Chronic Kidney Disease Analysis Prediction using ML Algorithms

1. INTRODUCTION

1.1 Overview

The field of biosciences have advanced to a larger extent and have generated large amounts of information from Electronic Health Records. This have given rise to the acute need of knowledge generation from this enormous amount of data. Data mining methods and machine learning play a major role in this aspect of biosciences. Chronic Kidney Disease (CKD) is a condition in which the kidneys are damaged and cannot filter blood as they always do. A family history of kidney diseases or failure, high blood pressure, type 2 diabetes may lead to CKD. This is a lasting damage to the kidney and chances of getting worse by time is high. The very common complications that results due to a kidney failure are heart diseases, anemia, bone diseases, high potassium and calcium. The worst case situation leads to complete kidney failure and necessitates kidney transplant to live. An early detection of CKD can improve the quality of life to a greater extent. This calls for good prediction algorithm to predict CKD at an earlier stage . Literature shows a wide range of machine learning algorithms employed for the prediction of CKD. This paper uses data preprocessing, data transformation and various classifiers to predict CKD and also proposes best Prediction framework for CKD. The results of the framework show promising results of better prediction at an early stage of CKD.

1.2 Purpose

The disability of the kidneys to perform their regular blood filtering function and others is called Chronic Kidney Disease (CKD). The term “chronic” describes the slow degradation of the kidney cells over a long period of time. This disease is a major kidney failure where the kidney sans blood filtering process and there is a heavy fluid buildup in the body. This leads to alarming increase of potasium and calcium salts in the body. Existence of high levels of these salts result in various other ailments in the body. The prime job of kidneys is to filter extra water and wastes from blood. The efficient functioning of this process is important to balance the salts and minerals present in our body. The right balance of salts are necessary to control blood pressure, activate hormones, build red blood cells, etc. A high concentration of calcium leads to various bone diseases and cystic ovaries in women. CKD also may lead to sudden illness or allergy to certain medicines. This state is called as Acute Kidney Injury(AKI). An increased blood pressure may lead to heart problems and heart attacks. CKD in many cases leads to permenant dialysis or kidney transplants. A history of kidney disease in the family also leads to high probability of CKD. Literature shows that almost one out of three people diagnosed with diabetes have CKD. Literature also presents evidences of early identification and care of CKD can improve the quality of the patients life. Prediction algorithms in machine learning can be intelligently used to predict the occurence of CKD and presents a method of early medication. The detailed review on literature shows the application of various machine learning algorithms to predict CKD.

2. LITERATURE SURVEY

2.1 Existing Problem

In recent years, few studies have been done on the classification or diagnosis of chronic kidney disease. In 2013, T. Di Noia et al. , presented a software tool that used the artificial neural network ANN to classify patient status, which is likely to lead to end-stage renal disease (ESRD). The classifiers were trained using the data collected at the University of Bari over a 38-year period, and the evaluation was done based on precision, recall, and F-measure. The presented software tool has been made available as both an Android mobile application and online web application.

Using data from Electronic Health Records (EHR) in 2014, H. S. Chase et al. identified two groups of patients in stage 3: 117 progressor patients (eGFR declined >3 ml/min/1.73m2 /year) and 364 non progressor patients (eGFR declined <1 ml/min/1.73m2) .Where GFR is a glomerular filtration rate that commonly used to detect CKD. Based on initial lab data recorded, the authors used Naïve Bayes and Logistic Regression classifiers to develop a predictive model for progression from stage 3 to stage 4. They compared the metabolic complications between the two groups and found that phosphate values were significantly higher, but bicarbonate, hemoglobin, calcium, and albumin values were significantly lower in progressors compared to non progressors, even if initial eGFR values were similar. Finally, they found that the probability of progression in patients classified as progressors was 81% (73% – 86%) and non- progressors was 17% (13% – 23%).

Later in 2016, K. A. Padmanaban and G. Parthiban aimed in their work to detect chronic kidney disease for diabetic patients using machine learning methods. In their research, they used 600 clinical records collected from a leading Chennai-based diabetes research center. The authors have tested the dataset using the decision tree and Naïve Bayes methods for classification using the WEKA tool. They concluded that the decision tree algorithm outweighs the Naïve Bayes with an accuracy of 91%.

P. Yildirim studied the effect of sampling algorithms in predicting chronic kidney disease. The experiment was done by comparing the effect of the three sampling algorithms: Resample, SMOTE, and Spread Sup Sample on the prediction by multilayer perceptron classification algorithm. The study showed that sampling algorithms could improve the classification algorithm performance, and the resample method has a higher accuracy among the sampling algorithms. On the other hand, Spread Sub Sample was better in terms of execution time.

A. J. Aljaaf et al. examined in their study the ability of four machine learning (ML) models for early prediction of CKD, which were: support vector machine (SVM), classification and regression tree (CART), logistic regression (LR), and multilayer perceptron neural network (MLP). By using the CKD dataset from UCI and seven features out of 24, they compared the performance of these ML models. The results showed that the MLP model had the highest AUC and sensitivity. It was also noticeable that logistic regression almost had the same performance as MLP but with the advantage of the simplicity of the LR algorithm. Therefore, in our study, we can use the LR algorithm as a start or a benchmark and then use more complex algorithms.

Lastly in 2019, J. Xiao et al. in their study established and compared nine ML models, including

LR, Elastic Net, ridge regression lasso regression SVM, RF, XGBoost, knearest neighbor and neural network to predict the progression of CKD. They used available clinical features from 551 CKD follow-up patients. They conclude that linear models have the overall predictive power with an average AUC above 0.87 and precision above 0.8 and 0.8, respectively

2.2 Proposed System

This part describes the dataset and contains block diagrams, flow diagrams,
evaluation matrices, and the study's procedure and methodology. The below figure(1) depicts the block diagram of the proposed system. The framework utilizes the CKD prediction dataset. After preprocessing and feature selection, the KNN, and logistic regression algorithms have been used. All the components of this diagram have been discussed in the following sub sections.

3. THEORITICAL ANALYSIS

3.1 Block Diagram

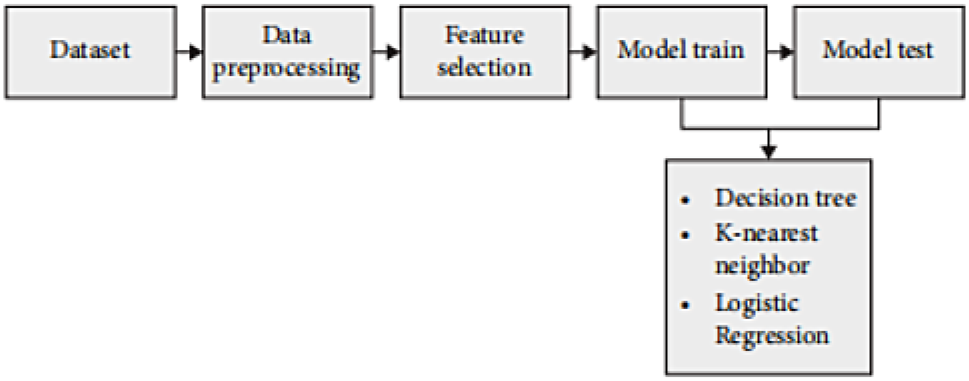


Figure-1 Block Diagram of the Proposed System

3.2 Hardware/Software Designing

- Google Co-Laboratory
- VS(Visual Studio) Code
- Python 3.10.4

4. EXPERIMENTAL INVESTIGATIONS

4.1 Dataset

The research was conducted using the CKD dataset . There are 400 rows and 25 columns in this dataset. The output column “class” has a value of either “1” or “0.” The value “0” indicates that the patient is not a CKD patient, while the value “1” shows that the patient is a CKD patient. Before preprocessing, Figure-(2) displays the total number of CKD and non-CKD entries in the output column. The overall number of CKD data is 250, whereas the total number of non-CKD data is 150.

