**A Project Report on**

**Renal disease prediction using k-means clustering algorithm**

submitted in partial fulfillment for the award of

**Bachelor of Technology**

in

**Computer Science & Engineering**

by

**K. Meghamala (Y20ACS487) P. Bhargavi (Y20ACS528)**

**K. Bhanu Prakash (Y20ACS467) K. Navya Sri (Y20ACS465)**



Under the guidance of

**Mr. M. M. Meera Durga, M. Tech**

**Assistant Professor.**

Department of Computer Science and Engineering

**Bapatla Engineering College**

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

**BAPATLA – 522 102, Andhra Pradesh, INDIA**

**2023-2024**

**Department of**

**Computer Science & Engineering**



**CERTIFICATE**

This is to certify that the project report entitled **Renal disease prediction using k-means clustering algorithm** that is being submitted by K. Meghamala (Y20ACS487), P. Bhargavi (Y20ACS528), K. BhanuPrakash (Y20ACS467) and K. Navya Sri (Y20ACS465) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

**Signature of the Guide Signature of the HOD**

**Mr. M. M. Meera Durga Dr. M. Rajesh Babu**

**Assistant Professor Assoc. Prof. & Head**

**DECLARATION**

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

**K. Meghamala (Y20ACS487)**

**P. Bhargavi (Y20ACS528)**

**K. Bhanu Prakash (Y20ACS467)**

**K. Navya Sri (Y20ACS465)**

**Acknowledgement**

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Mr. M. M. Meera Durga, Assistant Professor,** Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. M. Rajesh Babu**, Assoc. Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. Shaik Nazeer** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. N. Sudhakar,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

**K. Meghamala (Y20ACS487)**

**P. Bhargavi (Y20ACS528)**

**K. Bhanu Prakash (Y20ACS467)**

**K. Navya Sri (Y20ACS465)**

Table of Contents

[List of Figures vi](#_Toc164337829)

[List of Tables vii](#_Toc164337830)

[Abstract viii](#_Toc164337831)

[1 Introduction 1](#_Toc164337832)

[1.1 Problem Statement 2](#_Toc164337833)

[1.2 Objective 2](#_Toc164337834)

[1.3 Machine Learning 3](#_Toc164337835)

[1.4 Feature Selection/Extraction 3](#_Toc164337836)

[1.4.1 Principal Component Analysis (PCA) 4](#_Toc164337837)

[1.4.2 t-Distributed Stochastic Neighbor Embedding (t-SNE) 5](#_Toc164337838)

[2 Literature Review 6](#_Toc164337839)

[3 Technology used 8](#_Toc164337840)

[4 Existing System 9](#_Toc164337841)

[4.1 Drawbacks 9](#_Toc164337842)

[5 Proposed System 10](#_Toc164337843)

[5.1 Dataset 10](#_Toc164337844)

[5.2 Pre-processing 11](#_Toc164337845)

[5.3 Model 13](#_Toc164337846)

[6 System Design 14](#_Toc164337847)

[6.1 Use-case Diagram 14](#_Toc164337848)

[6.2 Class Diagram 15](#_Toc164337849)

[6.3 Activity Diagram 16](#_Toc164337850)

[6.4 Sequence Diagram 17](#_Toc164337851)

[7 Requirements 18](#_Toc164337852)

[7.1 Functional Requirements 18](#_Toc164337853)

[7.2 Non-Functional Requirements 18](#_Toc164337854)

[7.3 Software Requirements 18](#_Toc164337855)

[7.4 Hardware Requirements 18](#_Toc164337856)

[8 Implementation 19](#_Toc164337857)

[8.1 Data Collection 19](#_Toc164337858)

[8.2 Importing Libraries 19](#_Toc164337859)

[8.2.1 NumPy 20](#_Toc164337860)

[8.2.2 Pandas 20](#_Toc164337861)

[8.2.3 Seaborn 21](#_Toc164337862)

[8.2.4 Scikit-learn 21](#_Toc164337863)

[8.3 Data pre-processing 22](#_Toc164337864)

[8.4 Exploratory Data Analysis 24](#_Toc164337865)

[8.5 Clustering 25](#_Toc164337866)

[9 Results and conclusion 30](#_Toc164337867)

[10 Future Work 31](#_Toc164337868)

[11 References 32](#_Toc164337869)

List of Figures

[**Figure 6‑1 Use Case Diagram** 14](#_Toc164336664)

[**Figure 6‑2 Class Diagram** 15](#_Toc164336665)

[**Figure 6‑3 Activity Diagram** 16](#_Toc164336666)

[**Figure 6‑4 Sequence Diagram** 17](#_Toc164336667)

[**Figure 8‑1 Dataset** 19](#_Toc164336668)

[**Figure 8‑2 Importing and loading data** 22](#_Toc164336669)

[**Figure 8‑3 Data description** 23](#_Toc164336670)

[**Figure 8‑4 Proportions of missing values** 23](#_Toc164336671)

[**Figure 8‑5 PCA components ranked by variation** 26](#_Toc164336672)

[**Figure 8‑6 k-means clustering flowchart** 28](#_Toc164336673)

[**Figure 8‑7 PCA + kmeans vs target variable** 29](#_Toc164336674)

[**Figure 9‑1 Results on testing data** 30](#_Toc164336675)

[**Figure 9‑2 Result** 30](#_Toc164336676)

List of Tables

[**Table 1 Dataset Description** 10](#_Toc164337749)

Abstract

Renal diseases present a significant health challenge worldwide, impacting millions of lives annually. Early detection and timely intervention are crucial to mitigate their adverse effects on health outcomes. In this study, we propose a novel approach for renal disease prediction leveraging machine learning techniques, particularly K-Means clustering.

Our project focuses on the development of a predictive model that utilizes K-Means clustering algorithm to analyse and classify patterns within renal health datasets. The primary objective is to identify distinct clusters of patients based on a set of relevant features such as demographic information, medical history, and clinical measurements. By partitioning the data into clusters, our model aims to uncover hidden patterns and associations that may signify the onset or progression of renal diseases.

We will employ a comprehensive dataset comprising anonymized patient records, encompassing a diverse range of attributes pertinent to renal health. Through rigorous preprocessing and feature engineering, we will refine the dataset to enhance the quality and relevance of input variables. Subsequently, the K-Means algorithm will be applied to segment the dataset into clusters, with each cluster representing a distinct subgroup of patients with similar characteristics.

