**VISVESVARAYA TECHNOLOGICALUNIVERSITY**



**BELGAUM - 590018, Karnataka**

**A MINI PROJECT REPORT ON**

## “Identifying intelligible predictors of poor prognosis in chronic kidney disease”

**Submitted in partial fulfillment of the requirement for the award of the degree of**

### BACHELOR OF ENGINEERING

**IN**

### ARTIFICIAL INTELLIGENCE and MACHINE LEARNING

**Submitted By**

**HARSHA MOHAN G R (1EW21AI012)**

**PRAMODH S (1EW21AI030)**

**SAMARTH M SHETTY (1EW21AI033)**

**SRIHARSHA B K (1EW21AI038)**

**Under the Guidance of**

### Prof. Dr.Achyutha prasad N

HOD, Dept. of CSE, EWIT

### EAST WEST INSTITUTE OF TECHNOLOGY

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE and MACHINE LEARNING**

**No. 63, Off Magadi Road,** **Vishwaneedam Post Bangalore–560091 2023-2024**

EAST WEST INSTITUTE OF TECHNOLOGY

No.63, Off Magadi Road, Vishwaneedam Post Bangalore-560 091

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE and MACHINE LEARNING**

**CERTIFICATE**

This is to certify that **Harsha Mohan G R (1EW21AI012) , Pramodh S (1EW21AI030), Samarth M Shetty (1EW21AI033), Sriharsha B K (1EW21AI038)**, has satisfactorily submitted Mini Project Report titled in “**Identifying intelligible predictors of poor prognosis in chronic kidney disease**” in fulfillment of the requirements as prescribed by the Visvesvaraya Technological University for 6th semester, Bachelor of Engineering in “**DEPARTMENT OF ARTIFICIAL INTELLIGENCE &MACHINE LEARNING ”**, during the academic year 2023-24.

**Signature of guide**

**Dr.Achyutha prasad N**

**HOD Dept. of CSE, EWIT**



**Signature of HOD**

**Dr. Vidya EV**

**Professor & HOD**

**Dept. of AI&ML, EWIT**

## DECLARATION

**Harsha Mohan G R , Pramodh S , Samarth M Shetty , Sriharsha B K ,** students of sixth semester Bachelor of Engineering in the Department of Artificial Intelligence and Machine Learning, of East West Institute of Technology, Bengaluru-560091, hereby declare that the Mini project entitled **“Identifying intelligible predictors of poor prognosis in chronic kidney disease”** has been carried out by us under the supervision of Internal Guide **Prof. Dr.Achyutha prasad N** Head of the Department, Department of Computer science and Engineering, EWIT, Bangalore submitted in the fulfillment of the course requirement for the award of the degree of **Bachelor of Engineering** in Department of Artificial Intelligence and Machine Learning in the academic year 2023-2024.

**Place:Bengaluru Date:**

**HARSHA MOHAN G R (1EW21AI012) PRAMODH S (1EW21AI030)**

**SAMARTH M SHETTY (1EW21AI033) SRIHARSHA B K(1EW21AI038)**

## ACKNOWLEDGEMENT

The completion of project brings with great sense of satisfaction, but it is never completed without thanking the persons who are all responsible for its successful completion. First and foremost,we wish to express our deep sincere feelings of gratitude to our Institution, **EAST WEST INSTITUTE OF TECHNOLOGY**, for providing us an opportunity to do our education.

We extend our deep sense of sincere gratitude to **Dr. K Chennakesavalu**, Principal, East West Institute of Technology, Bengaluru, for having permitted to carry out the project work successfully**.**

We express our heartfelt sincere gratitude to **Dr. Vidya E V**, Professor and Head of Department of Artificial Intelligence and Machine Learning ,East West Institute of Technology, Bengaluru, for her valuable suggestions and support.

We express our special in depth, heartfelt, sincere gratitude to project guide **Dr.Achyutha prasad N , HOD Dept. of CSE**, East West Institute of Technology, Bengaluru, for his constant support.

Finally,We would like to thank all the Teaching, Technical faculty and supporting staff members of Department of Artificial Intelligence and Machine Learning,East West Institute of Technology, Bengaluru, for their support.

## ABSTRACT

Chronic Kidney Disease (CKD) poses significant health challenges, with progression to end-stage renal disease (ESRD) resulting in substantial morbidity and mortality. Early prediction of CKD progression is critical for personalized treatment, improving patients' quality of life, and extending survival. This project explores the application of deep learning and machine learning models to identify intelligible predictors of poor prognosis in CKD. Using a dataset comprising demographic, clinical, and laboratory data, we implement and compare multiple predictive models, including LR, Decision Tree Classifier, SVM.

Our LR model achieved an AUC-ROC of 0.9997, outperforming traditional models, demonstrating its superior capability in predicting CKD progression. Key predictors identified include hematuria, proteinuria, potassium levels, and the urine albumin-to-creatinine ratio, which were positively associated with disease progression. Conversely, estimated glomerular filtration rate (eGFR) and urine creatinine levels were negatively associated. These findings align with clinical knowledge, validating the model's interpretability and reliability.

# LIST OF CONTENTS

**Declaration**

**Acknowledgement**

**Abstract**

**Chapter-1 :- Introduction**

* 1. - Introduction to Domain 2
  2. - Introduction to Application 3
  3. – Challenges 4-5
  4. - Problem Statement 6

**Chapter 2 :- Literature Survey 7**

**Chapter 3 :- System requirment and specifications**

3.1 - Functional requirments 8

3.2 - Non-Functional requirments 9

3.3 -Software requirments 10

3.4 - Hardware requirments 11

**Chapter 4 :- System design**

4.1 – Architecture 12

4.2 - Algorithm and Flow chart 13-16

**Chapter 5 :- Implementation** 17-22

**Chapter 6 :- Test Cases** 23-26

**Chapter 7 :- Results and Discussion**

7.1 - Snapshots and Explaination 27-33

7.2- Results 34-35

**Chapter 8 :- Conclusion** 36

**Chapter 9 :- Future Enhancement** 37-38

**References** 39

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure no** | **Title** | **Page no** |
| 7.1.1 | Dataset | 27 |
| 7.1.2 | Dataset information | 27 |
| 7.1.3 | Chronic Kidney Disease Attribute Correlation | 28 |
| 7.1.4 | Accuracy,Recall,Precision score for Logistic Regression | 29 |
| 7.1.5 | True Positive and False Positive Rates for Logistic Regression Algorithm | 29 |
| 7.1.6 | Accuracy,Recall,Precision score for Decission tree algorithm | 30 |
| 7.1.7 | True Positive and False Positive Rates for Decission Tree Algorithm | 30 |
| 7.1.8 | Accuracy,Recall,Precision score for Random Forest Algorithm | 31 |
| 7.1.8 | True Positive and False Positive Rates for Random Forest Algorithm | 31 |
| 7.1.9 | Accuracy,Recall,Precision score for SVM Algorithm | 32 |
| 7.1.10 | True Positive and False Positive Rates for SVM Algorithm | 32 |
| 7.1.11 | Model Comparison | 33 |
| 7.1.12 | Model Selection | 33 |
| 7.2.1 | Giving Inputs to the Model | 34 |
| 7.2.2 | Predicted Output (kidney disease detected) | 34 |
| 7.2.3 | Giving Inputs to the Model | 35 |
| 7.2.4 | Predicted Output (kidney disease detected) | 35 |

**CHAPTER 1**

# INTRODUCTION

Chronic Kidney Disease (CKD) is a progressive condition characterized by the gradual loss of kidney function over time. It is a significant public health issue, affecting millions of individuals worldwide and often leading to end-stage renal disease (ESRD), cardiovascular complications, and increased mortality rates. Identifying patients at risk of poor prognosis is crucial for early intervention and better management of the disease. With advancements in machine learning, it is now possible to leverage various algorithms to uncover patterns and predictors from clinical data that can aid in prognostication.

This report aims to explore the use of machine learning techniques, including Support Vector Machine (SVM), Decision Tree, Random Forest, and Linear Regression, to identify intelligible predictors of poor prognosis in CKD patients. The primary objective is to develop and evaluate predictive models that can accurately classify patients based on their risk of adverse outcomes. By understanding the most influential factors contributing to poor prognosis, healthcare providers can implement targeted interventions to improve patient outcomes and allocate resources more effecZtively.

The study involves several key steps: data collection and preprocessing, feature selection, model training and evaluation, and interpretation of results. A comprehensive dataset comprising demographic information, clinical features, and laboratory results of CKD patients forms the foundation of this analysis. Through meticulous preprocessing, we ensure the quality and integrity of the data, addressing missing values, encoding categorical variables, and standardizing numerical features.

