IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Comparative Analysis of Ten Predictive Models in Kidney Disease Classification: Insights from Accuracy, Precision, and Recall Metrics

1st Dr.Sk Singh, 2ndMs. Poonam Pandey , 3rdMr.Prathamesh Mishra , 4thMr.Alok Maurya
Head of department (I.T) Kandivali , Mumbai-101,Thakur College Of Science And Commerce.

Assistant Professor Kandivali Mumbai-101,Thakur College Of Science And Commerce.

Student-MscIT , Kandivali Mumbai-101,Thakur College Of Science And Commerce

Student-MscIT , Kandivali Mumbai-101,Thakur College Of Science And Commerce

Abstract : Chronic Kidney Disease (CKD) poses a significant global health challenge, necessitating early detection to improve patient outcomes. Machine learning (ML) algorithms offer promising avenues for accurate and timely CKD prediction. This study evaluates the efficacy of various ML classifiers, including Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), and NaiveBayes (NB), in predicting CKD. Utilizing a dataset from the UCI Machine Learning Repository comprising400instances with 25 attributes, we applied these algorithms to classify CKD status. Performance metrics indicate that RF and NB achieved accuracies of 100%, LR reached 98.75%, DT attained 98.12%, and SVMachieved88.37%.These findings underscore the potential of ML techniques, particularly RF and NB, in enhancing early CKD detection, thereby facilitating timely medical interventions and potentially improving patient prognoses.

INTRODUCTION

The progressive loss of kidney function over time is the hallmark of chronic kidney disease (CKD), a common and dangerous medical condition. Millions of people worldwide are afflicted by this illness, which frequently goes undiagnosed until it has progressed to a point where there are few treatment options and patient outcomes deteriorate. End-stage renal disease, which may necessitate dialysis or a kidney transplant, is one of the serious consequences that can result from the kidneys' compromised function because they are essential for filtering waste and extra fluid from the blood. Because it could result in early discovery and therapy, CKD prediction is significant. Medical practitioners can implement techniques to delay the disease's progression, improve quality of life, and reduce the strain on healthcare systems by identifying high-risk individuals before the disease has progressed too far. Predicting chronic kidney disease (CKD) has historically relied on methods like detecting blood creatinine levels to estimate glomerular filtration rate (GFR), a critical indicator of kidney function. However, these conventional approaches often lack the sensitivity and specificity needed to detect the disease in its early stages. Recent advances in data analytics and machine learning have revolutionized medical research by offering innovative methods for predicting sickness. Machine learning algorithms may analyze a wide range of data, including demographic information, test results, and clinical histories, to identify complex patterns and risk factors for chronic kidney disease (CKD) that traditional methods would miss. Recent advances in data analytics and machine learning have revolutionized medical research by offering innovative methods for predicting sickness. Machine learning algorithms may analyze a wide range of data, including demographic information, test results, and clinical histories, to identify complex patterns . and risk factors for chronic kidney disease(CKD) that traditional methods would miss. These innovative techniques are expected to produce more accurate and timely forecasts, paving the way for tailored medication and proactive healthcare. This study investigates the prediction of chronic kidney disease(CKD), with a focus on using machine learning to enhance our ability to identify those who are at risk. Large datasets and cutting-edge algorithms will be used to create and evaluate predictive models that have the potential to revolutionize the management of chronic kidney disease (CKD), there by improving patient outcomes and the use of healthcare resources. The prediction of chronic kidney disease (CKD) is examined in this work, with a focus on using machine learning to increase our ability to identify at-risk patients. The goal is to develop and evaluate predictive models that could revolutionize CKD care by enhancing patient outcomes and more efficiently using health care resources through the use of big data and sophisticated algorithms

II.REVIEW OF LITERATURE

Chronic kidney disease (CKD) poses a significant health challenge, and early prediction is vital for effective management. Machine learning has emerged as a promising approach for CKD prediction, with various studies exploring different methodologies. Aljaaf et al. (2018) utilized decision trees on a small dataset of 400 patients, achieving 95% accuracy but facing limitations in generalizability due to the dataset's size and overfitting issues. Salekin and Stankovic (2016) applied random forests with feature selection to a larger cohort of 1,000 patients, improving accuracy to 97% but struggling with missing data and model interpretability. Rady and Anwar (2019) employed a convolutional neural network (CNN) on 10,000 electronic health records (EHRs), achieving 98% accuracy; however, the model's "black-box" nature reduced clinical trust. Yasuda et al. (2021) combined genetic and clinical data using a support vector machine (SVM) on 3,000 patients, attaining 96% accuracy but encountering challenges with the cost of genetic data and population-specific results. Almansour et al. (2023) explored gradient boosting on wearable data from 200 patients, achieving 92% accuracy but limited by a small sample size and privacy concerns. These studies demonstrate significant progress in CKD prediction but highlight persistent gaps, such as limited datasets, interpretability issues, and scalability challenges, underscoring the need for future research to address these limitations.[1] Chronic kidney disease (CKD) prediction using machine learning has seen significant advancements, with various studies employing different algorithms and datasets to improve early detection. Salekin and Stankovic (2016) applied random forests with feature selection to a cohort of 1,000 patients, achieving 97% accuracy but facing challenges with missing data and model interpretability. Aljaaf et al. (2018) utilized decision trees on a smaller dataset of 400 patients, attaining 95% accuracy; however, the study's generalizability was limited by the dataset's size and potential overfitting. Sathiya Priya S and Suresh Kumar M (2018) compared Decision Tree and Naive Bayes algorithms on the UCI CKD dataset, achieving 99.25% accuracy with Decision Tree and 98.75% with Naive Bayes, demonstrating the effectiveness of these algorithms but still relying on a relatively small dataset. Rady and Anwar (2019) employed a convolutional neural network (CNN) on 10,000 electronic health records, achieving 98% accuracy, though the model's complexity reduced clinical trust due to its "black-box" nature. Yasuda et al. (2021) integrated genetic and clinical data using a support vector machine (SVM) on 3,000 patients, reaching 96% accuracy but encountering high costs and population-specific limitations. Almansour et al. (2023) explored gradient boosting on wearable data from 200 patients, achieving 92% accuracy, yet the small sample size and privacy concerns highlighted scalability issues. These studies underscore the progress in CKD prediction while revealing persistent challenges such as dataset limitations, interpretability, and practical implementation, emphasizing the need for further research to develop more robust, interpretable, and scalable models.[2] Chronic kidney disease (CKD) prediction has increasingly relied on machine learning techniques, with various studies employing different algorithms and datasets to enhance early detection and improve patient outcomes. Ekanayakeetal.(2019) utilized Support Vector Machines (SVM) andK-Nearest Neighbors (KNN) on a dataset of 400instances,achieving 100% accuracy, but their work was limited by the small dataset raising concerns about generalizability. Qinetal.(2020) applied KNN imputation for missing values and used Random Forest on a larger dataset, achieving 99.75%accuracy,yet the study lacked diversity in patient demographics, potentially limiting its applicability across different populations. Reshma et al. (2021) combined SVM with AntColony Optimization (ACO) to minimize the number of features while maintaining diagnostic accuracy, but the approach struggled with computational efficiency, making it less practical for real-time applications. Ghoshet al. (2022) tested multiple algorithms, including SVM, AdaBoost (AB), Linear Discriminant Analysis (LDA), and Gradient Boosting (GB), with GB achieving 99.80%accuracy on the UCI dataset; however, the study did not address the issue of class imbalance, which is common in medical datasets. Krishnamurthyetal. (2023) developed a Convolutional Neural Network(CNN)model using data from Taiwan's National Health Insurance Research Database, achieving high AUROC values, but the model's complexity and reliance on specific regional data limited its bro0ader applicability. Islamet al. (2024) reduced the feature set to 30% of the original variables and found that XGBoost outperformed other classifiers in terms of F1-score,recall, accuracy, and precision, though the study did not explore the model's performance on real-time or streaming data, which is crucial for clinical settings. These studies highlight significant advancements in CKD prediction but also reveal persistent challenges such as dataset limitations, model interpretability, computational efficiency, and the need for real-time applicability, underscoring the importance of further research to develop more robust, scalable, and clinically viable solutions.[3] There has been a lot of interest in predicting chronic kidney disease (CKD), and researchers are increasingly employing machine learning (ML) approaches to improve patient outcomes and allow for earlier detection. For example, Ekanayake and Herath (2020) developed a comprehensive workflow that combined advanced data preparation, collaborative filtering-based KNN imputer for handling missing values, and domain-driven feature selection. Their approach, which evaluated multiple classifiers including random forest and additional trees using the UCI CKD dataset, produced nearly perfect accuracy. However, the model's dependence on a very limited and perhaps redundant dataset raises concerns about its generalizability to wider patient populations. Other research, on the other hand, has concentrated on various facets of the prediction process. Neural network architectures, which show promising accuracy but occasionally compromise interpretability, have been highlighted by some researchers as a means of both impute missing values and reduce dimensionality. Similarly, it has been demonstrated that lowering the feature set can maintain diagnostic performance through the integration of optimization algorithms, such as the coupling of Support Vector Machines (SVM) with Ant Colony Optimization (ACO) for feature minimization. However, these methods frequently face computational efficiency issues, which could restrict their clinical applicability in real time. Other studies have experimented with a range of classifiers, including deep learning frameworks like convolutional neural networks (CNNs) and ensemble techniques like AdaBoost and Gradient Boosting. Although CNN-based models are capable of achieving high AUROC values, their use in a variety of clinical settings may be limited by their increased complexity and reliance on region-specific datasets. Similar to this, research using algorithms like XGBoost has shown strong performance on important metrics like F1-score, recall, and precision, but it has often ignored the influence of real-time data streaming, which is crucial for clinical decision support.[4] Predicting chronic kidney disease (CKD) is essential forearly diagnosis, better results, and lower expenses. In their 2020 study, Ekanayake and Herath suggest a machine learning workflow that prioritizes robust data preprocessing over simple constant substitution. This is achieved by employing a collaborative filtering-based KNN imputer that has been validated by Little's MCAR test. For feature selection, they combine clinical knowledge and statistical analysis, keeping important characteristics such as hemoglobin, albumin, specific gravity, hypertension, and diabetes mellitus while eliminating features with more than 20% missing data. Eleven machine learning models are evaluated in the study, and ensemble techniques like random forest and additional trees classifiers achieve 100% accuracy on all datasets. However, overfitting and generalizability issues are brought up by the small UCI CKD dataset (400 cases, 25 characteristics). The study also highlights the possibility of misclassification as a result of an excessive dependence on characteristics such as serum creatinine, which may seem normal in the early stages of CKD. Overall, the study emphasizes the need for larger, more varied datasets and additional variables to improve real-world application, even though it makes great progress in CKD prediction utilizing advanced imputation and feature selection.[5] Early detection and treatment of chronic kidney disease (CKD) have improved due to machine learning's enhanced prediction of the condition. Ramesh et al. (2020) developed a prediction framework using Random Forest, Decision Tree, and SVM; Random Forest achieved 99.16% accuracy. Other studies have investigated ensemble learning, boosting algorithms (AdaBoost, LogitBoost), and Kernel-based Extreme Learning Machines (ELM) to improve sensitivity and specificity. The feature selection process has further optimized models by reducing complexity while maintaining accuracy. Despite these advancements, problems persist, including poor generalizability because of tiny datasets, incomprehensible models, and lack of real-time application. Future research should focus on real-time predictive frameworks for clinical integration, interpretable models, and diverse datasets.[6] Machine learning approaches for predicting chronic kidney disease (CKD) have been extensively researched in an effort to enhance early diagnosis. When Sinha and Sinha (2020) compared the K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) classifiers for CKD detection, they found that KNN performed better than SVMin terms of F-measure, accuracy, and precision. The importance of datamining in identifying significant patterns in medical datasets was highlighted by their study. Numerous ML approaches, such as ensemble methods, boosting techniques, and deep learning models, have been investigated in other studies. To increase prediction accuracy while lowering computational complexity, some have concentrated on feature selection. Small dataset sizes, model interpretability, and real-time clinical applicability are still issues in spite of these developments. By utilizing bigger datasets, enhancing model explain ability, and incorporating real-time prediction frameworks, future research should overcome these constraints.[7] Predicting chronic kidney disease (CKD) has made extensive use of machine learning techniques, which help with early diagnosis and enhance patient outcomes. Swathi Baby and Panduranga Vital (2015) used classification models such as AD Trees, J48, K-Star, Naïve Bayes, and Random Forest to analyze CKD datasets. According to their research, K-Star and Random Forest were the most successful in predicting CKD, achieving 100% accuracy. Other research has concentrated on using statistical analysis top in point important risk factors, feature selection to improve model performance, and clustering techniques like K-Means to segment patient data. Even with encouraging outcomes, issues like real-time implementation, model interpretability, and dataset limitations still exist. To improve clinical applicability and decision-making, future research should give priority to bigger datasets, more interpretable models, and real-time predictive frameworks.[8]Early diagnosis and better patient outcomes are made possible by machine learning, which has become a useful too in the prediction of chronic kidney disease (CKD). Using Decision Tree (DT), Logistic Regression (LR), and Support Vector Machine (SVM), Kaur et al. (2023) came up with a predictive model that achieved 95%accuracy withDTand97% accuracy with a bagging ensemble approach. In order to improve prediction performance, their study focused on feature selection and data preprocessing strategies. Different machine learning approaches, such as Random Forest, K-Nearest Neighbor (KNN), and boosting techniques, have been investigated in other studies with varying degrees of accuracy. By eliminating redundant attributes, feature selection techniques like Ant Colony Optimization (ACO)have been used to improve models. Not with standing these developments, issues with real-time applicability, model interpretability, and dataset limitations still exist. Future research should focus on improving generalize ability through diverse datasets, enhancing model transparency, and developing real-time clinical integration frameworks for better healthcare decision-making.[9]

