A

Major Project

On

A MACHINE LEARNING METHODOLOGY FOR DIAGNOSING CHRONIC KIDNEY DISEASE

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

SARA	(207R1A05N8)
ANUMULA ABHITEJ REDDY	(207R1A05J7)

Under the Guidance of

Dr. V. NARESH KUMAR

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V) , ,Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled **"A MACHINE LEARNING METHODOLOGY FOR DIAGNOSING CHRONIC KIDNEY DISEASE"** being submitted by **SARA(207R1A05N8) & ANUMULA ABHITEJ REDDY (207R1A05J7)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-2024.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

DR.V. NARESH KUMAR
(Associate Professor)
INTERNAL GUIDE

DR. A. RAJIREDDY
DIRECTOR

DR. K. SRUJAN RAJU
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on____________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Dr. V. Naresh Kumar,** Associate Professor for his exemplary guidance, monitoring, and constantencouragement throughout the project work. The blessing, help, and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G.Vinesh Shanker, Dr. J. Narasimharao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughoutthe course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

SARA	**(207R1A05N8)**
ANUMULA ABHITEJ REDDY	**(207R1A05J7)**

ABSTRACT

Chronic kidney disease (CKD) is a global health problem with a high morbidity and mortality rate, and it induces other diseases. Since there are no obvious symptoms during the early stages of CKD, patients often fail to notice the disease. Early detection of CKD enables patients to receive timely treatment to ameliorate the progression of this disease. Machine learning models can effectively aid clinicians achieve this goal due to their fast and accurate recognition performance. In this study, we propose a machine-learning methodology for diagnosing CKD. The CKD data set was obtained from the University of California Irvine (UCI) machine learning repository, which has a large number of missing values.

KNN imputation was used to fill in the missing values, which selects several complete samples with the most similar measurements to process the missing data for each incomplete sample. Missing values are usually seen in real-life medical situations because patients may miss some measurements for various reasons. After effectively filling out the incomplete data set, six machine learning algorithms (logistic regression, random forest, support vector machine, k-nearest neighbor, naive Bayes classifier, and feed-forward neural network) were used to establish models. Among these machine learning models, random forest achieved the best performance with 99.75% diagnosis accuracy. By analyzing the misjudgments generated by the established models, we proposed an integrated model that combines logistic regression and random forest by using perceptron, which could achieve an average accuracy of 99.83% after ten times of simulation. Hence, we speculated that this methodology could be applicable to more complicated clinical data for disease diagnosis.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

Chronic kidney disease (CKD) is a global health concern affecting about 10% of the population worldwide. Prevalence rates vary, with 10.8% in China, 10%-15% in the United States, and 14.7% in Mexico. CKD is characterized by a gradual decline in renal function, often remaining asymptomatic until around 25% of kidney function is lost. It carries a high morbidity and mortality risk, notably contributing to cardiovascular disease. As a progressive, irreversible condition, early detection and diagnosis are crucial for timely intervention and improved outcomes.

1.2 PROJECT PURPOSE

The purpose of developing a machine learning methodology for diagnosing Chronic Kidney Disease (CKD) is to revolutionize the early detection and accuracy of CKD diagnosis. By harnessing advanced algorithms, this project aims to identify CKD in its initial, often asymptomatic stages, enabling timely intervention and personalized treatment plans. Through data-driven insights and predictive modeling, the methodology seeks to reduce false positives and negatives, streamline healthcare processes, and improve patient outcomes. Additionally, this innovative approach will contribute to a more efficient and cost-effective healthcare system while addressing the growing global burden of CKD and its associated complications.

1.3 PROJECT FEATURES

The innovative machine learning methodology for diagnosing Chronic Kidney Disease (CKD) represents a groundbreaking approach aimed at transforming CKD diagnosis and treatment. Through the utilization of advanced algorithms, it offers early and remarkably precise detection of CKD, playing a pivotal role in enhancing patient outcomes. Beyond diagnosis, this methodology tailors treatment plans to the unique characteristics of each patient, ensuring individualized care. Moreover, it provides healthcare professionals with invaluable data-driven insights, empowering them to make informed decisions and optimize patient management.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “What must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

Diagnosing Chronic Kidney Disease (CKD) seeks to enhance early detection, accuracy, and personalization of CKD diagnosis. It addresses challenges posed by asymptomatic early stages, while also considering data quality, privacy, and ethical concerns in handling medical information. The primary goal is to create an effective system that advances CKD diagnosis and seamlessly integrates into healthcare, all while maintaining ethical and regulatory compliance.

2.2 EXISTING SYSTEM

Several studies have explored Chronic Kidney Disease (CKD) and its management. One study highlighted the benefit of blood pressure control in HIV patients. Another developed a data imputation and diagnosis model using the KNN algorithm. A third study suggested that probabilistic neural networks could improve CKD diagnostics. Yet another found that feed-forward neural networks performed well in CKD data analysis. Additional research focused on attribute assessment and classification models, with the JRip algorithm showing promise. Feature selection was improved using Ant Lion Optimization, enhancing classification accuracy.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- The system lacks the capability to identify CKD stages.
- MLP separators prove effective in predicting liver disease, particularly in cases of HBV-induced hepatic cirrhosis and liver failure.

2.3 PROPOSED SYSTEM

In the proposed system for Chronic Kidney Disease (CKD) classification, a diverse set of machine-learning algorithms is harnessed to determine whether a patient has CKD. Six models, namely K-nearest neighbors (KNN), Support Vector Machine (SVM), Random Forest, Decision Tree, AdaBoost, and XG Boost, are utilized, in addition to a simple deep neural network (DNN). For binary classification, SVM is employed as a supervised learning model, while KNN predicts values by comparing patient features with the training dataset. Decision trees are used for visual representation but are often limited in accuracy. Random Forest addresses this limitation by combining multiple decision trees. AdaBoost and XG Boost are boosting techniques that enhance the performance of weak classifiers, while DNNs, based on feature-rich deep neural networks, enable the detection of critical diseases by processing input data through multiple layers of nodes. Prior to applying these classification algorithms, a feature selection method is applied to optimize the model's performance by eliminating unnecessary features. This comprehensive approach leverages various machine learning techniques and neural networks to improve CKD classification accuracy, ensuring a more effective and robust diagnostic system.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- RFE (Recursive Feature Elimination) helps select the most crucial features for predicting the target outcome in the dataset.
- Diverse Algorithmic Approach harnesses a variety of machine-learning algorithms including K-nearest neighbors (KNN), Support Vector Machine (SVM), Random Forest, and Logistic Regression.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited**.** The expenditures must be justified. Thus the developed system as 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.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements:

- **System** : Pentium IV
- **Hard Disk** : 20 GB.
- **RAM** : 4 GB(min)

2.5.2 SOFTWARE REQUIREMENTS:

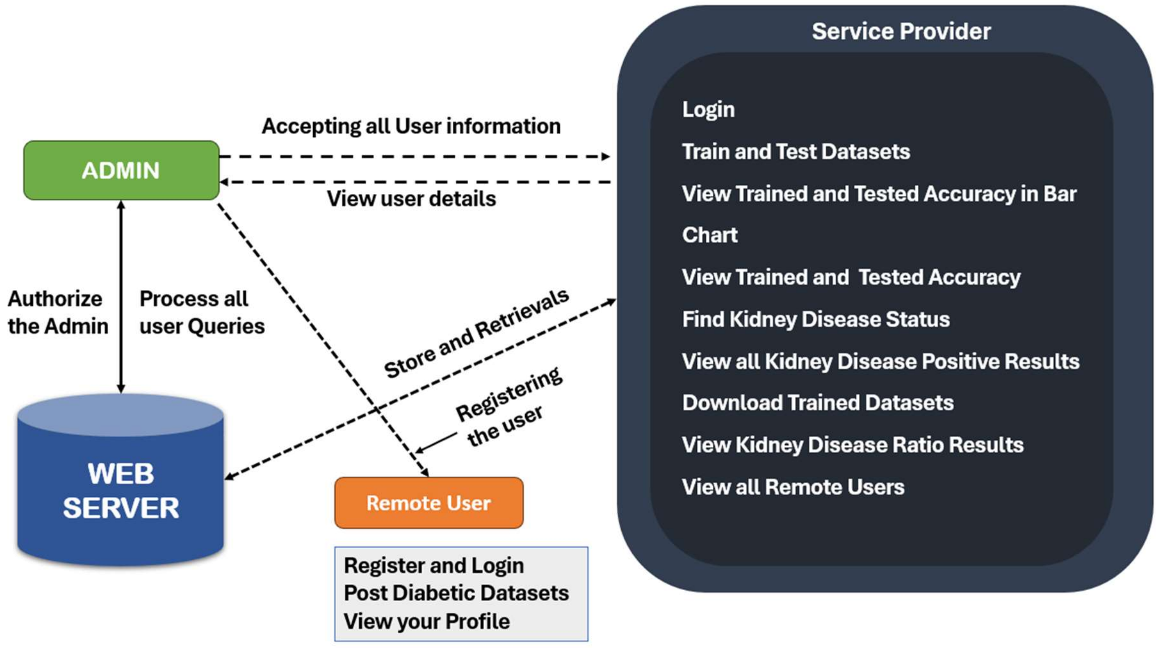
Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements:

• **Coding Language**	:	Python3.7
• **Database**	:	MySQL (XAMPP Server)
• **Front End**	:	HTML, CSS, JavaScript
• **Back End**	:	Django-ORM

3.ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE



This project architecture shows the procedure followed for classification, starting from input to final prediction.

Figure 3.1: Project Architecture of the proposed schema for A Machine Learning Methodology for Diagnosing Chronic Kidney Disease

3.2 DESCRIPTION

The proposed system architecture comprises four main components: Admin, Web Server, Remote User, and Service Provider. Admin manages user accounts and datasets, while the Web Server facilitates communication and data processing among components. Remote Users register, submit diabetic datasets, and manage their profiles, accessing results through the web interface. The Service Provider handles dataset processing, model training/testing, and result generation, offering insights into kidney disease status. Through secure authentication and data management mechanisms, the system enables efficient interaction, ensuring accurate predictions and seamless user experience in accessing and analyzing diabetic datasets for kidney disease diagnosis.

