A Comparative Analysis of Classification Techniques for Chronic Kidney Disease

Prajwal G Alewoor, Niranjan S Naik, Lalam Aakash from Department of Information Science and Engineering,
Under the guidance of Dr. Sreelatha R, Assistant Professor of information Science and Engineering,
BMS College of Engineering, Bangalore, 560019

K E Y W O R D S

Machine Learning
Classifiers
Sampling Techniques
KNN
Decision Tree
SMOTE

A B S T R A C T

Chronic Kidney Disease (CKD) is a serious medical issue that has a considerable effect on patient well-being and healthcare infrastructure. Accurate CKD prediction can help with early intervention and enhance patient results. In order to predict CKD using a dataset, we used a variety of classification algorithms in this study, including K-Nearest Neighbour (KNN), Decision Tree, Logistic Regression, and Naive Bayes. In addition, we assessed how well four sampling techniques—namely, SMOTE, under-sampling, over-sampling, and without sampling—handled class imbalance.

Purpose: This study provides an in-depth comparison of different classification methods for predicting chronic kidney disease (CKD) using a dataset. The study assesses how well various classifiers perform both with and without sampling strategies and with various missing value imputing algorithms. The best method for CKD prediction is determined by comparing the accuracy results from various classifiers.

Methodology: After imputed missing data with four different sampling strategies (without sampling, SMOTE, under sampling, and oversampling), several classifiers, such as Decision tree, KNN, Logistic regression, and naive Bayes are utilized.

Findings: Our findings reveal notable variations in accuracy across different classifiers and sampling techniques. The Decision Tree classifier achieved the highest accuracy of 98%, followed by Naive Bayes (96%), Logistic Regression (88%), and KNN (56%). Regarding the sampling techniques, over-sampling and SMOTE consistently improved the accuracy of the classifiers, achieving accuracies ranging from 69% to 95%, while under-sampling demonstrated moderate improvements in certain cases.

The results of this study have important repercussions for CKD predictions. The best combination for precise CKD prediction is the Decision Tree classifier combined with over-sampling or SMOTE. This emphasizes how crucial it is to take class imbalance into account when selecting a classification technique as well as the right sampling technique to increase prediction accuracy. An early intervention, individualized treatment programmes, and better patient outcomes are made possible by the accurate prediction of CKD.

Cross ponding Author: Tel: +0-000-000-000; fax: +0-000-000-000
Email address:

1. Introduction

1.1 Background and Significance:

A common and devastating medical illness, chronic kidney disease has a big impact on people's lives and healthcare systems. Accurate CKD prediction can aid in early intervention, better patient outcomes, and management. For CKD prediction, a variety of classification approaches have been used, but their relative effectiveness has not been well investigated.

Our research demonstrates the efficacy of several classifiers when combined with various sampling strategies and imputing methodologies, allowing us to determine the most precise strategy for CKD prediction.

1.2 Objectives:

This study uses a dataset to examine the effectiveness of various classification approaches for predicting chronic kidney disease. We specifically look into the precision of naive Bayes classifiers, logistic regression, decision trees, and KNN classifiers. In addition, we examine how sampling methods such as SMOTE, undersampling, oversampling, and without sampling affect classification accuracy.

1.3 Research Questions:

a. How does the accuracy of different classification techniques vary for CKD prediction?

b. What is the impact of sampling techniques on classification accuracy?

c. Which classification technique, combined with a specific sampling technique, yields the highest accuracy in CKD prediction?

1.4 Organization of the Paper:

The paper is organized as follows: Section 2 provides a literature review on chronic kidney disease and classification techniques. Section 3 describes the dataset, pre-processing steps, and the classification techniques employed. Section 4 presents the results and discussion, comparing the accuracy of different classifiers and the impact of sampling techniques. Section 5 discusses the limitations of the study and provides recommendations for future research. Finally, Section 6 concludes the paper and summarizes the key findings.

