PREDICTION OF STATUS OF CHRONIC KIDNEY DISEASE

Manikant Revankar[1], Rohith D[1], Sarthak Dixit[1], Sidharth Kumar[1]

Mr. Rajeev Bilagi, Assistant Professor[1]

[1]Acharya Institutes of Technology, Department of Computer Science and Engineering, Acharya Doctor Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

[1]rajeev@acharya.ac.in

[2]manikantd.19.becs@acharya.ac.in

[3]drohith7070@gmail.com

[4]sarthaks.19.becs@acharya.ac.in

[5]sidharths.19.becs@acharya.ac.in

Abstract —A major medical problem affecting millions of individuals globally is chronic kidney disease (CKD). According to estimates, one in ten persons worldwide have CKD, and the frequency is rising. Early detection and treatment of CKD are essential for improving patient outcomes and lowering healthcare costs because it is a progressive disease that frequently remains unnoticed until it reaches late stages. Early CKD diagnosis and management can greatly enhance patient outcomes and lower healthcare expenditures. In this project, the objective is to want to use machine learning to create a CKD predictive model. The most crucial CKD predictors were found using a variety of feature selection and modelling techniques on a dataset of patient records that included demographic data, test findings, and medical history.

I. INTRODUCTION

With a high prevalence of morbidity and mortality, chronic kidney disease (CKD) is an international medical problem that also causes other illnesses. Since no overt symptoms exist Patients frequently miss signs of chronic disease in the early stages. Early disease detection enables patients to get medical therapy in a timely manner to slow the disease's progression. With the right methodology, machine learning models may efficiently assist the patients. Due to the rising prevalence of CKD around the world, which encompasses illnesses that steadily harm the kidneys, CKD is one amongst the most important health problems. The goal of this project is to create a machine learning model that, using patient data, can promptly and accurately identify the stage of CKD. We can determine the most crucial CKD variables and develop a predictive model that can assist healthcare professionals in making wise patient care decisions by utilizing the power of machine learning algorithms. The project comes under the machine learning domain and the methodologies related to the same is used in the implementation of the proposed model. The CKD prediction project involves the use of several technical areas, including data pre-processing, feature selection, machine learning algorithms, and model evaluation. Data pre-processing involves cleaning and transforming raw data to ensure that it is consistent, complete, and ready for analysis. In this project, the performing of data pre-processing tasks such as handling missing data, and normalizing data were done. Feature selection involves selecting the most important variables from the dataset that are relevant for predicting CKD. This is important to reduce the complexity of the model, improve its accuracy, and prevent overfitting. The creation of a predictive model for CKD involves the application of machine learning methods. To find the machine learning technique that performed the best at predicting CKD, a testing of number of them in this study is done, including logistic regression, decision trees, random forests, and support vector machines (SVMs).

To create a machine learning model that could accurately and quickly predict the state of CKD based on patient data, improving early illness detection and enabling prompt intervention to stop or halt the disease's progression. To appropriately categorize patients with CKD or not, the model will use patient data, including demographic details, clinical data, and laboratory results.

The selection and weighting of features can significantly impact the model's predictive performance.

RELATED WORKS

The report of paper [1] is the result of two KDIGO Controversies Conferences held in 2004 and 2006. The conferences brought together experts and stakeholders in the field of chronic kidney disease (CKD) to discuss various aspects related to CKD, including its definition and classification, screening and surveillance, public policy, and its associations with cardiovascular disease, chronic

