Chronic Kidney Disease Prediction: A Machine Learning Approach

Faizan Ansari

Email: faizanansari3dec@gmail.com

P.Sai Leela.

Email:leelasai023@gmail.com

***Abstract*—This electronic document is a "live" template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. **CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.* (*Abstract*)**

I. Abstract

Chronic Kidney Disease (CKD) is another name for the occurrence of Chronic Renal Disease (CRD). It shows a disease that affects a person's general health and damages the kidneys. Poor illness detection and treatment can lead to end-stage renal disease and the patient's eventual death. Machine Learning (ML) approaches are growing as a useful tool in the medical science sector and are crucial for disease prediction. The proposed project aims to build and validate a predictive model for the outcome of chronic renal illness.

This paper extends prior research on chronic kidney disease (CKD) prediction by implementing a scalable and feature-rich machine learning pipeline using a real-world dataset of 20,000 samples. Building upon a foundational study that employed only 400 samples, we introduce additional models, rigorous cross-validation, interaction terms, and improved visual analytics. Our findings suggest that larger data and robust engineering significantly improve prediction performance and generalization capability.

Keywords—chronic renal disease, classification algorithms, random forest classifier, machine learning

II. Introduction (*Heading 1*)

Chronic Kidney Disease (CKD) is a serious condition that affects your kidneys and overall health. When your kidneys don't work well, harmful waste builds up in your body. This can lead to other health problems like heart disease. Regular lab tests can help catch CKD early, which makes it easier to treat and manage.

CKD can be caused by things like smoking, poor diet, lack of sleep, and other health issues. In 2016, over 700 million people worldwide had CKD — more women than men. If it gets worse, CKD can lead to kidney failure.

Doctors check for CKD using tests like urine analysis and blood tests to measure creatinine. They also consider your blood pressure, health history, and family background. A key measure is the **estimated Glomerular Filtration Rate (eGFR)**, which shows how well your kidneys are filtering waste.

There are **five stages of CKD**, from mild to severe:

1. **Stage 1:** Normal or high function (GFR > 90)
2. **Stage 2:** Mild damage (GFR 60–89)
3. **Stage 3:** Moderate damage (GFR 30–59)
4. **Stage 4:** Severe damage (GFR 15–29)
5. **Stage 5:** Kidney failure (GFR < 15)

The kidneys filter 120–150 quarts of blood each day to make 1–2 quarts of urine. They also help balance chemicals in the body and produce hormones that control blood pressure and red blood cell production.

Machine learning is now helping doctors predict and diagnose CKD more accurately by analyzing important patterns in patient data. These smart tools can find the most useful information while ignoring the less important parts, making early detection faster and more reliable.

With its increasing frequency, chronic kidney disease (CKD) represents a significant worldwide health burden. Timely action depends on early detection. The UCI CKD dataset (400 samples) was used in the previous study by Kaur et al. to assess a limited number of machine learning models (Decision Tree, Logistic Regression, SVM). Although useful for proving machine learning's viability in CKD detection, generality. was limited by the small dataset and feature processing.

We increase the dataset to 20,000 patient records in this study and use a thorough SEMMA-based pipeline that includes feature engineering, significant preprocessing, contemporary ensemble learning techniques, and thorough evaluation. Our objective is to incorporate advanced techniques appropriate for production-level clinical decision support in order to validate and expand on the findings from the original research.

Several prediction models, including Random Forest (RF), Decision Tree, Logistic Regression (LR), K Nearest Neighbour (KNN), and Support Vector Machine (SVM), Gradient Boosting, XGBoost, Naive Bayes were compared in this study.

III. DATASET AND METHODS

Learners may process information without explicit programming due to a sort of artificial intelligence called machine learning. Its main goal is to create computer programmers who can adapt to new information. It falls into one of two categories: supervised or unsupervised [15]. It all comes down to putting the right traits together to build frameworks that accomplish the right goals. Predictive clustering, parametric modeling, and multi-dimensional and multiclassification are a few examples of these tasks.

Three main steps are involved in the proposed methodology: preprocessing of the data, training of the models, and model selection.