3.3 USE CASE DIAGRAM:

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

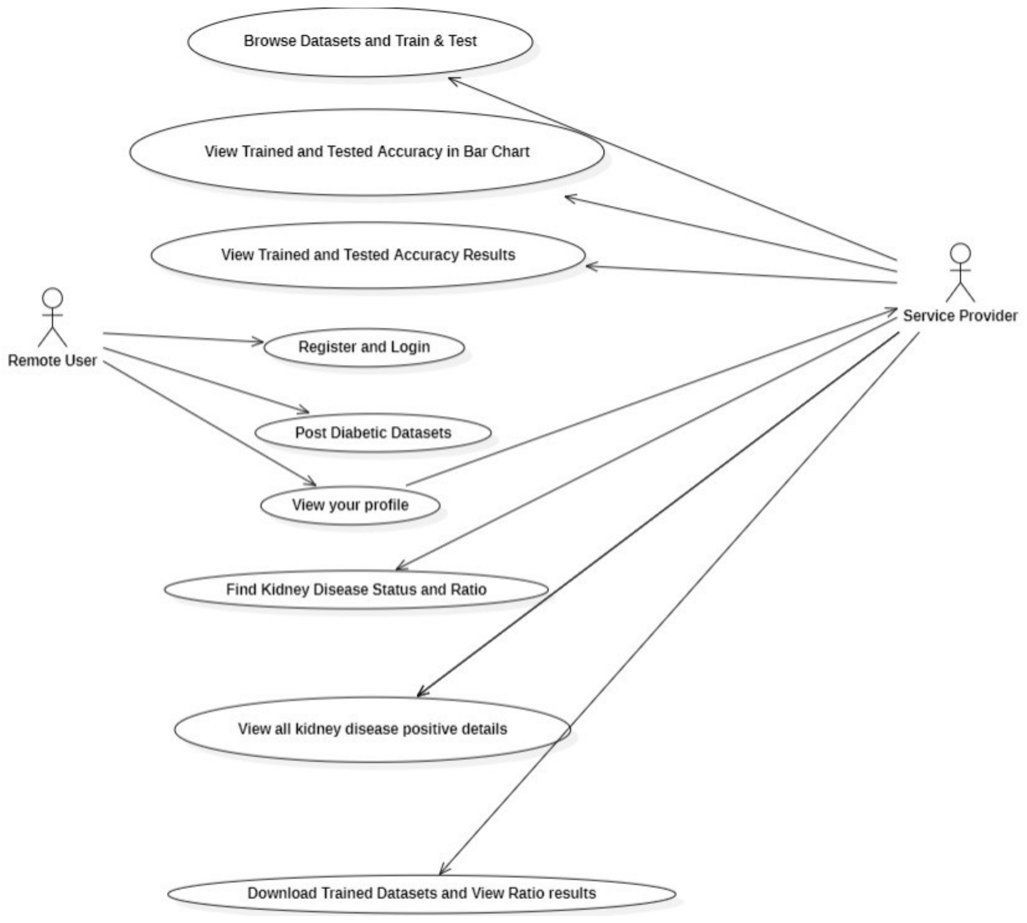

Browse Datasets and Train & Test
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Service Provider
Remote User
Register and Login
Post Diabetic Datasets
View your profile
Find Kidney Disease Status and Ratio
View all kidney disease positive details
Download Trained Datasets and View Ratio results

Figure 3.2: Use Case Diagram for A Machine Learning Methodology for Diagnosing Chronic Kidney Disease

3.4 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

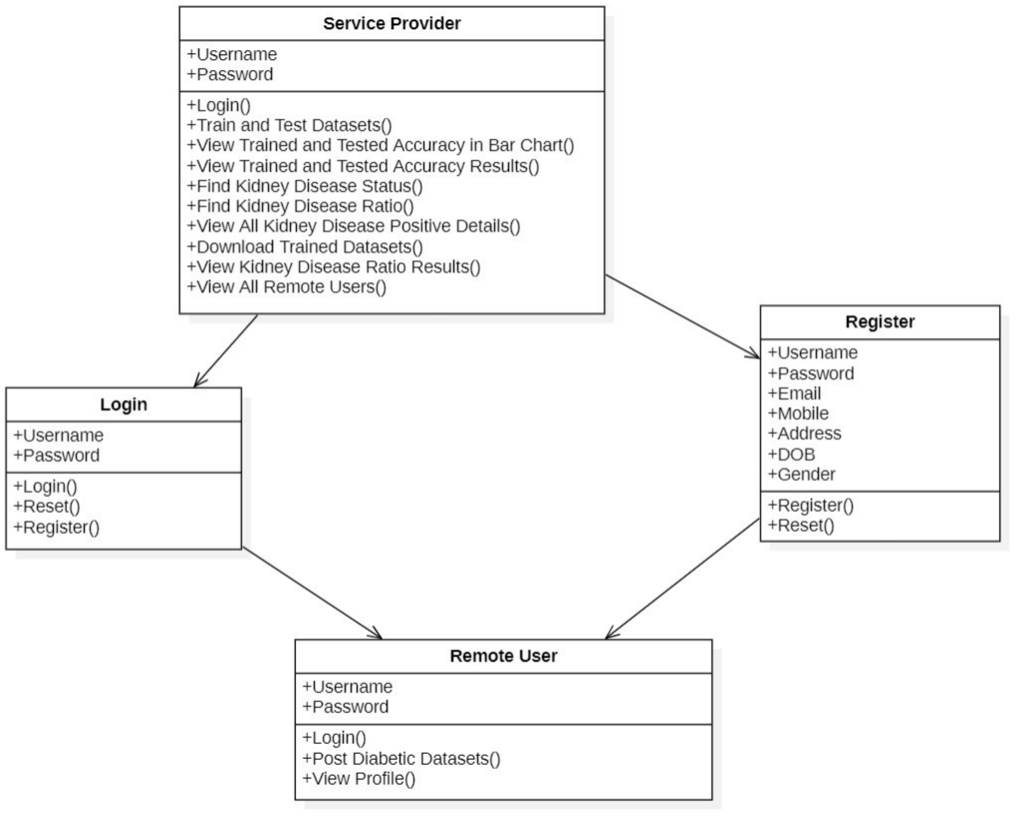

Service Provider
+Username
+Password
+Login()
+Train and Test Datasets()
+View Trained and Tested Accuracy in Bar Chart()
+View Trained and Tested Accuracy Results()
+Find Kidney Disease Status()
+Find Kidney Disease Ratio()
+View All Kidney Disease Positive Details()
+Download Trained Datasets()
+View Kidney Disease Ratio Results()
+View All Remote Users()
Register
+Username
+Password
+Email
+Mobile
+Address
+DOB
+Gender
+Register()
+Reset()
Login
+Username
+Password
+Login()
+Reset()
+Register()
Remote User
+Username
+Password
+Login()
+Post Diabetic Datasets()
+View Profile()

Figure 3.3: Class Diagram for A Machine Learning Methodology for Diagnosing Chronic Kidney Disease

3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

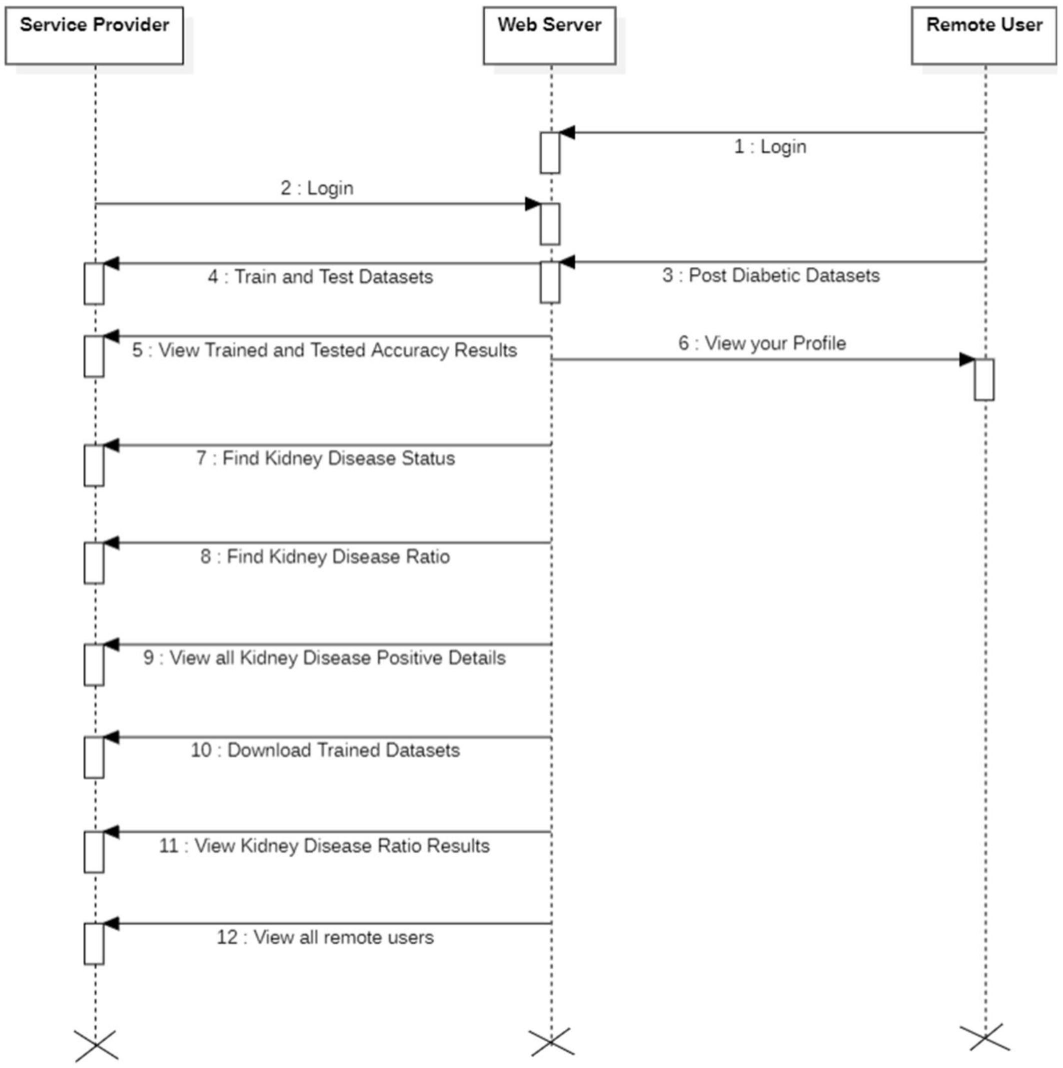

Service Provider
Web Server
Remote User
1 : Login
2 : Login
4 : Train and Test Datasets
3 : Post Diabetic Datasets
5 : View Trained and Tested Accuracy Results
6 : View your Profile
7 : Find Kidney Disease Status
8 : Find Kidney Disease Ratio
9 : View all Kidney Disease Positive Details
10 : Download Trained Datasets
11 : View Kidney Disease Ratio Results
12 : View all remote users

Figure 3.4: Sequence Diagram for A Machine Learning Methodology for Diagnosing Chronic Kidney Disease