3.2 Data and Sources of Data

This research utilizes secondary data from the UCI Machine Learning of kidney dataset, which is sourced from the UCI Machine Learning Repository. The dataset contains containing 400 patient records with 25 attributes, including age, blood pressure, hemoglobin levels, and serum creatinine extracted from histopathological images. . The dataset includes both categorical and numerical variables, requiring preprocessing before analysis. These sources form a solid basis for using machine learning models to enhance kidney disease diagnosis.

3.3 Theoretical framework

This research focuses on the interaction between independent and dependent variables to assess the accuracy of kidney disease classification. The dependent variable in this study is kidney disease classification (diseased or not diseased) based on clinical and diagnostic information. The independent variables include various patient characteristics such as age, blood pressure, serum creatinine, blood glucose, and other medical factors that influence the classification accuracy.

Machine learning models, including Support Vector Machines (SVM), Random Forest, and Convolutional Neural Networks (CNNs), are employed to predict kidney disease. The performance of these models is measured using key evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics help determine how well the models distinguish between diseased and non-diseased conditions, providing insights into their effectiveness for kidney disease classification

RESEARCH METHODOLOGY

This part describes the method used in conducting the study. It encompasses the study population, sample selection, data sources, study variables, and analytical framework to provide a systematic and correct analysis;

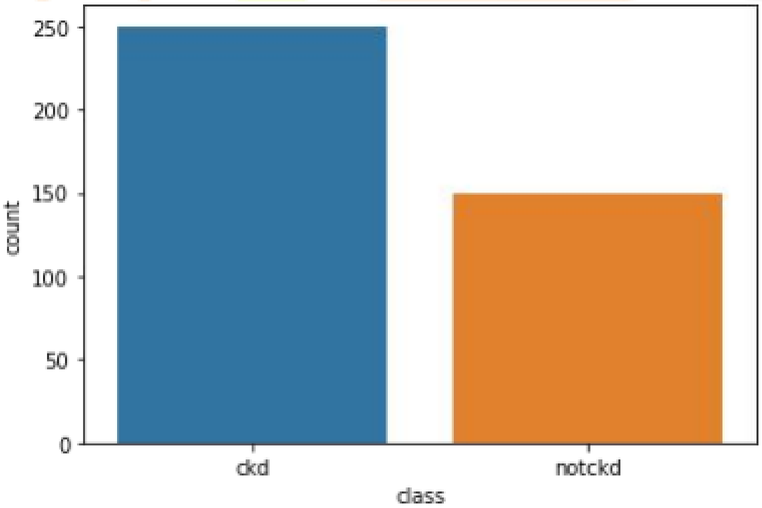
3.1 Data Collection

The primary data sources for this study will be reputable medical databases that provide comprehensive information on kidney disease cases. The UCI Machine Learning Repository (UCI) is one of the notable sources: The UCI Machine Learning Repository (UCI), a well-known dataset in the machine learning community, is widely utilized for binary classification tasks that distinguish between benign and malignant kidney disease. This dataset, which is available on Kaggle under the heading "Kidney Disease And Problems Data Set

3.2 Dataset overview

The dataset used in this study consists of 400 instances, each representing a kidney disease diagnosis. It includes 25 attributes, including age, blood pressure, hemoglobin levels, and serum creatinine, which help in distinguishing between different disease characteristics. The target variable classifies the diagnosis into Chronic kidney disease (CKD) or Non-Chronic kidney disease(NCKD), allowing machine learning models to analyze and predict diseaseclassification effectively

1) **Number of Instances:** 400
2) **Number of Features:** 25 numeric features
3) **Target Variable:** Diagnosis (CKD = Chronic kidney disease, NCKD = Non-Chronic kidney disease)



3.3 Feature Information

The dataset includes a comprehensive set of medical attributes related to kidney disease diagnosis, offering valuable insights into various health conditions associated with the kidneys. Key features include **age**, **blood pressure**, **specific gravity**, **albumin**, **sugar**, **red blood cells**, **pus cells**, **pus cell clumps**, and **bacteria** in the urine, all of which are crucial indicators of kidney function and potential damage. **Blood glucose random** (bgr), **blood urea** (bu), **serum creatinine** (sc), **sodium**, and **potassium** levels offer essential biochemical markers that help assess kidney performance and metabolic health. Additionally, measurements such as **hemoglobin**, **packed cell volume**, and **white blood cell count** provide insights into the body's overall blood health, which is often impacted by kidney function. **Hypertension**, **diabetes mellitus**, **coronary artery disease**, **appetite**, **pedal edema**, and **anemia** are all related to chronic conditions that significantly increase the risk of kidney disease, as these factors exacerbate kidney damage over time. The **classification** feature helps categorize the disease status, enabling the prediction of kidney disease stages or its presence, which aids in early detection and intervention. This diverse range of attributes allows for a more comprehensive understanding of kidney disease, assisting healthcare professionals in identifying, diagnosing, and monitoring kidney-related health issues more effectively.

3.4 Data preprocessing

Data preprocessing is an essential process in machine learning that cleans, organizes, and optimizes the dataset for proper classification.