2.Literature survey:

Chronic Kidney Disease (CKD) is a widespread and severe health condition that affects a significant portion of the global population. The UCI repository dataset for Chronic Kidney Disease (CKD) was used in this study. Early detection and accurate prediction of CKD play a crucial role in improving patient outcomes, optimizing treatment strategies, and reducing the burden on healthcare systems. In recent years, machine learning techniques have gained attention as valuable tools for predicting CKD and assisting in clinical decision-making. This literature review aims to provide an overview of the existing research on CKD prediction using classification techniques and sampling techniques. This literature review highlights the significance of accurate prediction in early CKD detection. It discusses the challenges involved in CKD prediction and the existing literature on classification techniques for CKD prediction. The research gaps identified emphasize the need for a comprehensive comparative analysis of classification techniques and the exploration of sampling techniques to address class imbalance. This study aims to contribute to the existing knowledge by providing insights into the most effective classification and sampling techniques for CKD prediction.

In Ref [04] the authors have mentioned regarding classifying the patients with kidney diseases based on three performance metrics i.e, accuracy, sensitivity, specificity using support vector machine classifier using 10-fold cross-validation without any sampling methods and the features were selected through KNN classifier. It highlights the importance of early detection and accurate classification of CKD to improve patient outcomes and guide appropriate treatment strategies. It emphasizes the use of machine learning algorithms as effective tools for CKD prediction and classification. It provides an overview of the relevance of machine learning techniques for CKD classification. It specifically highlights the use of SVM as a classifier and the importance of feature selection using the KNN algorithm.

Three machine learning classifiers i.e. Logistic regression, Decision tree, Support vector machine for analysis and then used bagging ensemble method to improve the results of the developed model [07]. The authors also discuss the use of the bagging ensemble method to improve the performance of the developed model. Bagging is an ensemble technique that combines multiple base models (in this case, LR, DT, and SVM) to make predictions. By aggregating the predictions of multiple models, the ensemble model aims to enhance the accuracy and robustness of the prediction results. The research paper highlights the utilization of LR, DT, and SVM classifiers for chronic kidney disease prediction. The bagging ensemble method is employed to potentially improve the accuracy of the model. The absence of sampling methods suggests that the researchers focused on the original distribution of the dataset. Lastly, the authors suggest exploring feature selection methods to optimize the model's performance in future research.

2.1 Significance of Early CKD Detection:

Early detection of CKD allows healthcare professionals to implement appropriate interventions, slow down disease progression, and prevent complications. It enables personalized treatment plans and lifestyle modifications, leading to improved patient outcomes and reduced healthcare costs. Therefore, accurate prediction of CKD at an early stage is of utmost importance.

2.2 Challenges in CKD Prediction:

CKD prediction poses several challenges due to the complex and multifactorial nature of the disease. The presence of numerous risk factors, both genetic and environmental, makes it challenging to identify the most relevant features for accurate prediction. Additionally, class imbalance, where the number of CKD cases is significantly smaller than non-CKD cases, poses a challenge for classifiers to accurately predict the minority class. Handling

Cross ponding Author: Tel: +0-000-000-000; fax: +0-000-000-000
Email address:

missing values and selecting appropriate evaluation metrics are also important considerations in CKD prediction studies.

2.3 Previous Studies on CKD Prediction Using Classification Techniques:

Several studies have explored the application of classification techniques in predicting CKD. Various algorithms, including but not limited to Logistic Regression, Decision Trees, Support Vector Machines, and Naive Bayes, have been employed for this purpose. These studies have reported varying levels of success in accurately predicting CKD. However, there is a need for a comprehensive comparative analysis of these techniques to identify the most effective approach for CKD prediction.