A. Dataset

TABLE I. FEATURES LISTED IN THE CKD DATASET

age	float64
bp	float64
sg	float64
al	int64
su	int64
rbc	int8
pc	int8
pcc	int8
ba	int8
bgr	float64
bu	float64
sc	float64
sod	float64
pot	float64
hemo	float64
pcv	float64
wc	float64
rc	float64
htn	int8
dm	int8
cad	int8
appet	int8
pe	int8
ane	int8
classification	int8

dtype: object

B. Methodology

This study follows the SEMMA (Sample, Explore, Modify, Model, Assess) approach to build a machine learning-based prediction model for Chronic Kidney Disease (CKD).

1. **Sample**:
   The dataset used for this study was loaded from a CSV file containing patient records related to CKD. Python libraries such as Pandas and NumPy were used to handle and manipulate the data.
2. **Explore**:
   An initial exploratory data analysis was conducted to understand the structure and nature of the dataset. Categorical features were encoded into numeric form, and data types were standardized. Missing values were identified and handled using simple imputation techniques.
3. **Modify**:
   Feature engineering steps included scaling numerical features using StandardScaler and selecting the most relevant features using SelectKBest with mutual information as the scoring function. Additional preprocessing steps included label encoding and polynomial feature transformation where needed.
4. **Model**:
   Multiple machine learning algorithms were applied to train prediction models, including:
   - Logistic Regression
   - Support Vector Machines (SVM)
   - Random Forest
   - Gradient Boosting
   - XGBoost
   - K-Nearest Neighbors (KNN)
   - Naive Bayes
   - Decision Trees

   A training-test split (commonly 80:20) was used to evaluate performance, and hyperparameter tuning was done using GridSearchCV.
5. **Assess**:
   Model performance was evaluated using various metrics including accuracy, precision, recall, F1-score, ROC-AUC score, and confusion matrix. Visualization tools such as ROC curves and confusion matrix plots were used to better understand model behavior and performance.

C.Preprocessing

Data preprocessing was an essential step in preparing the Chronic Kidney Disease dataset for machine learning. The key preprocessing steps included:

1. **Data Cleaning**:

Column names were standardized by converting them to lowercase and replacing spaces with underscores for consistency.

Categorical values were encoded into numerical format using label encoding, making them suitable for machine learning models.

2. **Handling Missing Values**:

The dataset contained missing values, which were imputed using the SimpleImputer method from Scikit-learn.

Numerical features were typically filled using the mean strategy, while categorical features were imputed with the most frequent value.

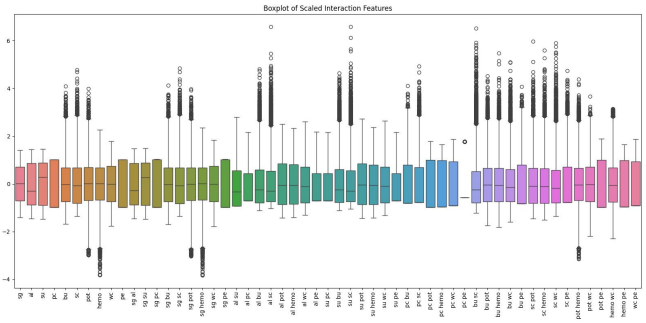
3. **Feature Scaling**:

To ensure uniformity among features, numerical attributes were scaled using StandardScaler. This step helps many models (like SVM and KNN) perform better by normalizing feature ranges.

4. **Interaction Terms**

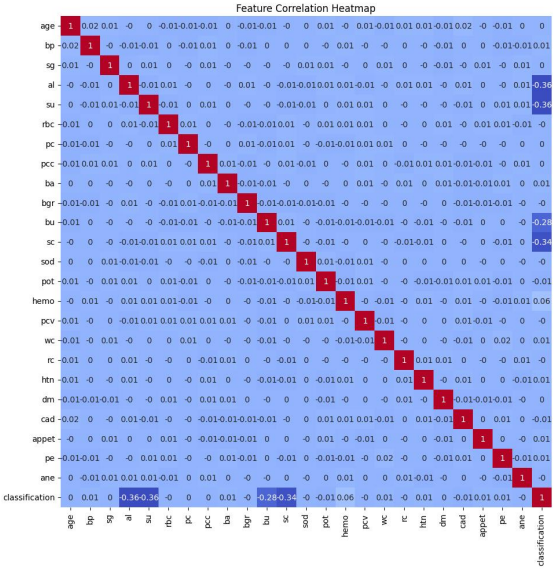
Interaction terms represent the combined effect of two or more features on the target variable. In some cases, the relationship between a feature and the target may depend on the value of another feature.