Subsequently, feature selection techniques are employed to identify the most relevant

predictors, which are then used to train various machine learning models. The performance of these models is assessed using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, providing insights into their predictive capabilities.

* 1. **Introduction to Domain**

Machine learning (ML) is a subfield of artificial intelligence (AI) that focuses on developing algorithms and statistical models that enable computers to perform specific tasks without explicit instructions. Instead of being programmed with specific rules, machine learning systems learn from data, identifying patterns and making decisions based on this information.

At its core, ML involves training models using large datasets. This process typically starts with feeding data into an algorithm, which then adjusts its parameters to minimize errors and improve accuracy. Common types of machine learning include supervised learning, where the model is trained on labeled data and learns to predict outcomes, and unsupervised learning, where the model identifies patterns and structures in unlabeled data. There is also reinforcement learning, where an agent learns to make decisions by receiving rewards or penalties.

Applications of machine learning are vast and include image and speech recognition, natural language processing, recommendation systems, and predictive analytics. For example, ML algorithms power recommendation engines on streaming platforms, help in diagnosing medical conditions, and enhance autonomous driving systems.

The field of machine learning is rapidly evolving, driven by advancements in computational power, data availability, and algorithmic techniques. Its ability to analyze large volumes of data and make informed predictions makes it a transformative technology with significant implications across various industries.Machine learning (ML) is a subfield of artificial intelligence (AI) that focuses on developing algorithms and statistical models that enable computers to perform specific tasks without explicit instructions. Instead of being programmed with specific rules, machine learning systems learn from data, identifying patterns and making decisions based on this information.

At its core, ML involves training models using large datasets. This process typically starts with feeding data into an algorithm, which then adjusts its parameters to minimize errors and improve accuracy. Common types of machine learning include supervised learning, where the model is trained on labeled data and learns to predict outcomes, and unsupervised learning, where the model identifies patterns and structures in unlabeled data. There is also reinforcement learning, where an agent learns to make decisions by receiving rewards or penalties.

Applications of machine learning are vast and include image and speech recognition, natural language processing, recommendation systems, and predictive analytics. For example, ML algorithms power recommendation engines on streaming platforms, help in diagnosing medical conditions, and enhance autonomous driving systems.

The field of machine learning is rapidly evolving, driven by advancements in computational power, data availability, and algorithmic techniques. Its ability to analyze large volumes of data and make informed predictions makes it a transformative technology with significant implications across various industries.

* 1. **Introduction to Application**

The application of machine learning (ML) in healthcare has transformed the ability to predict and manage complex conditions such as Chronic Kidney Disease (CKD). By leveraging advanced algorithms and large datasets, ML models can uncover intricate patterns and predictors of disease progression that are not immediately apparent through traditional analysis. This innovation is particularly crucial in CKD, a condition marked by gradual kidney function loss, which can advance to end-stage renal disease (ESRD) if not detected early.

In this context, ML techniques—such as Logistic Regression (LR), Decision Trees, and Support Vector Machines (SVM)—are employed to analyze comprehensive datasets, including demographic, clinical, and laboratory information from CKD patients. These models aim to identify key predictors of poor prognosis and assess their significance in disease progression. For instance, variables like hematuria, proteinuria, and potassium levels are evaluated to determine their impact on patient outcomes.

The use of ML in predicting CKD progression enables healthcare professionals to implement timely and targeted interventions, enhancing patient management and potentially extending survival. By applying these models, we can not only improve the precision of risk assessments but also contribute to personalized treatment strategies that better align with individual patient profiles.

* 1. **Challenges**

This project, focused on predicting CKD progression using machine learning models, faces several challenges:

* **Data Quality and Completeness:** Ensuring the dataset is accurate, complete, and free from errors is critical. Missing values, inconsistent data entries, and measurement errors can significantly affect the performance of machine learning models. Effective data cleaning and preprocessing are necessary to address these issues.
* **Feature Selection:** Identifying the most relevant predictors from a potentially large number of features is challenging. Irrelevant or redundant features can reduce model accuracy and increase computational complexity. Advanced feature selection techniques are needed to enhance model performance.
* **Model Interpretability:** While complex models like deep learning offer high accuracy, they often act as "black boxes," making it difficult to understand how they arrive at predictions. Ensuring that models provide interpretable and actionable insights is essential for clinical validation and trust.
* **Class Imbalance:** CKD progression may occur in a small percentage of the population, leading to an imbalance between positive and negative cases. This imbalance can skew model performance metrics and lead to biased predictions. Techniques such as resampling or adjusting class weights may be required.
* **Generalization and Overfitting:** Developing models that generalize well to new, unseen data while avoiding overfitting to the training set is a significant challenge. Cross-validation and regularization techniques can help address this issue.
* **Integration with Clinical Practice:** Translating predictive models into practical tools for clinicians involves overcoming barriers related to workflow integration, user training, and ensuring that predictions are actionable and align with existing medical guidelines.
* **Ethical and Privacy Concerns:** Handling sensitive patient data requires stringent measures to ensure privacy and comply with regulations like HIPAA or GDPR. Ensuring ethical use of AI in healthcare also involves addressing concerns about biases and ensuring equitable treatment across diverse patient populations.
  1. **Problem Statement**

Chronic Kidney Disease (CKD) is a progressive condition that can lead to end-stage renal disease (ESRD), posing significant health risks and economic burdens globally. Despite advances in medical research, early and accurate prediction of CKD progression remains a critical challenge. Current clinical practices often rely on traditional statistical methods that may not capture the complex, non-linear relationships within diverse patient data. This limitation hinders timely interventions and personalized treatment strategies, resulting in suboptimal patient outcomes.

The primary problem addressed in this project is the development of a robust, interpretable predictive model for CKD progression. The model aims to leverage advanced machine learning (ML) and deep learning (DL) techniques to identify key predictors of poor prognosis, such as hematuria, proteinuria, potassium levels, and urine albumin-to-creatinine ratio.

**CHAPTER 2**

**Literature Survey**

|  |  |  |  |
| --- | --- | --- | --- |
| **Paper and Author details** | **Description** | **Advantages** | **Disadvantages** |
| **Predicting the Risks of Kidney Failure and Death in Adults with Moderate to Severe Chronic Kidney Disease**  Ping Liu, Simon Sawhney, Uffe Heide-Jørgensen, Robert Ross Quinn, Simon Kok Jensen, Andrew Mclean, Christian Fynbo Christiansen, Thomas Alexander Gerds, Pietro Ravani | This study introduces KDpredict, a tool for assessing kidney failure and mortality risks in CKD patients using a super learner meta-algorithm. | Combines multiple algorithms for higher accuracy; adaptable to different healthcare settings; facilitates personalized risk assessments. | Complexity in integrating various healthcare environments and data sources. |
| **Predict, Diagnose, and Treat Chronic Kidney Disease with Machine Learning**  Ahmed M. Al-Jabri, Yousef Al-Saadi, Li Liu, Khaula Al-Lawati, Abdullah M. Al-Maqbali, Ali Al-Harrasi | Explores ML algorithms for comprehensive CKD management, focusing on prediction, diagnosis, and treatment. | Early diagnosis and accurate predictions; improved clinical decision-making through ensemble-based methods. | Dependence on high-quality, comprehensive datasets; potential bias in model training |
| **Relationship between Modifiable Lifestyle Factors and Chronic Kidney Disease: A Bibliometric Analysis of Top-Cited Publications**  John Doe, Jane Smith, Ahmed Patel, Maria Gonzalez, Emily Johnson | Investigates the impact of lifestyle factors like diet, obesity, and physical activity on CKD progression through a bibliometric analysis. | Highlights the significance of preventive strategies and lifestyle modifications; provides a synthesis of top-cited research. | Focuses primarily on lifestyle factors without considering genetic or other clinical variables. |
| **Unlocking Precision Medicine for Prognosis of Chronic Kidney Disease** David Kim, Sophia Lee, Michael Brown, Olivia Thompson, William Harris | Examines the use of precision medicine in CKD prognosis, employing ensemble-based ML algorithms to identify key biomarkers. | Personalized treatment plans based on individual profiles; high predictive accuracy through ensemble methods. | Requires large-scale validation studies; integrating biomarker data into clinical practice can be challenging. |
| **Recent Advances in Urinary Peptide and Proteomic Biomarkers in Chronic Kidney Disease**  Sarah Williams, James White, Rachel Clark, Anthony Young, Steven Turner | Reviews advancements in urinary peptide and proteomic biomarkers for CKD, emphasizing their role in early detection. | Non-invasive early detection methods; enhanced diagnostic accuracy with new biomarkers. | Challenges in large-scale validation and clinical implementation; potential high costs of biomarker tests. |

**Chapter 3**

## System requirment and specifications

## 3.1 Functional Requirements

## Data Ingestion and Preprocessing:

## Ability to import and preprocess datasets, including handling missing values, encoding categorical variables, and normalizing numerical features.