2.4 Research Gaps and Objectives of the Study:

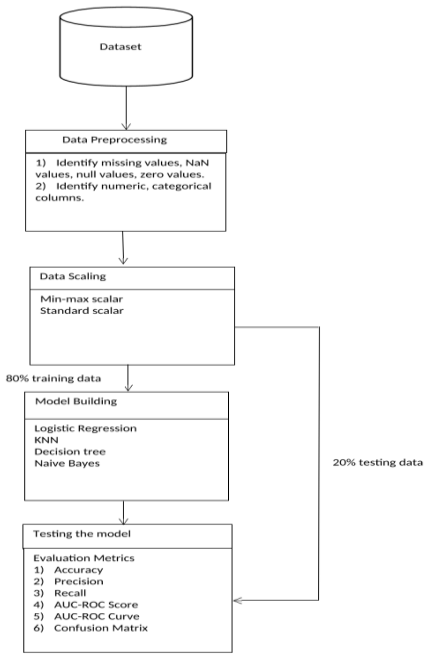
Despite the progress in CKD prediction using classification techniques, there are still research gaps that need to be addressed. Existing studies often focus on individual classification techniques without providing a comprehensive comparative analysis. Moreover, the impact of class imbalance and the effectiveness of sampling techniques in handling this issue have not been extensively explored. This study aims to bridge these gaps by conducting a comparative analysis of multiple classification techniques and evaluating the performance of different sampling techniques in CKD prediction.

3.Methodology:

3.1 Dataset Description:

The Chronic Kidney Disease dataset available in the UCI Machine Learning Repository provides information related to the diagnosis of chronic kidney disease. The dataset was collected from multiple hospitals and includes various clinical and laboratory measurements. It aims to aid in the prediction and analysis of CKD based on the provided attributes.

The dataset consists of a total of 400 instances or patient records.



The CKD dataset contains a total of 25 features, including both numerical and categorical variables

a.Numerical features (age, blood pressure, specific gravity, albumin, sugar, rbc, pus cells, pus cell clumps, bacteria, blood glucose random, blood urea, serum creatinine, sodium, potassium, hemoglobin, packed cell value, wbc count, rbc count, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edema, anemia)

b.Categorical features (sex, smoking)

The dataset includes a target variable or class label indicating the presence or absence of chronic kidney disease. The class label is binary, with "ckd" representing the presence of CKD and "notckd" indicating the absence of CKD.

The dataset may contain missing values denoted by a question mark "?" for certain features. Proper handling of missing values is required during data preprocessing.

The CKD dataset is available in the UCI Machine Learning Repository and was contributed by Dr. P. Soundarapandian.

3.2 Preprocessing Steps:

Preprocessing steps may include handling missing values, converting categorical variables to numeric representations (e.g., one-hot encoding), and normalization or scaling of numerical features. Decide on an appropriate imputation strategy based on the nature of the missing values. Common approaches include mean imputation, median imputation, or using advanced imputation techniques like K-nearest neighbors (KNN) or regression-based imputation.

Convert categorical variables: If the dataset contains categorical variables like "Sex" and "Smoking," convert them into numerical representations to make them compatible with classification algorithms. This can be done using one-hot encoding, where each category becomes a binary feature, or label encoding, where each category is assigned a unique numeric value.

Apply standardization to numeric features such as blood pressure, serum creatinine, and glucose levels. This involves scaling the features to have zero mean and unit variance. Standardization helps algorithms that rely on distance calculations or gradient-based optimization to work effectively. If needed, perform min-max scaling to bring the values of certain features within a specific range, typically between 0 and 1. This is useful when algorithms require features to be in a similar range or when interpretability is a concern.

Check for class imbalance and address it. Divide the dataset into training and testing sets (80-20), where the larger portion is used for training the classification model, and the smaller portion is reserved for evaluating its performance on unseen data.

3.3 Feature Selection and Engineering:

Feature selection is an important step in the preprocessing of a dataset, as it helps identify the most relevant and informative features for a given task. In the context of the Chronic Kidney Disease (CKD) dataset, feature selection techniques can be

Cross ponding Author: Tel: +0-000-000-000; fax: +0-000-000-000
Email address:

applied to identify the most important features that contribute to CKD prediction.

