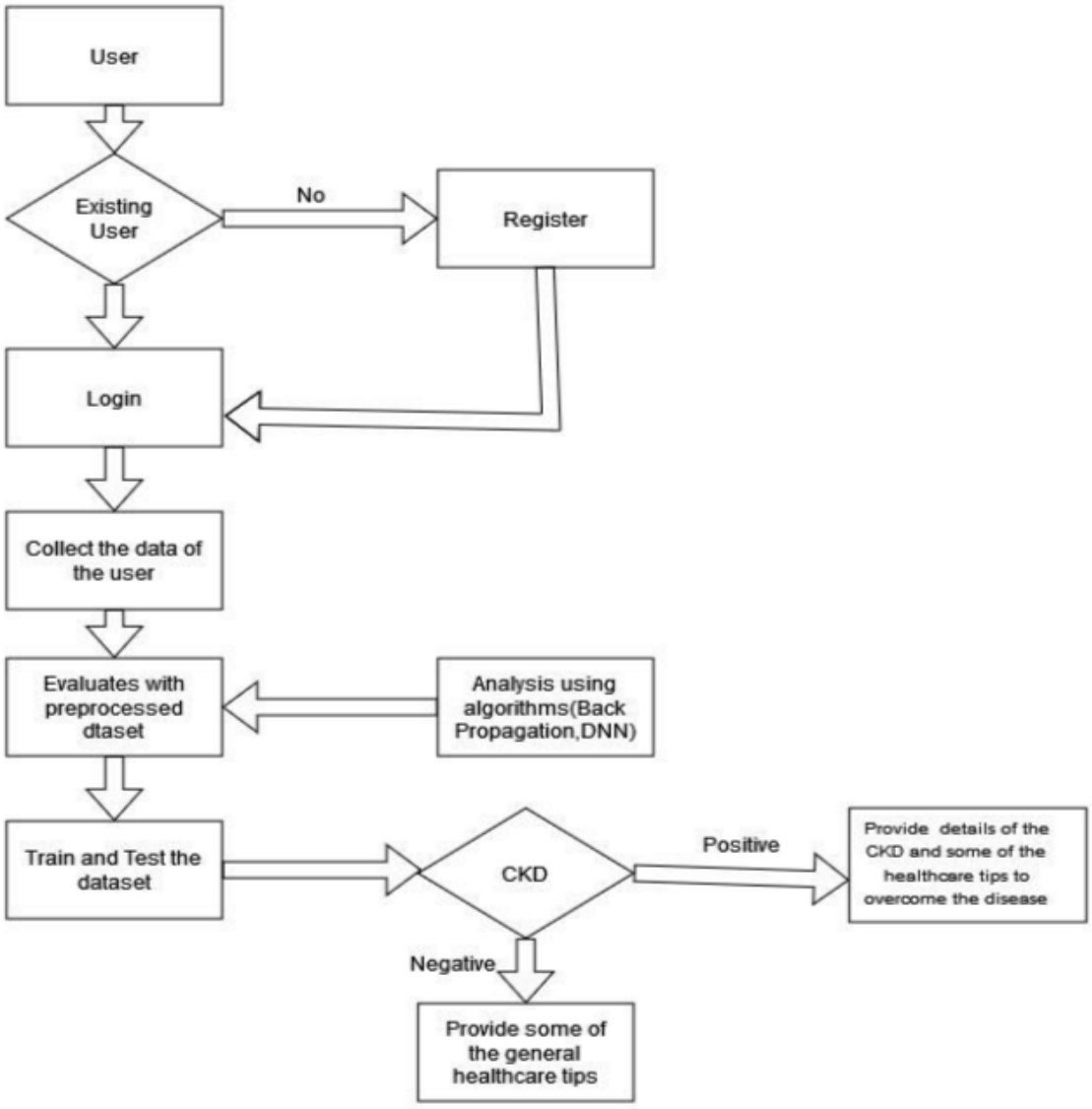
The Non functional requirements of the proposed solution are categorized into different ways.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Creating a machine learning model that uses the attributes of medical tests taken for different purposes to detect chronic kidney disease at early stage.
NFR-2	**Security**	The reports are maintained confidentially to the customer.
NFR-3	**Reliability**	The model will identify and detect the kidney disease earlier, so more number of clients will approach us and it results how the model is more reliable to the customers.
NFR-4	**Performance**	By using DNN, we can predict the chronic kidney disease with more than 95% of accuracy. In the DNN we have more hidden layers and hence its accuracy also high.
NFR-5	**Availability**	It is used a website(UI) and trained model to predict it will work at any time.
NFR-6	**Scalability**	This model can be expanded to include more attributes for more accurate detection. Training the model with even more attributes will increase the efficiency further.