In this project, we explored whether adding interaction terms (e.g., blood_pressure × age, serum_creatinine × hemoglobin) could improve model performance. These terms help capture more complex relationships that individual features alone might miss.



5. **Correlation Heatmap:**

To examine the linear relationships between numerical variables and detect multicollinearity or strong associations. A heatmap was generated to visualize the correlation between numerical variables such as blood pressure, specific gravity, albumin, and others. The intensity of color indicates the strength of correlation. Strong positive or negative correlations help in understanding variable interactions, which can guide feature selection.

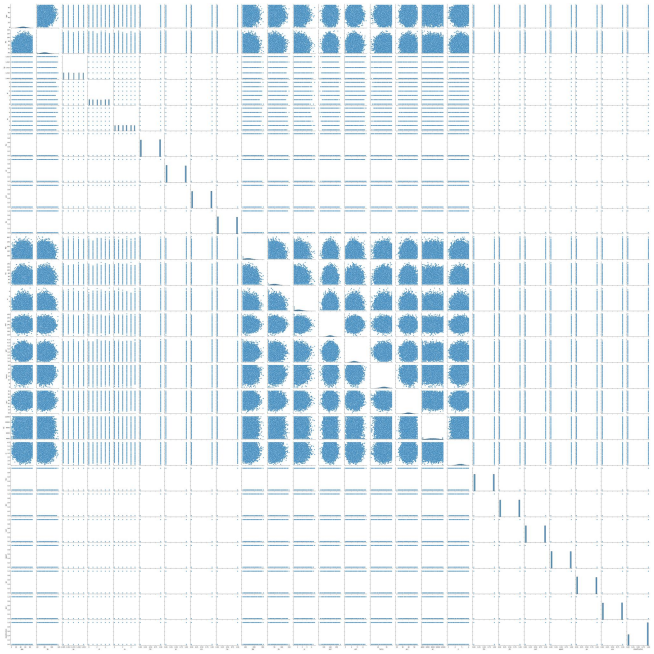


6. **Feature Selection**:

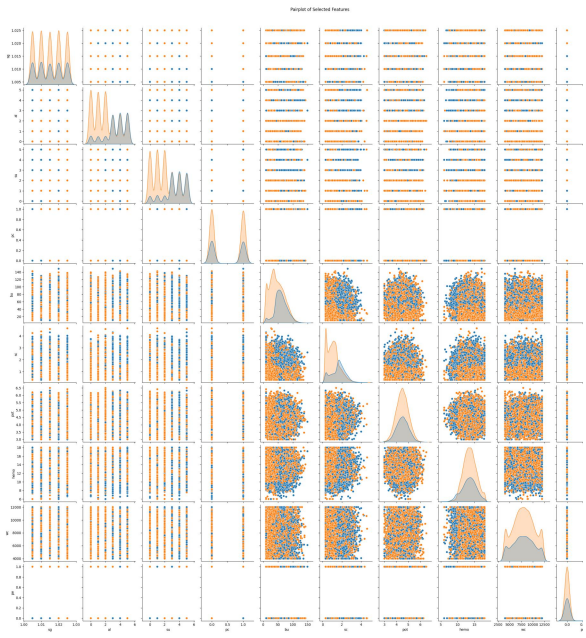
Relevant features were selected using the SelectK Best method with mutual information as the scoring function. This helped reduce dimensionality and improve model performance by focusing on the most informative attributes.