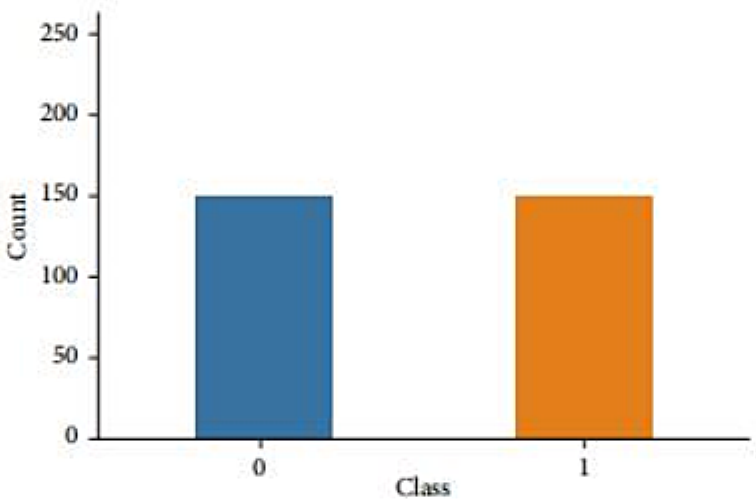
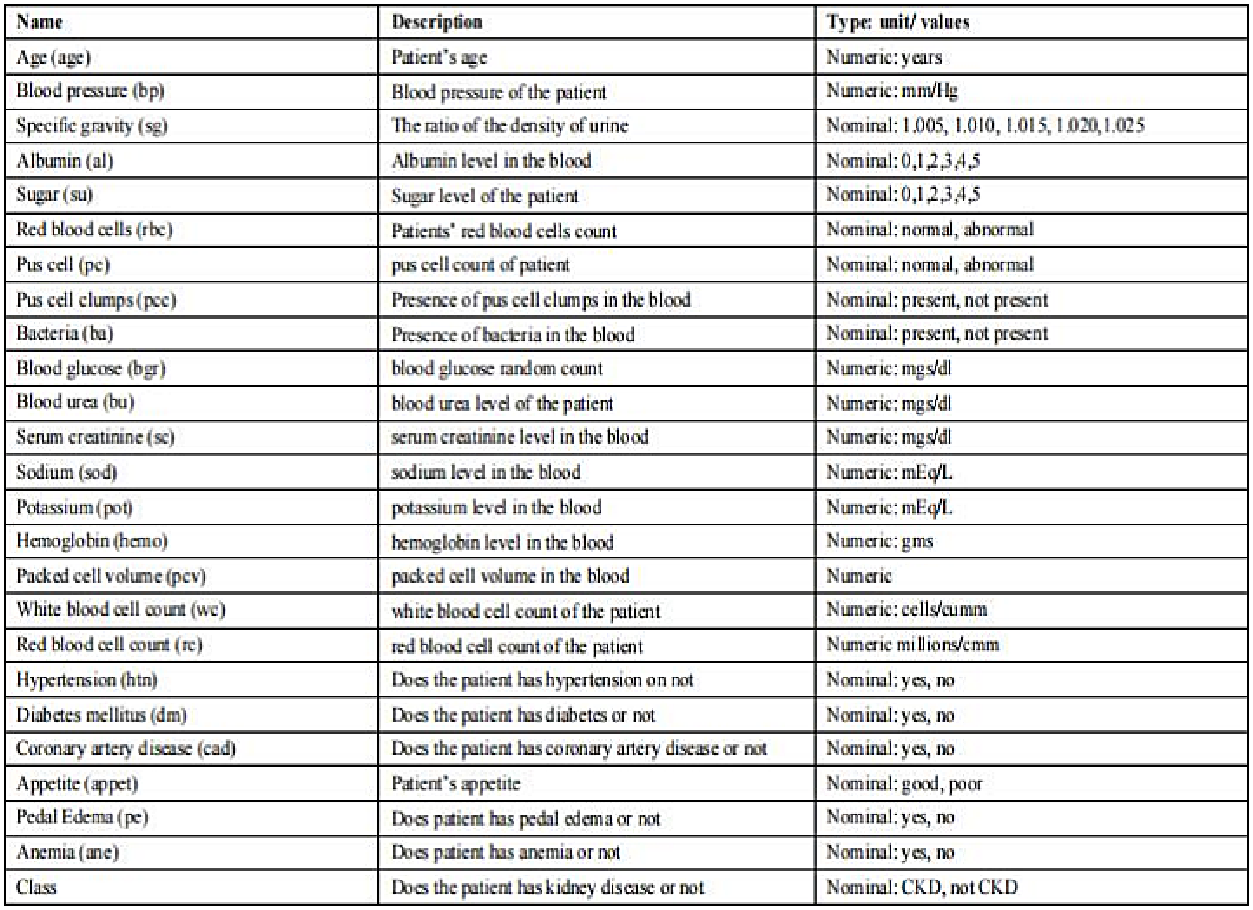


Figure-2 Total Amount of CKD and non-CKD data

4.2 Data Description

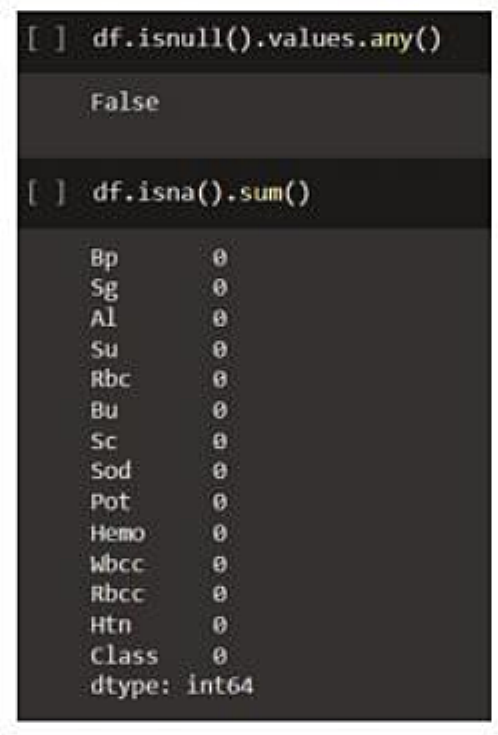


Name	Description	Type: unit/ values
Age (age)	Patient's age	Numeric: years
Blood pressure (bp)	Blood pressure of the patient	Numeric: mm/Hg
Specific gravity (sg)	The ratio of the density of urine	Nominal: 1.005, 1.010, 1.015, 1.020,1.025
Albumin (al)	Albumin level in the blood	Nominal: 0,1,2,3,4,5
Sugar (su)	Sugar level of the patient	Nominal: 0,1,2,3,4,5
Red blood cells (rbc)	Patients' red blood cells count	Nominal: normal, abnormal
Pus cell (pc)	pus cell count of patient	Nominal: normal, abnormal
Pus cell clumps (pcc)	Presence of pus cell clumps in the blood	Nominal: present, not present
Bacteria (ba)	Presence of bacteria in the blood	Nominal: present, not present
Blood glucose (bgr)	blood glucose random count	Numeric: mgs/dl
Blood urea (bu)	blood urea level of the patient	Numeric: mgs/dl
Serum creatinine (sc)	serum creatinine level in the blood	Numeric: mgs/dl
Sodium (sod)	sodium level in the blood	Numeric: mEq/L
Potassium (pot)	potassium level in the blood	Numeric: mEq/L
Hemoglobin (hemo)	hemoglobin level in the blood	Numeric: gms
Packed cell volume (pcv)	packed cell volume in the blood	Numeric
White blood cell count (wc)	white blood cell count of the patient	Numeric: cells/cumm
Red blood cell count (rc)	red blood cell count of the patient	Numeric millions/cmm
Hypertension (htn)	Does the patient has hypertension on not	Nominal: yes, no
Diabetes mellitus (dm)	Does the patient has diabetes or not	Nominal: yes, no
Coronary artery disease (cad)	Does the patient has coronary artery disease or not	Nominal: yes, no
Appetite (appet)	Patient's appetite	Nominal: good, poor
Pedal Edema (pe)	Does patient has pedal edema or not	Nominal: yes, no
Anemia (ane)	Does patient has anemia or not	Nominal: yes, no
Class	Does the patient has kidney disease or not	Nominal: CKD, not CKD

Table-1 Description of CKD Data

4.3 Data Preprocessing

Data Preprocessing. Prior to model building, data preprocessing is required to remove unwanted noise and outliers from the dataset that might cause the model to diverge from the proper training set. This stage tackles anything that is impeding the model's efficiency. After collecting the necessary data, it must be cleaned and prepared for model construction. The dataset is next searched for null values. However, this dataset contains no null values. Figure 3 shows there is no missing data available in this dataset. Here, the output values "False" and "0" indicate the absence of null values. After completing data preparation and handling the unbalanced dataset, the next step is to build the model. To increase the accuracy and efficiency of this task, the data is split into training and testing segments, with an 80/20 ratio of training to testing. Following the model's splitting, it is trained using a number of classification techniques. The classification methods used in this research include the decision tree classification method, K-nearest neighbor, and LR.