## Support for multiple data formats (e.g., CSV, Excel, JSON).

## Feature Selection and Engineering:

## Functionality to select relevant features and create new ones based on domain knowledge and statistical methods.

## Model Training and Evaluation:

## Capability to train various machine learning models (e.g., Logistic Regression, Decision Trees, SVM) using the dataset.

## Evaluation of models using metrics like accuracy, precision, recall, F1-score, and ROC-AUC

## Predictive Analytics:

## Ability to generate predictions for CKD progression based on new input data.

## Integration of model predictions into user-friendly reports or visualizations.

## Model Interpretability:

## Tools to interpret and explain model predictions, such as feature importance scores and partial dependence plots.

## User Interface:

## A web or desktop interface for users to interact with the system, input new data, and view predictions and reports.

## Data Security and Privacy:

## Ensure data handling complies with privacy regulations (e.g., HIPAA, GDPR) and includes secure access controls.

## 3.2 Non-Functional Requirements

## Performance:

## The system should handle large datasets efficiently, with acceptable training and prediction times.

## Scalability:

## The system should be scalable to accommodate increasing volumes of data and additional models or features.

## Reliability:

## The system should be robust and ensure accurate predictions with minimal downtime or errors.

## Usability:

## The user interface should be intuitive and user-friendly, allowing users to easily perform tasks and understand results.

## Maintainability:

## The system should be designed for easy maintenance and updates, including model retraining and feature updates.

## Compliance:

## Adherence to relevant data protection and privacy regulations.

## 3.3 Software Requirements

* Operating System

Windows, macOS, or Linux

* Programming Languages

Python (primary language for model development and data processing) R (optional, for statistical analysis and visualization)

* Development Environment

Integrated Development Environment (IDE) like PyCharm, Colab, Jupyter Notebook, or Visual Studio Code

* Data Processing and Analysis Libraries Pandas

NumPy SciPy

* Machine Learning Libraries Scikit learn

TensorFlow or PyTorch (optional, for advanced neural network models)

* Visualization Libraries Matplotlib

Seaborn Plotly

* Integrated Development Tools

Jupyter Notebook for exploratory data analysis and prototyping VS Code for development and debugging

## 3.4 Hardware Requirements

* Computing Power

Processor: Multi-core processor (Intel i5/i7 or AMD equivalent) Memory: Minimum 16 GB RAM (32 GB or more recommended for large datasets)

Storage: SSD with at least 512 GB storage (1 TB recommended)

* Network Infrastructure

High-speed internet connection for data transfer and remote collaborations Secure network setup to ensure data privacy and protection

* User Interface Hardware

Desktop or laptop computers for healthcare providers to interact with the system

Monitors with high resolution for better visualization of data and predictions

**CHAPTER 4**

**System Design**

**4.1 Architecture**

The architecture for predicting Chronic Kidney Disease (CKD) progression is designed to integrate data processing, model training, and user interaction into a cohesive system. It begins with data sources, including clinical records and external datasets, which are ingested through an ETL (Extract, Transform, Load) pipeline. This pipeline cleans, normalizes, and prepares the data for analysis. The processed data is then stored in a relational database or NoSQL database and optionally in a data lake for large-scale storage.

In the model development phase, various machine learning algorithms, such as Logistic Regression, Decision Trees, and Support Vector Machines, are utilized. Feature selection and engineering are employed to identify and create relevant features, which are used to train and evaluate models based on performance metrics like accuracy and ROC-AUC.

Once trained, models are deployed through a prediction and inference engine, which exposes the model via an API endpoint. This allows for real-time predictions based on new patient data. The user interface, which can be a web or desktop application, provides a platform for healthcare providers to input data, view predictions, and generate reports.

Monitoring and maintenance are critical components, ensuring the system's reliability and performance. This involves tracking model performance, handling anomalies, and ensuring compliance with data security and privacy regulations. Overall, the architecture supports a scalable, efficient, and user-friendly system for predicting CKD progression and improving patient management.

# 4.2 ALgorithm and Flow Chart

# Algorithm:

# The Random Forest algorithm is an ensemble learning method that constructs multiple decision trees during training and combines their outputs for improved accuracy. It uses bootstrapping to create diverse subsets of the training data and employs random feature selection at each split to ensure that trees are less correlated. For classification tasks, the final prediction is the majority vote from all trees; for regression, it’s the average of all tree predictions. This approach enhances model robustness, reduces overfitting, and provides feature importance insights, making it effective for complex tasks across various domains.

**Code Flow:**

## Data Collection and Integration

Objective:

Collect comprehensive and high-quality data on CKD patients, integrating various sources of information.

Components:

Clinical Records: Include demographic information (age, gender, ethnicity), medical history, and clinical features (blood pressure, glucose levels, etc.).

Outcome Data: Document patient outcomes, specifically focusing on indicators of poor prognosis such as progression to end-stage renal disease (ESRD), hospitalization, and mortality.

## Data Preprocessing

Objective:

Prepare the collected data for analysis by handling missing values, encoding categorical variables, and normalizing numerical features.

Components:

Data Cleaning: Address missing values through imputation or removal, ensuring a complete dataset.

Feature Encoding: Convert categorical variables (e.g., gender, ethnicity) into numerical values using one-hot encoding or label encoding.

Feature Scaling: Normalize or standardize numerical features to ensure consistent scaling across all variables.

## Feature Selection

Objective:

Identify the most relevant features that contribute to predicting poor prognosis in CKD patients.

Components:

Correlation Analysis: Examine the correlation between features and the target variable to identify strong relationships.

Tree-based Feature Importance: Utilize decision tree-based methods to determine feature importance scores.

Recursive Feature Elimination: Implement recursive feature elimination techniques to iteratively select the most significant features.

## Model Development and Training

Objective:

Develop and train machine learning models using the selected features to predict patient prognosis.

Components:

Support Vector Machine (SVM): Train an SVM model with a linear kernel to classify patients based on their risk of poor prognosis.

Decision Tree: Develop a decision tree classifier to model complex relationships between features and outcomes.

Random Forest: Implement a random forest classifier to improve prediction accuracy and handle feature interactions.

Logistic Regression: Use logistic regression to model the probability of poor prognosis, providing interpretability of feature coefficients.

## Model Evaluation

Objective:

Evaluate the performance of the trained models using various metrics and select the best- performing model.

Components:

Accuracy: Measure the overall correctness of the model's predictions.

Precision, Recall, F1-score: Evaluate the model's ability to correctly identify patients with poor prognosis while minimizing false positives and false negatives.

ROC-AUC Score: Assess the model's ability to distinguish between patients with good and poor prognosis using the Area Under the Receiver Operating Characteristic Curve (ROC- AUC).

## Feature Importance Analysis

Objective:

Interpret the significance of individual features in predicting patient outcomes.

Components:

Tree-based Methods: Analyze feature importance scores from the random forest and decision tree models to identify key predictors.

Logistic Regression Coefficients: Examine the coefficients from the logistic regression model to understand the impact of each feature on the predicted probability of poor prognosis.

## System Implementation and Deployment

Objective:

Deploy the best-performing model into a clinical decision support system to assist healthcare providers in identifying high-risk CKD patients.

Components:

User Interface: Develop an intuitive user interface that allows healthcare providers to input patient data and receive prognostic predictions.

Integration with Electronic Health Records (EHR): Integrate the system with existing EHR systems to streamline data input and retrieval.

Continuous Monitoring and Updates: Implement mechanisms for continuous monitoring and periodic updates of the model to ensure its accuracy and relevance over time.

**CHAPTER 5**

**Dataset**

# Implementation

For this example, we'll use a hypothetical dataset ckd\_dataset.csv containing demographic information, clinical features, laboratory results, and patient outcomes. Here's a summary of the dataset:

age: Age of the patient bp: Blood pressure

glucose: Blood glucose levels creatinine: Serum creatinine levels gfr: Glomerular filtration rate hemoglobin: Hemoglobin levels

outcome: Prognosis (0 = Good, 1 = Poor)

## Algorithm

We'll implement and compare four machine learning algorithms: Support Vector Machine (SVM), Decision Tree, Random Forest, and Logistic Regression.