CHAPTER 5
PROJECT DESIGN

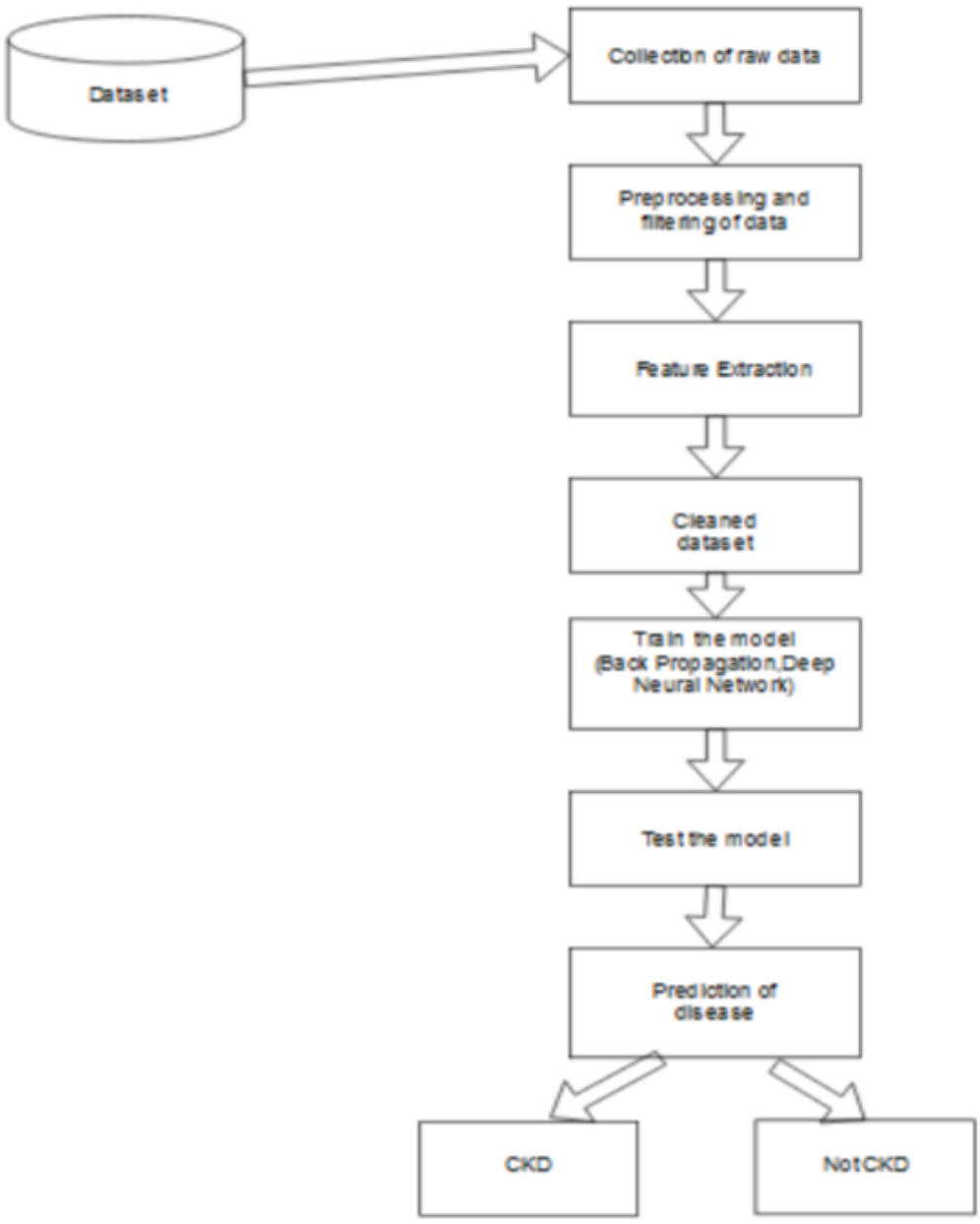
5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