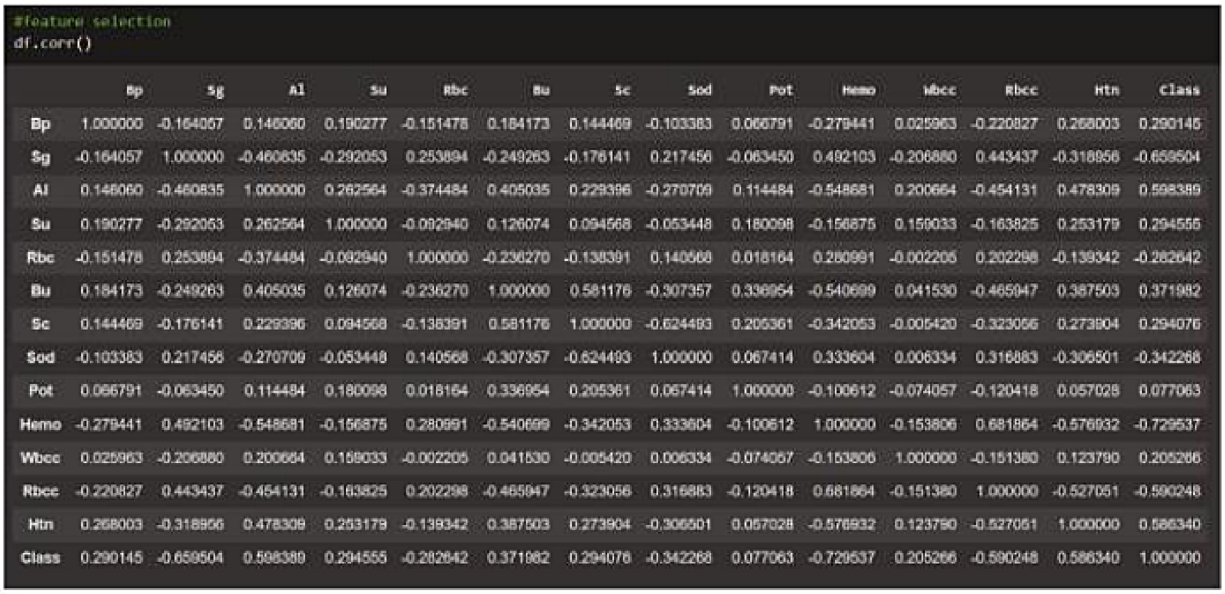
```
[ ]  df.isnull().values.any()

     False

[ ]  df.isna().sum()

     Bp       0
     Sg       0
     Al       0
     Su       0
     Rbc      0
     Bu       0
     Sc       0
     Sod      0
     Pot      0
     Hemo     0
     Wbcc     0
     Rbcc     0
     Htn      0
     Class    0
     dtype: int64
```

Figure-3 No Missing Data

4.4 Feature Selection

In the heat map, the absolute values of the correlations between features and the class label show that blood pressure, albumin, sugar, blood urea, serum, creatinine, potassium, white blood cell count, and hypertension all have positive links. Figures 4 and 5 show the feature correlation value and heat map, respectively. All the positively correlated features are considered for further prediction. Each albumin molecule has just five distinct sets of values. The quantity of albumin is assessed using a urine protein test.

A high protein level in the urine means that the filtration units in the kidneys have been damaged by disease, fever, or intense activity. Numerous tests should be performed over many weeks to establish the diagnosis. The term serum creatinine is used interchangeably with blood creatinine and creatinine. Creatinine is the byproduct of muscle breakdown of the chemical creatine. The kidneys eliminate creatinine from the body. This test is done to fifind out how much creatinine is in your blood. Creatine is an element of the metabolic cycle that produces the energy needed for muscle contraction. The body produces both creatine and creatinine at the same rate. Creatinine levels in the blood can rise due to a high-protein diet, congestive heart failure, diabetic issues, and dehydration, among other factors. Creatinine levels in women should be between 0.6 and 1.1 mg/dL, while those in males should be between 0.7 and 1.3 mg/dL. Additionally, hypertension, or high blood pressure, develops whenever blood pressure against the walls of blood vessels rises. Hyper tension can lead to heart attacks, strokes, and chronic kidney disease if it is not treated or managed properly. Nonetheless, CKD may result in hypertension.



```
#feature selection
df.corr()
```

	Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn	Class
Bp	1.000000	-0.164057	0.146060	0.190277	-0.151478	0.184173	0.144469	-0.103383	0.066791	-0.279441	0.025963	-0.220827	0.268003	0.290145
Sg	-0.164057	1.000000	-0.460835	-0.292053	0.253894	-0.249263	-0.176141	0.217456	-0.063450	0.492103	-0.206880	0.443437	-0.318956	-0.659504
Al	0.146060	-0.460835	1.000000	0.262564	-0.374484	0.405035	0.229396	-0.270709	0.114484	-0.548681	0.200664	-0.454131	0.478309	0.598389
Su	0.190277	-0.292053	0.262564	1.000000	-0.092940	0.126074	0.094568	-0.053448	0.180098	-0.156875	0.159033	-0.163825	0.253179	0.294555
Rbc	-0.151478	0.253894	-0.374484	-0.092940	1.000000	-0.236270	-0.138391	0.140568	0.018164	0.280991	-0.002205	0.202298	-0.139342	-0.282642
Bu	0.184173	-0.249263	0.405035	0.126074	-0.236270	1.000000	0.581176	-0.307357	0.336954	-0.540699	0.041530	-0.465947	0.387503	0.371982
Sc	0.144469	-0.176141	0.229396	0.094568	-0.138391	0.581176	1.000000	-0.624493	0.205361	-0.342053	-0.005420	-0.323056	0.273904	0.294076
Sod	-0.103383	0.217456	-0.270709	-0.053448	0.140568	-0.307357	-0.624493	1.000000	0.067414	0.333604	0.006334	0.316883	-0.306501	-0.342268
Pot	0.066791	-0.063450	0.114484	0.180098	0.018164	0.336954	0.205361	0.067414	1.000000	-0.100612	-0.074057	-0.120418	0.057028	0.077063
Hemo	-0.279441	0.492103	-0.548681	-0.156875	0.280991	-0.540699	-0.342053	0.333604	-0.100612	1.000000	-0.153806	0.681864	-0.576932	-0.729537
Wbcc	0.025963	-0.206880	0.200664	0.159033	-0.002205	0.041530	-0.005420	0.006334	-0.074057	-0.153806	1.000000	-0.151380	0.123790	0.205266
Rbcc	-0.220827	0.443437	-0.454131	-0.163825	0.202298	-0.465947	-0.323056	0.316883	-0.120418	0.681864	-0.151380	1.000000	-0.527051	-0.590248
Htn	0.268003	-0.318956	0.478309	0.253179	-0.139342	0.387503	0.273904	-0.306501	0.057028	-0.576932	0.123790	-0.527051	1.000000	0.586340
Class	0.290145	-0.659504	0.598389	0.294555	-0.282642	0.371982	0.294076	-0.342268	0.077063	-0.729537	0.205266	-0.590248	0.586340	1.000000

