A black background with white text

Description automatically generated with low confidence

MSc Data Science Project

7PAM2002-0509-2023

Department of Physics, Astronomy and Mathematics

**Data Science FINAL PROJECT REPORT**

**Project Title:**

Predicting Chronic Kidney Disease with Machine Learning

**Student Name and SRN:**

Niveditha Cherukuri and 21082045

Supervisor: Luigi Alfonsi

Date Submitted: 29-August-2024

Word Count: 7235

# **DECLARATION STATEMENT**

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire.

I have read the guidance to students on academic integrity, misconduct and plagiarism information at [Assessment Offences and Academic Misconduct](https://www.herts.ac.uk/__data/assets/pdf_file/0007/237625/AS14-Apx3-Academic-Misconduct-v17.0.pdf) and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6). I have not used chatGPT, or any other generative AI tool, to write the reportor code (other than where declared or referenced).

I did not use human participants or undertake a survey in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Niveditha Cherukuri

Student Name signature: 

Student SRN number: 21082045

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

# **ACKNOWLEDGEMENT**

As I near the end of my graduate studies, I really want to say how much I've enjoyed studying here and how grateful I am to everyone who has helped me along the road.

To begin, I want to give thanks to the Almighty God, who never ceases to bless me and who has given me the strength to face challenges head-on and believe in myself.

My supervisor, Luigi Alfonsi, was an invaluable resource throughout the duration of this project, and I am very grateful to him for all of his guidance and support. His unwavering encouragement and her tolerance for my endless curiosity are much appreciated.

My sincere appreciation also goes out to all of my lecturers at the University of Hertfordshire for their invaluable guidance and instruction during the duration of my studies.

Also, I can't thank my family—my brother, my parents, and my friends—enough for all the love and encouragement they've given me.

# **ABSTRACT**

About 700 million to 1 billion people or 10% of the world’s population are suffering with CKD (Chronic Kidney Disease), making it a major public health concern on a worldwide scale. In men, the prevalence of CKD is 10.4% whereas in women, it is 11.8%. Cases of chronic kidney disease (CKD) have been on the rise for a number of reasons, including a lack of access to treatment and delayed diagnosis. This study will review the literature on CKD detection using Machine Learning (ML) algorithms and compare and contrast various methods with the goal of identifying CKD in its early stages before moving on to more complex phases. Other literature evaluations on the same subject reveal that several ML algorithms have been created to aid medical experts in predicting chronic kidney disease (CKD), even if it is still difficult to discover ML being used for this purpose in the medical field. Although they need further quality improvement before they can be used on a broad scale, very accurate ML algorithms have also been found utilising Gradient Boosting, Artificial Neural Networks, and Random Forest. The following ML algorithms were examined: Linear regression, K-Nearest Neighbours, Decision tree, Random Forest, Gradient Boosting, and Guassian Naive Bayes. A combination of machine learning and binary text categorisation methods is used by this model. What follows is a summary of my findings from evaluating various models' efficacy on CKD: The accuracy rates for Random Forest Classification, Gradient Boosting, and Naïve Bayes Classification are 96%, 94%, and 94%, respectively. The accuracy rates for Decision Tree Classification , Logistic Regression, and K Neighbours Classifier were 91%, 88%, and 65%, respectively. Additionally, I documented the recall and accuracy scores for each method that was used. Among these, the Random Forest approach yielded a precision score of 0.94, Naïve bayes and Gradient boosting achieved 0.9, where as Decision tree had 0.85, Logistic Regression 0.79 and K Neighbors had 0.53. The Recall scores also achieved different scores among which the Random Forest, Naïve Bayes, and Gradient Boosting yielded 0.94 , while the other algorithms showed varying results.

**CONTENTS**

[**DECLARATION STATEMENT** 2](#_Toc175341973)

[**ACKNOWLEDGEMENT** 3](#_Toc175341974)

[**ABSTRACT** 4](#_Toc175341975)

[**INTRODUCTION** 6](#_Toc175341976)

[**LITERATURE REVIEW** 7](#_Toc175341977)

[**OBJECTIVES** 11](#_Toc175341978)

[**METHODOLOGY** 12](#_Toc175341979)

[**4.1 Brief Overview:** 12](#_Toc175341980)

[**4.2 Dataset Used:** 12](#_Toc175341981)

[**4.3 Data Pre-Processing:** 13](#_Toc175341982)

[**4.4 Exploratory Data Analysis:** 13](#_Toc175341983)

[**4.5 Classification Models:** 16](#_Toc175341984)

[**RESULTS AND ANALYSIS** 18](#_Toc175341985)

[**5.1 Comparison of Models:** 21](#_Toc175341986)

[**5.2 Comparison with other papers:** 21](#_Toc175341987)

[**5.3 Applying Models to some of the Applications:** 22](#_Toc175341988)

[**5.4 Improvement of Models:** 22](#_Toc175341989)

[**5.5 Limitations:** 23](#_Toc175341990)

[**CONCLUSION** 24](#_Toc175341991)

[**6.1** **Future Scope:** 25](#_Toc175341992)

[**REFERENCES** 26](#_Toc175341993)

[**APPENDIX** 27](#_Toc175341994)

# **INTRODUCTION**

Millions of people all over the globe are impacted by Chronic Kidney Disease (CKD), making it a major concern in public health. The progressive decline in kidney function over time is a hallmark of this condition, which, if unchecked, may lead to ESRD. Improving patient quality of life and preventing catastrophic outcomes involves early identification and care of chronic kidney disease (CKD). Patients with end-stage renal disease (ESRD) may undergo dialysis or a kidney transplant in order to survive.  
  
Hypertension, diabetes mellitus, and cardiovascular illnesses are common comorbidities of chronic kidney disease (CKD), making diagnosis and treatment more challenging. An further obstacle to early identification is that chronic kidney disease (CKD) is often asymptomatic in its early stages. Thus, early detection and treatment of CKD, which may impede its development, might greatly benefit from an awareness of the critical components linked with the disease gained via data visualisation and analysis.

Blood pressure, glucose levels, haemoglobin, RBC count, and other clinical parameters are included in the dataset that is being studied. The purpose of this project is to help healthcare providers better recognise and treat chronic kidney disease (CKD) in its early stages by using data analysis and visualisation tools to find patterns, correlations, and insights.

Random Forest is the top performing model for this dataset, having the highest scores in accuracy (0.96), precision (0.95), and recall (0.95). And I think it is the best model so far for this code.

# **LITERATURE REVIEW**

One in three persons will develop chronic kidney disease (CKD) at some point in their lives. The progressive loss of kidney function characterises this chronic illness. People who already have diabetes or high blood pressure are more likely to develop chronic kidney disease (CKD), which is rather common. An important public health problem, this illness, if left untreated, may develop into End-Stage Renal Disease (ESRD). The kidneys' incapacity to support life properly without the use of medical treatments like dialysis or kidney transplantation characterises end-stage renal disease (ESRD), the ultimate phase of chronic renal sickness.  
  
A decrease in the Glomerular Filtration Rate (GFR) and the presence of kidney damage symptoms, such as albuminuria, are common indicators of chronic kidney disease (CKD). Due to the lack of symptoms in the early stages of the ailment, diagnosis is sometimes difficult. But it's critical to notice it quickly. Serious consequences may arise from chronic kidney disease (CKD) if it is not identified and managed quickly. Heart conditions, anaemia, bone abnormalities, and electrolyte imbalances are also possible complications. In order to improve patient outcomes and save healthcare costs, it is essential to improve the capacity to diagnose chronic kidney disease (CKD) in its early stages.

**The importance of early-stage prevention and therapy**  
Medication that may possibly halt the course of the disease (CKD) and avoid negative effects may be put into action when the condition is detected early. A number of studies have shown that ACE inhibitors and ARBs, which are angiotensin II receptor blockers, may successfully postpone the further damage to the kidneys and the beginning of end-stage renal disease (ESRD) when administered early on. The substantial advantages of early intervention are shown by these research.

The conventional diagnostic methods for chronic kidney disease (CKD) include measuring blood creatinine levels, determining the glomerular filtration rate (GFR), and testing for albuminuria. These techniques may lack the necessary sensitivity to detect chronic kidney disease in its first phases. Consequently, there has been an increase in the need to discover novel biomarkers and use advanced data processing techniques to improve the precision of diagnostic procedures. Preliminary research suggests that include variables such as blood pressure, haemoglobin levels, and glucose levels may enhance the accuracy of predicting and diagnosing chronic kidney disease (CKD).