Technology Architecture



Components and Technologies

S.No	Component	Description	Technology
1.	User Interface	User interact with our application through web User Interface.	HTML, CSS and Python flask.
2.	Registration	The user details will be stored and it will be used for further process.	HTML ,CSS, Python flask
3.	Login	Logic for a process in the application	IBM Watson STT service
4.	Client's input collection	User enters their diagnose report	Front end- HTML ,CSS ,MySQL,Pytjon flask Back end-Python
5.	Database	For user registration and login process	MySQL
6.	Machine Learning Model	Deep Learning Model gives 98% accuracy	Deep Learning Neural Network (DNN).

Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	International Business Machines	Cloud
2.	Security Implementations	Authentication using stored data for login and CAPTCHA	Encryptions and Authentication
3.	Scalable Architecture	This model can be expanded to include more attributes for more accurate detection. Training the model with even more attributes will increase the efficiency further.	Performance optimization
4.	Availability	It is used a website(UI) and trained model to predict , it will work at any time.	Web development
5.	Performance	By using DNN, we can predict the chronic kidney disease with more than 95% of accuracy. In the DNN we have more hidden layers and hence its accuracy also high.	Deep Neural Network and back propagation

5.3 User Stories

Use the below template to list all the user stories for the product

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	New user enters into the System He/ She can register into the Application by entering user details such as username and mobile number .	I can access my account / dashboard	High	Sprint-1
		USN-2	The user will receive OTP through SMS.	I can receive OTP & click confirm	High	Sprint-2
	Login	USN-3	After Successful registration the user can Log into the application by entering the registered Username and Password	I can register & access the dashboard	High	Sprint-1
		USN-4	CAPTCHA will be provided to reduce the network traffic.		Medium	Sprint-1
	Dashboard	USN-5	User can get into the Dashboard only when the Verification Successful. After the user can access the displayed information in the Dashboard		Low	Sprint-2
	Data collection	USN-6	Diagnosed result data will be entered by the user	Data will be collected in standard format	Medium	Sprint-3
	Prediction result	USN-7	By the collected data the trained model will predict and display the result	Display Result to the user.	High	Sprint - 4
		USN-8	Based on the result the suggestion varies.	Suggestions to improve	Low	Sprint-4
Customer Care Executive	Further Clarification	USN-9	The problems which are faced by the user while using the application can be clarified	If any doubt arises they can contact the customer care	Medium	Sprint-4

CHAPTER 6
PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	New user enters into the System. He/ She can register into the Application by entering user details such as username and mobile number	2	High	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-1	Login	USN-2	After Successful registration the user can Log into the application by entering the registered Username and Password	2	High	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-1		USN-3	CAPTCHA will be provided to reduce the network traffic.	2	Medium	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-2	Dashboard	USN-4	User can get into the Dashboard only when the Verification Successful. After the user can access the displayed information in the Dashboard	3	Medium	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-3	Data collection	USN-5	Diagnosed result data will be entered by the user.	2	Medium	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-4	Prediction result	USN-6	By the collected data the trained model will predict and display the result.	2	High	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-4		USN-7	Based on the result the suggestion varies.	2	Low	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-1	Dataset Collection	USN-8	Chronic Kidney Disease dataset identification	2	High	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore

Sprint-1	Clean the Dataset	USN-9	The dataset had to be cleaned. Cleaning process includes removing null values, Replacing missing values, segregation of test and train data.	3	High	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-2	Train ML Model in IBM	USN-10	The model will be trained in IBM.	4	High	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-3	Save Model	USN-11	The model will be stored in pickle format	3	Medium	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-3	Model Testing	USN-12	The model will be tested using the test data	3	High	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-3	Integration	USN-14	HTML file and python Code Integration	2	Medium	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore
Sprint-4	Deployment	USN-15	The model will be deployed in Cloud	3	Medium	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore

Sprint-4	Further Clarification	USN-16	The problems which are faced by the user while using the application can be clarified	2	Medium	A.Anandh, P.Hariharasudhan, S.Divyashri, R.Ishwarya, V.Ram Kishore

6.2 Sprint Delivery Schedule

Project Tracker

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	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	7	6 Days	07 Nov 2022	12 Nov 2022	7	07 Nov 2022
Sprint-4	9	6 Days	14 Nov 2022	19 Nov 2022	9	14 Nov 2022

Velocity

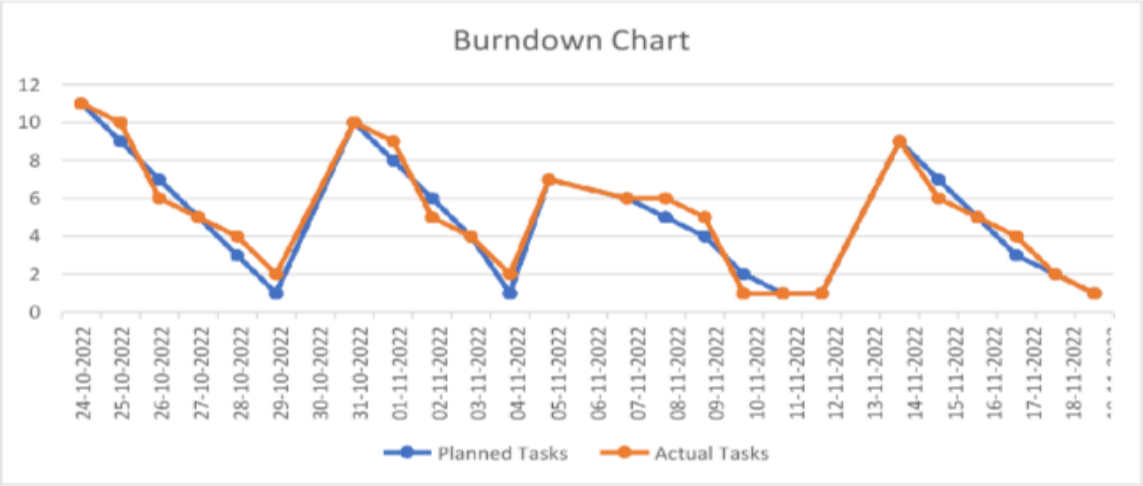
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

Sprint 1 AV = Sprint duration/velocity = 11/6 = 1.83
Sprint 2 AV = Sprint duration/velocity = 10/6 = 1.67
Sprint 3 AV = Sprint duration/velocity = 7/6 = 1.16
Sprint 4 AV = Sprint duration/velocity = 9/6 = 1.5