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

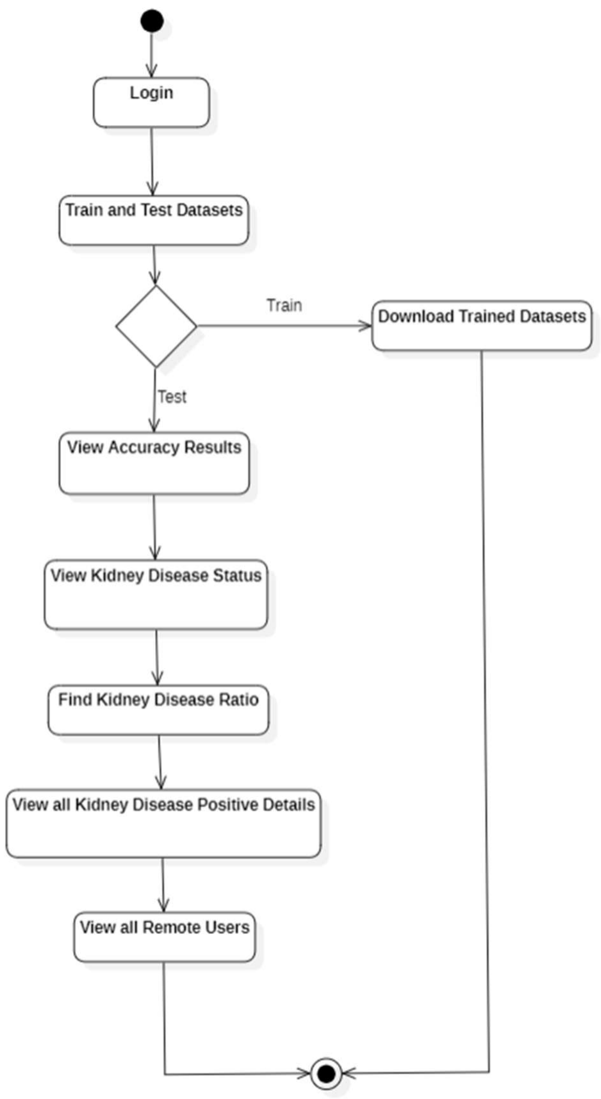

Login
Train and Test Datasets
Train
Download Trained Datasets
Test
View Accuracy Results
View Kidney Disease Status
Find Kidney Disease Ratio
View all Kidney Disease Positive Details
View all Remote Users

Figure 3.5: Activity Diagram for A Machine Learning Methodology for Diagnosing Chronic Kidney Disease

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from django.db.models import  Count, Avg
from django.shortcuts import render, redirect
from django.db.models import Count
from django.db.models import Q
import datetime
import xlwt
from django.http import HttpResponse
# Importing the libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_curve, auc, confusion_matrix,
classification_report,accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
# Create your views here.
from Remote_User.models import
ClientRegister_Model,kidney_model,kidney_disease_model,detection_ratio_model,detection_ac
curacy_model


def serviceproviderlogin(request):
    if request.method  == "POST":
        admin = request.POST.get('username')
        password = request.POST.get('password')
        if admin == "Admin" and password =="Admin":
            detection_accuracy_model.objects.all().delete()
            return redirect('View_Remote_Users')

    return render(request,'SProvider/serviceproviderlogin.html')
```

```
def Find_Kidney_Disease_Ratio(request):
    detection_ratio_model.objects.all().delete()
    ratio = ""
    kword = 'Positive'
    print(kword)
    obj = kidney_disease_model.objects.all().filter(Q(prediction=kword))
    obj1 = kidney_disease_model.objects.all()
    count = obj.count();
    count1 = obj1.count();
    ratio = (count / count1) * 100
    if ratio != 0:
        detection_ratio_model.objects.create(names=kword, ratio=ratio)

    ratio1 = ""
    kword1 = 'Negative'
    print(kword1)
    obj1 = kidney_disease_model.objects.all().filter(Q(prediction=kword1))
    obj11 = kidney_disease_model.objects.all()
    count1 = obj1.count();
    count11 = obj11.count();
    ratio1 = (count1 / count11) * 100
    if ratio1 != 0:
        detection_ratio_model.objects.create(names=kword1, ratio=ratio1)


    obj = detection_ratio_model.objects.all()
    return render(request, 'SProvider/Find_Kidney_Disease_Ratio.html', {'objs': obj})

def View_Kidney_Disease_Positive_Details(request):

    keyword="Positive"

    obj = kidney_disease_model.objects.all().filter(prediction=keyword)
    return render(request, 'SProvider/View_Kidney_Disease_Positive_Details.html', {'objs': obj})

def View_Remote_Users(request):
    obj=ClientRegister_Model.objects.all()
    return render(request,'SProvider/View_Remote_Users.html',{'objects':obj})

def ViewTrendings(request):
    topic =
kidney_disease_model.objects.values('topics').annotate(dcount=Count('topics')).order_by('-
dcount')
    return  render(request,'SProvider/ViewTrendings.html',{'objects':topic})
```

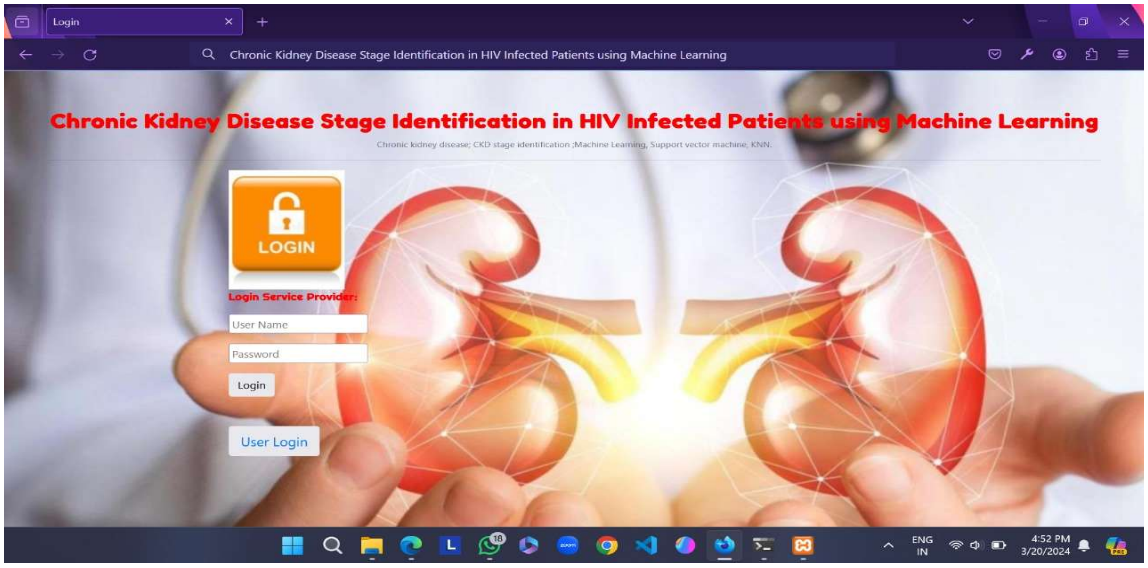
```
def charts(request,chart_type):
    chart1 = detection_ratio_model.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts.html", {'form':chart1, 'chart_type':chart_type})

def charts1(request,chart_type):
    chart1 = detection_accuracy_model.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts1.html", {'form':chart1, 'chart_type':chart_type})

def Find_Kidney_Disease_Status(request):

    obj =kidney_disease_model.objects.all()
    return render(request, 'SProvider/Find_Kidney_Disease_Status.html', {'list_objects': obj})

def likeschart(request,like_chart):
    charts =detection_accuracy_model.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/likeschart.html", {'form':charts, 'like_chart':like_chart})

def Download_Trained_DataSets(request):

    response = HttpResponse(content_type='application/ms-excel')
    # decide file name
    response['Content-Disposition'] = 'attachment; filename="TrainedData.xls"'
    # creating workbook
    wb = xlwt.Workbook(encoding='utf-8')
    # adding sheet
    ws = wb.add_sheet("sheet1")
    # Sheet header, first row
    row_num = 0
    font_style = xlwt.XFStyle()
    # headers are bold
    font_style.font.bold = True
    # writer = csv.writer(response)
    obj = kidney_disease_model.objects.all()
    data = obj  # dummy method to fetch data.
    for my_row in data:
        row_num = row_num + 1
    ws.write(row_num, 0, my_row.id1, font_style)
    ws.write(row_num, 1, my_row.age, font_style)
    ws.write(row_num, 2, my_row.bp, font_style)
    ws.write(row_num, 3, my_row.sg, font_style)
    ws.write(row_num, 4, my_row.al, font_style)
    ws.write(row_num, 5, my_row.su, font_style)
    ws.write(row_num, 6, my_row.rbc, font_style)
    ws.write(row_num, 7, my_row.pc, font_style)
    ws.write(row_num, 8, my_row.pcc, font_style)
    ws.write(row_num, 9, my_row.ba, font_style)
    ws.write(row_num, 10, my_row.bgr, font_style)
    ws.write(row_num, 11, my_row.bu, font_style)
    ws.write(row_num, 12, my_row.sc, font_style)
    ws.write(row_num, 13, my_row.sod, font_style)
    ws.write(row_num, 14, my_row.pot, font_style)
    ws.write(row_num, 15, my_row.hemo, font_style)
    ws.write(row_num, 16, my_row.pcv, font_style)
    ws.write(row_num, 17, my_row.wc, font_style)
    ws.write(row_num, 18, my_row.rc, font_style)
    ws.write(row_num, 19, my_row.htn, font_style)
    ws.write(row_num, 20, my_row.dm, font_style)
    ws.write(row_num, 21, my_row.cad, font_style)
    ws.write(row_num, 22, my_row.appet, font_style)
    ws.write(row_num, 23, my_row.pe, font_style)
    ws.write(row_num, 24, my_row.ane, font_style)
    ws.write(row_num, 25, my_row.prediction, font_style)

wb.save(response)
return response
```

```
def train_model(request):
    detection_accuracy_model.objects.all().delete()
    # Reading the dataset
    kidney = pd.read_csv("kidney_disease.csv")
    kidney.head()
    # Information about the dataset
    kidney.info()
    # Description of the dataset
    kidney.describe()
    # To see what are the column names in our dataset
    print(kidney.columns)
    # Mapping the text to 1/0 and cleaning the dataset
    kidney[['htn', 'dm', 'cad', 'pe', 'ane']] = kidney[['htn', 'dm', 'cad', 'pe', 'ane']].replace(
        to_replace={'yes': 1, 'no': 0})
    kidney[['rbc', 'pc']] = kidney[['rbc', 'pc']].replace(to_replace={'abnormal': 1, 'normal': 0})
    kidney[['pcc', 'ba']] = kidney[['pcc', 'ba']].replace(to_replace={'present': 1, 'notpresent': 0})
    kidney[['appet']] = kidney[['appet']].replace(to_replace={'good': 1, 'poor': 0, 'no': np.nan})
    kidney['classification'] = kidney['classification'].replace(
        to_replace={'ckd': 1.0, 'ckd\t': 1.0, 'notckd': 0.0, 'no': 0.0})
    kidney.rename(columns={'classification': 'class'}, inplace=True)

    kidney['pe'] = kidney['pe'].replace(to_replace='good', value=0)  # Not having pedal edema is
good
    kidney['appet'] = kidney['appet'].replace(to_replace='no', value=0)
    kidney['cad'] = kidney['cad'].replace(to_replace='\tno', value=0)
    kidney['dm'] = kidney['dm'].replace(to_replace={'\tno': 0, '\tyes': 1, ' yes': 1, '': np.nan})
    kidney.drop('id', axis=1, inplace=True)
    kidney.head()