infections, and cancer. The paper titled [3] "Chronic kidney disease as a global public health problem: approaches and initiatives" describes the use of four machine learning algorithms for predicting chronic kidney disease (CKD) and improving accuracy. The algorithms utilized in the study are Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest, and Decision Tree. The dataset employed for training the algorithms is obtained from the UCI repository. The paper introduces the Adaptive Hybridized Deep Convolutional Neural Network (AHDCNN) as a tool for early prediction and diagnosis of Chronic Kidney Disease (CKD). The focus is on using deep learning techniques to identify specific types of lesions from CT images in renal cancer cases. The collected data is analyzed, and missing values are replaced with median value estimates. Various features associated with kidney disease are extracted from the noise-free data and inputted into a classifier to detect variations in kidney patterns. The system trains the features in each hidden layer by measuring weight and bias values. The trained features are further refined by multiple layers of a deep-belief network, enabling the recognition of irregular patterns. The paper presents a machine learning methodology for diagnosing Chronic Kidney Disease (CKD) using a dataset obtained from the University of California Irvine (UCI) machine learning repository. The dataset contains missing values, which were imputed using the K-Nearest Neighbor (KNN) imputation method. Six machine learning algorithms, including logistic regression, random forest, support vector machine, k-nearest neighbor, naive Bayes classifier, and feed-forward neural network, were employed to establish models. Among these models, random forest achieved the highest accuracy of 99.75% in diagnosing CKD. By analyzing the misjudgments made by the established models, an integrated model combining logistic regression and random forest using a perceptron was proposed. This integrated model achieved an average accuracy of 99.83% after ten simulations.

PROPOSED METHODOLOGY

In this system, The Model will take in the dataset for collection and processing further into the cleaning phase where it has been cleaned and removing unwanted data. It is then undergoing Feature encoding so as to select the necessary features for to feed in to the algorithm to make predictions. The model is then expected to predict accurately the results and further been monitored and updated.

Flow chart

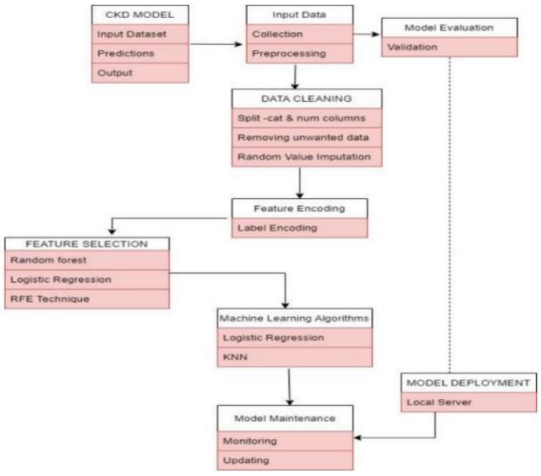

Fig 1. Design Dataflow

The above diagram shows the flow chart of the CKD Prediction model. The first step is to collect data related to various attributes of patients from various sources such as hospitals, clinics, and research studies. The data is then preprocessed to clean and transform it into a format that is suitable for analysis. The data is then split into categorical and numerical variables, and irrelevant data that does not contribute to the prediction model is removed. Missing values in the dataset are filled using random value imputation, and categorical variables are converted into numerical values using label encoding. Two popular machine learning algorithms used for predicting CKD are Random Forest and Logistic Regression. Random forest is an ensemble learning algorithm that builds multiple decision trees and combines their predictions to generate the final output, while logistic regression is a statistical method that uses a logistic function to model a binary dependent variable. Once the model is built, it is essential to validate it by testing it on a separate set of data that was not used during the training phase. The model also needs to be monitored and updated regularly to ensure that it continues to provide accurate predictions as new data becomes available.