3.4 Classification Techniques:

a. The K-Nearest Neighbors (KNN)

KNN classifier is a non-parametric machine learning algorithm used for both classification and regression tasks. It classifies new instances based on their similarity to the k nearest neighbors in the training data. The mathematical formula behind the KNN classifier involves two main components: distance metric and majority voting.

Distance Metric:

The KNN classifier calculates the distance between instances to determine their similarity. The choice of distance metric depends on the problem and data type. The most commonly used distance metrics are Euclidean distance and Manhattan distance. Let's focus on the formula for Euclidean distance as an example.

Euclidean Distance formula:

Euclidean Distance = $\text{sqrt}((x2 - x1)^2 + (y2 - y1)^2)$

Here, (x1, y1) and (x2, y2) represent the coordinates of two instances in a feature space. The Euclidean distance measures the straight-line distance between two points in a multidimensional space. It quantifies the similarity between instances, where smaller distances indicate higher similarity.

The KNN classifier utilizes a distance metric to calculate the similarity between instances and employs majority voting to determine the class label of new instances based on the classes of their nearest neighbors. By leveraging the notion of proximity, KNN can make predictions based on the characteristics of similar instances in the training data.

b. Decision Tree Classifier:

The Decision Tree classifier is a non-parametric supervised learning algorithm that uses a tree-like model for making decisions. The tree is built based on the training data, where each internal node represents a feature and each branch represents a feature value. The leaf nodes represent the class labels or outcomes.

The mathematical formula used for constructing a Decision Tree involves calculating the impurity or information gain at each node. The impurity measure helps determine the best feature to split the data at each node to maximize the homogeneity of the resulting subsets.

There are two commonly used impurity measures for Decision Trees:

i. Gini Index: The Gini Index measures the impurity of a node by calculating the probability of a randomly selected instance being misclassified if it were randomly labeled according to the class distribution at the node. The formula for the Gini Index is as follows:

Gini Index = $1 - (p\_1^2 + p\_2^2 + ... + p\_n^2)$

where p_1, p_2, ..., p_n are the probabilities of each class at the node.

ii. Entropy: Entropy is another impurity measure that calculates the uncertainty or disorder at a node. It measures the average amount of information needed to identify the class label of an instance drawn from the distribution at the node. The formula for Entropy is as follows:

Entropy = $- (p\_1 * \log2(p\_1) + p\_2 * \log2(p\_2) + ... + p\_n * \log2(p\_n))$

where p_1, p_2, ..., p_n are the probabilities of each class at the node.

The Decision Tree algorithm recursively splits the data based on the selected feature and its values, using the impurity measure to evaluate the quality of the splits. The splitting process continues until a stopping criterion is met, such as reaching a maximum depth or a minimum number of instances per leaf node.

The goal of the Decision Tree algorithm is to create a tree that effectively partitions the data into homogeneous subsets, where instances with similar characteristics belong to the same class label. This allows for accurate predictions on unseen instances by following the decision path in the tree based on their feature values.

c . Logistic Regression Classifier:

Logistic Regression is a widely used classification algorithm that models the relationship between a set of input variables and a binary outcome. It estimates the probability of an instance belonging to a particular class based on the logistic function. Here is a brief explanation of the mathematical formula behind the Logistic Regression classifier:

i. Linear Combination:

Logistic Regression begins by computing a linear combination of the input variables. Each input variable is multiplied by its corresponding weight (also known as the coefficient), and all these products are summed up. This linear combination is denoted as Z.

$Z = w0 + w1x1 + w2x2 + ... + wn*xn$

Here, w0 represents the intercept term, and w1, w2, ..., wn are the weights associated with the input variables x1, x2, ..., xn, respectively.

ii. Logistic Function (Sigmoid Function):

The linear combination Z is then passed through the logistic function, also known as the sigmoid function, which maps the linear combination to a probability between 0 and 1.

