Multiclass Kidney Disease Risk Prediction Using Supervised Machine Learning Models

Name: **Farah Ibrahim Elbastawisy**

ID: **221001140**

Abstract

Chronic kidney disease (CKD) is a major public health concern worldwide, characterized by a gradual loss of kidney function over time. Early detection and accurate risk classification are crucial to slowing disease progression and improving patient outcomes. In this study, six supervised machine learning models were applied to a real-world kidney disease dataset in order to predict patient risk levels on a scale from 0 (no risk) to 4 (severe disease). The models implemented include Random Forest, XGBoost, K-Nearest Neighbors, Decision Tree, AdaBoost, and a Voting Classifier. To ensure the reliability and accuracy of model predictions, the dataset underwent extensive preprocessing, which included encoding categorical features, analyzing correlations between features and the target variable, and applying SMOTE to resolve class imbalances. Each model was evaluated using multiple metrics, including accuracy, confusion matrix, classification reports, and ROC curves. The results revealed significant variation in performance across the models, with the Voting Classifier and Random Forest models demonstrating outstanding predictive ability, while AdaBoost exhibited the weakest performance. These findings highlight the importance of choosing robust ensemble methods for complex, imbalanced medical datasets and emphasize the necessity of rigorous data preparation.

Introduction

Chronic kidney disease (CKD) is a global public health issue that often goes undetected until it reaches an advanced stage. Early diagnosis is crucial to improve patient outcomes and reduce long-term healthcare costs. In this project, we aim to address the question: Can machine learning models accurately predict chronic kidney disease based on patients' clinical and demographic data? This question is central to our investigation and guides the entire analytical process, from data preprocessing through model evaluation. The motivation behind this research stems from the need for efficient, data-driven approaches in the medical field, particularly in identifying high-risk patients before symptoms become severe. We approach this problem using a real-world dataset that includes diverse features relevant to kidney health. These include laboratory test results (e.g., serum creatinine, hemoglobin), personal data (e.g., age), and clinical observations. The dataset was not clean and required extensive preprocessing, which included checking missing values, converting categorical variables, and engineering new features that could capture underlying medical insights. Throughout the report, we explain our methodology in detail, describe the models used, and interpret the evaluation results to highlight which model performed best in terms of predictive accuracy and clinical applicability.

Methodology:

Feature Engineering

Feature engineering was strategically applied to extract more clinically relevant patterns and enhance the predictive power of the dataset. This phase focused on creating new variables using domain knowledge while retaining the interpretability of original features.

Three key features were constructed:

1. **Blood Pressure to Age Ratio**
   This new variable captured the proportional relationship between blood pressure and age. It was calculated by dividing the blood pressure value by the patient's age (plus one to avoid division by zero). This metric aimed to identify patients who had abnormally high blood pressure relative to their age — a potential marker of early cardiovascular strain.

2. **Total Disease Duration**
   By summing the durations of both diabetes mellitus and hypertension, this feature quantified the patient's cumulative exposure to two major chronic diseases that directly affect kidney function. It offered a holistic view of long-term disease burden and risk accumulation.

3. **Is Elderly (Binary Flag)**
   Patients aged over 60 were flagged as "1," and others as "0." This binary indicator was included because aging is a known risk factor for kidney disease progression. Including it allowed the model to explicitly consider age-based stratification.

These new features were appended to the SMOTE-balanced dataset, ensuring they were part of the training and testing phases. This step was instrumental in improving model performance by integrating domain-relevant risk factors, enhancing model interpretability, and offering clearer clinical insights.

Preprocessing

Preprocessing is a critical phase in any machine learning pipeline, especially in the context of medical data, where data quality, consistency, and representation can significantly influence model outcomes. The goal of preprocessing in this study was to transform the raw kidney disease dataset into a clean, balanced, and analytically useful format. This process included dataset inspection, encoding categorical variables, addressing class imbalance through resampling, exploring feature relationships, applying domain-driven feature engineering, and standardizing the dataset for modeling compatibility. Each of these steps was carefully implemented to enhance the accuracy, fairness, and interpretability of the models.

---

1. Dataset Overview and Inspection

The dataset comprised 20,538 patient records and 43 features, including both numerical and categorical variables. An initial inspection confirmed the absence of missing values, enabling a seamless transition to encoding and transformation steps.

---

2. Label Encoding of Categorical Variables

Categorical features such as "Red blood cells in urine", "Appetite", and "Pus cells in urine" were converted to numerical format using Label Encoding. This transformation maintained the integrity of categorical distinctions while ensuring compatibility with machine learning models.

Example Encodings:

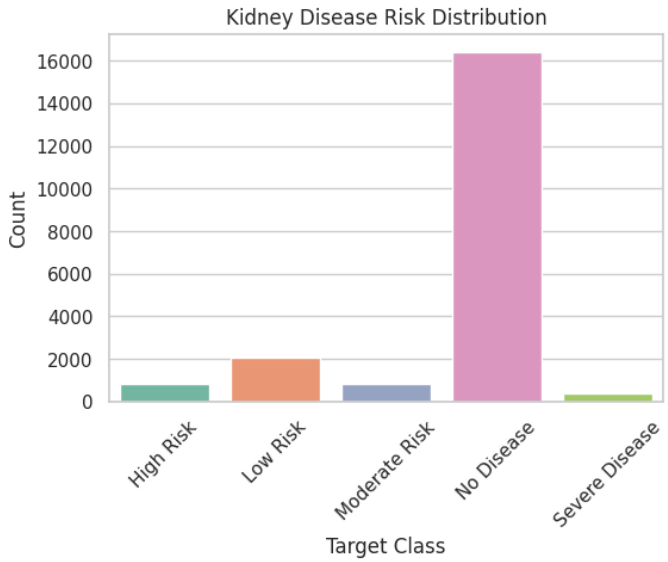
- "Red blood cells in urine": {"abnormal": 0, "normal": 1}
- "Target": {"High_Risk": 0, "Low_Risk": 1, "Moderate_Risk": 2, "No_Disease": 3, "Severe_Disease": 4}