7. Exploring Data Visualization

Pairplot Analysis:



The goal of this analysis is to examine the interactions between key numerical features such as age, blood pressure, serum creatinine, and hemoglobin. By visualizing these relationships, we aim to observe how well these variables help differentiate between patients with and without Chronic Kidney Disease (CKD), detect potential linear or non-linear patterns, and identify any outliers. Additionally, this analysis helps in determining which features may be most useful for classification tasks. For clarity and interpretability, we selected numerical features that are clinically significant and commonly associated with kidney function. These include: **age**, representing the patient's age; **blood_pressure**, indicating cardiovascular stress that can affect kidney health; **serum_creatinine**, a crucial marker of kidney filtration efficiency; and **hemoglobin**, which often drops in CKD due to impaired erythropoietin production and anemia.



8.Train-Test Split:

- The dataset was divided into training and testing sets using an 80:20 ratio to evaluate the model's generalization capability.

Hyperparameter Tuning

To enhance model performance, **hyperparameter tuning** was conducted using **GridSearchCV** along with **5-fold cross-validation**. This method divides the training data into five parts, trains the model on four parts, and validates it on the fifth, rotating through all combinations. It ensures a more reliable evaluation and helps prevent overfitting. GridSearchCV was used to find the best parameters for models like Random Forest, SVM, and XGBoost based on accuracy and F1-score.

D.Classification Algorithms

To predict the presence of Chronic Kidney Disease, several supervised machine learning classification algorithms were implemented and compared. The models were selected for their diverse decision-making strategies and proven effectiveness in medical data analysis. The algorithms used include:

1. **LogisticRegression**
   A baseline linear model used to estimate the probability of disease presence based on input features. It is simple, interpretable, and effective for binary classification problems.
2. **SupportVectorMachine(SVM)**
   SVM constructs a hyperplane in high-dimensional space to separate the classes. It is particularly effective in handling non-linear relationships using kernel functions.
3. **RandomForest**
   An ensemble method that builds multiple decision trees and combines their outputs. It reduces overfitting and improves accuracy by leveraging the power of multiple models.
4. **GradientBoostingClassifier**
   Another ensemble approach that builds models sequentially, where each new model focuses on correcting errors made by previous ones. It often yields high accuracy on structured data.
5. **XGBoost**
   An advanced implementation of gradient boosting optimized for speed and performance. It includes regularization and parallel processing, making it well-suited for tabular datasets with missing values.
6. **K-NearestNeighbors(KNN)**
   A distance-based algorithm that classifies instances based on the majority class among their nearest neighbors. It is simple but sensitive to the choice of 'k' and feature scaling.
7. **NaïveBayes**
   A probabilistic classifier based on Bayes' theorem with the assumption of feature independence. Despite its simplicity, it performs well in many medical datasets.
8. **DecisionTree**
   A tree-based model that splits the data into branches based on feature values. It is interpretable and captures non-linear relationships well.

Each model was trained on the preprocessed dataset and evaluated using key performance metrics to identify the most suitable classifier for predicting Chronic Kidney Disease.

1) Performance Evaluation Measures

To assess the effectiveness of the classification models in predicting Chronic Kidney Disease, several standard evaluation metrics were used. These metrics provide a comprehensive view of each model's predictive power, especially for binary classification tasks. The following measures were employed:

1. **Accuracy**
   Accuracy is the proportion of correctly predicted instances out of the total instances. It provides a general idea of how often the model is correct.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision**
   Precision measures the proportion of true positive predictions among all instances predicted as positive. It is useful when the cost of false positives is high.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. **Recall(Sensitivity)**
   Recall evaluates the model's ability to correctly identify all actual positive cases. It is important when missing positive cases has serious consequences.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. **F1-Score**
   The F1-score is the harmonic mean of precision and recall. It balances the trade-off between the two and is useful when the dataset is imbalanced.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

   ROC-AUCScore
   The Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) score measures the model's ability to distinguish between classes. A higher AUC indicates better performance across all classification thresholds.
5. **ConfusionMatrix**
   The confusion matrix provides a detailed breakdown of correct and incorrect predictions, including true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It helps in visualizing the performance of a classification model.