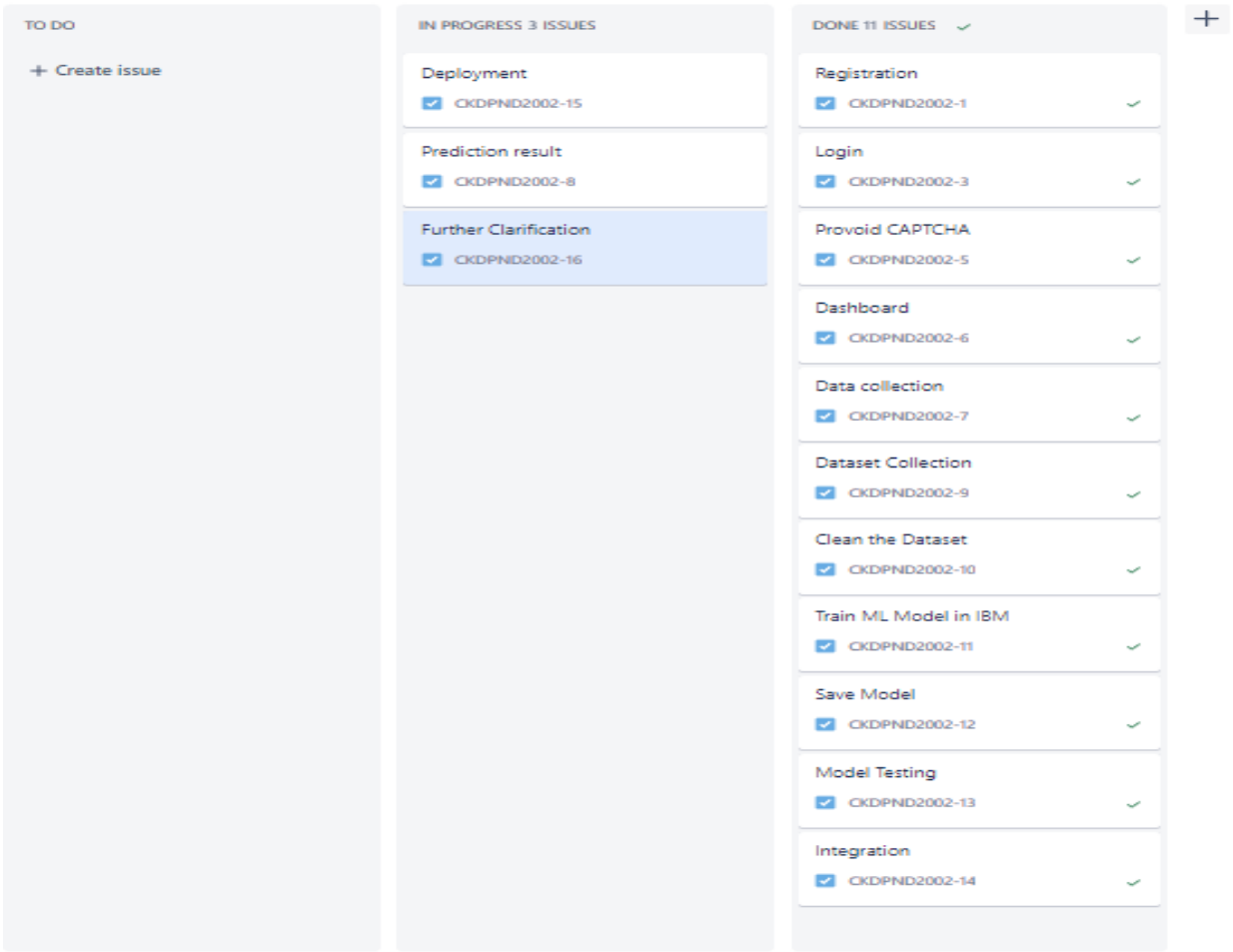
Burn down Chart

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. However,

burn down charts can be applied to any project containing measurable progress over time.



6.3 Reports from JIRA



CHAPTER 7

CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

 # Splitting Dependent and Independent Variable

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30,
random_state = 121)#101
print("X train value")
print(X_train)
print("Y train value")
print(y_train)
```

The sklearn. metrics module implements several loss, score, and utility functions to measure classification performance. Some metrics might require probability estimates of the positive class, confidence values, or binary decisions values.

test_size:float or int, default=None

Float- Between 0.0 and 1.0 and represent the proportion of the dataset to include in the test split.

Int- Represents the absolute number of test samples.

None- The value is set to the complement of the train size. If train_size is also None, it will be set to 0.30.

train_sizefloat or int, default=None

Float- Between 0.0 and 1.0 and represent the proportion of the dataset to include in the train split.

Int-Represents the absolute number of train samples.

None- The value is automatically set to the complement of the test size.

random_stateint, RandomState instance or None, default=None

Controls the shuffling applied to the data before applying the split. Pass an int for reproducible output across multiple function calls.

7.2 Feature 2

#Hidden and Output Layer