IMPLEMENTATION

A. Data Cleaning

Data cleaning is essential for the chronic kidney disease (CKD) prediction model to produce accurate and trustworthy predictions. The act of cleaning the data makes sure that it is complete, accurate, and consistent. After being cleaned, the data can be pre-processed to create a format that is appropriate for analysis and modelling. The data cleaning module aims to achieve the functional requirement of ensuring that the dataset used for building the CKD prediction model is of high quality, consistent, and error-free. This is important because the quality of the input data significantly affects the accuracy and reliability of the model's predictions. In fig 2, we have replaced the unwanted particles in the data with cleaned data. "\t" is removed in each of these characteristics which contains it. It showcases the cleaned data with the values free of '\t'.

```
In [13]: dataset['diabetes_mellitus'].replace(to_replace = {'\tno':'no','\tyes':'yes',' yes':'yes'},inplace=True)
         dataset['coronary_artery_disease'] = dataset['coronary_artery_disease'].replace(to_replace = '\tno', value='no')
         dataset['class'] = dataset['class'].replace(to_replace = {'ckd\t': 'ckd', 'notckd': 'not ckd'})

In [14]: dataset['class'] = dataset['class'].map({'ckd': 0, 'not ckd': 1})
         dataset['class'] = pd.to_numeric(dataset['class'], errors='coerce')

In [15]: cols = ['diabetes_mellitus', 'coronary_artery_disease', 'class']

         for col in cols:
             print(f"{col} has {dataset[col].unique()} values\n")

         diabetes_mellitus has ['yes' 'no' nan] values

         coronary_artery_disease has ['no' 'yes' nan] values

         class has [0 1] values
```

Fig 2. Data Cleaning

B. Data Visualization

The objectives of the data visualization module in the CKD prediction project are to gain insights and understanding of the dataset, identify patterns and relationships among variables. The module involves creating graphical representations of the dataset using various visualization techniques, such as histograms, scatter plots, bar charts, and heat maps. Fig 3 indicates the heatmap of the

prediction model in machine learning and it generally indicates the correlation between the various features or input variables in the dataset. show how the different features are related to each other in terms of their correlation coefficient values. A high positive correlation between two features indicates that they are strongly correlated, while a high negative correlation indicates that they are inversely correlated.

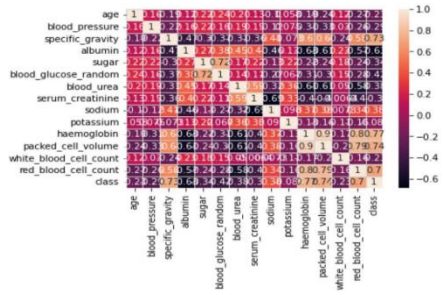

Fig 3. Data Visualization

C. DATA PREPROCESSING

At this phase Random value imputation is used to fill in missing values in a dataset with random values drawn from the same distribution as the existing values in that column. This technique is commonly used when a small number of values are missing in the dataset, and we don't have any prior knowledge about the missing values. Random value imputation is relatively simple to implement and is a fast way to fill in missing values. The Fig 4 shows the output shows that there are no missing or null values left in the numerical columns of the dataset after applying the random value imputation technique. The code is iterating over all numerical columns of the dataset, and for each column, the random_value_imputation() function is called. Label encoding is done, to do feature encoding in achieving the functional requirement of converting categorical data to numerical data. It assigns a unique numerical label to each category in the data.

```
def random_value_imputation(feature):
    random_sample = dataset[feature].dropna().sample(dataset[feature].isna().sum())
    random_sample.index = dataset[dataset[feature].isnull()].index
    dataset.loc[dataset[feature].isnull(), feature] = random_sample

def impute_mode(feature):
    mode = dataset[feature].mode()[0]
    dataset[feature] = dataset[feature].fillna(mode)
```

```
for col in num_cols:
    random_value_imputation(col)
dataset[num_cols].isnull().sum()
```

```
age                     0
blood_pressure          0
specific_gravity        0
albumin                 0
sugar                   0
blood_glucose_random    0
blood_urea              0
serum_creatinine        0
sodium                  0
potassium               0
haemoglobin             0
packed_cell_volume      0
white_blood_cell_count  0
red_blood_cell_count    0
class                   0
dtype: int64
```

Fig 4. Random Value Imputation

D. MODEL BUILDING/ALGORITHMS

The building of a model requires one last step and that is to select the algorithm so as to train. This model is tested with different algorithms to study how the model performs with different algorithms under the influence of a dataset. The interface is then made using with the model giving higher accuracy. Model building is a crucial step in developing this project. It involves creating a model using the data available that can be used to make predictions or classify new data based on patterns learned from the available data.