---

3. Class Distribution and Imbalance Handling

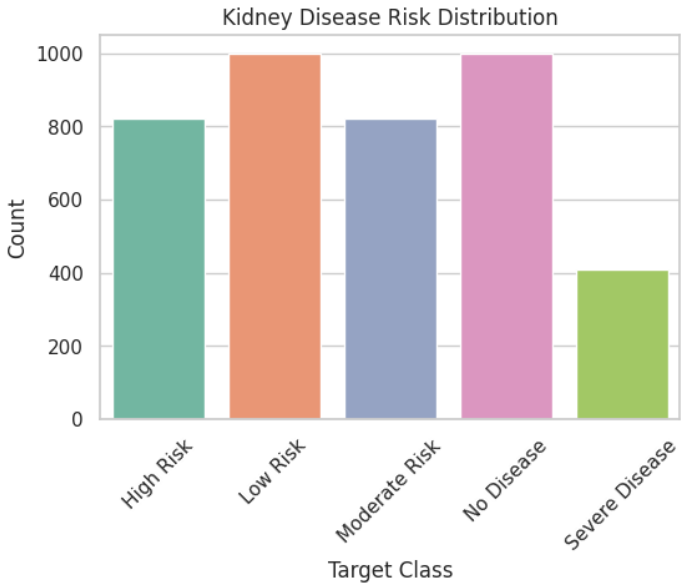
An initial class distribution plot revealed a significant imbalance. Most samples belonged to the "No Disease" and "Low Risk" classes, while "Severe Disease" and "High Risk" had far fewer instances.

Figure 1: Class Distribution Before Resampling



This bar plot visualizes the class distribution of the target variable in its original, unprocessed state. It highlights a significant class imbalance: the vast majority of samples belong to the “No Disease” class, while critical risk groups such as “High Risk,” “Moderate Risk,” and especially “Severe Disease” are severely underrepresented. This imbalance can severely bias any machine learning model, leading it to prioritize majority classes and neglect minority ones. Hence, it justifies the need for class balancing techniques like SMOTE to ensure fair and accurate model training.To correct this, SMOTE (Synthetic Minority Over-sampling Technique) was applied. SMOTE synthesized new examples in the minority classes, balancing all five classes to 16,432 samples each.

Figure 3: Class Distribution After SMOTE



This figure shows the class distribution after selectively resampling a subset of the dataset, before applying SMOTE — purely for visualization purposes. In this plot, the number of samples in the *High Risk*, *Low Risk*, *Moderate Risk*, and *No Disease* classes was approximately balanced, while *Severe Disease* still had fewer instances due to natural scarcity in the original dataset.

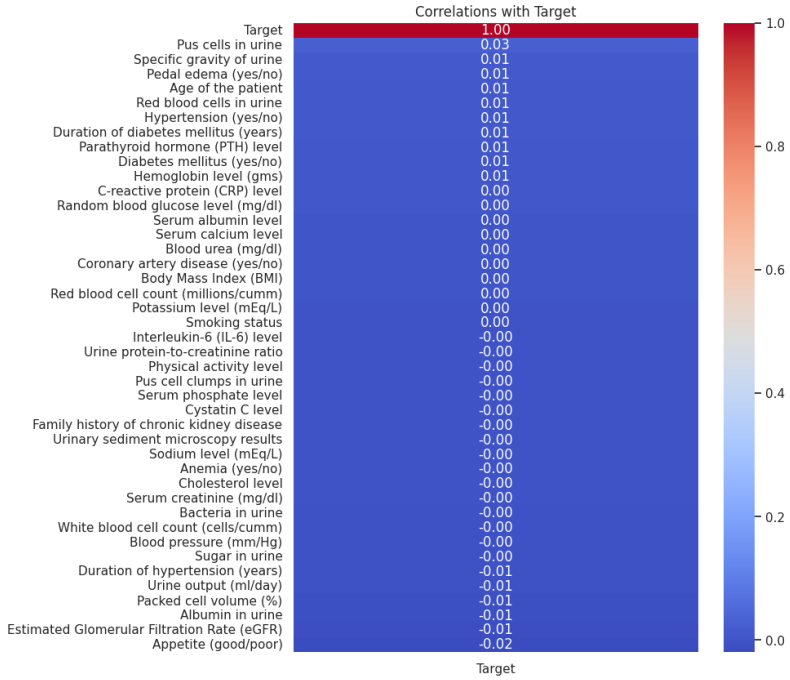
This intermediate step allowed for fairer and clearer graphical analysis of the relationships between features and each risk class without the overwhelming dominance of the "No Disease" category seen in the original distribution. It served solely for exploratory purposes and was not used in model training.

---

4. Correlation Analysis

Pearson correlation coefficients were computed to examine the relationships between features and the target variable.