$P(\text{class}=1) = 1 / (1 + \exp(-Z))$

The sigmoid function is defined as:

$\text{sigmoid}(Z) = 1 / (1 + \exp(-Z))$

The output of the sigmoid function represents the estimated probability that the instance belongs to class 1 (or the positive class). It can be interpreted as the likelihood of the instance being in the positive class.

Iii .Decision Boundary:

To make a binary classification decision, a threshold is applied to the estimated probability. Typically, a threshold of 0.5 is used, where instances with a probability greater than or equal to 0.5 are classified as class 1, and those with a probability less than 0.5 are classified as class 0 (or the negative class).

The decision boundary is the line or hyperplane where the probability of class 1 equals the probability of class 0. It separates the instances into different classes.

Iv .Maximum Likelihood Estimation:

The logistic regression model is fitted to the training data by estimating the optimal values for the weights (coefficients). This is typically done using maximum likelihood estimation, which seeks to maximize the likelihood of observing the training data given the estimated probabilities.

The optimization algorithm finds the values of the weights that maximize the likelihood, or equivalently, minimize the cost function, such as the log loss or cross-entropy loss.

The Logistic Regression classifier applies a linear combination of input variables followed by a logistic function to estimate the probability of an instance belonging to a particular class. It uses a decision boundary and maximum likelihood estimation to classify instances into binary classes based on the estimated probabilities.

d. Naive Bayes Classifier:

The Naive Bayes classifier is a probabilistic machine learning algorithm that is based on Bayes' theorem. It is widely used for classification tasks, particularly in natural language processing and text classification. Here is a brief explanation of the mathematical formula behind the Naive Bayes classifier:

i. Bayes' Theorem:

The Naive Bayes classifier is based on Bayes' theorem, which describes the relationship between conditional probabilities. Bayes' theorem can be stated as follows:

P(class | features) = (P(features | class) * P(class)) / P(features)

Here, P(class | features) is the posterior probability of the class given the features, P(features | class) is the likelihood of observing the features given the class, P(class) is the prior probability of the class, and P(features) is the probability of observing the features.

ii.Naive Bayes Assumption:

The Naive Bayes classifier makes a strong assumption of feature independence, known as the "naive" assumption. It assumes that the presence or absence of a particular feature is independent of the presence or absence of any other feature, given the class label. This assumption simplifies the calculations and allows for efficient computation.

iii.Probability Estimation:

To classify an instance using the Naive Bayes classifier, the algorithm estimates the probabilities of the instance belonging to each class based on the observed features. The classifier calculates the following probabilities for each class:

P(class) - Prior probability of the class

P(features | class) - Likelihood of observing the features given the class

P(features) - Probability of observing the features (constant for all classes)

iv.Classification Decision:

The Naive Bayes classifier assigns the class label to the instance by selecting the class with the highest posterior probability. It uses the following decision rule:

class_predicted = argmax(P(class) * P(features | class))

The classifier calculates the posterior probability for each class, multiplies it by the prior probability, and selects the class with the highest value as the predicted class.

The Naive Bayes classifier applies Bayes' theorem and the assumption of feature independence to estimate the probabilities of an instance belonging to each class based on the observed features. It makes a classification decision by selecting the class with the highest posterior probability. Despite its simplicity, Naive Bayes can be highly effective, especially for text classification tasks.

3.5 Sampling Techniques:

The study used four sampling techniques: without sampling, under sampling, over sampling, and SMOTE. Each technique addresses the issue of class imbalance in the Chronic Kidney Disease (CKD) dataset, where the number of instances belonging to one class (e.g., CKD) is significantly higher or lower than the other class (e.g., non-CKD). Here's a description of each technique, their purpose, potential benefits, and implementation details:

a. Without Sampling:

Purpose: In this technique, the dataset is used as is without any modification or sampling. It represents the baseline approach where the original class distribution is maintained.