- LOGISTIC REGRESSION: It models the probability of a binary outcome (i.e., 0 or 1) by transforming the output of a linear regression model using the sigmoid function. The algorithm Logistic Regression and it is found out the following results: The accuracy score for training data is 0.9219. Similarly, the accuracy score for testing data is 0.9250

- SUPPORT VECTOR MACHINES: It is an algorithm used in machine learning for both classification and regression tasks. In the training results, an accuracy score of 0.6188 was found, which is not good. The classification report shows that the precision and recall scores for the second class are 0.0, indicating that the algorithm failed to predict any instance in the second class correctly. In the testing results, an accuracy score of 0.65 is found, which is not good either. The algorithm performed the same way on the testing dataset as it did on the training dataset.

- DECISION TREE: Decision Tree is a supervised learning algorithm used for solving classification and regression problems. The training results are shown as accuracy score is 1.0, indicating that the model correctly classified all the samples in the training dataset. The testing results are shown next, which include the confusion matrix, accuracy score, and F1-score. The accuracy score is 0.9875, indicating that the model correctly classified 98.75% of the samples in the testing dataset. The F1-score is also high, indicating good performance of the model.

- RANDOM FOREST CLASSIFIER: The accuracy score for the training set is 0.9844, which means that the model correctly classified 98.44% of the instances in the training set. The classification report shows the precision, recall, and F1-score for each class (0 and 1), as well as the accuracy, macro average, and weighted average. For the testing set, the confusion matrix shows that all 52 true negatives and 28 true positives were predicted correctly. The accuracy score for the testing set is 1.0000, which means that the model correctly classified all instances in the testing set.

- GRADIENT BOOST ALGORITHM: Gradient Boost Algorithm is another ensemble method that combines several weak learners to form a strong learner. The training data has been fit to the model, resulting in an

accuracy score of 1.0, which indicates that the model has correctly classified all of the observations in the training data. The testing data has also been evaluated using the same model, resulting in an accuracy score of 0.9875. The confusion matrix for the testing data shows that there was one false positive and no false negatives. The precision, recall, and f1-score for both classes are 1.0, which indicates that the model's predictions are perfect for both classes.

- XGBOOST CLASSIFIER: The accuracy score for the training set is 0.9688, which indicates that the model has achieved a high level of accuracy. The accuracy score for the testing set is 0.9875, which indicates that the model has achieved a high level of accuracy on the unseen data as well. The confusion matrix in the test results reveals that, of the 51 items in the testing set for class 0, 50 were correctly classified and 1, incorrectly, as class 1. In a test set for class 1, there were 28 samples, and each of them was correctly classified.

- ADABOOST CLASSIFIER: AdaBoost (Adaptive Boosting) is a machine learning algorithm that is used for classification tasks. The classification report confirms this perfect performance, with precision, recall, and F1-score all equal to 1 for both classes. On the testing data, the model achieved an accuracy score of 0.9875, which indicates that the model performs very well on unseen data.

- ENSEMBLE MODEL: Ensemble refers to the technique of combining multiple machine learning models to improve the overall performance of the system. For the training data, the ensemble model achieved an accuracy of 0.9906, with a confusion matrix showing that out of 320 samples, there were 195 true negatives, 122 true positives, 3 false negatives, and 0 false positives. For the testing data, the ensemble model achieved a perfect accuracy of 1.0, with a confusion matrix showing that out of 80 samples, there were 52 true negatives and 28 true positives.