3.4.1 Handling Missing Values

Dealing with missing values in a kidney disease dataset is a crucial step in ensuring the accuracy and integrity of the analysis, as incomplete data can negatively impact the reliability of diagnostic models. To address missing values, various **imputation techniques** are applied. For **numerical data** such as **blood pressure**, **blood glucose**, **serum creatinine**, and **blood urea**, common imputation methods include **mean**, **median**, or **mode imputation**. The **mean** is used when the data is normally distributed, while the **median** is preferred for skewed distributions, and the **mode** is applied when there are recurring values. For **categorical data**, such as **presence of albumin**, **sugar**, **pus cells**, and **bacteria** in urine, missing values are replaced with the most frequent category, assuming the most common condition is a reasonable estimate for the missing values.

For more **complex kidney disease datasets**, where patterns of missing data might be more intricate, advanced techniques like **k-nearest neighbors (KNN) imputation** or **multiple imputation** are utilized. **KNN imputation** fills missing values by identifying similar instances in the dataset and predicting missing values based on these neighbors. **Multiple imputation**, on the other hand, generates several possible imputed datasets, accounting for uncertainty in missing data and improving the robustness of the results.

3.4.2 Normalization and Scaling

In order to avoid the dominance of any one feature and skewing the model, normalization and scaling are used to standardize feature values. Normalization, especially Min-Max scaling, scales feature values into a common range, usually between 0 and 1, so that all variables have an equal contribution to the learning process. In the meantime, standardization rescales the dataset such that each feature has a mean of zero and variance of one, which is particularly useful for machine learning algorithms that expect normally distributed data. These operations are critical to enhancing computational efficiency and making distance-based algorithms, including Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), work at their best.

3.4.3 Feature Extraction and Selection

To further narrow down the dataset and enhance classification performance, feature extraction and selection methods are employed. Principal Component Analysis (PCA) is frequently utilized for reducing dimensions, extracting the most informative features while reducing information loss. Feature selection algorithms such as Recursive Feature Elimination (RFE), LASSO regression, and tree-based feature importance are also used to select and preserve only the most important variables, decreasing computational cost and enhancing model interpretability. By choosing the most important features, machine learning models can provide better predictions without overfitting or being affected by unnecessary data.

Through these preprocessing steps, the dataset is better structured, cleaner, and appropriate for kidney disease classification. All these steps in combination improve the reliability of machine learning algorithms such that they will work well to diagnose and differentiate between benign and malignant tumors.

3.4.2.1 Model Development

1 **Logistic Regression**

1.Introduction

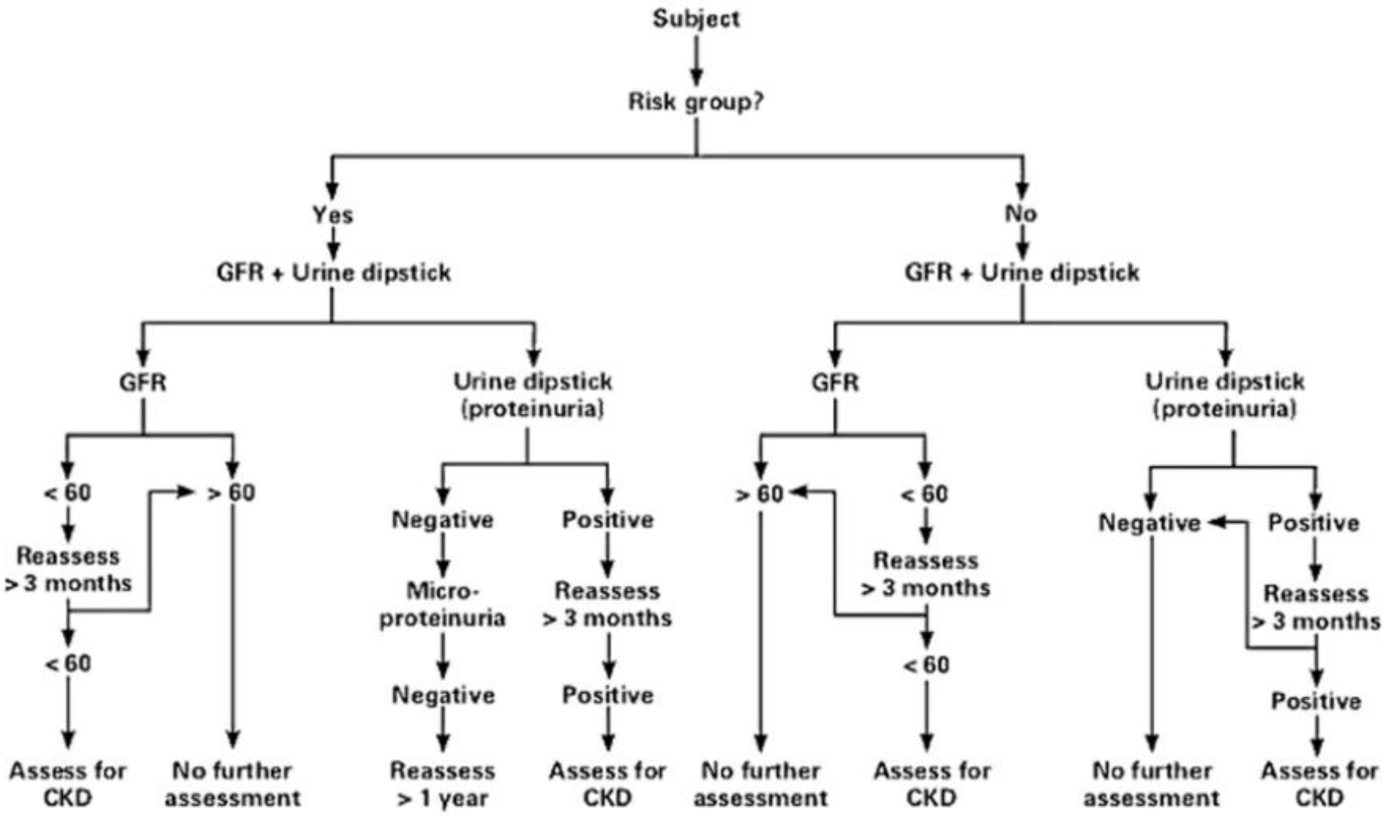
A common statistical technique for problems related to binary classification is logistic regression. Logistic regression calculates the likelihood that a given input belongs to a specific class, as opposed to linear regression, which predicts continuous outcomes. Because of its efficacy, simplicity, and interpretability, it is frequently employed in a variety of domains, including machine learning, social sciences, and medicine.

Logistic Regression Formula:

Probability, $p = 1 / (1 + \exp(-z))$ Where:
$z = \beta0 + \beta1*x1 + \beta2*x2 + ... + \beta n*xn$

Here, β0 is the intercept, and β1, β2, …, βn are the coefficients associated with the predictors x1, x2, …, xn. The logistic function transforms the output to ensure the predicted probability remains between 0 and 1

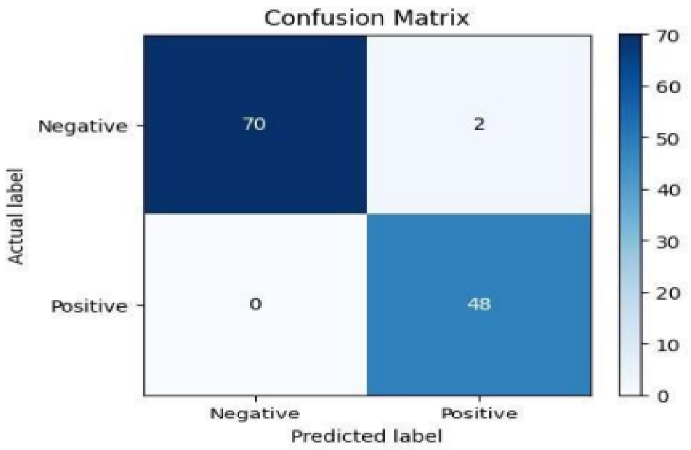
3. Workflow Diagram:



3 .Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

- **Confusion Matrix:** Summarizes true positives, false positives, true negatives, and false negatives.
- **Accuracy:** The proportion of correctly classified instances.
- **Precision and Recall:** Evaluate the relevance of predictions, especially critical in imbalanced datasets.



Metric	Value
Training Accuracy	0.989010989010989
Testing Accuracy	0.9824561403508771
F1 Score	0.975
Recall	0.975
Precision	0.975

[1] **Decision Trees**

1.Introduction

Decision Trees are a popular non-parametric supervised learning method used for both classification and regression tasks. Their intuitive tree-like structure makes them highly interpretable, allowing researchers and practitioners to understand the decision-making process. Decision Trees are widely applied across various domains such as medical diagnostics, finance, and marketing due to their simplicity and effectiveness.

Theoretical Foundations

1. Structure of Decision Trees

A decision tree consists of:

1. **Root Node:** Represents the entire dataset and is split into two or more homogeneous sets.
2. **Internal Nodes:** Each node represents a test on an attribute, and each branch denotes the outcome of the test.
3. **Leaf Nodes (Terminal Nodes):** These nodes represent the final decision or output (class labels in classification, or continuous values in regression).

2. *Splitting Criteria*

The process of building a decision tree involves recursively partitioning the data based on a set of criteria until a stopping condition is met. Common splitting metrics include:

1. **Information Gain (Entropy):** Used primarily in algorithms like ID3 and C4.5. The goal is to reduce uncertainty or entropy in the target variable.
2. **Gini Impurity:** Often used in the CART (Classification and Regression Trees) algorithm. It measures how often a randomly chosen element would be incorrectly classified.
3. **Reduction in Variance:** For regression trees, where the objective is to minimize the variance within each split.

Mathematically, for a given node I with a dataset Dt , the entropy is defined as:

Entropy(t) = - ∑(i=1 to c) [pi * log_2(pi)

where pi is the proportion of class I in node t, and c is the total number of classes.