    # This helps us to count how many NaN are there in each column
    len(kidney) - kidney.count()
    # This shows number of rows with missing data
    kidney.isnull().sum(axis=1)
    # This is a visualization of missing data in the dataset
    sns.heatmap(kidney.isnull(), yticklabels=False, cbar=False, cmap='viridis')
    # This shows number of complete cases and also removes all the rows with NaN
    kidney2 = kidney.dropna()
    print(kidney2.shape)
    # Now our dataset is clean
    sns.heatmap(kidney2.isnull(), yticklabels=False, cbar=False, cmap='viridis')
    sns.heatmap(kidney2.corr())
```

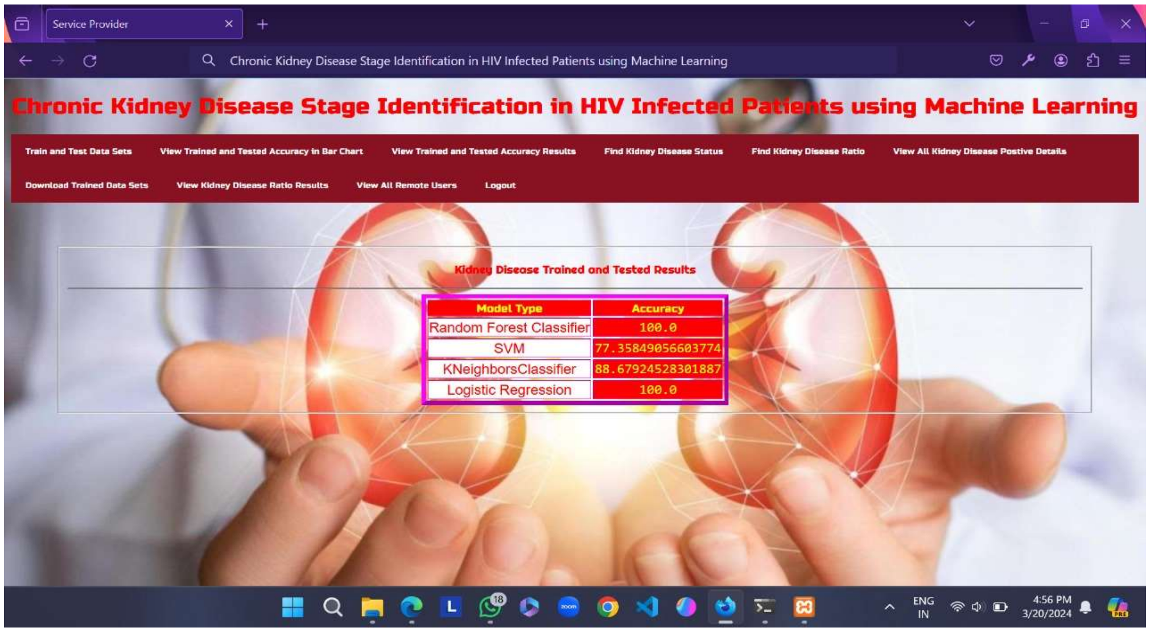
```
# Counting number of normal vs. abnormal red blood cells of people having chronic kidney
disease
    print(kidney2.groupby('rbc').rbc.count().plot(kind="bar"))

    # This plot shows the patient's sugar level compared to their ages
    kidney2.plot(kind='scatter', x='age', y='su');


# plt.show()
    # Shows the maximum blood pressure having chronic kidney disease
    print(kidney2.groupby('class').bp.max())
    print(kidney2['dm'].value_counts(dropna=False))
    X_train, X_test, y_train, y_test = train_test_split(kidney2.iloc[:, :-1], kidney2['class'], test_size=0.33,
                                                        random_state=44, stratify=kidney2['class'])
    print(X_train.shape)
    y_train.value_counts()

    print("RANDOM FOREST CLASSFIER")

    rfc = RandomForestClassifier(random_state=22)
    rfc_fit = rfc.fit(X_train, y_train)
    rfc_pred = rfc_fit.predict(X_test)
    print(confusion_matrix(y_test, rfc_pred))
    print(classification_report(y_test, rfc_pred))
    accuracy_score(y_test, rfc_pred)

    print("ACCURACY")
    print(accuracy_score(y_test, rfc_pred) * 100)

    detection_accuracy_model.objects.create(names="Random Forest Classifier",
ratio=accuracy_score(y_test, rfc_pred) * 100)

    print("SVM CLASSFIER")

    svm = SVC()
    svm_fit = svm.fit(X_train, y_train)
    svm_pred = svm_fit.predict(X_test)
    print(confusion_matrix(y_test, svm_pred))
    print(classification_report(y_test, svm_pred))
    accuracy_score(y_test, svm_pred)
    print("ACCURACY")
    print(accuracy_score(y_test, svm_pred) * 100)

  detection_accuracy_model.objects.create(names="SVM",ratio=accuracy_score(y_test, svm_pred)*100

  print("KNeighborsClassifier")
print("LogisticRegression")

    logmodel = LogisticRegression()
    logmodel.fit(X_train, y_train)
    predictions = logmodel.predict(X_test)
    print(classification_report(y_test, predictions))
    print(confusion_matrix(y_test, predictions))
    accuracy_score(y_test, predictions)

    print("ACCURACY")
    print(accuracy_score(y_test, predictions) * 100)

    detection_accuracy_model.objects.create(names="Logistic Regression",
ratio=accuracy_score(y_test, predictions) * 100)