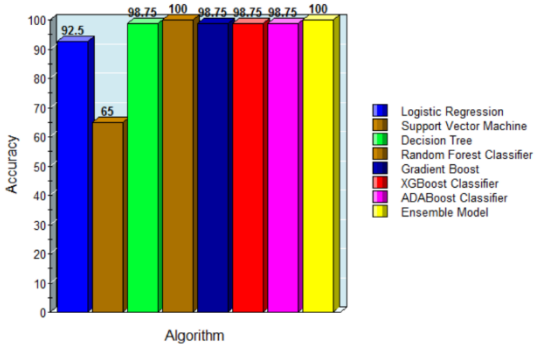

Fig 5. Accuracy Graph

REQUIREMENTS

A. Hardware Requirements

The minimum hardware requirements to run the CKD prediction model would be

1. CPU

A modern processor with at least 2 cores (e.g., Intel Core i3 or AMD Ryzen 3). A modern processor with at least 2 cores (e.g., Intel Core i3 or AMD Ryzen 3) is essential for this project as it can significantly improve the speed and efficiency of the training and testing process. Machine learning algorithms require a lot of computation, especially when working with large datasets, complex models, or deep learning architectures Having a processor with at least 2 cores allows for parallel processing, which means that multiple tasks can be executed simultaneously, speeding up the overall computation time.

2. RAM

RAM: At least 4 GB of RAM. Having at least 4 GB of RAM is essential for a smooth and efficient computing experience, especially when working with memory-intensive applications like machine learning. RAM, or Random Access Memory, is a type of volatile memory that is used by the computer to store and access data temporarily while it is running. It is different from permanent storage devices like hard drives or solid-state drives, which are used to store data permanently. Having sufficient RAM is crucial as it can significantly impact the performance and speed of the algorithms.

3. STORAGE

Having at least 20 GB of free disk space is necessary for storing the software, datasets, and models in a machine learning project. Disk space refers to the amount of storage available on the computer's hard drive or other storage devices like solid-state drives (SSDs) or external hard drives. The software used for training and testing models can take up a significant amount of disk space. The datasets used for training and testing models can also be quite large, especially if they contain high-resolution images or other types of multimedia data. Additionally, the trained models can also take up a considerable amount of disk space.

B. SOFWARE USED

The various software used for this project are mentioned and explained in this section.

1) ANACONDA NAVIGATOR

The Anaconda is a popular distribution of Python and R for scientific computing, data science, and machine learning. It includes a package manager, an environment manager, and other tools that make it easy to manage dependencies and packages. Anaconda Navigator is a user-friendly graphical interface that allows users to easily launch

and manage applications, such as Jupyter Notebook. Anaconda Prompt allows you to create and manage multiple Python environments. Each environment can have its own set of packages and dependencies, making it easy to work on different projects with different requirements. Fig 5 shows the image of an anaconda prompt.

Fig 6. Anaconda Navigator

2) JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It is widely used in data science and machine learning for interactive data analysis, prototyping, and visualization. Jupyter Notebook supports a variety of programming languages, including Python, R etc. Jupyter Notebook allows you to interactively explore and manipulate data, which can be especially useful for data exploration and analysis. You can write and run code, display visualizations, and document your work all in one place. Jupyter Notebook allows you to document your work and make it reproducible. You can save your notebooks, version control them using Git, and create a record of the code, data, and analysis used to produce your results. Fig 6 shows the useability of jupyter notebook.

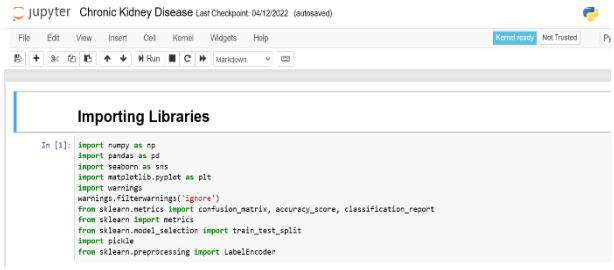

Fig 7. Jupyter Notebook

3) PYTHON3

Python is a widely used high-level programming language that is popular in data science and machine learning due to its simplicity, readability, and vast array of libraries and frameworks. Python 3 has an improved syntax, making it more consistent and easier to read than Python 2. It also includes new features like type annotations and f-strings, which can make code more efficient and easier to maintain. Python 3 includes several new modules and libraries, including asyncio, pathlib, and unittest.mock, which make it easier to work with asynchronous programming, file handling, and testing.Fig 7 shows the use case of Python3.