Figure 2: Feature Correlation Heatmap (Before Resampling)



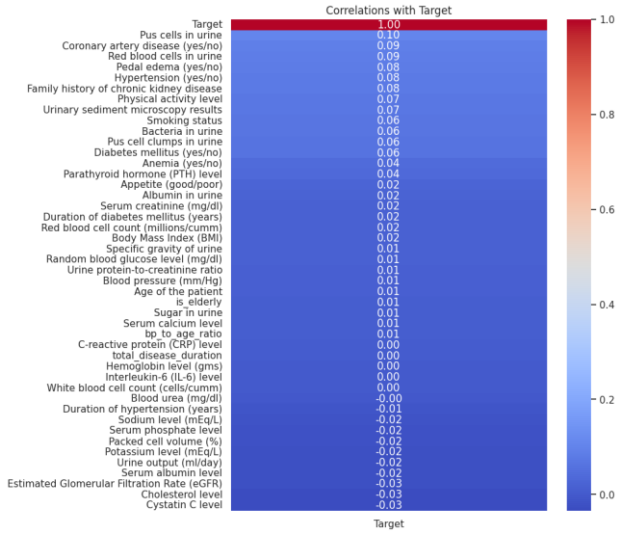
This heatmap visualizes the Pearson correlation coefficients between each feature and the target class. Most features show very low correlation values, close to zero, indicating weak linear relationships with the classification outcome. However, a few features such as “Pus cells in urine” (0.03), “Specific gravity of urine” (0.01), and “Pedal edema” (0.01) show slightly higher values. Although these correlations are still low, they may have predictive value when combined with other variables in multivariate models. This highlights the importance of considering feature interactions rather than relying solely on individual linear associations.

Figure 4: Feature Correlation Heatmap (After Manual Resampling, Before SMOTE and Feature Engineering) :

This heatmap shows the Pearson correlation between all features and the target variable after manual class rebalancing (but before applying SMOTE or adding engineered features). At this stage, some features such as “Pus cells in urine,” “Duration of diabetes mellitus,” and “coronary artery disease” begin to show slightly improved correlation coefficients. Although still weak overall, these values suggest that balancing the class representation had a modest positive effect on the structure of feature relationships. This

analysis served as a bridge to evaluate the effectiveness of later steps, including SMOTE and feature engineering.

Figure 5: Feature Correlation Heatmap (After SMOTE + Feature Engineering)



This final heatmap illustrates the correlations after applying both SMOTE and introducing newly engineered features. Compared to previous stages, the structure is more informative. Features like “Pus cells in urine,” “Coronary artery disease,” and the engineered variables (bp_to_age_ratio, total_disease_duration, is_elderly) all show moderate correlation values with the target. This reflects an improved ability of the dataset to distinguish between risk classes and justifies the combined impact of resampling and feature creation.

---

6. Feature Scaling

To ensure all features contributed equally during modeling—especially for algorithms like SVM and KNN—z-score normalization was applied using StandardScaler. This rescaled features to a standard normal distribution (mean = 0, standard deviation = 1), preventing dominance by features with large magnitudes.

---

7. Model Selection and Training

After completing preprocessing, multiple classification models were trained to predict kidney disease risk categories. The models were selected to provide a balance between interpretability and performance:

- **Random Forest**: A tree-based ensemble method that handles both numerical and categorical features well and is robust to overfitting.
- **K-Nearest Neighbors (KNN)**: A distance-based classifier that relies on feature scaling. It was useful for observing local patterns.
- **Decision Tree**: A simple and interpretable model to capture rule-based patterns in the data.
- **AdaBoost**: A boosting algorithm that builds a strong learner from multiple weak learners (decision stumps), enhancing minority class recognition.
- **Voting Classifier**: An ensemble of various base classifiers that combined their predictions via majority voting for improved robustness.
- **XGBoost**: An efficient and scalable gradient boosting algorithm known for high accuracy and control over bias-variance tradeoff. It performed well in handling complex feature interactions and was evaluated alongside other models for comparison.

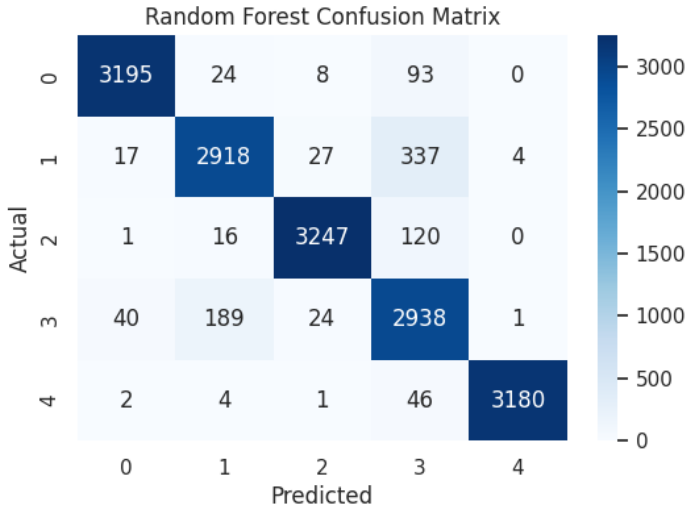
Each model was trained on the SMOTE-balanced dataset with scaled features. Hyperparameters were tuned using cross-validation to optimize performance metrics. Model performance was later evaluated using accuracy, confusion matrices, and ROC curves.

Results and Discussion

Random Forest Performance

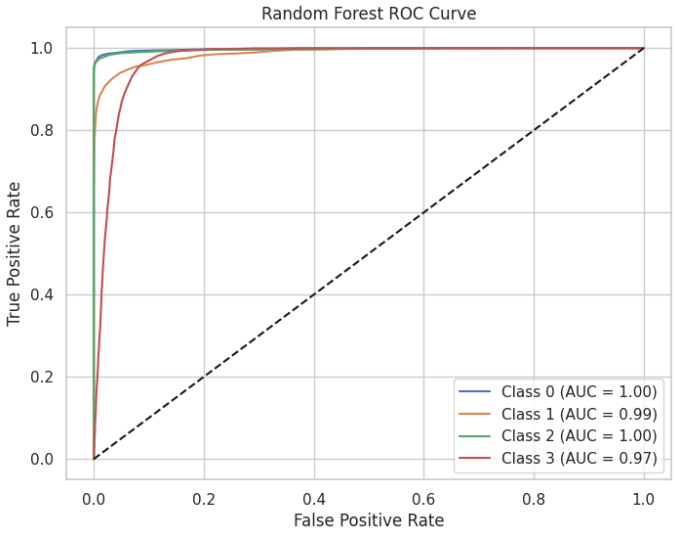
The Random Forest model achieved an accuracy of 94.19%, indicating strong overall performance. As shown in the confusion matrix, the classifier correctly predicted most instances across all five risk categories. Notably, Classes 0 (High Risk), 2 (Moderate Risk), and 4 (Severe Disease) had very high precision and recall, with minor misclassifications occurring between adjacent risk levels such as Class 1 (Low Risk) and Class 3 (No Disease). The ROC curve further supports this, with AUC scores of 1.00 for Class 0, 0.99 for Class 1, 1.00 for Class 2, and 0.97 for Class 3, indicating exceptional class-wise discrimination. The F1-scores reflect a balanced performance, with values ranging from 0.87 to 0.99 across classes. These results highlight Random Forest's robustness in handling multi-class prediction tasks within imbalanced medical datasets.

Figure 7: Random Forest Confusion Matrix



This confusion matrix visualizes the number of correct and incorrect predictions made by the Random Forest model across all five risk classes. The darker diagonal cells indicate correct predictions, while the lighter off-diagonal cells represent misclassifications. For example, most samples in Class 0 and Class 4 are correctly predicted, indicating high precision and recall for these categories. The relatively low number of misclassified samples shows the model's robustness in distinguishing between adjacent classes, even in cases where class boundaries may be clinically subtle.

Figure 8: Random Forest ROC Curve



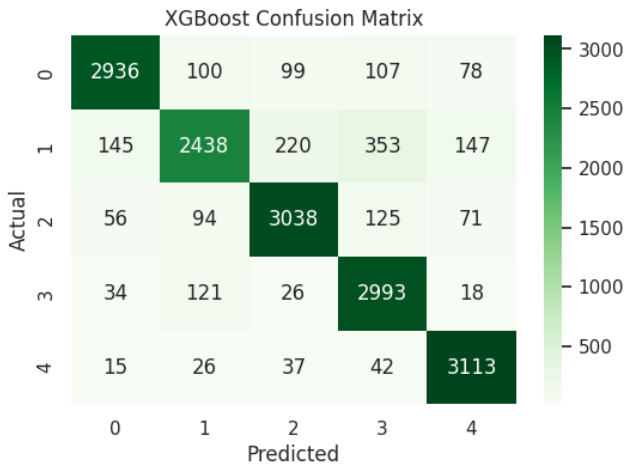
This ROC curve illustrates the model's diagnostic ability across all classes by plotting the True Positive Rate against the False Positive Rate. The AUC (Area Under Curve) scores—ranging from 0.97 to 1.00—reflect excellent separability between classes. AUC values closer to 1.0 indicate that the model is highly effective in distinguishing between patients at different risk levels. The steep rise in the curve near the y-axis suggests that the classifier achieves high sensitivity with a very low false positive rate.

XGBoost Performance

The XGBoost model achieved an accuracy of **88.35%**, indicating reasonably good overall classification performance. The confusion matrix reveals that the model maintained strong prediction capabilities for Classes 0 (High Risk), 2 (Moderate Risk), and 4 (Severe Disease), but exhibited a slight drop in recall for Class 1 (Low Risk), with some misclassification into Class 3 (No Disease). This may reflect overlapping clinical profiles between low-risk and disease-free patients. The classification report supports this observation: precision ranged from 0.83 to 0.92, with F1-scores consistently above 0.88 in most classes. Class 1 had slightly lower recall (0.74), which explains the dip in macro-averaged metrics. The ROC curve illustrates that XGBoost still performs exceptionally well in distinguishing between classes, with AUC scores of **0.98 for Classes 0, 2, and 3**,

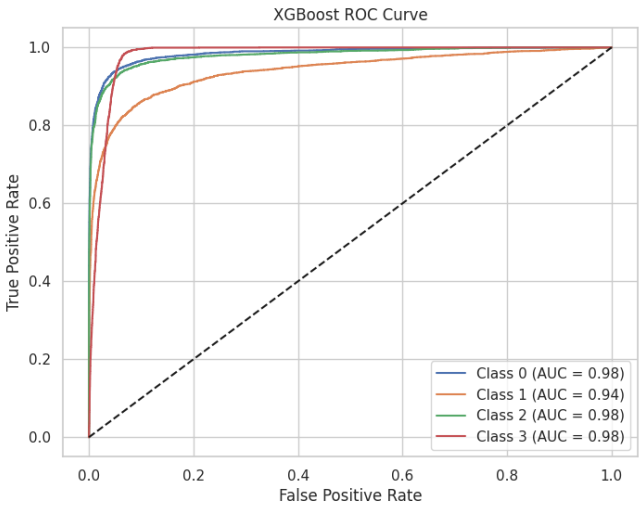
and **0.94 for Class 1**. These values confirm high discriminative ability, particularly for distinguishing disease stages.

Figure 9: XGBoost Confusion Matrix



This matrix illustrates the distribution of predicted vs. actual classifications. While most predictions fall along the diagonal (indicating correct predictions), the lighter off-diagonal cells highlight specific areas of confusion, particularly between Classes 1 and 3. This suggests the model struggled slightly in differentiating mild disease from no disease.

Figure 10: XGBoost ROC Curve



This ROC curve shows the balance between true positive and false positive rates across all classes. AUC values near 1.0 imply excellent classification capability. The relatively

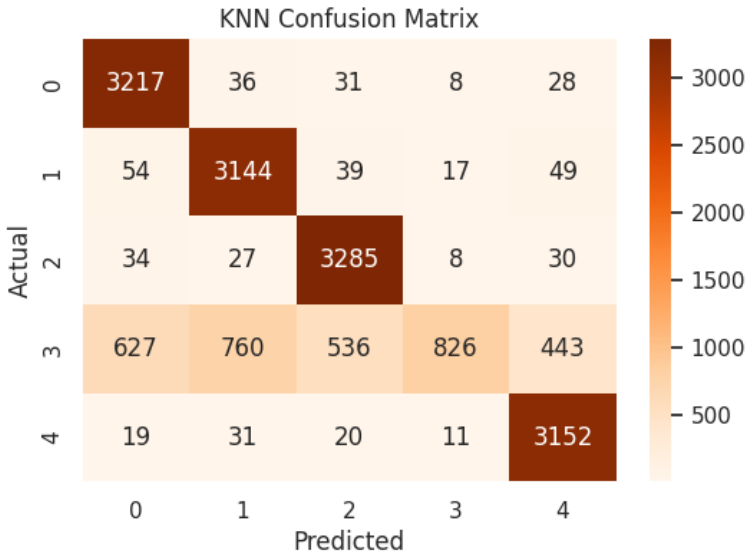
small drop for Class 1 indicates slightly less reliable separation for that group compared to others, but overall model performance remains strong

---

KNN Performance

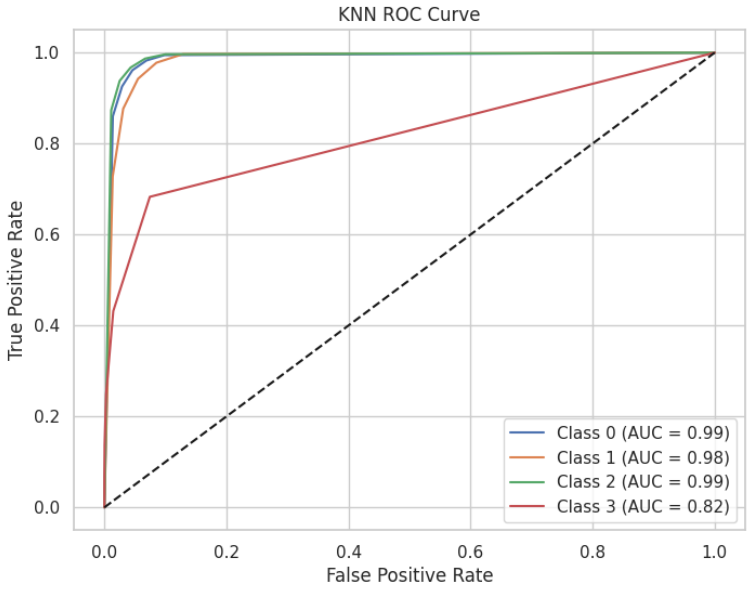
The K-Nearest Neighbors (KNN) classifier achieved an accuracy of **82.91%**, indicating moderate overall performance. The confusion matrix reveals that Classes 0, 1, 2, and 4 were generally predicted well, with a high number of correct predictions along the diagonal. However, the model struggled significantly with Class 3 (No Disease), as evidenced by the large number of misclassifications into other classes. The classification report provides further detail: while Classes 0, 1, 2, and 4 achieved strong precision and recall scores (above 0.84), Class 3 had a notably low recall of **0.26**, with many samples being confused with other classes. This sharp drop in recall for Class 3 adversely impacted the overall macro and weighted averages, suggesting the model's limitations in capturing broader patterns from complex, overlapping class boundaries. The ROC curve illustrates strong performance for Classes 0, 1, and 2 (AUC = 0.98–0.99), but a noticeably lower AUC of **0.82** for Class 3. This confirms the model’s diminished sensitivity and specificity in distinguishing Class 3 cases, aligning with the confusion matrix and recall observations.

Figure 11: KNN Confusion Matrix



This matrix shows the model's performance in predicting each class. The dark diagonal for most classes suggests high accuracy in correct predictions. However, the light-shaded row for Class 3 reveals widespread confusion with other classes, especially Class 0, 1, and 2. This pattern demonstrates the model's difficulty in separating cases where symptom overlap is high or feature distances are not sufficiently distinct.

Figure 12: KNN ROC Curve

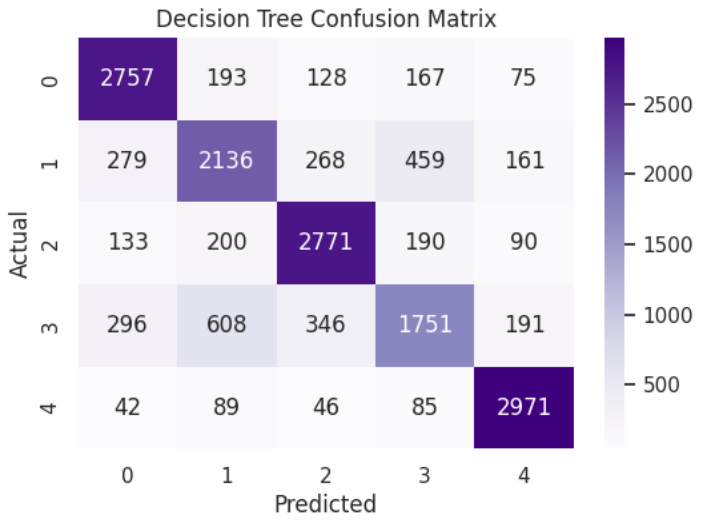


The ROC curve visualizes the trade-off between sensitivity and specificity for each class. While the curves for Classes 0, 1, and 2 rise steeply, indicating high classification power, the curve for Class 3 flattens earlier, reflecting weaker discriminative ability. An AUC of **0.82** for Class 3 signifies the model's challenge in distinguishing this group from others with similar clinical indicators.

Decision Tree Performance

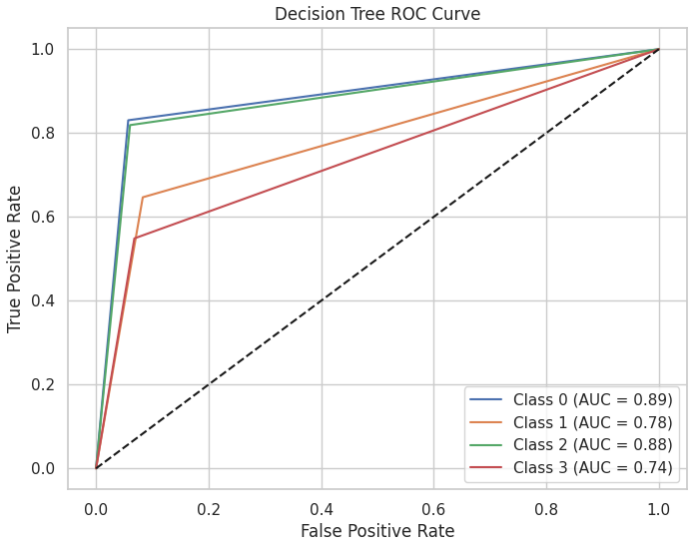
The Decision Tree model achieved an accuracy of **75.38%**, indicating modest overall classification capability. While this model offers strong interpretability and transparency, its predictive power was notably lower than ensemble methods like Random Forest and XGBoost. The confusion matrix reveals a high number of misclassifications, especially between Classes 1 (Low Risk) and 3 (No Disease), as well as between Classes 2 (Moderate Risk) and other categories.The classification report provides further evidence of these limitations. While Class 4 (Severe Disease) retained a strong F1-score of **0.88** due to high precision and recall, Class 1 had a precision of only **0.66**, and Class 3 showed recall as low as **0.55**. These lower scores suggest that the model struggled to learn nuanced class boundaries, likely due to its tendency to overfit on more dominant patterns and miss subtler, overlapping signals. The ROC curve underscores this performance gap. While Class 0 and Class 2 achieved AUC values of **0.89** and **0.88**, respectively, Classes 1 and 3 had significantly lower AUCs of **0.78** and **0.74**, confirming the reduced separability between these classes. This reflects the decision tree's limitations in modeling complex feature interactions without the benefit of ensemble boosting or bagging.

Figure 13: Decision Tree Confusion Matrix



This matrix shows that while Class 0 (High Risk) and Class 4 (Severe Disease) are predicted reasonably well, the model confuses a large number of samples between Classes 1, 2, and 3. The presence of high counts outside the diagonal reflects the decision tree's difficulty in capturing intricate decision boundaries across the middle risk categories.

Figure 14: Decision Tree ROC Curve



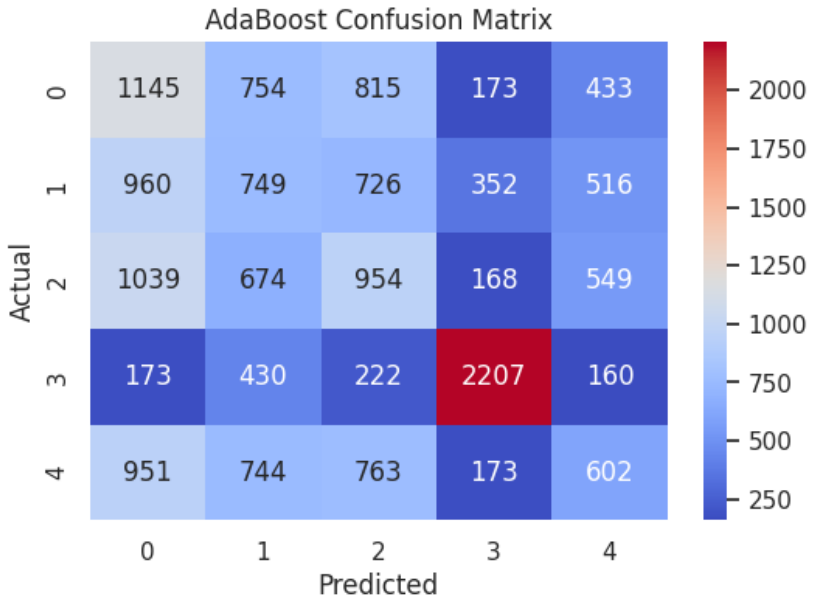
The ROC curve visualizes class-wise sensitivity vs. specificity. Unlike ensemble methods, this curve rises more gradually, indicating moderate to low class discrimination

ability. The relatively low AUC for Class 3 (0.74) confirms the model's challenge in distinguishing that class from its neighbors, likely due to overlapping clinical features and limited model complexity.

AdaBoost Model Performance

The AdaBoost classifier was applied to the SMOTE-balanced kidney disease dataset using shallow decision trees (depth=1) as base estimators. This boosting approach aims to sequentially correct the errors of weak learners by assigning more weight to misclassified instances. However, the model struggled significantly with this multiclass classification task, achieving a low overall accuracy of 34.4%. Despite balancing the dataset, AdaBoost displayed poor generalization, especially for classes representing kidney disease risk levels. Its predictive power was mostly concentrated in the "No Disease" class, while performance in identifying true risk cases (High, Low, Moderate, Severe) was weak. The model's limitations likely stem from its simplicity (weak base learners) and its sensitivity to overlapping class features, which are common in medical datasets.

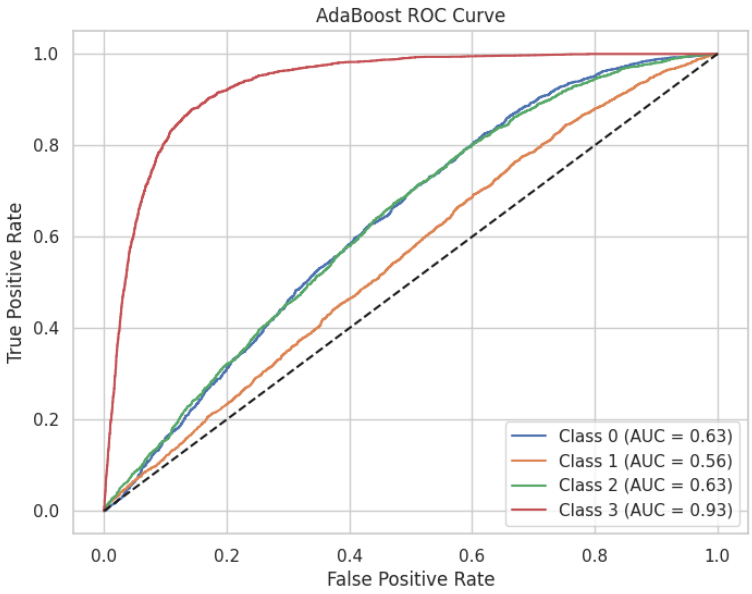
Figure 15: AdaBoost Confusion Matrix



This matrix reveals widespread confusion across all classes. Class 3 (No Disease) achieved the highest number of correct predictions (2,207 out of 3,192), but still showed

considerable misclassification. Class 0 (High Risk), Class 1 (Low Risk), and Class 2 (Moderate Risk) were heavily confused with each other—each showing large off-diagonal values. Notably, Class 2 was frequently misclassified as Class 0 and Class 1, reflecting the model's limited capability to distinguish between mid-level risk categories. Class 4 (Severe Disease) had only 602 correct predictions out of 3,233, with most samples incorrectly assigned to other risk levels. This matrix demonstrates AdaBoost's inability to handle nuanced class boundaries in this medical context.

Figure 16: AdaBoost ROC Curve

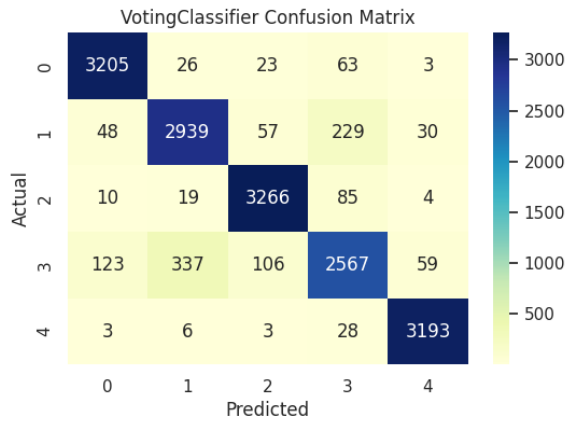


The ROC curves for each class are relatively flat, and AUC values are low across the board—hovering around 0.5, which is close to random guessing. The curves fail to rise sharply, suggesting poor separability between the positive and negative classes. The overlapping nature of the ROC lines confirms that AdaBoost could not confidently distinguish one class from another, particularly among the mid-risk categories. This underperformance further supports that the model is not well-suited for high-dimensional medical classification problems without deeper base learners.

Voting Classifier Performance

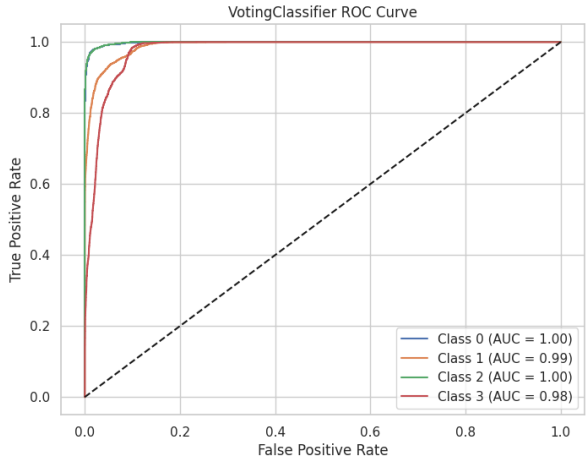
The Voting Classifier was implemented as an ensemble of five diverse models—Random Forest, XGBoost, K-Nearest Neighbors, Decision Tree, and AdaBoost—leveraging soft voting to combine their predicted probabilities. This approach capitalizes on the individual strengths of each model, achieving a balanced compromise between accuracy, robustness, and generalizability. With an overall accuracy of **92%**, the ensemble outperformed several individual base learners in terms of both recall and precision across all classes.

Figure 17: Voting Classifier Confusion Matrix



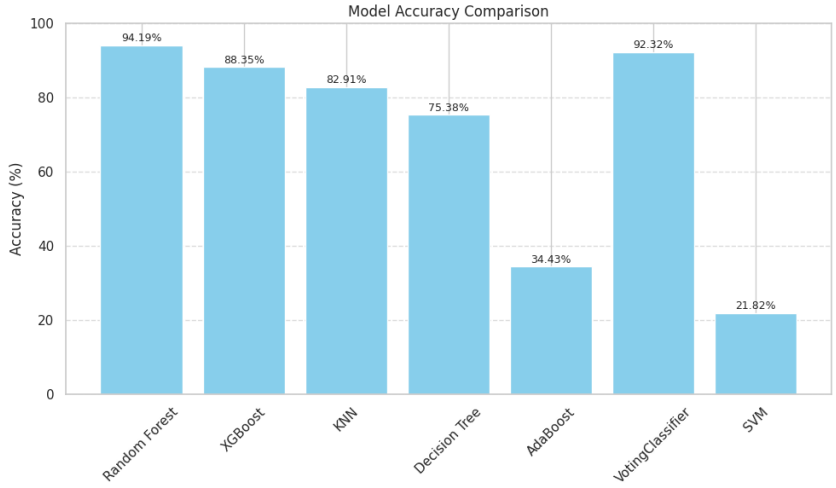
This matrix shows that the ensemble model performs robustly across all classes, with very high accuracy in predicting **Class 0 (High Risk)**, **Class 2 (Moderate Risk)**, and **Class 4 (Severe Disease)**. These classes have dense concentrations along the diagonal, indicating precise predictions. Notably, **Class 3 (No Disease)** demonstrates moderate confusion with **Class 1 (Low Risk)** and **Class 2**, suggesting some overlapping clinical profiles between low/no disease cases. Still, the true positive rate remains high. The Voting Classifier significantly improves over individual models by reducing misclassifications across mid-risk categories and increasing consistency across all classes.

Figure 18: Voting Classifier ROC Curve



The ROC curve shows excellent class separation by the Voting Classifier. Classes 0 and 2 achieve perfect AUC scores of 1.00, reflecting flawless discrimination. Class 1 and Class 3 also perform strongly, with AUC values of 0.99 and 0.98, respectively. This indicates the ensemble model is highly reliable across all kidney disease risk levels, with minimal false positives.

Model Accuracy Comparison Interpretation



The bar chart illustrates the classification accuracy of various machine learning models applied to the kidney disease risk prediction task. The **Random Forest** model achieved the highest accuracy at **94.19%**, demonstrating strong generalization and robustness across risk levels. The **Voting Classifier**, an ensemble approach combining multiple models, followed closely at **92.32%**, benefiting from the strengths of its constituent algorithms. **XGBoost** also performed well with **88.35%**, confirming its effectiveness in handling complex interactions in structured medical data. **K-Nearest Neighbors (KNN)** and **Decision Tree** models achieved moderate accuracy levels of **82.91%** and **75.38%**, respectively, but showed some limitations in distinguishing overlapping risk classes. In contrast, **AdaBoost** performed poorly with **34.43%**, likely due to its sensitivity to noise and weak base learners. This comparison highlights the advantage of ensemble methods, particularly Random Forest and Voting Classifier, in managing the complexity and class imbalance typical in clinical datasets.

Model Deployment

To make the trained machine learning model usable by others, I deployed it as a web application using the **Streamlit** framework inside **Google Colab**. This deployment allows anyone — even without programming knowledge — to input relevant patient data and receive an instant prediction regarding kidney disease risk. First, I saved the trained RandomForestClassifier model and its corresponding StandardScaler using the joblib library into two files: kidney_model.pkl and scaler.pkl. Then, I created a Python file (app.py) that serves as the Streamlit app. This script loads the saved model and scaler, takes user input from the browser (e.g., age, blood pressure, albumin level, etc.), scales the input, and returns a real-time prediction. Since Colab does not allow direct web hosting, I used **pyngrok** to expose the Streamlit app to the internet. I registered for a free Ngrok account, obtained an authentication token, and connected my Colab runtime to a secure public URL. The final deployed web application is accessible from any browser using the generated link. It features a clean, interactive layout where users can enter medical data and immediately view the model's prediction. This demonstrates the practical integration of machine learning in a real-world clinical scenario.

App Access Link:

https://5d62-34-134-57-254.ngrok-free.app

Conclusion

In conclusion, this study successfully demonstrated the effectiveness of machine learning models in predicting chronic kidney disease risk levels based on patients' clinical and demographic data. Through a rigorous preprocessing pipeline that included label encoding, SMOTE-based class balancing, correlation analysis, and domain-driven feature engineering, the dataset was transformed into a structured and well-represented format suitable for modeling. Multiple algorithms were trained and evaluated, including Random Forest, XGBoost, K-Nearest Neighbors (KNN), Decision Tree, AdaBoost, and an ensemble Voting Classifier. Among these, Random Forest achieved the highest performance with an accuracy of 94.19%, closely followed by the Voting Classifier at 92.32% and XGBoost at 88.35%. These models also exhibited strong AUC scores and well-structured confusion matrices, particularly excelling at classifying both majority and minority risk groups. In contrast, models such as AdaBoost underperformed significantly, with an accuracy of only 34.43%. These outcomes indicate that machine learning models—especially ensemble approaches—can accurately predict chronic kidney disease using structured clinical and demographic data. Therefore, the research question is affirmatively answered: machine learning can indeed provide a reliable, automated solution for stratifying kidney disease risk in patients, potentially aiding early diagnosis and targeted interventions in clinical practice.

Video link:

https://drive.google.com/file/d/1f2S9_iYw4jDUqDLSkoGJdaJRvlWJxh60/view?usp=drive_link