The outcomes of this research hold significant implications for the early detection and management of renal diseases. By leveraging K-Means clustering, our predictive model aims to provide healthcare practitioners with valuable insights into patient stratification and risk assessment, thereby facilitating personalized treatment strategies and improving overall patient outcomes.

# Introduction

Chronic Kidney Disease (CKD) is a significant health concern affecting millions worldwide, with its prevalence steadily rising over the past few decades. Early detection and timely intervention are crucial for effective management and prevention of complications associated with CKD.

Chronic kidney disease (CKD) is a global healthcare challenge with significant morbidity and mortality. Early identification of high-risk CKD patients for end-stage kidney disease (ESKD) is crucial.[5]

“Around 10% of the Indian population suffers from CKD and every year over 1 lakh cases of renal failure are reported.” – Nephrologist Dr. Gurudev K C said. The kidneys’ main job is to get rid of waste and extra water from the blood. Renal disease means gradual failure of kidneys. This is two types- Acute Kidney Injury (AKI) and Chronic Kidney Disease (CKD).

AKI occurs when the kidneys suddenly fail due to an injury, medication or illness. Some disease and conditions can damage your kidneys and lead to AKI. Some examples include: a type of severe, life-threatening infection called “sepsis,” a type of cancer called “multiple myeloma”. AKI includes 4 stages: i) initiation, ii) oligo-anuria, iii) polyuria and iv) restitution. In this dynamic process, clinical signs of renal dysfunction emerge during stage 2.

CKD is the gradual loss of kidney function mainly caused by high blood pressure, diabetes and an inflammatory condition known as glomerulonephritis. This condition is called chronic because the damage happens slowly over a long period of time.

We can also identify in which stage of CKD the person is in according to the GFR (Glomerular Filtration Rate). CKD includes 5 stages: Stage 1 with normal or high GFR (GFR > 90 mL/min), Stage 2 Mild CKD (GFR = 60-89 mL/min), Stage 3A Moderate CKD (GFR = 45-59 mL/min), Stage 3B Moderate CKD (GFR = 30-44 mL/min),

Stage 4 Severe CKD (GFR = 15-29 mL/min), Stage 5 End Stage CKD (GFR < 15 mL/min). To figure out eGFR (estimated GFR), the doctor usually uses the result of blood test that measures your creatinine level. The use of GFR as the best indicator of renal function to identify different stages of CKD with each successive stage defining a more severe decrease in GFR and the last stage defining kidney failure with a GFR <15 ml/min/1.73 m2 [8] were also advocated.

Symptoms of CKD are nausea, vomiting, loss of appetite, fatigue and weakness, sleep problems, urinating more or less, decreased mental sharpness, muscle cramps, swelling of feet and ankles, dry, itchy skin, high blood pressure that’s difficult to control, shortness of breath if fluid builds up in the lungs, chest pain if fluid builds up around the lining of the heart.

## Problem Statement

Chronic Kidney Disease (CKD) is a significant public health concern globally, leading to increased morbidity and mortality rates.

## Objective

The goal is to design, develop and implement a machine learning based robust prediction system using K-Means algorithm that can accurately identify individuals at risk of developing CKD based on various clinical and demographic factors.

## Machine Learning

It is a branch of Artificial Intelligence (AI) that involves the development of algorithms and models. In essence, it’s about teaching machines to learn from data and improve overtime without human intervention. This enables computers to learn patterns and make predictions or decisions based on the data, without being explicitly programmed to perform specific tasks. [9] Because there is such a wide diversity of health datasets, machine learning algorithms are the most appropriate method for enhancing the accuracy of diagnosis prediction.

ML algorithms can be broadly categorized into supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves training a model on labeled data, where the algorithm learns to predict the output from input data. Unsupervised learning involves finding patterns or hidden structures in unlabeled data. Reinforcement learning involves training agents to make decisions by trial and error, receiving feedback in the form of rewards or penalties.

ML is widely used in various applications such as image and speech recognition, natural language processing, recommendation systems, autonomous vehicles, and healthcare. Its versatility and ability to handle complex, large-scale data make it a powerful tool in solving real-world problems and driving innovation across industries.

## Feature Selection/Extraction

Feature extraction is a crucial step in data analysis and machine learning, where the goal is to transform raw data into a format that is more suitable for modeling. It involves selecting or creating a subset of relevant features (variables) from the original dataset. These features should capture essential information while reducing dimensionality and removing noise.

Various techniques are used for feature extraction depending on the type of data and the specific problem. This can include statistical methods, dimensionality reduction algorithms like PCA (Principal Component Analysis) or t-SNE (t-distributed Stochastic Neighbor Embedding), and domain-specific knowledge.

Effective feature extraction is vital for improving model performance, reducing computational complexity, and enhancing interpretability. It helps uncover underlying patterns and relationships in data, enabling better decision-making and insight generation.

### Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique used to simplify complex data while retaining its essential features. By transforming the data into a new coordinate system, PCA identifies the directions (principal components) that capture the maximum variance in the data. These principal components are orthogonal to each other, meaning they are uncorrelated.

PCA is widely used in various fields such as image processing, pattern recognition, and finance. It helps in visualizing high-dimensional data, identifying patterns, reducing noise, and speeding up machine learning algorithms by reducing the number of features.

In essence, PCA offers a powerful way to summarize and compress data while preserving its inherent structure, making it a valuable tool in data analysis and machine learning.

### t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a powerful dimensionality reduction technique commonly used in machine learning and data visualization. It's particularly effective at visualizing high-dimensional data in lower dimensions, while preserving the local structure of the data points as much as possible. By measuring the similarity between data points in high-dimensional space and embedding them in a lower-dimensional space, t-SNE reveals patterns and relationships that might be obscured in the original data. It's widely used in various fields such as natural language processing, image analysis, and bioinformatics for exploratory data analysis and visualization.

# Literature Review

In this paper [1], the research aimed to build an intelligent machine learning model that was used reliably to establishCKD diagnosis. Algorithms such as, K-Means clustering, Isolation Forest, DB-Scan, and Autoencoder are implemented on various sets of selected features.

In this paper [2], the study aimed at applying machine learning (ML) techniques for CKD’s classification and prediction using small features’ subset provided in the dataset they were going to use so that it helped healthcare professionals, stakeholders and healthcare facilities around the world to determine and classify patients with either CKD or without CKD.

In this paper [3], several typical and recent AI algorithms are studied in the context of CKD and the extreme gradient boosting (XGBoost) is chosen as their base model for its high performance.

In this paper [4], the prediction models used include Random Forest (RF), Support Vector Machine (SVM) and Decision Tree (DT). The results from the experiments indicated that RF based on recursive feature elimination with cross validation has better performance than SVM and DT.

Tekale et al. [6] worked on “Prediction of Chronic Kidney Disease Using Machine Learning Algorithm” with a dataset consists of 400 instances and 14 features.

Mohammed and Beshah [7] conducted their research on developing a self-learning knowledge-based system for diagnosis and treatment of the first three stages of chronic kidney have been conducted using machine learning. A small number of data have been used in this research and they have developed prototype which enables the patient to query KBS to see the delivery of advice. They used decision tree in order to generate the rules. The overall performance of the prototype has been stated as 91% accurate.

Xiao et  al. [10] proposed prediction of  chronic kidney disease progression using logistic regression, Elastic Net, lasso regression, ridge regression, support vector machine, random forest, XGBoost, neural network and k-nearest neighbor and compared the models based on their performance.

# Technology used

1. Python:

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming

1. Jupyter Notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

1. Google Colab:

Colaboratory, or Colab for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

# Existing System

Several studies have explored the application of machine learning unsupervised techniques, including k-means clustering, db-scan, i-forest, autoencoder and some of machine learning supervised techniques, including random forest, decision tree, adaboost for renal disease prediction. These studies typically involve the analysis of large clinical datasets containing patient demographics, laboratory results, imaging data, and other relevant clinical parameters.

Many studies integrate multiple data sources, such as electronic health records (EHRs), genetic data, and patient-reported outcomes, to improve the accuracy and robustness of predictive models. Integrating diverse data sources allows for a comprehensive understanding of the underlying factors contributing to renal diseases.

## Drawbacks

The challenges may include

* Model interpretability,
* Generalizability across different patient populations,
* Usage of old Pre-processing techniques,
* Given less priority to the models individually.

# Proposed System

A model using k-means clustering algorithm is proposed to predict CKD in individuals. The dataset contains data of 400 patients. It consists of 24 clinical attributes and 1 class attribute. The datasets consist of 250 CKD cases and 150 Non-CKD cases. Missing data is a significant problem in real-world datasets, especially in the medical field. On average, every patient record and attribute have a few missing values.

## Dataset

**Table 1 Dataset Description**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Role** | **Type** | **Description** | **Demographic** | **Units** | **Missing Values** |
| Age | Feature | Integer |  | Age | year | Yes |
| Bp | Feature | Integer | Blood pressure |  | Mm/Hg | Yes |
| Sg | Feature | Categorical | Specific gravity |  |  | Yes |
| Al | Feature | Categorical | Albumin |  |  | Yes |
| su | Feature | Categorical | Sugar |  |  | Yes |
| Rbc | Feature | Binary | Red blood cells |  |  | Yes |
| Pc | Feature | Binary | Pus cell |  |  | Yes |
| Pcc | Feature | Binary | Pus cell clumps |  |  | Yes |
| Ba | Feature | Binary | Bacteria |  |  | Yes |
| Bgr | Feature | Integer | Blood glucose random |  | Mgs/dl | Yes |
| Bu | Feature | Integer | Blood urea |  | Mgs/dl | Yes |
| Sc | Feature | Continuous | Serum creatinine |  | Mgs/dl | Yes |
| Sod | Feature | Integer | Sodium |  | mEq/L | Yes |
| Pot | Feature | Continuous | Potassium |  | mEq/L | Yes |
| Hemo | Feature | Continuous | Hemoglobin |  | gms | Yes |
| Pcv | Feature | Integer | Packed cell volume |  |  | Yes |
| Wbcc | Feature | Integer | White blood cell count |  | Cells/cmm | Yes |
| Rbcc | Feature | Continuous | Red blood cell count |  | Millions/cmm | Yes |
| Htn | Feature | Binary | Hypertension |  |  | Yes |
| Dm | Feature | Binary | Diabetes mellitus |  |  | Yes |
| Cad | Feature | Binary | Coronary artery disease |  |  | Yes |
| Appet | Feature | Binary | Appetite |  |  | Yes |
| Pe | Feature | Binary | Pedal edema |  |  | Yes |
| Ane | Feature | Binary | Anemia |  |  | Yes |
| class | Target | binary | Ckd or notckd |  |  | no |

## Pre-processing

Preprocessing is a crucial step in data analysis and machine learning workflows. It involves cleaning, transforming, and organizing raw data into a format suitable for further analysis or modeling. This process typically includes steps like:

Data Cleaning: Removing or correcting errors, inconsistencies, or missing values in the dataset.

Normalization/Scaling: Bringing different features to a similar scale to prevent certain features from dominating the model.

Feature Selection/Extraction: Choosing relevant features or creating new ones to improve model performance and reduce dimensionality.

Encoding Categorical Variables: Converting categorical data into a numerical format that can be easily understood by machine learning algorithms.

Handling Imbalanced Data: Addressing situations where the distribution of classes in the dataset is skewed, which can lead to biased models.

Feature Engineering: Creating new features from existing ones to capture more information or improve model performance.

Firstly, in our project after a bit of exploration, data cleaning is performed in a way to remove typos, null values, to replace missing values. Next, one-hot encoding is performed, which is a encoding technique to convert each categorical value into new categorical column and assign a binary value of 1 or 0 to those columns. One-hot encoding is perfect here as every categorical feature has only 2 values, which means we won't be increasing dimensions.

Secondly, we are performing knn-imputation, which is designed to find k-nearest neighbors for missing incomplete instance from all complete instances (without missing values) in a given dataset and then filling the missing datum with the most frequent one occurring in the neighbors with the target feature is categorical referred to as majority rule or with the mean of the neighbors if the target feature is numerical and referred to mean rule.

Next, Exploratory data analysis is done on the data, which is an analysis approach that identifies general patterns in the data. These patterns include outliers and features of the data that might be unexpected.

## Model

A machine learning model is a mathematical representation or algorithm that learns patterns and relationships from data in order to make predictions, decisions, or identify patterns in new data. It's essentially the computational embodiment of a problem-solving or decision-making process.

**k-means**

k-means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into a predetermined number of clusters. The goal of k-means is to group similar data points together and discover underlying patterns or structures in the data.

The algorithm works by iteratively assigning each data point to the nearest cluster centroid and then updating the centroids to the mean of all data points assigned to that cluster. This process continues until the centroids no longer change significantly, or a specified number of iterations is reached.

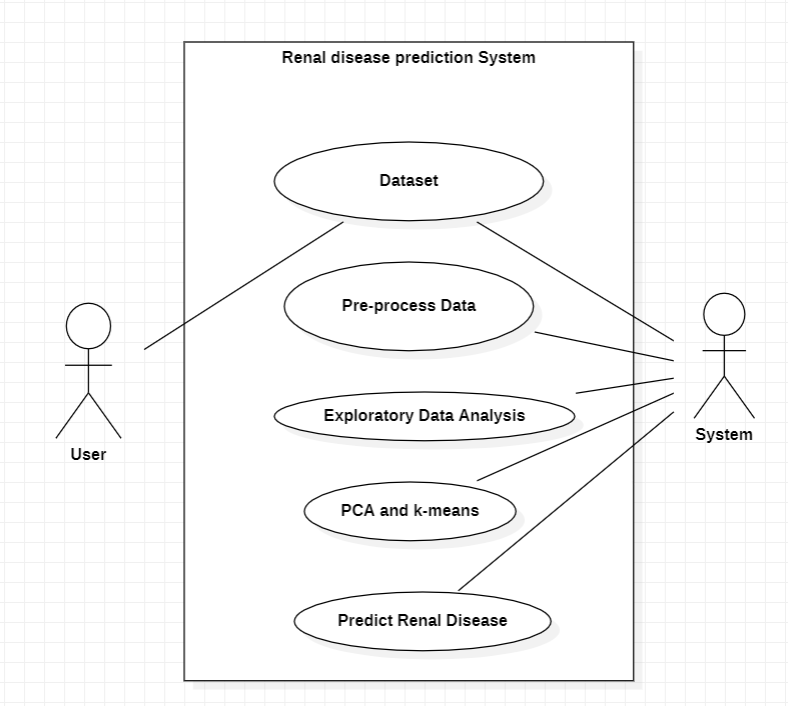
k-means clustering is effective for datasets where the number of clusters is known in advance or can be reasonably estimated. It is widely used in various applications such as customer segmentation, image compression, anomaly detection, and pattern recognition.

k-means clustering algorithm’s advantage over all other clustering algorithms is it guarantees convergence, can warm-start the positions of centroids, easily adapts to new examples, generalizes to clusters of different shapes and sizes such as elliptical clusters.

# System Design

## Use-case Diagram

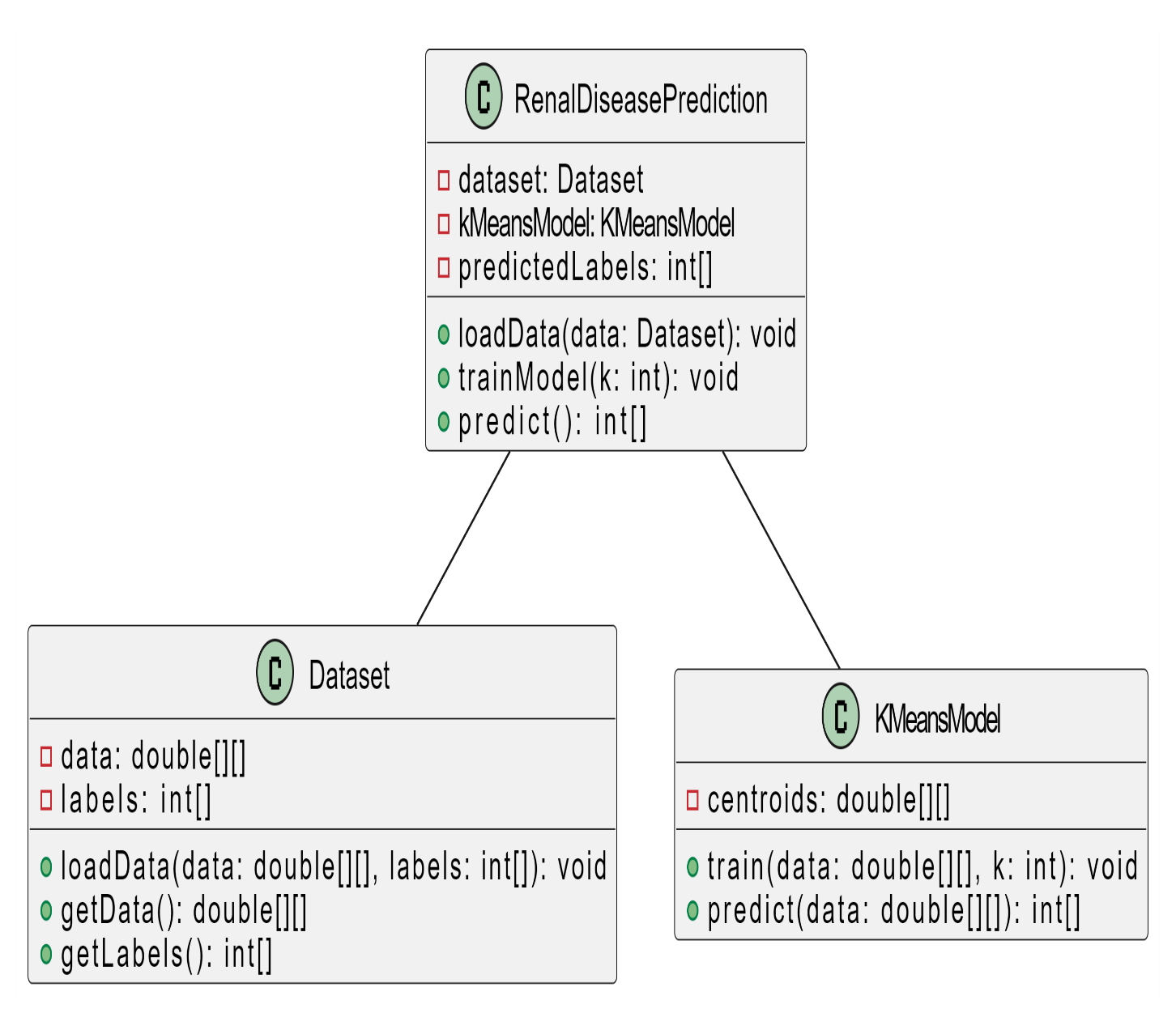
Use-case diagram describes the high-level functions and scope of a system. This diagrams also identify the interaction between the system and its actors. The fig 6.1 shows the use-case diagram representation of the system.



**Figure 6‑1 Use Case Diagram**

## Class Diagram

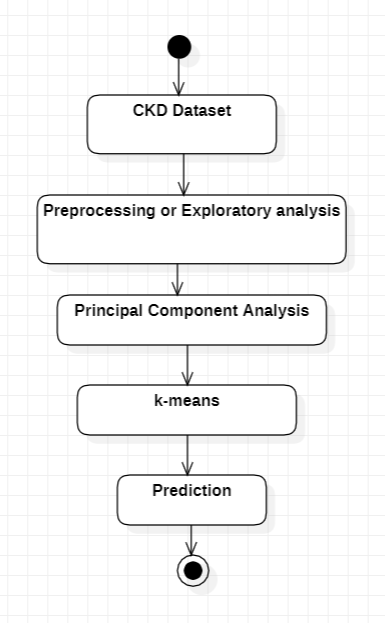
Class diagram is basically graphical representation of the static view of the system and represents different aspects of the application. The fig 6.2 shows the class diagram representation of the system.



**Figure 6‑2 Class Diagram**

## Activity Diagram

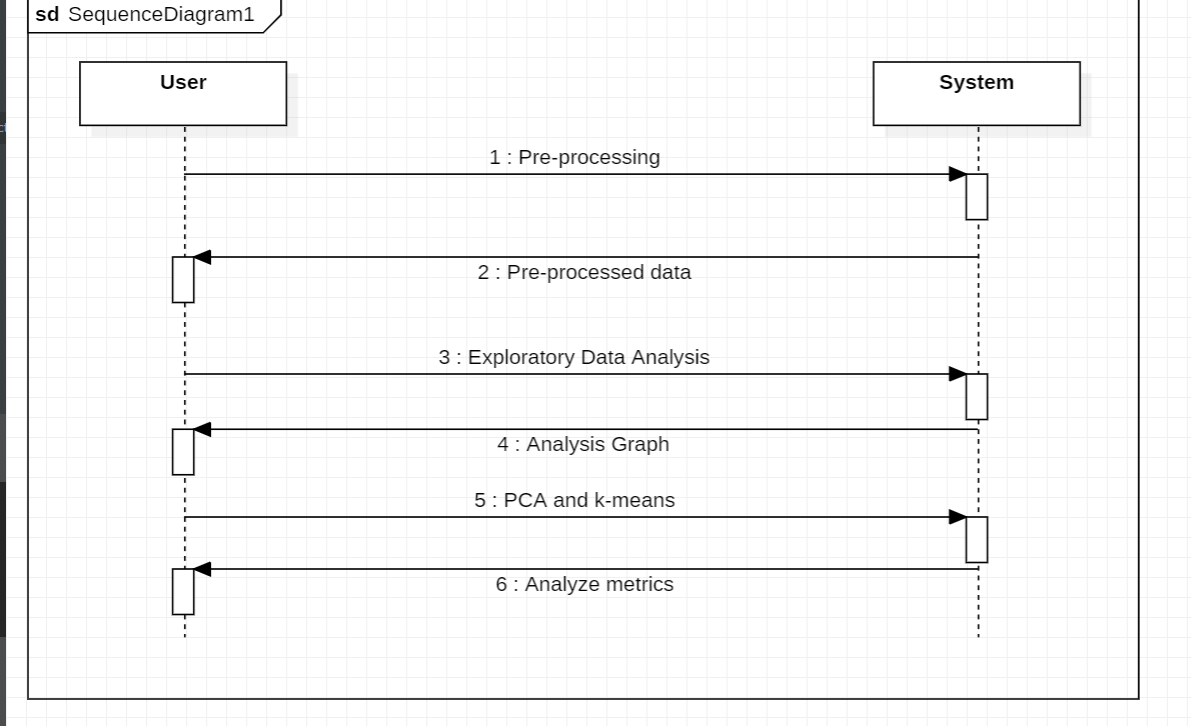
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. In UML, an activity diagram provides a view of the behavior of a system by describing the sequence of actions in a process. The Figure 6.3 shows the activity diagram representation of the system.



**Figure 6‑3 Activity Diagram**

## Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. The Figure 5.4 shows the sequence diagram representation of the system. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them.



**Figure 6‑4 Sequence Diagram**

# Requirements

## Functional Requirements

1. Input data to train the model.

2. Data preprocessing to clean the raw input data.

3. Accurate algorithm for training model.

4. Post-processing to manage the output.

## Non-Functional Requirements

1. Accuracy: Accuracy is major part as we require accurate results.

2. Reliability: Reliability is needed so that model is not disturbed.

3. Adaptability: The model should adapt automatically according to user input.

4. Security: Security has to be provided to the data model in case of sensitive data.

## Software Requirements

1. Operating System: Windows or Linux or macOS
2. Workspace: Google Colab
3. Libraries: numpy, scikit-learn, seaborn, matplotlib

## Hardware Requirements

1. RAM: >=8GB
2. Hard disk or SSD: >=500GB
3. Processor: 7th Gen Intel core i5 or 2nd Gen Ryzen or higher.

# Implementation

## Data Collection

The dataset is collected from Kaggle. It contains data of 400 patients. It consists of 24 clinical attributes and 1 class attribute. The datasets consist of 250 CKD cases and 150 Non-CKD cases. Missing data is a significant problem in real-world datasets, especially in the medical field. On average, every patient record and attribute have a few missing values.

The following are instances from the dataset:

A white grid with many people

Description automatically generated

**Figure 8‑1 Dataset**

## Importing Libraries

The necessary libraries installed and imported in our project are numpy, matplotlib, pandas, seaborn, sci-kit learn.

The following is the explanation of how the libraries work:

### NumPy

NumPy is a fundamental library in Python for numerical computing, providing powerful tools for handling large arrays and matrices of numeric data. It offers functionalities for mathematical operations like linear algebra, Fourier transforms, and random number generation. NumPy's main object is the multidimensional array, ndarray, which allows efficient manipulation of large datasets. Its array operations are much faster than equivalent operations performed using Python lists, making it essential for scientific computing and data analysis tasks. NumPy also integrates seamlessly with other Python libraries like SciPy, Matplotlib, and Pandas, forming a robust ecosystem for scientific computing and data analysis in Python. Overall, NumPy is a cornerstone of numerical computing in Python, enabling efficient manipulation and analysis of large datasets with ease.

### Pandas

Pandas is a powerful Python library widely used for data manipulation and analysis. It provides data structures and functions for efficiently handling structured data, primarily in the form of dataframes. Dataframes are two-dimensional labeled arrays capable of holding heterogeneous data types. One of Pandas' key features is its ability to load data from various file formats such as CSV, Excel, SQL databases, and more, making it highly versatile for working with real-world data. With Pandas, users can perform essential data operations like indexing, filtering, grouping, and aggregation with ease. Its integration with other Python libraries like NumPy and Matplotlib facilitates seamless data processing, analysis, and visualization workflows. Additionally, Pandas offers robust functionality for handling missing data, time series data, and data alignment, making it indispensable for data scientists, analysts, and researchers across various domains.