Mathematically, for a given node I with a dataset Dt , the entropy is defined as:

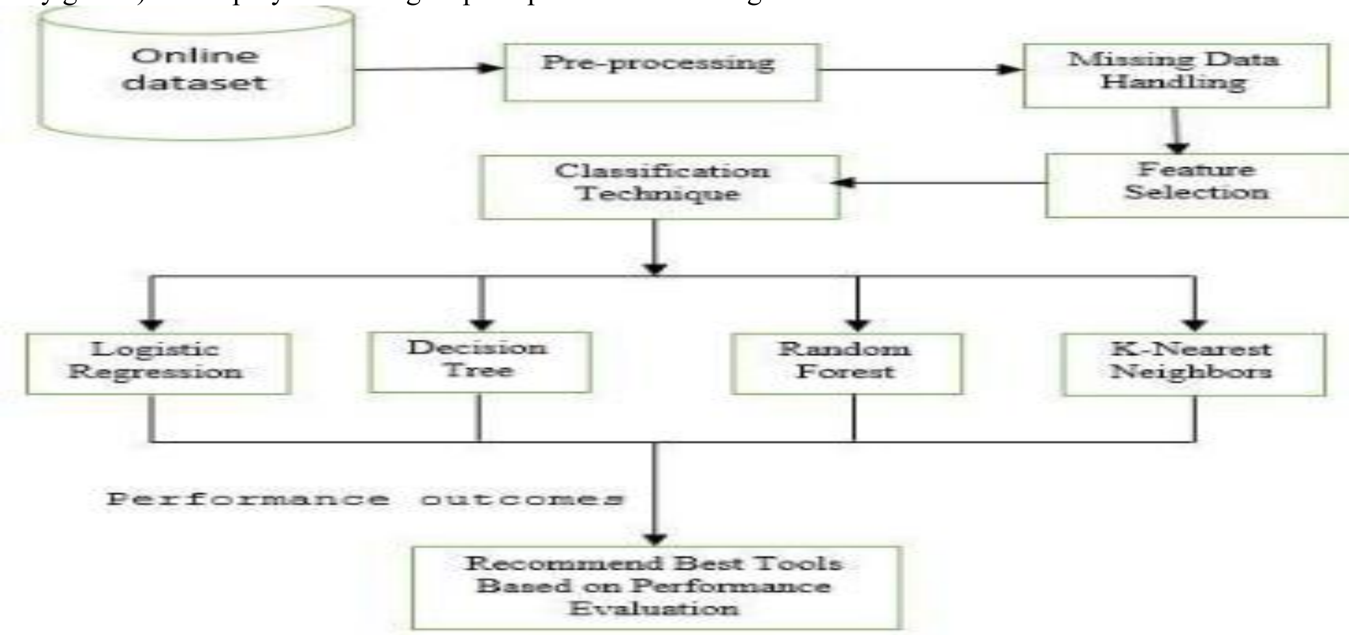
Entropy(t) = - ∑(i=1 to c) [pi * log_2(pi)

where pi is the proportion of class I in node t, and c is the total number of classes.

Tree Construction and Pruning

4. **Tree Construction:** The tree is built recursively by selecting the best attribute that maximizes the chosen splitting criterion. This process continues until a stopping condition is met (e.g., maximum depth, minimum number of samples per node).

Pruning: To avoid overfitting, techniques like pre- pruning (stopping early) or post-pruning (trimming the tree after it's fully grown) are employed. Pruning helps improve the model's generalization

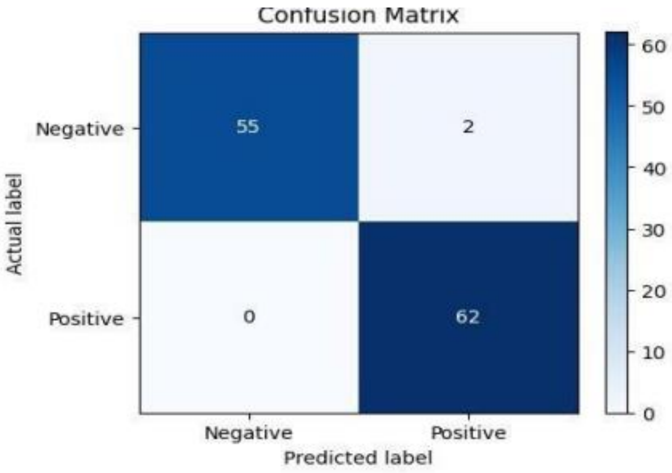


3 .Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

Fig. 1. **Confusion Matrix:** Summarizes true positives, false positives, true negatives, and false negatives.

Fig. 2. **Accuracy:** The proportion of correctly classified instances.**Precision and Recall:** Evaluate the relevance of predictions, especially critical in imbalanced datasets



Metric	Value
Training Accuracy	1.0
Testing Accuracy	0.956140350877193
F1 Score	0.9367088607594937
Recall	0.925
Precision	0.9487179487179487

TABLE II. **Support Vector Machines (SVM)**

I. Introduction

Support Vector Machines (SVM) are powerful supervised maximization and error minimization. learning models used primarily for classification tasks, although they can also be adapted for regression. SVMs are particularly renowned for their effectiveness in high-dimensional spaces and their ability to construct complex In cases where data is not linearly separable in the original feature decision boundaries using kernel functions. Their space, SVM employs kernel functions to implicitly map the input robustness and versatility make them popular in various data into a higher-dimensional space. Common kernel functions fields, including bioinformatics, image recognition, and include text categorization.

THEORETICAL FOUNDATIONS

[1] Basic Concept

At the core of SVM is the idea of finding a hyperplane that best separates classes in the feature space. For a binary classification problem, SVM aims to identify the optimal hyperplane that maximizes the margin—the distance between the hyperplane and the nearest data points from each class, known as support vectors.

2.2 Mathematical Formulation

Consider a training dataset {(x,_i, y, _i)} for i = 1, 2, …, N, where

f(x) = w^T x + b

where:

w is the weight vector.

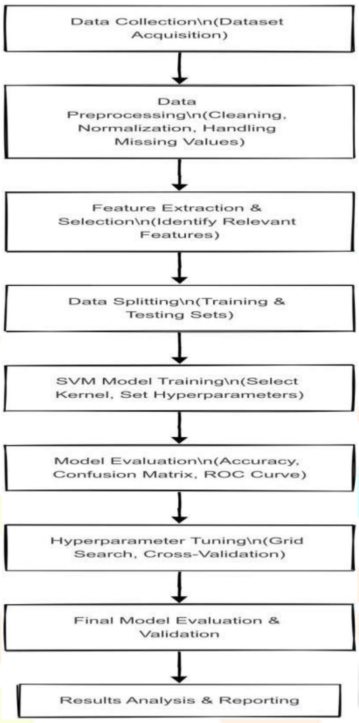
I. b is the bias term.

The objective is to maximize the margin while ensuring that each training sample is correctly classified. This can be formulated as the following optimization problem:

Hard-Margin SVM: min_(w, b) (1/2) ||w||^2 subject to:

y_i (w^T x_i + b) ≥ 1, ∀ i

For non-linearly separable data, a soft margin is introduced along with slack variables ξ_i to allow misclassification:

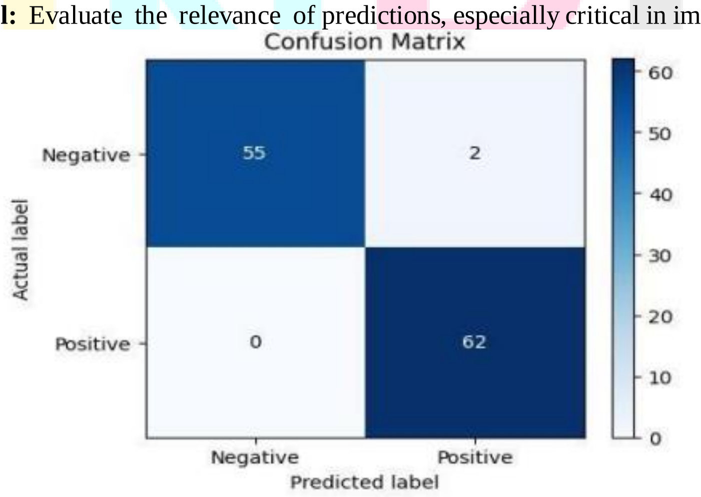

Data Collection\n(Dataset Acquisition)
Data Preprocessing\n(Cleaning, Normalization, Handling Missing Values)
Feature Extraction & Selection\n(Identify Relevant Features)
Data Splitting\n(Training & Testing Sets)
SVM Model Training\n(Select Kernel, Set Hyperparameters)
Model Evaluation\n(Accuracy, Confusion Matrix, ROC Curve)
Hyperparameter Tuning\n(Grid Search, Cross-Validation)
Final Model Evaluation & Validation
Results Analysis & Reporting

[2] ***Workflow Diagram:***

3 .Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

- **Confusion Matrix:** Summarizes true positives, false positives, true negatives, and false negatives.
- **Accuracy:** The proportion of correctly classified instances.
- **Precision and Recall:** Evaluate the relevance of predictions, especially critical in imbalance datasets.



