Enhancing Chronic Kidney Disease Prediction through Machine Learning

MSc Research Project
Data Analytics

Revanth Vijay Kumar
Student ID: x21218374

School of Computing
National College of Ireland

Supervisor: Dr. Syed Muslim Jameel

School of Computing
Project Submission Sheet

Student Name:	Revanth Vijay Kumar
Student ID:	x21218374
Program:	Data Analytics
Year:	2023
Module:	MSc Research Project1
Supervisor:	Dr. Syed Muslim Jameel
Submission Due Date:	14/08/2023
Project Title:	Enhancing Chronic Kidney Disease Prediction through Machine Learning
Word Count:	6937
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Revanth Vijay Kumar
Date:	12/08/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Chronic Kidney Disease Prediction through Machine Learning

Revanth Vijay Kumar
x212128374

Abstract:

CKD affects millions of people worldwide and causes symptoms such as heart disease, kidney failure. To help people with CKD, it is imperative to diagnose it early, and begin treatment as soon as possible. The goal of this study is to examine existing approaches for imputing missing data from healthcare datasets, as well as to use machine learning models to automate the prediction and analysis of chronic kidney disease. The first step is to examine the different methods that researchers have used to fill in missing data from medical datasets. The next step will be to compare algorithms such as decision trees, K-nearest neighbors, and its variations to determine which is the best at predicting CKD progression. Thus, it is important to identify the most accurate model for forecasting the outcomes. Based on previous studies, it appears that automated machine learning may dramatically improve the precision of CKD prediction. For prediction and analysis, this process can be applied to any binary-classification problem.

1.Introduction

Chronic kidney disease (CKD) represents a significant global health issue. The early diagnosis and treatment of CKD can lessen disease severity and mitigate future complications. Machine learning models show promise for identifying at-risk populations.

Incomplete healthcare data presents both challenges and opportunities for medical professionals. A major challenge lies in effectively analyzing datasets containing missing values, as this can skew results and hinder accurate insights. Techniques employed for addressing missing gaps include data imputation and data augmentation.

Data augmentation offers an intriguing approach to developing enhanced predictive models for various diseases. Early diagnosis of many conditions proves difficult due to limited sample sizes within medical datasets resulting from challenges obtaining patient information. However, data augmentation allows artificial generation of synthetic samples while maintaining the original distribution and trends. This permits training machine learning algorithms on more extensive, nuanced datasets. Specifically, SMOTE methods prove especially useful for rare diseases by oversampling minority classes to achieve balance. Addressing missing values and slightly modifying existing records via operations such as data imputation further expands informative sample sizes, respecting ethical and privacy concerns.

Algorithms were evaluated to determine the most accurate predictor of kidney disease progression. Decision trees, K-nearest neighbors, Bernoulli's Naive Bayes, and related techniques were assessed based on criteria like precision, efficiency and scalability. Establishing a robust prognostic tool could meaningfully impact patient outcomes and resource management. With earlier CKD prediction, expensive therapies may be preemptively avoided. Moreover, automated solutions allow clinicians to focus on care over data processing.

In closing, this study applied various machine learning and data science techniques to progress prediction and management of CKD. Further research stemming from this work holds potential to improve diagnosis and treatment of this widespread condition.

1.1 Research Question

Analysing how well machine learning models classify the stages of chronic kidney disease, and how automated dataset imputation and algorithm choice increase accuracy and generalization performance?

1.2 Research Objectives

1. Studying the different factors that develop chronic kidney disease (CKD) over time.
2. The goal is to employ machine learning models to forecast the clinical progression of CKD for patients.
3. One challenge lies in medical records containing missing information. Gaps in the data could skew results and hinder developing a comprehensive understanding.
4. To determine the most precise predictive model, common machine learning techniques will be compared - decision trees, k-nearest neighbours, Naive Bayes and related variants. Models will be evaluated based on criteria like accuracy, efficiency and scalability to extensive datasets.
5. Medical data requires substantial resources to compile due to privacy concerns. However, data augmentation can assist by generating synthetic samples while preserving the original distribution patterns.

The following paper is organised as following pattern, section 2 provides the review of previous work on this domain (literature review), section 3 discusses the Research Methodology Approach used, section 4 discusses the Chronic Kidney Methodology Approach, section 5 & 6 discusses about the Implementation and Results of models, section 7 compares and analyses the models used, finally section 8 concludes the paper.

2. Literature Review

Millions of people throughout the world suffer with CKD, commonly known as chronic kidney disease. It's a major issue that affects a lot of people. It's crucial to identify CKD as early as possible and forecast how it will develop over time to assist patients and prevent further health difficulties in the future. Machine learning algorithms can help predict a person's CKD. These algorithms succeed in analysing large volumes of data, recognizing patterns, and making accurate predictions. In this study, I have looked at previous research that has utilized machine learning to forecast how CKD will progress for specific people. While large healthcare databases can contain a wealth of useful data, they may also contain gaps. How other researchers have attempted to close those gaps also piques my attention. (Charumathi Sabanayagam et al 2022) In this paper, the use of machine learning algorithms to anticipate the people who may develop diabetic kidney disease in the future is researched. For six years, the researchers monitored a group of Asian people who had diabetes but no renal problems. Over time, about 12% of people had it, to determine which machine learning (ML) method best predicted a person's risk based on variables like age, blood sugar management, cholesterol levels, etc., they tested out a variety of ML algorithms, including elastic net and neural networks. The fundamental approach now used by doctors was outperformed by the elastic net model. Additionally, it's discovered that several brand-new predictors, such as sugar and fat-related metabolites. This demonstrates how ML could identify individuals who should be evaluated for chronic kidney disease in advance of the need for preventative medications. Applying comparable prediction models and feature selection techniques to other diseases, as in my own study. To identify which hospital patients are shortly ready to return home, the researcher used a neural network, a sort of AI (K. C. Safavi et al 2019). Over 15,000 surgical patients' medical records were used to train the system. The program used an extensive amount of information, including medication records, vital signs, test results, and nurse notes, to predict which patients will be discharged in the upcoming 24 hours. Compared to simply looking at how long most patients who have a particular surgery typically remain, it was fairly accurate. When the researchers used it, they discovered that it could identify patients who ought to have gone home but instead got stranded. Many were kept in hospitals for administrative or other non-medical reasons. A neural network was developed by the researchers to forecast the progression of chronic renal disease (Md. Rashed-Al-Mahfuz, A. Haque, A. Azad, S. A. Alyami, J. M. W. Quinn, and M. A. Moni, et al 2021). To train it, they used the medical records of more than 400 individuals. Age, blood pressure, lab results, and medications were considered by the AI to determine who will eventually develop advanced renal failure. To improve accuracy, they experimented with several neural net topologies. The most accurate one, which correctly identified which patients became worse 96.8% of the time, comprised three hidden layers and performed significantly better than a logistic regression model. The neural network also exhibited a low number of false alarms. This demonstrates that these AI models are capable of accurately projecting the progression of ailments such as renal disease using data from patient records. To determine whether individuals with chronic kidney disease might deteriorate over time, the researchers examined various machine learning algorithms (Md. A. Islam, S. Akter, Md. S. Hossen, S. A. Keya, S. A. Tisha, and S. Hossain, et al 2020). They examined information such as age, blood pressure and lab results using the medical records of more than 1,700 patients. They have developed techniques like neural networks and random forests to predict which individual would get kidney failure in five years. With a success rate of 83%, the neural network outperformed the rest. It was discovered that old age and hypertension were the most crucial elements for the forecasts.