### Seaborn

Seaborn is a powerful Python visualization library built on top of matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics. Seaborn is particularly adept at creating complex visualizations with minimal code, making it a favorite among data scientists and statisticians alike. Its strengths lie in its ability to handle structured data, allowing users to easily create plots such as scatter plots, bar plots, box plots, and heatmaps with just a few lines of code. Seaborn also offers a variety of customization options, allowing users to fine-tune the appearance of their plots to better convey their data's insights. Additionally, Seaborn integrates well with Pandas dataframes, simplifying the process of visualizing data stored in this common format. Overall, Seaborn is a valuable tool for anyone looking to explore and communicate patterns within their data through compelling visualizations.

### Scikit-learn

Scikit-learn is a versatile machine learning library in Python, renowned for its simplicity and efficiency in implementing various algorithms. It provides a unified interface for implementing popular machine learning techniques, including classification, regression, clustering, and dimensionality reduction. With a rich set of functionalities, Scikit-learn enables users to preprocess data, select relevant features, and evaluate model performance seamlessly. Its robustness lies in its extensive documentation, making it accessible to both novice and expert users. Additionally, Scikit-learn integrates well with other Python libraries such as NumPy, Pandas, and Matplotlib, facilitating a smooth workflow in data preprocessing, analysis, and visualization. Its active community ensures continuous development and support, making it an indispensable tool for machine learning practitioners and researchers alike.

## Data pre-processing

For pre-processing, we must first load the data.

A screenshot of a computer

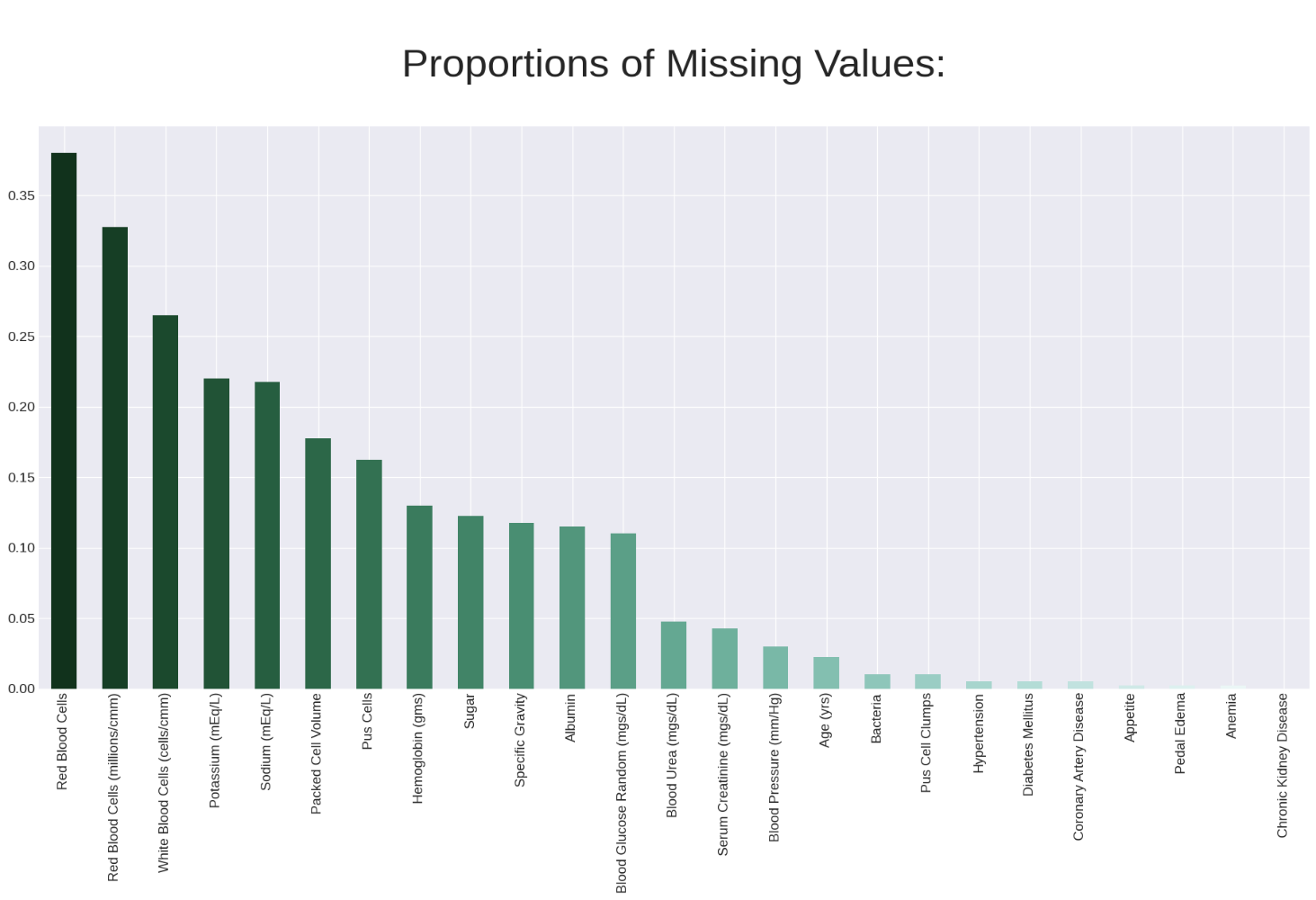
Description automatically generated

**Figure 8‑2 Importing and loading data**

Description of data: A screenshot of a computer screen

Description automatically generated

**Figure 8‑3 Data description**



**Figure 8‑4 Proportions of missing values**

We are using one-hot encoding and knn-imputation to fill missing values.

One-hot encoding is a popular technique used in machine learning and natural language processing to represent categorical data numerically. In this method, each category is represented by a binary vector where all elements are zero except for one element which corresponds to the category being represented. One-hot encoding is useful because it allows algorithms to efficiently interpret categorical data, as it ensures that the numerical representation of categories does not imply any ordinal relationship between them. However, it can lead to high-dimensional data, especially when dealing with a large number of categories, which might introduce computational overhead and memory issues. Despite this drawback, one-hot encoding remains a widely used method for handling categorical variables in machine learning tasks.