Figure-4 Correlation Feature Values

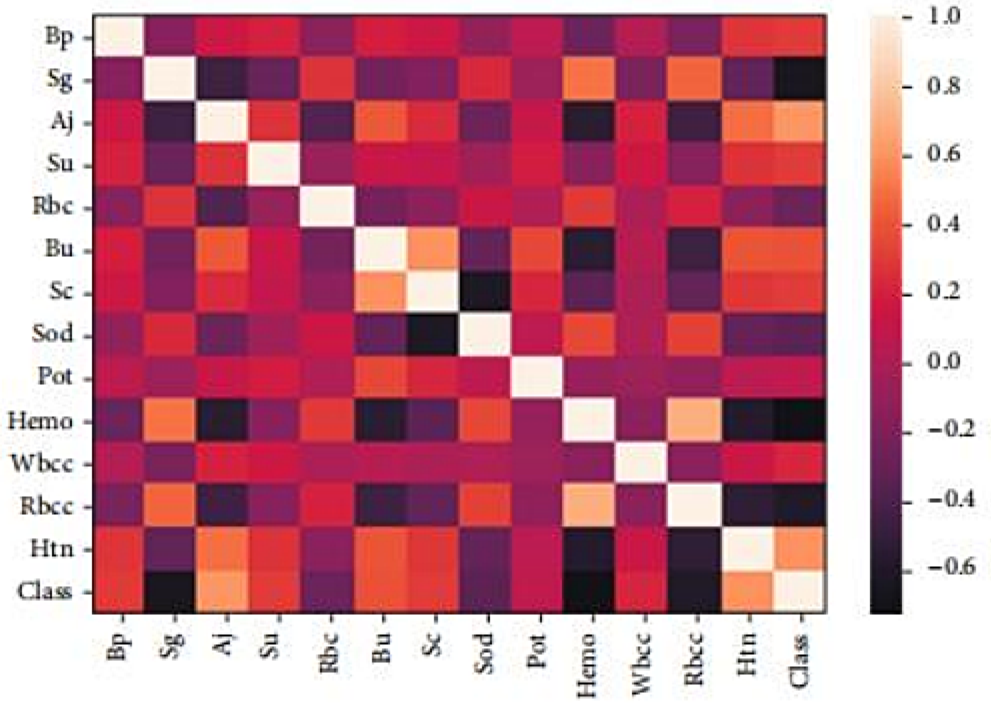


Figure-5 Heat Map

4.5 Algorithms

The following machine learning algorithms have been used to predict chronic kidney disease.

1. K – Nearest Neighbor
2. Logistic Regression

4.5.1 K-Nearest Neighbor

Figure-(6) depicts the whole KNN model's flowchart. One of the simplest ML algorithms is KNN, which uses the supervised learning approach. A new case is assigned to a category based on how closely it resembles prior categories. This is known as the KNN technique. With the KNN method, you can store all the data you have and then classify new data based on how similar it is to the old. This suggests that the KNN technique can rapidly classify new data into well-defined categories. Though it is often utilized for classification problems, the KNN method may be used for regression as well. There are no data assumptions made by the KNN technique, which is nonparametric and also called a "lazy learner algorithm," since it does not instantly learn from the training set but rather keeps and categorizes the data for later. If it receives new data, the KNN classifies it into a category that is quite close to the new data that was stored during training. For classification, one of the most frequently used ML methods is the K-nearest neighbor classifier. The nonparametric slow learning approach, K-nearest neighbor, may be used to categorize data. This classifier sorts objects according to how far they are from each other and how close they are. It prioritizes the immediate surroundings of the item above, the dissemination of essential information.

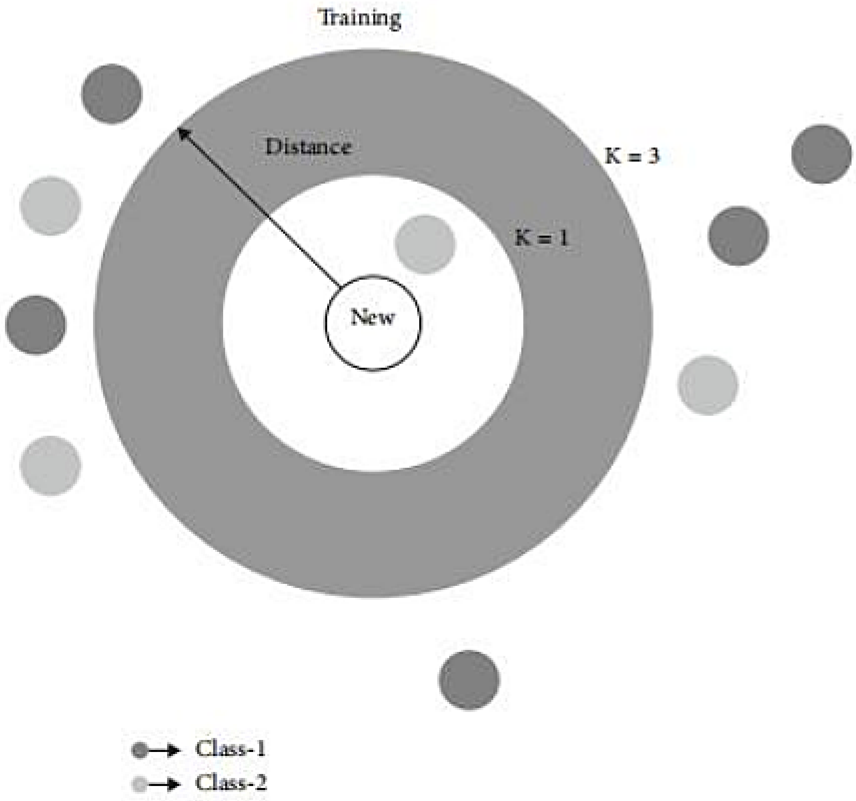


Figure-6 KNN classifier working procedure

4.5.2 Logistic Regression

Binary outcomes are modeled using the statistical method of logistic regression, which is well known in the field. Different learning methods are used to execute logistic regression in statistical research. A variant of the neural network method was used to create the LR algorithm. This method resembles neural networks in many ways, but it is simpler to set up and use. Figure-(7) shows the block diagram of LR. Utilizing logistic regression, the output of a categorical dependent variable is predicted. So, the output must be discrete or categorical. It may be yes or no, 0 or 1, true or false, etc., but probability values between 0 and 1 are given. Logistic regression and linear regression are used in very similar ways. Classification problems are addressed with logistic regression, and regression problems are addressed using linear regression. Instead of a regression line, we use an “S” shaped logistic function that predicts two maximum values (0 or 1). The logistic function’s curve indicates the probability of anything, such as whether cells are malignant or not, or if an animal is fat or not. Since it can classify new data using both discrete and continuous datasets, logistic regression is a common ML technique.

Binary outcomes are modeled using the statistical method of logistic regression, which is well known in the field. Different learning methods are used to execute logistic regression in statistical research. A variant of the neural network method was used to create the LR algorithm. This method resembles neural networks in many ways, but it is simpler to set up and use. Figure-(7) shows the block diagram of LR. Utilizing logistic regression, the output of a categorical dependent variable is predicted. So, the output must be discrete or categorical. It may be yes or no, 0 or 1, true or false, etc., but probability values between 0 and 1 are given. Logistic regression and linear regression are used in very similar ways. Classification problems are addressed with logistic regression, and regression problems are addressed using linear regression. Instead of a regression line, we use an “S” shaped logistic function that predicts two maximum values (0 or 1). The logistic function’s curve indicates the probability of anything, such as whether cells are malignant or not, or if an animal is fat or not. Since it can classify new data using both discrete and continuous datasets, logistic regression is a common ML technique.