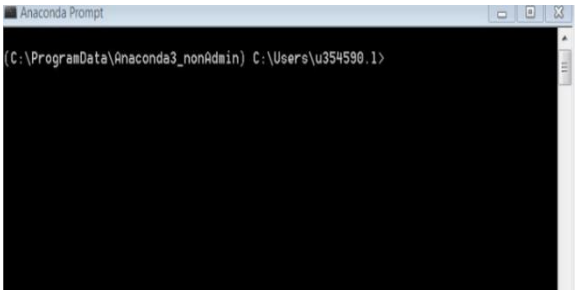

Fig 8. Python3

4) FLASK

It is a micro web framework written in Python. It is designed to be lightweight and modular, with minimal dependencies. Flask provides tools, libraries, and technologies that allow developers to build web applications quickly and easily. It is a micro-framework, which means that it only includes the bare essentials for building a web application. This allows developers to choose the tools and libraries that best fit their needs, rather than being restricted by a pre-determined set of conventions. Flask is a lightweight framework and does not require a lot of resources to run. This makes it ideal for building small to medium-sized applications that do not require heavy processing or complex logic.

5) VISUAL STUDIO CODE

It is a free and open-source code editor developed by Microsoft. It is a cross-platform tool that can be used for various programming languages, including Python. Visual Studio Code provides a rich set of features, including intelligent code completion, debugging, syntax highlighting, and code formatting. It also supports the integration of various extensions and plugins, making it a versatile and customizable code editor. Fig 8 shows how visual studio code can be used to view and manipulate data on the model.

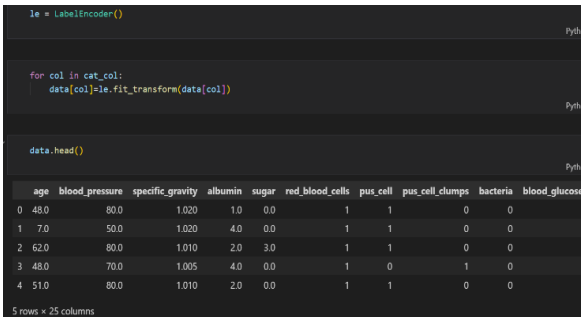

Fig 9. Visual Studio Code

RESULTS

User Interface for Home Page: Fig 9 shows the home page of the CKD prediction web application features two important options for users to access the platform: sign in and sign up. The sign-in option is designed for users who have previously registered on the platform and already have an account. These users can simply enter their email and password to access their account and make use of the prediction model. On the other hand, the sign-up option is intended for new users who want to register and create an account on the platform. This option allows users to enter their personal details, to create an account on the platform.

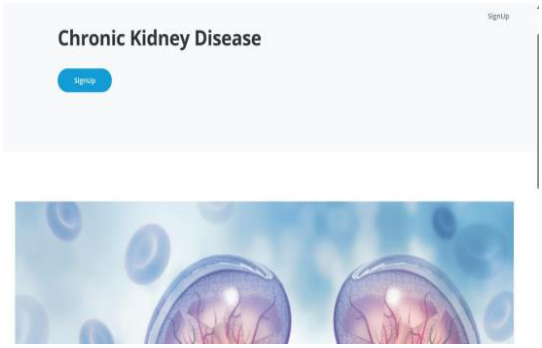

Fig 10. Home Page

Sign Up Page: The Fig 10 sign-up page is where new users can create an account to access the CKD prediction system. The page should be designed with a user-friendly interface that makes it easy for users to enter their information. The page should include fields for the user's name, email address, password, and any other relevant information that is required to create an account. The user should be able to easily navigate the page, filling in the necessary information and submitting their details. Once the user submits their information, the system should verify the information and create a new account for the user.