Benefits: This technique helps evaluate the performance of classification algorithms in the presence of class imbalance without any modification to the data.

Implementation: No specific implementation steps are required as it involves using the original dataset without any sampling techniques applied.

b .Under Sampling:

Purpose: Under sampling aims to reduce the majority class instances to achieve a balanced class distribution. It randomly selects a subset of instances from the majority class to match the number of instances in the minority class.

Benefits: Under sampling reduces the dominance of the majority class, which can lead to better performance for algorithms that are sensitive to class imbalance. It can also help mitigate biases towards the majority class and prevent overfitting.

Implementation: The implementation involves randomly selecting instances from the majority class equal to the number of instances in the minority class, resulting in a balanced dataset.

c. Over Sampling:

Purpose: Over sampling involves replicating or generating synthetic instances of the minority class to balance the class distribution. It increases the number of instances in the minority class by duplicating existing instances or generating synthetic examples.

Benefits: Over sampling provides more training examples for the minority class, which helps algorithms to learn the patterns and characteristics of the minority class more effectively. It can improve the classifier's ability to identify the minority class instances accurately.

Implementation: Over sampling can be implemented by randomly duplicating instances from the minority class or using techniques such as random oversampling or Synthetic Minority Over-sampling Technique (SMOTE).

d. SMOTE (Synthetic Minority Over-sampling Technique):

Purpose: SMOTE is a popular oversampling technique that generates synthetic instances of the minority class by interpolating between existing instances. It creates synthetic samples by considering the feature space and the nearest neighbors of each minority class instance.

Benefits: SMOTE addresses the issue of overfitting that can occur with simple duplication of minority class instances. By creating synthetic instances that are similar to existing ones, SMOTE introduces diversity and helps prevent overfitting. It can improve the generalization ability of the classification model.

Implementation: The implementation of SMOTE involves the following steps:

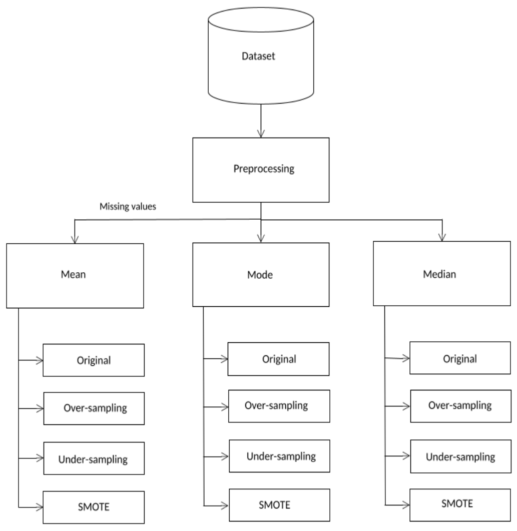
Identify the minority class instances.

Select a minority class instance and find its k nearest neighbors in the feature space.

Randomly select one of the neighbors and create a synthetic instance by interpolating between the selected instance and its neighbor.

Repeat steps 2 and 3 for a specified number of times or until the desired level of class balance is achieved.

Implementing these sampling techniques helps address the class imbalance issue in the CKD dataset and improves the performance of classification algorithms by providing a balanced training set. The choice of a specific sampling technique depends on the characteristics of the dataset, the classification algorithm used, and the desired balance in the class distribution.