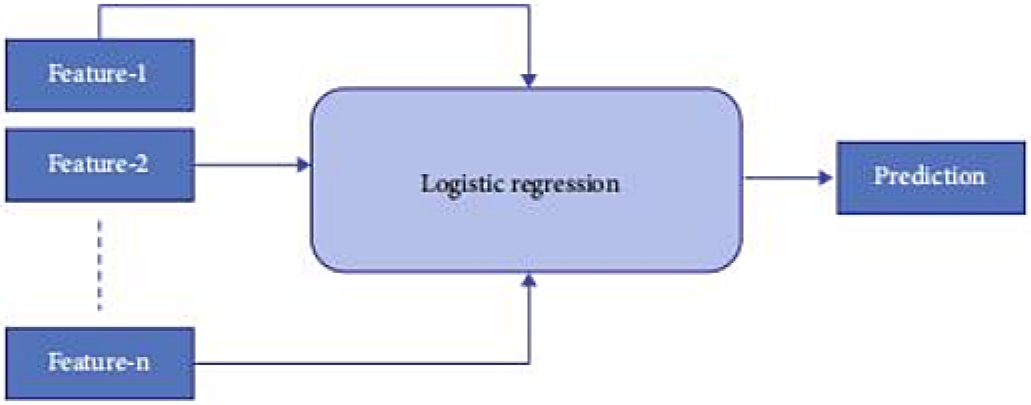
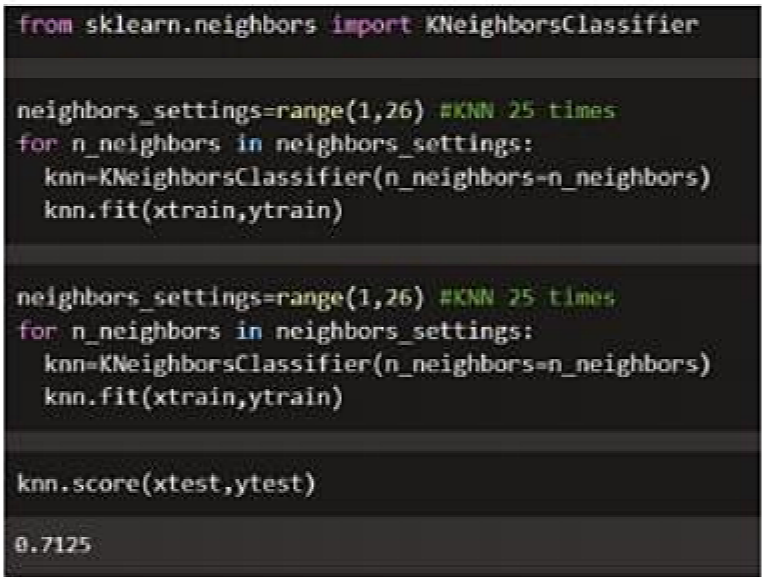


Figure-7 Block Diagram of LR Classifier

5. RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

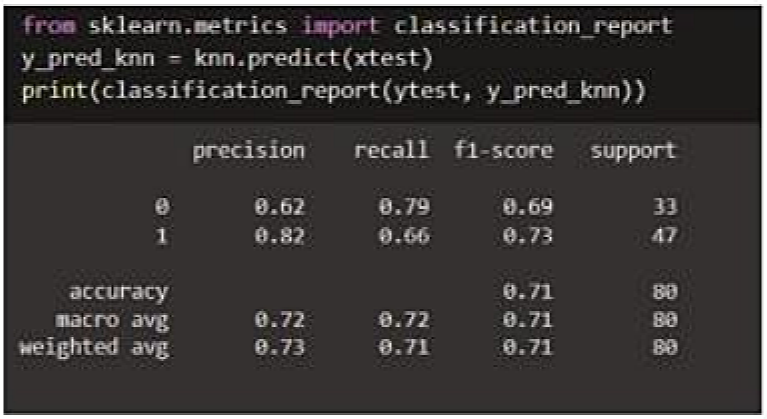
5.1 K-Nearest Neighbor

Figure–(8) shows the K-NN classifier’s classification accuracy. Here, the
accuracy is lower than with other algorithms. Even after fine tuning, this accuracy did not improve further. Figure –(9) shows the classification report of the KNN algorithm. The overall performance of KNN is unsatisfactory. The overall r2-score obtained here is 71%. Individual r2-scores for non-CKD are 69 percent and 73 percent for CKD. Figure-(10) depicts the KNN classifier’s AUC curve. The KNN classifier has a 73 percent accuracy under the curve. As well as the model’s calculated performance, the confusion matrix displays the predicted outcome. There were 57 accurate predictions, but there were also 23 erroneous predictions



```
from sklearn.neighbors import KNeighborsClassifier

neighbors_settings=range(1,26) #KNN 25 times
for n_neighbors in neighbors_settings:
  knn=KNeighborsClassifier(n_neighbors=n_neighbors)
  knn.fit(xtrain,ytrain)

neighbors_settings=range(1,26) #KNN 25 times
for n_neighbors in neighbors_settings:
  knn=KNeighborsClassifier(n_neighbors=n_neighbors)
  knn.fit(xtrain,ytrain)

knn.score(xtest,ytest)

0.7125
```

Figure-8 Accuracy KNN Classifier



```
from sklearn.metrics import classification_report
y_pred_knn = knn.predict(xtest)
print(classification_report(ytest, y_pred_knn))

              precision    recall  f1-score   support

           0       0.62      0.79      0.69        33
           1       0.82      0.66      0.73        47

    accuracy                           0.71        80
   macro avg       0.72      0.72      0.71        80
weighted avg       0.73      0.71      0.71        80
```

Figure-9 Classification Report of KNN Classifier

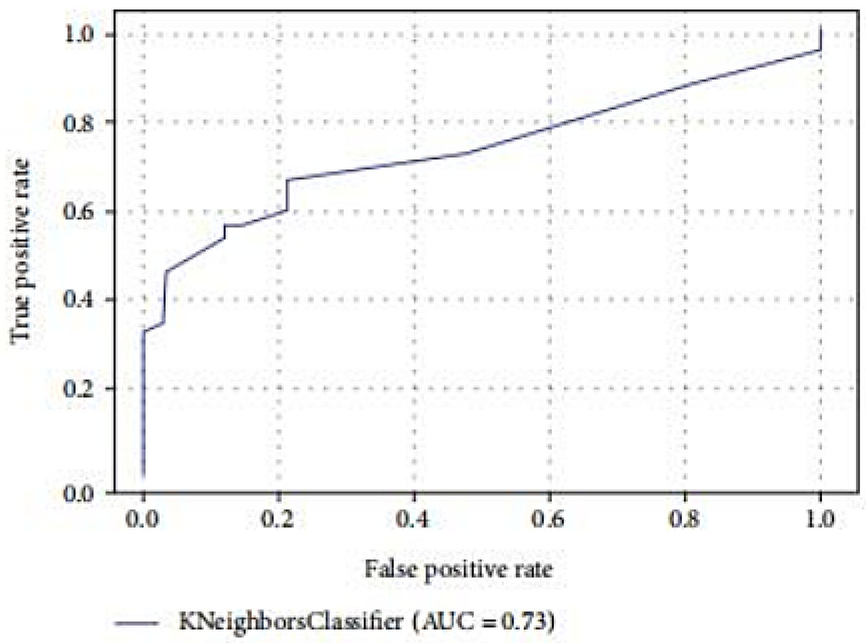
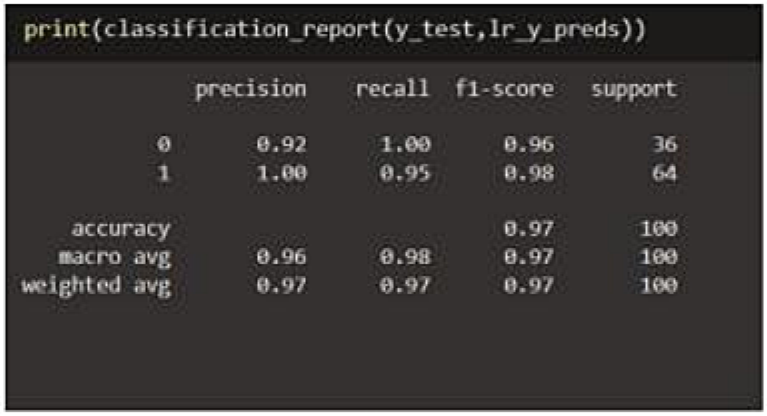