To help physicians comprehend what powers the forecasts, scientists made the AI models simple to understand. To identify HIV patients who may later develop chronic kidney disease, the researchers tested various machine learning algorithms (J. A. Roth et al 2020). They trained the AI using medical records from over 12,000 participants in a large Swiss research study. To calculate a person's likelihood of seeing a decline in kidney function within the following several months to a year, the models examined a vast amount of data, including demographics, lab results, and medications. Compared to a traditional statistical approach, models like neural networks and random forests performed incredibly well. The most effective ones were able to predict who would get chronic renal disease more than 95% of the time. Reviewing several research that attempted to use machine learning to forecast the course of chronic kidney disease, (Sanmarchi, F., Fanconi, C., Golinelli, D., Gori, D., Hernandez-Boussard, T. and Capodici, A, et al 2023) this paper reviews their results. The researchers discovered that there is some encouraging evidence demonstrating that AI can predict who may experience worsening CKD. But there are also some significant problems. Many the studies used incredibly little patient data or evaluated their models in questionable methods. Additionally, the medical information that was provided to the algorithms varied greatly between research. To make accurate predictions, it can be challenging to determine which machine learning method or dataset is ideal. To determine the ideal ML algorithm and data combination to accurately predict how CKD will develop, the authors stated that more research is unquestionably required. Number of Machine Learning algorithms was put to the test by the authors to see how effectively they could forecast which person having chronic renal disease will eventually develop kidney failure (Su, C.-T., Chang, Y.-P., Ku, Y.-T. and Lin, C.-M., et al 2022). Experiments included techniques like support vector machines, decision trees, and random forests with access to the medical information of more than 1500 patients with CKD. As per to the result of this paper, Random Forest had the highest accuracy and AUC_ROC score. However, researchers also highlighted some significant drawbacks, such as missing data and the dataset's lack of specific traits. Thus, it is difficult to say if random forests will perform similarly on other patient groups.

The next paper discusses the ways to address the significant problem of missing data in healthcare databases (Phung, S., Kumar, A. and Kim, J., et al 2019). The authors emphasize that there are many forms of missing data, such as missing completely at random or missing based on other variables. These methods include mean imputation, which uses the average value to fill in missing variables, also more advanced techniques like regression imputation and multiple imputation. Considering factors including the amount of missing data and the intended use of the dataset, researchers are advised to carefully consider which strategy will be most effective. The authors recommend selecting an imputation approach rather than a conventional choice with extreme caution. Variety of methods for filling in or "imputing" missing data in healthcare databases (Chowdhury, M.H., Islam, M.K. and Khan, S.I., et al 2020). Simple techniques like mean imputation, which uses the average to fill in for missing information, were put to the test in the study, also explored more sophisticated techniques like regression imputation and multiple imputation. The study used parameters like sensitivity and specificity to compare the accuracy of each procedure. In general, it was discovered that multiple imputation techniques outperformed single imputation techniques. Furthermore, approaches based on regression performed better than other kinds. Using this research will help me choose the most effective imputation strategy. Using a large database of over 7000 patients, the scientists evaluated several machine learning algorithms to see if they could predict which person has the more mortality rate, following the heart surgery (Y. Yu et al 2022). Experiment was done with several AI techniques, such as neural networks and decision trees, and trained them using data collected immediately following surgery, such as test results, medicine, and vital signs. The model that worked best in the end was AdaBoost; it predicted people's deaths within four years over 80% of the time. The factors that were considered including red blood cell diameter, renal tests, and scoring systems were crucial for the forecasts.

Utilizing deep belief networks to automatically complete blank fields in patients' electronic health records (BEAULIEU-JONES, B.K. and MOORE, J.H. et al 2016) . Deep Learning is an example of AI. The study found out that, this method outperformed others in handling problems including missing values, noisy data, and complex interactions between variables. On a dataset related to healthcare, deep belief network approach was applied to the test. The disadvantage to this was, they only tested their strategy on a single dataset and did not evaluate it against other approaches already in use. Therefore, it is difficult for us to say for certain if deep belief networks are superior for inputting all kinds of healthcare data. The rapid use of deep learning, a form of artificial intelligence, in medicine and healthcare (S. Mittal, Yasha Hasija, et al 2020). Research is on how deep neural networks can detect patterns in complex health data such as medical pictures, genetics, lab tests, and so on. The research provides numerous examples, including the use of deep learning to decode scans, predict protein structures, examine DNA, RNA, etc. Based on the unique facts of individual patient, the study shows that the AI models could enhance diagnosis, treatment, and forecasts. Some of the medical data are scattered, uneven and missing, to fill in these gaps in the changing medical records, this paper focuses on a deep learning model called a generative adversarial network. In this, one neural network is used to create the missing numbers, while another is used to determine how realistic they appear to be (Y. Zhang, B. Zhou, X. Cai, W. Guo, X. Ding, and X. Yuan et al 2021). To initialize the generative model, encoder is used to compress the missing patient data into a summary vector. This improved the degree to which the filled-in values matched the actual ones. To improve the consistency of the outcomes, devised a method of combining created and real data during training. The study's methodology outperformed other approaches at accurately imputed missing data on three health datasets, according to the results. To predict which ICU patients are at risk, based on their medical histories (H. Jiang, C. Wan, K. Yang, Y. Ding, and S. Xue et al 2021), the researchers experimented with a variety of machine learning algorithms. Models like logistic regression, random forests, and neural networks was trained using data from the starting few days in the ICU, such as vital signs and lab results. Random Forest Algorithm predicted the risk of deaths with 81% as the highest accuracy compared to other algorithms. Creating artificial intelligence (AI) methods to fill in blanks in sensor data from a bridge that continuously monitors its condition (J. Hou et al 2022). It also measures gaps when sensors malfunction. Deep Learning networks like GANs and LSTM were used to model the link between sensors and to predict the missing values. While LSTM performed better with various types of sensors, GANs performed better with similar sensor types. Self-supervised learning, a novel AI technique, which enables models to train on medical data without requiring all the labels and annotations made by human specialists (R. Krishnan, P. Rajpura, and E. J. Topol et al 2022), without the need for supervised instruction, methods like different learning and generative pre-training can identify patterns in data sets like DNA sequences, hospital records, and photographs. By more effectively utilizing all that untapped data, the researchers believe self-supervision has a lot of promise to improve healthcare AI. However, it should be noted that to not assuming the diversity of patients in real-world settings will be reflected in the training data. To analyse how autoML, which automates machine learning using AI, has been used in healthcare (Mustafa, A. and Rahimi Azghadi, M. et al 2021). The paper discusses the development of autoML, its applications. Also, how autoML has aided in the decision-making of medical professionals for the discovery of new medications and the analysis of medical images. The goal of the autoML approach is to make and using ML models much easier. Also discussed in the paper is the need to carefully consider the ethics as AI plays larger roles in healthcare. Based on the data from brain scans, the researchers have experimented with utilizing autoML to identify autism spectrum disease (Subah, F.Z., Deb, K., Dhar, P.K. and Koshiba, T. et al 2021). When compared to previous research which just used one machine learning strategy, they intended to increase the accuracy of autism prediction. To identify crucial features in the brain images and train prediction models, autoML system used various AI techniques. The autoML approach could discover the ideal feature selector and model combination for classifying autism by automating the testing of numerous feature selectors and models. In this paper, also discussed about how previous research was constrained by utilizing just one machine learning technique and not testing the models with fresh external data. Using a deep learning model called an autoencoder to fill in blanks in time series records, such as measurements of the air quality over time (S. Singh, S. Sharma, and S. Bhadula, et al 2022). Recurrent layers are used in the model to learn long-term dependencies, whereas convolution layers are used to learn local patterns. As, this enables it to forecast the missing values by modelling both closest and farthest relationships. In comparison to other straightforward approaches like just replacing the missing data with the mean, the autoencoder model performed significantly better in terms of accuracy. A recurrent neural network model was employed in this paper to fill in missing values from medical time series data, which are measurements of things like a patient's vital signs over time (J. Aswini, B. Yamini, Rajaram Jatothu, K. Sankara Nayaki, and M. Nalini, et al 2021). To concentrate on relevant time steps and enhance the imputations, architecture used adversarial training and attention layers. On actual clinical data, such as incomplete ECG readings, the test was conducted. RNN model outperformed when it came to accurately completing the time series. Additionally, RNN has increased the precision of models that used the filled-in data to predict patient outcomes like mortality.