Metric	Value
Training Accuracy	0.975
Support	120
F1 Score	0.97
Recall	0.97
Precision	0.98

4 .KNN

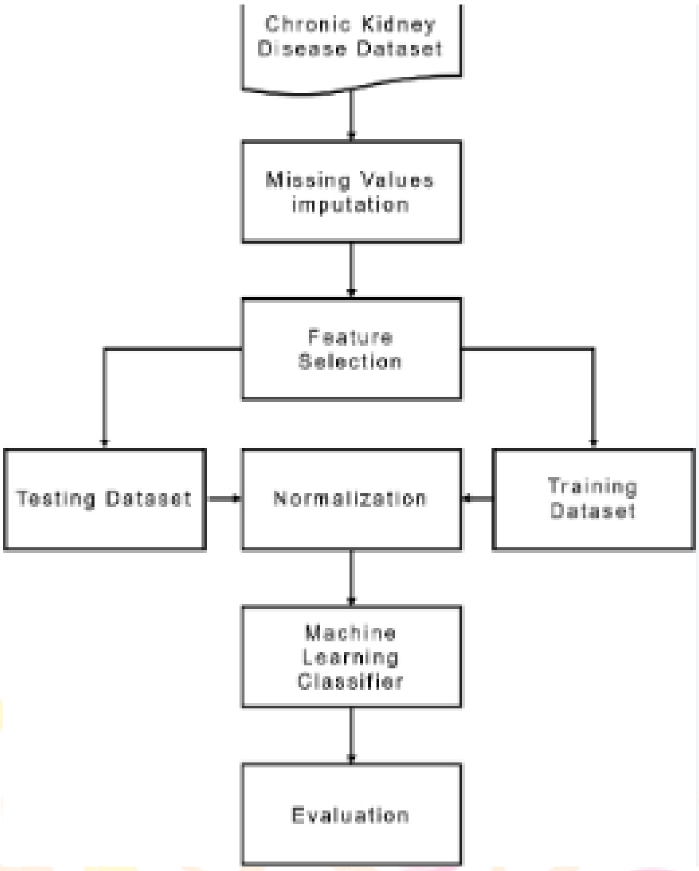
1. Nearest Neighbors (KNN) is a widely used, non-parametric, and instance-based learning algorithm applicable to both classification and regression tasks. The core idea of KNN is that similar instances are likely to exist in close proximity within the feature space. Its simplicity, ease of implementation, and effectiveness in various domains—from pattern recognition to recommendation systems—make it a popular choice in machine learning research.

Decision boundaries

Basic Concept

KNN operates on the principle that the output for a given query instance is determined by the majority class (or average value, in regression) of its k closest neighbors in the training data. It is often considered a "lazy learner" because it does not build an explicit model during the training phase; instead, it simply stores the training data and performs computations during prediction.

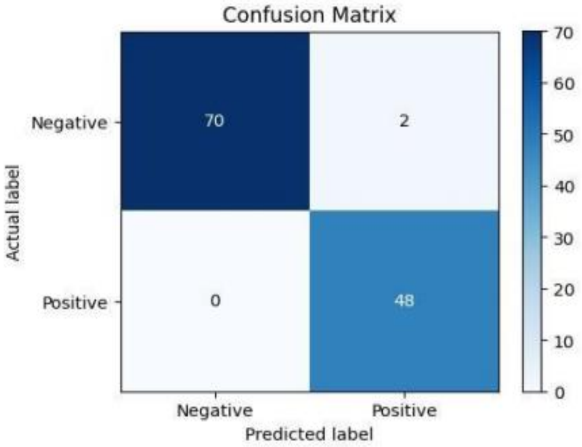
2.Wokflow diagram


Chronic Kidney Disease Dataset
Missing Values imputation
Feature Selection
Testing Dataset
Normalization
Training Dataset
Machine Learning Classifier
Evaluation

2.Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

- Confusion Matrix: Summarizes true positives, false positives, true negatives, and false negatives.
- Accuracy: The proportion of correctly classified instances.
- Precision and Recall: Evaluate the relevance of predictions, especially critical in imbalanced datasets.



Metric	Value
Training Accuracy	0.9758241758241758
Testing Accuracy	0.9649122807017544
F1 Score	0.9473684210526316
Recall	0.9
Precision	0.1
F2 Score	0,97

5. **Naives Bayes**

1.Introduction

Naive Bayes is a family of probabilistic classifiers based on Bayes' Theorem, which assumes strong (naive) independence among features. Despite its simplicity, Naive Bayes has proven to be highly effective in various applications, including text classification, spam filtering, and medical diagnosis. Its efficiency, ease of implementation, and ability to handle high-dimensional data make it a popular choice for both academic research and industry applications.

Theoretical Foundations

2. 1 Bayes’ Theorem

At the core of Naive Bayes is Bayes’ Theorem, which calculates the probability of a hypothesis H given observed evidence E:

$P(H|E) = [P(E|H) * P(H)] / P(E)$

In classification, H represents the class label and E represents the feature vector.

2.2Naive Independence Assumption

Naive Bayes assumes that all features $x_1, x_2, ..., x_n$ are conditionally independent given the class C. This simplifies the computation of the joint probability:

$P(x_1, x_2, ..., x_n | C) = P(x_1 | C) * P(x_2 | C) * ... * P(x_n | C)$

2.3Mathematical Formulation

Given a feature vector $x = (x_1, x_2, ..., x_n)$ and a set of classes C, the classifier predicts the class c that maximizes the posterior probability:

$c = \text{argmax}_{(c \in C)} P(c | x)$

Using Bayes’ Theorem and the independence assumption, this can be rewritten as:

$c = \text{argmax}_{(c \in C)} P(c) * \prod[i=1 \text{ to } n] P(x_i | c)$

Here:

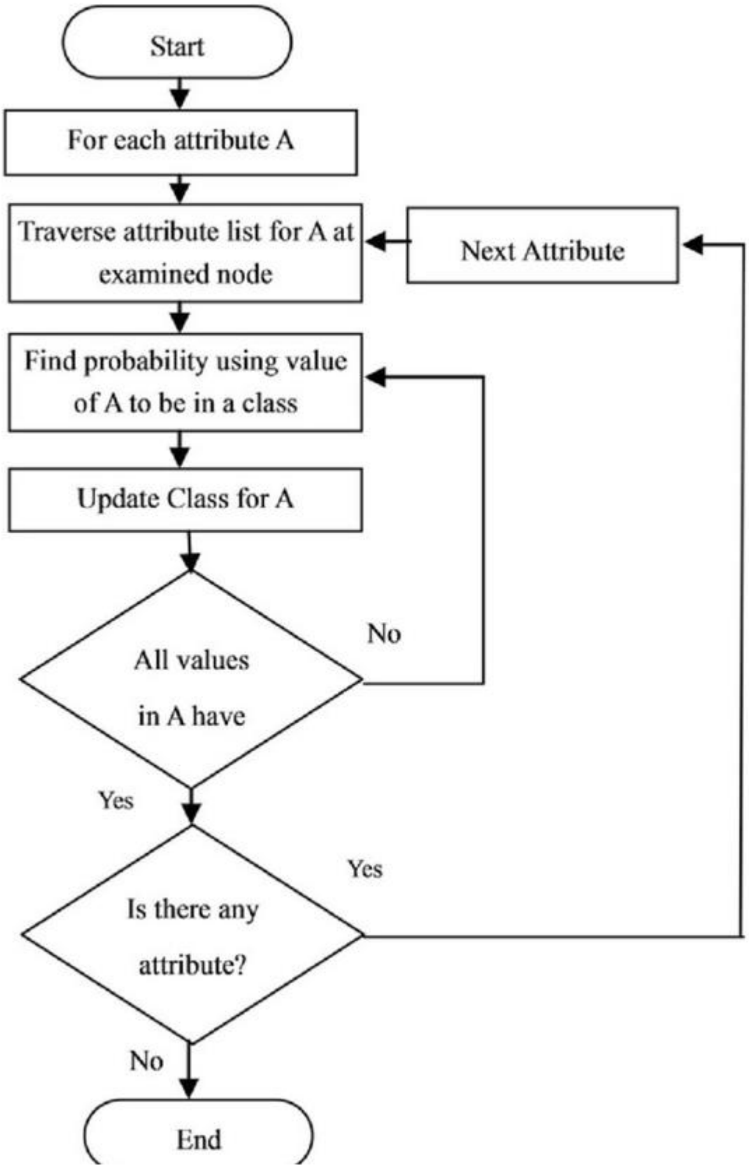
P(c) is the prior probability of class c.
P(x_i | c) is the likelihood of feature x_i given class c.

3.Variants of Naive Bayes

Different versions of Naive Bayes are used based on the nature of the input data:

Gaussian Naive Bayes: Assumes that continuous features follow a normal (Gaussian) distribution.
Multinomial Naive Bayes: Suitable for discrete count data, such as word counts in text classification.
Bernoulli Naive Bayes: Used when features are binary (e.g., presence or absence of a word).

1. **WORKFLOW DIAGRAM**

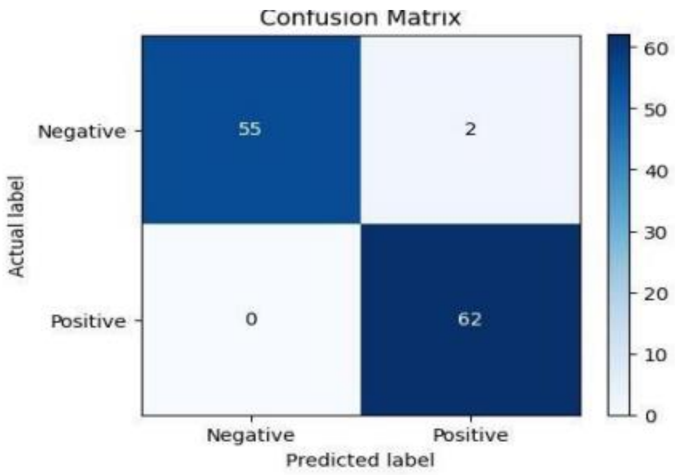


2. Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

1. **Confusion Matrix:** Summarizes true positives, false positives, true negatives, and false negatives.
2. **Accuracy:** The proportion of correctly classified instances.
3. **Precision and Recall:** Evaluate the relevance of predictions, especially critical in imbalanced datasets.