Figure-10 AUC Curve of KNN Classifier

5.2 Logistic Regression

The report of the LR model is shown in Figure 11. This model has the best accuracy in classifying objects. In this case, the total F1-score obtained is percent. Individuals without CKD have an r2-score of 96 percent, whereas those with CKD have a score of 98 percent. The AUC curve for the logistic regression classifier is shown in Figure 12. The accuracy under the curve is 100 percent in this case. Figure 13 shows the final prediction of the logistic regression model. As well as the model's calculated performance, the confusion matrix displays the predicted outcome. There were 97 correct forecasts and three incorrect forecasts, for a total 16accuracy of 100 percent.



```
print(classification_report(y_test,lr_y_preds))

              precision    recall  f1-score   support

           0       0.92      1.00      0.96        36
           1       1.00      0.95      0.98        64

    accuracy                           0.97       100
   macro avg       0.96      0.98      0.97       100
weighted avg       0.97      0.97      0.97       100
```

Figure-11 Logistic Regression Classification Report

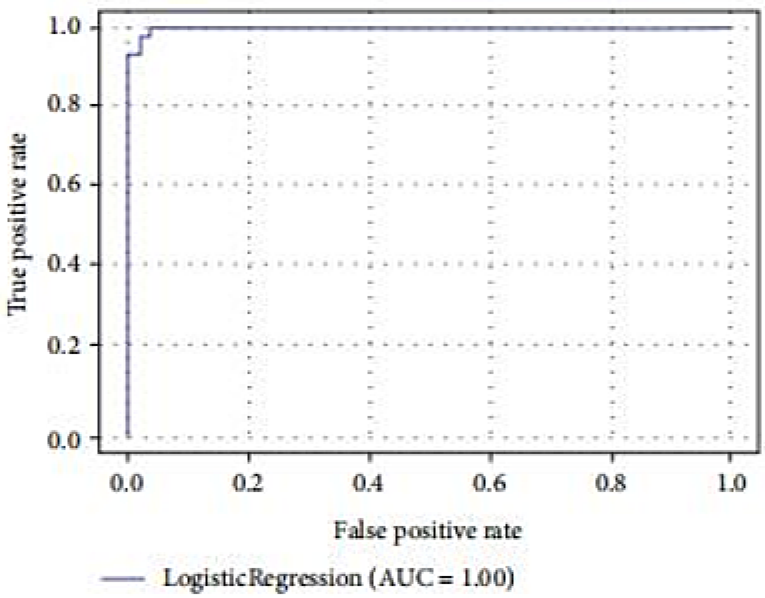


Figure-12 Logistic Regression AUC Curve

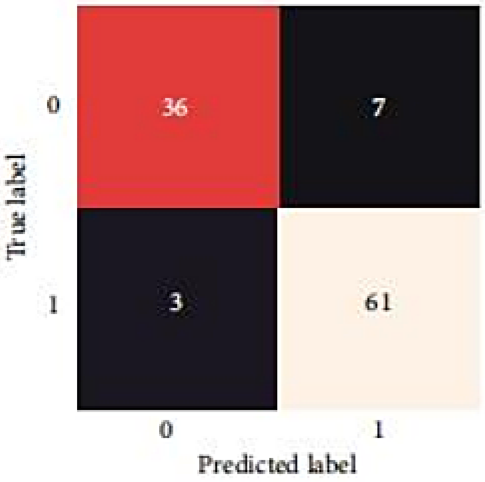
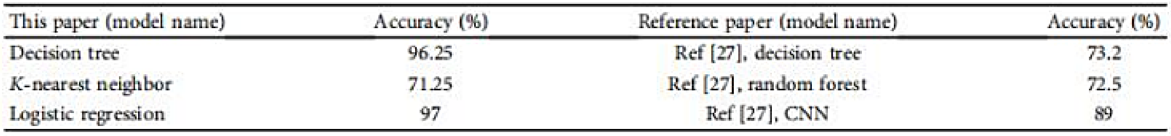


Figure-13 Logistic Regression Confusion Matrix

5.3 Model Comparison



This paper (model name)	Accuracy (%)	Reference paper (model name)	Accuracy (%)
Decision tree	96.25	Ref [27], decision tree	73.2
K-nearest neighbor	71.25	Ref [27], random forest	72.5
Logistic regression	97	Ref [27], CNN	89

Figure-14

LR Model got most accuracy in predicting output results for Prediction of CKD.

6. SUMMARY AND CONCLUSIONS

According to the findings of the study, the decision tree approach and logistic regression can be used to predict chronic kidney disease more accurately. According to the study, their precision was 96.25 percent, and their accuracy was 97 percent. Compared to prior research, the accuracy percent of the models used in this investigation is considerably higher, indicating that the models used in this study are more reliable than those used in previous studies. When cross validation measurements are used in the prediction of chronic kidney disease, the LR method outperforms the other processes. Future research may build on this work by developing a web application that incorporates these algorithms and using a bigger dataset than the one utilized in this study. This will aid in the achievement of improved out comes as well as the accuracy and efficiency with which healthcare practitioners can anticipate kidney issues. This will enhance the dependability of the framework as well as the framework's presentation. The hope is that it would encourage people to seek early treatment for chronic renal disease and to make improvements in their lives

Data Availability

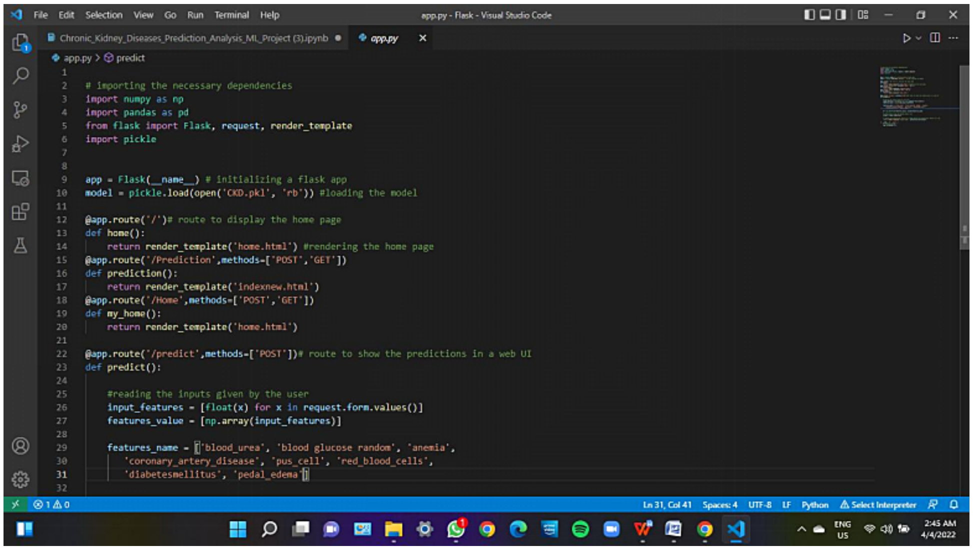
The data utilized to support these research findings is accessible online at https://www.kaggle.com/datasets/mansoordaku/ckdisease.