3. Research Methodology Approach

The standard process for an ML research project is shown in the flowchart below: First, data will be gathered and examined, followed by the preparation stage which involves handling missing values and converting categories into numbers. Next, models such as KNNs and Decision Trees will be constructed, and the settings will be altered, cross-validated, and tested using real data. After comparing the models, the best-performing one will be selected, and the outcomes will be examined to understand the predictions. This approach guides the fundamental phases of data.

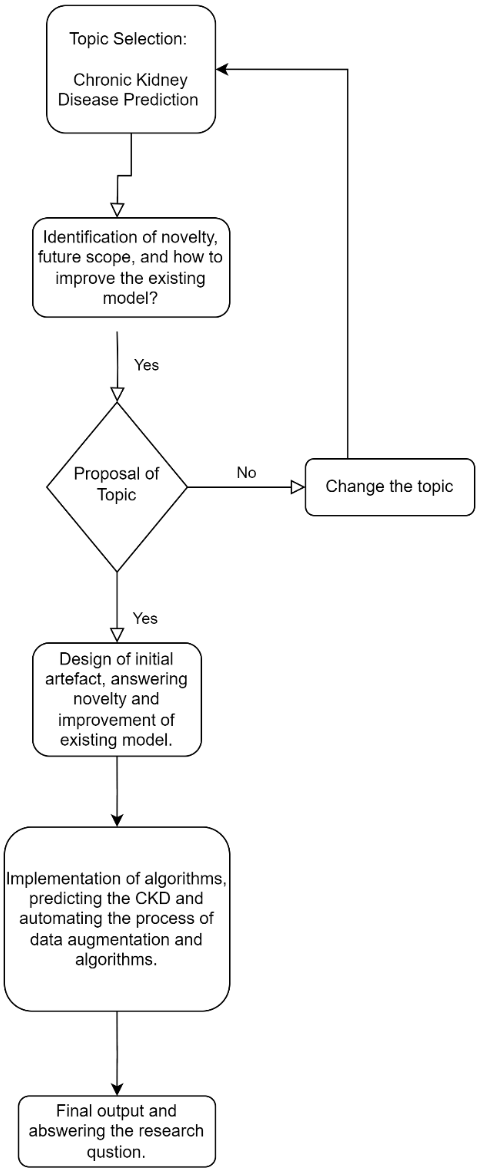

Topic Selection:
Chronic Kidney Disease Prediction
Identification of novelty, future scope, and how to improve the existing model?
Yes
Proposal of Topic
No
Change the topic
Yes
Design of initial artefact, answering novelty and improvement of existing model.
Implementation of algorithms, predicting the CKD and automating the process of data augmentation and algorithms.
Final output and abswering the research qustion.

Figure 1. Research Methodology

4. Chronic Kidney Disease Methodology Approach

Chronic kidney disease (CKD) is a progressive condition that can cause damage to the kidneys. Currently, there is no known treatment or cure for CKD; however, early detection and intervention can help delay the progression of the disease. One option to improve early diagnosis is the development of better tools for predicting CKD stages. Predicting the stages of CKD is challenging due to the complexity of the disease and the absence of a single reliable test. However, machine learning has the potential to enhance CKD predictions. By analysing large quantities of patient data using machine learning techniques, patterns that may otherwise go unnoticed can be identified. My approach to CKD methodology consists of three primary components. First, the data is prepared by filling in any gaps using various approaches to determine the most effective one. The second phase involves developing and testing machine learning techniques such as K-nearest neighbour, decision tree, Bernoulli Naive Bayes, and their variants. The final step is the visualization of the results. Through the creation of graphs and charts, we aim to understand what the models are learning and identify any biases or inaccuracies.

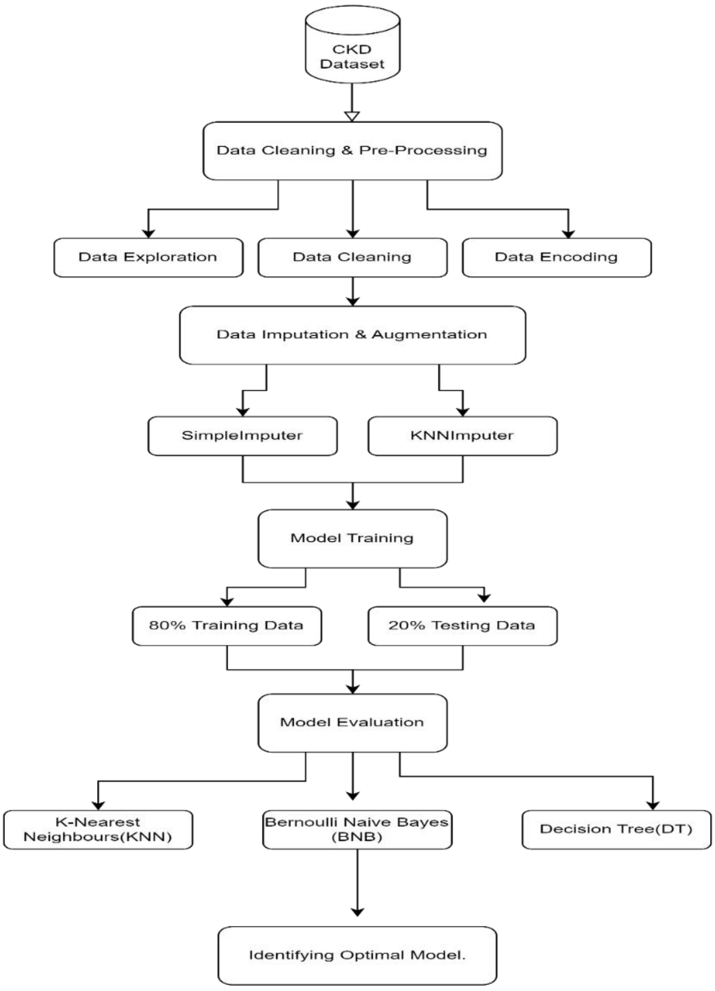


Figure 2. Methodology for Chronic Kidney Disease Prediction.

4.1 Dataset Description

There are a total of 400 rows and 24 columns of clinical data for patients who may or may not have chronic kidney disease (CKD), which needs to be researched/investigated from the kidney disease dataset. The NHI-produced dataset is available in the UCI Machine Learning Repository. The classification variable, which indicates whether the patient has CKD (1), is the primary variable. The remaining 23 columns list the patients' clinical characteristics such as age, blood pressure, albumin, blood sugar, red blood cells, proteinuria, blood urea nitrogen, serum creatinine clearance, serum sodium, serum potassium, haemoglobin, haematocrit, white blood cell count, red blood cell count, history of hypertension, diabetes mellitus, coronary artery disease, appetite, pedal enema, and anaemia. Data Augmentation is used to generate new synthetic data and additional rows were created. The total number of rows after applying SMOTE based Simple_Strategy augmentation is 4000. This dataset is well balanced, with 2000 patients in each of the two classes, CKD or NotCKD. The scatter plot in Figure 3 displays the relationship between the two variables. The distribution of the classifications is shown in the bar chart.