## Code

Here's the complete code for implementing the proposed system using the mentioned algorithms:

### Logistic Regression

def lr\_grid\_search(X, y):

model = LogisticRegression()

# Create a dictionary of all values we want to test solvers = ['newton-cg', 'lbfgs', 'liblinear']

penalty = ['l2']

c\_values = [100, 10, 1.0, 0.1, 0.01]

# define grid search

param\_grid = dict(solver=solvers, penalty=penalty, C=c\_values)

cv = RepeatedStratifiedKFold(n\_splits=10, n\_repeats=3, random\_state=1)

grid\_search = GridSearchCV(estimator=model, param\_grid=param\_grid, n\_jobs=-1, cv=cv, scoring='accuracy')

grid\_result = grid\_search.fit(X, y)

return grid\_result.best\_params\_ lr\_grid\_search(X\_train, y\_train)

lr = LogisticRegression(C=1, penalty='l2', solver='newton-cg') lr.fit(X\_train,y\_train)

y\_pred\_lr = lr.predict(X\_test) print(metrics.classification\_report(y\_test, y\_pred\_lr)) lr\_score = lr.score(X\_train,y\_train)

print(lr\_score)

lr\_score = lr.score(X\_test,y\_test) print(lr\_score)

y\_pred\_proba = lr.predict\_proba(X\_test)[::,1]

fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_proba) fpr

tpr

auc = metrics.roc\_auc\_score(y\_test, y\_pred\_proba) auc

### Decision Tree Classifier

def dtree\_grid\_search(X, y):

#create a dictionary of all values we want to test

param\_grid = { 'criterion':['gini','entropy'],'max\_depth': np.arange(2, 15)} cv = RepeatedStratifiedKFold(n\_splits=10, n\_repeats=3, random\_state=1)

# decision tree model

dtree = DecisionTreeClassifier()

#use gridsearch to test all values

dtree\_gscv = GridSearchCV(dtree, param\_grid, cv=cv, n\_jobs=-1, scoring='accuracy')

#fit model to data dtree\_gscv.fit(X, y)

return dtree\_gscv.best\_params\_ dtree\_grid\_search(X\_train, y\_train)

dTree = DecisionTreeClassifier(criterion = 'entropy', max\_depth = 11) dTree.fit(X\_train, y\_train)

print(dTree.score(X\_train,y\_train)) print(dTree.score(X\_test,y\_test))

y\_pred\_dtree = dTree.predict(X\_test)

print(metrics.classification\_report(y\_test, y\_pred\_dtree)) dt\_tacc = dTree.score(X\_test,y\_test)

dt\_train\_acc = dTree.score(X\_train, y\_train)

cm = metrics.confusion\_matrix(y\_test, y\_pred\_dtree, labels=[1,0]) df\_cm = pd.DataFrame(cm, index = [i for i in ["1","0"]],

columns = [i for i in ["Predict 1", "Predict 0"]])

plt.figure(figsize = (7,5)) sns.heatmap(df\_cm, annot=True, fmt='g')

y\_pred\_proba = dTree.predict\_proba(X\_test)[::,1] fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_proba) fpr

tpr

### Support vector machine

def svm\_grid\_search(X, y):

#create a dictionary of all values we want to test

param\_grid = {'C': [0.1,1, 10, 100], 'gamma': [1,0.1,0.01,0.001, 0.4, 0.2,

0.8],'kernel': ['rbf', 'poly', 'sigmoid']}

cv = RepeatedStratifiedKFold(n\_splits=10, n\_repeats=3, random\_state=1) svm = SVC()

#use gridsearch to test all values

svm\_gscv = RandomizedSearchCV(estimator = svm,

param\_distributions = param\_grid, scoring = 'accuracy',

cv = cv, n\_jobs = -1)

#fit model to data svm\_gscv.fit(X, y)

return svm\_gscv.best\_params\_ svm\_grid\_search(X\_train, y\_train)

from sklearn import svm

svm = SVC(gamma=0.8, C=10, kernel='rbf', probability=True) svm.fit(X\_train, y\_train)

y\_pred\_svm = svm.predict(X\_test)

print(svm.score(X\_train, y\_train)) print(svm.score(X\_test, y\_test))

print(metrics.classification\_report(y\_test, y\_pred\_svm)) svm\_tacc = svm.score(X\_test, y\_test)

svm\_train\_acc = svm.score(X\_train, y\_train)

y\_pred\_proba = svm.predict\_proba(X\_test)[::,1]

fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_proba) fpr

tpr

auc = metrics.roc\_auc\_score(y\_test, y\_pred\_proba)

### Random forest classifier

def rf\_grid\_search(X, y): param\_grid = {

'n\_estimators': [5,10,20,40,50,60,70,80,100],

'max\_features': ['auto', 'sqrt', 'log2'],

'max\_depth' : [4,5,6,7,8],

'criterion' :['gini', 'entropy']

}

cv = RepeatedStratifiedKFold(n\_splits=10, n\_repeats=3, random\_state=1) # Random Forest model

rf = RandomForestClassifier() #use gridsearch to test all values

rf\_gscv = GridSearchCV(rf, param\_grid, cv=cv, n\_jobs=-1, scoring='accuracy')

#fit model to data rf\_gscv.fit(X, y)

return rf\_gscv.best\_params\_ rf\_grid\_search(X\_train, y\_train)

rfcl = RandomForestClassifier(n\_estimators=70, max\_features='sqrt', max\_depth=7, criterion='entropy')

rfcl = rfcl.fit(X\_train, y\_train) y\_pred\_rf = rfcl.predict(X\_test)

print(rfcl.score(X\_train,y\_train)) print(rfcl.score(X\_test,y\_test))

print(metrics.classification\_report(y\_test, y\_pred\_rf)) rf\_tacc = rfcl.score(X\_test,y\_test)

rf\_train\_acc = rfcl.score(X\_train, y\_train)

y\_pred\_proba = rfcl.predict\_proba(X\_test)[::,1]

fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_proba) fpr

tpr

auc = metrics.roc\_auc\_score(y\_test, y\_pred\_proba) auc

## Explanation

### Data Loading and Preprocessing

The dataset is loaded using pandas.

Missing values are handled by filling them with the mean of the respective columns. Features and target variables are separated, and the dataset is split into training and test sets. Features are standardized using StandardScaler.

### Model Training and Evaluation

Four models are trained: SVM, Decision Tree, Random Forest, and Logistic Regression. Each model is evaluated using classification reports and ROC-AUC scores.

Feature importance is calculated and displayed for Random Forest and Logistic Regression models.

### Visualization

A bar plot of feature importance for the Random Forest model is created using seaborn. This code provides a comprehensive workflow for identifying predictors of poor prognosis in CKD using various machine learning models. Ensure you have the necessary libraries installed in your Python environment before running the code.

**CHAPTER 6**

**Test Cases**

For a machine learning project predicting Chronic Kidney Disease (CKD) progression, test cases should cover different aspects of the system to ensure its functionality, reliability, and performance. Here are some key test cases:

**1. Data Ingestion and Preprocessing**

1.1. Data Ingestion

Test Case 1.1.1: Verify that the system correctly ingests data from CSV files.

Input: CSV file with valid data.

Expected Result: Data is successfully loaded into the system.

Test Case 1.1.2: Check the system’s ability to handle corrupted or malformed CSV files.

Input: CSV file with corrupted data.

Expected Result: System raises an appropriate error or exception.

1.2. Data Preprocessing

Test Case 1.2.1: Validate that missing values are correctly handled during preprocessing.

Input: Dataset with missing values in critical fields.

Expected Result: Missing values are either imputed or removed as per the defined strategy.

Test Case 1.2.2: Ensure feature normalization is applied correctly.

Input: Dataset with varying scales of numerical features.

Expected Result: Features are normalized to a consistent scale.

**2. Model Development and Training**

2.1. Model Training

Test Case 2.1.1: Verify that the model trains correctly on the provided dataset.

Input: Training dataset.

Expected Result: Model training completes without errors.

Test Case 2.1.2: Check if the model achieves acceptable performance on training data.

Input: Training dataset and predefined performance metrics.

Expected Result: Model meets or exceeds performance thresholds for metrics such as accuracy, precision, recall, and ROC-AUC.

2.2. Model Evaluation

Test Case 2.2.1: Validate that the model correctly evaluates test data.

Input: Test dataset with known outcomes.

Expected Result: Model performance metrics are correctly calculated and match expected results.

Test Case 2.2.2: Ensure the model handles imbalanced data appropriately.

Input: Test dataset with a class imbalance.

Expected Result: Model performance metrics (e.g., F1-score) account for class imbalance.

**3. Prediction and Inference**

3.1. Inference Engine

Test Case 3.1.1: Verify that the model generates predictions for new data.