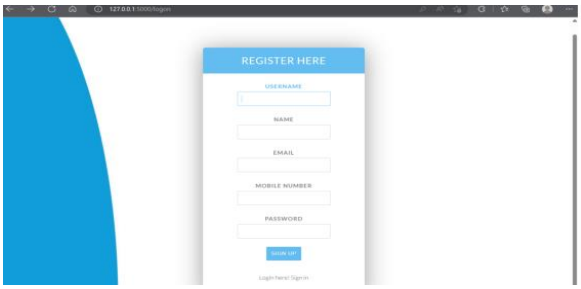

Fig 11. Sign Up Page

Data Input: Fig 11 shows the data input page, which is the core of the CKD prediction system, where users can input their relevant health data to get predictions on their likelihood of having chronic kidney disease. The input form contains various fields for users to input their personal information, such as age as well as medical information such as blood pressure, serum creatinine, and albumin levels. The page includes clear instructions and helpful tips for users to ensure they enter accurate data. It also includes error checking functionality to prevent users from inputting invalid or incorrect data.

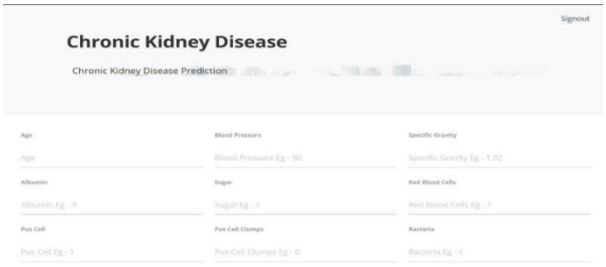

Fig 12. Data Input

Log In Page: The Fig 12 showcases a login page, which is the gateway for the users to access their accounts on the CKD prediction website. It is designed to allow registered users to securely access their accounts by entering their unique login credentials, including their username and password. The page should be easily accessible from the homepage and provide clear instructions on how to sign in. Once the user enters their login information, the system should validate their credentials and redirect them to their dashboard.

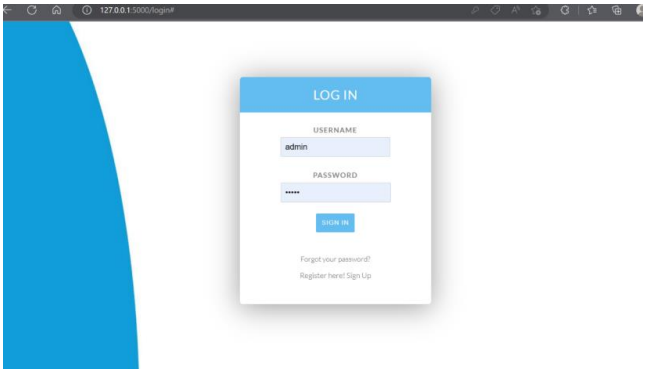

Fig 13. Log In Page

Results Page: The results page is the final page of the CKD prediction system, where the user can view the output of the prediction model. Fig 13 displays a clear message indicating whether the user is likely to have CKD or not based on the input data provided. The results page is an essential component of the CKD prediction system, providing users with valuable insights into their health status and empowering them to take action towards improving their health.