    status = ''
    type = ''
    obj1 = kidney_model.objects.values('id1',
    'age',
    'bp',
    'sg',
    'al',
    'su',
    'rbc',
    'pc',
    'pcc',
    'ba',
    'bgr',
    'bu',
    'sc',
    'sod',
    'pot',
    'hemo',
    'pcv',
    'wc',
    'rc',
    'htn',
    'dm',
    'cad',
    'appet',
    'pe',
    'ane'
    )
```

5. SCREENSHOTS

Screenshot 5.1: Service Provider Login


Login
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic kidney disease; CKD stage identification ;Machine Learning, Support vector machine, KNN.
LOGIN
Login Service Provider:
User Name
Password
Login
User Login

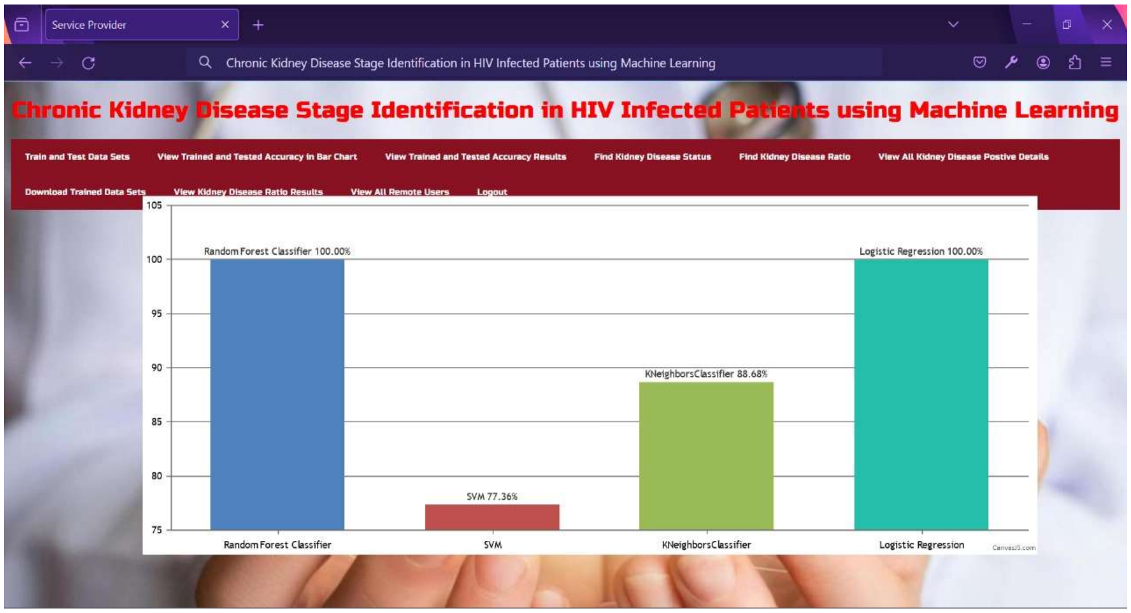
Screenshot 5.2: Train and Test Datasets


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout

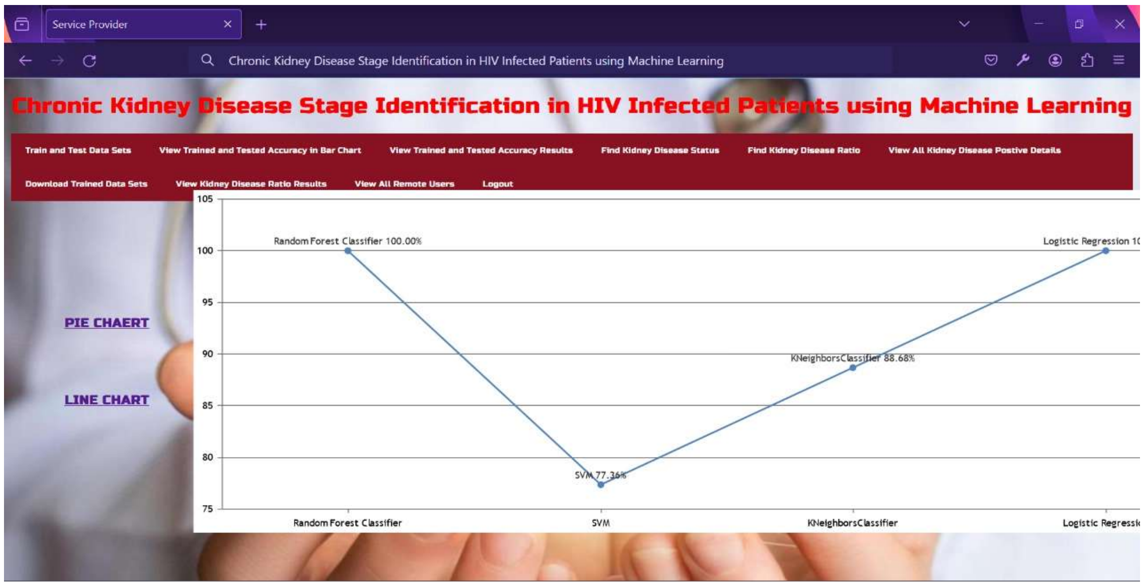
Kidney Disease Trained and Tested Results

Model Type	Accuracy
Random Forest Classifier	100.0
SVM	77.35849056603774
KNeighborsClassifier	88.67924528301887
Logistic Regression	100.0

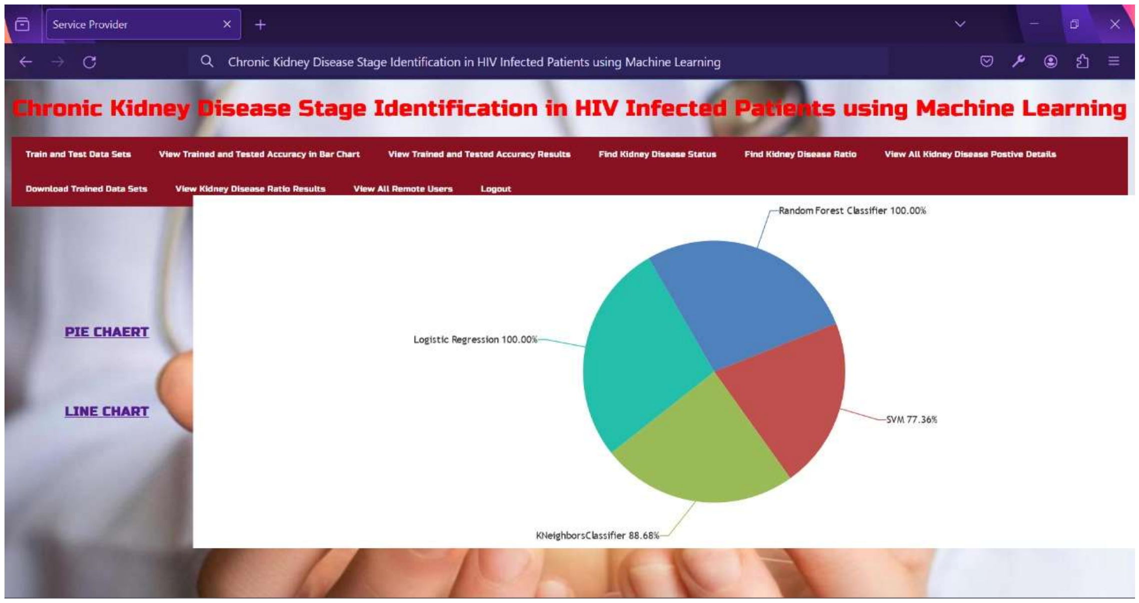
Screenshot 5.3: View Trained and Tested Accuracy in Bar Chart


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout
Random Forest Classifier 100.00%
SVM 77.36%
KNeighborsClassifier 88.68%
Logistic Regression 100.00%
Random Forest Classifier
SVM
KNeighborsClassifier
Logistic Regression

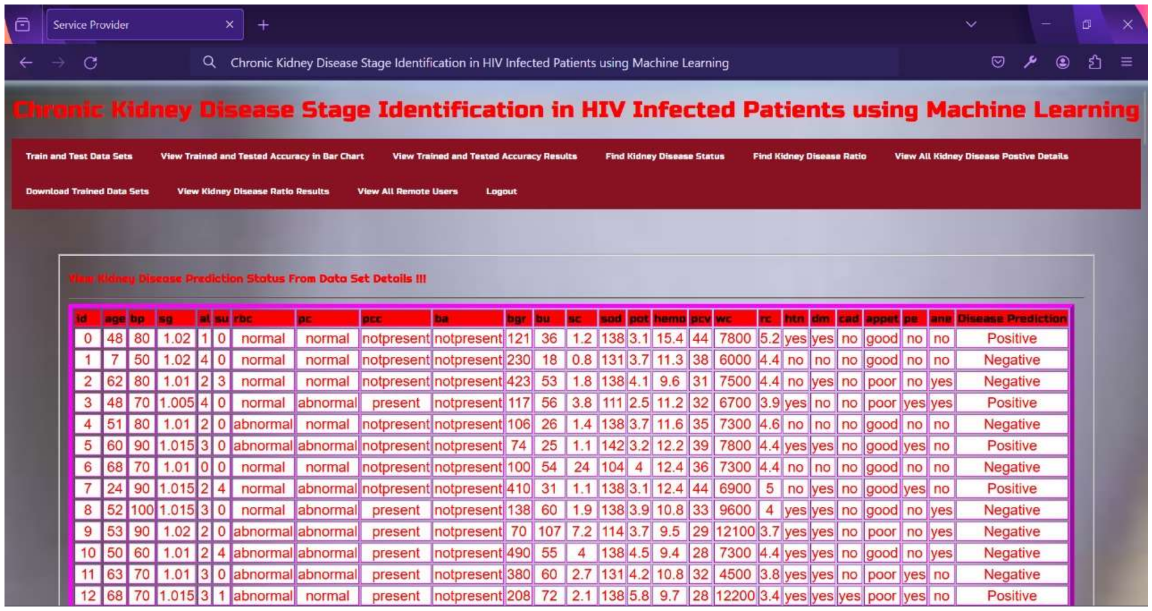
Screenshot 5.4: View Trained and Tested Accuracy Results in Line Chart


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Random Forest Classifier 100.00%
SVM 77.36%
KNeighborsClassifier 88.68%
Logistic Regression 1
Random Forest Classifier
SVM
KNeighborsClassifier
Logistic Regressi

Screenshot 5.5: View Trained and Tested Accuracy Results in Pie Chart


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Random Forest Classifier 100.00%
Logistic Regression 100.00%
SVM 77.36%
KNeighborsClassifier 88.68%

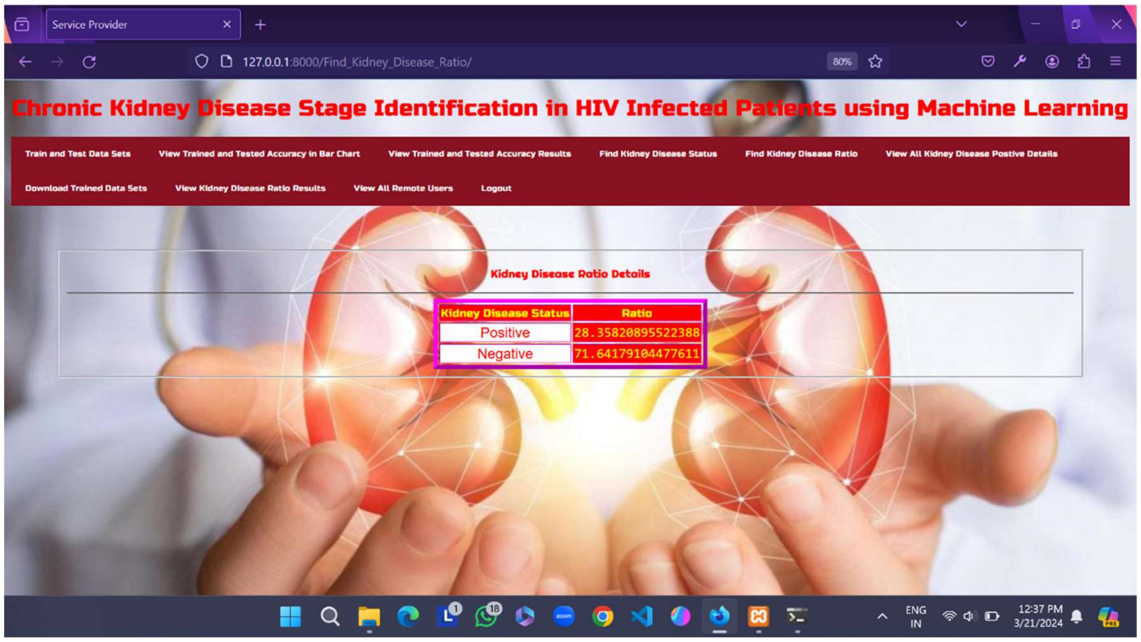
Screenshot 5.6: Find Kidney Disease Status


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout

View Kidney Disease Prediction Status From Data Set Details !!!

id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	Disease Prediction
0	48	80	1.02	1	0	normal	normal	notpresent	notpresent	121	36	1.2	138	3.1	15.4	44	7800	5.2	yes	yes	no	good	no	no	Positive
1	7	50	1.02	4	0	normal	normal	notpresent	notpresent	230	18	0.8	131	3.7	11.3	38	6000	4.4	no	no	no	good	no	no	Negative
2	62	80	1.01	2	3	normal	normal	notpresent	notpresent	423	53	1.8	138	4.1	9.6	31	7500	4.4	no	yes	no	poor	no	yes	Negative
3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	3.8	111	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes	Positive
4	51	80	1.01	2	0	abnormal	normal	notpresent	notpresent	106	26	1.4	138	3.7	11.6	35	7300	4.6	no	no	no	good	no	no	Negative
5	60	90	1.015	3	0	abnormal	abnormal	notpresent	notpresent	74	25	1.1	142	3.2	12.2	39	7800	4.4	yes	yes	no	good	yes	no	Positive
6	68	70	1.01	0	0	normal	normal	notpresent	notpresent	100	54	24	104	4	12.4	36	7300	4.4	no	no	no	good	no	no	Negative
7	24	90	1.015	2	4	normal	abnormal	notpresent	notpresent	410	31	1.1	138	3.1	12.4	44	6900	5	no	yes	no	good	yes	no	Positive
8	52	100	1.015	3	0	normal	abnormal	present	notpresent	138	60	1.9	138	3.9	10.8	33	9600	4	yes	yes	no	good	no	yes	Negative
9	53	90	1.02	2	0	abnormal	abnormal	present	notpresent	70	107	7.2	114	3.7	9.5	29	12100	3.7	yes	yes	no	poor	no	yes	Negative
10	50	60	1.01	2	4	abnormal	abnormal	present	notpresent	490	55	4	138	4.5	9.4	28	7300	4.4	yes	yes	no	good	no	yes	Negative
11	63	70	1.01	3	0	abnormal	abnormal	present	notpresent	380	60	2.7	131	4.2	10.8	32	4500	3.8	yes	yes	no	poor	yes	no	Negative
12	68	70	1.015	3	1	abnormal	normal	present	notpresent	208	72	2.1	138	5.8	9.7	28	12200	3.4	yes	yes	yes	poor	yes	no	Positive

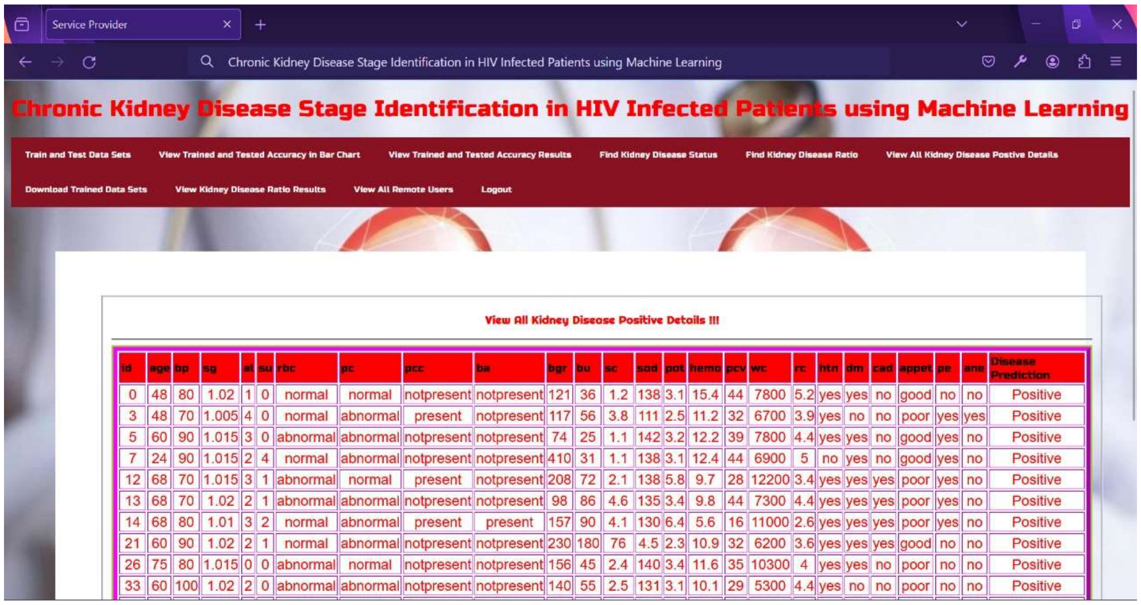
Screenshot 5.7: Kidney Disease Ratio Details


Service Provider
127.0.0.1:8000/Find_Kidney_Disease_Ratio/
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout
ENG IN
12:37 PM 3/21/2024

Kidney Disease Ratio Details

Kidney Disease Status	Ratio
Positive	28.35820895522388
Negative	71.64179104477611

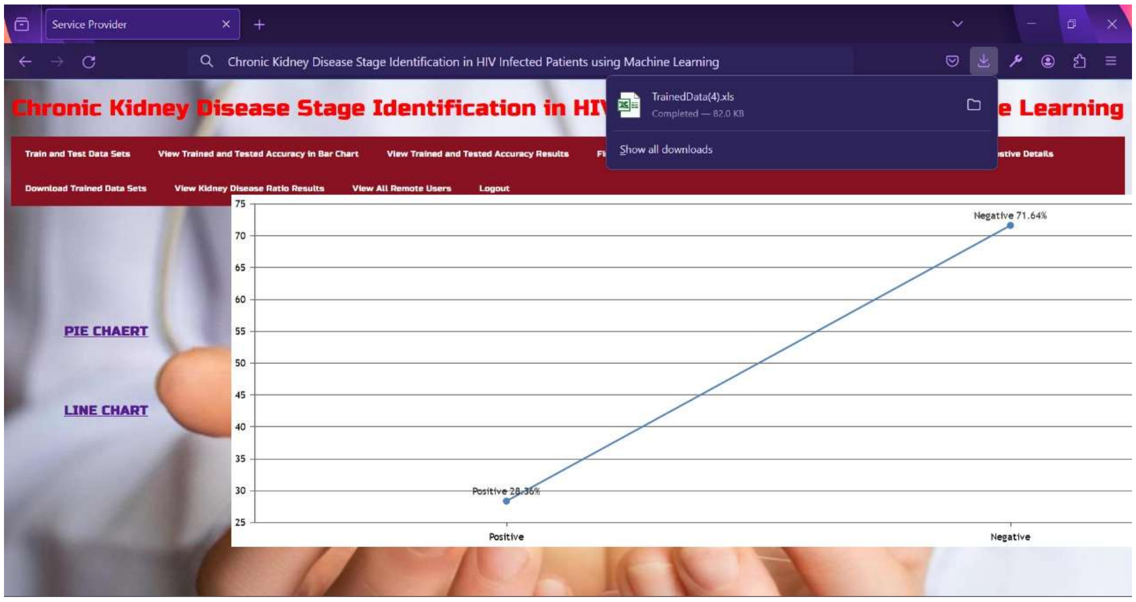
Screenshot 5.8: View All Kidney Disease Positive Details


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout

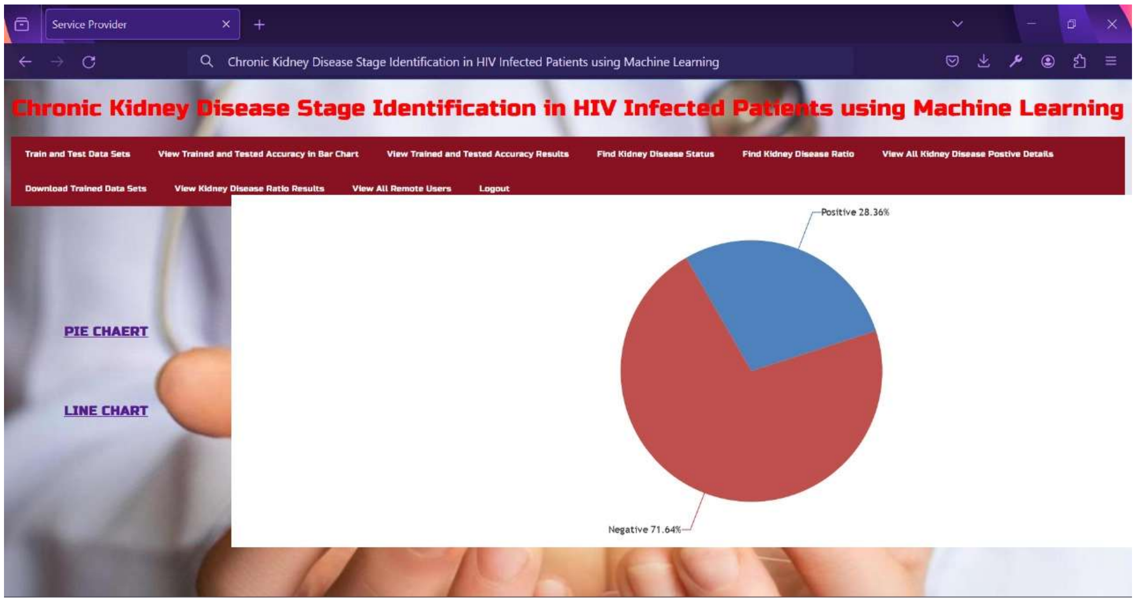
View All Kidney Disease Positive Details !!!

id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	Disease Prediction
0	48	80	1.02	1	0	normal	normal	notpresent	notpresent	121	36	1.2	138	3.1	15.4	44	7800	5.2	yes	yes	no	good	no	no	Positive
3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	3.8	111	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes	Positive
5	60	90	1.015	3	0	abnormal	abnormal	notpresent	notpresent	74	25	1.1	142	3.2	12.2	39	7800	4.4	yes	yes	no	good	yes	no	Positive
7	24	90	1.015	2	4	normal	abnormal	notpresent	notpresent	410	31	1.1	138	3.1	12.4	44	6900	5	no	yes	no	good	yes	no	Positive
12	68	70	1.015	3	1	abnormal	normal	present	notpresent	208	72	2.1	138	5.8	9.7	28	12200	3.4	yes	yes	yes	poor	yes	no	Positive