```
model.add(tf.layers.Dense(128,activation='relu'))
model.add(tf.layers.Dense(256,activation='relu'))
model.add(tf.layers.Dense(512,activation='relu'))
model.add(tf.layers.Dense(1,activation='sigmoid'))
#Compile Model
model.compile(loss="binary_crossentropy",optimizer='adam',metrics=['accuracy'])
model.fit(X_train,y_train,epochs=1000)
```

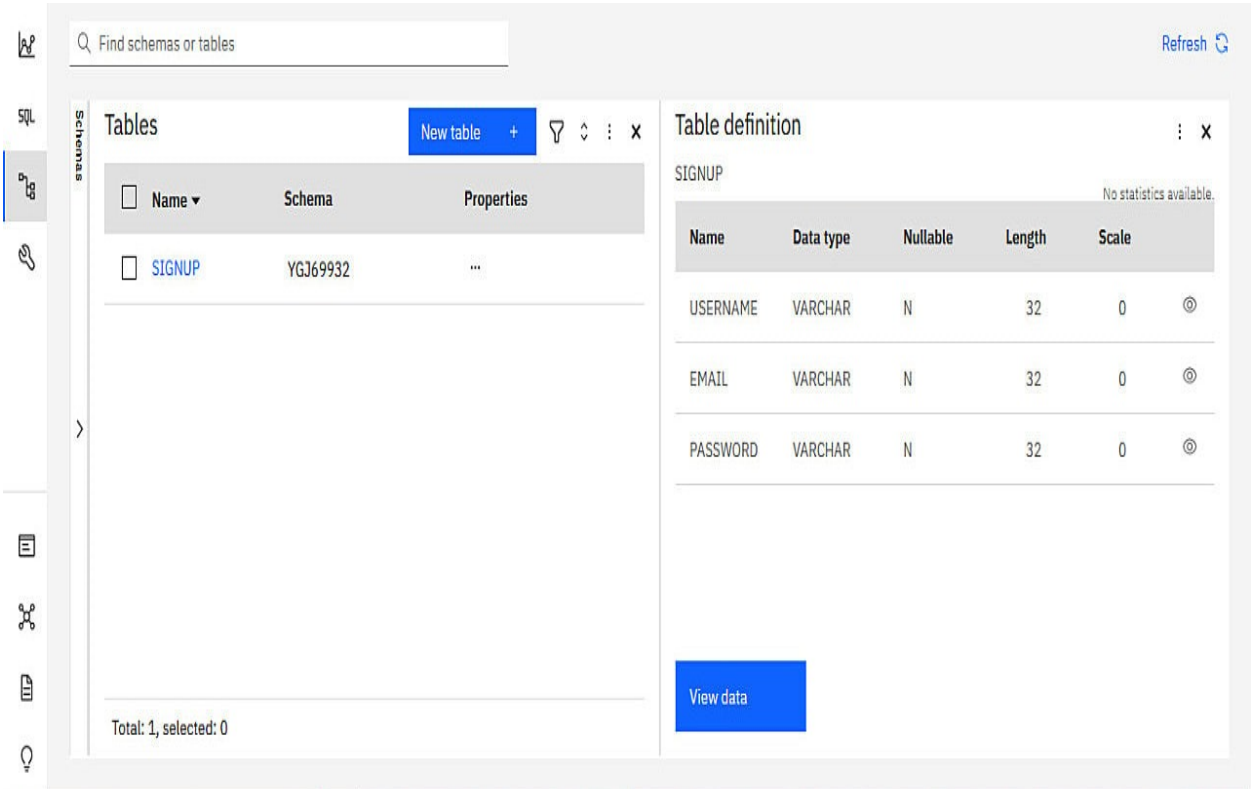
Sigmoid activation function

Applies the sigmoid activation function. For small values (<-5), sigmoid returns a value close to zero, and for large values (>5) the result of the function gets close to 1.Sigmoid is equivalent to a 2-element Softmax, where the second element is assumed to be zero. The sigmoid function always returns a value between 0 and 1

RELU activation function

The rectified linear activation function or ReLU is a non-linear function or piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.It is the most commonly used activation function in neural networks, especially in Convolutional Neural Networks (CNNs) & Multilayer perceptrons.

7.3 Database Schema (if Applicable)



CHAPTER 8

TESTING

8.1 Test Cases

CKD-Test case

[6.00001, 1.02500, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000, 9.60001, 3.30001, 9.0001, 1.47002,
4.50000, 1.69001, 4.10001, 7.20003, 5.00000, 0.00000,
0.00000, 0.00000, 1.00000, 0.00000, 0.00000]

No CKD-Test case

[1.10002, 1.01500, 3.00000, 0.00000, 1.00000, 0.00000,
1.00000, 0.00000, 1.06002, 2.15002, 1.52001, 1.20002,
5.70000, 8.60000, 2.60001, 5.00003, 2.50000, 1.00000,
0.00000, 1.00000, 1.00000, 0.00000, 1.00000]

8.2 User Acceptance Testing

The main goal of UAT is to demonstrate that the given solution satisfies the business process, and all business processes, including batch, periodic, and scheduled processes, will be exercised end-to-end. In general, verification of records, record content, values, and volumes will also be carried out to make sure that nothing has been lost and functionality has not been negatively impacted by the introduction of the new or updated version of the software, especially in financial applications.Deep neural network works on vast datasets and,functions like the brain, which performs repetitive learning and automatically predicts the result with better accuracy.

CHAPTER 9

RESULTS

9.1 Performance Metrics