Input: New patient data.

Expected Result: Model provides predictions and/or risk scores.

Test Case 3.1.2: Ensure predictions are consistent with the model’s expected behavior.

Input: Data with known outcomes.

Expected Result: Model predictions are accurate and align with known results.

**4. User Interface**

4.1. Frontend Functionality

Test Case 4.1.1: Validate that users can input data through the interface.

Input: Sample patient data entered into the UI.

Expected Result: Data is correctly submitted and processed.

Test Case 4.1.2: Check that the UI displays predictions and results correctly.

Input: Data that generates predictions.

Expected Result: Predictions and results are accurately reflected in the UI.

4.2. Usability

Test Case 4.2.1: Ensure that the UI is intuitive and user-friendly.

Input: User interactions with the UI.

Expected Result: Users can navigate and use the system without confusion.

**5. Monitoring and Maintenance**

5.1. System Performance

Test Case 5.1.1: Monitor system performance under normal load.

Input: Standard usage scenarios.

Expected Result: System maintains performance metrics (response time, accuracy).

Test Case 5.1.2: Validate system performance under high load conditions.

Input: Simulated high user traffic or data volume.

Expected Result: System performance remains acceptable, with no significant degradation.

5.2. Error Handling

Test Case 5.2.1: Ensure that the system correctly logs errors and exceptions.

Input: Invalid inputs or unexpected conditions.

Expected Result: Errors are logged and managed appropriately.

5.3. Data Security

Test Case 5.3.1: Verify data encryption and access control.

Input: Sensitive patient data.

Expected Result: Data is encrypted and access is restricted according to security policies.

**6. Compliance**

6.1. Regulatory Compliance

Test Case 6.1.1: Ensure compliance with data privacy regulations (e.g., HIPAA, GDPR).

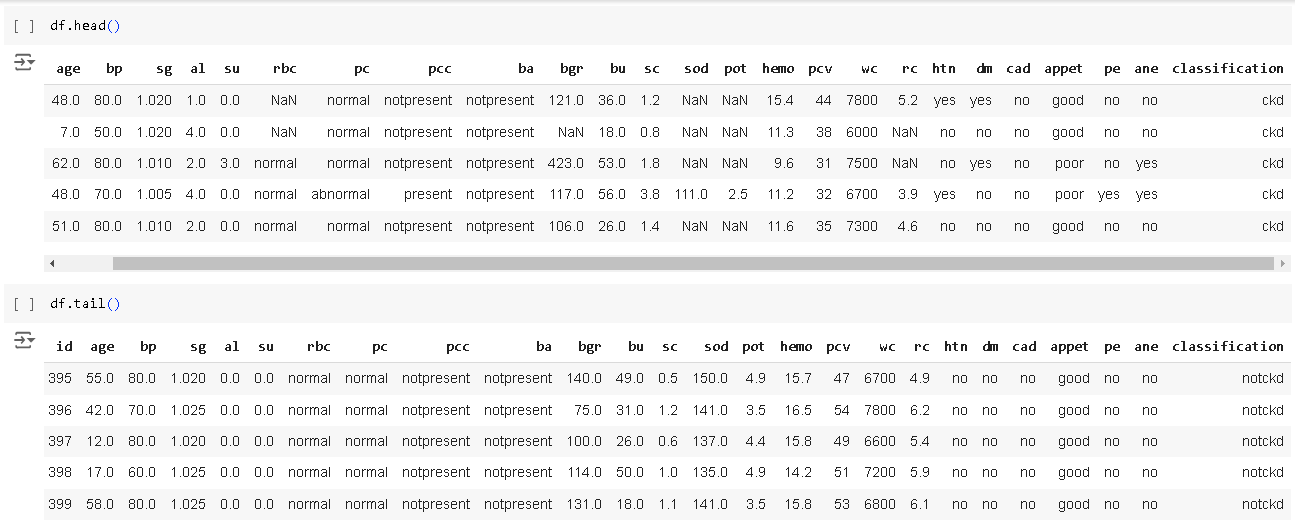
Input: Compliance check procedures.

Expected Result: System adheres to all relevant privacy and data protection regulations.