References

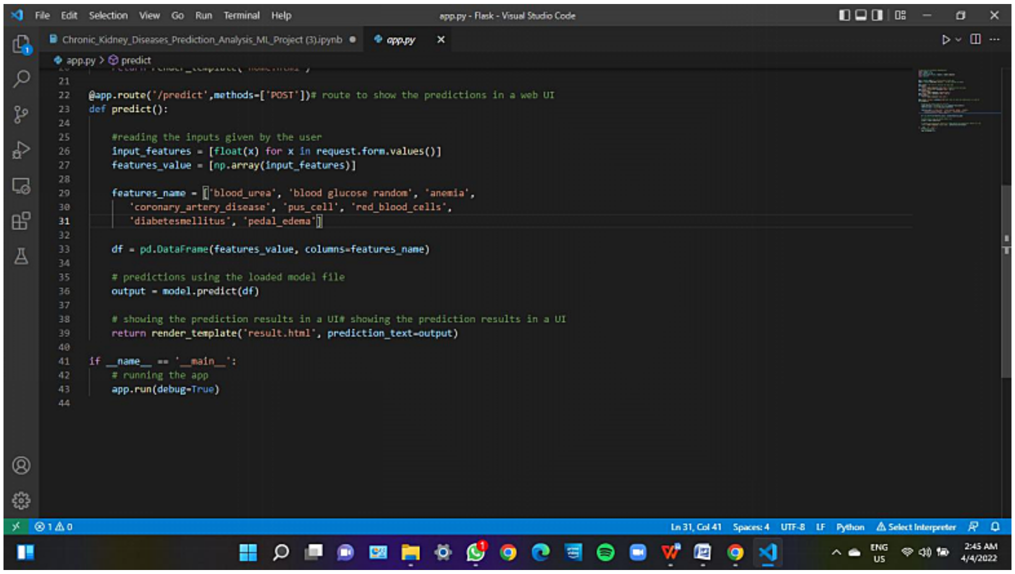
- All Information is collected from the Internet sources like Google…

Appendix

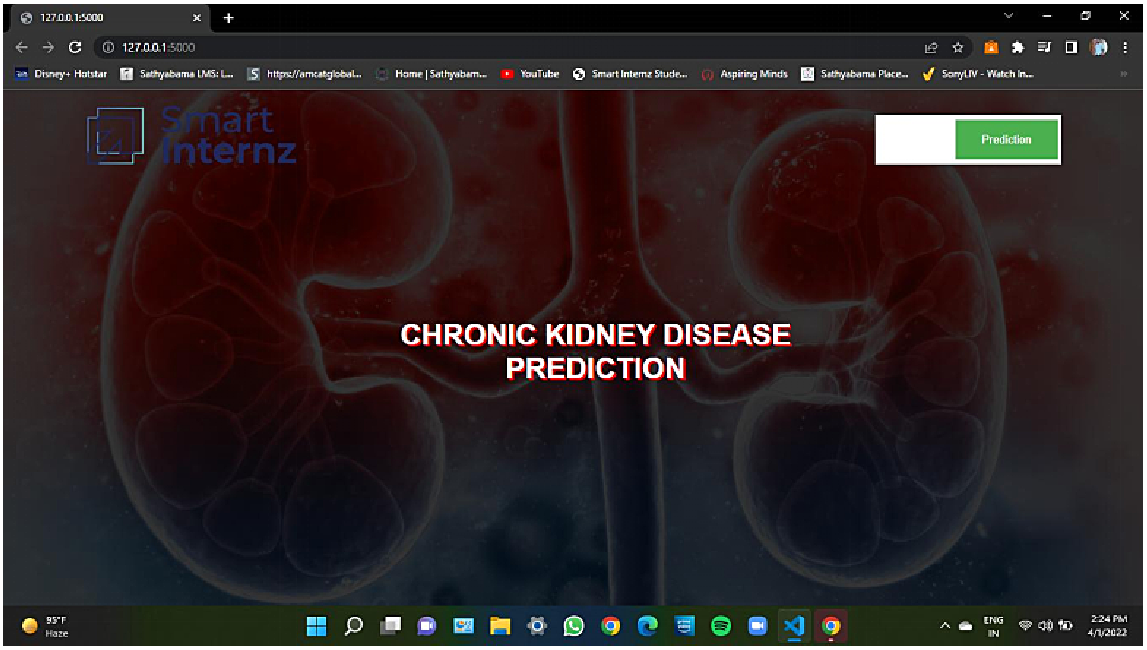
Main Code & Output Screenshots


File Edit Selection View Go Run Terminal Help
app.py - Flask - Visual Studio Code
Chronic_Kidney_Diseases_Prediction_Analysis_ML_Project (3).ipynb
app.py
app.py > predict
Ln 31, Col 41 Spaces: 4 UTF-8 LF Python Select Interpreter
ENG US
2:45 AM 4/4/2022

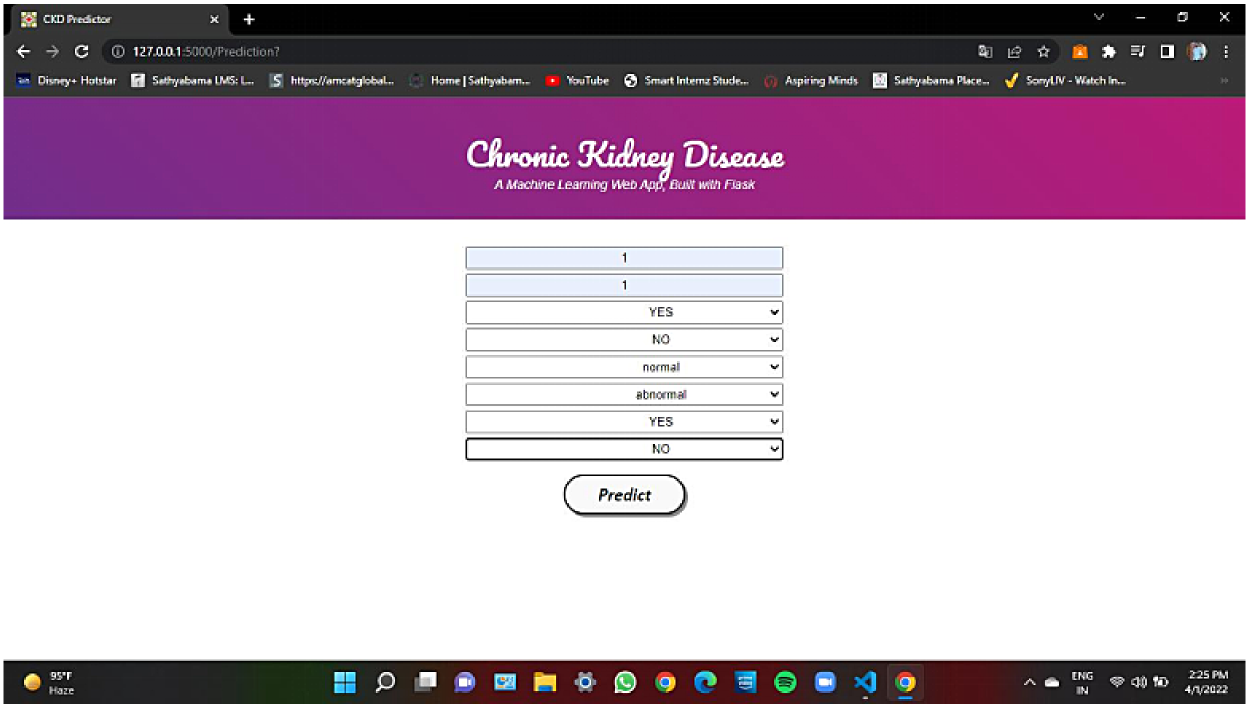
```python

# importing the necessary dependencies
import numpy as np
import pandas as pd
from flask import Flask, request, render_template
import pickle


app = Flask(__name__) # initializing a flask app
model = pickle.load(open('CKD.pkl', 'rb')) #loading the model

@app.route('/')# route to display the home page
def home():
    return render_template('home.html') #rendering the home page
@app.route('/Prediction',methods=['POST','GET'])
def prediction():
    return render_template('indexnew.html')
@app.route('/Home',methods=['POST','GET'])
def my_home():
    return render_template('home.html')

@app.route('/predict',methods=['POST'])# route to show the predictions in a web UI
def predict():

    #reading the inputs given by the user
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]

    features_name = ['blood_urea', 'blood glucose random', 'anemia',
       'coronary_artery_disease', 'pus_cell', 'red_blood_cells',
       'diabetesmellitus', 'pedal_edema']