K-Nearest Neighbors (KNN) imputation is a popular method used to fill in missing values in datasets. It operates on the principle of similarity: for each missing value, the algorithm identifies the k closest data points based on some distance metric, typically Euclidean distance. These neighboring points' known values are then used to estimate the missing value, often by averaging or weighting them. KNN imputation is versatile and applicable to various data types, including numerical and categorical variables. However, its effectiveness heavily relies on the choice of k and the appropriate distance metric, which can impact imputation accuracy and computational efficiency. Moreover, KNN imputation may struggle with high-dimensional data or when dealing with sparse datasets. Nonetheless, with careful parameter tuning and preprocessing, KNN imputation can be a valuable tool for handling missing data and maintaining dataset integrity in machine learning and data analysis tasks.

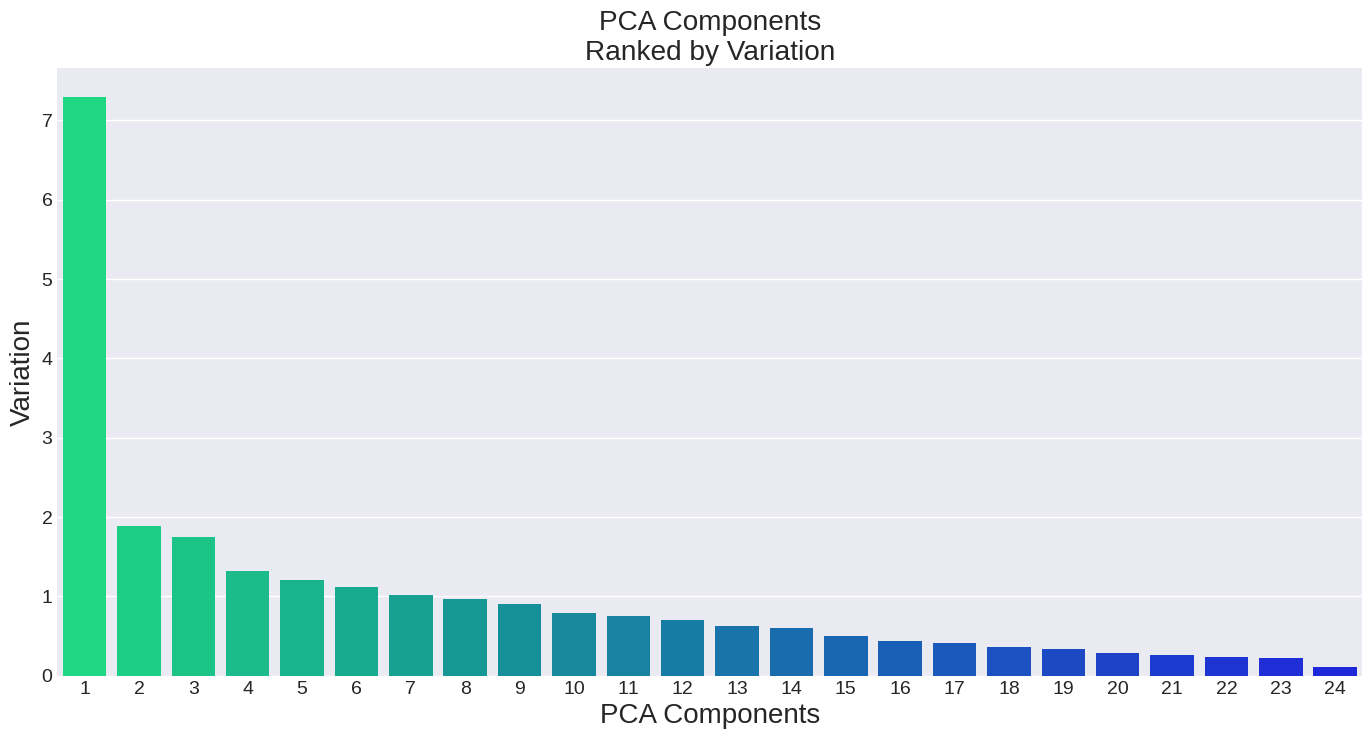
## Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical phase in the data analysis process, serving as a preliminary step to understand the structure, patterns, and potential insights hidden within a dataset. It involves visually inspecting, summarizing, and interpreting data to uncover trends, anomalies, and relationships. Through techniques such as summary statistics, data visualization, and correlation analysis, EDA helps analysts gain familiarity with the data, identify outliers or missing values, and formulate hypotheses for further investigation. EDA also plays a crucial role in guiding subsequent modeling and decision-making processes by providing insights into the data's characteristics and informing appropriate strategies for data preprocessing and feature engineering. Ultimately, effective EDA empowers data scientists and analysts to make informed decisions and derive meaningful conclusions from the data, laying the groundwork for more advanced analytics and modeling tasks.

## Clustering

**Unsupervised machine learning:** It encompasses a range of techniques where the model is tasked with extracting patterns or structures from input data without explicit guidance or labeled examples. Clustering is a fundamental task within this domain, wherein the algorithm groups similar data points together based on certain features or characteristics. Dimensionality reduction techniques like principal component analysis (PCA) aim to condense complex datasets into a lower-dimensional space while preserving important information. Another key approach is anomaly detection, which identifies outliers or irregularities in data that deviate significantly from the norm. Unsupervised learning is particularly valuable in exploratory data analysis, uncovering hidden patterns, and understanding underlying structures within datasets without the need for labeled training data. However, its effectiveness relies heavily on the quality and nature of the data, as well as the appropriate selection and configuration of algorithms.

**Principal Component Analysis:** Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction in data analysis. Its primary goal is to simplify complex datasets by transforming them into a lower-dimensional space while preserving most of the original variance. PCA achieves this by identifying the principal components, which are the orthogonal vectors that maximize the variance of the data when projected onto them. These principal components are ordered such that the first component captures the most variance, the second captures the second most, and so on. By retaining only a subset of these components, PCA allows for the reduction of data dimensionality, which can aid in visualization, noise reduction, and speeding up subsequent analysis. PCA is widely used in various fields such as image processing, finance, genetics, and neuroscience for exploratory data analysis and feature extraction.



**Figure 8‑5 PCA components ranked by variation**

**k-means clustering:** K-means is a widely used clustering algorithm in machine learning and data mining for partitioning a dataset into K distinct, non-overlapping clusters. The algorithm works by iteratively assigning each data point to the nearest centroid, which represents the center of a cluster. After assigning all data points, the centroids are recalculated as the mean of all points assigned to each cluster. This process continues until convergence, typically when the centroids no longer change significantly or after a specified number of iterations. K-means aims to minimize the within-cluster variance, making it particularly efficient for large datasets. However, its performance can be sensitive to the initial placement of centroids and the choice of K, and it may converge to local optima rather than the global optimum. Regularization techniques and initialization strategies can help mitigate these issues, ensuring more robust clustering results.