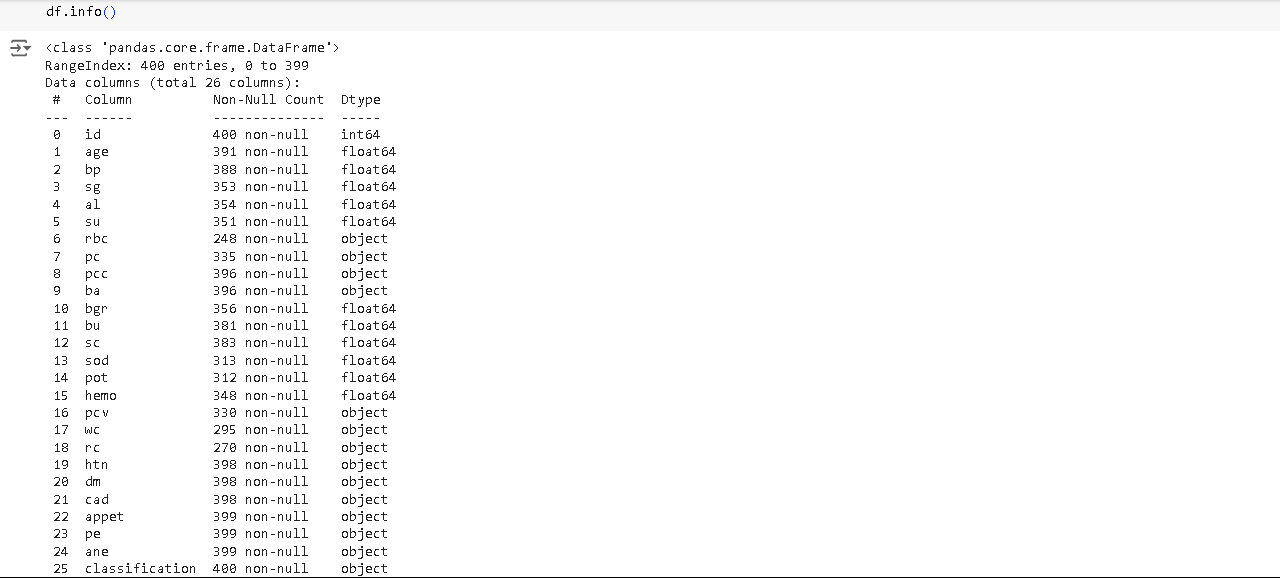
**Chapter 7**

# Results and Discussion

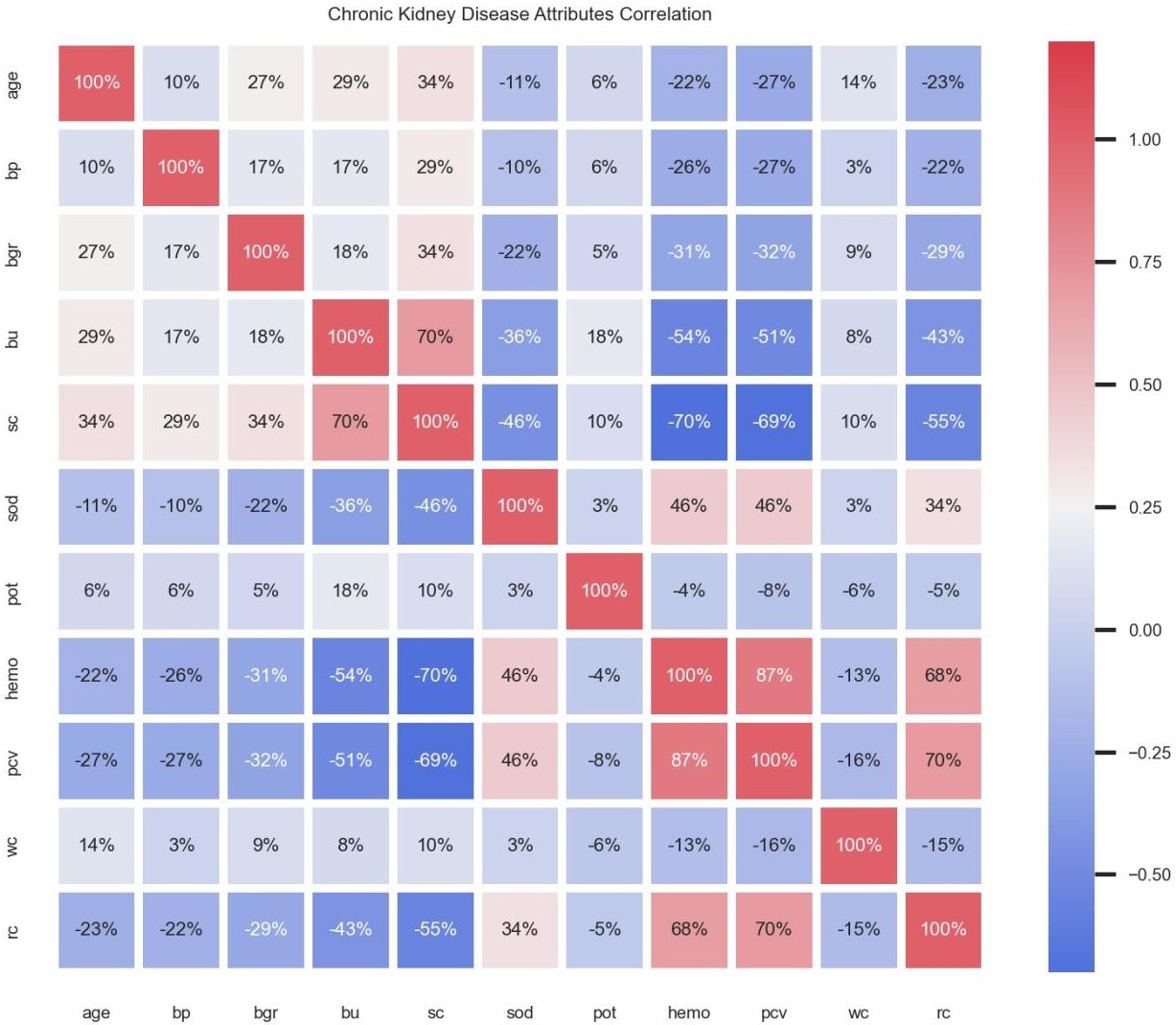
# 7.1 Snapshots



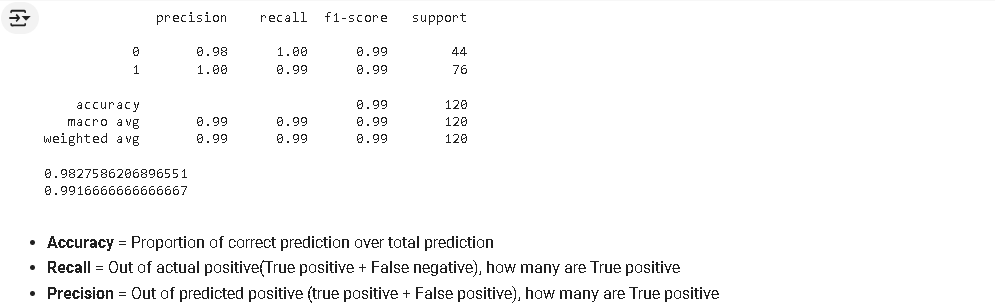
**Fig 7.1.1: Dataset**



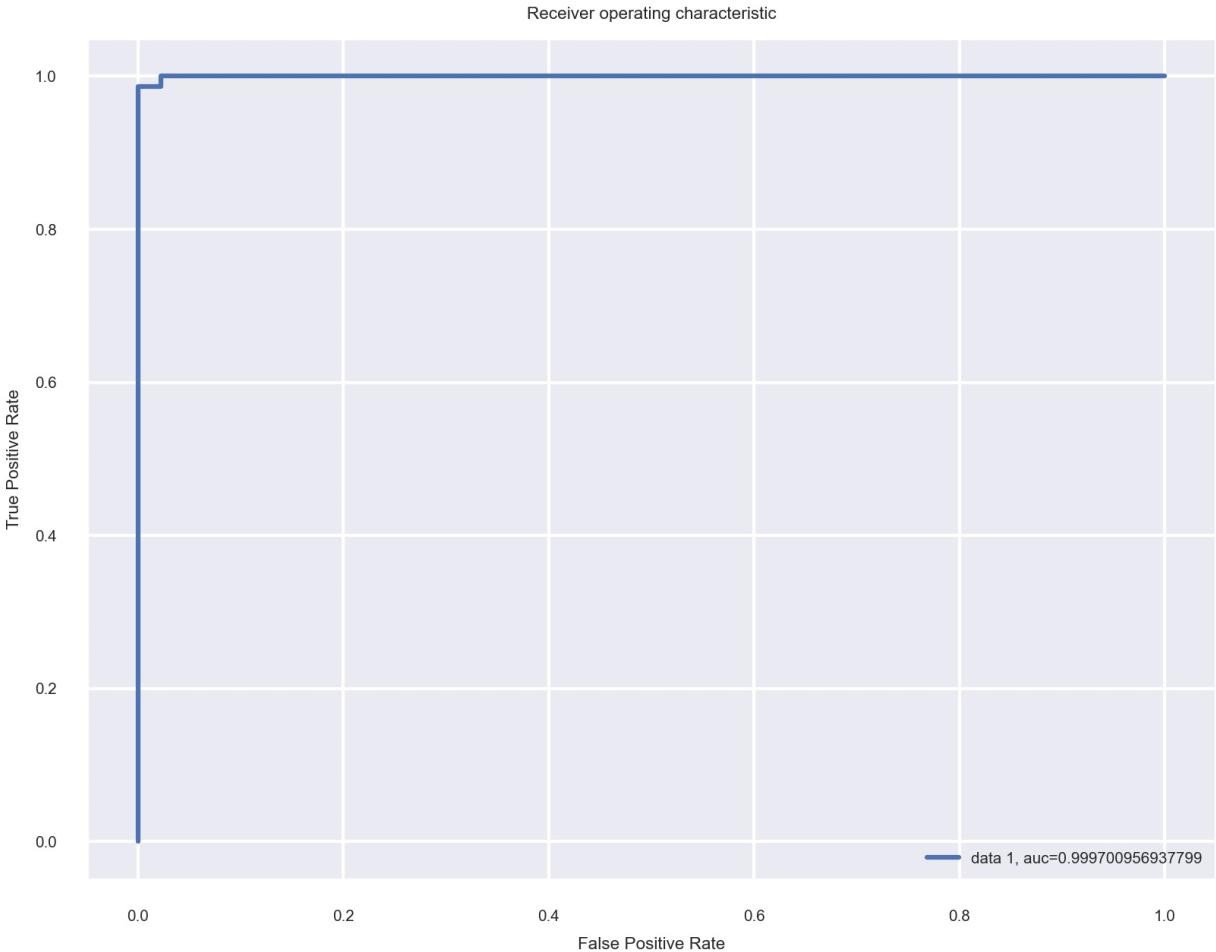
**Fig 7.1.2: Dataset information**



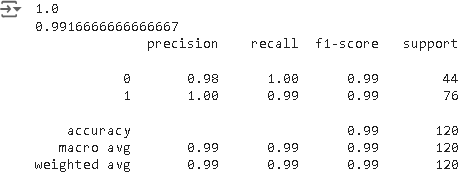
**Fig 7.1.3: Chronic Kidney Disease Attribute Correlation**