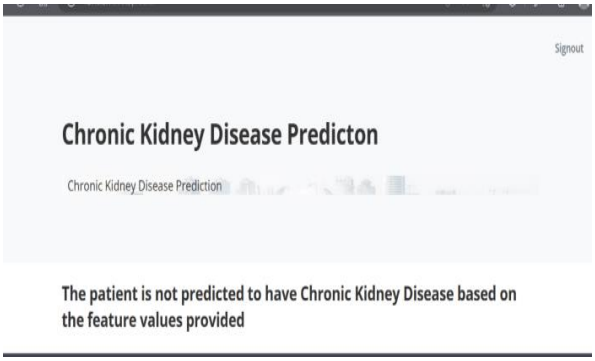

Fig 14. Result

CONCLUSION

The CKD prediction project aimed to predict whether a person has chronic kidney disease or not using machine learning algorithms. The project used the CKD dataset, which contained 25 features related to kidney health. The project successfully implemented four different algorithms: Logistic Regression, Random Forest, XGBoost, and AdaBoost, achieving different accuracies which are exceptional. The high accuracy rate of the project shows that machine learning algorithms can be effectively applied in the field of healthcare to improve diagnostic accuracy and provide better medical care to patients. This project has significant importance, as early detection of CKD can help prevent complications and delay the progression of the disease. With the implementation of this project, individuals can quickly and easily assess their risk for CKD and take appropriate measures to prevent the development of the disease. The importance of this initiative lies in chronic kidney disease early identification, which is essential for efficient management and treatment. By examining numerous health indicators and seeing trends that point to kidney disease, machine learning algorithms can help in the early detection of CKD. This may lessen the chance of renal failure and delay the progression of CKD, improving patient outcomes.

FUTURE SCOPE

There are several potential areas for future works and enhancements or advancements of this CKD prediction project. The online application's user interface is another possible area for improvement. There is need for improvement, even though the current implementation offers a straightforward and user-friendly interface for entering patient data and returning CKD prediction results. For instance, one may think about incorporating more interactive elements, like data visualizations and patient-specific advice based on their CKD risk factors. Adding more features or applying more complex machine learning techniques is one such enhancement. One may, for instance, investigate the use of deep learning techniques like neural networks, which have the ability to learn intricate patterns in data and may enhance the precision of the CKD prediction. The current model uses all the available features from the dataset. Selecting the most relevant features and engineering new ones could improve the model's performance.

REFERENCES

[1] A. S. Levey, R. Atkins, J. Coresh et al., “Chronic kidney disease as a global public health problem: approaches and initiatives–a position statement from kidney disease improving global outcomes,” Kidney International, vol. 72, no. 3, pp. 247–259, 2007.

[2] V. Jha, G. Garcia-Garcia, K. Iseki et al., “Chronic kidney disease: global dimension and perspectives,” The Lancet, vol. 382, no. 9888, pp. 260–272, 2013

[3] N. R. Hill, S. T. Fatoba, J. L. Oke et al., “Global prevalence of chronic kidney disease – a systematic review and meta-analysis,” PLoS One, vol. 11, no. 7, article e0158765, 2016.

[4] Prediction of Chronic Disease in Kidneys Using Machine Learning Classifiers Chilakamarthi Prem Kashyap; Gollapudi Sai Dayakar Reddy; M Balamurugan 2022 1st International Conference on Computational Science and Technology (ICCST)

[5] G. Abraham, S. Varughese, T. Thandavan et al., “Chronic kidney disease hotspots in developing countries in South Asia,” Clinical Kidney Journal, vol. 9, no. 1, pp. 135– 141, 2016

[6] Multilevel Ensemble Method to Identify Risks in Chronic Kidney Disease Using Hybrid Synthetic Data Karamsetty Shouryadhar; P Kiran Rao; Subarna Chatterjee 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT).

[7] Variable Selection in Genetic Algorithm Model with Logistic Regression for Prediction of Progression to Diseases Avijit Kumar Chaudhuri; Anirban Das 2020 IEEE International Conference for Innovation in Technology (INOCON).

[8] Chronic Kidney Disease Detection Using Machine Learning Technique Rama AlMomani; Ghada Al-Mustafa; Razan Zeidan; Hiam Alquran; Wan Azani Mustafa; Ahmed Alkhayyat 2022 5th International Conference on Engineering Technology and its Applications (IICETA).

[9] Analyze the Medical Threshold for Chronical Kidney Diseases and Cardio Vascular Diseases using Internet of Things S. Chitra; V. Jayalakshmi 2021 4th International Conference on Computing and Communications Technologies (ICCCT).