```


File Edit Selection View Go Run Terminal Help
app.py - Flask - Visual Studio Code
Chronic_Kidney_Diseases_Prediction_Analysis_ML_Project (3).ipynb
app.py
app.py > predict
Ln 31, Col 41 Spaces: 4 UTF-8 LF Python Select Interpreter
ENG US
2:45 AM 4/4/2022

```python

@app.route('/predict',methods=['POST'])# route to show the predictions in a web UI
def predict():

    #reading the inputs given by the user
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]

    features_name = ['blood_urea', 'blood glucose random', 'anemia',
       'coronary_artery_disease', 'pus_cell', 'red_blood_cells',
       'diabetesmellitus', 'pedal_edema']

    df = pd.DataFrame(features_value, columns=features_name)

    # predictions using the loaded model file
    output = model.predict(df)

    # showing the prediction results in a UI# showing the prediction results in a UI
    return render_template('result.html', prediction_text=output)

if __name__ == '__main__':
    # running the app
    app.run(debug=True)

```



CHRONIC KIDNEY DISEASE PREDICTION



Chronic Kidney Disease

A Machine Learning Web App, Built with Flask

1

1

YES

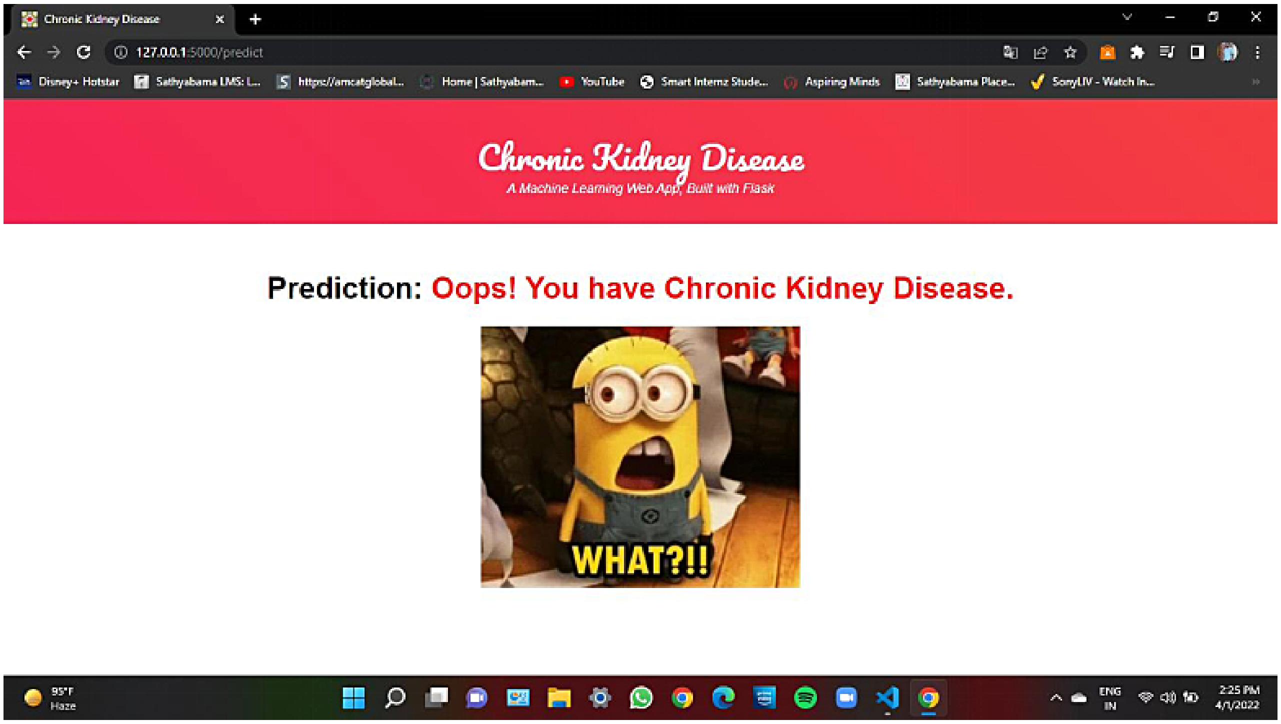
NO

normal

abnormal

YES

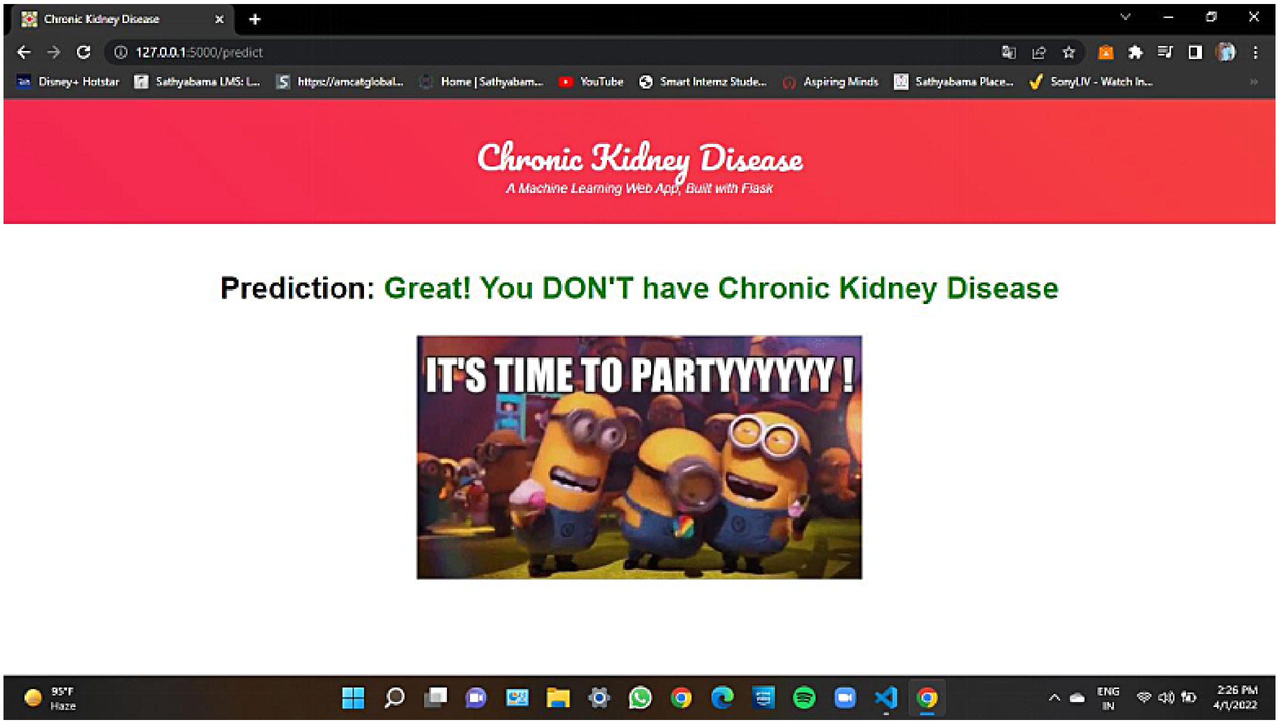
NO



Chronic Kidney Disease

A Machine Learning Web App, Built with Flask

Prediction: Oops! You have Chronic Kidney Disease.



Chronic Kidney Disease

A Machine Learning Web App, Built with Flask

Prediction: Great! You DON'T have Chronic Kidney Disease