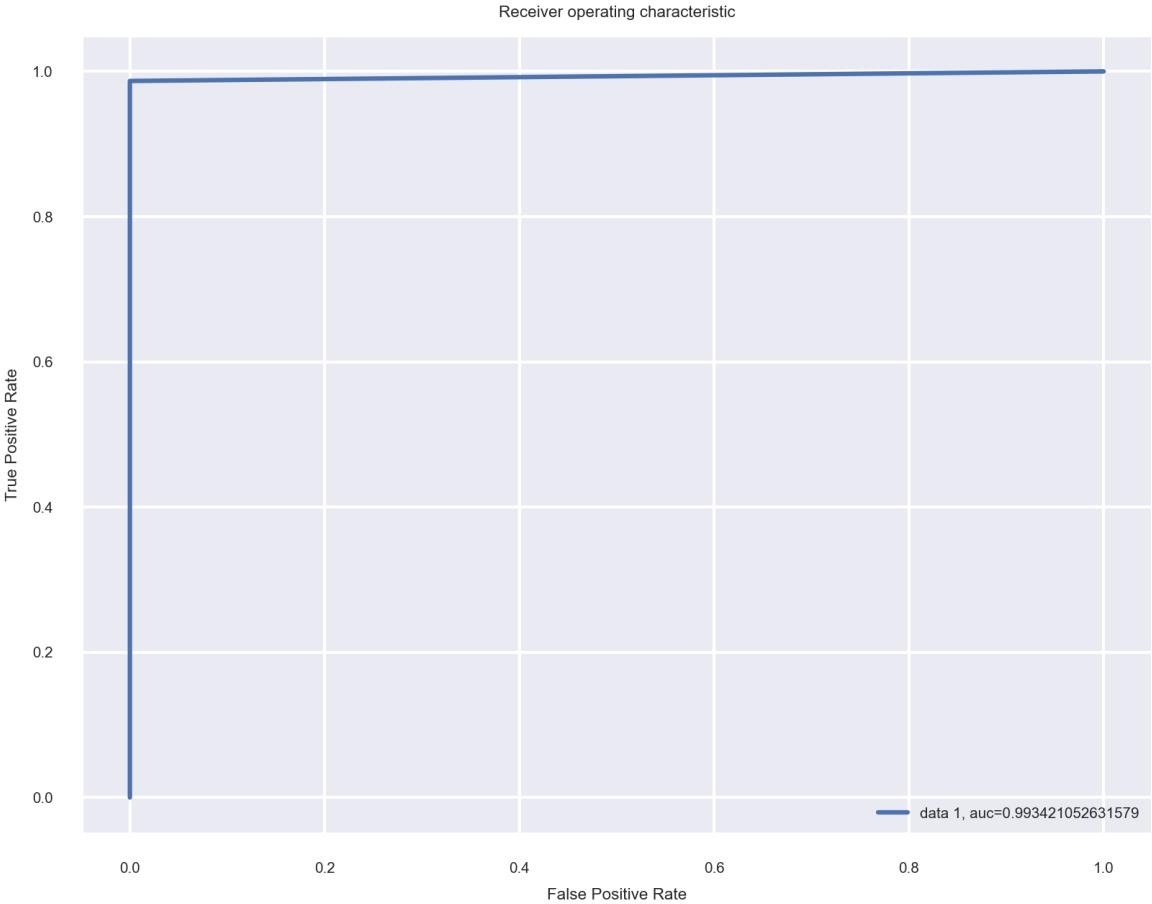
**Fig 7.1.4: Accuracy, Recall, Precision score for Logistic Regression**



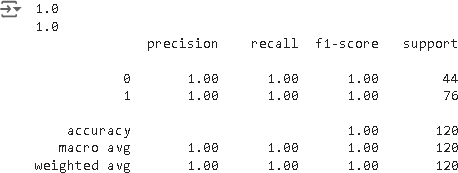
**Fig 7.1.5: True Positive and False Positive Rates for Logistic Regression Algorithm**



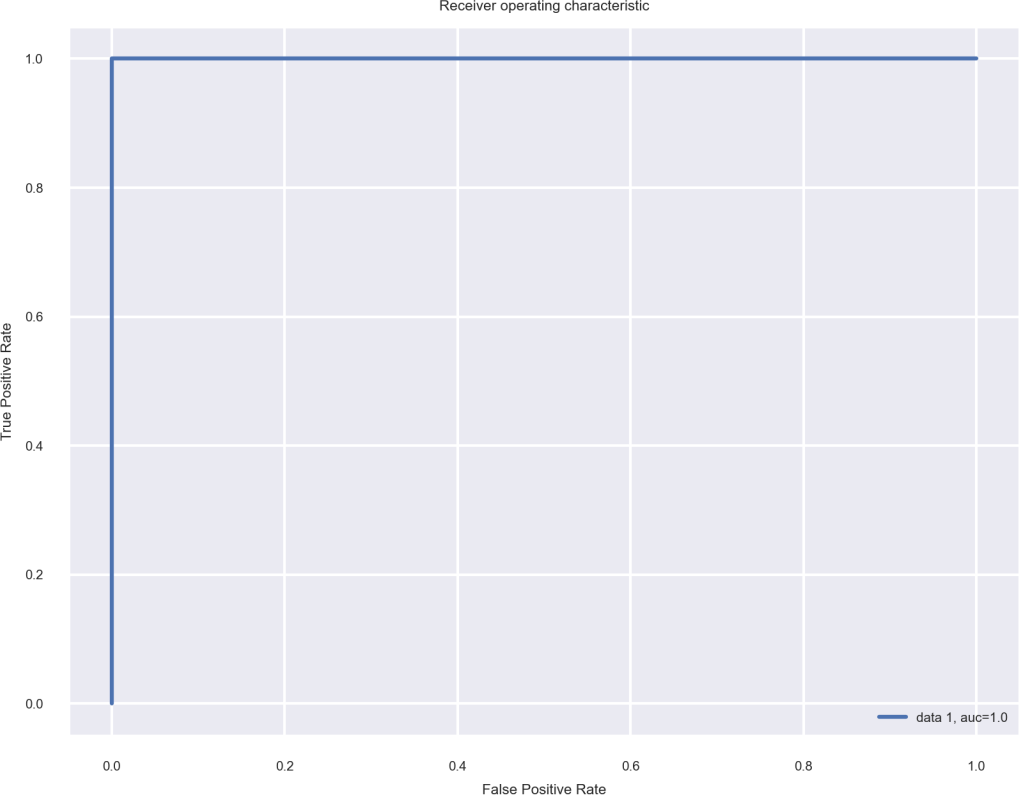
**Fig 7.1.6: Accuracy, Recall, Precision score for Decision tree algorithm**



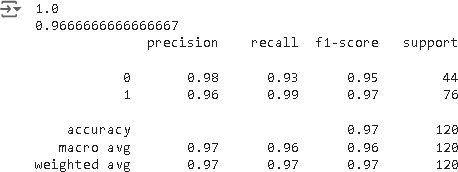
**Fig 7.1.7: True Positive and False Positive Rates for Decision Tree Algorithm**



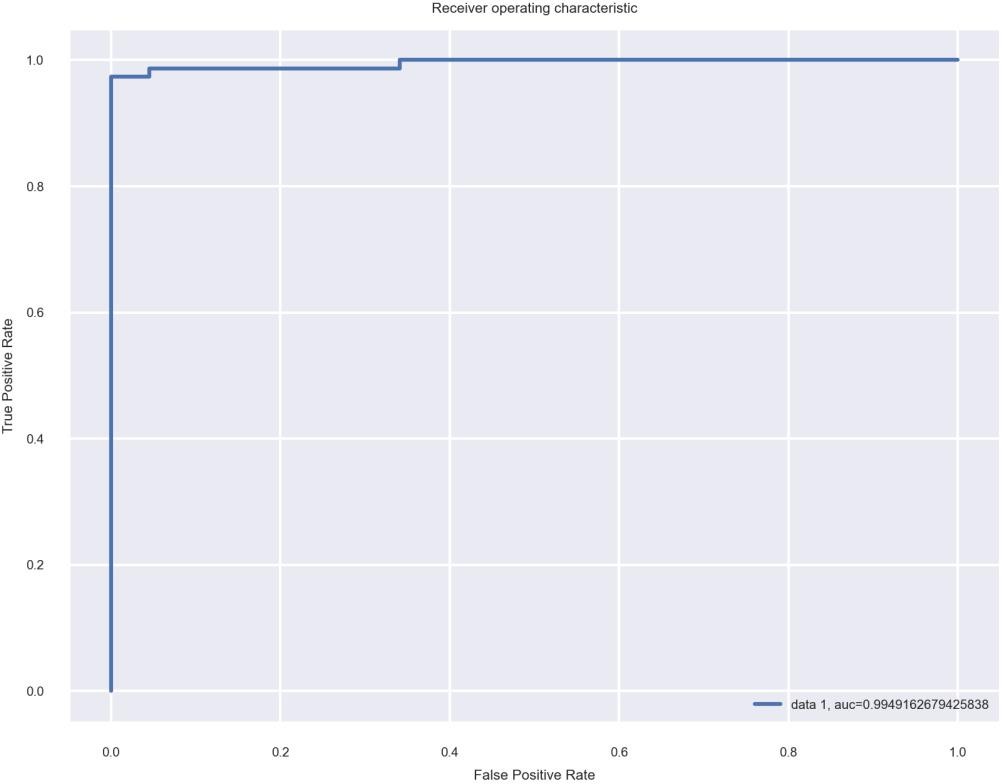
**Fig 7.1.8: Accuracy, Recall, Precision score for Random Forest Algorithm**