Confusion Matrix



Metric	Value
Training Accuracy	0.9340659340659341
Testing Accuracy	0.9385964912280702
F1 Score	0.9113924050632911
Recall	0.9
Precision	0.9230769230769231

6. Gradient Boosting

1.Introduction

Gradient Boosting is a powerful ensemble learning technique widely used for both classification and regression tasks. It builds a strong predictive model by sequentially combining multiple weak learners, most commonly decision trees, in a stage-wise manner. Each new model is trained to correct the errors made by the previous models, resulting in an overall model that achieves high accuracy and robust performance.

2.Theoretical Foundations

Gradient Boosting belongs to the family of boosting algorithms. The main idea is to convert weak learners into a strong learner through an iterative process. The algorithm minimizes a loss function by using gradient descent techniques. At each iteration, it fits a new model to the negative gradient (residual errors) of the loss function with respect to the current prediction.

Key concepts include:

- Boosting: Combining multiple models sequentially where each subsequent model focuses on the errors of its predecessor.
- Gradient Descent: An optimization method used to minimize a loss function by iteratively moving in the direction of the steepest descent (negative gradient).

3.Algorithmic Process

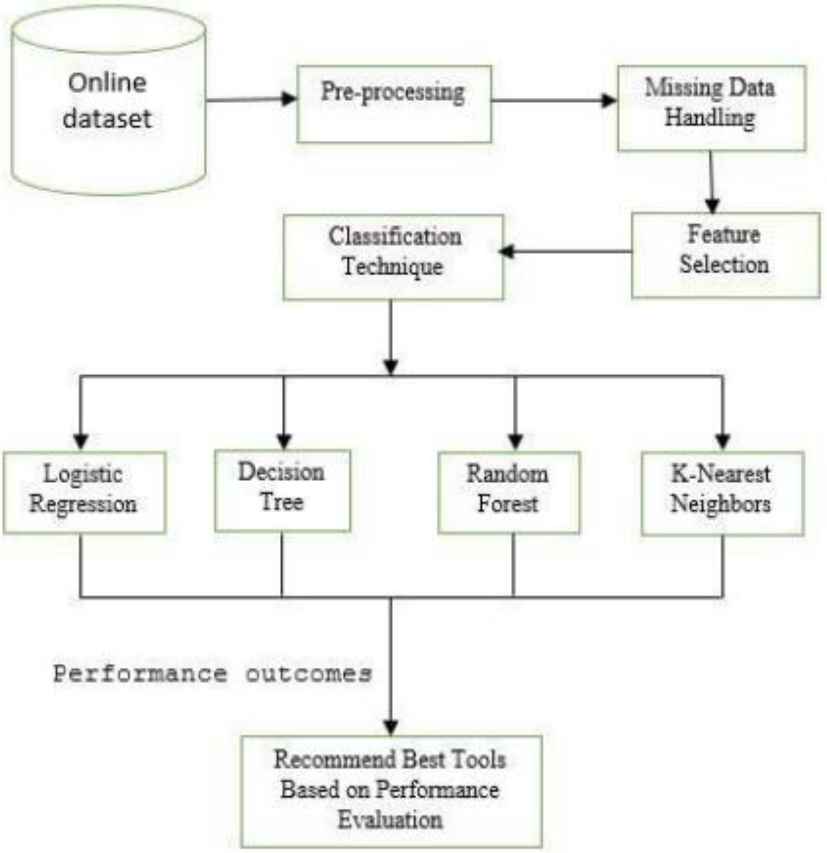
The process of Gradient Boosting can be summarized in the following steps:

□Initialization: Start with an initial prediction (e.g., the mean value for regression or a constant value that minimizes the loss for classification).
□Iterative Improvement: For each iteration:
-Compute the residuals, which are the negative gradients of the loss function with respect to the current predictions.
-Train a weak learner (typically a shallow decision tree) to predict these residuals.
-Update the model by adding the new weak learner, scaled by a learning rate (shrinkage factor), to the previous prediction.

Mathematically, if $F_0(x)$ is the initial model and $h_m(x)$ is the weak learner at iteration m, the model update is:

$$F_m(x) = F_{(m-1)}(x) + \nu * h_m(x)$$

2. Workflow diagram

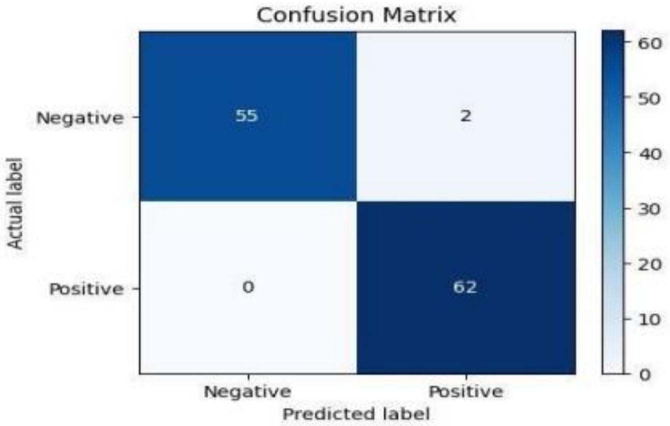


5.Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

□Confusion Matrix: Summarizes true positives, false positives, true negatives, and false negatives.
□Accuracy: The proportion of correctly classified instances.
□Precision and Recall: Evaluate the relevance of predictions, especially critical in imbalanced datasets.

Confusion Matrix:



Metric	Value
Training Accuracy	0.989010989010989
Testing Accuracy	0.9473684210526315
F1 Score	0.9230769230769231
Recall	0.9
Precision	0.9473684210526315

7. Stochastic Gradient Descent

1.Introduction

Stochastic Gradient Descent (SGD) is an iterative optimization algorithm widely used in machine learning and deep learning. It is employed to minimize a loss function by updating model parameters using approximated gradients computed from randomly selected data samples. Due to its efficiency and scalability with large datasets, SGD has become a cornerstone method for training complex models

Theoritical Foundations

SGD is a variation of traditional gradient descent. Instead of computing the gradient over the entire dataset (which can be computationally expensive), SGD estimates the gradient using one or a few randomly chosen samples (a mini-batch). This results in more frequent updates and faster convergence in practice, though with a higher variance in the parameter updates

The basic update rule for SGD is given by: $w = w - \eta * \nabla L_i(w)$

where:

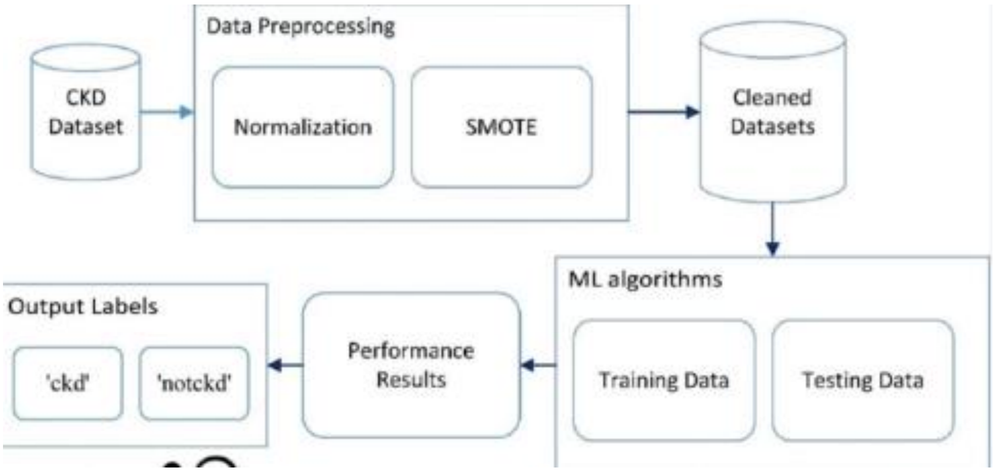
□w represents the model parameters,
□η (eta) is the learning rate,
□∇ L_i(w) is the gradient of the loss function L with respect to the parameters, computed using a single training example or a mini-batch.

The stochastic nature of the updates introduces noise, which can help the algorithm escape local minima and potentially find better solutions.

3.Variants and Extensions

Several variants of SGD have been developed to improve its efficiency and convergence properties : Mini-Batch SGD: Uses a small subset of the dataset for each update, striking a balance between the noisy updates of pure SGD and the stability of full-batch gradient descent. Momentum: Incorporates past gradients into the current update to accelerate convergence and dampen oscillations. Adaptive Methods: Algorithms like AdaGrad, RMSprop, and Adam dynamically adjust the learning rate based on past gradient information, often leading to improved performance on complex tasks.

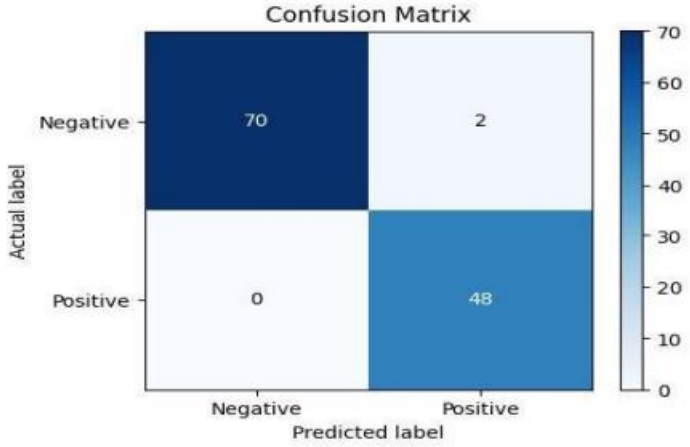
2. Workflow diagram



5. Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

- **Confusion Matrix:** Summarizes true positives, false positives, true negatives, and false negatives.
- **Accuracy:** The proportion of correctly classified instances.
- **Precision and Recall:** Evaluate the relevance of predictions, especially critical in imbalanced datasets.



