**Predicting Renal Health Conditions Using Machine Learning Models**

**ABSTRACT: -**

The primary objective of this research is to predict renal health conditions using machine learning models, specifically leveraging the power of the Decision Tree (DT) algorithm. Renal health conditions can significantly impact the quality of life, making early detection essential. This study proposes a system that assesses kidney health by predicting whether a patient is at risk of kidney-related issues based on various health attributes. The model is trained on a dataset containing features like Age, Blood Pressure (Bp), Serum Urea (Su), Red Blood Cells (Rbc), and several other parameters such as Blood Glucose Ratio (BGR), Hemoglobin (Hemo), and more.A Decision Tree classifier is employed for its interpretability and high performance, achieving an impressive 99% accuracy in classification. The system uses a form-based input collection where users provide values related to their health, such as age, blood pressure, serum urea levels, and other key medical attributes. Once the user inputs this data, the Decision Tree model predicts whether the kidney is in danger or not, helping individuals make informed decisions about seeking medical advice or treatment.The decision tree model has been trained to classify kidney health conditions efficiently by learning complex patterns from the data. It then outputs a clear result: indicating whether the kidney is in danger or not based on the input values, ensuring both accuracy and accessibility for users. This method enhances the healthcare experience by providing a quick, reliable, and easy-to-use tool for early detection of renal health problems.

|  |  |
| --- | --- |
| **EXISTING SYSTEM** | **PROPOSED SYSTEM** |
| * The existing system for predicting renal health conditions primarily utilized traditional machine learning algorithms such as Logistic Regression and Support Vector Machines (SVM). While these methods are commonly used in classification tasks, they do not perform as well in this specific domain. Logistic Regression, while simple and effective for linear data, struggles to capture the complex, non-linear relationships in renal health data, leading to reduced accuracy in predictions. On the other hand, Support Vector Machines, although powerful for classification, require extensive tuning of hyperparameters to optimize performance, which is often a time-consuming process. * Moreover, SVMs lack transparency, making it difficult for healthcare professionals to interpret and trust the reasoning behind the predictions. | * The proposed system aims to provide a reliable and efficient tool for predicting renal health conditions by utilizing machine learning, specifically the Decision Tree (DT) algorithm. This system is designed to predict whether a patient’s kidney is in danger based on key health parameters such as age, blood pressure, serum urea levels, red blood cell count, and other relevant factors. * In this system, the user inputs their health data via a simple form, which includes values such as blood pressure (Bp), serum urea (Su), red blood cells (Rbc), hemoglobin (Hemo), and more. These inputs are then fed into the trained Decision Tree model, which classifies the kidney’s health condition as either "in danger" or "not in danger." The Decision Tree model has been trained on a large dataset and has achieved a high accuracy of 99%, ensuring that the predictions made by the system are both accurate and reliable. |
| **EXISTING ALGORITHM**   * Support Vector Machines & Logistic Regression | **PROPOSED ALGORITHM: -**   * Decision Tree Algorithm |
| **ALGORITHM DEFINITION: -**  The existing system employs Logistic Regression and Support Vector Machines (SVM) as the primary algorithms for predicting renal health conditions. Logistic Regression is a simple, yet commonly used method for binary classification tasks. It models the relationship between the features and the binary outcome using a logistic function. While it performs well with linearly separable data, it struggles to capture more complex, non-linear relationships, which can lead to inaccurate predictions in scenarios like predicting kidney conditions where the data can be highly intricate.  On the other hand, Support Vector Machines (SVM) are more powerful, particularly for classification problems with non-linear boundaries. SVMs work by finding the optimal hyperplane that maximizes the margin between different classes. However, SVMs require careful tuning of hyperparameters like the choice of the kernel, the penalty parameter (C), and the margin, which can be computationally expensive and time-consuming. | **ALGORITHM DEFINITION: -**  The Decision Tree (DT) algorithm is a powerful and widely used supervised machine learning technique for both classification and regression tasks. In this system, the Decision Tree is employed to predict whether a patient’s kidney is in danger based on various input health parameters such as age, blood pressure, red blood cell count, and other critical features. The Decision Tree works by recursively splitting the dataset into subsets based on the most significant attribute (feature), creating a tree-like structure where each internal node represents a decision based on an attribute, and each leaf node represents a classification outcome.  For the renal health prediction system, the Decision Tree is trained on a dataset that contains labeled examples of kidney conditions, with features such as age, blood pressure, and blood sugar levels, among others. Each input provided by the user in the form is processed and used to navigate through the tree to determine whether the kidney is “in danger” or “not in danger.” |
| **DRAWBACKS: -**   * Limited Accuracy * Computational Complexity * Parameter Tuning | **ADVANTAGES: -**   * High Accuracy and Predictive Power * Interpretability and Transparency * Ease of Use and Automation |

**SYSTEM ARCHITECTURE:**

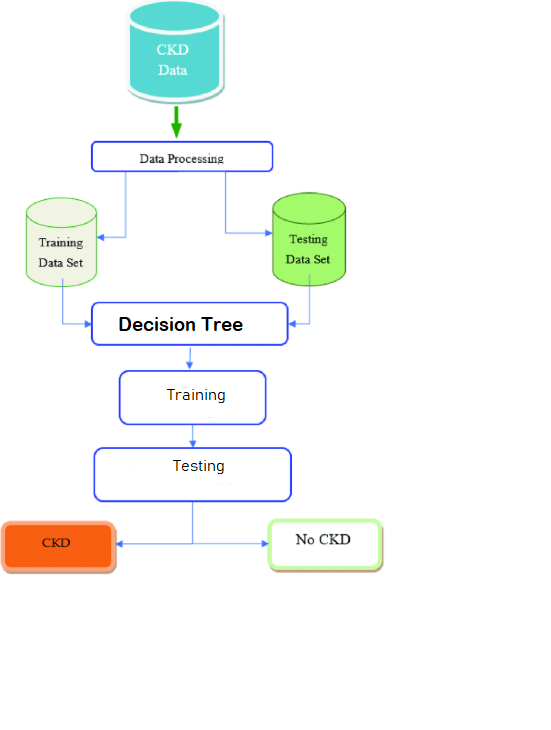


Fig: - proposed model

**MINIMUM SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

* PROCESSOR : Pentium i3 Processor
* RAM : 4GB DD RAM
* HARD DISK : 450 GB

**SOFTWARE REQUIREMENTS**

* BACK END : PYTHON
* OPERATING SYSTEM : WINDOWS 10
* IDE : Spyder3