**Pseudo code of k-means:**

1. Initialize K centroids randomly

2. Repeat until convergence:

a. Assign each data point to the nearest centroid:

for each data point x:

compute distance between x and each centroid

assign x to the nearest centroid

b. Update centroids:

for each centroid:

compute mean of all data points assigned to it

update centroid position to the mean

3. Convergence criteria:

- Centroids no longer change significantly

- Maximum number of iterations reached

A diagram of a algorithm

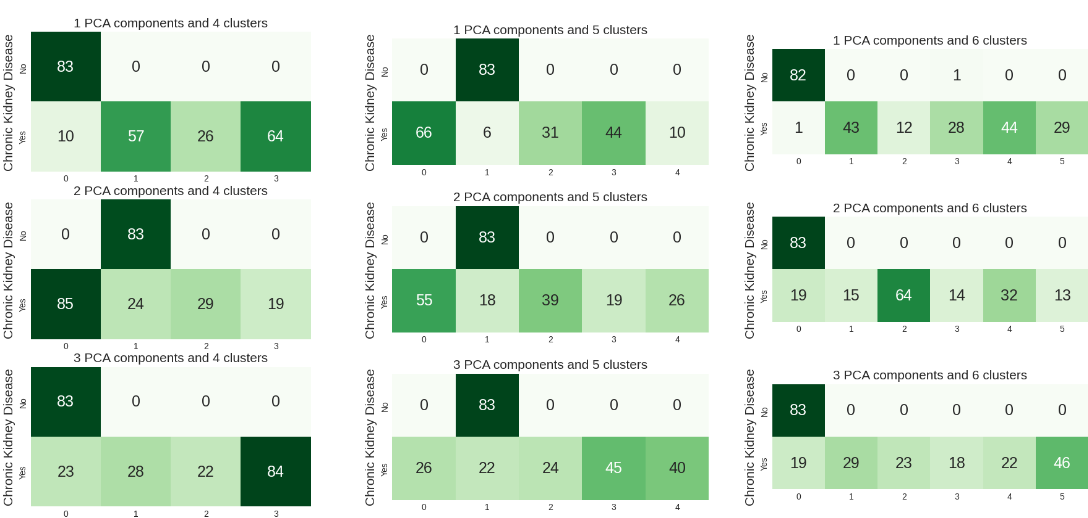
Description automatically generated

**Figure 8‑6 k-means clustering flowchart**

**Training and Testing of data:**

Training and testing data are pivotal components in the development and evaluation of machine learning models. The training data set is utilized to train the model, enabling it to learn the underlying patterns and relationships within the data. This process involves exposing the model to a vast array of examples, allowing it to adjust its internal parameters through techniques like gradient descent. Conversely, the testing data set serves as a benchmark to assess the model's performance once training is complete. It comprises data that the model has not encountered during training, ensuring an unbiased evaluation of its predictive capabilities. By evaluating the model's performance on unseen data, researchers can gauge its generalization ability and identify potential issues like overfitting or underfitting. Effective management and selection of training and testing data are crucial for developing robust and reliable machine learning models.

The data is split into 80% for training and 20% for testing. The clusters and pca components are checked through a loop and fixed on optimal choice which is 1 pca component with 6 clusters due to the uniform distribution.



**Figure 8‑7 PCA + kmeans vs target variable**

# Results and conclusion

There are only a limited number of studies, using unsupervised systems and algorithms to solve the issue of early detection of CKD. However, in detecting CKD, there were some studies based on semi-supervised and supervised learning which were worth mentioning.

We are achieving a 99.37% accuracy rate in renal disease prediction using the k-means algorithm is a significant accomplishment, indicating the effectiveness of the model accurately identifying patterns and making predictions based on patient data. In conclusion, renal disease prediction using k-means clustering algorithm was able to identify patterns or groupings within the data related to renal disease.

A screenshot of a graph

Description automatically generated

**Figure 9‑1 Results on testing data**

A screenshot of a computer code

Description automatically generated

**Figure 9‑2 Result**

# Future Work

* Live streaming data from the hospitals can be utilized without using previously existing data.
* Other advance preprocessing techniques and clustering algorithms can be used according to the data collected.

# References

[1] L. Antony et al., "A Comprehensive Unsupervised Framework for Chronic Kidney Disease Prediction," in IEEE Access, vol. 9, pp. 126481-126501, 2021, doi: 10.1109/ACCESS.2021.3109168.

[2] O. Abuomar and P. Sogbe, "Classification and Detection of Chronic Kidney Disease (CKD) Using Machine Learning Algorithms," 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 2021, pp. 1-8, doi: 10.1109/ICECET52533.2021.9698666.

[3] A. Ogunleye and Q.-G. Wang, ``XGBoost model for chronic kidney disease diagnosis,'' IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 17, no. 6, pp. 2131?2140, Nov. 2020.

[4] S. Drall, G. S. Drall, S. Singh, and B. B. Naib, ``Chronic kidney disease prediction using machine learning: A new approach,'' Int. J. Manage., Technol. Eng., vol. 8, pp. 278\_287, May 2018.

[5] Bai Q, Su C, Tang W, Li Y. Machine learning to predict end stage kidney disease in chronic kidney disease. Sci Rep. 2022 May 19;12(1):8377. doi: 10.1038/s41598-022-12316-z. PMID: 35589908; PMCID: PMC9120106.

[6] Tekale S, Shingavi P, Wandhekar S, Chatorikar A. Prediction of chronic kidney disease using machine learning algorithm. Disease. 2018;7(10):92–6.

[7] Priyanka K, Science BC. Chronic kidney disease prediction based on naive Bayes technique. 2019. p. 1653–9.

[8] Islam M.A., Akter S., Hossen M.S., Keya S.A., Tisha S.A., Hossain S. *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)* IEEE; 2020. Risk factor prediction of chronic kidney disease based on machine learning algorithms; pp. 952–957.

[9] Akhter T., Islam M.A., Islam S. Artificial neural network based covid-19 suspected area identification. *J Eng Adv.*2020;1:188–194.

[10] Xiao J, et al. Comparison and development of machine learning tools in the prediction of chronic kidney disease progression. J Transl Med. 2019;17(1):1–13.