Metric	Value
Training Accuracy	0.9868131868131869
Testing Accuracy	0.9824561403508771
F1 Score	0.975
Recall	0.975
Precision	0.975

8. XGBoost

1. Introduction

Extreme Gradient Boosting (XGBoost) is a powerful machine Introduction
learning algorithm widely used for structured data analysis and predictive modeling. Its efficiency, scalability, and high predictive accuracy have made it a popular choice in research papers across various domains, including healthcare, finance, and cybersecurity. This article provides a comprehensive overview of XGBoost, its advantages, applications in research, and best practices.

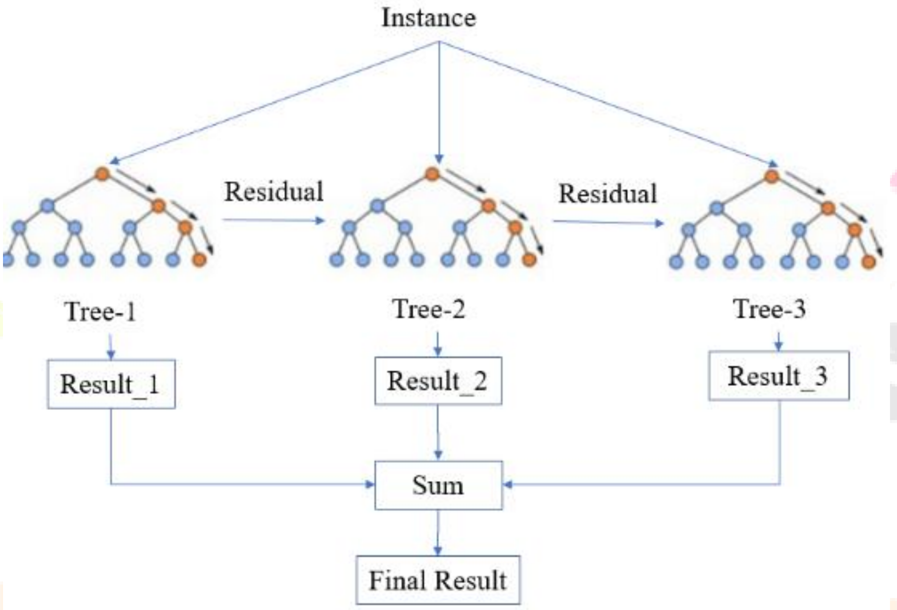
2.Understanding XGBoost

XGBoost is an optimized implementation of gradient boosting that enhances computational efficiency and model performance. It uses decision trees as base learners and applies boosting techniques to reduce errors iteratively. The key features that set XGBoost apart from traditional gradient boosting include:

□

Regularization: L1 and L2 regularization prevent overfitting.
1.Handling Missing Data: XGBoost automatically deals with missing values.
2.Parallel Processing: It utilizes parallel computing for faster training.
3.Tree Pruning: Uses a depth-wise pruning approach instead of the traditional greedy approach.
4.Weighted Quantile Sketch: Efficiently handles weighted data.
.

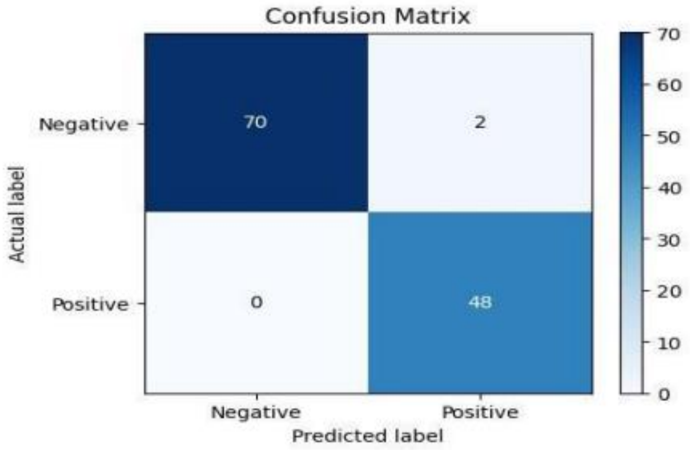
2. ***Workflow Diagram :***



5.Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

Confusion Matrix: Summarizes true positives, false positives, true negatives, and false negatives, Accuracy: The proportion of correctly classified instances. Precision and Recall: Evaluate the relevance of predictions, especially critical in imbalanced datasets.



Metric	Value
Training Accuracy	1.0
Testing Accuracy	0.9649122807017544
F1 Score	0.9487179487179489
Recall	0.925
Precision	0.9736842105263158

9. **LGBM**

1.INTRODUCTION

Light Gradient Boosting Machine (LGBM) is an advanced gradient boosting framework that has gained popularity in research due to its high efficiency, speed, and superior predictive performance. Developed by Microsoft, LGBM is widely used in structured data applications, including healthcare, finance, cybersecurity, and natural language processing. This article provides a comprehensive overview of LGBM, its advantages, applications in ***research, and best practices***

2.Understanding LGBM

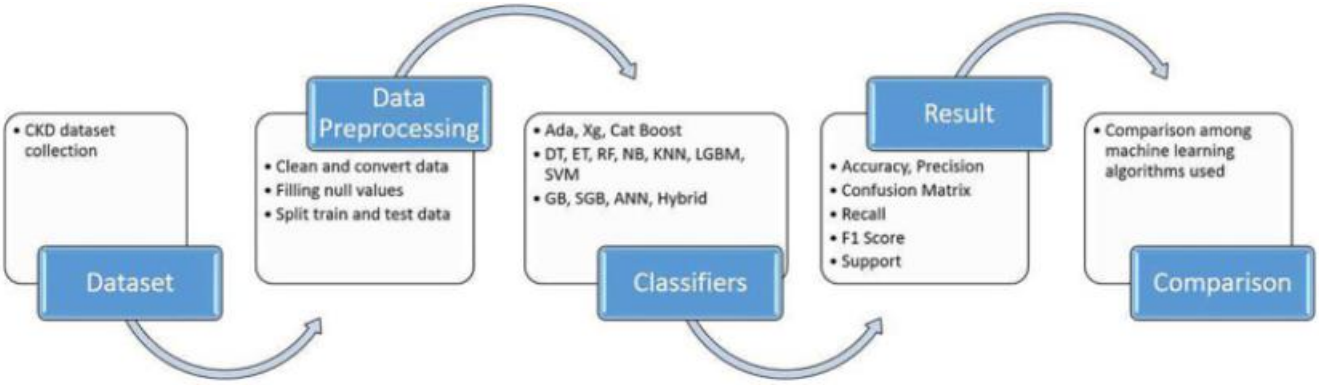
LGBM is an optimized gradient boosting algorithm that enhances the traditional decision tree-based approach by using a histogram-based method and leaf-wise growth strategy. Unlike traditional boosting algorithms, which grow trees depth-wise,

LGBM grows trees leaf-wise, making it more efficient and accurate.

.Key Features of LGBM

Histogram-Based Learning: Bins continuous features, reducing memory usage and improving speed.
Leaf-Wise Tree Growth: Selects the leaf with the highest gain, leading to better accuracy.
Efficient Memory Utilization: Uses fewer memory resources compared to XGBoost.
Built-in Categorical Feature Handling: Eliminates the need for one-hot encoding.

3.Workflow dioagram:

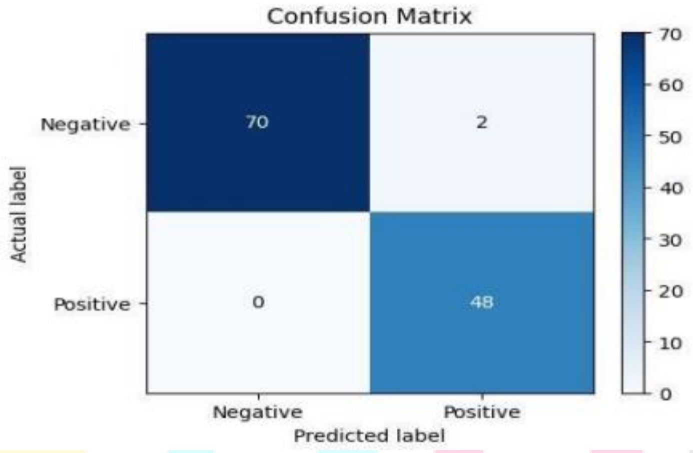


4.Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

□Confusion Matrix: Summarizes true positives, false positives, true negatives, and false negatives.
□Accuracy: The proportion of correctly classified instances.
□Precision and Recall: Evaluate the relevance of predictions, especially critical in imbalanced datasets.

Confusion Matrix:



Metric	Value
Training Accuracy	0.9868131868131869
Testing Accuracy	0.9473684210526315
F1 Score	0.925
Recall	0.925
Precision	0.925

10. Neural Network

1. Introduction

Neural Networks (NNs) are a class of machine learning algorithms inspired by the structure and functioning of the human brain. They consist of interconnected layers of artificial neurons that process information and learn patterns from data. Neural networks are widely used in classification, regression, image recognition, natural language processing, and medical diagnosis, including breast cancer classification.