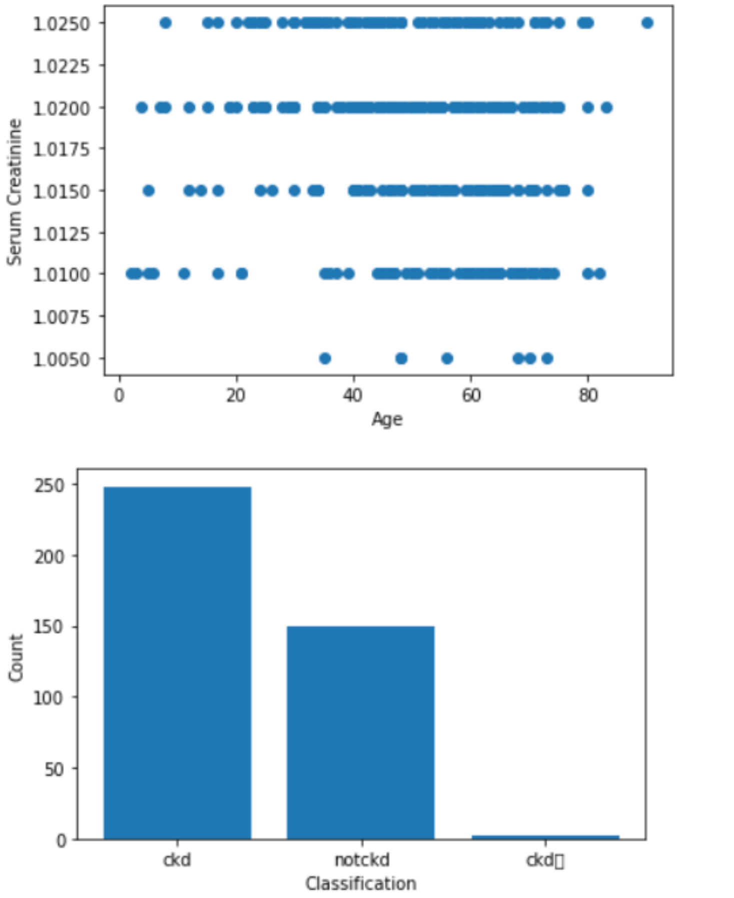


Figure 3. The Distribution of Classification

4.2 Data Pre-Processing

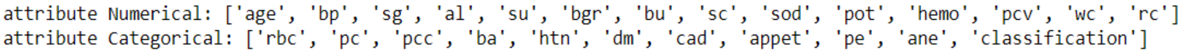
Data pre-processing is an essential step after gathering the dataset from a reliable source. Raw data may contain noise and have inconsistent types, requiring pre-processing before applying machine learning algorithms for classification and analysis.

Data exploration: The dataset is first read into a Pandas Data Frame using the algorithm. It consists of 24 variables, including age, blood pressure, serum creatinine level, and the classification of chronic kidney disease. A scatter plot of age versus serum creatinine level and a bar chart showing the distribution of classification are displayed. The scatter plot in Figure 3 indicates a positive association between age and serum creatinine, suggesting that as people age, their levels of these substances tend to increase.

Data cleaning: The next step involves identifying columns with missing values by analyzing the dataset. The columns "Pcv, " "wc, " and "rc" are found to have missing values. The algorithm and code impute the missing values in categorical fields using the most frequent values. In the case of the "Pcv" column, for example, the code replaces missing values with 43, as it is the most frequent value. The KNN imputation algorithm is then used to impute missing values in numerical columns. This method mimics missing values by averaging the values of the nearest neighbors.

Data encoding: Categorical columns are label-encoded in the next step. Label encoding assigns a different integer value to each category. In this dataset, the categorization column has two categories, "ckd" and "notckd, " which are represented as 0 and 1, respectively.

Data splitting: A binary value (0 or 1) is created from the categorization column using code, as machine learning models can only understand numerical data. The data is then divided into training and testing sets using code. The machine learning models are trained using the training data, and their effectiveness is evaluated using the testing data.



```
attribute Numerical: ['age', 'bp', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc']
attribute Categorical: ['rbc', 'pc', 'pcc', 'ba', 'htn', 'dm', 'cad', 'appet', 'pe', 'ane', 'classification']
```

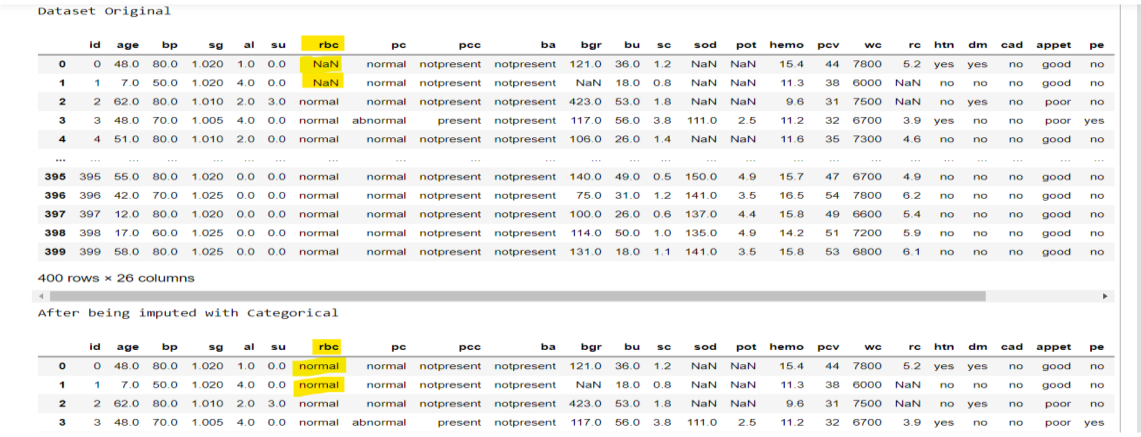
Figure 4. Attribute classification

4.3 Data Imputation

Before imputation, the raw dataset is analyzed to determine the number of missing values in each column. In this research, the "SimpleImputer" method is used to impute missing values in categorical columns. It fills in the missing information with the most frequent value. The imputed dataset is then ready for encoding the categorical columns using the 'LabelEncoder' function, which transforms categorical values into integers.

The KNNImputer method is applied to handle missing values in numerical columns. The code sets the k-value to five, which imputes missing values by finding the closest possible value based on the k-nearest neighbors.

Figure 5. Imputing the Categorical Variables.



Dataset Original

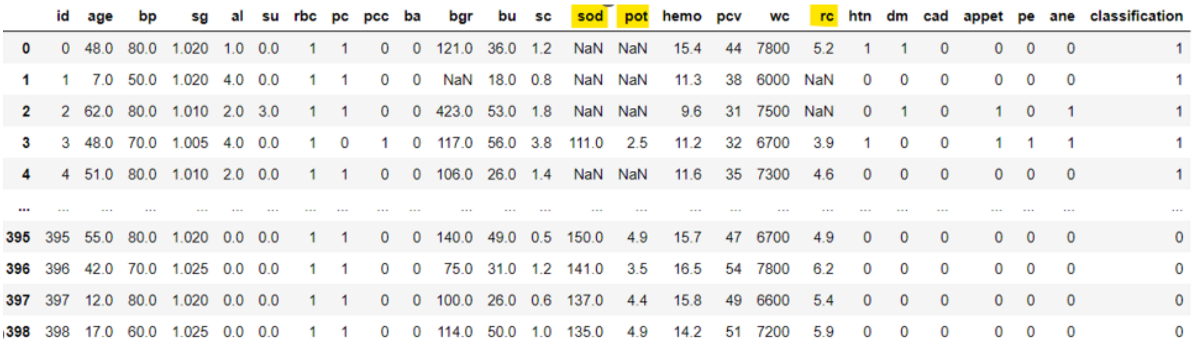
	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	121.0	36.0	1.2	NaN	NaN	15.4	44	7800	5.2	yes	yes	no	good	no
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	NaN	18.0	0.8	NaN	NaN	11.3	38	6000	NaN	no	no	no	good	no
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	53.0	1.8	NaN	NaN	9.6	31	7500	NaN	no	yes	no	poor	no
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	106.0	26.0	1.4	NaN	NaN	11.6	35	7300	4.6	no	no	no	good	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	140.0	49.0	0.5	150.0	4.9	15.7	47	6700	4.9	no	no	no	good	no
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	75.0	31.0	1.2	141.0	3.5	16.5	54	7800	6.2	no	no	no	good	no
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	100.0	26.0	0.6	137.0	4.4	15.8	49	6600	5.4	no	no	no	good	no
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	114.0	50.0	1.0	135.0	4.9	14.2	51	7200	5.9	no	no	no	good	no
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	131.0	18.0	1.1	141.0	3.5	15.8	53	6800	6.1	no	no	no	good	no

400 rows × 26 columns

After being imputed with Categorical

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe
0	0	48.0	80.0	1.020	1.0	0.0	normal	normal	notpresent	notpresent	121.0	36.0	1.2	NaN	NaN	15.4	44	7800	5.2	yes	yes	no	good	no
1	1	7.0	50.0	1.020	4.0	0.0	normal	normal	notpresent	notpresent	NaN	18.0	0.8	NaN	NaN	11.3	38	6000	NaN	no	no	no	good	no
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	423.0	53.0	1.8	NaN	NaN	9.6	31	7500	NaN	no	yes	no	poor	no
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes

Figure 6. Imputing the Numerical Variables.



	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	1	1	0	0	121.0	36.0	1.2	NaN	NaN	15.4	44	7800	5.2	1	1	0	0	0	0	1
1	1	7.0	50.0	1.020	4.0	0.0	1	1	0	0	NaN	18.0	0.8	NaN	NaN	11.3	38	6000	NaN	0	0	0	0	0	0	1
2	2	62.0	80.0	1.010	2.0	3.0	1	1	0	0	423.0	53.0	1.8	NaN	NaN	9.6	31	7500	NaN	0	1	0	1	0	1	1
3	3	48.0	70.0	1.005	4.0	0.0	1	0	1	0	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	1	0	0	1	1	1	1
4	4	51.0	80.0	1.010	2.0	0.0	1	1	0	0	106.0	26.0	1.4	NaN	NaN	11.6	35	7300	4.6	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	395	55.0	80.0	1.020	0.0	0.0	1	1	0	0	140.0	49.0	0.5	150.0	4.9	15.7	47	6700	4.9	0	0	0	0	0	0	0
396	396	42.0	70.0	1.025	0.0	0.0	1	1	0	0	75.0	31.0	1.2	141.0	3.5	16.5	54	7800	6.2	0	0	0	0	0	0	0
397	397	12.0	80.0	1.020	0.0	0.0	1	1	0	0	100.0	26.0	0.6	137.0	4.4	15.8	49	6600	5.4	0	0	0	0	0	0	0
398	398	17.0	60.0	1.025	0.0	0.0	1	1	0	0	114.0	50.0	1.0	135.0	4.9	14.2	51	7200	5.9	0	0	0	0	0	0	0

4.4 Data Augmentation

In data augmentation, the goal is to generate new synthetic data for the minority class in order to balance an unbalanced dataset. The Synthetic Minority Over-sampling Technique (SMOTE) is employed to produce fresh samples. A total of 50 new samples are added using SMOTE to achieve the desired number of samples per class. In this case, the target would be 150 samples per class, considering that there were initially 100 samples and 50 more were added to reach a total of 4000 rows. The Sampling_Strategy is used to keep track of the desired quantity for each class, and the SMOTE object is configured to generate synthetic data.

4.5 Model Training

In the model definition, two key variables, X and Y, are defined. X represents all attributes in the dataset except the classification column, while Y represents the classification column as the target variable. The dataset is divided into training and test sets using the train_test_split() function, with a test size of 20%. The training set, test set, y_train variable (the goal variable for training), and y_test variable (the target variable for testing) areCheck plagiarism for "Data augmentation techniques are used to generate new synthetic data for the minority class in order to balance an imbalanced dataset. One popular technique is the Synthetic Minority Over-sampling Technique (SMOTE), which creates new samples by interpolating between existing samples of the minority class. Another technique is the Random Over-sampling Examples (ROSE), which randomly duplicates samples of the minority class to increase their representation. Both techniques aim to increase the number of samples in the minority class, making the dataset more balanced and improving the performance of machine learning models.

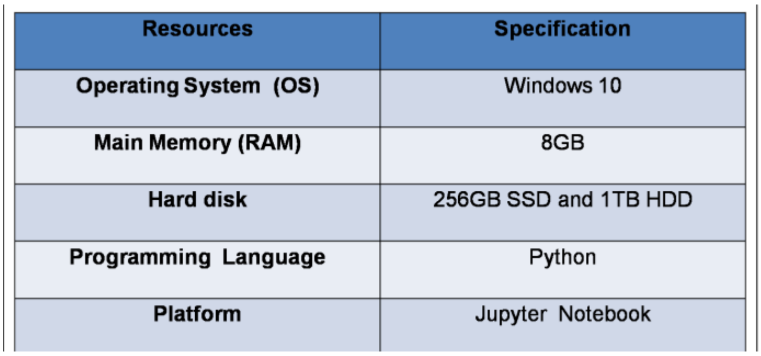
4.6 Model Evaluation

After successfully training the models on the training data, each model was evaluated using the test data. Classification-based metrics were used to assess the model because of the nature of this study. Accuracy, precision, and recall are the measurements considered to identify the best algorithm for predicting the chronic kidney disease.

5. Implementation of CKD Model

Multiple machine learning algorithms, like "KNN Original," "KNN CV," "BNB Original," "BNB CV," "DT Original," "DT CV," "KNN Scaling," "KNN Scaling CV," "KNN Feature Selection," "KNN PCA," "KNN PCA CV," and "BNB PCA," are used in this research. The libraries used in the implementation were LabelEncoder, MinMaxScaler, KNNImputer, SimpleImputer, NumPy, Seaborn, and pandas. Python was used for scripting. To implement the model, the following criteria must be met.

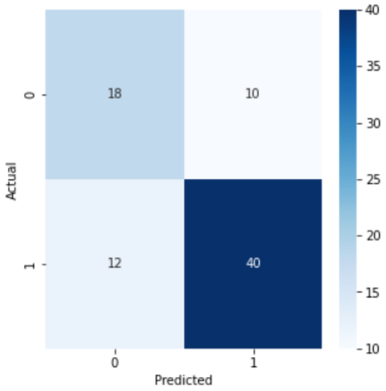
Figure 8: Configurations



Resources	Specification
Operating System (OS)	Windows 10
Main Memory (RAM)	8GB
Hard disk	256GB SSD and 1TB HDD
Programming Language	Python
Platform	Jupyter Notebook

6. Results and Evaluation

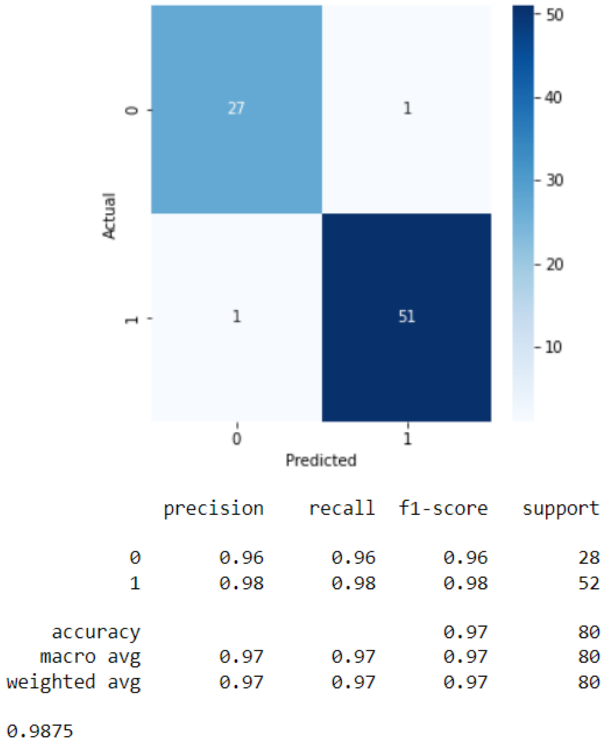
6.1 KNN Original & KNN CV: For KNN, Python's scikit-learn module was used for the creation of the KNN method, with n_neighbors set to 5. The model was trained on the training dataset and measures such as precision, recall, F1-score, and accuracy, which were then used to evaluate the performance of the model on the test dataset. The findings demonstrate that the KNN classifier attained an accuracy of 73%, with a precision and recall of 60% and 80% for classes 0 and 1, respectively. The F1 scores for classes 0 and 1 were 0.62 and 0.78, respectively, with macro-and weighted average F1 values of 0.70 and 0.73. Better performance for class 1 predictions was suggested by the confusion matrix visualization. The cross-validation showed an average accuracy of 60.625 %. The results demonstrate how well the KNN classifier performs in binary classification, which justifies further investigation through feature selection and hyperparameter adjustment.



```
Cross Validation Score Accuracy: [0.546875 0.65625  0.6875   0.609375 0.53125 ]
Cross Validation Score Accuracy Mean: 0.60625
```

Figure 9. Output for KNN Original & KNN CV

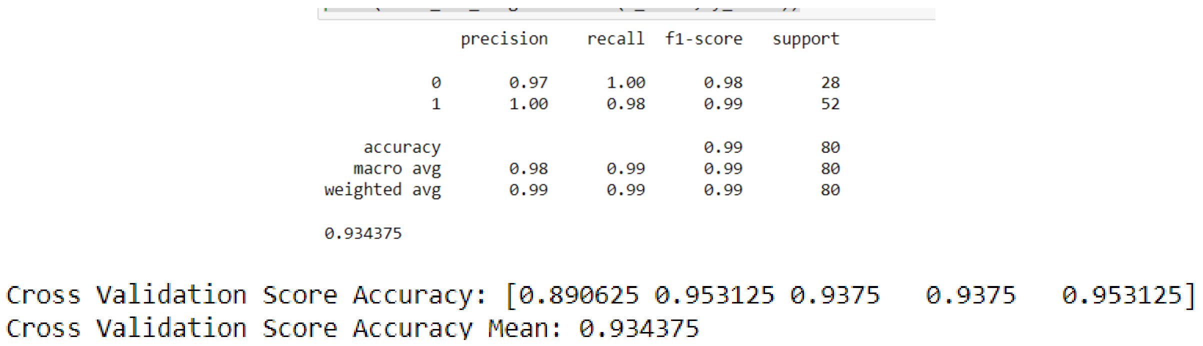
6.2 KNN Original & KNN CV (After Feature Selection):_After performing feature selection using min–max scaling, the KNN classifier is trained on the scaled data. The classifier's accuracy, precision, recall, and F1-score were assessed on the test dataset, and the classification outcomes were analyzed using a confusion matrix. From the results, it can be concluded that feature scaling considerably increased the accuracy and performance of the KNN classifier, resulting in high precision and recall values for both classes and an overall accuracy of 0.97. With an average accuracy of 0.971875, cross-validation analysis further supported the predictability of the model. The findings underline the importance of data preparation methods for improving the efficiency of the KNN classifier in binary classification tasks.



```
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        28
           1       0.98      0.98      0.98        52

    accuracy                           0.97        80
   macro avg       0.97      0.97      0.97        80
weighted avg       0.97      0.97      0.97        80

0.9875
```

Figure 10. Output for KNN Original & KNN CV after Min-Max feature Selection

6.3 BNB Original & BNB CV:_After the dataset is pre-processed, feature selection methods such as mutual information and chi-square are applied. Additionally, dimensionality reduction was accomplished using Principal Component Analysis (PCA). The classifiers were assessed on the test dataset using methods such as precision, recall, and F1-score. With the approaches used, the performance increased significantly according to the cross-validation scores and confusion matrices. The results show that for both classes, the Gaussian Naive Bayes classifier had a remarkable accuracy of 99%, with great precision and recall. Cross-validation revealed a steady performance with an accuracy mean of 93.44%. These findings imply that the Gaussian Naive Bayes classifier exhibits encouraging promise for binary classification tasks and may be a good option for real-world applications requiring high accuracy and efficiency.

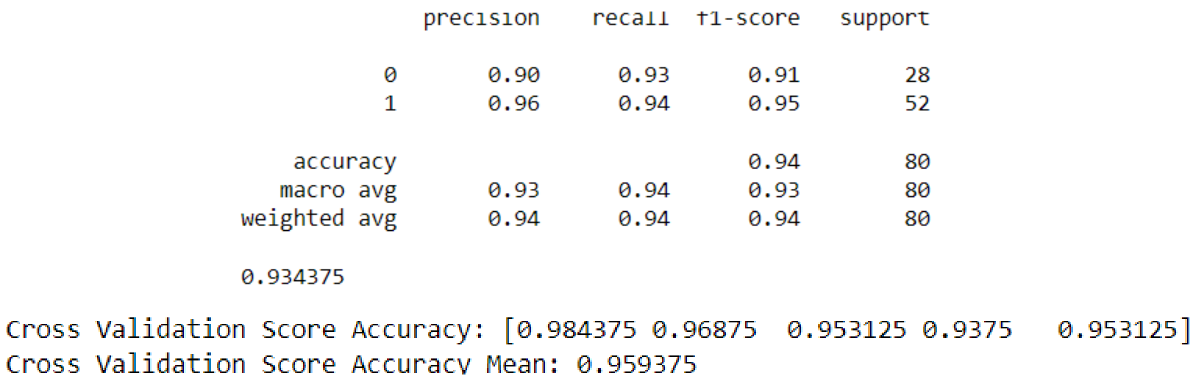


```
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        28
           1       1.00      0.98      0.99        52

    accuracy                           0.99        80
   macro avg       0.98      0.99      0.99        80
weighted avg       0.99      0.99      0.99        80

0.934375
```

```
Cross Validation Score Accuracy: [0.890625 0.953125 0.9375   0.9375   0.953125]
Cross Validation Score Accuracy Mean: 0.934375
```

Figure 11. Output for BNB Original & BNB CV

6.4 DT Original & DT CV:_With a maximum depth of five, to prevent overfitting, the Decision Tree model is built. The dataset was divided into training and test sets, and a variety of metrics, including precision, recall, and F1-score, were used to evaluate the classifier's performance. The results showed that the Decision Tree classifier had an accuracy of 94%, and that its precision and recall for classes 0 and 1 were 90% and 96 %, respectively. Balanced performance of the classifier was demonstrated by macro-average and weighted-average F1-scores of 0.93. The consistency of the model was further supported by a cross-validation analysis, which had an average accuracy of 95.94%.



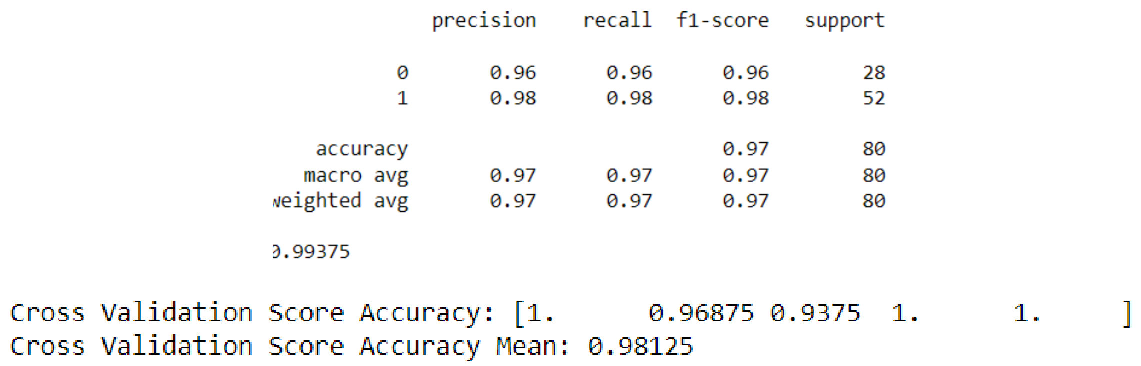
```
              precision    recall  f1-score   support

           0       0.90      0.93      0.91        28
           1       0.96      0.94      0.95        52

    accuracy                           0.94        80
   macro avg       0.93      0.94      0.93        80
weighted avg       0.94      0.94      0.94        80

0.934375
```

```
Cross Validation Score Accuracy: [0.984375 0.96875  0.953125 0.9375   0.953125]
Cross Validation Score Accuracy Mean: 0.959375
```

Figure 12. Output for DT Original & DT CV

6.5 Feature Selection Techniques for Binary Classification: Chi-square and Mutual Information Analysis: The dataset consists of both category and numerical features. First, using the Chi-square test, the association between category variables and classification was investigated. The relationship between the numerical qualities and class labels is measured using mutual information in a similar manner. The results show that 'rbc,' measured by the p-value, is the least significant categorical feature, while 'su, ' 'wc,' and 'age' are rated as having lower mutual information scores. These less useful traits/unwanted attributes were removed from the dataset.

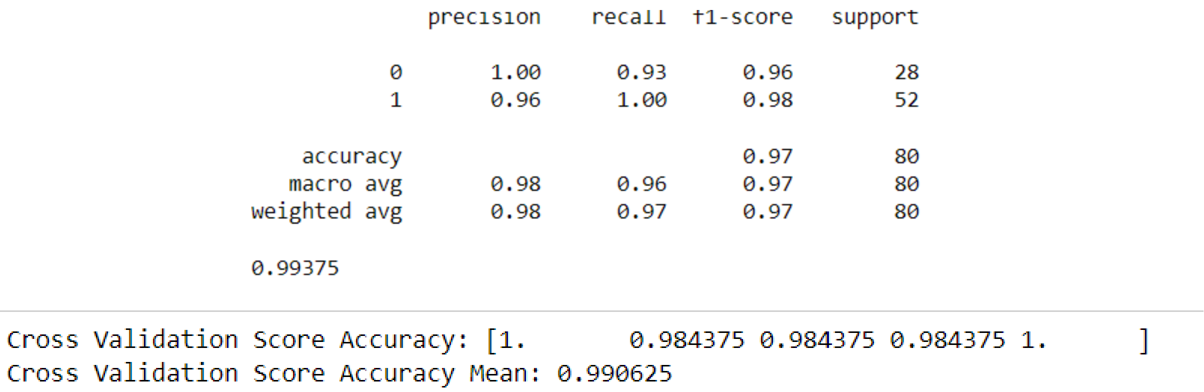
6.6 'KNN Scaling', 'KNN Scaling CV' & 'KNN Feature Selection': Analysing the impact of feature selection on K-Nearest Neighbors (KNN) classifier performance for binary classification problems. using chi-square and mutual information analysis, the features that are less informative in the dataset is subjected to feature selection. The KNN classifier was then trained on a new version of the features. The classification outcomes show that feature selection increases the accuracy of the KNN classifier by 97%, while also improving its recall, precision, and F1-score. The ability of the model to provide precise predictions for both classes is supported by the confusion matrix. With an average accuracy of 98.125 %, the cross-validation analysis revealed consistent performance. These results show that feature selection significantly affects the performance of the KNN classifier, improving its accuracy and efficiency for binary classification tasks.



```
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        28
           1       0.98      0.98      0.98        52

    accuracy                           0.97        80
   macro avg       0.97      0.97      0.97        80
weighted avg       0.97      0.97      0.97        80

0.99375

Cross Validation Score Accuracy: [1.       0.96875 0.9375  1.       1.      ]
Cross Validation Score Accuracy Mean: 0.98125
```

Figure 13. Output for KNN Original & KNN CV after Chi-Square

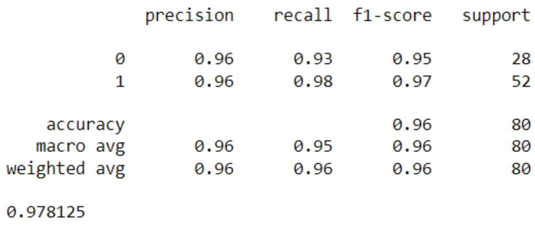
6.7 'KNN PCA' & 'KNN PCA CV': Principal Component Analysis (PCA) is used for dimensionality reduction to enhance the K-Nearest Neighbors (KNN) classifier's performance in binary classification tasks. The original features of the dataset were converted into three principal components through PCA. The KNN classifier was then trained using the transformed data. The classification outcomes showed that PCA considerably increased the accuracy, precision, recall, and F1-score of the KNN classifier, resulting in an accuracy of 97%. The model's capability to produce precise predictions for both groups was validated using a confusion matrix. Additionally, cross-validation analysis, which achieved an average accuracy of 99.06%, showed a constant performance of the KNN classifier with PCA. These results show that PCA, as a dimensionality reduction technique, successfully improves the performance and accuracy of the KNN classifier.



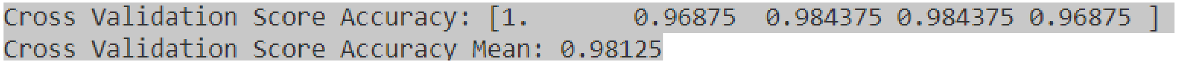
```
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        28
           1       0.96      1.00      0.98        52

    accuracy                           0.97        80
   macro avg       0.98      0.96      0.97        80
weighted avg       0.98      0.97      0.97        80

0.99375

Cross Validation Score Accuracy: [1.       0.984375 0.984375 0.984375 1.      ]
Cross Validation Score Accuracy Mean: 0.990625
```

Figure 14. Output for KNN PCA & KNN PCA CV

6.8 'BNB PCA' & 'BNB PCA CV': Principal Component Analysis (PCA) is used as a method for dimensionality reduction with the Gaussian Naive Bayes classifier to improve the effectiveness of binary classification. According to the classification findings, PCA improved the classifier's accuracy, precision, recall, and F1-score, resulting in a 96% accuracy rate. Cross-validation analysis also revealed a constant performance with 98.12% average accuracy. According to these findings, by integrating Gaussian Naive Bayes with PCA, the same produces accurate and efficient results.



```
              precision    recall  f1-score   support

           0       0.96      0.93      0.95        28
           1       0.96      0.98      0.97        52

    accuracy                           0.96        80
   macro avg       0.96      0.95      0.96        80
weighted avg       0.96      0.96      0.96        80

0.978125
```



```
Cross Validation Score Accuracy: [1.       0.96875  0.984375 0.984375 0.96875 ]
Cross Validation Score Accuracy Mean: 0.98125
```

Figure 15. Output for BNB PCA & BNB PCA CV

6.9 Automating the file: In this process, the entire file is automated. So, whenever a binary classification dataset is passed, with minor tweaking the entire process would take place. That is,

"Dataset is cleaned and transformed.
Dataset is pre-processed.
Dataset is Imputed and Augmented
Application of multiple machine learning algorithms to identify the best performing algorithm in terms of Accuracy, Precision, Recall and F1-Score."

7. Discussion & Comparison of Developed Models

Comprehensive comparative analysis of various classification models for binary classification tasks: This study aims to evaluate and compare the performance of different classification models, including K-Nearest Neighbors (KNN), Gaussian Naive Bayes (BNB), and Decision Tree (DT) classifiers. The models are evaluated in their original form as well as after applying feature scaling, feature selection, and Principal Component Analysis (PCA) for dimensionality reduction. The evaluation metrics considered include accuracy, precision, recall, and F1-score.

The results reveal that the KNN model with PCA (KNN PCA CV) achieves the highest accuracy of 99. 06% and outperforms the other models. The BNB classifier in its original form (BNB Original) and with PCA (BNB PCA CV) also demonstrate strong performance, with accuracies of 99. 00% and 98. 12%, respectively.

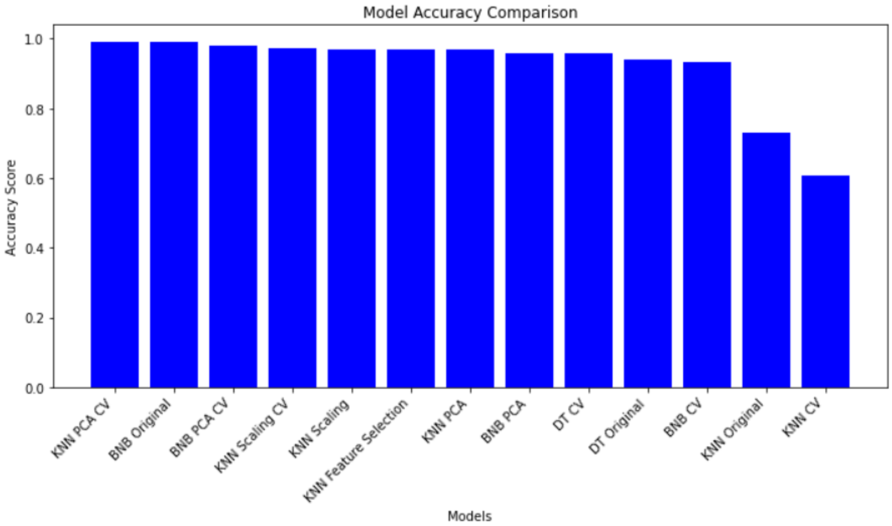


Figure 16. Model Comparison

8. Conclusion & Recommended Future

As a result, this study evaluates the effectiveness of various classification models for binary classification tasks using a variety of preprocessing methods. The findings demonstrate the value of feature selection and dimensionality reduction techniques in improving model efficacy and accuracy. The best-performing model was the K-Nearest Neighbors (KNN) classifier using Principal Component Analysis (PCA), which attained a remarkable accuracy of 99.06%. Gaussian Naive Bayes (BNB) models also performed well. The study highlights the significance of making the right preprocessing decisions to enhance the performance of classification models. To further improve the effectiveness and usability of binary classification models, researchers are encouraged to investigate additional preprocessing approaches, ensemble methods, hyperparameter tweaking, robustness analysis, and real-world applications.

9. Acknowledgements

I would like to express my gratitude to my lecturer Dr. Syed Muslim Jameel for his supervision, guidance, and support throughout the research.

10. References

Charumathi Sabanayagam et al., “Prediction of diabetic kidney disease risk using machine learning models: a population-based cohort study of Asian adults,” medRxiv (Cold Spring Harbor Laboratory), Aug. 2022, doi: https://doi.org/10.1101/2022.08.17.22278900.

K. C. Safavi et al., “Development and Validation of a Machine Learning Model to Aid Discharge Processes for Inpatient Surgical Care,” JAMA Network Open, vol. 2, no. 12, p. e1917221, Dec. 2019, doi: https://doi.org/10.1001/jamanetworkopen.2019.17221.

Md. Rashed-Al-Mahfuz, A. Haque, A. Azad, S. A. Alyami, J. M. W. Quinn, and M. A. Moni, “Clinically Applicable Machine Learning Approaches to Identify Attributes of Chronic Kidney Disease (CKD) for Use in Low-Cost Diagnostic Screening,” IEEE Journal of Translational Engineering in Health and Medicine, vol. 9, pp. 1–11, 2021, doi: https://doi.org/10.1109/jtehm.2021.3073629.

Md. A. Islam, S. Akter, Md. S. Hossen, S. A. Keya, S. A. Tisha, and S. Hossain, “Risk Factor Prediction of Chronic Kidney Disease based on Machine Learning Algorithms,” 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), Dec. 2020, https://doi.org/10.1109/iciss49785.2020.9315878.

J. A. Roth et al., “Cohort-Derived Machine Learning Models for Individual Prediction of Chronic Kidney Disease in People Living With Human Immunodeficiency Virus: A Prospective Multicenter Cohort Study,” The Journal of Infectious Diseases, vol. 224, no. 7, pp. 1198–1208, May 2020, doi: https://doi.org/10.1093/infdis/jiaa236.

Sanmarchi, F., Fanconi, C., Golinelli, D., Gori, D., Hernandez-Boussard, T. and Capodici, A. (2023). Predict, diagnose, and treat chronic kidney disease with machine learning: a systematic literature review. Journal of Nephrology. doi:https://doi.org/10.1007/s40620-023-01573-4

Su, C.-T., Chang, Y.-P., Ku, Y.-T. and Lin, C.-M. (2022). Machine Learning Models for the Prediction of Renal Failure in Chronic Kidney Disease: A Retrospective Cohort Study. Diagnostics, [online] 12(10), p.2454.

Phung, S., Kumar, A. and Kim, J. (2019). A deep learning technique for imputing missing healthcare data. [online] IEEE Xplore. doi:https://doi.org/10.1109/EMBC.2019.8856760.

Chowdhury, M.H., Islam, M.K. and Khan, S.I. (2017). Imputation of missing healthcare data. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICCITECHN.2017.8281805

Y. Yu et al., “Machine Learning Methods for Predicting Long-Term Mortality in Patients After Cardiac Surgery,” Frontiers in Cardiovascular Medicine, vol. 9, May 2022, doi: https://doi.org/10.3389/fcvm.2022.831390.

BEAULIEU-JONES, B.K. and MOORE, J.H. (2016). MISSING DATA IMPUTATION IN THE ELECTRONIC HEALTH RECORD USING DEEPLY LEARNED AUTOENCODERS. Biocomputing 2017.https://doi.org/10.1142/9789813207813_0021.

S. Mittal and Yasha Hasija, “Applications of Deep Learning in Healthcare and Biomedicine,” Studies in big data, Jan. 2020, doi: https://doi.org/10.1007/978-3-030-33966-1_4.

Y. Zhang, B. Zhou, X. Cai, W. Guo, X. Ding, and X. Yuan, “Missing value imputation in multivariate time series with end-to-end generative adversarial networks,” Information Sciences, vol. 551, pp. 67–82, Apr. 2021, doi: https://doi.org/10.1016/j.ins.2020.11.035.

H. Jiang, C. Wan, K. Yang, Y. Ding, and S. Xue, “Continuous missing data imputation with incomplete dataset by generative adversarial networks–based unsupervised learning for long-term bridge health monitoring,” Structural Health Monitoring, p. 147592172110219, Jun. 2021, doi: https://doi.org/10.1177/14759217211021942.

J. Hou et al., “Deep learning and data augmentation based data imputation for structural health monitoring system in multi-sensor damaged state,” Measurement, vol. 196, p. 111206, Jun. 2022, doi: https://doi.org/10.1016/j.measurement.2022.111206.

R. Krishnan, P. Rajpurkar, and E. J. Topol, “Self-supervised learning in medicine and healthcare,” Nature Biomedical Engineering, Aug. 2022, doi: https://doi.org/10.1038/s41551-022-00914-1.

Mustafa, A. and Rahimi Azghadi, M. (2021). Automated Machine Learning for Healthcare and Clinical Notes Analysis. Computers, 10(2), p.24. doi:https://doi.org/10.3390/computers10020024

Subah, F.Z., Deb, K., Dhar, P.K. and Koshiba, T. (2021). A Deep Learning Approach to Predict Autism Spectrum Disorder Using Multisite Resting-State fMRI. Applied Sciences, 11(8), p.3636. doi:https://doi.org/10.3390/app11083636.

S. Singh, S. Sharma, and S. Bhadula, “Automated Deep Learning based Disease Prediction Using Skin Health Records: Issues, Challenges and Future Directions,” IEEE Xplore, Mar. 01, 2022. https://ieeexplore.ieee.org/abstract/document/9752422/ (accessed Jul. 08, 2023).

J. Aswini, B. Yamini, Rajaram Jatothu, K. Sankara Nayaki, and M. Nalini, “An efficient cloud-based healthcare services paradigm for chronic kidney disease prediction application using boosted support vector machine,” Concurrency and Computation: Practice and Experience, vol. 34, no. 10, Dec. 2021, doi: https://doi.org/10.1002/cpe.6722.