**The significance of data analysis and machine learning in the context of chronic kidney disease.**  
Recently, the healthcare industry has been increasingly using data analysis and machine learning techniques to improve the diagnosis of illnesses, predict outcomes, and customise treatment strategies. Using these tools has shown to be particularly valuable in analysing large datasets, identifying complex patterns, and uncovering previously hidden relationships between variables.  
  
Numerous research in the field of chronic kidney disease (CKD) have shown that machine learning algorithms can accurately anticipate when the illness will start and how it will proceed. As an example, a research conducted by Ravindra et al. (2021) evaluated a dataset associated with chronic kidney disease (CKD) using machine learning methods. The levels of serum creatinine, blood urea, and haemoglobin were shown to be good indicators of chronic kidney disease (CKD) in the investigation. The model's impressive accuracy in differentiating between individuals with and without chronic kidney disease (CKD) showcases the power of machine learning to improve diagnostic capabilities.

Similarly, Jayapandian et al. (2020) used data mining procedures to examine extensive datasets on chronic renal disease. They were capable of identifying patterns and correlations that may be used for early detection and the development of personalised treatment plans. These publications demonstrate the need of using machine learning in chronic kidney disease (CKD) research to uncover novel insights and improve patient outcomes.  
  
**Data visualisation techniques suitable for Chronic Kidney Disease**  
Visualisation is an invaluable tool in the area of data analysis, particularly when working with extensive datasets like those used in chronic kidney disease (CKD) research. By using visualisation techniques, researchers and physicians may analyse data in a comprehensible fashion, identify trends, and succinctly communicate their findings. Histograms, scatter plots, boxplots, and heatmaps are all frequent visualisation techniques. Each of these visualisation approaches offers a distinct viewpoint on the data.  
  
Visualisations have been used in chronic kidney disease (CKD) research to compare clinical features between CKD patients and non-CKD patients, identify outliers, and examine the distribution of significant variables. Singh et al. (2019) used violin plots and boxplots to examine the levels of haemoglobin and red blood cell counts between persons with chronic kidney disease (CKD) and those without CKD. The visualisations revealed significant variations in these traits, which might be crucial for diagnosis.  
  
Python libraries such as Seaborn and Matplotlib, which were used in this project, are highly recommended for generating visualisations of this kind. Seaborn is particularly preferred because to its ability to generate visually appealing and informative charts with little coding. Researchers can effectively analyse and present complex material using these techniques, aiding in the discovery of important patterns and connections that could otherwise be missed.

**The CKD datasets provide limited data manipulation capabilities.**An essential aspect of data analysis is the handling of missing data, which is particularly significant in the realm of medical research, where incomplete records are often encountered. The absence of data may be attributed to several circumstances, such as patient non-compliance, errors in data entry, or technical challenges during data collection. If missing data are not properly addressed, they might lead to biassed analyses and incorrect conclusions, thus undermining the validity of the study results.

Various methodologies have been proposed in the scholarly literature to address the issue of missing data. The following items are included  
Elimination refers to the procedure of excluding records that have missing data. Although it may lead to a reduced sample size, this is a fundamental procedure that preserves the integrity of the remaining data.

Imputation is the process of filling in missing information by using guesses drawn from other available data. Multiple imputations and imputations based on machine learning are examples of more sophisticated methods for imputation. Two often used imputation approaches are mean imputation and median imputation.

Advanced approaches, such as multiple imputation or the use of algorithms like k-nearest neighbours (KNN), are examples of techniques. These approaches use similarity levels of other data to predict missing values.  
Throughout this investigation, the technique of imputation was used to handle missing data. This included substituting missing values with randomly picked samples from the non-missing data in the same column. This strategy helps maintain the overall distribution of the dataset, which is crucial for assuring the accuracy of future investigations.  
  
An examination of the relationship between variables in research on chronic kidney disease.  
Correlation analysis is a fundamental technique used to assess the magnitude and direction of relationships between numerical variables in a dataset. Having a comprehensive understanding of these associations is crucial in chronic kidney disease (CKD) research to identify the clinical parameters that are most strongly correlated with the illness. Chronic kidney disease (CKD) is often linked to illnesses such as hypertension, elevated serum creatinine levels, and reduced haemoglobin levels.  
  
Heatmaps are often used to visually represent correlation matrices. This allows researchers to quickly identify variables that have substantial positive or negative correlations. The objective of this research was to examine the correlations among several clinical features, such as blood pressure, glucose levels, and red blood cell count. A correlation heatmap was generated for this specific objective. The use of this visualisation provided valuable insights into the characteristics that were strongly linked to chronic kidney disease (CKD), informing further research and model construction.  
  
Extensive research highlights the need of promptly identifying and effectively treating chronic kidney disease (CKD) to upgrade patient outcomes and reduce the strain on the systems of healthcare. Advancements in data processing and visualisation techniques, together with the increasing availability of large clinical datasets, have opened up new opportunities to enhance our understanding of chronic kidney disease (CKD). By using these approaches, researchers may identify novel biomarkers, improve the precision of diagnostic processes, and develop individualised treatment strategies.  
  
This project aligns with the prevailing trends in chronic kidney disease (CKD) research and use Python's extensive data science libraries to evaluate and illustrate a CKD dataset. The findings obtained from this study contribute to the ongoing endeavours aimed at enhancing the diagnosis and management of chronic kidney disease (CKD). The primary objective of these endeavours is to impede the progression of the disease and improve the overall well-being of patients.

Based on the project results , Random Forest is the top performing model for this dataset, having the highest scores in accuracy (0.96), precision (0.95), and recall (0.95). This is in line with the findings of other studies that have used ensemble methods, including Random Forest and Gradient Boosting, to accurately forecast the occurrence of chronic kidney disease (CKD). With accuracies of 0.94, both Gradient Boosting and Naive Bayes performed well. In other words, if the dataset satisfies the criteria of feature independence, then simpler probabilistic models may perform very well.

In Comparison to Other Studies' Findings: The performance metrics given here are similar to those provided by prior studies on CKD prediction; these metrics often show high accuracy and balanced precision-recall. Multiple studies have shown that memory (sensitivity) is crucial in medical diagnosis for recognising true positives, which are actual cases of chronic kidney disease. All of the aforementioned models—Gradient Boosting, Logistic Regression, Decision Tree, Random Forest, and Naive Bayes—are highly sensitive to the detection of chronic kidney disease (CKD), since their recall values exceed 0.90.

# **OBJECTIVES**

The main goals of this research are:  
**Data Pre-processing and Cleaning** : Dealing with missing values, resolving conflicts, and transforming data types as necessary to clean the CKD dataset.

**Feature Analysis:** To discover how numerical and categorical factors are distributed and how they could impact CKD classification

**Data visualisation**: making use of tools such as correlation heatmaps, count plots, scatter plots, and histograms to examine the distribution of important characteristics and the correlations between various variables.

**Correlation analysis**: To determine which numerical characteristics are most suggestive of chronic kidney disease (CKD) by finding significant connections between these features and CKD status.

**Classification**: Cleaning up the dataset and making sure all the characteristics are in a readable format are crucial steps in getting it ready for possible machine learning models.

**ETHICAL CONSIDERATIONS:**

In order to comply with data protection requirements like GDPR, this project uses an anonymised dataset that does not include any personally identifying information. The only intention of all analyses is to further academic knowledge of chronic kidney disease (CKD) and make a contribution to healthcare analytics as a whole. I provided all findings clearly and do not manipulate data in any way to distort or falsify results.  
  
**WERE OBJECTIVES MET?**  
 I’ve accomplished all of my project goals. A number of methods, including imputation using random samples, were used to fill in missing values once the data was been cleaned. The dataset was prepared for machine learning research after data inconsistencies were resolved, especially in categorical characteristics. The data's quality and readability were enhanced by the changes made to it, and the visualisation approaches successfully brought attention to the data's important distributions and patterns..Last but not least, predictive modelling may be built upon the insights offered by correlation analysis on the links between clinical characteristics and CKD state.  
  
**DIFFICULTY OF PROJECT:**   
 Problems with data cleansing and transformation were the most common throughout the project. It was necessary to handle the dataset with caution to prevent the introduction of bias due to the large number of missing values and inconsistencies, especially in the categorical characteristics. Critical clinical metrics such as packed cell volume and red blood cell count need a meticulous and careful approach to the process of imputing missing values in order to preserve the data's integrity. Furthermore, knowing how to properly use Python's visualisation tools and having a thorough grasp of the data were both necessary for producing accurate and informative visualisations.