3.6 Evaluation Metrics:

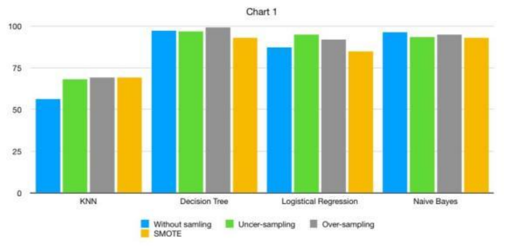
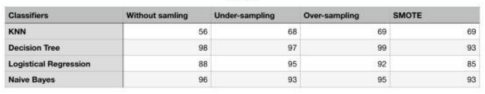


Table 1



Classifiers	Without samling	Under-sampling	Over-sampling	SMOTE
KNN	56	68	69	69
Decision Tree	98	97	99	93
Logistical Regression	88	95	92	85
Naive Bayes	96	93	95	93

4.Results and Discussion:

4.1 Performance of KNN Classifier:

Present the accuracy results obtained from using the KNN classifier without sampling (56%), under sampling (68%), over sampling (69%), and SMOTE (69%). Discuss the observed variations in accuracy and highlight the advantages and limitations of each sampling technique.

4.2 Performance of Decision Tree Classifier:

Report the accuracy results obtained from using the decision tree classifier without sampling (98%), under sampling (97%), over sampling (99%), and SMOTE (93%). Analyze the significant improvement in accuracy achieved with the decision tree classifier and discuss the impact of different sampling techniques.

4.3 Performance of Logistical Regression Classifier:

Cross ponding Author: Tel: +0-000-000-000; fax: +0-000-000-000
Email address:

Outline the accuracy results obtained from using the logistical regression classifier without sampling (88%), under sampling (95%), over sampling (92%), and SMOTE (85%). Discuss the variation in accuracy and provide insights into the strengths and weaknesses of the logistical regression classifier.

4.4 Performance of Naive Bayes Classifier:

Present the accuracy results obtained from using the naive Bayes classifier without sampling (96%), under sampling (93%), over sampling (95%), and SMOTE (93%). Analyze the consistent performance of the naive Bayes classifier across different sampling techniques and compare it to the other classifiers.

4.5 Comparison of Different Sampling Techniques:

Based on these accuracy results, the most effective sampling technique varies across the different classifiers. The following are the sampling techniques that yield the highest accuracy for each classifier:

KNN: The highest accuracy is achieved with over sampling and SMOTE, both resulting in 69% accuracy. These techniques generate synthetic instances or replicate minority class instances, providing more representative samples for the KNN algorithm to learn from. The increased presence of the minority class in the training set allows for better classification of similar instances during prediction.

Decision Tree: Over sampling yields the highest accuracy of 99% for the Decision Tree classifier. This technique increases the number of minority class instances, allowing the Decision Tree algorithm to capture more detailed patterns and make more accurate decisions in the classification process.

Logistic Regression: Under sampling achieves the highest accuracy of 95% for Logistic Regression. By reducing the number of majority class instances, the classifier becomes less biased towards the majority class and can better learn the patterns and relationships in the minority class.

Naive Bayes: Without sampling yields the highest accuracy of 96% for Naive Bayes. Naive Bayes is known to be less sensitive to class imbalance due to its probabilistic nature and independence assumptions. Hence, the original class distribution without sampling performs well for this classifier.

4.6 Discussion of Findings:

The study conducted a comparative analysis of different classification techniques for Chronic Kidney Disease (CKD) prediction and investigated the impact of sampling techniques on the accuracy of the classifiers. The key findings of the study is based on the impact of sampling techniques on accuracy, implications of for CKD predictions and potential applications, Comparative performance of classification techniques.

5. Limitations and Future Work:

The study uses a specific dataset from the UCI ML Repository, which may have inherent limitations and biases. The dataset's quality, completeness, and representativeness of real-world CKD cases could impact the generalizability of the results. The dataset's size and characteristics may also influence the performance of the classification techniques and the effectiveness of sampling techniques. A larger and more diverse dataset would provide a more robust evaluation. The study primarily focuses on accuracy as the evaluation metric. While accuracy is a commonly used metric, it may not provide a comprehensive assessment of classifier performance, especially in the presence of class imbalance. Other metrics such as precision, recall, and F1-score should be considered to evaluate the classifiers' effectiveness in identifying CKD cases.