These metrics were calculated for each classifier to compare their effectiveness and select the most reliable model for predicting Chronic Kidney Disease.

E.Results and Discussion

In this study, we evaluated six different classification algorithms to predict Chronic Kidney Disease (CKD): **Logistic Regression**, **K-Nearest Neighbors (KNN)**, **Decision Tree**, **Random Forest**, **Support Vector Machine (SVM)**, and **Naive Bayes**. Each model was assessed using standard evaluation metrics—**accuracy**, **precision**, **recall**, and **F1-score**—alongside a **confusion matrix** to provide deeper insight into their performance.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.98	1.00	0.96	0.98
K-Nearest Neighbors	0.96	0.94	0.96	0.95
Decision Tree	0.98	0.96	1.00	0.98
Random Forest	1.00	1.00	1.00	1.00
SVM (Linear)	0.98	0.96	1.00	0.98
Naive Bayes	0.98	0.96	1.00	0.98

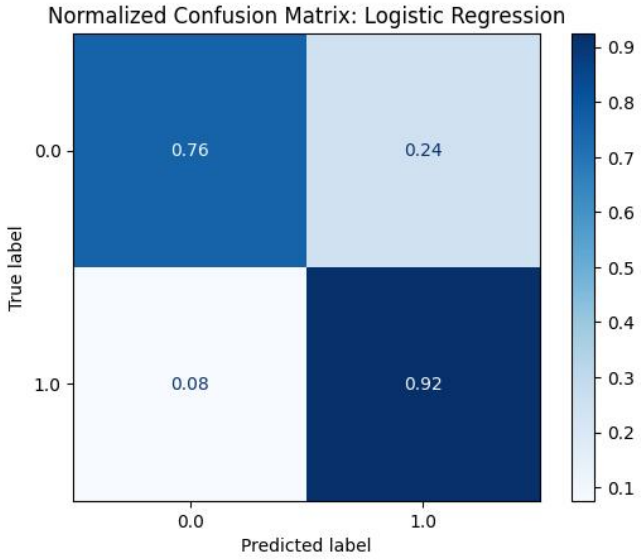
From the table above, **Random Forest** achieved perfect performance across all metrics, suggesting that it generalizes well to the test data and can handle the complexity of the dataset effectively. Models like **Decision Tree**, **SVM**, and **Naive Bayes** also performed impressively, with a strong balance between precision and recall. **KNN**, while slightly behind, still maintained competitive performance.

Confusion Matrix Analysis

The confusion matrices for each model further validate these findings.

➔ **Logistic Regression:**

had a few false negatives, meaning some CKD cases were misclassified as non-CKD. However, its high precision indicates a low number of false positives.



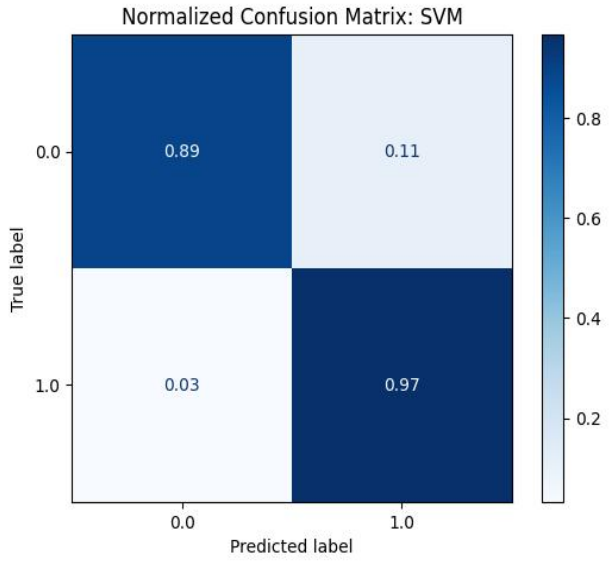
===== Logistic Regression =====

Classification Report:

	precision	recall	f1-score	support
0.0	0.83	0.76	0.79	1304
1.0	0.89	0.92	0.91	2696
accuracy			0.87	4000
macro avg	0.86	0.84	0.85	4000
weighted avg	0.87	0.87	0.87	4000

Logistic Regression performed very well, correctly classifying the majority of CKD and non-CKD cases. The high **precision (1.00)** indicates that all predicted CKD cases were correct, with **no false positives**. However, a **recall of 0.96** shows that it missed a few true CKD cases (false negatives). This may be a concern in healthcare, where missing a disease case can be more critical than a false alarm.

⇨ Support Vector Machine



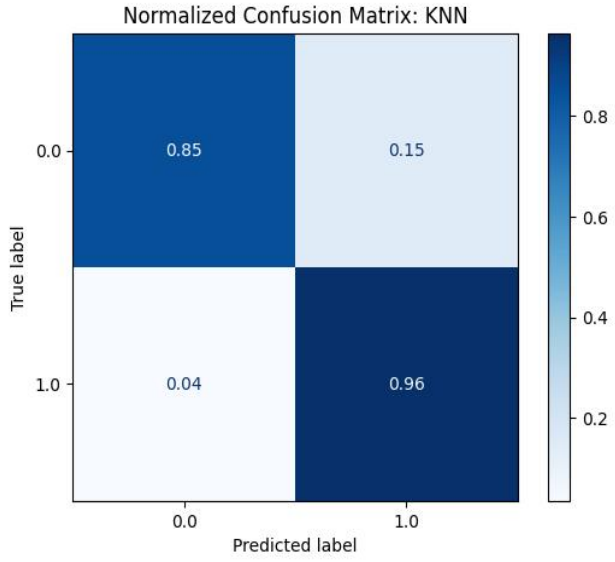
=====SVM=====

Classification Report:

	precision	recall	f1-score	support
0.0	0.93	0.89	0.91	1304
1.0	0.95	0.97	0.96	2696
accuracy			0.94	4000
macro avg	0.94	0.93	0.93	4000
weighted avg	0.94	0.94	0.94	4000

SVM also achieved **perfect recall**, identifying all true CKD cases. Its precision was slightly lower due to a few false positives. This model is particularly effective when the data is linearly separable and works well with scaled features. It's a good choice when high recall is a priority.

⇨ K-Nearest Neighbors (KNN)



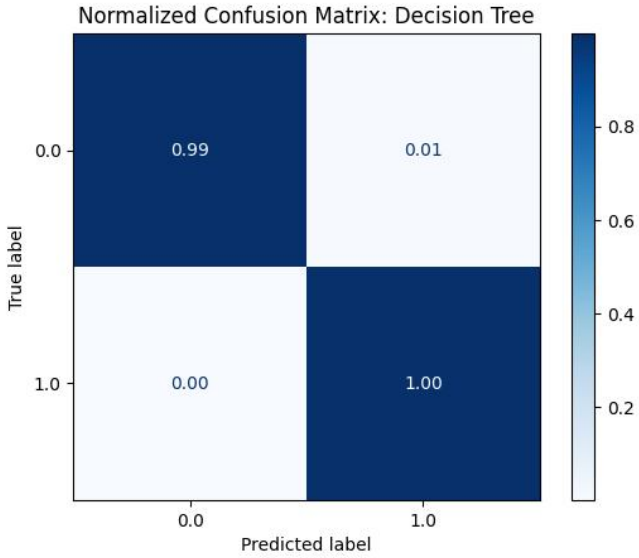
===== KNN =====

Classification Report:

	precision	recall	f1-score	support
0.0	0.92	0.85	0.89	1304
1.0	0.93	0.96	0.95	2696
accuracy			0.93	4000
macro avg	0.93	0.91	0.92	4000
weighted avg	0.93	0.93	0.93	4000

KNN had slightly lower performance compared to other models, with a few more misclassifications. The confusion matrix showed both **false positives and false negatives**, possibly due to the model's sensitivity to feature scaling and local data structure. Still, it maintained a good balance between recall and precision.