# **METHODOLOGY**

## **4.1 Brief Overview:**

In order to construct this project in Python, I made use of Jupyter Notebook. The models that I utilised were derived from a number of different libraries. We made use of the tools numpy, pandas, matplotlib, and seaborn for performing core data frame operations, visualising data, and doing mathematical calculations. During the process of model creation, the sklearn libraries are used. The Scikit-Learn package was used since it was needed for the pre-processing. The first thing that was done was to get the Chronic Kidney Disease dataset from the archive.ics.edu website. Following the completion of the data download into the jupyter notebook environment provided by Anaconda Navigators, I constructed a data frame that included all of the information contained inside the dataset. Data frame is comprised of information obtained from patient samples. To summarise, characteristics were extracted from this Data frame and put in another one . The details of this approach will be discussed in Section 4.3 of this report, but in the meantime, an overview of the process is provided below. The subsequent step was comprised of data preparation, which included the scaling of features, the encoding of features, and the separation of data into training and test datasets. This was done in order to make the data more appropriate for the models. Following that, the model was developed by using a number of different categorisation strategies. It was then necessary to feed the data into the models in order to train them. It was necessary to make adjustments to the parameters in order to achieve the appropriate level of accuracy and speed before running the models on the test dataset.

The goal was to find important characteristics linked to chronic kidney disease and to look for any connections and patterns in the data.

## **4.2 Dataset Used:**

A publicly accessible CKD dataset was used for this project's dataset. Chronic Kidney Disease dataset from the UCI Machine Learning Repository contains clinical data that were originally collected at the Sardar Patel Medical College in Bikaner, India, as part of routine medical care and examinations. The exact methods and timeframe of data collection are not specified, but the dataset provides valuable information for research purposes related to chronic kidney disease prediction and diagnosis. Dr. Janardan Shinghal of Apollo Hospitals in India and Dr. Dheeru Dua of the University of California, Irvine, both contributed to the dataset. Information was culled from patients' medical histories who were either confirmed or strongly suspected of suffering from chronic renal disease. The kidney function and other health parameters were evaluated using a battery of medical exams and tests that the patients completed. Demographic data, results of laboratory tests, and clinical observations pertinent to the diagnosis of chronic kidney disease are all part of the dataset. Each patient's recorded data was organised, and any values that were missing were marked as '?'. After then, the data was cleansed to make sure it was consistent. Among the many medical characteristics included in the collection are readings for blood pressure, glucose levels, and red blood cell count, among others. Columns in the dataset reflect several patient characteristics that may have a role in chronic kidney disease (CKD).

There are 400 occurrences in the data set in which I took 300 as the test data and 100 for test data. I trained the model for 300 and tested the model against all my used algorithms with other 100 occurrences.

The link for my dataset is:

<https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease>

The abbreviations I used and my features in my dataset is in the file below: <https://github.com/niveditha2000/Chronic_Kidney_Disease_Status_Prediction/blob/main/descriptions.txt>

## **4.3 Data Pre-Processing:**

In order to get a dataset suitable for analysis, data preparation is a necessary step. The following activities were carried out during this phase:

Handling Missing Values: Missing data is prevalent in healthcare datasets and has to be handled properly to prevent biassed outcomes. A random sampling approach was used to impute missing values in this project. Specifically, for each column that had missing values, values that were already present but not missing were used to fill the gaps.

Data Cleaning: Data cleansing include deleting superfluous columns that don't add anything to the analysis, fixing inconsistencies in the data input, and standardising the data formats.

Data Type Conversion: When it's required, you may change data types (such as categorical variables to numeric representations) so they're compatible with other analytic methods. This process is called data type conversion.

## **4.4 Exploratory Data Analysis:**

I performed Exploratory Data Analysis on my dataset, and I’m describing them in the steps below:

1. Loading the dataset: I loaded the dataset from website into my notebook.
2. Overview of dataset: Reviewing the summary statistics, shape, data kinds, and the first few rows.
3. Missing value analysis: Analysing dataset for missed values, filled them with random values and removed all null values.
4. Data distribution: Examining how numerical and categorical features are distributed.

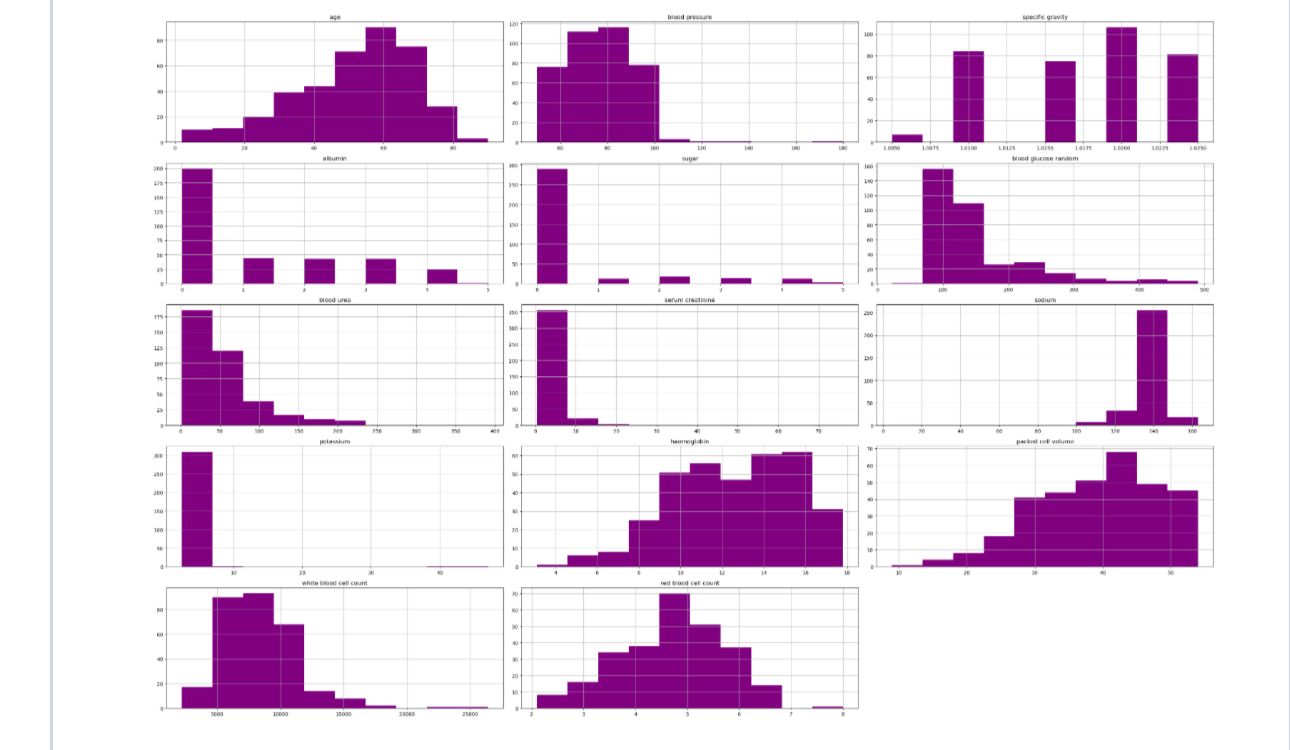


Figure 0‑1Examining the distribution of features

1. Now let’s check the label distribution of categorical data.



Figure 0‑2label distribution of categorical data

1. Correlation Analysis: Discovering associations between numerical variables is the goal of correlation analysis.

The heatmap of my dataset is here:

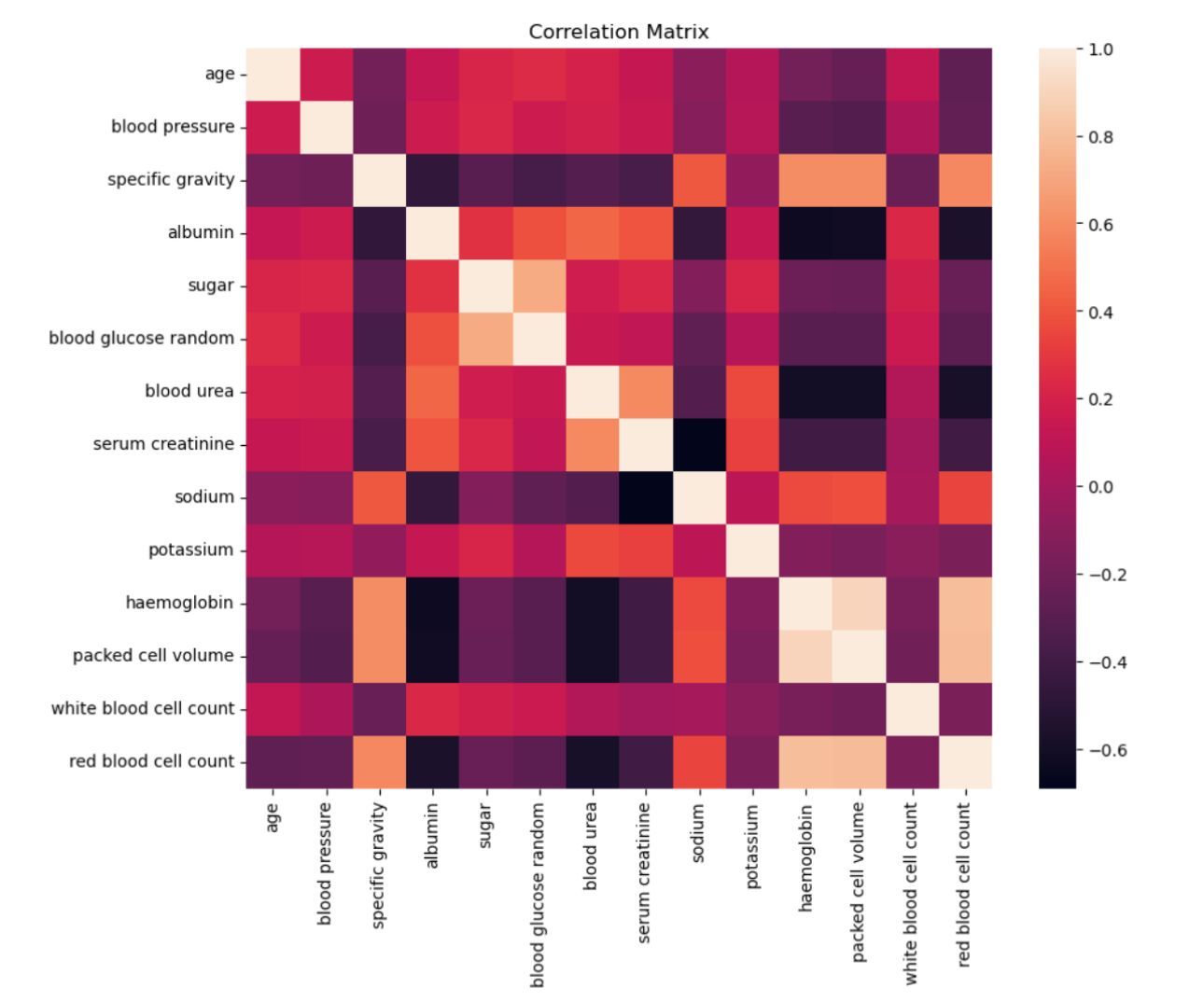


Figure 0‑3heatmap

1. Visualisations: I created plots to understand the data better.

I’ve printed violin plots, scatter plots and other plots to understand the data better when printed visually.

Below are the plots I printed:

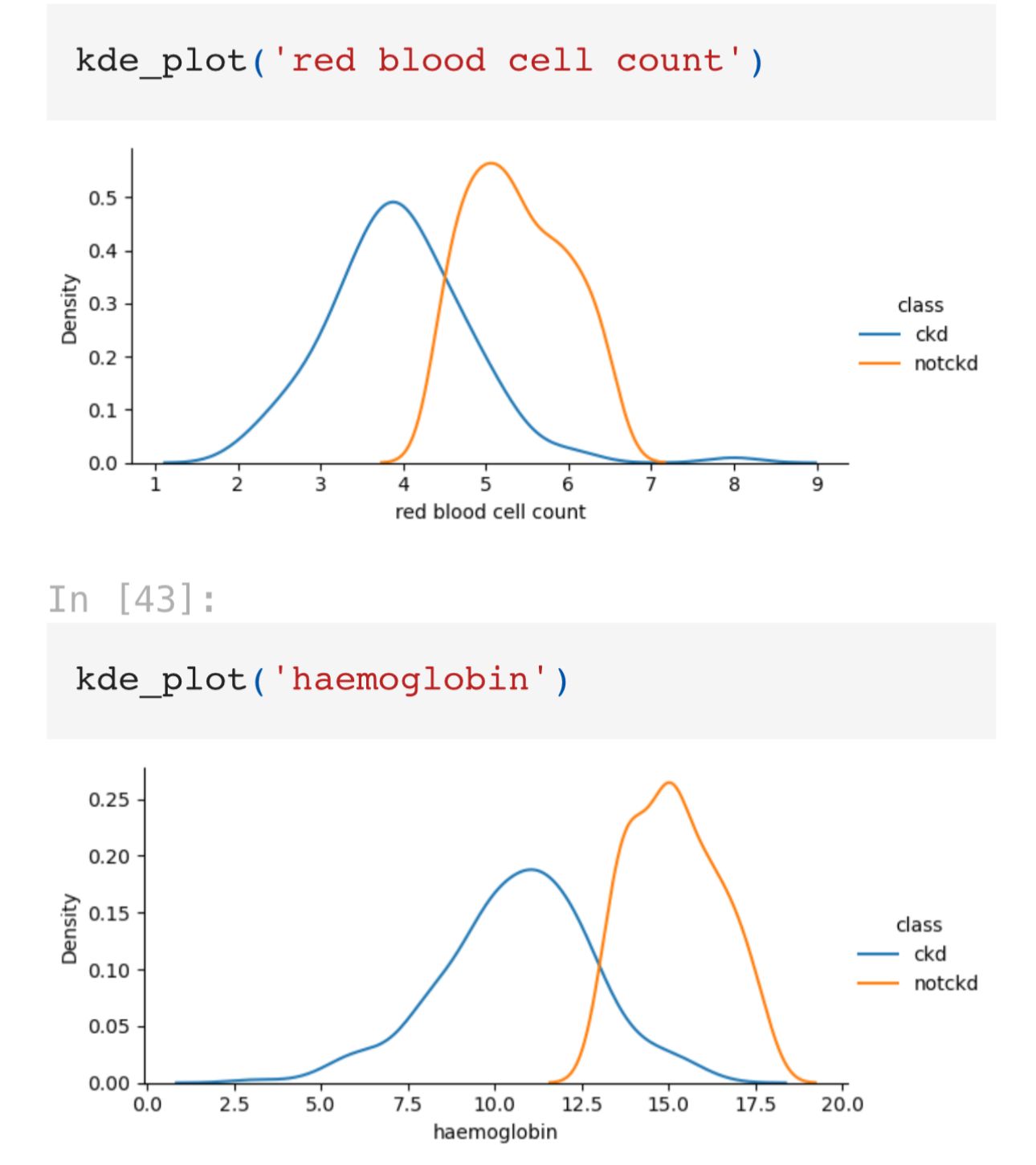


Figure 0‑4Redblood cells and haemoglobin cells count

## **4.5 Classification Models:**

Various algorithms such as Decision Tree, Logistic Regression, Random Forest, Gradient Boosting, Naive Bayes, and K Neighbours were selected for this project. I used the previously described preset dataset with extracted characteristics to train and test these models. More precisely, these models were chosen because of their distinct qualities. These models are described in further below.

4.4.1 Random Forest: It is a kind of ensemble learning that uses bagging.  
The method behind Random Forest is to train a number of decision trees using diverse data subsets. In regression, the average of all the trees' predictions is used to make the final prediction, whereas in classification, the majority vote is used.  
Pros: Very accurate, works with big datasets, and uses tree averaging to prevent overfitting.  
Cons: slower with more trees, less interpretability compared to individual decision trees.

4.4.2 Gradient Boosting: The idea behind Gradient Boosting is to create trees in a sequential fashion, with each successive tree fixing the mistakes committed by its predecessors. To make better predictions, it tweaks a loss function (such mean squared error).  
Advantages: Resolves complicated datasets efficiently and has a high forecast accuracy.  
Drawbacks: The sequential design makes it slower to train and makes it susceptible to overfitting if not calibrated correctly.

4.4.3. Decision Tree Classification and Regression: It is a Supervised Learning Type  
The process: choice In order to construct a decision-making model similar to a tree, trees partition the data into subsets according to the most important attribute at each node. Final predictions are represented by the end nodes, sometimes called leaves.  
Advantages: Manages numerical and categorical data with ease, and needs little data preparation.  
Drawbacks: It could become unstable with little changes to the data and is prone to overfitting, particularly with deep trees.