13	68	70	1.02	2	1	abnormal	abnormal	notpresent	notpresent	98	86	4.6	135	3.4	9.8	44	7300	4.4	yes	yes	yes	poor	yes	no	Positive
14	68	80	1.01	3	2	normal	abnormal	present	present	157	90	4.1	130	6.4	5.6	16	11000	2.6	yes	yes	yes	poor	yes	no	Positive
21	60	90	1.02	2	1	normal	abnormal	notpresent	notpresent	230	180	76	4.5	2.3	10.9	32	6200	3.6	yes	yes	yes	good	no	no	Positive
26	75	80	1.015	0	0	abnormal	normal	notpresent	notpresent	156	45	2.4	140	3.4	11.6	35	10300	4	yes	yes	no	poor	no	no	Positive
33	60	100	1.02	2	0	abnormal	abnormal	notpresent	notpresent	140	55	2.5	131	3.1	10.1	29	5300	4.4	yes	no	no	poor	no	no	Positive

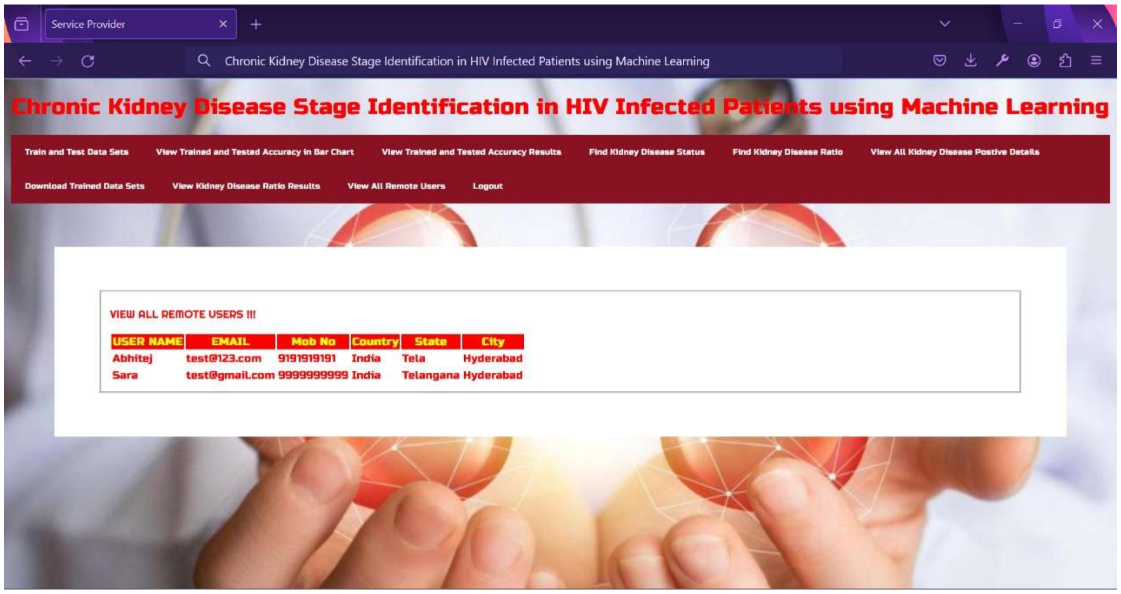
Screenshot 5.9: Download Trained Datasets


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
TrainedData(4).xls
Completed — 82.0 KB
Show all downloads
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Negative 71.64%
Positive 28.36%
Positive
Negative

Screenshot 5.10: View Kidney Disease Ratio Results


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Positive 28.36%
Negative 71.64%

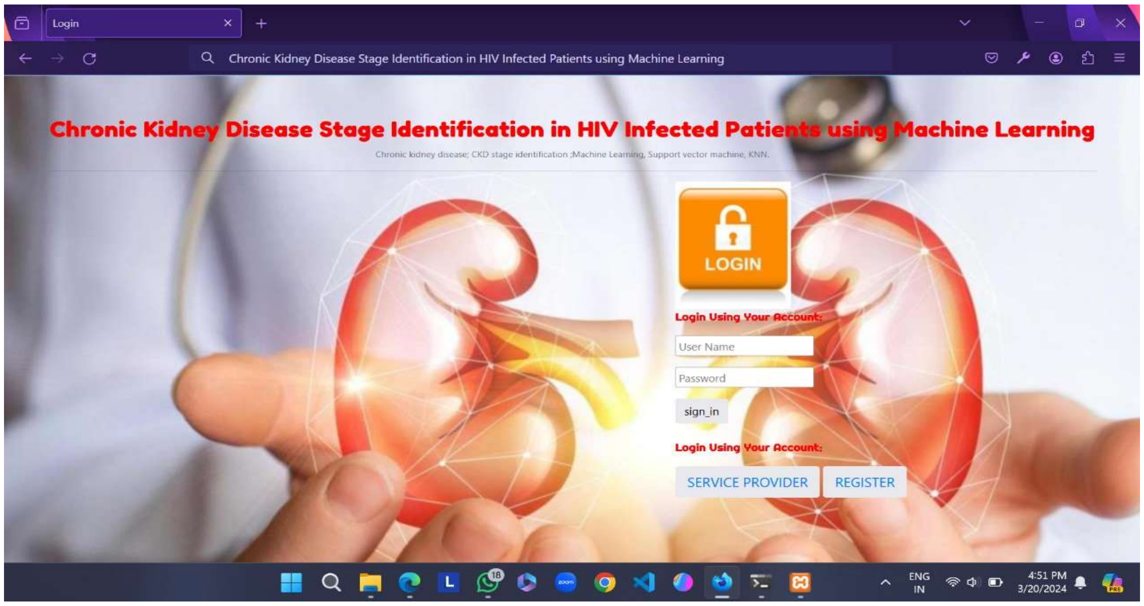
Screenshot 5.11: View All Remote Users


Service Provider
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Train and Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Find Kidney Disease Status
Find Kidney Disease Ratio
View All Kidney Disease Postive Details
Download Trained Data Sets
View Kidney Disease Ratio Results
View All Remote Users
Logout

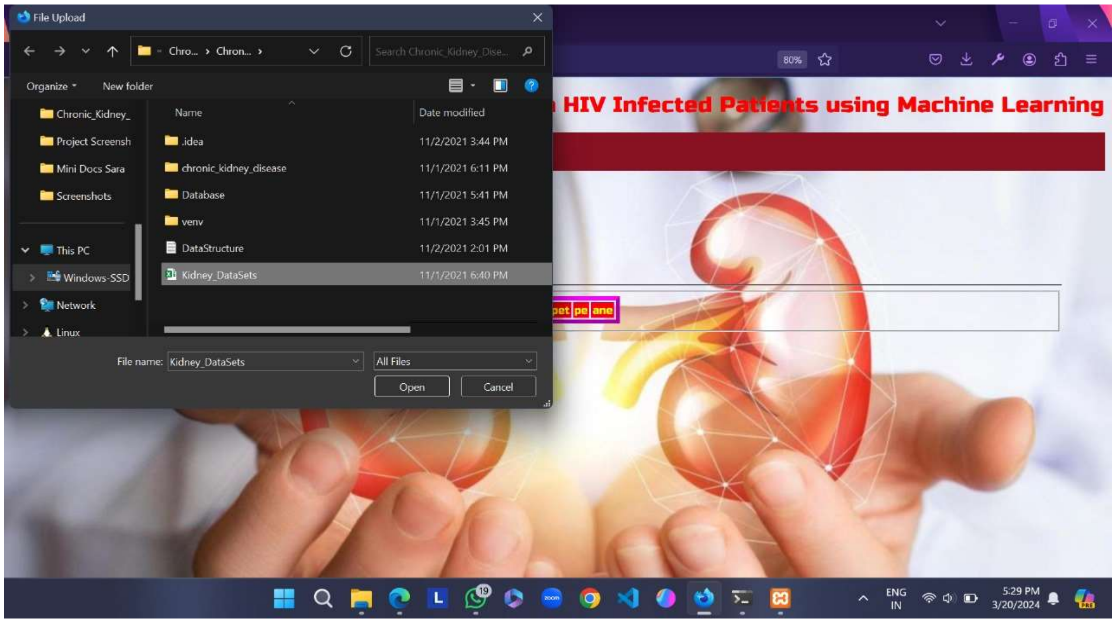
VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	Mob No	Country	State	City
Abhitej	test@123.com	9191919191	India	Tela	Hyderabad
Sara	test@gmail.com	9999999999	India	Telangana	Hyderabad

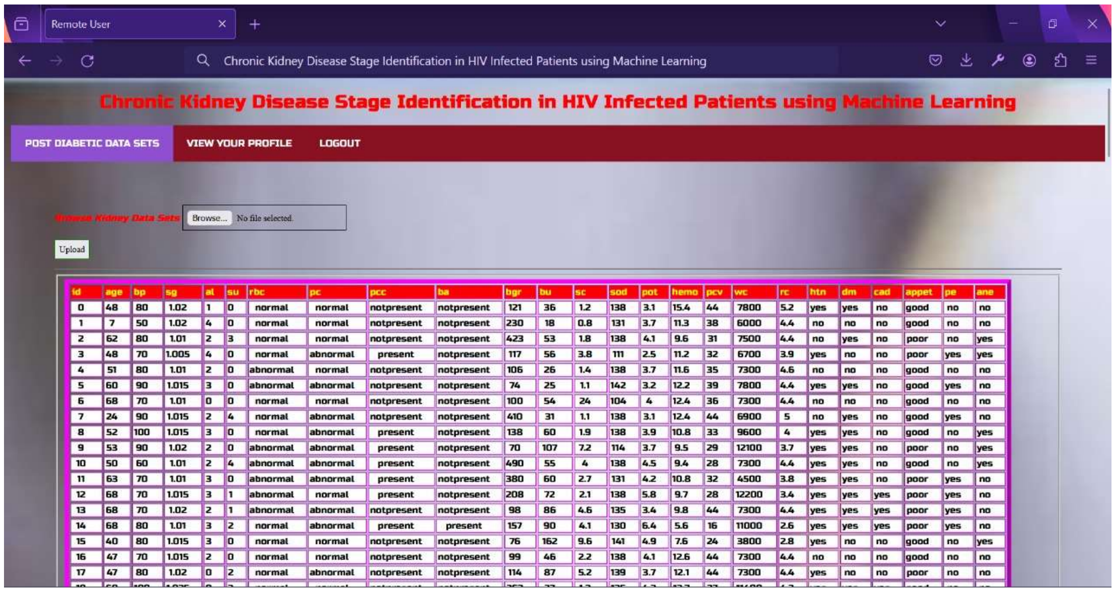
Screenshot 5.12: Remote User Login


Login
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic kidney disease; CKD stage identification ;Machine Learning, Support vector machine, KNN.
LOGIN
Login Using Your Account:
User Name
Password
sign_in
Login Using Your Account:
SERVICE PROVIDER
REGISTER
ENG
IN
4:51 PM
3/20/2024

Screenshot 5.13: Post Diabetic Datasets

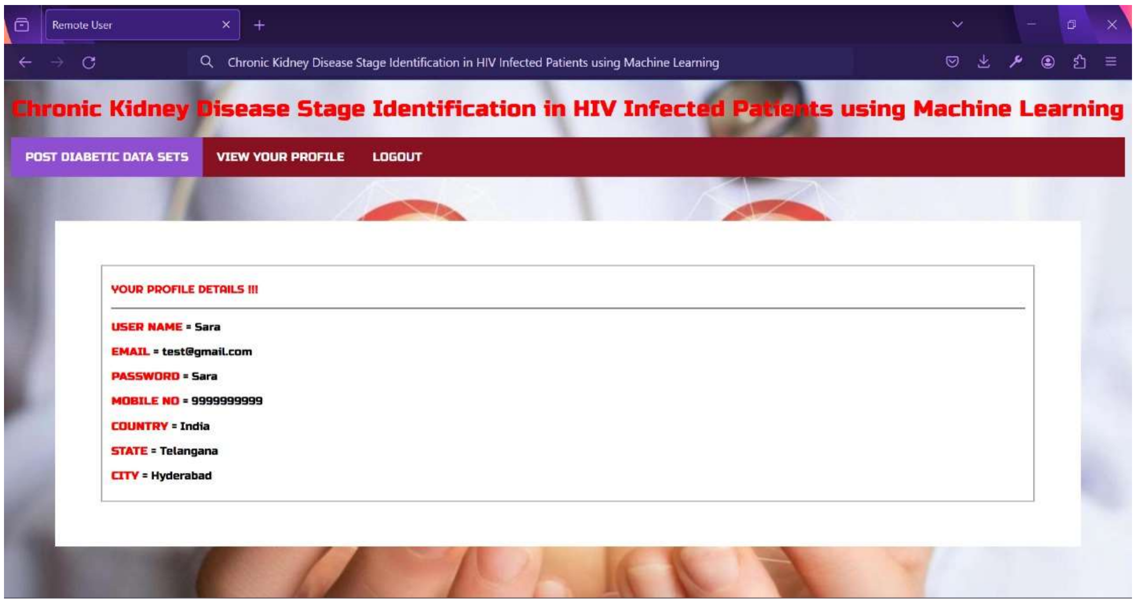


Screenshot 5.14: View Posted Diabetic Datasets



id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane
0	48	80	1.02	1	0	normal	normal	notpresent	notpresent	121	36	1.2	138	3.1	15.4	44	7800	5.2	yes	yes	no	good	no	no
1	7	50	1.02	4	0	normal	normal	notpresent	notpresent	230	18	0.8	131	3.7	11.3	38	6000	4.4	no	no	no	good	no	no
2	62	80	1.01	2	3	normal	normal	notpresent	notpresent	423	53	1.8	138	4.1	9.6	31	7500	4.4	no	yes	no	poor	no	yes
3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	3.8	111	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes
4	51	80	1.01	2	0	abnormal	normal	notpresent	notpresent	106	26	1.4	138	3.7	11.6	35	7300	4.6	no	no	no	good	no	no
5	60	90	1.015	3	0	abnormal	abnormal	notpresent	notpresent	74	25	1.1	142	3.2	12.2	39	7800	4.4	yes	yes	no	good	yes	no
6	68	70	1.01	0	0	normal	normal	notpresent	notpresent	100	54	24	104	4	12.4	36	7300	4.4	no	no	no	good	no	no
7	24	90	1.015	2	4	normal	abnormal	notpresent	notpresent	410	31	1.1	138	3.1	12.4	44	6900	5	no	yes	no	good	yes	no
8	52	100	1.015	3	0	normal	abnormal	present	notpresent	138	60	1.9	138	3.9	10.8	33	9600	4	yes	yes	no	good	no	yes
9	53	90	1.02	2	0	abnormal	abnormal	present	notpresent	70	107	7.2	114	3.7	9.5	29	12100	3.7	yes	yes	no	poor	no	yes
10	50	60	1.01	2	4	abnormal	abnormal	present	notpresent	490	55	4	138	4.5	9.4	28	7300	4.4	yes	yes	no	good	no	yes
11	63	70	1.01	3	0	abnormal	abnormal	present	notpresent	380	60	2.7	131	4.2	10.8	32	4500	3.8	yes	yes	no	poor	yes	no
12	68	70	1.015	3	1	abnormal	normal	present	notpresent	208	72	2.1	138	5.8	9.7	28	12200	3.4	yes	yes	yes	poor	yes	no
13	68	70	1.02	2	1	abnormal	abnormal	notpresent	notpresent	98	86	4.6	135	3.4	9.8	44	7300	4.4	yes	yes	yes	poor	yes	no
14	68	80	1.01	3	2	normal	abnormal	present	present	157	90	4.1	130	6.4	5.6	16	11000	2.6	yes	yes	yes	poor	yes	no
15	40	80	1.015	3	0	normal	normal	notpresent	notpresent	76	162	9.6	141	4.9	7.6	24	3800	2.8	yes	no	no	good	no	yes
16	47	70	1.015	2	0	normal	normal	notpresent	notpresent	99	46	2.2	138	4.1	12.6	44	7300	4.4	no	no	no	good	no	no
17	47	80	1.02	0	2	normal	abnormal	notpresent	notpresent	114	87	5.2	139	3.7	12.1	44	7300	4.4	yes	no	no	poor	no	no

Screenshot 5.15: View your Profile


Remote User
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
Chronic Kidney Disease Stage Identification in HIV Infected Patients using Machine Learning
POST DIABETIC DATA SETS
VIEW YOUR PROFILE
LOGOUT

YOUR PROFILE DETAILS !!!

USER NAME = Sara
EMAIL = test@gmail.com
PASSWORD = Sara
MOBILE NO = 9999999999
COUNTRY = India
STATE = Telangana
CITY = Hyderabad

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

Testing is an integral and indispensable part of the software development and quality assurance process. Its fundamental purpose is to unearth errors and flaws within a work product, ensuring that the final software system is robust, reliable, and capable of meeting user expectations. Testing serves as the systematic process of meticulously scrutinizing every conceivable fault or weakness that might be present in a software product, ranging from individual components and subassemblies to complete assemblies or finished products. At its core, testing functions as a comprehensive evaluation mechanism, verifying the functionality, performance, and reliability of software components and systems.

The overarching objective is to confirm that the software not only adheres to its specified requirements but also aligns with user expectations. Equally important is the assurance that the software does not fail in any unacceptable manner under different operational conditions. To achieve these objectives, testing encompasses a wide array of test types, each addressing specific testing requirements.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Integration testing, like other testing methodologies, is fundamentally designed to uncover errors and inconsistencies. It aims to expose issues that may arise when previously tested components come together and interact. This testing approach endeavors to verify that the combined system behaves as expected, adhering to predefined requirements and meeting user expectations. Integration testing examines different integration points, including APIs, databases, communication protocols, and data transfer mechanisms.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstration that functions tested are available as specified by the business and technical requirements, system documentation and user manuals.

Functional testing is centered on the following items:

Valid Input	: Identified classes of valid input must be accepted.
Invalid Input	: Identified classes of invalid input must be rejected.
Functions	: Identified functions must be exercised.
Output	: Identified classes of application output must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases

6.2.4 SYSTEM TEST

System testing is a critical phase in the software testing process, with the primary objective of ensuring that the entire integrated software system aligns with the specified requirements. This type of testing evaluates the software configuration as a whole to verify that it produces known and predictable results. System testing often involves comprehensive scenarios and use cases, mimicking real-world interactions, in order to assess the software's behavior in a holistic manner.

A noteworthy example of system testing is the configuration-oriented system integration test, which is focused on examining how different components and modules are configured to function as a coherent whole. This type of testing is fundamentally grounded in process descriptions and flows, emphasizing pre-determined process links and integration points. It ensures that the software system functions seamlessly when all its components are integrated, ultimately meeting user expectations and fulfilling its intended purpose.

6.2.5 WHITE BOX TESTING

White Box Testing is a sophisticated testing approach that requires the software tester to possess knowledge of the inner workings, structure, and often the programming language of the software being tested. In this method, the tester has insight into the software's internal architecture, allowing themto delve into the logic and algorithms that drive its functionality. White Box Testing is typically employed to scrutinize areas of the software that are not easily accessible through black box testing, providing a more detailed and in-depth assessment of its internal mechanisms.

This form of testing is especially useful for identifying coding errors, logical flaws, and security vulnerabilities within the software. It aids in uncovering issues related to the software's control flow, data flow, and overall code quality. White Box Testing aims to improve the structural quality of the software by ensuring that it adheres to coding standards, meets design specifications, and performs as intended at the code level.

6.2.6 BLACK BOX TESTING

Black Box Testing stands in contrast to White Box Testing, as it involves testing the software without any knowledge of its inner workings, structure, or programming language. Testers approach the software as a "black box," focusing solely on its external behavior and functionality. This method does not require an understanding of the software's code or internal logic; instead, it evaluates how the software performs based on the inputs provided and the outputs it generates. Black Box Testing is conducted without considering the software's underlying mechanisms, making it ideal for assessing the software from an end-user perspective.

Test cases are designed based on defined specifications or requirements documents, and the test focuses on validating that the software functions as expected and complies with the documented criteria. This approach is valuable for validating user- facing features, usability, and overall system behavior. In essence, Black Box Testing is centered on evaluating what the software does rather than how it accomplishes it, making it an essential part of ensuring that the software meets user expectations and operates as intended.