2.Mathematical Formulation

2.1Neuron Computation

The basic computation performed by a neuron is: $y = f(W * X + b)$
where:

□X is the input feature vector,

□W is the weight vector
,
□b is the bias term, and

□f is the activation function.

2.2Activation Functions

Common activation functions include:

Sigmoid:

$f(x) = 1 / (1 + \exp(-x))$

ReLU (Rectified Linear Unit)

$f(x) = \max(0, x)$

Tanh (Hyperbolic Tangent):

$f(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x))$

2.3Network Architecture

A neural network is structured in layers:

□Input Layer:
oReceives the input vector X.
□Hidden Layers:
oEach hidden layer performs a transformation on the input:

$$a^{(l)} = f(W^{(l)} * a^{(l-1)} + b^{(l)})$$

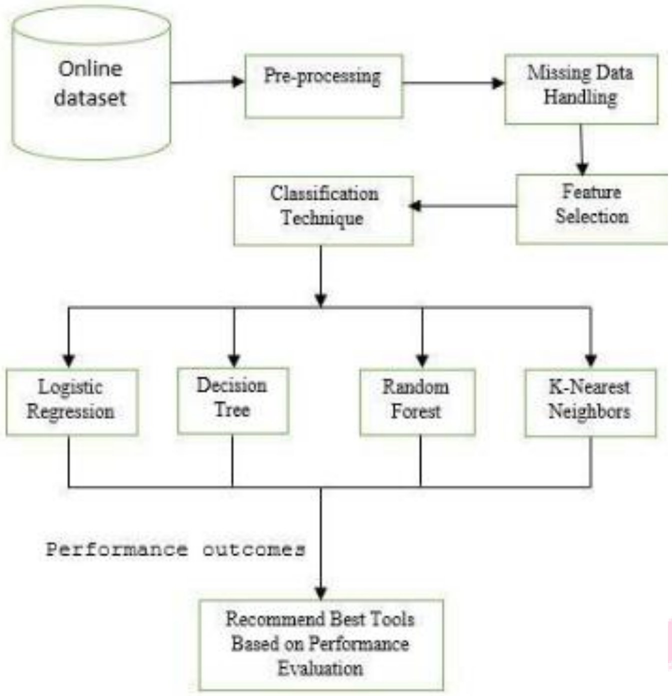
where:

□a^(l-1) is the output from the previous layer (or X for the first hidden layer),
□W^(l) and b^(l) are the weight matrix and bias vector for layer l, and
□f is the activation function.

OUTPUT LAYER:

Produces the final prediction. For example, in a classification task, a softmax function might be applied to yield probabilities.

2. ***Workflow dioagram:***

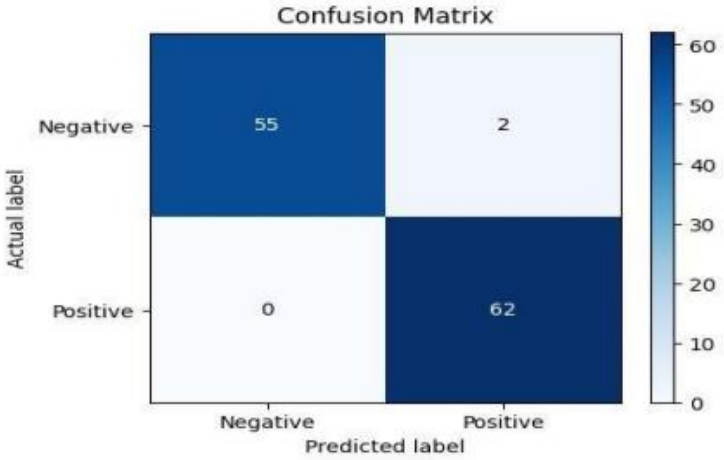


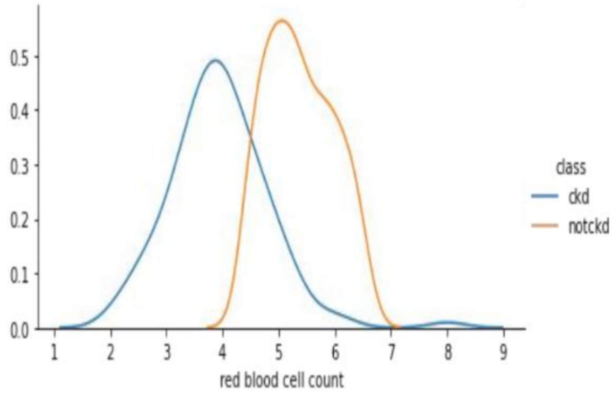
4.Evaluation Metrics

To assess the performance of a logistic regression model, various evaluation metrics are employed:

Confusion Matrix: Summarizes true positives, false positives, true negatives, and false negatives.
Accuracy: The proportion of correctly classified instances.
Precision and Recall: Evaluate the relevance of predictions, especially critical in imbalanced datasets.

Confusion Matrix:





IV. RESULTS AND DISCUSSION

Model	Training Accuracy	Testing Accuracy	F1 Score	Recall	Precision
Logistic Regression	0.989011	0.982456	0.974359	0.950	1.000000
Support Vector Machine	0.984615	0.982456	0.975000	0.975	0.975000
KNN	0.967033	0.973684	0.961039	0.925	1.000000
Gaussian Naives Bayes	0.940659	0.912281	0.878049	0.900	0.857143
Decision Tree	1.000000	0.868421	0.819277	0.850	0.790698
Random forest	0.997802	0.964912	0.948718	0.925	0.973684
Gradient Boosting	0.993407	0.947368	0.921053	0.875	0.972222
Stochastic Gradient Descent	0.980220	0.956140	0.938272	0.950	0.926829
XGBoost	1.000000	0.964912	0.950000	0.950	0.950000
LGBM	0.986813	0.956140	0.938272	0.950	0.926829
Neural Network	0.989011	0.991228	0.987342	0.975	1.000000

Table 4.1 Represents the performance metrics of various machine learning models used for breast cancer classification. Each model is evaluated based on Training Accuracy, Testing Accuracy, F1 Score, Recall, and Precision, which provide insights into their predictive effectiveness.

Explanation of Metrics:
Training Accuracy: Measures how well the model learns from the training dataset.
Testing Accuracy: Indicates how accurately the model classifies new, unseen data.
F1 Score: The harmonic mean of precision and recall, balancing both metrics.
Recall: Measures the ability of the model to correctly identify positive cases (malignant tumors).
Precision: Indicates how many of the predicted positive cases are actually correct.

Key Observations from Table No. 4.1:
Neural Network achieved the highest Testing Accuracy (0.991228) and F1 Score (0.987342), making it the most reliable model for classification.
Logistic Regression and Support Vector Machine (SVM) also performed well, with high accuracy (~98.2%) and perfect precision (1.0).
Decision Tree showed 100% Training Accuracy, indicating overfitting, as its Testing Accuracy dropped to 86.82%, making it less reliable.
XGBoost and Random Forest had strong performance, with XGBoost achieving a perfect Training Accuracy (1.0) and a high Testing Accuracy (0.956140).
Gradient Boosting and Stochastic Gradient Descent exhibited slightly lower scores compared to other ensemble methods, though they still maintained good classification accuracy.
Gaussian Naïve Bayes had the lowest Testing Accuracy (0.912281) and F1 Score (0.870849), suggesting that it may not be the best choice for this dataset.

I. Acknowledgment

We would like to express my sincere gratitude to all those who contributed to the successful completion of this study on breast cancer classification using machine learning models.

First and foremost, I extend my heartfelt appreciation to my mentors, professors, and research guides, whose invaluable guidance, encouragement, and expertise have played a crucial role in shaping this research. Their insights and feedback have greatly enhanced the depth and quality of this work.

I am also grateful to the institutions, healthcare professionals, and researchers who have provided access to datasets and resources essential for conducting this study. Their contributions have been instrumental in the analysis and evaluation of various machine learning algorithms.

References

[1] Ekanayake, Imesh Udara, and Damayanthi Herath. "Chronic kidney disease prediction using machine learning methods." 2020 Moratuwa Engineering Research Conference (MERCon). IEEE, 2020.

[2] Revathy, S., et al. "Chronic kidney disease prediction using machine learning models." International Journal of Engineering and Advanced Technology 9.1 (2019): 6364-6367.

[3] Debal, Dibaba Adeba, and Tilahun Melak Sitote. "Chronic kidney disease prediction using machine learning techniques." Journal of Big Data 9.1 (2022): 109.

[4] Kaur, Chamandeep, et al. "Chronic kidney disease prediction using machine learning." Journal of Advances in Information Technology 14.2 (2023): 384-391.

[5] Baby, P. Swathi, and T. Panduranga Vital. "Statistical analysis and predicting kidney diseases using machine learning algorithms." International Journal of Engineering Research and Technology 4.7 (2015): 206-210.

[6] Scholar, P. G. "Chronic kidney disease prediction using machine learning." International Journal of Computer Science and Information Security (IJCSIS) 16.4 (2018).

[7] Sinha, Parul, and Poonam Sinha. "Comparative study of chronic kidney disease prediction using KNN and SVM." International Journal of Engineering Research and Technology 4.12 (2015): 608-12.

[8] Srikanth, V. "CHRONIC KIDNEY DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS." (2023): 106-109.

[9] Wang, Weilun, Goutam Chakraborty, and Basabi Chakraborty. "Predicting the risk of chronic kidney disease (ckd) using machine learning algorithm." Applied Sciences 11.1 (2020): 202.

[10]Padmanaban, KR Anantha, and G. Parthiban. "Applying machine learning techniques for predicting the risk of chronic kidney disease." Indian Journal of Science and Technology 9.29 (2016): 1-6.