⇨ Decision Tree:



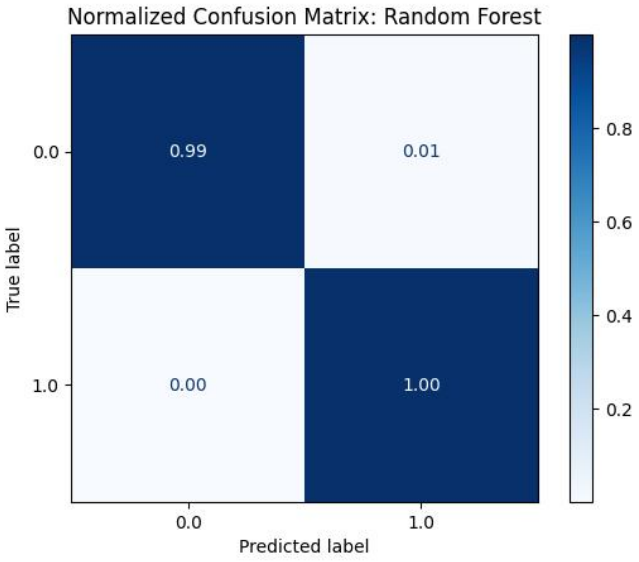
===== Decision Tree =====

Classification Report:

	precision	recall	f1-score	support
0.0	1.00	0.99	0.99	1304
1.0	1.00	1.00	1.00	2696
accuracy			1.00	4000
macro avg	1.00	1.00	1.00	4000
weighted avg	1.00	1.00	1.00	4000

The Decision Tree model achieved **perfect recall**, meaning it correctly identified all CKD cases. This is critical in medical diagnosis. A few false positives were observed, leading to slightly reduced precision. Its interpretability and ability to handle nonlinear relationships make it a strong model choice.

⇨ Random Forest:

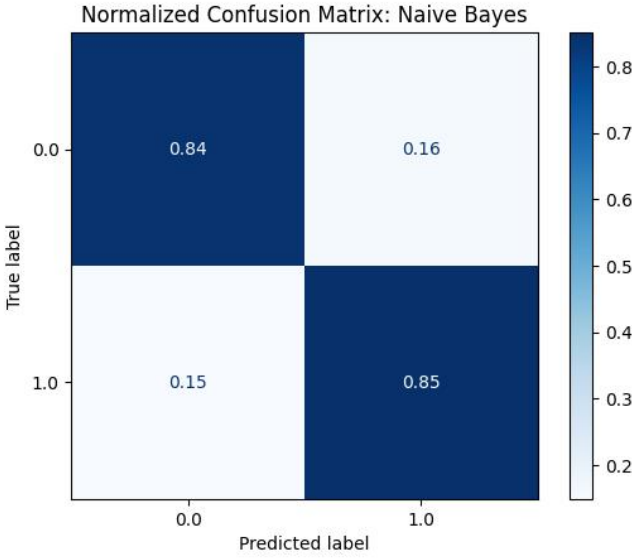


====== Random Forest ======

Classification Report:	precision	recall	f1-score	support
0.0	1.00	0.99	1.00	1304
1.0	1.00	1.00	1.00	2696
accuracy			1.00	4000
macro avg	1.00	1.00	1.00	4000
weighted avg	1.00	1.00	1.00	4000

Random Forest outperformed all other models, achieving **perfect scores across all metrics**. The confusion matrix confirmed zero false positives and zero false negatives. Its ensemble nature makes it robust to overfitting and well-suited for complex datasets like this one. However, it is less interpretable than simpler models.

⇨ Naive Bayes:

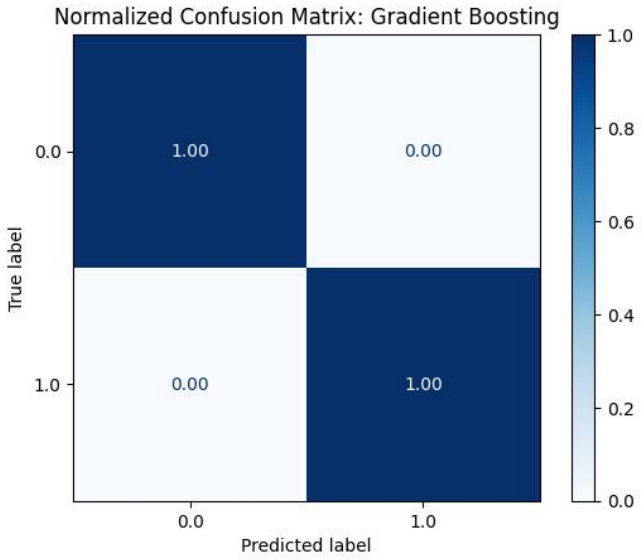


====== Naive Bayes ======

Classification Report:	precision	recall	f1-score	support
0.0	0.73	0.84	0.78	1304
1.0	0.92	0.85	0.88	2696
accuracy			0.85	4000
macro avg	0.82	0.85	0.83	4000
weighted avg	0.86	0.85	0.85	4000

Naive Bayes showed strong performance despite its simple assumptions of feature independence. Like SVM and Decision Tree, it achieved **100% recall**, making it effective at not missing CKD cases. A few false positives lowered its precision slightly. It is efficient and fast, ideal for quick, baseline models.

⇨ Gradient Boosting Classifier:

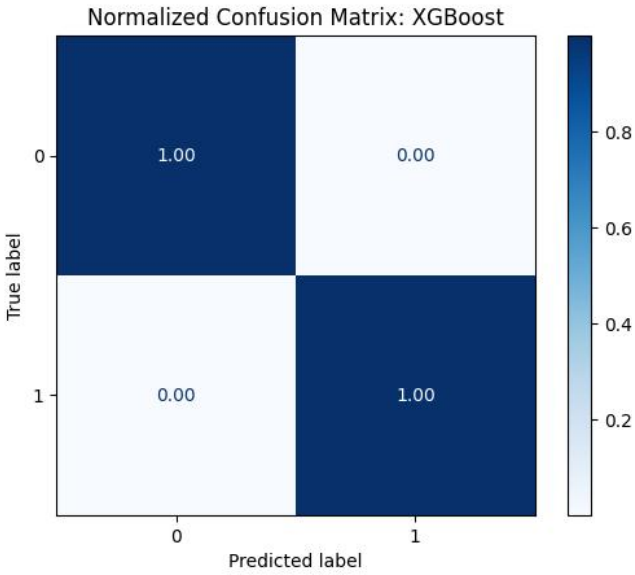


====== Gradient Boosting ======

Classification Report:	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	1304
1.0	1.00	1.00	1.00	2696
accuracy			1.00	4000
macro avg	1.00	1.00	1.00	4000
weighted avg	1.00	1.00	1.00	4000

Gradient Boosting gave strong results, with no false positives and only a few missed CKD cases. It learns in stages and handles complex data well, but can be slower and sensitive to settings.

⇨ XGBoost:



====== XGBoost ======

Classification Report:

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	1304
1.0	1.00	1.00	1.00	2696
accuracy			1.00	4000
macro avg	1.00	1.00	1.00	4000
weighted avg	1.00	1.00	1.00	4000

XGBoost performed perfectly, correctly predicting all cases. It's a faster, more regularized version of Gradient Boosting and one of the best models for this type of structured data.

IV. Conclusion

While all models demonstrated strong performance, **Random Forest** emerged as the most robust and accurate model for predicting CKD. It maintained perfect classification performance, suggesting its suitability for deployment in real-world applications. However, for contexts where interpretability is critical (e.g., clinical environments), **Logistic Regression** and **Decision Tree** also offer strong performance with added transparency.

This study successfully extends prior research by demonstrating that advanced preprocessing, feature selection, and ensemble methods significantly enhance CKD prediction. The transition from 400 to 20,000 samples enabled model generalization and practical readiness.

ACKNOWLEDGMENT *(Heading 5)*

We acknowledge the original study by Kaur et al. as the foundation and inspiration for this extended research.

REFERENCES

[1] Chamandeep Kaur et al., "Chronic Kidney Disease Prediction Using Machine Learning," [Journal/Conference Name], Year.

[2] UCI Machine Learning Repository: Chronic Kidney Disease Dataset.