

ABSTRACT

Chronic Kidney Disease is a serious lifelong condition induced by either kidney pathology or reduced kidney functions. Early prediction and proper treatments can stop, chronic disease or slow the progression of the chronic disease to the end stage, where dialysis or kidney transplantation is the only way to save a patient's life. The ability of several machine learning methods such as Random Forest, Decision Tree, and K-Nearest Neighbors (KNN) used for the early prediction of Chronic Kidney Disease. Predictive analytics is used to examine the relationship between data parameters as well as with the target class attribute. It enables us to introduce the optimal subset of parameters to feed machine learning to build a set of predictive models. The dataset used for this study was obtained from the UCI Machine Learning Repository and consisted of 24 clinical and laboratory features of 400 patients diagnosed with CKD. The data was preprocessed and feature selection was performed to remove irrelevant and redundant features. The remaining features were used to train and test the models. The results showed that Random Forest had the highest prediction of accuracy, and followed by Decision Tree, and KNN. The precision and recall scores for Random Forest were also higher than the other two algorithms. These findings suggest that Random Forest is the most effective algorithm for predicting of CKD.

Keywords: Chronic Kidney Disease, Decision Tree, K-Nearest Neighbor, Light Gradient Boosted Machine, Machine Learning, prediction, Random Forest.

LIST OF FIGURES

4.1	Architecture Diagram	11
4.2	Data Flow Diagram	12
4.3	Use Case Diagram	13
4.4	Sequence Diagram	14
4.5	Collaboration Diagram	15
4.6	Activity Diagram	16
5.1	Input Design	19
5.2	Output Design	20
5.3	Unit Testing Test Result	21
5.4	KNN Test Result	23
5.5	Decision Tree Test Result	23
6.1	KNN Test Result	25
9.1	Poster Presentation	30

LIST OF ACRONYMS AND ABBREVIATIONS

CKD	Chronic Kidney Disease
DT	Decision Tree
ESRD	End Stage Renal Disease
GFR	Glamorise Filtration Rate
GBM	Gradient Boosting Framework
KNN	K-Nearest Neighbor
LGBM	Light Gradient Boosted Machine
ML	Machine Learning

TABLE OF CONTENTS

	Page.No
ABSTRACT	v
LIST OF FIGURES	vi
LIST OF ACRONYMS AND ABBREVIATIONS	vii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Aim of the project	2
1.3 Project Domain	3
1.4 Scope of the Project	3
2 LITERATURE REVIEW	4
3 PROJECT DESCRIPTION	7
3.1 Existing System	7
3.2 Disadvantages	7
3.3 Proposed System	7
3.4 Advantages	8
3.5 Feasibility Study	8
3.5.1 Economic Feasibility	9
3.5.2 Technical Feasibility	9
3.5.3 Social Feasibility	9
3.6 System Specification	9
3.6.1 Hardware Specification	9
3.6.2 Software Specification	10
3.6.3 Standards and Policies	10
4 METHODOLOGY	11
4.1 General Architecture	11
4.2 Design Phase	12

4.2.1	Data Flow Diagram	12
4.2.2	Use Case Diagram	13
4.2.3	Sequence Diagram	14
4.2.4	Collaboration Diagram	15
4.2.5	Activity Diagram	16
4.3	Module Description	17
4.3.1	Data Collection	17
4.3.2	Data pre-processing	17
4.3.3	Training and Testing the Data	17
4.3.4	Prediction of CKD	17
4.4	Execution And Implementation Of The Project	18
4.4.1	Installation Of Google Colab Notebook	18
4.4.2	Create a File	18
4.4.3	Execution of Python Code	18
4.4.4	Prediction Of Accuracy	18
5	IMPLEMENTATION AND TESTING	19
5.1	Input and Output	19
5.1.1	Input Design	19
5.1.2	Output Design	20
5.2	Types of Testing	21
5.2.1	Unit testing	21
5.2.2	Integration Testing	22
6	RESULTS AND DISCUSSIONS	24
6.1	Efficiency of the Proposed System	24
6.2	Comparison of Existing and Proposed System	24
6.3	Sample Code	25
7	CONCLUSION AND FUTURE ENHANCEMENTS	26
7.1	Conclusion	26
7.2	Future Enhancements	26
8	PLAGIARISM REPORT	27

9	SOURCE CODE & POSTER PRESENTATION	28
9.1	Source Code	28
9.2	Poster Presentation	30
	References	31

Chapter 1

INTRODUCTION

1.1 Introduction

Chronic kidney disease (CKD) is a significant public health problem worldwide, especially in low and medium-income countries. CKD means that the kidney does not work and cannot correctly filter the blood. About 10% of the population worldwide suffer from (CKD), and millions of people die each year because they couldn't get affordable treatment, with the number increasing in the elderly. Chronic kidney Disease (CKD) means that kidneys are damaged and not filtering your blood the way it should. The primary role of kidneys is to filter extra water and waste from your blood to produce urine and if the person has suffered from CKD, it means that wastes are collected in the body. The chronic kidney disease can damage gradually over a long period. It is flatterer a common disease worldwide Due to CKD may have some health troubles. There are many causes for CKD like diabetes, high blood pressure, heart disease. Along with these critical diseases, CKD also depends on age and gender.

If your kidney is not working, then you may notice one or more symptoms like abdominal pain, back pain, diarrhea, fever, nosebleeds, rash, vomiting. There are two main diseases of CKD diabetes and high blood pressure. So that controlling of these diseases is the prevention of CKD. Usually, CKD does not give any signal till kidney is damaged badly. CKD is being increased rapidly as per the studies hospitalization cases increase 6.23% per year but the global mortality rate remains fixed. There are few diagnostic tests to check the condition of CKD, estimated glomerular filtration rate(eGFR), urine test and blood pressure. The eGFR value shows that how your kidney cleaning the blood. If your eGFR value is greater than 90, that means the kidney is normal. The value is less than 60, that means you have CKD. The Urine test is for kidney functionality if the urine contains blood and protein, that means your kidney is not working properly. if the Blood pressure range shows that how your heart is pumping blood. If the eGFR value reaches less than 15, that

means the patient has end-stage kidney disease. At this point, there are only available treatments dialysis and kidney transplant. Patient's life after dialysis depends on such factors as age, gender, frequency and duration of dialysis, physical movement of the body and mental health. If dialysis is not possible, the doctor has only one solution, i.e., kidney transplantation. However, it is extremely expensive. In 2010, a study was conducted by the International Society of Nephrology (ISN) on global burden disease, they reported that CKD has been raised an important cause of mortality worldwide with the number of deaths increasing by 82.3% in the last two decades. Also, the number of patients reaching End-Stage Renal Disease (ESRD) increasing, which requires kidney transplantation or dialysis to save the patients' lives. The worst possible outcome of chronic kidney disease and symptoms causing any reduced kidney functioning lead to kidney failure. When the symptoms become severe and uncontrollable they can be treated through dialysis and transplantation. Early detection and treatment of CKD can slow or stop the progression of the kidney disease. But the CKD, in its early stages, has no symptoms. Proper diagnosis and testing may be the only way to find out whether the patient has been affected by kidney disease. Early detection of CKD in its initial stages can help the patient get effective treatment and then prohibit the progression to ESRD. For this critical situation we are going with Machine learning algorithms such as, KNN, Decision tree, Random forest for early prediction of chronic kidney disease of patient.

1.2 Aim of the project

The main aim is to identify whether a particular patient is affected by CKD (or) not and it has to be accurate and precise. So, for that, we are going to purpose a correlation of four pre-existing Machine Learning Algorithms to find the best among all. For this purpose, we gathered a CKD data set from the UCI machine learning repository and we examined the correlation between the development of the CKD and predictors using a predictive approach to the analysis. This will help us to reduce the number of required parameters to predict the CKD disease occurrence as well as eliminate the missing, redundant, and noisy data. And we have to use certain features to measure its accuracy and predictions.

1.3 Project Domain

Machine Learning is a sub-branch of artificial intelligence that is mostly used for predictive analysis of data, mainly in ML, To predict the outcome or the class label, and in deep learning, it is the exact opposite. Most of the predictive models are built on the principle of Machine Learning.

1.4 Scope of the Project

Chronic kidney disease (CKD) prediction using machine learning algorithms such as Random Forest, Decision Tree, and K-Nearest Neighbors (KNN) can have significant scope in healthcare. The early detection and prediction of CKD can help healthcare professionals intervene early and potentially prevent the progression of the disease. These models can assist in identifying high-risk individuals and help healthcare professionals make informed decisions about interventions and treatment plans. Random Forest and Decision Tree algorithms are suitable for identifying the most important risk factors associated with CKD, while KNN can be used to identify patients with similar CKD risk profiles. The scope of CKD prediction using these algorithms includes improving the accuracy and efficiency of diagnosis, developing personalized treatment plans, and identifying high-risk patients who require more intensive management. Additionally, these can be used to assess the effectiveness of treatment plans and predict patient outcomes. The ultimate role is to Predict the kidney disease and whether the patient is suffering from the kidney disease by using Random Forest, Decision Tree, and K-Nearest Neighbors (KNN). To minimize the risk caused by the diseases because “prevention is better than cure”.

Chapter 2

LITERATURE REVIEW

[1]. Abhishek et al.(2021) proposed an efficient neural network training model for kidney stone diagnosis. The authors highlight the limitations of the traditional diagnostic methods and the need for artificial intelligence techniques in improving diagnosis accuracy. It discusses existing research on kidney stone diagnosis using neural networks, comparing different approaches such as support vector machines and decision trees.

[2]. A. J. Aljaaf et al.(2020) proposed the method to detect CKD using machine learning algorithms while considering the least number of tests or features. They approach this aim by applying four machine learning classifiers: logistic regression, SVM, random forest, the gradient boosting on a small data set of 400 records. To reduce the number of features and remove redundancy, the association between the variables has been studied. A filter feature selection method has been applied to the remaining attributes and found that there are hemoglobin, albumin, and specific gravity have the most impact to predict CKD.

[3]. C.T. Tran, et al.(2019) proposed the ability of machine learning algorithms, for the early prediction of Chronic Kidney Disease, an experimental procedure has been undertaken in this study, considering a data set collected from Apollo Hospitals India, containing 400 instances. Two class labels were used as targets in the study (i.e. patients with CKD and healthy individuals), over which four machine-learning methods were simulated.

[4]. Eknayan G et al.(2020) discussed the role of proteinuria and other markers in the diagnosis and management of chronic kidney disease. The authors represent the National Kidney Foundation (NKF) and the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK). It provides an overview of the understanding of proteinuria and other markers in chronic kidney disease. The position statement highlights the importance of these markers in the assessment and monitoring to the kidney function. It serves as a significant resource for understanding the guidelines and recommendations related to proteinuria and chronic kidney disease.

[5]. Esra Mahsereci Karabulut et al.(2021) Proposed the comparative study on the effect of feature selection on classification accuracy. The authors investigate to the impact of feature selection techniques on the performance of classification models. The literature review discusses existing research on feature selection methods and their importance in improving classification accuracy. The methodology section provides details of the comparative study, including the selection of different feature selection techniques and the evaluation of classification accuracy. The results highlight the significant influence of feature selection on classification accuracy and provide insights into the most effective techniques for feature selection in classification tasks.

[6]. Jaymin Patel et al.(2020) focused on the prediction of chronic kidney disease is very important and nowadays it is the leading cause of death. The performance of the Decision tree method was found to be 99.25% accurate compared to the naive Bayes method. Classification algorithm on chronic kidney disease data set the performance is obtained as 99.33% Specificity and 99.20% Sensitivity. They are also further working on enhancing the performance of prediction system accuracy in neural networks and deep learning algorithm.

[7]. J.Brak et al.(2022) proposed the interaction between feature selection methods and linear classification models. The authors investigate how different feature selection techniques impact the performance of linear classification models in the context of text learning. It provides an overview of the existing research on feature selection methods and their relationship with linear classification models. The methodology section presents the details of the experimental setup and evaluation of the interaction between feature selection and linear classification. The results shed light on the influence of feature selection methods on the performance of linear classification models in the specific domain of text learning.

[8]. Koushal Kumar et al.(2021) discussed the use of artificial neural networks for the diagnosis of kidney stone disease. The authors investigate the application of neural networks as a computational approach for diagnosing kidney stones. The literature review provides an overview of existing research on kidney stone disease and the role of artificial neural networks in medical diagnosis. The methodology section presents the details of the experimental setup and implementation of neural networks for kidney stone diagnosis.

[9]. S. Vijayarani et al.(2021) proposed the prediction algorithm to predict CKD at an early stage. The dataset shows input parameters to collected from the CKD

patients and the models are trained and validated for the given input parameters. The Decision tree, Random Forest, and Support Vector Machine learning models are constructed to carry out the diagnosis of CKD. The performance of the models is evaluated based on the accuracy of prediction. The results of the research showed that the Random Forest Classifier model better predicts CKD in comparison to the Decision trees and Supports Vector machines.

[10]. T Shaikhina, et al.(2020) proposed the classification algorithms to analyze and predict Chronic Kidney Disease, and compared the performance of five classifiers in the prognosis of CKD. The results of proposed method have demonstrated that RF and XGB have produced superior prediction performance in terms of classification accuracy for our considered data set. In the future, we are going to work for enhancing the performance of prediction system accuracy by ensemble different classifier algorithms.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

Chronic kidney disease (CKD) is a long-term condition that damages the kidneys and impairs their ability to filter blood effectively. There are several methods used to predict and diagnose CKD, including eGFR, proteinuria, UACR, blood pressure, family history, and diabetes. eGFR is a measure of kidney function and is calculated using a mathematical equation. Proteinuria and UACR are measures of the presence of excess protein in urine and are used to screen for kidney damage. Monitoring blood pressure and managing hypertension is also an important part of preventing and managing CKD. Family history and diabetes are also risk factors for CKD and may be used in screening and diagnosis.

3.2 Disadvantages

- The Main limitations of these methods may not give always accurately reflect the true extent of kidney damage. Such as eGFR may not be accurate in certain populations, such as older adults, people with low muscle mass, or pregnant women.
- Proteinuria and UACR may be affected by factors of dehydration or exercise, which can lead to false positive results.
- These methods may not be able to predict the progression of CKD or the risk of developing other complications, such as cardiovascular disease.

3.3 Proposed System

In this proposed system for predicting of kidney disease using decision tree and random forest algorithms, as well as the K-nearest neighbors (KNN) algorithm.

These algorithms are trained using a dataset of patient information, such as age, sex, blood pressure, and laboratory values, to predict the likelihood of kidney disease. The Decision tree and the Random forest algorithms are both supervised learning algorithms that use a tree-like structure to model decisions based on the input data. The decision tree algorithm builds a tree model to represent possible decisions and their consequences, The random forest algorithm builds multiple decision trees and combines them to improve accuracy and reduce overfitting. The KNN algorithm is a non-parametric method that uses a distance metric to find the k-nearest neighbors to a given data point and uses their labels to predict the label of the new data point. These ML algorithms can be used to improve the accuracy and efficiency of predicting kidney disease, compared to traditional methods. And also it will give the accurate output of the patient.

3.4 Advantages

- Large datasets of patient information can be easily handled and accurately to predict the likelihood of kidney disease. This can improve accuracy compared to traditional methods.
- It processes large amounts of data quickly and efficiently, making them useful in a clinical setting where timely diagnoses are important.
- Reduce the risk of human error such as manual calculations or interpretations of test results.
- Identify patients at high risk of kidney disease at an early stage, allowing for timely interventions and management.

3.5 Feasibility Study

This study evaluates all of the project's crucial parameters, such as including the economic, technical, and social issues, these are to maximize the chances of the project's successful completion. And also it deals with accuracy and performance.

3.5.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into its search and development of the system is limited. The expenditures must be justified. Thus, the developed system is well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.5.2 Technical Feasibility

This study is processed to verify the technical feasibility, that is, the technical requirements of the proposed system. The new proposed model might have the best possibility of stealing the data from the existing or the old models. So, in that case, we have to verify whether there is a loss in data or not. Any system that has developed must not have demand on the available technical resources.

3.5.3 Social Feasibility

The main motto of social feasibility is to verify whether the proposed system is adequate for the user or not. That will also include whether the user can access everything which we have proposed. The user must not be threatened, as provided by the system.

3.6 System Specification

3.6.1 Hardware Specification

- Processor : Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz 2.60 GHz
- RAM: 8.00 GB (7.78 GB usable)
- Hard Disc Space: 8GB
- Graphics card: Nvidia GeForce GTX or AMD Radeon RX
- High-Resolution Display: 4K monitor, for viewing and analyzing complex data.

3.6.2 Software Specification

- Operating System: Windows 7,8,10 (or) Mac (or) Linux
- Coding Language: Python - version(3.5)
- Platform Requirement: Google Colab (or) Jupyter Notebook
- Framework: Kara's (or) Tensor flow (or) Kaggle
- Database Management System: DBMS, MySQL, PostgreSQL, MongoDB, and Cassandra.

3.6.3 Standards and Policies

IS 15784: Healthcare facilities - Particular requirements, Section 3.8.2, 2007 Documented established quality assurance program for imaging service and data gathering. This program will address the verification 11 and validation of imaging methods, and surveillance of all equipment with documentation of the corrective and preventive actions.

IS 1885-52-15, Data Processing, Section 15 - Programming languages, 1986. Electronic data processing is utilized for numerous exchanges and the analyses of information of both intellectual and material nature. Application of electronic data processing becomes more difficult, either because of the great variety of terms used in various fields to express the same concept, or because of the absence of or the imprecision of useful concepts. To overcome these barriers, this standard has been formulated.

IS 2003: Health Insurance Portability and Accountability Act(HIPAA) - This policy sets national standards for the protection of individually identifiable health information, including electronically protected health information (ePHI). The policy requires that covered entities, such as healthcare providers and insurers, implement safeguards to protect patient privacy, such as encryption, access controls, and audit trails.

Chapter 4

METHODOLOGY

4.1 General Architecture

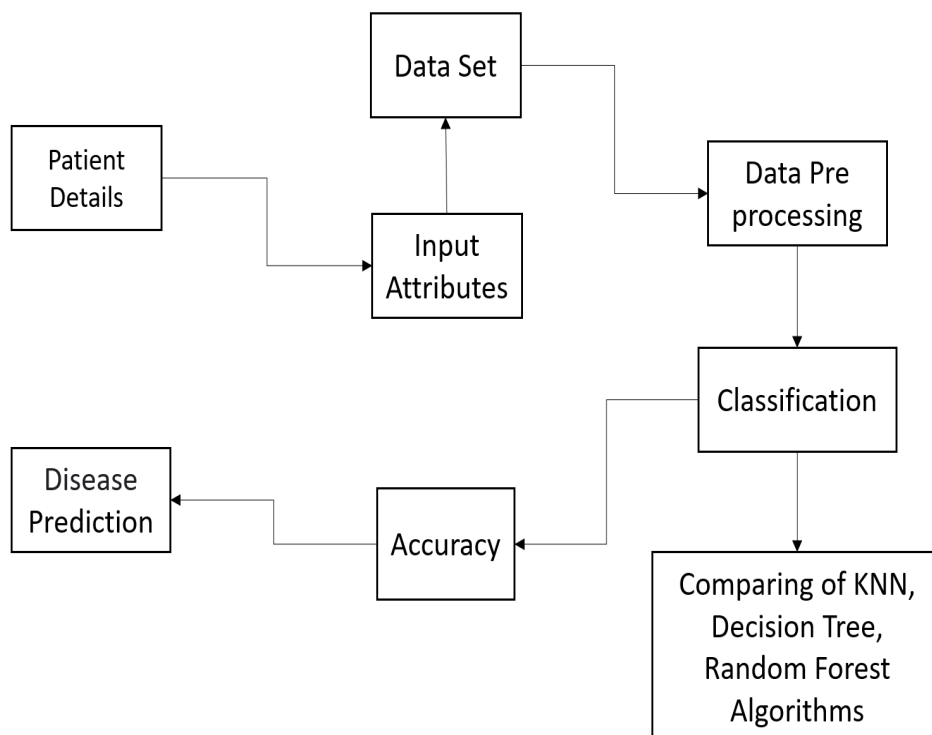


Figure 4.1: **Architecture Diagram**

In Figure 4.1 describes the architecture diagram in that the first step is to collect the data and convert it into the dataset. Now, pre-process the data that is stored in the data set based on the attributes. After that, New cleaned data is generated. Now classify this cleaned data into training data and test data. Once classification is done, apply training data to the decision tree and random forest as well as K-nearest neighbors (KNN) algorithms and get the predicted value. Test the predicted value which we got by applying the Algorithms and find the optimal output of the patient.

4.2 Design Phase

4.2.1 Data Flow Diagram

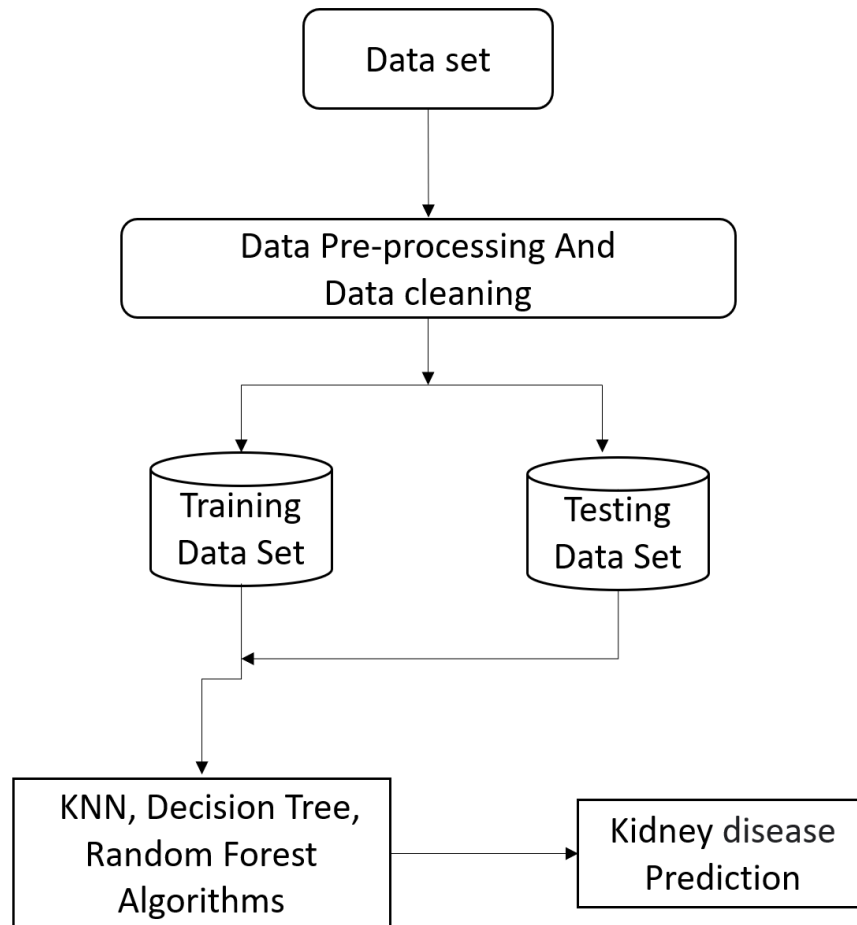


Figure 4.2: Data Flow Diagram

In Figure 4.2 describes the data flow diagram in that initially, we have to collect the data from various Patients. Import libraries and collected data set into the program. Then, the collected data undergoes preprocessing. After that, we will get created perfect data. Now apply the decision tree and random forest, K-nearest neighbors (KNN) algorithms to the perfect data. Find the optimal output of the patient.

4.2.2 Use Case Diagram

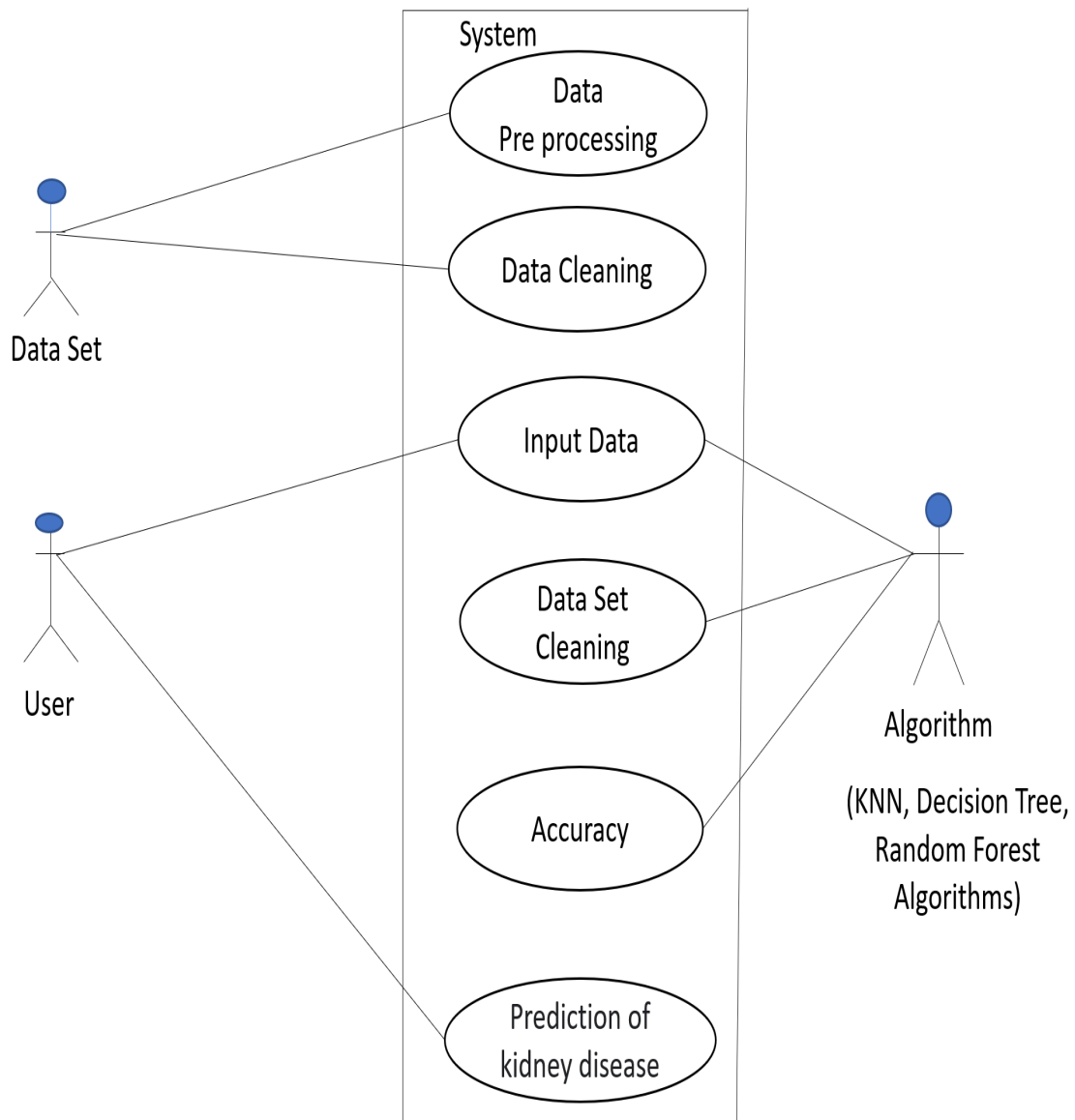


Figure 4.3: Use Case Diagram

In Figure 4.3 is a graphical depiction of a user's possible interactions with the system. Use cases are represented by ellipses. As you can see in above figure 4.3, the actor user has two possible interactions with the system in the form of giving input and viewing the output displayed on the screen. The actor server has three possible interactions i.e. taking the input given by the user, applying decision tree and random forest algorithms, as well as the K-nearest neighbors(KN) algorithms to it, and displaying the output.

4.2.3 Sequence Diagram

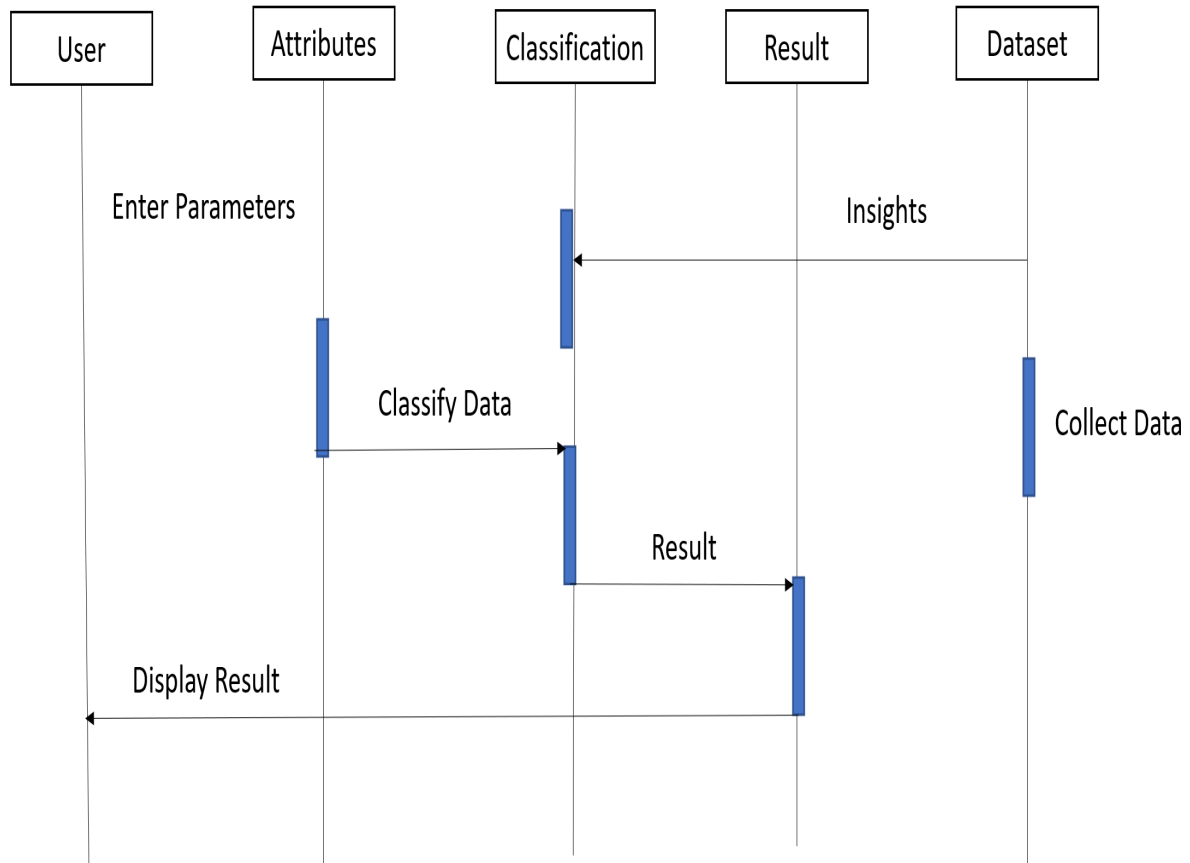


Figure 4.4: Sequence Diagram

In Figure 4.4 shows the sequence diagram. It depicts the process involved and the typical sequence of messages exchanged between the processes needed to carry out the functionality. It also describes how and in what order a group of objects works are together. The sequence diagram involves collecting relevant data from the data set and then preprocessing the data to clean, transform, and after applying the classification of machine learning algorithms such as decision tree and random forest algorithms, as well as the K-nearest neighbors (KNN) algorithms on the data based on the output it will predict the condition of the patient.

4.2.4 Collaboration Diagram

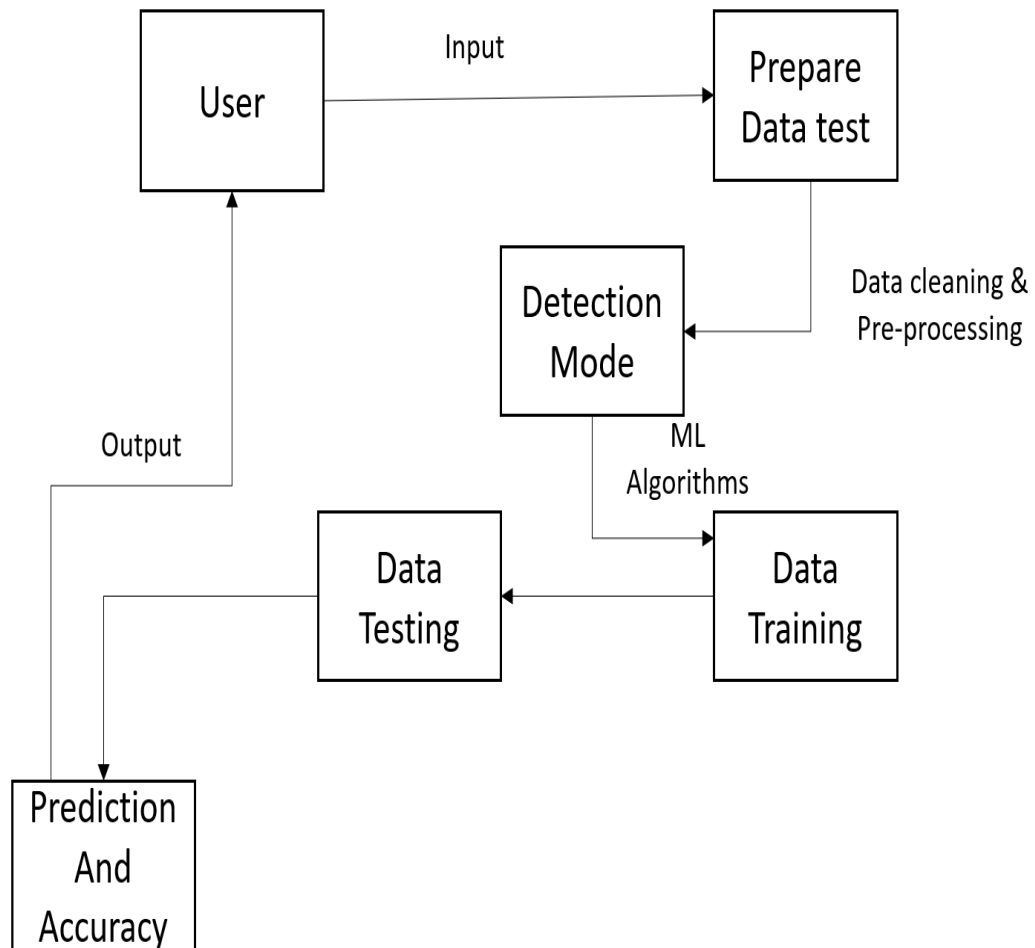


Figure 4.5: Collaboration Diagram

In Figure 4.5 shows the collaboration diagram is used to display the dynamic behavior of particular use case and define the role of each object. It is designed by first identifying the structural elements required to carry out the functionality of an interaction. As shown in Figure 4.5, we can say the collaboration diagram is a communication diagram as it illustrates the relationships and interactions among the objects.

4.2.5 Activity Diagram

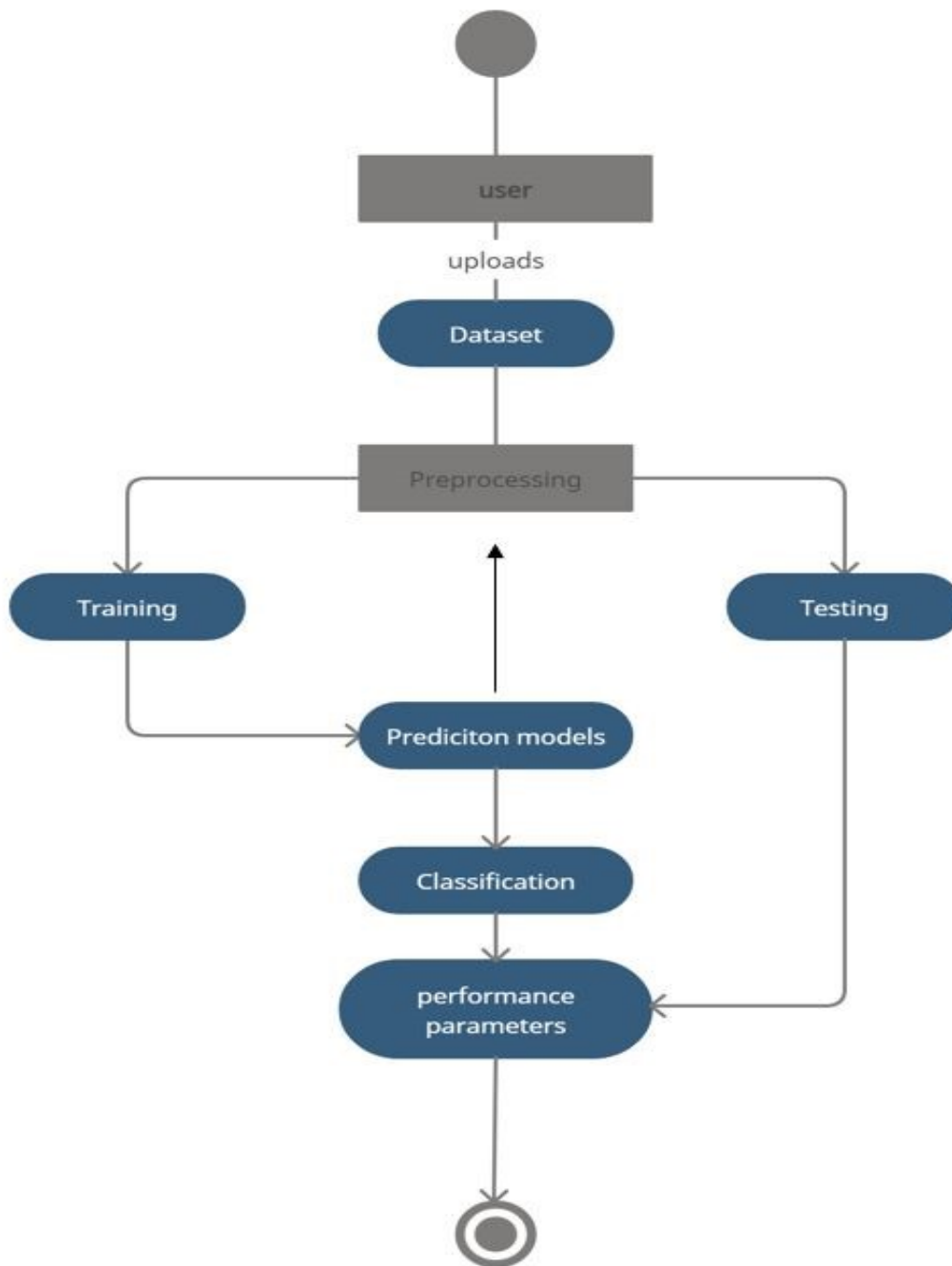


Figure 4.6: Activity Diagram

In Figure 4.6 describes the activity diagram for machine learning algorithms shows a series of steps, from collecting the data set and preprocessing the data to the selecting, training, evaluating, optimizing, and deploying the model. These types of activities are very essential for building accurate and effective machine-learning models that can be used for the prediction of kidney disease.

4.3 Module Description

4.3.1 Data Collection

Collecting data from the patient's clinical and laboratory characteristics, medical history, and family history of kidney disease. The data includes variables such as age, gender, sg, blood pressure, body mass index (BMI), and comorbidities such as diabetes and hypertension. After collecting all this data prepare a dataset, and upload it in Google Colab notebook. Before uploading the data test, should create a notebook in Google Colab and import Python Libraries numpy, pandas.

4.3.2 Data pre-processing

The pre-processing is to clean, transform, and prepare the dataset for further analysis. In the prediction of chronic kidney disease, the dataset may contain missing values, outliers, or irrelevant features that need to be addressed. Additionally, the data may need to be normalized or standardized to ensure that all features are on the same scale. Overall, after the pre-processing the dataset clean data will be obtained. It is essential to ensure accurate and reliable predictions.

4.3.3 Training and Testing the Data

The dataset is trained to recognize patterns and it makes the accurate predictions related to disease progression, treatment response, and patient outcomes. The machine learning algorithm such as decision tree and random forest algorithms, and K-nearest neighbors (KNN) also plays an important role in the performance of the model. The outcome of training dataset that can generalize well to new, unseen data and makes for accurate predictions. After test the data set by applying the machine learning algorithms such as, decision tree and random forest, and K-nearest neighbors (KNN), after applying the algorithms it displays the accuracy of each and every model of the dataset.

4.3.4 Prediction of CKD

After applying the machine learning algorithms such as, decision tree and random forest, and K-nearest neighbors (KNN) on the train and test data the accuracy will be

obtained based on the accuracy the model will predict the patients who are effected by the chronic kidney disease.

4.4 Execution And Implementation Of The Project

4.4.1 Installation Of Google Colab Notebook

Google Colab Notebook is a cloud-based platform that allows users to write, run, and share Python code. Google Colab Notebook provides access to various built-in libraries for data analysis, machine learning, and deep learning.

4.4.2 Create a File

Create a file in Google Colab Notebook Once the notebook is open, Now open the Python file and run each cell until the process is complete.

4.4.3 Execution of Python Code

Execute the python code in Google Colab Notebook involves creating a new notebook and importing the necessary Python packages for data analysis and machine learning. The next step is to load and preprocess the chronic kidney disease dataset by cleaning the data, scaling features, and splitting the data into training and testing sets. Afterward, that apply the a machine learning model such as, decision tree and random forest, and K-nearest neighbors (KNN) algorithms can be built using sci-kit-learn or other libraries and trained on the training set. The performance of the model is evaluated on the testing set using appropriate metrics, So after applying the algorithms accuracy will be obtained, and that should be displayed on using plots and charts.

4.4.4 Prediction Of Accuracy

Finally, it will give accuracy of all algorithms and it will produce a graph for that GUI is implemented and by filling in all the details of the patient. Based on the accuracy it will predict the patients who are effected by the chronic kidney disease.

Chapter 5

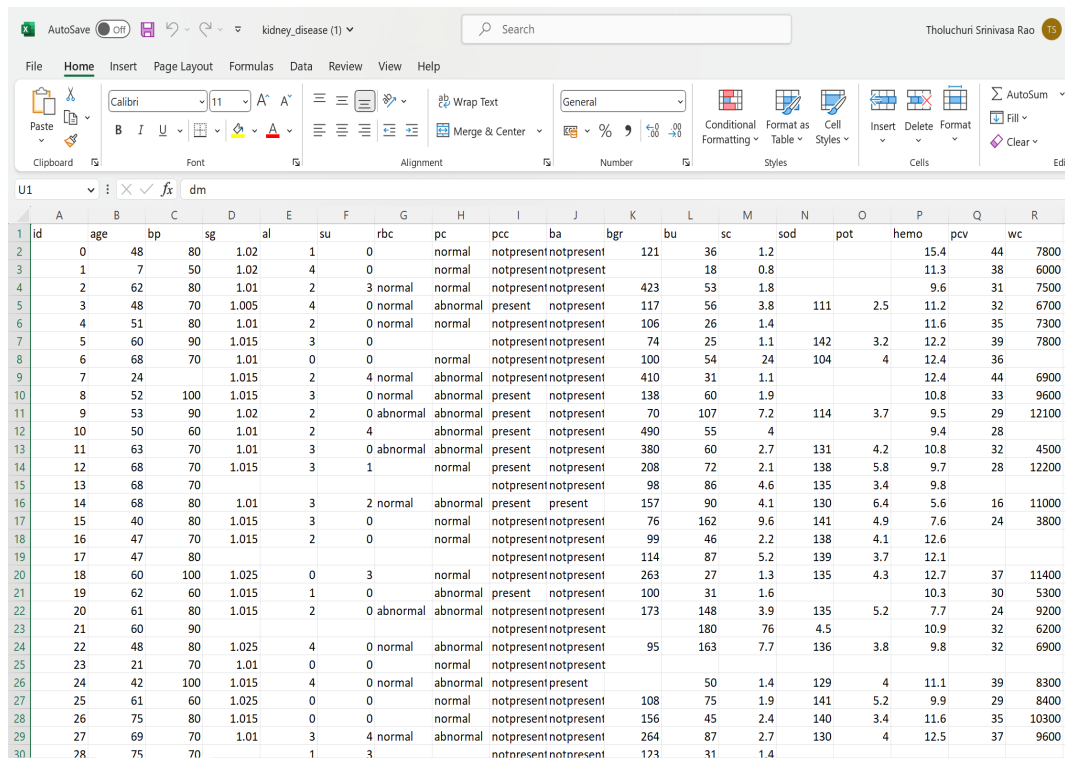
IMPLEMENTATION AND TESTING

5.1 Input and Output

Input will have to be designed effectively to minimize the error occurring. The output of the computer is essential to create an efficient method of communication between the project leader and his team members, in other words, the administrator and his clients.

5.1.1 Input Design

The input will be given in terms of file format and the Excel sheet will be provided with features in it. It can also increase the instances along with it. Currently, we are having 400 instances and a predictive model will work just fine with more instances.



id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	r
0	48	80	1.02	1	0	normal	normal	notpresent	notpresent	121	36	1.2			15.4	44	7800	
1	7	50	1.02	4	0	normal	normal	notpresent	notpresent	18	0.8				11.3	38	6000	
2	62	80	1.01	2	3	normal	normal	notpresent	notpresent	423	53	1.8			9.6	31	7500	
3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	3.8	111	2.5	11.2	32	6700	
4	51	80	1.01	2	0	normal	normal	notpresent	notpresent	106	26	1.4			11.6	35	7300	
5	60	90	1.015	3	0			notpresent	notpresent	74	25	1.1	142	3.2	12.2	39	7800	
6	68	70	1.01	0	0	normal	normal	notpresent	notpresent	100	54	24	104	4	12.4	36		
7	24		1.015	2	4	normal	abnormal	notpresent	notpresent	410	31	1.1			12.4	44	6900	
8	52	100	1.015	3	0	normal	abnormal	present	notpresent	138	60	1.9			10.8	33	9600	
9	53	90	1.02	2	0	abnormal	abnormal	present	notpresent	70	107	7.2	114	3.7	9.5	29	12100	
10	50	60	1.01	2	4	abnormal	abnormal	present	notpresent	490	55	4			9.4	28		
11	63	70	1.01	3	0	abnormal	abnormal	present	notpresent	380	60	2.7	131	4.2	10.8	32	4500	
12	68	70	1.015	3	1	normal	normal	present	notpresent	208	72	2.1	138	5.8	9.7	28	12200	
13	68	70						notpresent	notpresent	98	86	4.6	135	3.4	9.8			
14	68	80	1.01	3	2	normal	abnormal	present	present	157	90	4.1	130	6.4	5.6	16	11000	
15	40	80	1.015	3	0	normal	normal	notpresent	notpresent	76	162	9.6	141	4.9	7.6	24	3800	
16	47	70	1.015	2	0	normal	normal	notpresent	notpresent	99	46	2.2	138	4.1	12.6			
17	47	80						notpresent	notpresent	114	87	5.2	139	3.7	12.1			
18	60	100	1.025	0	3	normal	normal	notpresent	notpresent	263	27	1.3	135	4.3	12.7	37	11400	
19	62	60	1.015	1	0	abnormal	abnormal	present	notpresent	100	31	1.6			10.3	30	5300	
20	61	80	1.015	2	0	abnormal	abnormal	notpresent	notpresent	173	148	3.9	135	5.2	7.7	24	9200	
21	60	90						notpresent	notpresent		180	76	4.5		10.9	32	6200	
22	48	80	1.025	4	0	normal	abnormal	notpresent	notpresent	95	163	7.7	136	3.8	9.8	32	6900	
23	21	70	1.01	0	0		normal	notpresent	notpresent									
24	42	100	1.015	4	0	normal	abnormal	notpresent	present		50	1.4	129	4	11.1	39	8300	
25	61	60	1.025	0	0	normal	normal	notpresent	notpresent	108	75	1.9	141	5.2	9.9	29	8400	
26	75	80	1.015	0	0	normal	normal	notpresent	notpresent	156	45	2.4	140	3.4	11.6	35	10300	
27	69	70	1.01	3	4	normal	abnormal	notpresent	notpresent	264	87	2.7	130	4	12.5	37	9600	
28	75	70		1	3			notpresent	notpresent	173	31	1.4						

Figure 5.1: Input Design

5.1.2 Output Design

The output of this predictive is represented numerically ranging between 0 to 1 because we are using logistic regression here and this initiative gives the output in the range 0 and 1. And this is common for every model, the output will be in the range of 0 and 1.

```
[ ] 'model' : ['KNN','Decision Tree Classifier','RandomForestClassifier'],  
    'score' : [knn_acc,dtc_acc,rd_clf_acc]}  
)  
models.sort_values(by = 'score',ascending = False)
```

	model	score
0	KNN	0.982759
2	RandomForestClassifier	0.982759
1	Decision Tree Classifier	0.965517

 y_test

```
[ ] 339  1  
    227  0  
    344  1  
    244  0  
    44   0  
    167  0  
    254  1  
    18   0  
    351  1  
    51   0  
    246  0  
    29   0  
    352  1  
    257  1  
    366  1  
    56   0  
    354  1  
    96   0  
    112  0  
    94   0
```

Figure 5.2: Output Design

5.2 Types of Testing

5.2.1 Unit testing

The unit testing will be carried out in stages, by testing the code, starting with the smallest and lowest level modules and progressing one by one. The code is run for each cell, allowing us to obtain correct code free of errors, as errors can be erased at the cell level.

```
1
2 #import libraries
3 import glob
4 from keras.models import Sequential, load_model
5 import numpy as np
6 import pandas as pd
7 from keras.layers import Dense
8 from sklearn.model_selection import train_test_split
9 from sklearn.preprocessing import LabelEncoder, MinMaxScaler
10 import matplotlib.pyplot as plt
11 import keras as k
12 from google.colab import files
13 uploaded = files.upload()
14 df = pd.read_csv('chronic kidney.csv')
15 df.head(6)
```

```
[ ] from google.colab import files
    uploaded = files.upload()
    df = pd.read_csv('chronic kidney.csv')
    df.head(10)
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving chronic kidney.csv to chronic kidney (2).csv

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	no	no	good	no	no	ckd
5	5	60.0	90.0	1.015	3.0	0.0	NaN	NaN	notpresent	notpresent	...	39	7800	4.4	yes	yes	no	good	yes	no	ckd
6	6	68.0	70.0	1.010	0.0	0.0	NaN	normal	notpresent	notpresent	...	36	NaN	NaN	no	no	no	good	no	no	ckd

✓ 4s completed at 15:03

Figure 5.3: Unit Testing Test Result

5.2.2 Integration Testing

Integration testing is a software testing practice for that focuses on testing the interaction between different modules or components of a software system to ensure that they work together correctly. The goal of integration testing is to identify and resolve any issues that may arise when the components are combined. In the context of a software application designed to support the management of chronic kidney disease, integration testing would involve the testing of both interactions between the different features and components of the application.

If the application includes the features for tracking medication schedules and monitoring blood pressure, integration testing would verify that these features work together correctly and that the data from one feature is correctly integrated with the data from the other feature. Integration testing is an important part of the software development process as it helps identify and resolve any issues that may arise when different components of the system are combined. By verifying that the components work together correctly, integration testing helps ensure that the software application is reliable, accurate, and easy to use for patients and healthcare providers.

```
1
2 from sklearn.neighbors import KNeighborsClassifier
3 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
4
5 knn = KNeighborsClassifier()
6 knn.fit(x_train, y_train)
7
8 knn_acc = accuracy_score(y_test, knn.predict(x_test))
9
10 print(f"Training Accuracy of KNN is {accuracy_score(y_train, knn.predict(x_train))}")
11 print(f"Test Accuracy of KNN is {knn_acc} \n")
12
13 print(f"Confusion Matrix :- \n{confusion_matrix(y_test, knn.predict(x_test))}\n")
14 print(f"Classification Report :- \n {classification_report(y_test, knn.predict(x_test))}")
15 from sklearn.tree import DecisionTreeClassifier
16
17 dtc = DecisionTreeClassifier()
18 dtc.fit(x_train, y_train)
19
20 # accuracy score, confusion matrix, and classification report of decision tree
21
22 dtc_acc = accuracy_score(y_test, dtc.predict(x_test))
23
```

```

24 print(f"Training Accuracy of Decision Tree Classifier is {accuracy_score(y_train , dtc.predict(
    x_train))}")
25 print(f"Test Accuracy of Decision Tree Classifier is {dtc_acc} \n")
26
27 print(f"Confusion Matrix :- \n{confusion_matrix(y_test , dtc.predict(x_test))}\n")
28 print(f"Classification Report :- \n {classification_report(y_test , dtc.predict(x_test))}")

```

```

☞ Training Accuracy of KNN is 0.9912663755458515
Test Accuracy of KNN is 0.9827586206896551

Confusion Matrix :-
[[31  1]
 [ 0 26]]

Classification Report :-

```

	precision	recall	f1-score	support
0	1.00	0.97	0.98	32
1	0.96	1.00	0.98	26
accuracy			0.98	58
macro avg	0.98	0.98	0.98	58
weighted avg	0.98	0.98	0.98	58

Figure 5.4: KNN Test Result

In Figure 5.4 depicts the output of KNN regression technique showing how the input is taken and how the data is trained using regression techniques and as well as how it is tested using the performance metrics . So based on the precision and Recall, upon using confusion matrix 98% of accuracy is obtained.

```

☞ Training Accuracy of Decision Tree Classifier is 1.0
Test Accuracy of Decision Tree Classifier is 0.9655172413793104

Confusion Matrix :-
[[30  2]
 [ 0 26]]

Classification Report :-

```

	precision	recall	f1-score	support
0	1.00	0.94	0.97	32
1	0.93	1.00	0.96	26
accuracy			0.97	58
macro avg	0.96	0.97	0.97	58
weighted avg	0.97	0.97	0.97	58

Figure 5.5: Decision Tree Test Result

In Figure 5.5 depicts the output of decision tree Based on the precision and Recall, upon using confusion matrix it display the 96% accuracy is obtained .

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system for predicting CKD using machine learning algorithms has the potential to significantly improve the efficiency of predicting the disease. Traditional methods of predicting kidney disease often rely on subjective assessments and are time-consuming, leading to delays in diagnosis and treatment. By contrast, machine learning algorithms can rapidly analyze large datasets of patient information to predict the likelihood of kidney disease with high accuracy. In particular, the decision tree, random forest, and K-nearest neighbors (KNN) algorithms used in the proposed system can effectively process large amounts of patient data to make predictions based on patterns and correlations. These algorithms can also be trained using a variety of different data sources, including electronic health records and laboratory data, allowing for more comprehensive predictions. Additionally, the proposed system can be continuously refined and updated using new data, leading to even more accurate and efficient predictions over time. The accuracy of the proposed system is 98% and it is very better than traditional methods. Overall, the proposed system has the potential to significantly improve the efficiency of predicting chronic kidney disease, leading to earlier diagnosis and more effective treatment.

6.2 Comparison of Existing and Proposed System

The existing system for predicting and diagnosing chronic kidney disease (CKD) relies on traditional methods such as eGFR, proteinuria, UACR, and blood pressure monitoring, family history, and diabetes screening. While these methods have been effective in identifying patients with CKD, they are limited in their accuracy and may not be able to predict the likelihood of kidney disease in patients who do not show obvious symptoms. In contrast, the proposed system for predicting kidney dis-

ease using machine learning algorithms such as decision trees, random forest, and K-nearest neighbors algorithms offers a more accurate and efficient approach to predicting and diagnosing CKD. Every algorithm uses a dataset of patient information to create a model that can accurately predict the likelihood of kidney disease based on a patient's age, sex, blood pressure, and laboratory values. By using these ML algorithms, healthcare providers can identify patients with CKD more accurately and efficiently, leading to earlier diagnosis and better treatment of outcomes. Therefore, the proposed system has the potential to improve the accuracy and efficiency of CKD diagnosis and management, compared to the existing traditional methods.

6.3 Sample Code

```

1
2 from sklearn.neighbors import KNeighborsClassifier
3 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
4
5 knn = KNeighborsClassifier()
6 knn.fit(x_train, y_train)
7
8 knn_acc = accuracy_score(y_test, knn.predict(x_test))
9
10 print(f"Training Accuracy of KNN is {accuracy_score(y_train, knn.predict(x_train))}")
11 print(f"Test Accuracy of KNN is {knn_acc} \n")
12
13 print(f"Confusion Matrix :- \n{confusion_matrix(y_test, knn.predict(x_test))}\n")
14 print(f"Classification Report :- \n {classification_report(y_test, knn.predict(x_test))}")

```

```

❏ Training Accuracy of KNN is 0.9912663755458515
   Test Accuracy of KNN is 0.9827586206896551

Confusion Matrix :-
[[31  1]
 [ 0 26]]

Classification Report :-

```

	precision	recall	f1-score	support
0	1.00	0.97	0.98	32
1	0.96	1.00	0.98	26
accuracy			0.98	58
macro avg	0.98	0.98	0.98	58
weighted avg	0.98	0.98	0.98	58

Figure 6.1: KNN Test Result

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The proposed system for predicting kidney disease using the machine learning algorithms such as decision trees, random forest, and K-nearest neighbors can also significantly improve the accuracy and efficiency of predicting kidney disease compared to traditional methods. By using a dataset of patient information such as age, sex, blood pressure, and laboratory values, these methodologies predict the likelihood of kidney disease and provide accurate outputs for patients. This system can ultimately aid in the early detection and prevention of chronic kidney disease, improving patient outcomes and reducing the people death.

7.2 Future Enhancements


As a complementary solution, should buy a mobile application, to make available to everyone to detect whether they are CKD disease or not and as of now will have a correct data set that could be used to predict many more diseases, if the application collaborates with a highly reputed company then it can allow making advancements to the existing model. Well, there would be disadvantages because of the need to have higher graphic card compatibility to run the model and the solution for that would be, the involvement of a framework such as Kara's or Tensor-Flow at the back end. One possible enhancement is to incorporate more advanced machine learning techniques, such as deep learning algorithms, to improve the accuracy and efficiency of the predictions. Another potential enhancement is to include more features in the dataset, such as genetic information, environmental factors, and lifestyle factors, to provide a more comprehensive and personalized prediction of kidney disease.

Chapter 8

PLAGIARISM REPORT



PLAGIARISM SCAN

Date	May 04, 2023			
Exclude URL:	NO			
	Unique	92	Word	1000
	Plagiarized	8	Records	0

CONTENT CHECKED FOR PLAGIARISM:

Chronic Kidney Disease is a serious lifelong condition induced by either kidney pathology or reduced kidney functions. Early prediction and proper treatments can stop, chronic disease or slow the progression of the chronic disease to the end stage, where dialysis or kidney transplantation is the only way to save a patient's life. The ability of several machine learning methods such as Random Forest, Decision Tree, and K-Nearest Neighbors (KNN) for the early prediction of Chronic Kidney Disease.

Predictive analytics is used to examine the relationship between data parameters as well as with the target class attribute. It enables us to introduce the optimal subset of parameters to feed machine learning to build a set of predictive models. The dataset used for this study was obtained from the UCI Machine Learning Repository and consisted of 24 clinical and laboratory features of 400 patients diagnosed with CKD. The

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
1 #import libraries
2 import glob
3 from keras.models import Sequential,load_model
4 import numpy as np
5 import pandas as pd
6 from keras.layers import Dense
7 from sklearn.model_selection import train_test_split
8 from sklearn.preprocessing import LabelEncoder,MinMaxScaler
9 import matplotlib.pyplot as plt
10 import keras as k
11 from google.colab import files
12 uploaded = files.upload()
13 df = pd.read_csv('chronic_kidney.csv')
14 df.head(6)
15 columns_to_retain = ['sg','al','sc','hemo','pcv','wbcc','rbcc','htn','classification']
16 df = df.drop([col for col in df.columns if not col in columns_to_retain],axis=1)
17 df = df.dropna(axis=0)
18 for column in df.columns:
19     if df[column].dtype == np.number:
20         continue
21     df[column] = LabelEncoder().fit_transform(df[column])
22 df.head()
23 x = df.drop(['classification'],axis=1)
24 y = df['classification']
25 x_scaler = MinMaxScaler()
26 x_scaler.fit(x)
27 column_names = x.columns
28 x[column_names] = x_scaler.transform(x)
29 x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,shuffle=True)
30 model = Sequential()
31 model.add(Dense(256, input_dim= len(x.columns), kernel_initializer=k.initializers.random_normal(
    seed=13), activation='relu'))
32 model.add(Dense(1, activation='hard_sigmoid'))
33 model.compile(loss='binary_crossentropy', optimizer='adam', metrics['accuracy'])
34 history = model.fit(x_train, y_train, epochs=2000, batch_size= x_train.shape[0])
```

```

35 model.save('ckd.model')
36 plt.plot(history.history['accuracy'])
37 plt.plot(history.history['loss'])
38 plt.title('model accuracy & loss')
39 plt.ylabel('accuracy and loss')
40 plt.xlabel('epoch')
41 print('shape of traning data:', x_train.shape)
42 print('shape of test data:', x_test.shape)
43 pred = model.predict(x_test)
44 pred = [1 if y>=0.5 else 0 for y in pred]
45 pred
46 print('Original : {0}'.format(", ".join(str(x) for x in y_test)))
47 print('Predicted :{0}'.format(", ".join(str(x) for x in pred )))
48 from sklearn.neighbors import KNeighborsClassifier
49 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
50 knn = KNeighborsClassifier()
51 knn.fit(x_train, y_train)
52 knn_acc = accuracy_score(y_test, knn.predict(x_test))
53 print(f"Training Accuracy of KNN is {accuracy_score(y_train, knn.predict(x_train))}")
54 print(f"Test Accuracy of KNN is {knn_acc} \n")
55 print(f"Confusion Matrix :- \n{confusion_matrix(y_test, knn.predict(x_test))}\n")
56 print(f"Classification Report :- \n {classification_report(y_test, knn.predict(x_test))}")
57 from sklearn.tree import DecisionTreeClassifier
58 dtc = DecisionTreeClassifier()
59 dtc.fit(x_train, y_train)
60 # accuracy score, confusion matrix and classification report of decision tree
61 dtc_acc = accuracy_score(y_test, dtc.predict(x_test))
62 print(f"Training Accuracy of Decision Tree Classifier is {accuracy_score(y_train, dtc.predict(
        x_train))}")
63 print(f"Test Accuracy of Decision Tree Classifier is {dtc_acc} \n")
64 print(f"Confusion Matrix :- \n{confusion_matrix(y_test, dtc.predict(x_test))}\n")
65 print(f"Classification Report :- \n {classification_report(y_test, dtc.predict(x_test))}")
66 from sklearn.ensemble import RandomForestClassifier
67 rd_clf = RandomForestClassifier(criterion = 'entropy', max_depth = 11, max_features = 'auto',
        min_samples_leaf = 2, min_samples_split = 3, n_estimators = 130)
68 rd_clf.fit(x_train, y_train)
69 # accuracy score, confusion matrix and classification report of random forest
70 rd_clf_acc = accuracy_score(y_test, rd_clf.predict(x_test))
71 print(f"Training Accuracy of Random Forest Classifier is {accuracy_score(y_train, rd_clf.predict(
        x_train))}")
72 print(f"Test Accuracy of Random Forest Classifier is {rd_clf_acc} \n")
73 models = pd.DataFrame({
74     'model' : ['KNN', 'Decision Tree Classifier', 'RandomForestClassifier'],
75     'score' : [knn_acc, dtc_acc, rd_clf_acc]}
76 )
77 models.sort_values(by = 'score', ascending = False)
78 y_test

```

References

- [1] Abhishek, Gour Sundar Mitra Thakur, Dolly Gupta “Proposing Efficient Neural Network Training Model for Kidney Stone Diagnosis”, International Journal of Computer Science and Information Technologies, Vol. 3 (3), pp. 3900-3904, 2021.
- [2] A. J. Aljaaf et al, ”Early prediction of chronic kidney disease using machine learning supported by predictive analytic,” in IEEE Congress on Evolutionary Computation (CEC), vol. 25, pp. 57-59, 2020.
- [3] C.T. Tran, et al., Multiple Imputation and Ensemble Learning for Classification with Incomplete Data, Springer Publishing, vol. 17, pp. 401-415, 2019.
- [4] Eknayan G, Hostetter T, Bakris GL, et al: Proteinuria and other markers of chronic kidney disease: A position statement of the National Kidney Foundation (NKF) and the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK). vol. 16, pp. 617-622, 2020.
- [5] Esra Mahsereci Karabulut*, Selma Ayşe Özelb, Turgay İbrikç,c. “A comparative study on the effect of feature selection on classification accuracy” vol. 6, pp. 232-332, 2021.
- [6] Jaymin Patel, Prof.Tejal Upadhyay, Dr. Samir Patel, et al., “Kidney Disease Prediction Using Machine Learning and Data Mining Technique”, International Journal Of Computer Science Communication, Vol. 7, pp. 129 – 137, 2020.
- [7] J. Brank, M. Grobelnik, N. Milic-Frayling, and D. Mladenic, Interaction of Feature Selection Methods and Linear Classification Models, Proceedings of the ICML-02 on Text Learning, vol. 15, pp. 22-45, 2022.

- [8] Koushal Kumar, Abhishek, “Artificial Neural Networks for Diagnosis of Kidney Stones Disease”, International Journal of Information Technology and Computer Science, vol.45 pp.20-25, 2021.

- [9] S. Vijayarani et al, ”Kidney Disease Prediction Using the SVM and ANN Algorithms”,(IJCBR), vol. 6, pp. 190, 2021.

- [10] T Shaikhina, et al. ”Decision tree and Random forest models for the outcome prediction in antibody incompatible kidney transplantation.” Biomedical Signal Processing and Control vol. 16, pp. 255-355, 2020.