```
[ ] from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test,ypred)
    print(cm)

    [[10  4]
     [ 0 34]]
```

Fig. Confusion Matrix

The confusion matrix of test data which is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. Confusion matrices are useful because they give direct comparisons of values like true positive,false positive, true negative and false negative.

```
[ ] from sklearn.metrics import accuracy_score
    accuracy_score(y_test,ypred)

    0.9166666666666666
```

Fig. Accuracy

CHAPTER 10
ADVANTAGES & DISADVANTAGES

Advantage

- Train deep learning algorithms and still obtain insights which are relevant to the purpose of the training. For instance, you can use deep learning algorithms to uncover any existing relations between industry analysis, social media chatter, and more to predict upcoming stock prices of a given organization.
- A deep learning model becomes able to perform thousands of routine, repetitive tasks within a relatively shorter period of time compared to what it would take for a human being. images become challenging because of different reasons, deep learning can account for those variations and learn valuable features to make the inspections robust.

Disadvantage

- Deep learning relies on data analysis to build its training process. However, the amount of time needed to ensure an effective training process is limited by the fast-moving and streaming input data.
- Deep learning software's inability to explain its reasoning for reaching a certain conclusion is a significant drawback. You cannot follow an algorithm, unlike in the case of conventional machine learning, to determine why your system determined that a photo was of a cat and not a dog. You must modify the entire algorithm in order to fix faults in Deep Learning algorithms.

CHAPTER 11

CONCLUSION

In our project we used Deep Neural Network for predicting chronic kidney disease. It is a one of the deep learning technique which can able to train and predict kidney disease based upon input data. We used chronic kidney disease dataset which holds over 24 features for training purpose. Once the model is trained with the help of training dataset, the DNN model is tested with the test dataset considered. Our model achieves more than 98% accuracy during testing andtraining. The predicted results by DNN is accurate and stable, its patterns are also matched with the existing dataset patterns. And hence our model is perfectly trained and it can able to predict the kidney disease with high stability. Our future work is to improve the accuracy by using hybrid approach.

CHAPTER 12

FUTURE SCOPE

Our future work is to improve the accuracy by using hybrid approach.Hybrid techniques are a combination of two or more computational techniques which provide more advantage to detect components than any other individual technique. So this approach help us to improvise the CKD model in different analysis which increases productivity. The hybrid technique helps qualitative research to be effective.

CHAPTER 13

APPENDIX

Source Code

```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
import tensorflow
from tensorflow import keras as tf
from keras.layers import Dense, Activation
from keras import Sequential
from keras.models import load_model as tl
from tensorflow.keras.optimizers import Adam
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3
def __iter__(self): return 0
# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='XkilApJJEZwMiGeavh4VUZe9m6Gq7QwGVc56JiwDLQrn',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
```

```
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')
bucket = 'chronickidneymodel-donotdelete-pr-lsxbf6kbtsagbx'
object_key = 'CKD.csv'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body)
data = pd.read_csv(body)
data.head()
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3
def __iter__(self): return 0
# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='fMSxo9CALQ88Bgm7xxzu4dYjvizNR4rudrxp7OSvak24',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')
bucket = 'ckdmodel-donotdelete-pr-gk6q4iwp7ye7qz'
object_key = 'CKD.csv'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
```

```
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body
)
data = pd.read_csv(body)
data.head()
print(data.info())
data.drop('id', inplace=True, axis=1)
data.info()
data=data.dropna(how="any")
print(data)
data['rbc'] = data['rbc'].map({"abnormal":1,"normal":0})
data['pc'] = data['pc'].map({"abnormal":1,"normal":0})
data['pcc'] = data['pcc'].map({"present":1,"notpresent":0})
data['ba'] = data['ba'].map({"present":1,"notpresent":0})
data['htn'] = data['htn'].map({"yes":1,"no":0})
data['dm'] = data['dm'].map({"yes":1,"no":0})
data['cad'] = data['cad'].map({"yes":1,"no":0})
data['pe'] = data['pe'].map({"yes":1,"no":0})
data['ane'] = data['ane'].map({"yes":1,"no":0})
data['appet'] = data['appet'].map({"poor":0,"good":1})
data['classification'] = data['classification'].map({"ckd":0,"notckd":1})
data['pcv'] = data['pcv'].astype('int')
data['wc'] = data['wc'].astype('int')
data['rc'] = data['rc'].astype('float')
print(data)
X = data.iloc[:,1:24].values
y = data.iloc[:, 24].values
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30,
random_state = 121)#101
print("X train value")
print(X_train)
print("Y train value")
print(y_train)
model = tf.Sequential()
model.add(tf.layers.Dense(64,input_dim=23,activation='relu')
model.add(tf.layers.Dense(128,activation='relu'))
model.add(tf.layers.Dense(256,activation='relu'))
model.add(tf.layers.Dense(512,activation='relu'))
model.add(tf.layers.Dense(1,activation='sigmoid'))
model.compile(loss="binary_crossentropy",optimizer='adam',metrics=['accuracy'])
model.fit(X_train,y_train,epochs=1000)
ypred=model.predict(X_test)
ypred = ypred.round()
print(ypred)
print(y_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,ypred)
print(cm)
!pip install ibm_watson_machine_learning
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey":"C7fEAUL0ReUvFh_BmNWalkIv8aFsOFonGjAIcXrGKm5P"
```

```
    }
client = APIClient (wml_credentials)
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    #print(space)
        return(next(item for item in space['resources'] if item['entity']["name"] ==
space_name)['metadata']['id'])
space_uid = guid_from_space_name(client,'CKDspace')
print("Space UID = "+ space_uid)
client.set.default_space(space_uid)
client.software_specifications.list(limit=100)
software_spec_uid                                                                    =
client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid
model.save('models/ckd.h5')
cd models
!tar -zcvf cloudDeployment.tgz ckd.h5
ls -1
!pip install watson-machine-learning-client --upgrade
!pip install -U ibm-watson-machine-learning
model_details     =     client.repository.store_model(model='cloudDeployment.tgz',
meta_props={
    client.repository.ModelMetaNames.NAME: "CKD_Model",
    client.repository.ModelMetaNames.TYPE: "tensorflow_2.7",
                          client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:
software_spec_uid })
model_details
```

```
model_id = client.repository.get_model_id(model_details)
model_id
client.repository.download(model_id,'CKD_model.tar.gz')
import tensorflow
from tensorflow import keras as tf
from keras.layers import Dense, Activation
from keras import Sequential
from keras.models import load_model
from tensorflow.keras.optimizers import Adam
model = load_model("ckd.h5")
```

GitHub & Project Demo Link

GitHub Link - https://github.com/IBM-EPBL/IBM-Project-32521-1660210481

Project Demo Link - http://surl.li/dtnue