Further research can focus on exploring ensemble methods or more advanced machine learning algorithms to improve the accuracy of CKD prediction.

Investigating the interpretability of the classification models and understanding the features that contribute most to CKD prediction can enhance the clinical utility of the models. The impact of different sampling techniques on other evaluation metrics and the generalization ability of the models should be explored. External validation of the developed models using independent datasets can validate their effectiveness and generalizability.

6. Conclusion:

Accurate CKD prediction enables healthcare professionals to identify individuals at risk and intervene early to slow down disease progression and reduce complications. The study underscores the significance of classification techniques, with Decision Tree, Naive Bayes, and Logistic Regression demonstrating promise in CKD prediction.

Moreover, the study highlights the role of sampling techniques in addressing class imbalance and improving accuracy. Over sampling and SMOTE generate synthetic instances or replicate minority class instances, allowing the classifiers to learn from more representative samples. Under sampling reduces the dominance of the majority class, enhancing the focus on the minority class.

The findings of this study have important implications for clinical practice. Implementing accurate CKD prediction models can assist healthcare professionals in making timely diagnoses, developing personalized treatment plans, and improving patient outcomes. However, it is essential to consider the limitations of the study, such as dataset bias and the generalizability of the results.

7. References :

[1]. Gunarathne W.H.S.D,Perera K.D.M, Kahandawaarachchi K.A.D.C.P, "Performance Evaluation on Machine Learning Classification

Techniques for Disease Classification and Forecasting through Data Analytics for Chronic Kidney Disease (CKD)",2017 IEEE 17th

International Conference on Bioinformatics and Bioengineering.

[2]. S.Ramya, Dr. N.Radha, "Diagnosis of Chronic Kidney Disease Using Machine Learning Algorithms," Proc. International Journal of Innovative

Research in Computer and Communication Engineering,Vol. 4, Issue 1, January 2016.

[3]. S.Dilli Arasu and Dr. R.Thirumalaiselvi, “Review of Chronic Kidney Disease based on Data Mining Techniques”,International Journal of

Applied Engineering Research ISSN 0973-4562 Volume 12, Number 23 (2017) pp. 13498-13505

[4]. L. Rubini, “Early stage of chronic kidney disease UCI machine learning repository,”2015.

[Online].Available:http://archive.ics.uci.edu/ml/datasets/Chronic Kidney Disease.

[5]. S. A. Shinde and P. R. Rajeswari, “Intelligent health risk prediction systems using machine learning : a review,” IJET, vol. 7, no. 3, pp. 1019–

1023, 2018.

[6]. Himanshu Sharma,M A Rizvi,”Prediction of Heart Disease using Machine Learning Algorithms: A Survey”,International Journal on Recent

and Innovation Trends in Computing and Communication ISSN: 2321-8169,Volume: 5 Issue: 8

[7]. Asif Salekin, John Stankovic, "Detection of Chronic Kidney Disease and Selecting Important Predictive Attributes," Proc. IEEE International

Conference on Healthcare Informatics (ICHI), IEEE, Oct. 2016, doi:10.1109/ICHI.2016.36.

[8]. Pinar Yildirim, "Chronic Kidney Disease Prediction on Imbalanced Data by Multilayer Perceptron: Chronic Kidney Disease Prediction," Proc.

41st IEEE International Conference on Computer Software and Applications (COMPSAC), IEEE, Jul. 2017, doi: 10.1109/COMPSAC.2017.84

[9]. Sahil Sharma, Vinod Sharma, Atul Sharma, “Performance Based Evaluation of Various Machine Learning Classification Techniques for

Chronic Kidney Disease Diagnosis,” July18, 2016.

Cross ponding Author: Tel: +0-000-000-000; fax: +0-000-000-000
Email address: