

**PREDICTING CHRONIC KIDNEY DISEASE USING MACHINE LEARNING
TECHNIQUES**

BY

**ABDUL LATIF
ID: 182-15-11375**

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Md. Sabab Zulfiker
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Co-Supervised By

Mohammad Jahangir Alam
Lecturer (Senior Scale)
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2023

APPROVAL

This Project titled “**Predicting Chronic Kidney Disease Using Machine Learning Techniques**”, submitted by Abdul Latif, ID No: 182-15-11375 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 23/01/2023.

BOARD OF EXAMINERS



Dr. Touhid Bhuiyan
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Dr. Md. Tarek Habib
Associate Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Tapasy Rabeya
Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner


23-01-23

Dr. Dewan Md Farid
Professor

Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

I hereby confirm that I completed this project under the supervision of **Md. Sabab Zulfiker, Lecturer (Senior Scale) in the Department of Computer Science and Engineering** at Daffodil International University. I further affirm that this entire work or any portion of this work, has not been submitted elsewhere for the purpose of receiving any kind of degree or certification.

Supervised by:



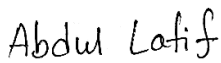
Md. Sabab Zulfiker
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Co-Supervised by:



Mohammad Jahangir Alam
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Submitted by:



Abdul Latif
ID: 182-15-11375
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

At first, we would like to express our sincere gratitude and thanks to the almighty God, whose divine favor made it possible for us to successfully complete the project for the final year.

I would like to express sincere appreciation and thanks to **Md. Sabab Zulfiker, Lecturer (Senior Scale)** in the Computer Science and Engineering Department at Daffodil International University, Dhaka. Our supervisor's knowledge and expertise in the discipline of "*Machine Learning*" gave us the confidence we needed to complete the task. His patience, intellectual direction, encouragement, frequent and active supervision, helpful advice, informative counsel, and reading many substandard drafts and editing them made this effort possible.

I would like to thank **Dr. Touhid Bhuiyan, Professor and Head, Department of CSE** for assisting us in the completion of our project. We would also wish our gratitude to the officers and other faculty members at DIU.

I want to express our thanks to all our classmates from Daffodil International University who took part in the discussion while completing the work.

Lastly, the devotion & continual assistance of our parents during our undergraduate studies must be acknowledged.

ABSTRACT

The term “Chronic Kidney Disease” (CKD) is used in medicine to describe a number of disorders that result in kidney damage or a poor Glomerular Filtration Rate (GFR). Medical advancements in recent years have allowed doctors to apply a wide range of techniques in the treatment of this illness. Recently, AI and ML have been increasingly adopted as a useful method for improving healthcare and medical research. The use of Machine Learning to detect the early symptoms of Kidney Condition is helpful as the disease may lead to a life-threatening condition. Different machine learning techniques, programs, and algorithms can be applied together to predict the steady progress of Chronic Kidney Disease. An appropriate result is produced by a machine-learning algorithm using this technique, and the algorithm with the highest performance among all others is chosen as the best one. The system could allow doctors to determine the formation of the disease as soon as they receive the dialysis report. Also, the report analysis can help to figure out which elements in the human body are the root cause of Chronic Kidney Disease. Complex and dynamic algorithms such as Naive Bayes, Random Forest, KNN, Decision Tree, AdaBoost & XGBoost etc. are needed in order to achieve optimal results in this system.

TABLE OF CONTENTS

CONTENTS	PAGE
Approval	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
List of Figures	xi
List of Tables	xi
CHAPTER 1	
INTRODUCTION	1-5
1.1 Introduction	1-2
1.2 Motivation	2-3
1.3 Rationale of the Study	3
1.4 Research Questions	3-4
1.5 Report Layout	4-5
CHAPTER 2	
BACKGROUND	6-9
2.1 Overview	6
2.2 Related Research Works	6-8
2.3 Scope of the Problem	8-9
2.5 Challenges	9

CHAPTER 3

RESEARCH METHODOLOGY 10-25

3.1 Introduction	10
3.2 Research Topic	10
3.3 Unsupervised Machine Learning Techniques	10
3.4 Supervised Machine Learning Techniques	11
3.5 Classification Techniques	11-12
3.6 Algorithm Specifications	12
3.6.1 Logistic Regression	12
3.6.2 K-Nearest Neighbors	13
3.6.3 Gaussian Naïve Bayes	13
3.6.4 Support Vector Machine	13
3.6.5 Perceptron	14
3.6.6 Decision Tree	14
3.6.7 Stochastic Scholar Gradient	15
3.6.8 Random Forest	15-16
3.6.9 XGBoost (Extreme Gradient Boosting)	16
3.6.10 AdaBoost (Adaptive Boosting)	16
3.7 Working Procedure of the Study	17
3.7.1 Data Collection	17-18
3.7.2 Dealing with Null Values	18
3.7.3 Dataset Utilization	19-20

3.7.4 Feature Importance	20-21
3.7.5 Data Pre-Processing	21
3.7.6 Data Normalization	21
3.7.7 Applying Different Algorithms for Classification	21
3.7.8 Analyze Model	22
3.7.9 Choosing the Best Algorithm	22
3.7.10 Model for Web Implementation	22-23
3.8 Implementation of a Web Interface	23
3.8.1 Execution of the Model	23
3.8.2 Values for the Input Field	23
3.8.3 Predicted Outcome	23-24
3.8.4 Architecture of the System	24
3.8.5 User Interface	24-25
3.8.6 Backend of the Web Application	25
3.8.7 Machine Learning Model for the Web Version	25

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION 26-38

4.1 Introduction	26
4.2 Experimental Results	26
4.2.1 Predicted Results & Discussion	27
4.4 Confusion Matrix	27-30
4.5 Classification Report	30-31

4.6 Analysis of the Result	31
4.6.1 Cross Validation Score	31
4.6.2 Accuracy	32
4.6.3 AUC (Area Under the Curve) Score	32-33
4.6.4 Jaccard Similarity Index	33-34
4.6.5 ROC (Receiver Operating Characteristic) Curve	34-35
4.6.6 Error & Misclassification	35-36
4.6.7 Standard Deviation	36
4.6.8 Web Implementation	37
4.6.9 User Interface of the Website	37
4.6.10 Analysis of the Website Output	37-38

CHAPTER 5

IMPACT ON SOCIETY AND SUSTAINABILITY 39-40

5.1 Introduction	39
5.2 Impact on Society	39
5.3 Ethical Aspects	39-40
5.4 Sustainability	40

CHAPTER 6

CONCLUSION AND IMPLICATION FOR FUTURE WORK 41-44

6.1 Introduction	41
6.2 Implications for Future Research	41

6.3 Recommendations	41-42
6.4 Conclusion	42
REFERENCES	43-44

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: Step by Step Process for Identifying Chronic Kidney Disease	17
Figure 3.2: Architecture of the System	24
Figure 4.1: Cross Validation Score Chart	31
Figure 4.2: Accuracy Score Chart	32
Figure 4.3: AUC (Area Under the Curve) Score Chart	33
Figure 4.4: Jaccard Index Chart	34
Figure 4.5: Receiver Operating Characteristic Curve	34
Figure 4.6: User Interface of the Website	37
Figure 4.7: Negative Web Output	38
Figure 4.8: Positive Web Output	38

LIST OF TABLES

TABLES	PAGE NO
Table 2.1: Comparative Analysis of the Study	8
Table 3.2: Description of the Used Attributes	18
Table 3.3: Statistical Features of the Dataset	19-20
Table 3.4: Feature Importance Values of different features of the dataset	20-21
Table 4.2: Specification of a Confusion Matrix	27
Table 4.3: CM (Confusion Matrix) for each Algorithm	27-28
Table 4.4: Performance Metrics of different ML Algorithms	30-31
Table 4.5: Cross Validated, Accuracy, AUC & Jaccard Score	35
Table 4.6: Errors & Misclassifications	36
Table 4.7: Standard Deviation	36

CHAPTER 1

INTRODUCTION

1.1 Introduction

Chronic Kidney Disease is an illness that affects the kidneys and lasts for a long time. This common condition is often linked to getting older. It can happen to anyone, but people of color, especially those from South Asia and the Caribbean, are more likely to have this condition. There is a chance that CKD could get worse over time and cause the kidneys to stop working completely, but this rarely happens. Many people live their lives to the fullest even though they have CKD. Scientists from all over the world have tried for a long time to figure out what causes this kidney disease.

In modern medicine, kidney disease has become a big problem. Chronic Kidney Disease, Glomerulonephritis, Kidney Stones, Urinary Tract Infections, and Polycystic Kidney Diseases are some of the most dangerous kidney diseases. Chronic Kidney Disease is a lot more likely to happen if you have high blood pressure. High blood pressure can damage the glomeruli, which are the parts of the kidneys that filter waste. Small blood vessels in the kidneys called glomeruli are in charge of filtering the blood. Over time, the higher pressure hurts the renal arteries, which makes the kidneys work less well. The kidneys will lose their ability to work at some point. In this case, the person would need treatment with dialysis. Dialysis is a way to clean the blood by getting rid of waste and excess fluid. Dialysis can help people who have failed kidneys, but it is not a cure. How well a kidney transplant works for a certain patient depends on many things. CKD is often caused directly by diabetes. It is one of many diseases that can cause blood sugar levels to rise. Consistently elevated blood sugar levels damage renal blood vessels. Because of this, the kidneys can't clean the blood as well as they would normally.

Pollution can cause kidney failure if the body is exposed to too much of it. Researchers from all over the world have done studies that show Chronic Kidney Disease kills more

people than was previously thought. Several studies have shown that men with Chronic Kidney Disease used to die more often than women with the same disease. As of 2020, the World Health Organization said that kidney disease was the cause of 10,841 deaths in Bangladesh. This is 1.51% of the total number of deaths in the country [1]. Patients with CKD who are exposed to COVID-19 are more likely to get a condition that could kill them. This makes it harder for them to get dialysis and other kinds of medical care [2]. Algorithms for machine learning can be put into many different groups. Popular methods include supervised, unsupervised, and semi-supervised machine learning techniques, as well as reinforcement learning. In this research, supervised machine learning algorithms like Decision Tree, KNN, Perceptron, Naive Bayes, Random Forest, AdaBoost, and XGBoost are used to figure out chronic renal disease. Anyone can use this method to figure out how likely they are to get a serious kidney disease. But if the pathologist and patient want to find out exactly how likely it is that they have CKD, a web-based implementation procedure may be the best way to do it. The goal of this work is to build a model using a good data set and a good algorithm to diagnose CKD at any stage. In this study, the most important Machine Learning algorithms are used to choose the best algorithm. The data is then sent to a web-based application where patients, doctors, and pathologists can access the results and use the values to predict the likelihood of kidney disease. Technology is likely to get better in the near future of the modern era. This study will help people better understand what causes kidney disease and take the right steps to treat it. It will do this by using the web to gather likely information.

1.2 Motivation

In Bangladesh, very few people know how to keep their kidneys healthy. No one knows whether or not they have kidney disease. It is thought that kidney problems kill about 1.2 million people around the world every year. Bangladesh is home to about 18 million people, and every year between 35,000 and 40,000 people with CKD reach kidney failure. In a study, researchers found that people over 40 had a much higher chance of getting kidney disease. There haven't been many studies on predicting CKD that have shown how well they work. Aside from that, we've seen that most of the modern world's attention is going toward recommendation systems. Because of this, people trust the system to give

them only the best options. A system that is meant to make suggestions must be able to make decisions on its own. You need classified information to make decisions without asking anyone else for help. All of these things made us want to do the study, in which we would use techniques for classification and prediction to help people avoid CKD.

1.3 Rationale of the Study

There are different ways to predict kidney disease. But it's not very common to use so many classification algorithms on a dataset about Chronic Kidney Illness. Even though a lot of research has been done on Chronic Kidney Disease, the results aren't that different from one study to the next. So, we use 10 different ways to classify things to see which one gives the most accurate results so that we can use that method in the future.

1.4 Research Questions

Many questions have been brought up by this study. In order to make this study more solid, a large number of questions were taken from different sources.

What prompted this research towards chronic kidney disease prediction?

Chronic kidney disease is a major health problem around the world. The condition of a person with CKD gets worse over time. It goes through several stages, with complete kidney failure being the last one. Because of this, a person will eventually die. But it's important to remember that CKD can be treated well with the right drugs, and if you follow a few rules and regulations, you can find it early on. Because of this, the study was mostly about CKD.

Why should we use machine learning? How dependable is it?

One of the most common ways to make predictions is by using machine learning. If a model has been trained on a large enough dataset, it can make accurate predictions. You

can make accurate predictions about Chronic Kidney Disease by using a medical dataset and a machine learning method. Right now, the whole world is going through a time of modernization. If you think back about ten years, when Artificial Intelligence and Machine Learning weren't as advanced as they are now, you'll remember that these main ideas were just names for some math logic. But about half of all technologies on Earth right now can't work without AI. So, this area can become even more reliable with more practice and better accuracy.

Why do people typically make use of a web interface?

A web interface is a useful tool for entering values into a machine. It gives regular users a lot of options for how to use the resource. Anyone with an internet connection and a web browser can use our app from anywhere. A web-based interface can also speed up the time it takes to get results. With this web interface, a person can find out at home how their kidneys are doing in general. Because of this, it is a useful way to predict chronic kidney disease.

Why do we need 10 different algorithms to do the same thing?

Ten algorithms were tried out to find the best one for the CKD dataset. By comparing 10 different algorithms, it was possible to find the one with the lowest error rate and the highest accuracy. If only one method was used, it would be hard to figure out which one is best because no one can predict which algorithm will work best with a given dataset.

1.5 Report Layout

The results of the study are broken up into six chapters so that the researcher can understand them better.

In Chapter 1, There is a very important introduction to the research project as a whole. This is about information about CKD, in a nutshell. This chapter talks about the study's

goal, why it was done, important research questions, expected results, how it will be run, and economic factors.

In Chapter 2, Background information for this study is talked about in detail, such as how data values are sorted, how machine learning systems work, and what other research has been done on similar topics. This chapter also talks about the problem statement's scope and what people think are the problems with comparative analysis.

In Chapter 3, The research methods, the proposed system, and the structure of the system are all talked about. This chapter goes over the details of each implemented algorithm, from their mathematical roots to their current state.

In Chapter 4, The complete analysis of the outcomes of each stage is provided. This chapter describes the best accuracy score, best algorithm, cross verified score, Jaccard score, classification report, and confusion matrix. Here are also the ROC-AUC curves for each algorithm. Error statistics, covering topics such as standard deviation, misclassification, MAE, and MSE, are discussed to conclude the chapter.

In Chapter 5, In "Ethical Aspects," it is explained how this research affects society. This is the most important part of any research work that will have an impact and be useful. In the last part of this chapter, we talk about how well this study will work in the long run.

In Chapter 6, We can see how the study will grow in the future by looking at how it will grow. In this last part of the research report, the most important findings are summed up for the reader's convenience.

CHAPTER 2

BACKGROUND

2.1 Overview

In recent years, chronic kidney disease has emerged as a major health issue. Therefore, many tragic incidents and deaths have occurred throughout the course of this disease's history. Many people's lives have been saved thanks to the extensive work done to avoid chronic kidney disease. The foundational information about this illness and its background have been covered in this chapter. A selection of the relevant research in this field has been presented in this chapter. Finally, some comparative analysis is offered to demonstrate the extent to which the work presented here is influenced by earlier work.

2.2 Related Research Works

The study by Lambert et al. attempts to predict CKD using just numerical and nominal factors. In this study, the CFS method was used to detect and classify critical characteristics as CKD or non CKD. This methodology is utilized in both the classification and the prediction phases of this strategy. CFS can be used with nominal, numerical, or nominal + numerical data to choose features. In order to pick functions, the results of the CFS are compared to three unique ranking algorithms. These techniques consist of the information gain, the ratio gain, and the relief approach. The selection accuracy based on correlation and minimal sequential optimization (CFS-SMO) approach generated results with an accuracy of 95.25% (for numerical), 98.5% (for nominal), and 98.5% (for nominal and numerical combined). The results of this experiment showed that the CFS selection was able to pull out features from the CKD dataset, and that SMO recognized CKD as a good benchmark condition. So, the CFS-SMO is seen as a good way to accurately diagnose renal disease and help doctors figure out the best way to treat it [3].

Nusinovici et al. look at how well ML algorithms can predict Cardiovascular Disease, Chronic Kidney Disease, Hypertension, and Diabetes Mellitus in a prospective observational study. Also, they look at how well simple clinical predictions work. Five more ML models were compared to the basic logistic regression model: a neural network

with one hidden layer, a support vector machine, a gradient boost, a random forest, and a KNN classifier. The CKD and DM operating curves showed that the best way to predict disease was with logistic regression (0.905 [0.88, 0.93] and 0.768 [0.73, 0.81] respectively). One that uses gradient boosting, neural networks, and logistic regression was found to be the best [4].

In their research, Ali et al. looked at the diagnosis of Chronic Kidney Disease (CKD) in developing countries in order to explore the issues with the widespread use of automated prediction methods. With the introduction of a cost-sensitive grouped feature ranking mechanism, this research provides a more useful method for group-based feature selection. This study is significant because it demonstrates for the first time that non-cutting groups with cost-sensitive ensemble ranks have a chance of reaching the desired outcomes of low cost and high accuracy. To show the usefulness of the strategy, they used seven different well-known classification algorithms and eight different comparison selection methods in the experiment. This study finds that by incorporating the cost factor in the objective space of solution formulations, the practicality of automatic CKD systems may be improved. As a result, a simple and effective method for identifying CKD has been found [5].

The proposed research algorithm, called ITLBO (Improved Teacher Learner Based Optimization), uses the distance requirement of Chebyshev to figure out the fitness function and often-used control parameters like the size of the population and generational time to find the best set of features for the first diagnosis of chronic conditions. When this function selection strategy was used on CKD data sets, it led to a 36% decrease. When compared to the old TLBO method, which led to a 25% drop in features, this is a very impressive feat. Both the TLBO and ITLBO methods use analysis of SVM, Gradient Boosting, and the Convolutional Neural Network method to confirm the best subsets of features found by the TLBO and ITLBO methods. Balakrishnan et al paper talks about a different ITLBO method than this one. [6], the results of experiments show that the three algorithms are better at classifying the created feature subset as a whole.

Segal et al. investigated commercial health insurance claims for ten million individuals collected from five hundred and fifty thousand individual profiles. The inclusion criteria

were patients older than 18 with a CKD stage 1–4 diagnosis. A total of 240 predicted indicators were compiled and arranged into six unique groupings of characteristics. Using the Word2Vec method and a technique known as feature embedding, they were able to acquire temporal characteristics on these three key data components (diagnostics, procedures, and drugs). For their investigation, they used the gradient booster approach (XGBoost) [7].

Table 2.1: Comparative Analysis of the Study

References	Sample Size	Model	Best Model with Performance
Segal et al. [1]	550,000	Word2Vec,XGB,GB	XGB(Accuracy 94.55%)
Harimoorthy et al. [2]	10,000	SVM, RF, DT	SVM(Accuracy 98.3%)
Nusinovici et al. [3]	3,000	SVM, RF, NN,GB	GB(Accuracy 88.7%)
Sambyat et al. [4]	15,000	DNN, RF, XGB	XGB (Accuracy 97.8%)
Chen et al. [5]	7,120	DL, NB, LR,DT	DT(Accuracy 98.11%)
Yashfi et al. [6]	20,000	RF, ANN	RF (Accuracy 93.75%)

2.4 Scope of the Problem

In the past few years, kidney disease has become a bigger public health problem. It keeps getting worse every day at a scary rate. There is no good model that could be used to make a prediction about it. Because of this, there are many ways to deal with this problem. One way is to look at the signs of kidney disease to see if a patient has chronic kidney problems or not. To solve this problem, you need to find the right dataset and use multiple machine learning techniques, such as a specific model that you train and test. We need to look at the correlations between the attributes of the dataset to see if there is a link between them. The quality of health care in Bangladesh will be better because of this kind of automated

diagnosis. If different hospitals have this kind of system, kidney patients can start getting different kinds of care sooner. If this method can find the disease in its early stages, it will be less likely to get worse and become chronic. This would be a big step forward from our

point of view. If patients don't let their conditions get to the point where they are chronic, doctors can quickly cure them with the right treatment. When our work leads to the creation of a system like this, there will be a big chance to help people live better lives.

2.5 Challenges

Before the kidneys can start making urine, they have to go through a complicated series of processes that can take hours to finish. This process has to work right for the body's chemicals to stay in balance. But this process is slowed down or stops in people with chronic kidney disease. If you don't know how much a person drinks, it might be hard to tell if they have kidney disease. Finding the exact criteria that need to be met in order to diagnose chronic renal illness has been the hardest part of this study. It is hard to get rid of all of the null values in the dataset. This could also be a long job that takes a lot of time. A big problem was also trying to find the right algorithm. Several algorithms were used to train the dataset, and the one that could find CKD the most accurately was chosen for further study. One of the hardest things the team had to do was make a user interface for this model so that anyone could enter data and predict CKD at any time.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Before starting any kind of research, the research method is very important. Before coming up with ways to solve a problem, you need to know what the problem is. This chapter tells you about the topic of the study's background. Also, the algorithms that were used to solve the problem were explained, and the method was shown in a picture to make it easier to understand. Also, the system's structure was shown to help people understand it better.

3.2 Research Topic

The main goal of a research project is to find a problem that can be used as the study's topic and then find a way to solve that problem. Chronic kidney disease is one of the most common diseases of old age. It's a long-term problem that might get worse over time. Kidney disease kills a lot of people every year. In the last stage of CKD, the kidneys only work about 15% of the time. Kidney disease can be treated if it is found and treated early on. That's why it's so important to study nephrology and come up with a way to predict chronic kidney disease (CKD) at different ages. Through an easy-to-use web interface, people will be able to get to their report from the comfort of their own homes.

3.3 Unsupervised Machine Learning Techniques

In unsupervised learning, models are trained on unlabeled datasets and then given the freedom to make their own predictions about the data. Since unsupervised learning only uses input data and doesn't use output data, it can't be used to solve a classification or regression problem. Unsupervised learning finds a dataset's underlying structure, groups similar data together, and cuts it down. We couldn't use unsupervised techniques for classification problems, so in our thesis we only used supervised ML techniques to predict the value of the target attribute.

3.4 Supervised Machine Learning Techniques

supervised machine learning and categorization make it easier to divide things into different groups. An ML system is one that can make decisions on its own, without help from a person, by collecting and analyzing data all the time. It is possible to make a system that keeps getting better over time by learning from past experiences, making analytical decisions, and using other methods. When it comes to machine learning techniques, there are many different ways to do things. In this work, supervised machine learning techniques are used quite often. With supervised machine learning methods, you can use data from the past that has been labeled to make predictions. The training method creates an inferred function by looking at a large dataset. This function is used to predict the output value. The system can then be taught as much as it needs to be. If the learning algorithm compares its results to the original model, it might find mistakes and change the model to fix them. In this thesis, different supervised learning methods are used to figure out how to predict chronic kidney disease.

3.5 Classification Techniques

Data classification is a way to analyze data that uses the data you give it to figure out which categories within that data are most important. It has become more popular and used than any other machine learning method. In supervised learning, these models, called classifiers, can make predictions about classes that have already been set up. Each prediction is made on its own. Classifier does not provide any intermediate values. A classifier can be made to figure out if an image shows a dog or a cat. "Dog" or "Cat" are the two possible answers. You can't use a classifier to get a value in the middle. To use a classification learning method, you must label the data. There are two different kinds of datasets that are used in classification learning. The first group is called "Training Data," and the second is called "Test Data." With the help of training data, the model is built, and then its accuracy is checked with test data. There are two parts to the classification process. During the training phase, a classifier is built using a good method and training data. The classifier is then tested in the real world. When you put together a classification method and a training dataset, you get a classifier. A classifier is really just a set of rules that can

be used in different ways in different situations. In the prediction phase, the model made in the learning phase is used to make a prediction about the class of the target attribute. This is done by applying the model to a set of unknown data. Here, we use the test data to figure out how accurate the model's predictions are.

3.6 Algorithm Specifications

Ten of the top Supervised ML approaches are applied in this study. An algorithm is often thought of as an organized sequence of instructions that instructs computer software how to transform a set of data inputs into useable data. Statistics are informational facts and pieces of knowledge that can assist a human, a robot, or a machine in performing a specific activity.

Machine learning algorithms operate on the same logic and mathematical principles. Every Machine Learning Algorithm employs its own set of mathematical transformations. Furthermore, the most popular machine learning algorithms are included in this research project, as are the relevant algorithmic processes that are part of the overall system design.

3.6.1 Logistic Regression

Logistic regression is a type of classification method that can tell you what will happen. Logistic regression can be in one of two states: 0 or 1. Since this system can tell if someone has CKD or not, the algorithm can easily figure out what will happen and plan for it. By putting together different parts, this model might be able to solve problems that are more complicated. The values on the Y-axis go from 0 to 1. This is because the sigmoid function uses these two points as the minimum and maximum, which is a great way to split a set of data into two groups. After figuring out X's sigmoid function value, the system gets a probability value between 0 and 1. The observation must fit into either of the two groups.

The formula for the sigmoid function is:

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

3.6.2 K-Nearest Neighbors

The KNN algorithm is a simple method for supervised classification. It can also be used to solve regression and classification problems. simultaneously. KNN is simple and easy to understand and use. The main idea behind KNN is the Euclidean distance. Because the dataset is split into two groups, KNN is used to classify it.

The formula for the KNN algorithm can be written as follows:

$$distance = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

3.6.3 Gaussian Naïve Bayes

Gaussian Naive Bayes is a type of Naive Bayes that works with both continuous data and Gaussian normal distributions. supervised algorithms are used to study more than one of the Naive Bayes classifiers. Based on the Bayes theorem, Bernoulli, Multinomial, and Gaussian Classifiers are three types of machine learning techniques. The Gaussian-Naive Bayes method is easy to understand and very useful. When working with continuous data, it is common to assume that the values in each class follow a Gaussian distribution. In the given equation, the full formulization process for the Gaussian Naive Bayes algorithm is shown.

$$P(X | Y = c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}}$$

3.6.4 Support Vector Machine

Since the SVM Classifier is a machine learning method, it's important to keep an eye on how well it works. It can be used to solve problems with classification and regression. Even though it can be used for many different tasks, the most common one is classifying. It was found that the people who made this technology couldn't use a graphing calculator to plot data points in n-dimensional space. This study looked at two-dimensional data charts that showed whether a person had CKD or not.

3.6.5 Perceptron

Perceptron is a model for an artificial neural network that is used in systems that divide things into two groups. A node or neuron takes information from a row of data and predicts how it should be grouped. In this way, a prediction is made using a feature vector and a prediction function, both of which are linear classification methods. In this part, we talked about how to make a threshold function that takes an input x and gives an output f that is a binary value (x). The binary number x that comes in is changed into the binary number f that comes out (x). The activation function for the perceptron algorithm can be stated as follows:

$$f(x) = \begin{cases} 0, & w \cdot x + b \leq 0 \\ 1, & \text{otherwise} \end{cases}$$

3.6.6 Decision Tree

In supervised learning, classification and regression problems can be solved by using decision trees. Most of the time, a decision tree is used to help solve classification problems. The internal nodes show the characteristics of the dataset, the branches show how decisions are made, and the leaf nodes show the results. The Decision Tree can be used to build a learning model with simple decision logic that can predict the values of the target attribute based on the training dataset. For a decision tree to be made, you need entropy. In information theory, it is a way to measure how pure or uncertain a set of observations is. It tells a decision tree how to divide the information. If we know what the value of entropy is, it is easy to figure out how much information a node has gained. The value of entropy can be given by the formula:

$$Entropy = -p \log_2 p - q \log_2 q$$

3.6.7 Stochastic Scholar Gradient

The Stochastic Gradient Descent method is a simple and effective way to build linear and regressive classifiers with convex loss functions, like the linear SVM and Logistic Regression. In the field of machine learning, SGD has been around for a while, but it has recently become very popular in the field of learning large-scale datasets. The class SGD Classifier gives you a basic way to learn SGD. It supports a number of loss functions that are used to solve classification and misclassification problems. A linear SGD Classifier like an SVM that was trained with the hinge loss can be thought of as the same as the Support Vector Machine. The algorithm for SGD classifiers includes a series of retraining instances like $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where the values of x and y are limited in some way. For the system to give accurate estimates for binary classification problems, it needs to take into account the sign of $f(x)$. Based on the model parameters, this system is needed to build a linear scoring function. The value given by the function $f(x) = w^T x + b$ is the one where the intercept b is less than R and the model parameters w are less than Rm . The normalized training error equation is used to determine model parameters.

$$E(w, b) = \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i)) + \alpha R(w)$$

Where L is the loss function that is used to judge a model and R is the normalization term that keeps the model from getting too complicated. The strength of normalization is controlled by a positive linear combination, > 0 , which is greater than 0.

3.6.8 Random Forest

Random decision making forests, also called "random forests," are a type of supervised learning method that uses multiple decision trees to classify, predict, and do other things. When used to sort things into groups, a random forest gives the category that most of its individual trees fit into. In regression problems, the average prediction from all the trees is returned. When we apply the Random Forest algorithm to a certain problem, the following equation shows the parameters we use to predict what will happen next.

$$\sigma = \sqrt{\frac{\sum_{b=1}^B (f_b(x') - \hat{f})^2}{B - 1}}$$

The value of "B" is the number of samples. Cross-validation or looking at the out-of-bag error can help find the best number of B trees: the mean error for each training sample of x' is the number of trees that don't have x' in their validation set. When it looks like several trees fit, mistakes in training and tests start to get worse.

3.6.9 XGBoost (Extreme Gradient Boosting)

The XGBoost method uses a framework called "gradient boosting," and it is an ML tool for making decisions. It starts with a set of categorized and regressed (CART) trees as its base learners. Then, it makes a set of trees that minimize a regularized target function. This makes the trees work better. The algorithm used split-wisdom discovery in separate trees, approximate division algorithms that work well with caches, and efficient out-of-core gradient boosting computing to make fast and accurate predictions.

3.6.10 AdaBoost (Adaptive Boosting)

"Adaptive classification boosting," which is short for "AdaBoost classification," is a method for learning in groups. It makes a strong classifier by putting together the results of several less good ones. In this method, a bad classifier learns from its mistakes and gets better. We need to think about an n-sample dataset. First, we give each sample a weight of 1/n.

This dataset is used to build a classifier that isn't very good. This classifier works out the error,, for the whole process of classifying. The effect of a classificatory,, is measured by the amount of error in classifying data samples. We use in the equation to change how the samples in the dataset are weighted, which gives us a new dataset.

$$\alpha = \frac{1}{2} \ln \left(\frac{1 - \epsilon}{\epsilon} \right)$$

3.7 Working Procedure of the Study

After thinking about each algorithm, the architecture of the system that is needed can be suggested. The dataset has been collected from the UCI repository. Figure 3.1 shows a system diagram that helps understand its procedure.

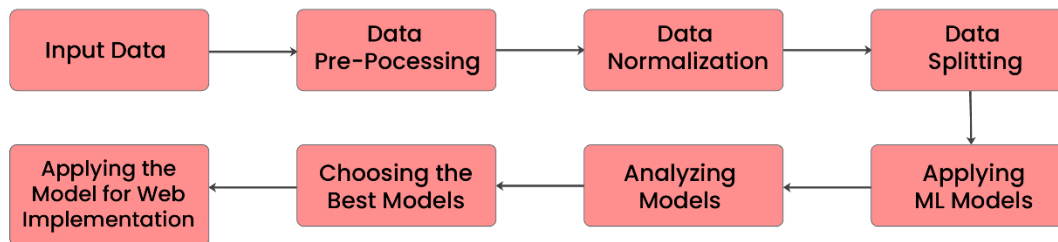


Figure 3.1: Step by Step Process for Identifying Chronic Kidney Disease

3.7.1 Data Collection

For the system to do CKD analysis, it had to have data from the real world. The University of California, Irvine provided the information used in this analysis (UCI). In the future, we want to do research in Bangladesh by getting information from places like the National Institute of Kidney Diseases and Urology, Bangladesh. The first step in this research was to put all of the information into a single comma-separated values (CSV) file to make it easier to read and understand. When we put all the data sets into one CSV file, the total number of rows was just right for using different Machine Learning Algorithms. For machine learning algorithms to make accurate predictions, they need a huge amount of data. The raw data set has 25 columns and 10321 rows, but there are some blanks in it. About 10321 data samples were used to train the model. This CKD dataset has a total of 24 variables per sample, 13 of which are nominal (categorical) and 11 of which are numeric. One of the numeric variables is a goal variable (class). There are two nominal values in each class: CKD and not CKD. The dataset has a lot of empty spots. A brief summary of the data set is provided in Table 3.2.

Table 3.2: Description of the Used Attributes

Attribute	Scale	Missing Values	Data Type
Age	age in years	2.27%	Numeric
Potassium	in mEq / L	0%	Numeric
Serum Creatinine	in mgs / dl	0%	Numeric
Blood Pressure	in mm / Hg	0%	Numeric
Sodium	in mEq / L	0.72%	Numeric
Bacteria	(Present, Not Present)	0.97%	Categorical
Pus Cell Clumps	(Present, Not Present)	0.97%	Categorical
Appetite	(Good, Poor)	0.25%	Categorical
Sugar	(0, 1, 2, 3, 4, 5)	0%	Categorical
Albumin	(0, 1, 2, 3, 4, 5)	0%	Categorical
Specific Gravity	(1.005, 1.010, 1.015, 1.020, 1.025)	0%	Categorical
Hemoglobin	in gms	0%	Numeric
White Blood Cell Count	in cells / cumm	0%	Numeric
Blood Urea	in mgs / dl	0%	Numeric
Packed Cell Volume	-	17.88%	Numeric
Blood Glucose Random	in mgs / dl	11.06%	Numeric
Red Blood Cell Count	in millions / cmm	0%	Numeric
Coronary Artery Disease	(Yes, No)	0.5%	Categorical
Diabetes Mellitus	(Yes, No)	0%	Categorical
Hypertension	(Yes, No)	0%	Categorical
Pus Cell	(Normal, Abnormal)	16.2%	Categorical
Red Blood Cells	(Normal, Abnormal)	0%	Categorical
Anemia	(Yes, No)	0.25%	Categorical
Pedal Edema	(Yes, No)	0.25%	Categorical
Class	(CKD, Not CKD)	0%	Categorical

3.7.2 Dealing with Null Values

To deal with rows or columns that are missing data, you can just delete them. If more than half of the rows in a column are null, the column can be deleted as a whole. You can also get rid of rows where one or more columns have a "null" value. In this case, we took the average of all the numbers in a column.

3.7.3 Dataset Utilization

Each category (nominal variable) had its own code, which made it much easier to manage the data in a computer system. Red blood cells that were healthy got a value of 1, and white blood cells that were not healthy got a value of 0. Pcc and ba were given a 1 or a 0 based on whether or not they were present. So, a 1 means a "yes" answer and a "0" means a "no" answer. These 1s and 0s were used to code the "yes" and "no" answers. A good appetite was given a value of 1 and a bad appetite was given a value of 0. Even though al, su, and sg were originally defined as nominal types, their values were decided by how they related to the numbers. All of the categorical variables were changed by using a method called "factorization." The samples had random numbers from 1 to 10321 written on them. This data set had a lot of empty spaces. Patients may not take the steps they need to take before a diagnosis for a number of different reasons. If you don't know what the sample diagnostic categories are, you must use the right imputation method to fill in the missing values. After categorical variables were coded, missing values in the main CKD dataset were filled in. After that, the data descriptions for each of the 25 attributes were taken out so that a better way to understand the dataset could be found. In Table 3.2, the count, minimum, maximum, mean, standard deviation, and quartiles of the data set are shown.

Table 3.3: Statistical Features of the Dataset

Column Name	Count	Max	Min	Mean	25%	50%	75%	Std
Age	10321	90	2	51.51	42	54	64	16.95
Blood Pressure	10321	1400	0	79.62	70	76	80	70.39
Pus Cell	10321	1	0	0.76	0.76	1	1	0.39
Bacteria	10321	1	0	0.06	0	0	0	0.23
Sugar	10321	5	0	0.4	0	0	0	1.03
Red Blood Cells	10321	1	0	0.88	1	1	1	0.32
Blood Urea	10321	391	1.5	57.73	27	44	64	49.63
Serum Creatinine	10321	76	0.4	3.04	0.9	1.4	3.07	5.31
Sodium	10321	1436	104	144.03	135	137.53	141	87.07
Potassium	10321	7.6	1.4	4.43	3.9	4.63	4.8	0.73
Specific Gravity	10321	1.03	1.01	1.02	1.02	1.02	1.02	0.01
Hemoglobin	10321	17.8	3.1	12.46	10.8	12.53	14.6	2.83
Pus Cell Clumps	10321	1	0	0.12	0	0	0	0.32
Packed Cell Volume	10321	54	9	38.75	34	38.75	44	8.09
White Blood Cell Count	10321	26400	2200	8403.41	7000	8406	9400	2534.28
Albumin	10321	5	0	1.02	0	1	2	1.27

Red Blood Cell Count	10321	58	2.1	4.85	4.5	4.71	5.1	2.81
Hypertension	10321	1	0	0.37	0	0	1	0.48
Diabetes Mellitus	10321	1	0	0.35	0	0	1	0.48
Coronary Artery Disease	10321	1	0	0.09	0	0	0	0.28
Appetite	10321	1	0	0.79	1	1	1	0.4
Blood Glucose Random	10321	490	22	148.4	101	127	150	74.87
Pedal Edema	10321	1	0	0.19	0	0	0	0.39
Anemia	10321	1	0	0.15	0	0	0	0.36
Class	10321	1	0	0.62	0	1	1	0.48

3.7.4 Feature Importance

The term "feature importance" refers to the method of giving an input feature a value based on how well it predicts a target attribute. It is used to talk about a set of ways to rank the importance of different features that are used as inputs to the predictive model. The feature importance score can be used to improve a predictive model and also to learn more about a dataset and the model. Table 4.4 shows how important each feature is for each algorithm.

In Table 3.4, it's clear that the highest value is for hemoglobin. In this way, Hemoglobin is a much more important feature than most other features.

Table 3.4: Feature Importance Values of different features of the dataset

Attribute	Algorithms				
	XGBoost Classifier	Decision Tree	Random Forest	Logistic Regression	AdaBoost Classifier
Age	.00524	.00529	.01393	.07557	.01
Hemoglobin	.36743	.7749	.4042	-3.32391	.36
Red Blood Cell Count	.07197	.07877	.26318	-4.10634	.27
Albumin	.00522	.00133	.00416	.03239	0
Sodium	.01034	.00334	.0105	-0.00714	.01
Blood Pressure	.00507	.00177	.00714	.04297	0
Appetite	.00483	.00033	.00182	-0.0092	0
Specific Gravity	.00804	0	.00528	-0.01694	0
White Blood Cell Count	.05247	.03789	.07531	.05221	.28
Hypertension	.38551	.07549	.12988	2.98574	.01
Sugar	.01002	.00071	.00292	.00864	0
Red Blood Cells	.00626	.00044	.00104	-0.02106	0
Diabetes Mellitus	.01044	0	.00402	.15624	.01
Potassium	.00671	.00482	.01041	.02034	0

Pus Cell	.00649	.00034	.00369	.0285	0
Pedal Edema	.00352	0	.00229	-0.06433	0
Blood Glucose (Random)	.00663	.00316	.01468	-0.04264	.02
Pus Cell Clumps	.00907	.00041	.00194	-0.00889	0
Packed Cell Volume	.00684	.00494	.01401	-0.08325	0
Anemia	.0043	.00034	.00221	.0338	0
Coronary Artery Disease	0	0	.00159	.07053	0
Bacteria	0	.00047	.00152	.03948	0
Serum Creatinine	.00713	.00079	.01145	-0.01995	.02
Blood Urea	.00649	.00445	.01284	-0.07015	.01

3.7.5 Data Pre-Processing

So that the algorithms can be used easily, the dataset needs to be changed to remove qualitative information and missing values. At first, the values of the data that were qualitative were turned into numbers. After that, we dealt with the problem of missing values. All of the blanks were filled in with the average value. The data was split into a set of variables that depended on each other (X) and variables that didn't depend on each other (Y).

3.7.6 Data Normalization

Normalization is the process of putting a set of numbers on the same scale while keeping the same ranges for each number. The next step, which made a big difference in accuracy, was to standardize the values of the independent attributes (X). As a method of standardization, Z-Score normalization was used in this case.

3.7.7 Applying Different Algorithms for Classification

Ten different algorithms were tried out and used to see which one gave the most accurate results. These are the ten algorithms: Decision Tree (DT), Support Vector Machine (SVM), Random Forests (RF), K-Nearest Neighbors (KNN), Stochastic Gradient Descent (SGD), Gaussian Naive Bayes (GNB), Adaptive Boosting (AdaBoost), Perceptron Algorithm (PA), eXtreme Gradient Boosting (XGBoost), and Logistic Regression (LR). Using these different algorithms, a number of different kinds of analytical results were found.

3.7.8 Analyze Model

After the Cross Validated Score, Jaccard Score, Accuracy Score, Confusion Matrix, Area Under the Curve (AUC), Misclassification, Mean Squared Error (MSE), and Mean Absolute Error were used to measure the data, it was put into tables (MAE). The confusion matrix is a quick way to see how well you can predict what will happen based on the facts. The Cross Validated Score, the Jaccard Score, the Accuracy Score, and the Area Under the Curve can be used to find out how often the predicted data is right (AUC). Misclassification, mean absolute error, and mean squared error are then used to evaluate an algorithm's performance.

3.7.9 Choosing the Best Algorithm

To find the best algorithm to use, all of the needed results were measured and recorded in tables. The chosen algorithm has the most accurate results and the lowest mean squared error of all the algorithms that could be used on that dataset.

The first step to making good use of the dataset is to come up with a good algorithm. As models, there are a lot of algorithms that can be used, and the best one can be chosen later. In this study, we looked at a lot of different ways to find the best one. Some of these were the Cross Validation Score, the Jaccard Score, the Accuracy Score, the Area Under the Curve (AUC), and others. Based on the findings of this study, the eXtreme Gradient Boosting (XGBoost) classifier is the best way to use the CKD dataset. It got the best marks for everything on the list above. The process can move forward once an algorithm has been chosen.

3.7.10 Model for Web Implementation

The authors picked the best way to do things and then made a web-based interface. To get the interface and the top algorithm to talk to each other, a "Pickle" was used. In this case, pickle is a Python package for serializing things. Machine learning algorithms can also be saved to a file using the pickling method.

The chosen model has been saved as a serialized model format ".sav" file. A model can't be made until there is at least one object of the chosen algorithm. The fit() method trains the model with the training dataset. Once the model has been trained with the right algorithm, it is ready to be used. The model is saved to a file in the step before this one. Then, it is loaded as what is called a "pickled model." The prediction accuracy score and the test data are then calculated based on the loaded model.

3.8 Implementation of a Web Interface

The "flask" module of Python was used to construct a web interface. You also needed to understand the fundamentals of HTML and CSS to create a user-friendly interface. The website's back end had a link to a pickle file.

3.8.1 Execution of the Model

After the model is made, it needs to be put in a folder. The dump() function of Pickle is used to save the model. This saves the object as model.pkl and turns it into a string.

Then, this model can be saved or added to Git and run on unknown test data without having to start over. The load() method of Pickle is used to deserialize a "pickled" model. The original model's predict() function can then be used to get predictions in the form of an array.

3.8.2 Values for the Input Field

Once the model has been made, flask can be used to make a simple interface for predicting CKD. Flask is just a way to make web apps with Python. Flask's framework is easier to understand than Django's, and it takes less code to make a simple web app with Flask.

3.8.3 Predicted Outcome

The user interface of the website makes it easy for anyone to put in information that can help predict Chronic Kidney Disease. If CKD was found, it would just show a positive result and suggest that the patient make an appointment with a doctor. Also, if the patient

does not have any symptoms of CKD, the website will show a negative result and reassure the patient that everything is fine.

3.8.4 Architecture of the System

System architecture is essential to simplify machine learning approach and online implementation for the whole project. In Figure 3.2, an overview of the proposed system's basic architecture is shown.

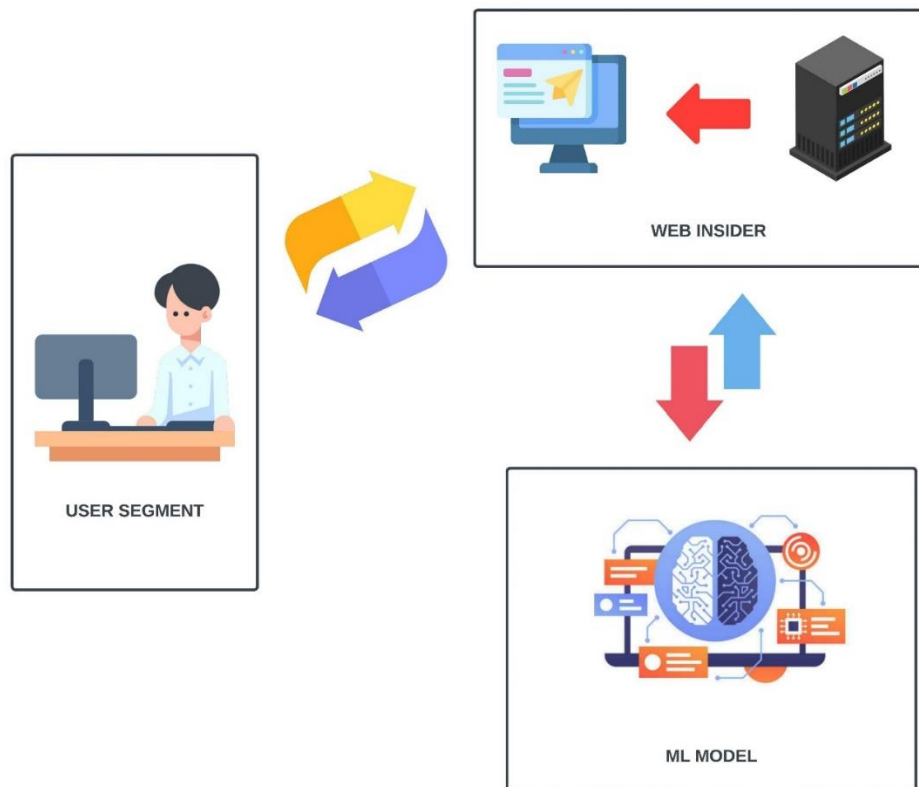


Figure 3.2: Architecture of the System

3.8.5 User Interface

In this user segment, a person uses a machine to check their own kidney health. A laptop user will put in the values for the diagnosis report's attributes. Once this is done, the system will give the user an overall picture of how healthy their kidneys are. The user segment

has a pretty simple look. All users will find it easy to figure out how to use the interface. The user only needs to type in the value and press the "Submit" button for the machine to make an accurate guess about the kidney's current state.

3.8.6 Backend of the Web Application

The "flask" Python library was used to make the web interface. You also needed to know the basics of HTML and CSS to make a user interface that worked and looked good. The back end of the website had a link to the pickle file. The website's user interface makes it easy for anyone to enter information that can be used to predict Chronic Kidney Disease. If a person has Chronic Kidney Disease, the website will show a positive result and suggest they see a doctor. Also, if a person has no signs of CKD, the website will show a negative result and let them know that everything is fine.

3.8.7 Machine Learning Model for the Web Version

We were able to figure out the best algorithm by comparing and contrasting the data in the tables. In that set of data, the chosen algorithm has the best accuracy and the fewest mistakes. After the best algorithm was found, a user interface for the web had to be made. A "pickle" was used to connect the user interface to the top algorithm. The pickling process can also be used to put ML algorithms in order and save them to a file. In this case, the model that was chosen was saved as a ".sav" file.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

The results are the most important part of any project or research. Taking into account that the end result is the sum of all the work. This chapter gives detailed results in the form of tables. This chapter shows how to understand the CKD dataset by going into detail about how the data were collected, how they were used, and how important the features were. Then, the results of the different algorithms are shown in a table called a confusion matrix. In a classification report, the accuracy, recall, and F1-Score are all added up. The accuracy results, Jaccard scores, cross-validated scores, AUC scores, and ROC curves are shown on several graphs. The information is also shown in a table to make it easy to look up. The standard deviation is also shown in a table. Last but not least, a table was given that showed the mistakes and wrong classifications.

4.2 Experimental Results

After the ML model was used successfully, each algorithm's accuracy and score were shown. To find the best way to predict Chronic Kidney Disease, these are needed. So, the Experimentation results have a section for analysis where all possible scores for each algorithmic application or technique can be looked at. Table 4.1 shows the accuracy of best three algorithms.

Table 4.1: Accuracy of Best 3 Algorithms

Algorithm	Accuracy (in percentage)
XGBoost	98.55
Decision Tree	98.11
AdaBoost	98.11

4.3 Predicted Results & Discussion

In this study, CKD got a good score and Not CKD got a bad score. Machine learning methods are judged with the help of the confusion matrix. Table 4.6 shows the confusion matrix template for several different kinds of algorithms.

4.4 Confusion Matrix

Make a confusion matrix to check the results from the point of view of how they will be used. A confusion matrix is a NN matrix with N target classes that can be used to measure how well a classification model works. Ways to use The Confusion Matrix is used to see how well machine learning models work. To do the evaluation, the actual and expected target values are compared. This way, both the algorithmic model's successes and its failures can be seen. Precision, Recall, and Accuracy are easier to figure out if you start with a binary classification. For multi-class classification, you also need to give these values a micro or a macro average. Before we talk about these, it's important to know the four basics of measuring. We use True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) to judge a Confusion Matrix (TN). Table 4.2 shows how you can make Confusion Matrices with these four values. Table 4.6 shows the confusion matrix for each algorithmic method.

Table 4.2: Specification of a Confusion Matrix

Confusion Matrix		
Predicted Class	Actual Class	
	TP (True Positive)	FP (False Positive)
	FN (False Negative)	TN (True Negative)

Table 4.3: CM (Confusion Matrix) for each Algorithm

Algorithm Used	CM (in %)			CM (value)	
		CKD	Not CKD	CKD	Not CKD
AdaBoost	CKD	62%	0%	1286	7
	Not CKD	2%	36%	32	740
SVM	CKD	59%	3%	1228	65
	Not CKD	3%	34%	70	702
Naive Bayes	CKD	51%	12%	1051	242
	Not CKD	2%	35%	47	725

Perceptron	CKD	57%	5%	1184	109
	Not CKD	6%	32%	117	655
Random Forest	CKD	61%	1%	1266	27
	Not CKD	1%	36%	28	744
Decision Tree	CKD	62%	1%	1273	20
	Not CKD	1%	37%	17	755
KNN	CKD	56%	7%	1153	140
	Not CKD	5%	33%	95	677
XGBoost	CKD	62%	0%	1287	6
	Not CKD	1%	36%	24	748
Logistic	CKD	60%	3%	1237	56
	Not CKD	3%	34%	67	705
SGD	CKD	60%	3%	1236	57
	Not CKD	3%	35%	52	720

- **True Positive (TP)**

"Positive tuples" are the ones that were correctly labeled by the classifier. It is shown by the letter TP in the acronym. About 62% of the TP (True Positive) values that came from XGBoost, AdaBoost, and Decision Tree. Then, 60% and 61% of the people chose the Logistic Regression and Random Forest methods.

- **True Negative (TN)**

Negative tuples are pairs that were meant to be positive but were put in the wrong group. You could use the letter TN to show that this is the case. The highest True Negative (TN) score is 37% for Decision Tree, followed by 36% for XGBoost, 36% for AdaBoost, and 36% for Random Forest. And 34% using "logistic regression."

- **False Positive (FP)**

Now, we're going to look at these pairs with negative labels that the classifier mistakenly thought were positive. FP can be used to show this kind of relationship. During this phase of analysis, the Random Forest, XGBoost, and Decision Tree classifiers had the lowest FP values (1%). After that, both AdaBoost and Naive Bayes gave 2% of the FP values, but Logistic Regression gave 3%.

- **False Negative (FN)**

The classifier made a mistake and gave each of these positive tuples a negative value. This idea is shown by the FN symbol. False Negative values happened 0% of the time with both AdaBoost and XGBoost. Next on the list are Random Forest and Decision Tree, both of which have 1% FN values. Logistic Regression comes next, with 3% FN values.

- **Precision**

Precision is a number that can be used to measure how accurate the given results are (i.e., what proportion of tuples that have been classified as positive are in fact positive). This shows how many positive values there are compared to the total number of positive outcomes. The given equation shows the formula for measuring with high accuracy.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall**

To figure out the recall value, we divide all of the samples that were predicted to be positive by the sum of the True Positive and False Negative values. Recall is a way to figure out how well a model can find true positives. As recall goes up, the number of identified positive samples goes up. The math formula for measuring Recall is shown in the next equation.

$$Recall (Sensitivity) = \frac{TP}{TP + FN}$$

- **F1-Measure**

The F1 score is based on the "harmonic mean" between "precision" and "recall." Since the F1 score is calculated by taking the average of the Precision and Recall scores, both Precision and Recall carry the same amount of weight when figuring out the F1 score. To get a high F1 score, a model must have good Precision and Recall. If your Precision and Recall scores are low, your F1 score will also be low. If one of a model's Precision or Recall

scores is low and the other is high, the model will get an F1 score, which is considered to be medium. Equation for the F1-Measure is given as below:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- **Accuracy**

The accuracy of a classifier is measured by how many tuples from a specific test set it correctly sorts. From the given equation, it may be easier to see how to measure accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

4.5 Classification Report

In machine learning, a "classification report" is one way to see how well a system is doing. It is used to show that a trained classification model is accurate, has a good F1 Score, and is valid. Simply put, it checks how well a machine learning model can classify things. You can tell how well a machine learning method works by looking at the metrics it shows, such as Recall, F1-measure, Precision, and Accuracy. It gives a more complete picture of how well the trained model is working. To understand classification reports made by machine learning models, you need to know about all of the measures in this study. In Table 4.4 all of the classification results for all of the used methods are shown as percentages for Precision, Recall, F1-Score, and Accuracy.

Table 4.4: Performance Metrics of different ML Algorithms

Algorithm	Class	Precision	Recall	F1-Score	Accuracy
Naive Bayes	Not CKD	0.75	0.94	0.83	86%
	CKD	0.96	0.81	0.88	
KNN	Not CKD	0.83	0.88	0.85	88.62%
	CKD	0.92	0.89	0.91	
SVM	Not CKD	0.92	0.91	0.91	93.46%
	CKD	0.95	0.95	0.95	
Perceptron	Not CKD	0.86	0.85	0.85	89.06%
	CKD	0.91	0.92	0.91	
XGBoost	Not CKD	0.99	0.97	0.98	98.55%

	CKD	0.98	1	0.99	
Random Forest	Not CKD	0.96	0.96	0.96	97.09%
	CKD	0.98	0.98	0.98	
SGD	Not CKD	0.92	0.92	0.92	93.95%
	CKD	0.95	0.95	0.95	
Logistic	Not CKD	0.93	0.91	0.92	94.04%
	CKD	0.95	0.96	0.95	
AdaBoost	Not CKD	0.99	0.96	0.97	98.11%
	CKD	0.98	0.99	0.99	
Decision Tree	Not CKD	0.97	0.98	0.97	98.11%
	CKD	0.99	0.98	0.98	

4.6 Analysis of the Result

Now that all the metrics, like Recall, F1-measure, Precision & Accuracy, etc., have been calculated, we can look at the results. We will look at how well different algorithms work and decide which ones are the best and which ones are the worst.

4.6.1 Cross Validation Score

Cross-validation is a statistical method that is used to measure how well a machine learning model works. To start Cross Validation, the data are mixed up and put into k different groups. Then, we fit k models to (k-1/k) of the data, and then we look at 1/k of the data. The final score is made by averaging the results of each evaluation, and the model that is made is used after it has been fit to the whole dataset. Figure 4.1 shows the results of the cross validation for each algorithm.

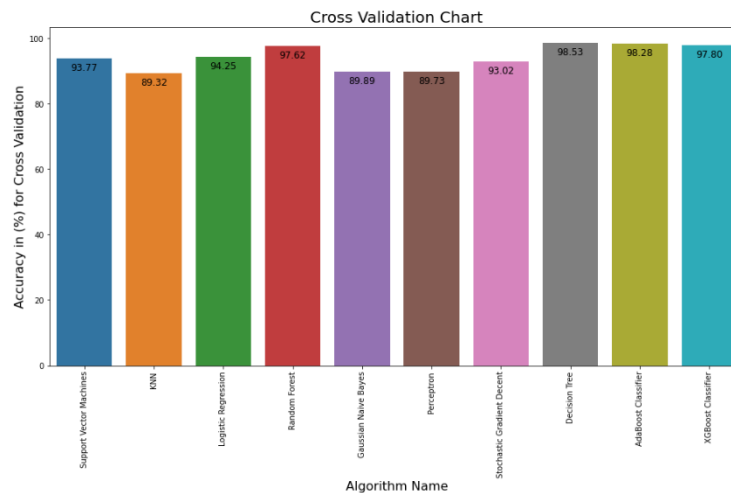


Figure 4.1: Cross Validation Score Chart

4.6.2 Accuracy

The best way to measure how well an algorithm works is by how accurate it is. How well it works depends on what it is told. The performance can be judged by how well it works, which can be done by using a probabilistic method to figure out how accurate it is. The most accurate of these methods is eXtreme Gradient Boosting, while the least accurate is Gaussian Nave Bayes. eXtreme Gradient Boosting is a powerful and scalable ML implementation of gradient boosting that can push the limits of what boosted trees methods can do with their computing power.

It was made to improve how well models work and how quickly computers can process data. Here are the accuracy charts (Figure 4.2) and percentages (Table 4.8) for all of the prediction algorithms that were used in this model.

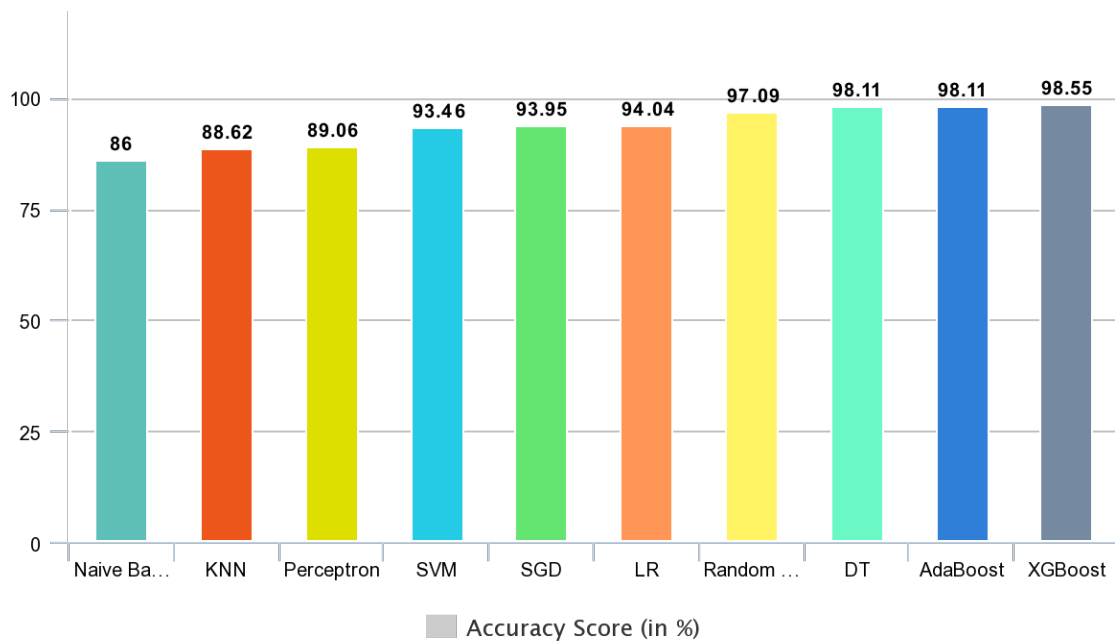


Figure 4.2: Accuracy Score Chart

4.6.3 AUC (Area Under the Curve) Score

When used with machine learning programs, this efficiency metric can be used to estimate how well a system will work at different levels of classification. You can figure out the AUC by comparing how well the model works on a percentile of randomly chosen positive cases to how well it works on a percentile of randomly chosen negative cases. This number

could be one of four things, but one is the most likely. The numbers range from 0 to 1, with 0 being the lowest possible number. Figure 4.3 shows how the AUC score turned out for each algorithm.

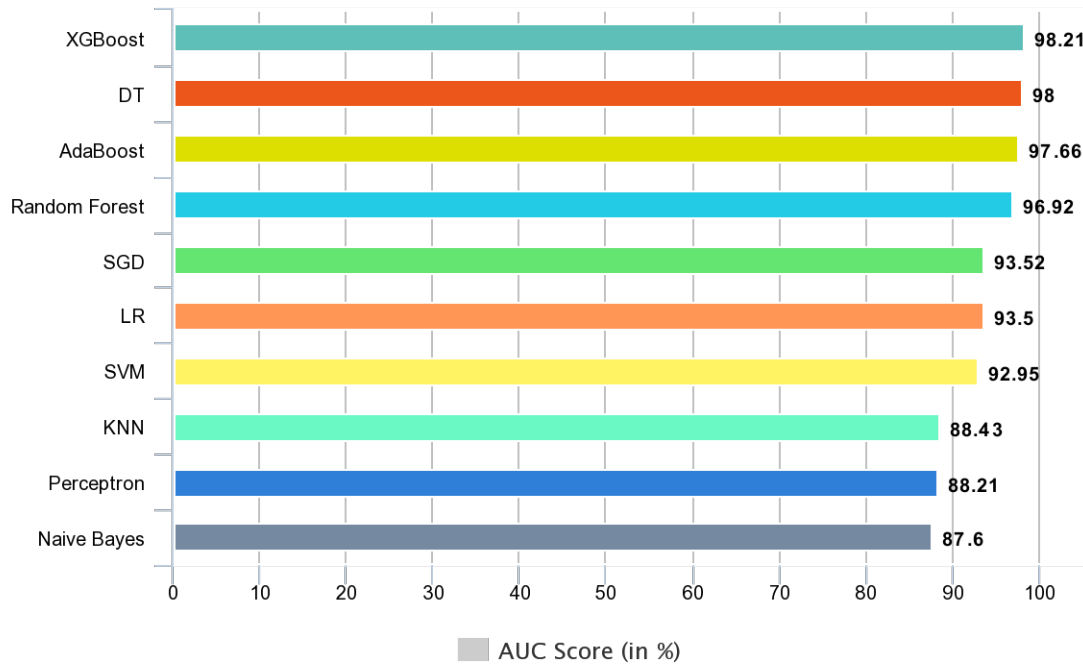


Figure 4.3: AUC (Area Under the Curve) Score Chart

4.6.4 Jaccard Similarity Index

A Jaccard score is one way to figure out how much two samples are alike or different from each other. Jaccard Similarity is a tool that is used a lot in the field of data science. With the Jaccard similarity index, which is the ratio of the size of the intersection to the size of the union, it is easy to compare how similar two finite sets are. The Jaccard Similarity index ranges from 0 to 1. The way to figure out the Jaccard Index is as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

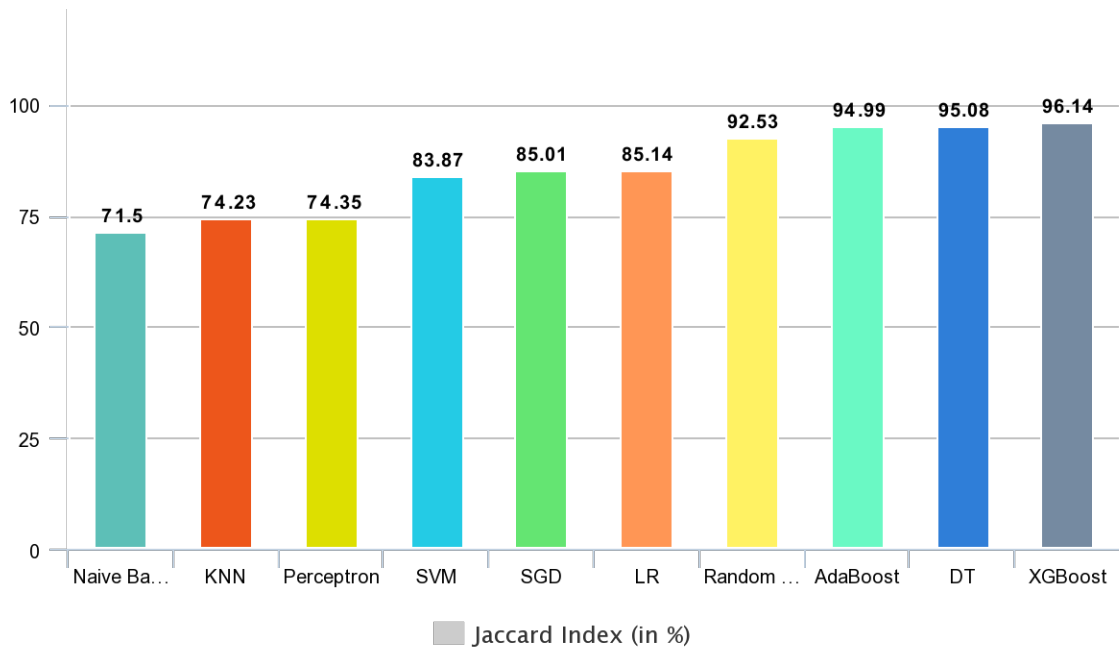


Figure 4.4: Jaccard Index Chart

4.6.5 ROC (Receiver Operating Characteristic) Curve

With ROC analysis, you can figure out how well a diagnostic test works and how accurate a statistical model is. It sorts people into those who have diseases and those who don't. The ROC curve analysis is a simple way to show how accurate a medical diagnostic test is. Figure 4.5 shows what this ROC curve looks like for each algorithm.

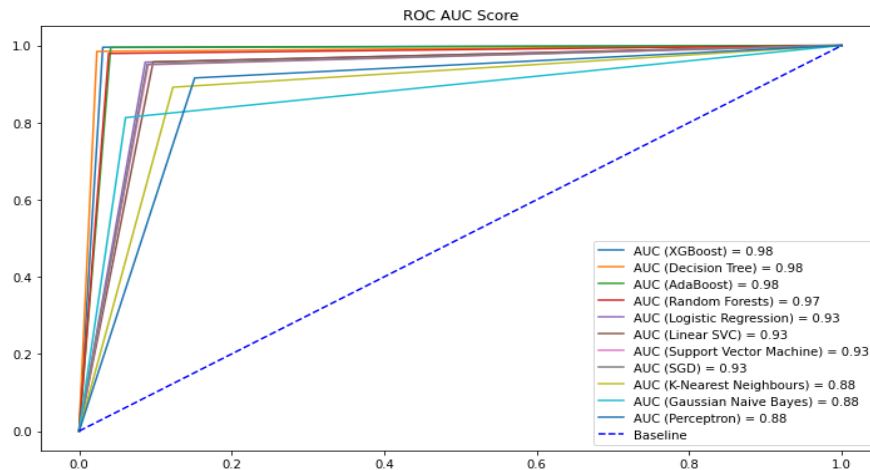


Figure 4.5: Receiver Operating Characteristic Curve

Overall, XGBoost Classifier did the best in terms of accuracy (98.55%), Jaccard Score (96.14%), Cross Validated Score (98.97%), and Area Under Curve (AUC) (98.21%). Table 4.4 gives a short summary of how accurate it is.

Table 4.5: Cross Validated, Accuracy, AUC & Jaccard Score

Algorithm	Cross Validated Score	Accuracy Score	AUC Score	Jaccard Score
KNN	87.71%	88.62%	88.43%	74.23%
Perceptron	88.65%	89.06%	88.21%	74.35%
Support Vector Machines	93.53%	93.46%	92.95%	83.87%
Stochastic Gradient Decent	93.34%	93.95%	93.52%	85.01%
Logistic Regression	94.1%	94.04%	93.5%	85.14%
Random Forest	97.61%	97.09%	96.92%	92.53%
AdaBoost Classifier	98.12%	98.11%	97.66%	94.99%
Decision Tree	98.54%	98.11%	98%	95.08%
Naive Bayes	85.45%	86%	87.6%	71.5%
XGBoost Classifier	98.97%	98.55%	98.21%	96.14%

4.6.6 Error & Misclassification

When trying to figure out how accurate an algorithm is, errors can be a problem. Mean square error and mean absolute error after misclassification are two ways to measure how accurate a machine-learning model is. Misclassification happens when the wrong kind of attribute is used. If the rate of error is the same for every possible way to divide a variable, then the variable is misclassified. Absolute error is a way to talk about how much of a measurement error there is. MAE is the average of the absolute errors in a measurement.

$$MAE = \frac{1}{n} \sum_{i=1}^n |x - x_i|$$

Mean Squared Error (MSE) shows how far away a set of points are from a regression line. The equation for the MSE can be written as,

$$MSE = \frac{1}{n} \sum_{i=1}^n |y - y_i|^2$$

Table 4.6 shows the wrong classification, the mean squared error, and the mean absolute error of the algorithms. XGBoost had a lower error rate on all levels (1.45%) than other popular methods.

Table 4.6: Errors & Misclassifications

Algorithm	MSE (%)	MAE (%)	Misclassification (%)
KNN	11.38	11.38	11.38
AdaBoost Classifier	1.89	1.89	1.89
Support Vector Machine	6.54	6.54	6.54
Naive Bayes	14	14	14
Logistic Regression	5.96	5.96	5.96
DT	1.89	1.89	1.89
Random Forest	2.91	2.91	2.91
SGD	6.05	6.05	6.05
XGBoost Classifier	1.45	1.45	1.45
Perceptron	10.94	10.94	10.94

4.6.7 Standard Deviation

You can also figure out the standard deviation from the data in this study. The standard deviation from the mean is a way to figure out how different a group of numbers is from each other. Take the square root of the difference between each piece of data to get the standard deviation. When the points are farther from the mean, the standard deviation goes up.

Table 4.7: Standard Deviation

Algorithm	S.D.
Decision Tree	0.15
AdaBoost Classifier	0.1
XGBoost Classifier	0.1
Random Forest	0.09

4.6.8 Web Implementation

The proper approach was selected and effectively applied in the development of machine learning models. To end this chapter, the System Architecture described in the last chapter will be shown by showing how the model described in the last chapter is put into action through the Web Interface.

4.6.9 User Interface of the Website

Chapter 3 says that the "Flask" method will be used to predict chronic kidney disease online. As we said in Chapter 3, we'll use the Flask framework to build a web interface. For the sole purpose of making sure that this task is done well, a website that is both very effective and fully functional has been made. This web-based user interface is shown in Figure 4.6.

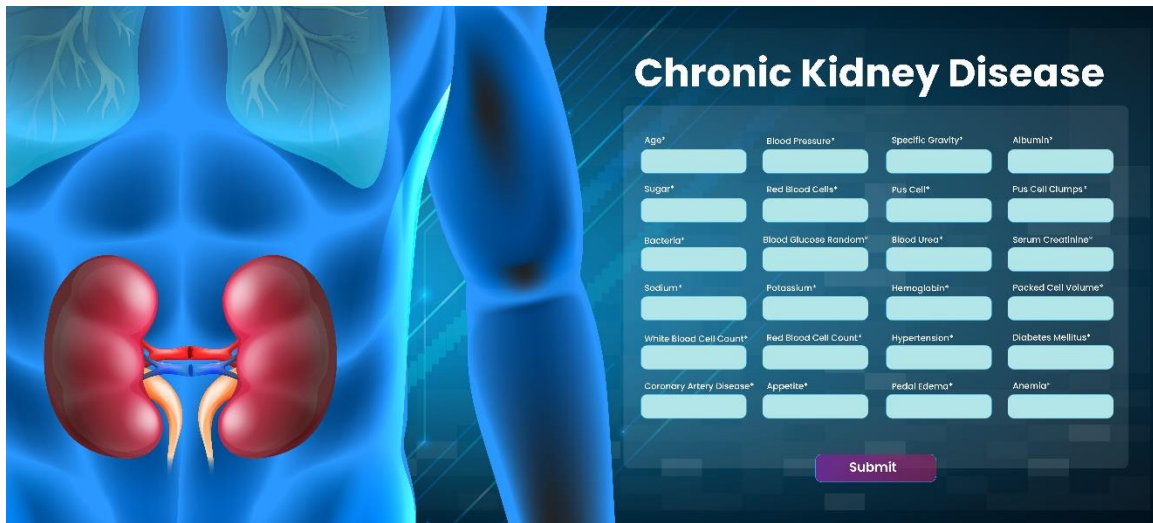


Figure 4.6: User Interface of the Website

4.6.10 Analysis of the Website Output

Based on the way we did things and the data we got, we can only divide people into two groups: those with chronic kidney disease (CKD) and those without it (Not CKD) (which stands for non-chronic kidney disease). Two separate output analyses will be needed for the whole of this section. Figure 4.7 shows what happens when all the data needed

to find Chronic Kidney Disease is entered randomly into a trained model. In Figure 4.8, a model that had already been trained was used to predict CKD based on random data from a test case. After doing all of these things, it is clear that the trained model for predicting Chronic Kidney Disease is correct.



Figure 4.7: Negative Web Output

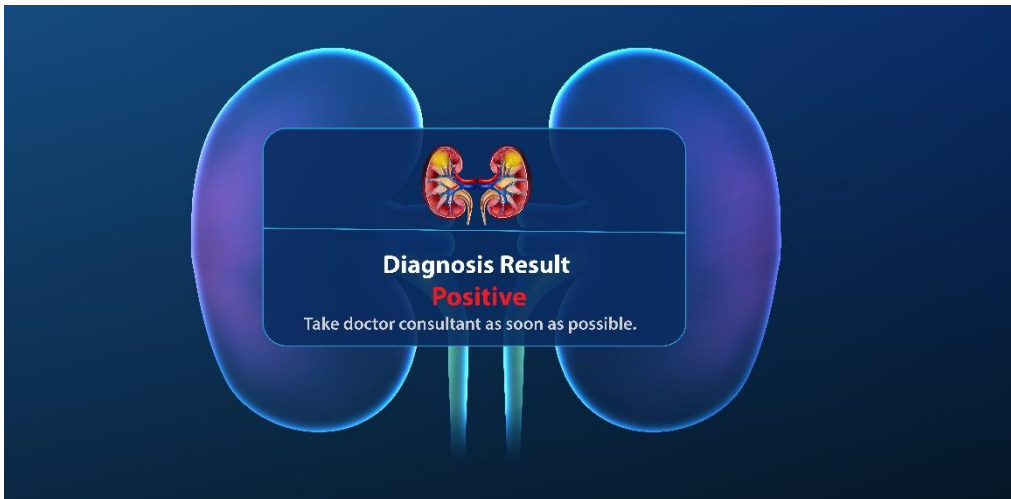


Figure 4.8: Positive Web Output

CHAPTER 5

IMPACT ON SOCIETY AND SUSTAINABILITY

5.1 Introduction

When a project is done, the effects it has on the community should be talked about and studied. In this chapter, the project's effects are broken up into three different sections. In the "Consequences on Society" section, it has been talked about how this work will be good for society. After that, we looked at the moral consequences. For this, the moral issues have been talked about in a way that helped people see how this work will help both doctors and patients. And at the end, we talked about how reliable the project was. In the last part, we talk about how this work can grow in the future to help more people.

5.2 Impact on Society

The results of this research will have a big impact on society as a whole. People's lives are so busy these days that it might be hard to find the time to go to the doctor to make sure they aren't getting a long-term illness. Through an easy-to-use interface, users can now send in the information needed to make a good prediction of chronic renal disease. When the COVID-19 virus is spreading around the world, it can be dangerous to get medical tests done in a hospital. It would be very helpful if people could look at their report at home. We plan to do a survey of patients and medical professionals in the future to see how well this study works. The results of that survey will almost certainly be good for society as a whole and for the relationship between doctors and patients.

5.3 Ethical Aspects

People could find out what was wrong with them without going to a hospital in this age of technology. People could learn about chronic kidney disease without having to leave their homes. Because of this, they would be able to make their own predictions. Since a user interface has been made, all of the data that has been collected and saved in the database will be added to the training set. This will help the model get better over time. Since the

project is based on machine learning, it looks like we can't rely on this model completely right now. A machine is incapable of foreseeing the future. It will take time for an ML model to get more accurate. When the dataset has a million records, the model will be a lot stronger, and maybe it will be possible to make good predictions. If AI and the IoT could be added to our project, people might not have to go to the hospital to get a diagnosis in the future. People will be able to figure out if they have chronic renal disease at home before going to see a doctor. If a portable device for testing blood and urine could be made, people could test for chronic kidney disease more accurately in their own homes.

5.4 Sustainability

A website can be used to figure out a person's risk of getting chronic kidney disease, which makes this study very reliable. At the moment, the website can only predict kidney disease by using algorithms that help computers learn on their own. Deep learning, AI, and the Internet of Things can be used together in the future to make this project better. So, this project can be used to do many different things in the future if it can be trusted. Predictions can also be made in the form of a smartphone app. So far, the study has only focused on predicting CKD, but this ML project and the web-based method can be used together to predict a wide range of kidney diseases and diseases in other organs as well.

CHAPTER 6

CONCLUSION AND IMPLICATION FOR FUTURE WORK

6.1 Introduction

The future scope and conclusion of our work are covered in this chapter. For example, We talk about how this project could help the company grow in the future and how a better machine could be made in the future. At the end of this chapter, there is a summary that is clean and easy to understand. At the end of this chapter, there is a list of sources for more reading.

6.2 Implications for Future Research

No matter how hard it is, the idea can be used to build a web app for any hospital that specializes in treating kidney disease. The people who worked on this project made a simple interface. In the near future, a website based on IoT could be built and made available to everyone online. Chronic Kidney Disease (CKD) can be easily predicted at home by filling out a web form with all the needed information. The person can choose whether or not to have CKD. Since this would be an IoT-based website and users would enter new information each time, the site would keep track of that information. In the end, the model will be able to get better by learning from the new information it gets. Adding algorithms from Artificial Intelligence and Neural Networks to a Deep Learning method could make it work better over time.

6.3 Recommendations

For every diagnostic test, there is a base level, and if an abnormal phase is found, this system would be able to tell what is going on. Urine Albumin, Hemoglobin, Serum Creatinine, Creatinine Clearance, etc. are the main things that can help predict Chronic Kidney Disease. Serum Creatinine is usually between 0.7 and 1.3 mg/dL for male patients and between 0.6 and 1.1 mg/dL for female patients. The normal ranges for males and women's creatinine clearance are 97-137 mL/min and 88-128 mL/min,

respectively. When the amount of protein in the urine goes up, it could be a sign of a problem with the kidneys. If someone wants to know how bad their condition is, they must check to see if their urine has protein in it. Normal range is between 0 and 8 mg/dL. If the amount of protein in your urine goes up in a way that doesn't make sense, it could mean that your kidneys aren't working right. Women should have a Hgb level between 12 and 16 g/dL, while men should have a Hgb level between 14 and 18 g/dL. Changes in the ranges may be an early sign of chronic renal disease if they happen often. So, there is still time to take precautions like changing the way you eat, getting more exercise, drinking more water, etc. When this happens, you need to talk to a doctor.

6.4 Conclusion

A person with CKD might be able to get better if their condition is found and treated quickly. Several lab tests can be used to find out if someone has chronic kidney disease (CKD). In real life, these tests could take a long time and cost a lot of money. An ML model that has been trained on the right dataset can predict any stage of chronic renal illness. For this project, the authors have made a user interface in which users can find out if they have CKD by filling out a set form.

On the UCI dataset, ten different ML algorithms were trained, and then their accuracy, AUC score, Cross Validated score, and Jaccard index were used to judge them. This is done so that the best model for the data set can be found. All of the algorithms are tested for their efficiency using three score values: accuracy, the Jaccard index, and cross validation. XGBoost does a better job than other classifiers when it comes to performance. Most of the time, going with XGBoost classifier was the best choice. After a web application for this project is made, it could be used in any hospital that treats kidney disease. Online CKD reports will be available around the clock, so patients won't have to take time out of their busy days to get them.

REFERENCES

- [1] World Health Rankings, deaths due to renal disease in Bangladesh, accessible at <<<https://www.worldlifeexpectancy.com/bangladesh-kidney-disease>>>, last visited on 09-01-2022 at 11:08 AM.
- [2] Center for Disease Prevention & Control, accessible at <<<https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/people-with-medical-conditions.html>>>, last visited on 09-01-2022 at 11:14 AM.
- [3] Lambert, J. R., Perumal, E., & Arulanthu, P., “Identification of Nominal Attributes for Intelligent Classification of Chronic Kidney Disease using Optimization Algorithm” in the 2020 ICCSP (International Conference on Communication and Signal Processing) IEEE, pp. 0119-0125, July 2020.
- [4] Nusinovici, S., Yan, M. Y. C., Tham, Y. C., Cheng, C. Y., Ting, D. S. W., Li, J., & Sabanayagam, C., “Logistic regression was as good as machine learning for predicting major chronic diseases” in the Journal of clinical epidemiology, 122, pp. 56-69, 2020.
- [5] Balakrishnan, S., “Feature Selection Using Improved Teaching Learning Based Algorithm on Chronic Kidney Disease Dataset” in Procedia CS, 171, pp. 1660-1669, 2020.
- [6] Ali, S. I., Hussain, M., Lee, S., Bilal, H. S. M., Hussain, J., & Satti, F. A., “Ensemble feature ranking for cost-based non-overlapping groups: A case study of chronic kidney disease diagnosis in developing countries” in IEEE-Access, 8, pp. 215623-215648, 2020.
- [7] Segal, Z., Ehrenberg, B., Koren, G., Kalifa D., Maor, G., Radinsky, K., & Elad, G., “Machine learning algorithm for early detection of end-stage renal disease” in BMC-Nephrology, 21(1), pp. 1-10, 2020.
- [8] Navaneeth, B., and Suchetha, M., “A dynamic pooling based convolutional neural network approach to detect chronic kidney disease”, Biomedical Signal Control & Processing Operation, 62, p. 102068, 2020.
- [9] Qin, J., Feng, C., Liu, C., Chen, B., Liu, Y., & Chen, L., “A machine learning methodology for diagnosing chronic kidney disease”, IEEE-Access, 8, pp. 20991-21002, 2019.
- [10] Abdelaziz A., Riad, A. M., Mahmoud, A. N., & Salama, A. S., “A machine learning model for predicting of chronic kidney disease-based internet of things and cloud computing in smart cities” in the Springer Cham, pp. 93-114, 2019.
- [11] Snegha, J., Bhavani, S., Charanya, R., Tharani, V., & Preetha, S. D. “Chronic Kidney Disease Prediction Using Data Mining” in the 2020 ic-ETITE (International Conference on Emerging Trends in Information Technology and Engineering) IEEE, pp. 1-5, February 2020.
- [12] Herath, D., & Ekanayake, I. U., “Chronic Kidney Disease Prediction Using Machine Learning Methods” in the 2020 MERCon (Moratuwa Engineering Research Conference) IEEE, pp. 260-265, July 2020.

- [13] Shahbaz, M., Yashfi, S. Y., Sakib, N., Islam, M. A., Pantho, S. S., & Islam, T., “Risk Prediction of Chronic Kidney Disease Using Machine Learning Algorithms” in the 2020 11th ICCCNT (International Conference on Computing, Communication and Networking Technologies) IEEE, pp. 1-5, July 2020.
- [14] Thangavelu, M., & Harimoorthy, K., “Multi-disease prediction model using improved SVM-radial bias technique in healthcare monitoring system” in the Journal of Humanized Computing & Ambient Intelligence, 12(3), pp. 3715-3723, 2021.
- [15] Ding, C., Ren, L., Chen, G., Li, X., Xue, W., & Li, Y., “Prediction of Chronic Kidney Disease Using Adaptive Hybridized Deep Convolutional Neural Network on the Internet of Medical Things Platform”, IEEE-Access, 8, pp. 100497-100508, 2020.

Predicting Chronic Kidney Disease

ORIGINALITY REPORT

17%

SIMILARITY INDEX

14%

INTERNET SOURCES

5%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	8%
2	Submitted to Daffodil International University Student Paper	4%
3	link.springer.com Internet Source	<1%
4	Ebrahim Mohammed Senan, Mosleh Hmoud Al-Adhaileh, Fawaz Waselallah Alsaade, Theyazn H. H. Aldhyani et al. "Diagnosis of Chronic Kidney Disease Using Effective Classification Algorithms and Recursive Feature Elimination Techniques", Journal of Healthcare Engineering, 2021 Publication	<1%
5	Submitted to University of Leeds Student Paper	<1%
6	downloads.hindawi.com Internet Source	<1%
7	"Intelligent and Fuzzy Techniques for Emerging Conditions and Digital	<1%