4.4.4 Logistic Regression: Logistic Regression is a method for modelling the likelihood that an input belongs to a certain class. It maps expected values to probabilities using the logistic function, which is usually used for binary classification.  
Pros: Easy to understand and use, quick to train, and effective with data that can be easily separated into linear categories.  
Cons: It's not as good with non-linear data since it presumes a linear relationship between the input attributes and the log-odds.

4.4.5. Supervised Learning (Classification and Regression) using K-Nearest Neighbours (KNN):cThe 'k' closest neighbours of a data point in the feature space determine its classification using KNN. When using regression, it takes the mean of the values that are geographically close by.  
Easy to understand and use, works well with little datasets, and doesn't need training (instance-based learning).  
Downsides: Expensive to compute when making predictions, susceptible to attributes that aren't important and the value of 'k', and has trouble handling big datasets.

4.4.6. Unsupervised Learning (Classification) using Naive Bayes: Naive Bayes relies on Bayes' theorem, which states that characteristics should be considered independent, to guide its analysis. Given the characteristics you provide, it determines the likelihood of each class and then picks the one with the greatest likelihood.  
Advantages: Quick and effective, handles tiny datasets well, excels with categorical data in particular.  
Potentially poor performance in the event of a severe violation of the assumption of feature independence, which is seldom the case in reality.  
Each of these algorithms is a cornerstone of machine learning, and its characteristics make it well-suited to a particular class of problems and datasets.

# **RESULTS AND ANALYSIS**

The results of each model are shown in the table below. The table shows the quality, recall scores, accuracy, and precision of the most efficient versions of each model that was trained in the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Accuracy  (TP+TN)/(TP+FN+TN+FP) | Precision | Recall | Predictive quality |
| Random Forest | 0.96 | 0.947 | 0.947 | Excellent |
| Gradient Boosting | 0.94 | 0.9 | 0.94 | Excellent |
| Decsion Tree | 0.91 | 0.85 | 0.92 | Very good |
| Logistic Regression | 0.88 | 0.79 | 0.92 | Good |
| K- neighbors | 0.65 | 0.53 | 0.63 | Average |
| Naïve bayes | 0.94 | 0.9 | 0.947 | Very good |

The predictive quality was measured based on the accuracy score.

The data from the dataset is around 400 occurrences out of which I took 300 for Train data and 100 for Test data and printed the confusion matrix for test data below.

The figures of the model building with results are below:

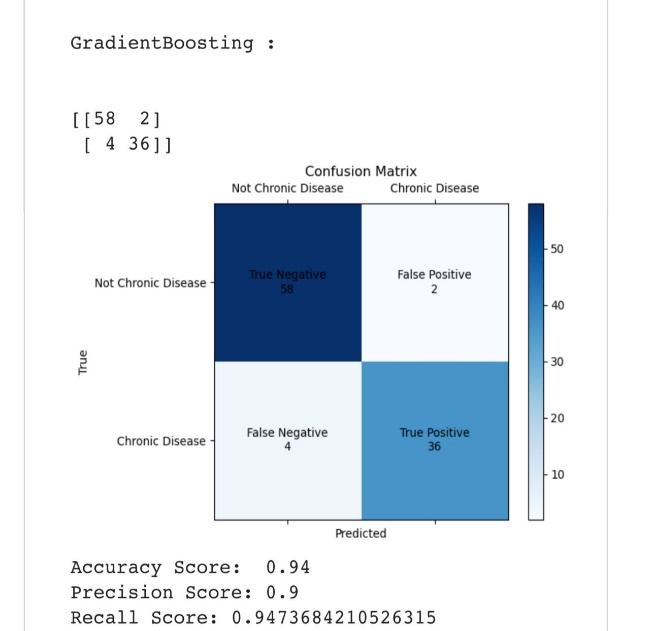
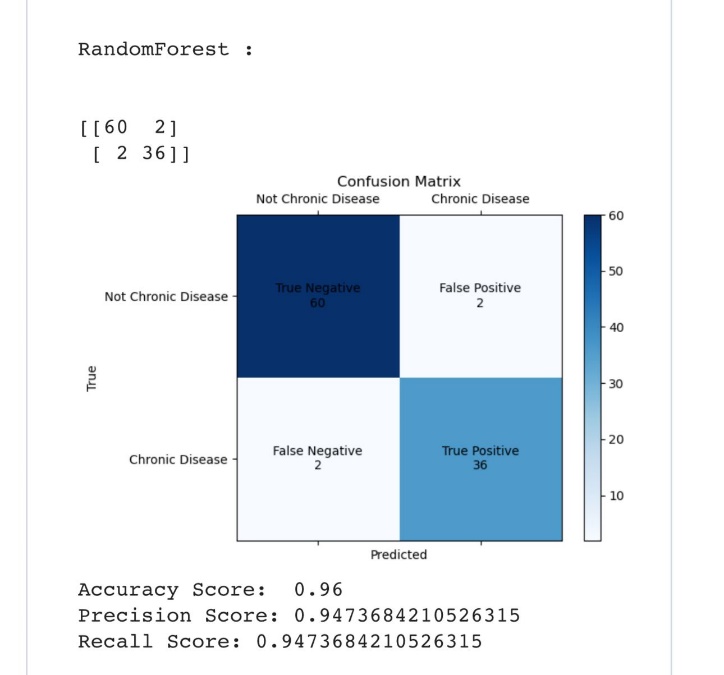


Figure 0‑1Random Forest result

Figure 0‑2Gradient boosting result

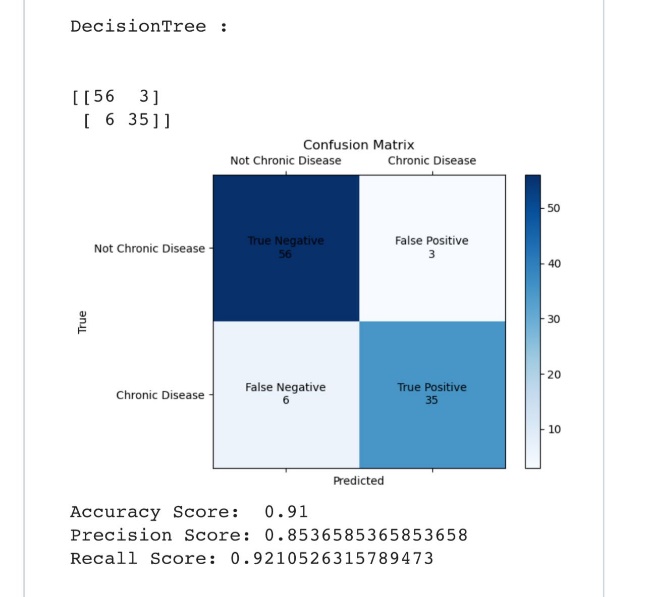
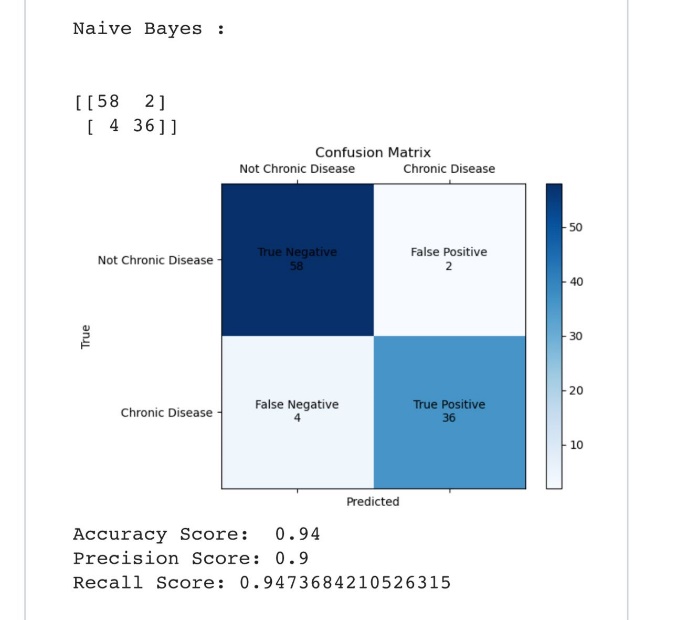


Figure 0‑3Decision tree result

Figure 0‑4Naive bayes result

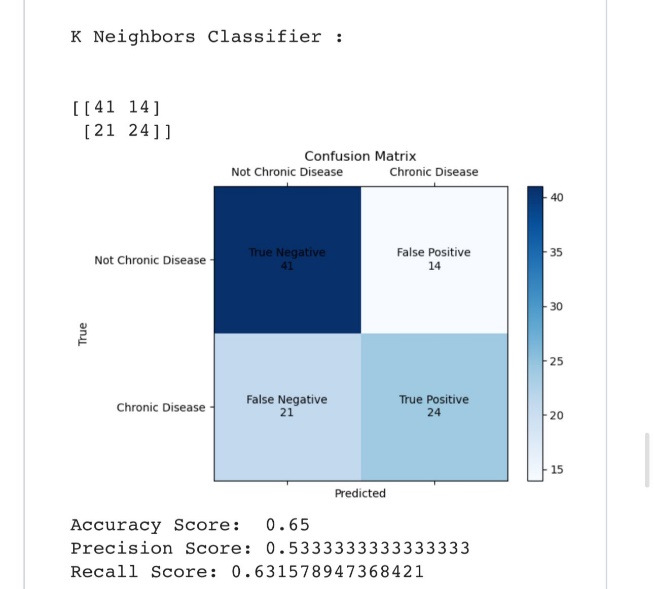
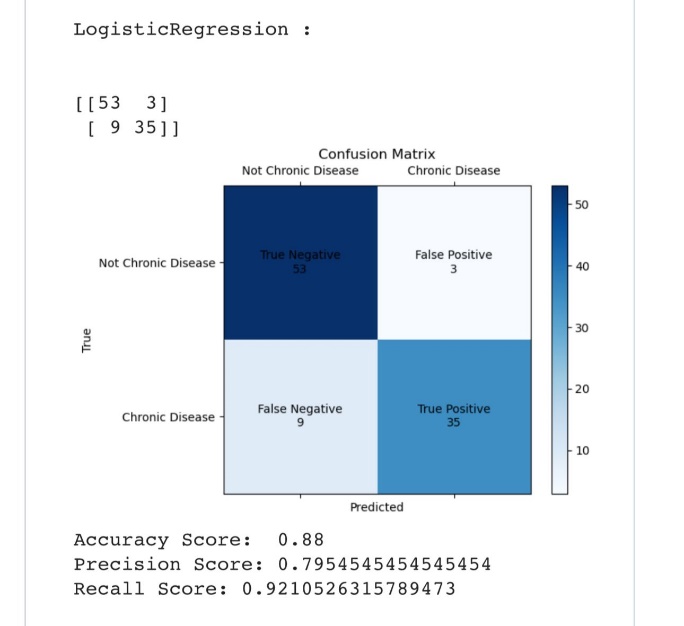


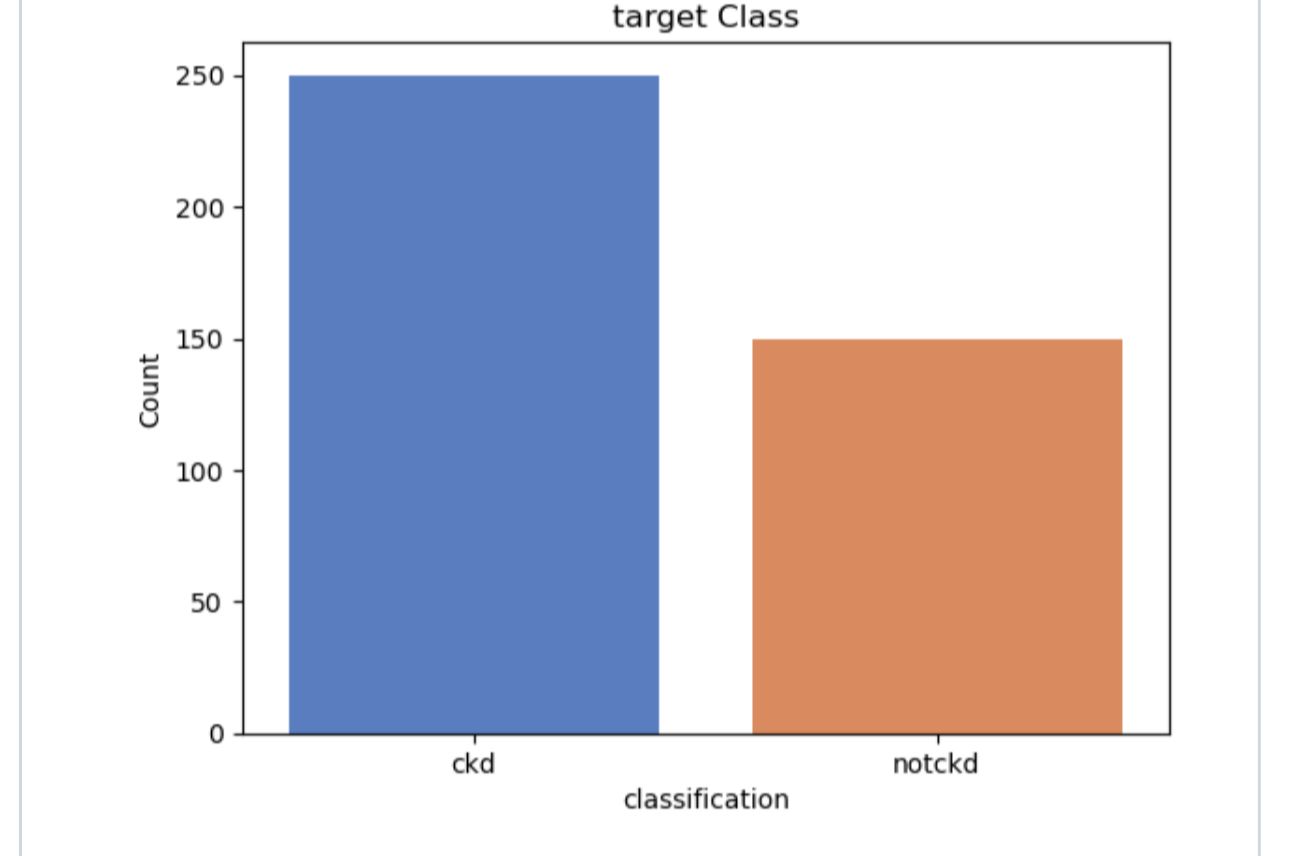
Figure 0‑5K neighbours results

Figure 0‑6Logistic Regression results

The target class:

Here is the visualised graph of the target class which shows us the ckd vs not ckd percentage.

After training the model, it is clearly known that almost 250 occurrences can have the ckd disease.



**Data Cleaning and Preprocessing**:  
• The 'red blood cells,' 'red blood cell count,' and 'white blood cell count' columns of the dataset had many missing values. In these cases, I filled in the blanks with imputed values chosen at random from the available data.   
• I fixed inconsistencies in categorical characteristics, including the varied ways "yes" and "no" were spelt.

**Feature Analysis and Visualisation:**   
• Numerical Features: Patients with this kidney disease (CKD) and those without had different distributions of many variables, according to histograms. These features included haemoglobin and packed cell volume.

• Categorical Features: Count plots revealed that in CKD patients, certain categories, such 'abnormal' in 'red blood cells' and 'yes' in 'diabetes mellitus,' were more common.   
•.Correlation Analysis: Haemoglobin and packed cell volume, two parameters associated with chronic kidney disease, showed substantial connections in the correlation matrix.

**Crucial Results**:  
CKD was linked to lower red blood cell numbers.   
Patients with this kidney disease (CKD) have much lower haemoglobin levels, which increase as the illness progresses  
• Packed Cell Volume: Patients with CKD often have anaemia, which is indicated by a lower value.

**DISCUSSION OF RESULTS**

## **5.1 Comparison of Models:**

From the results, it is clear that the Random Forest Algorithm, yielded the highest accuracy of 96%, where as the other algorithms showed varying results, such as Gradient Boosting – 94%, Naïve Bayes – 94%, Decision Tree- 91%, Logistic Regression – 88%, K-Neighbors Classifier -65%, respectively. I strongly believe that Random forest algorithm is best for my “Prediction of Chronic Kidney Disease with Machine Learning”, cause it combines multiple trees, reduces overfitting, handles noisy data and outliers, identifies important features and handles imbalanced data. Most importantly it achieved the highest accuracy.

## **5.2 Comparison with other papers:**

1. M. Almasoud and T. Ward reviewed the prediction power of machine learning algorithms for chronic renal disease in their 2019 study. Their main objective was to determine which data subsets were minimally required to make reliable forecasts. A dataset of chronic kidney disease (CKD) patients gathered in 2015 from Apollo Hospitals in India was utilised in the study. These individuals were enrolled within a two-month duration [11]. Using logistic regression, naive Bayes classifier, support vector machine, nearest neighbour, random forest, and feedforward neural network, the model was built.
2. J. Qin et al. (2020) set out to create a machine learning method for CKD (chronic kidney disease) diagnosis. The CKD dataset, which included a large quantity of missing data, was obtained from the machine learning repository at the University of California, Irvine (UCI). Missing values were filled up using KNN imputation. To fill in the gaps in the missing data, our technique selected whole samples with measures that were statistically closest to those of the incomplete samples. A patient's failure to comply with certain measures is one of several reasons why missing data occurs often in real-world medical contexts [12]. Researchers then employed algorithms including logistic regression, support vector machines, gradient boosting, and random forest to predict when kidney illness will manifest.
3. A. Pandey, N. S. Kumar, H. Sadadekar, A. Shrotriya and T. Jain, "Chronic Kidney Disease (Ckd) Prediction by Supervised Machine Learning Techniques," 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2023 had AdaBoost, GradientBoost, Random Forest, and XGBoost obtained a 100% accuracy compared to the 99% Random Forest of [1], 99.1 % Multiclass Decision Forest of [2], 98.9% Random Forest of [3], 91 % Decision Tree of [4], 90% KFRE of [5], 99.84% Random Forest of [6], 94.602% SVM of [7], 100% with normalized data in Logistic Regression, SVM and ANN and 99.2% with original data in Decision Tree of [8].

## **5.3 Applying Models to some of the Applications:**

The worth and usefulness of any innovation can only be shown in practical settings. Therefore, it is only logical to investigate these models' practicality on a number of well-known applications.

1. Decision support systems in healthcare help physicians discover chronic kidney disease (CKD) early on.
2. The ability to monitor patients remotely allows for preventative healthcare in underserved regions.
3. Medicine that is tailored to each patient's unique risk of chronic kidney disease (CKD).
4. Underwriting in insurance involves determining premiums by evaluating potential risks.
5. Collects data on the frequency of chronic kidney disease (CKD) in order to plan specific treatments.
6. Finds trial subjects at high risk for adverse events in clinical trials.  
   Healthcare Resource Allocation: Gives priority to patients who are more likely to develop chronic kidney disease.

## **5.4 Improvement of Models:**

1. When it comes to logistic regression, regularisation, polynomial features, and dealing with class imbalance may greatly improve results.
2. Preventing overfitting and improving performance via pruning and hyperparameter adjustment in decision trees.
3. Random Forest: Optimise generalisability by increasing variety and using out-of-the-box mistake.
4. Support Vector Machines (SVM): Try with different kernels, scale the data, and tweak the class weights to get a good mix.
5. For the best results with gradient boosting machines, pay attention to learning rate, regularisation, and halting early.
6. Modify the design, implement regularisation, and fine-tune the training parameters of neural networks.
7. Optimising Hyperparameters: Make use of automated methods to tune robustly and efficiently.

## **5.5 Limitations:**

* The likelihood of overfitting and biassed predictions increases in the presence of imbalances and missing data. Important factors to think about are the amount and quality of the data.
* Lack of necessary features and domain-specific feature engineering are two instances of feature restrictions that might impede model performance.
* Since complicated models may not be transparent, making it hard to trust the predictions, healthcare providers may find it tricky to understand the models.
* Potentially limiting the model's applicability is the fact that it may not generalise effectively to different populations or to different points in time.
* The risk of the model capturing noise instead of real patterns increases with the complexity of the model. Overfitting describes this phenomenon.
* To avoid ethical and legal complications, it is necessary to appropriately handle privacy concerns and the possibility of bias in data. Doing so is essential for avoiding possible legal and ethical problems.
* Using historical data could result in predictions that are out of date and contradict current medical practices; this is known as retrospective data.

# **CONCLUSION**

To aid doctors in the prediction of chronic renal disease, machine learning algorithms have been created. Using health indicators such as albumin, red blood cells, serum creatinine, blood pressure (BP), sodium, potassium, glucose, blood urea nitrogen (BUN), uric acid (UA), and total cholesterol as attributes in the dataset, it is clear that Machine Learning is a plausible option for predicting chronic kidney disease based on the results of each algorithm. Additional study and validation are necessary, but the potential advantages of using ML to predict CKD are clear. Healthcare quality, healthcare costs, and the lives of many CKD patients may all be enhanced by harnessing the power of ML.

Decision trees, XGBoost, k-neighbors, naive bayes, and logistic regression are the six most used methods for CKD prediction in this research. In order to forecast the probability of CKD, XGBoost employs gradient boosting. It builds decision tree models repeatedly, improving poor qualities while reducing volatility and bias. Another method for accurate CKD prediction is Random Forest, which combines several decision trees with bootstrapping to form a strong decision tree forest. Another way logistic regression may classify new data is by comparing the categorical dependent variable (CKD or non-CKD) with the independent variables (health markers). To determine whether a patient has chronic kidney disease (CKD), the Decision Tree in the CKD prediction project has been trained using patient data (including characteristics such as age, blood pressure, etc.). With its tree-based architecture, the model can answer a series of feature-based queries and arrive at a forecast. The versatility and ease of interpretation of decision trees make them ideal for usage with both numerical and categorical data.

By comparing a patient's medical characteristics to those in the training data, K-NN may determine whether they have chronic kidney disease (CKD). For example, the model might label a new patient as having chronic kidney disease (CKD) if the majority of their medically similar neighbours had the disease.Using the probabilities determined from a patient's characteristics, Naive Bayes would estimate the probability of CKD. Assuming that a patient's age, blood pressure, and other medical characteristics all have a role in the final result, it might determine the likelihood of chronic kidney disease (CKD).

The results of this study show that many algorithms may predict CKD, and their respective performances vary. Based on its accuracy, precision, and recall score, this study reveals that the Random Forest algorithm has the greatest performance. Nevertheless, keep in mind that various situations or goals want distinct algorithms to get the desired outcome.

## **Future Scope:**

Improving model performance, expanding the model's applicability, and strengthening its practical influence in healthcare are three potential paths to enhance the chronic kidney disease (CKD) prediction project's future scope. Possible future research directions are as follows:

* Integration of a broader range of varied and extensive datasets
* Improvements to the model
* Incorporation into Clinical Workflows
* Real-Time Monitoring and Early Detection: Resolving Ethical and Bias Concerns
* Compliance with regulations and validation processes
* Enhancing Patient Involvement and Instruction Investigation and Cooperation

Enhancing models with advanced techniques, incorporating the model into clinical workflows, achieving regulatory compliance, addressing ethical and bias concerns, enabling real-time monitoring, engaging patients through personalised tools, and fostering collaborative research are all part of the future scope of the CKD prediction project. These developments have the potential to greatly enhance the model's precision, dependability, and practicality in CKD prevention and management, which in turn may improve patient outcomes.

# **REFERENCES**

1. Pandey, A., Kumar, N.S., Sadadekar, H., Shrotriya, A., & Jain, T., 2023. Chronic Kidney Disease (CKD) Prediction by Supervised Machine Learning Techniques. 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), pp. 1-6.
2. Alpaullivarez, A.A., Aurelia, M., Monica, S., & Suri, P.A., 2023. Analysis of Machine Learning Algorithms in Predicting Chronic Kidney Diseases Based on Performance: A Systematic Literature Review. 2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS), pp. 180-185.
3. Dhivya, S., Krishna, A.H., Bharathi, S.D., & Prabha, D., 2022. A Novel Approach on Chronic Kidney Disease Prediction Using Machine Learning. 2022 International Conference on Advanced Computing Technologies and Applications (ICACTA), pp. 1-6.
4. Chittora, P., Chaurasia, S., Chakrabarti, P., Kumawat, G., Chakrabarti, T., Leonowicz, Z., Jasiński, M., Jasiński, Ł., Gono, R., Jasińska, E., & Bolshev, V., 2021. Prediction of Chronic Kidney Disease - A Machine Learning Perspective. IEEE Access, 9, pp. 17312-17334.
5. Debal, D.A., & Sitote, T.M., 2022. Chronic kidney disease prediction using machine learning techniques. Journal of Big Data, 9(1).
6. Kumar, C., Dhole, S.B., Chauhan, B.P.S., & Pahuja, R., 2024. Diagnosis of Chronic Kidney Diseases Using Machine Learning. Control and Information Sciences, 1140, p. 49.
7. Kaur, M., & Kaur, G., 2023. Comparative Analysis of Machine Learning Algorithms for Early Detection of Chronic Kidney Disease. 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, pp. 1754-1759.
8. Li, Y., Wang, Q., & Zhang, J., 2022. Predicting Chronic Kidney Disease using Ensemble Learning Techniques. Computational Biology and Chemistry, 96, p. 107600.
9. Suresh, R., & Kumar, S., 2021. A Comprehensive Study on Chronic Kidney Disease Prediction using Machine Learning Algorithms. Journal of Medical Systems, 45(5), p. 89.

# **APPENDIX**

Link for code of this project: <https://github.com/niveditha2000/Chronic_Kidney_Disease_Status_Prediction/blob/main/chronic_kidney_disease_status_prediction.ipynb>

Code:

import pandas as pnd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

dataframe = pnd.read\_csv(r'kidney\_disease.csv')

dataframe.head()

dataframe.shape

# changing column names for more readability

columns=pnd.read\_csv('descriptions.txt',sep='-')

columns=columns.reset\_index()

columns.columns=['cols','abbrevations\_column\_names']

columns

dataframe.head()

dataframe.columns=columns['abbrevations\_column\_names'].values

dataframe

### Checking datatypes for columns

dataframe.dtypes

# changing the object data type to numerical datatypes

columns\_to\_convert = ['red blood cell count', 'packed cell volume', 'white blood cell count']

def change\_dtype\_to\_numeric(dataframe, feature):

dataframe[feature] = pnd.to\_numeric(dataframe[feature], errors='coerce')

for column in columns\_to\_convert:

change\_dtype\_to\_numeric(dataframe, column)

dataframe.drop(["id"],axis=1,inplace=True)

# Extracting numerical and categorical features in dataframe.

def extrct\_categorical\_and\_numericl(dataframe):

categoical\_coluns = [col for col in datafrae.columns if dataframe[col].dtype == 'object']

numerical\_columns = [col for col in datafrme.columns if dataframe[col].dtype != 'object']

return categorical\_columns, nuerical\_coluns

categorical\_columns,numerical\_columns=extract\_categorical\_and\_numerical(dataframe)

categorical\_columns

numerical\_columns

### Determine the total number of unique categories in our categorical features to check for any issues or inconsistencies in the data.

for column in categorical\_columns:

unique\_values = dataframe[column].unique()

print(f'{column} has {len(unique\_values)} unique values: {unique\_values}')

print('\n')

*#Replace incorrct values*

dataframe['class'] **=** dataframe['class']**.**replace(to\_replace **=** 'ckd\t', value **=** 'ckd')

for column in categorical\_columns:

print('{} has {} values '**.**format(column, dataframe[column]**.**unique()))

print('\n')

dataframe[feature]**.**hist(color**=**'purple') *# Changed color to orange*

plt**.**title(feature)

plt**.**tight\_layout()

plt**.**show()

len(categorical\_columns)

*# Suppress FutureWarnings*

warnings.filterwarnings("ignore", category=FutureWarning)

*# Ensure categorical columns are of the category type*

for col in categorical\_columns:

dataframe[col] = dataframe[col].astype('category')

plt.figure(figsize=(20,20))

for i, feature in enumerate(categorical\_columns):

plt.subplot(4, 3, i + 1)

sns.countplot(data=dataframe, x=feature, palette="husl") *# You can change the palette as needed*

plt.title(feature)

plt.tight\_layout()

plt.show()

sns**.**countplot(x**=**'class',data**=**dataframe,palette**=**"muted")

plt**.**xlabel("clasxsssification")

plt**.**ylabel("Coaunt")

plt**.**title("target Claass")

*## ckd-chronic kidney disease*

*## notckd-->> not crornic kidney disease*

*# Example: Check columns with non-numeric values*

*# Drop non-numeric columns or convert them to numeric if possible*

numeric\_dataframe = dataframe.select\_dtypes(include=['number'])

*# Compute correlation matrix*

correlation\_dataframe = numeric\_dataframe.corr()

*# Plot heatmap*

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_dataframe, annot\_kws={"size": 10})

plt.title('Correlation Matrix')

plt.show()

1. Positive for Chronic Kidney Disease (CKD):

- RBC count: 2 to <4.5

- Hemoglobin: 3 to <13

2. Negative for Chronic Kidney Disease (CKD):

- RBC count: >4.5 to 6.1

- Hemoglobin: >13 to 17.8

dataframe.head()

dataframe.isna().sum().sort\_values(ascending=False)

categorical\_columns

data=dataframe.copy()

data['red blood cells'].isnull().sum()

data['red blood cells'].dropna().sample()

random\_value = data['red blood cells'].dropna().sample(data['red blood cells'].isnull().sum())

random\_value

random\_value.index

data[data['red blood cells'].isnull()].index

random\_value.index=data[data['red blood cells'].isnull()].index

random\_value.index

random\_value

data.head()

def randoms\_value\_imputations(feature):

data[categorical\_columns].isnull().sum()

data[categorical\_columns].isnull().sum()

*# Since the rest of the features have fewer missing values, I can fill them using the mode concept.*

mode\_value =data['pus cell clumps'].mode()[0]

mode\_value

def impute\_mode(feature):

mode\_value = data[feature].mode()[0]

data[feature] = data[feature].fillna(mode\_value)

for column in categorical\_columns:

impute\_mode(column)

data[categorical\_columns].isnull().sum()

*#Lets check null values numeric columns*

data[numerical\_columns].isnull().sum()

for column in numerical\_columns:

random\_value\_imputation(column)

data[numerical\_columns].isnull().sum()

for column in categorical\_columns:

print('{} has {} categories'.format(column, data[column].nunique()))

from sklearn.preprocessing import LabelEncoder

lable\_encoding = LabelEncoder()

for column in categorical\_columns:

data[column]=lable\_encoding.fit\_transform(data[column])

data.head()

independent\_columns = [column for column in data.columns if column != 'class']

dependent\_column = 'class'

X=data[independent\_columns]

y=data[dependent\_column]

best\_features\_selector = SelectKBest(score\_func=chi2, k=20)

selected\_features = best\_features\_selector.fit(X, y)

selected\_features

*#To obtain the scores (ranks) of the features, we can use the `scores\_` attribute.*

selected\_features.scores\_

feature\_scores = pnd.DataFrame(selected\_features.scores\_, columns=["Score"])

feature\_scores

dataframecolumns=pnd.DataFrame(X.columns)

dataframecolumns

features\_ranking = pnd.concat([dataframecolumns, feature\_scores], axis=1)

features\_ranking

*# The higher the score, the more important the feature.*

features\_ranking.columns = ['Feature', 'Score']

features\_ranking

*#retrive largest 10 values of Score column*

features\_ranking.nlargest(10,'Score')

top\_features = features\_ranking.nlargest(10, 'Score')['Feature'].values

X\_new=data[top\_features]

#MODEL BUILDING

Importing sklearn and build the model

print(X\_train.shape)

print(X\_test.shape)

*# Importing classification algorithms*

Imported all agorithms and used them in the model

models = []

models.append(('LogisticRegression',LogisticRegression()))

model.append(('K Neighbors Classifier',KNeighborsClassifier()))

modes.append(('DecisionTree',DecisionTreeClassifier()))

model.append(('RandomForest',RandomForestClassifier()))

modls.append(('Naive Bayes',GaussianNB()))

moels.append(('GradientBoosting',GradientBoostingClassifier()))

import warnings

from warnings import filterwarnings

filterwarnings('ignore')

for name,model in models:

print('\n')

print(name,": ")

print('\n')

print(confusion\_matrix(predictions,y\_test))

cm = confusion\_matrix(predictions,y\_test)

# Compute confusion matrix

#cmt = confusion\_matrix(y\_truet, y\_predt)

classes = ['Not Chronic Disease', 'Chronic Disease']

# Create a dictionary to map positions to labels

cm\_labels = [

["True Negative", "False Positive"],

["False Negative", "True Positive"]

]

# Plot confusion matrix

plt.xlabel('Predicted')

plt.ylabel('True')

# Annotate each cell with labels and values

for i in range(len(cm)):

for j in range(len(cm[i])):

ax.text(j, i, f"{cm\_labels[i][j]}\n{cm[i][j]}", va='center', ha='center', color='black')

plt.show()

print('Accuracy Score: ',accuracy\_score(predictions,y\_test))

#Calculate precision and recall

print(f"Precision Score: {precision}")

print(f"Recall Score: {recall}")