6.3 TEST CASES

6.3.1 CLASSIFICATION

S.No	Test Type	Test Objective	Test Case Description	Expected Outcome
1	Registration	Verify user registration functionality	Attempt to register a new user with valid credentials	User is successfully registered and can log in to access the system
2	Login	Verify user login functionality	Attempt to log in with correct username and password	User is successfully logged in and directed to the system dashboard
3	Dataset Submission	Verify dataset submission functionality	Upload a diabetic dataset file to the system	Dataset is successfully uploaded and stored in the system database
4	Model Training	Verify model training functionality	Initiate training of a machine learning model with provided dataset	Model is trained successfully without errors and ready for testing
5	Results Viewing	Verify result visualization functionality	Initiate prediction of kidney disease status based on input data	Accuracy metrics are displayed correctly, providing clear insights for analysis

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

The classification of Chronic Kidney Disease (CKD) stages in HIV-infected patients holds significant importance for both patients and healthcare providers, as it enables timely and accurate clinical decisions, ultimately leading to improved patient outcomes. In our project, we conducted a comprehensive analysis comparing the performance of state-of-the-art machine learning algorithms, including Deep Neural Networks (DNN), for the classification of CKD in patients with HIV. Our study revealed that DNN exhibited superior performance compared to other algorithms in accurately categorizing CKD stages. This finding underscores the potential of advanced machine learning techniques, particularly DNN, in enhancing the diagnostic capabilities for CKD in HIV-infected individuals.

In conclusion, our project underscores the significance of advanced machine learning methodologies, such as DNN, in the classification of CKD stages in HIV-infected patients. Through our findings, we aim to contribute to the ongoing efforts in leveraging technology to address the complex challenges associated with kidney disease management, with the ultimate goal of improving patient outcomes and quality of life.

7.2 FUTURE SCOPE

Refine Algorithms for Better Accuracy: Continuously improve algorithms by gathering diverse datasets to ensure accurate CKD detection across various demographics, regions, and medical conditions.

Integrate with Electronic Health Records (EHR): Connect the CKD identification system with healthcare providers' electronic health record systems for real-time diagnosis and monitoring, leading to quicker interventions and better patient outcomes.

Personalize Diagnosis and Treatment: Design CKD diagnosis and treatment plans based on individual patient characteristics like genetics, lifestyle, and existing conditions. This involves using machine learning to incorporate genetic and patient-reported data for personalized care.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] C. I. Bagnis, J. E. Heron, and D. M. Gracey, "Contemporary issues and new challenges in chronic kidney disease amongst people living with HIV," AIDS Res. T her., vol. 17.

[2] Y. Liu, C. Liu, J. Qin, L. Chen, C. Feng, and B. Chen, " A machine learning methodology for diagnosing chronic kidney disease," IEEE Access, vol. 8

[3] A. S. Anwar and E. H. A. Rady , " Prediction of kidney disease stages using data mining algorithms," Informatics Med. Unlocked, vol. 15, no. March.

[4] M. N. Amin, A. Al Imran, and F. T . Johora, "Classification of Chronic Kidney Disease using Logist ic Regression, Feedforward Neural Network and Wide Deep Learning," 2018 Int . Conf. Innov. Eng. Technol. ICIET 2018

[5] S. D. Sudarsan,N. Chet ty, K. S. Vaisla, " Role of attributes selection in classification of Chronic Kidney Disease patients".

[6] E. Perumal, P. Arulanthu, "Predicting the Chronic Kidney Disease using Various Classifiers," 4t h Int. Conf. Elect r. Electron. Commun. Comput. Technol. Optim. Tech. ICEECCOT 2019

[7] K. Shankar, G. Devika, P. Manickam, M. Ilayaraja, " Optimal Feature Selection for Chronic Kidney Disease Classification using Deep Learning Classifier," 2018 IEEE Int. Conf. Comput. Intell. Comput . Res. ICCIC 2018,

[8] R. Shinde, S. John, R. Jadhav, A. Maurya, R. Wable, and R. Dakshayani, "Chronic Kidney Disease Prediction and Recommendation of Suit able Diet Plan by using Machine Learning," 2019 Int. Conf. Nascent Technol. Eng. ICNTE 2019.

[9] S. C. Jat. R. Yadav, " Feature selection and dimensionality reduction methods for chronic disease prediction," Int. J. Sci. Technol. Res., vol. 9, no. 4.

[10] S. Saraswat , S. Vashisth, I. Dhall, CKD diagnosis using Multilayer Perceptron Classifier' 10t h International Conference on Cloud Computing, Data Science & Engineering (Confluence)" 2020.

8.2 GITHUB LINK

https://github.com/saramohd02/a-machine-learning-methodology-for-diagnosing-chronic-kidney-disease