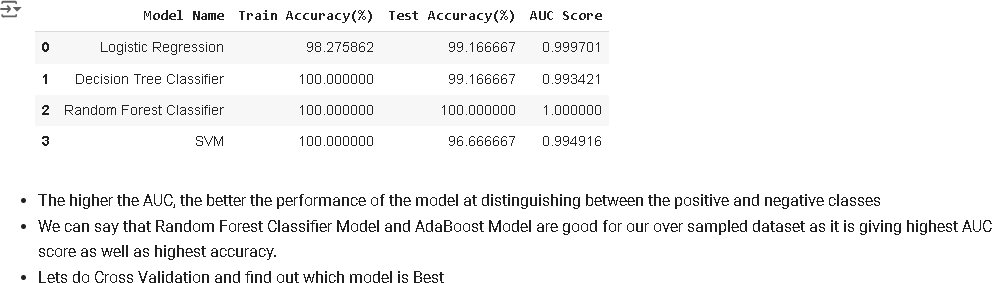
**Fig 7.1.8: True Positive and False Positive Rates for Random Forest Algorithm**



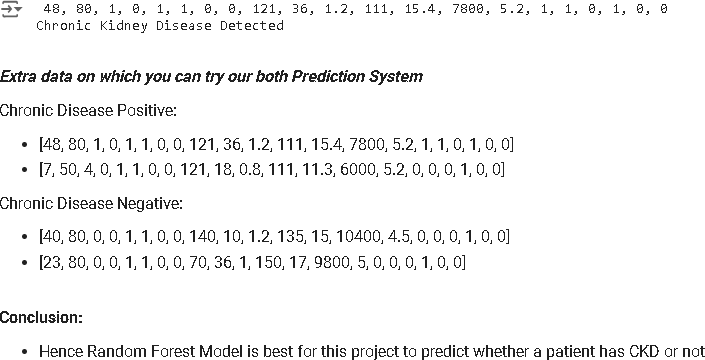
**Fig 7.1.9: Accuracy, Recall, Precision score for SVM Algorithm**



**Fig 7.1.10: True Positive and False Positive Rates for SVM Algorithm**

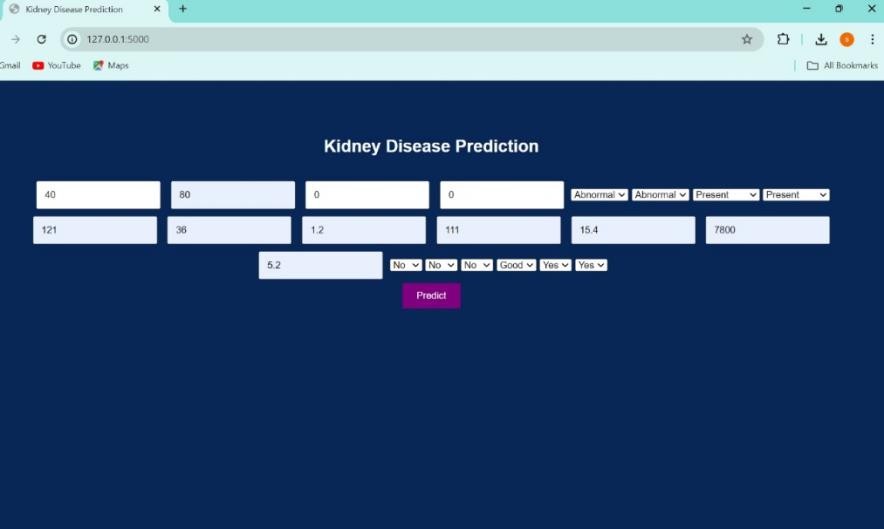


**Fig 7.1.11: Model Comparison**

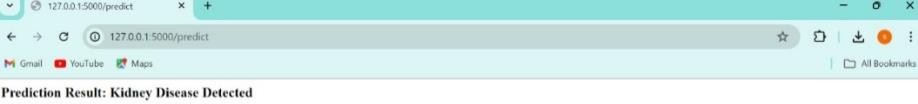


**Fig 7.1.12: Model Selection**

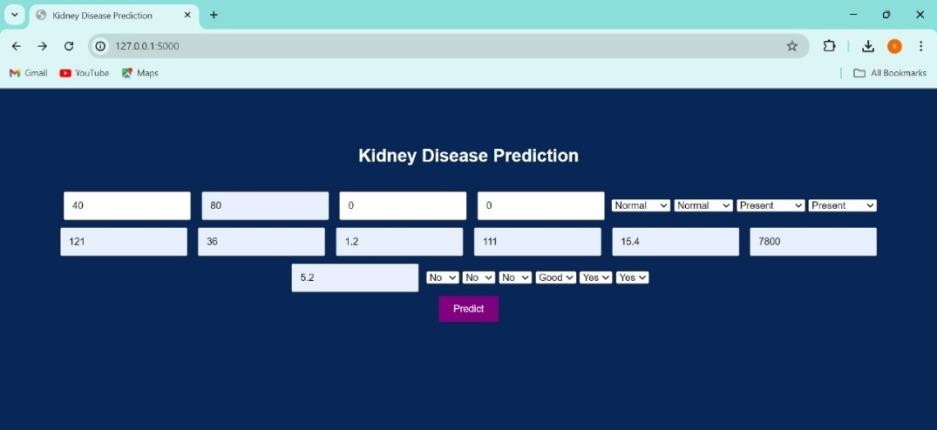
**7.2 Results**



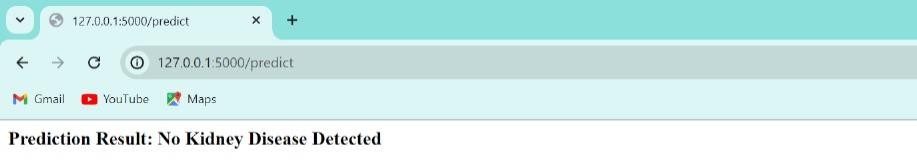
**Fig 7.2.1 Giving Inputs to the Model**



**Fig 7.2.2 Predicted Output (kidney disease detected)**



**Fig 7.2.3 Giving Inputs to the Model**



**Fig 7.2.4 Predicted Output (kidney disease detected)**

# Chapter 8

# Conclusion

This study highlights critical predictors of poor prognosis in chronic kidney disease (CKD), including reduced kidney function (eGFR), presence of albuminuria, uncontrolled hypertension, diabetes, advanced age, smoking, and comorbidities such as cardiovascular disease. These factors significantly impact disease progression and patient outcomes.

Effective management strategies targeting these predictors—such as early detection, optimal control of blood pressure and diabetes, smoking cessation, and tailored interventions based on individual risk profiles—are essential for improving prognosis in CKD patients. Addressing these factors comprehensively can help mitigate disease progression, reduce complications, and enhance quality of life for individuals with CKD.

Future research should continue to explore new biomarkers and therapeutic approaches that could further refine predictive models and enhance personalized care for CKD patients. By integrating these findings into clinical practice, healthcare providers can make meaningful strides towards improving outcomes and reducing the burden of chronic kidney disease.

# Chapter 9

# Future Enhancements

The findings of this study provide valuable insights into the predictors of poor prognosis in chronic kidney disease (CKD). Moving forward, several avenues for future enhancement and research could further advance our understanding and management of CKD:

1. Integration of Novel Biomarkers: Investigating and integrating novel biomarkers beyond eGFR and albuminuria could provide more precise indicators of kidney function and disease progression. Biomarkers related to inflammation, fibrosis, and oxidative stress may offer additional predictive value and aid in early detection of CKD complications.
2. Personalized Risk Stratification: Developing and implementing personalized risk stratification models based on a combination of clinical, genetic, and biomarker data could help tailor treatment strategies to individual patient profiles. Machine learning algorithms and artificial intelligence approaches may facilitate the identification of high-risk patients who would benefit most from intensive monitoring and early intervention.
3. Telemedicine and Remote Monitoring: Leveraging telemedicine and remote monitoring technologies could enhance patient care by facilitating frequent monitoring of kidney function and adherence to treatment plans outside of traditional clinical settings. This approach may improve patient outcomes by enabling timely adjustments to management strategies based on real-time data.
4. Lifestyle and Behavioral Interventions: Expanding interventions aimed at modifying lifestyle factors such as diet, physical activity, and smoking habits could complement medical treatments in slowing CKD progression. Behavioral interventions focusing on improving medication adherence and self-management

skills could also play a crucial role in optimizing outcomes.

1. Targeted Therapies and Precision Medicine: Advancing research into targeted therapies, including novel pharmacological agents and biologic treatments, tailored to specific molecular pathways implicated in CKD progression could potentially halt or reverse kidney damage. Precision medicine approaches based on genetic profiling may guide personalized treatment selection and optimize therapeutic efficacy.
2. Addressing Socioeconomic Determinants: Recognizing and addressing socioeconomic determinants of health—such as access to healthcare, education, and socioeconomic status—can help reduce disparities in CKD outcomes. Policy initiatives aimed at improving healthcare access and affordability for vulnerable populations may contribute to more equitable health outcomes.
3. Longitudinal Studies and Real-World Evidence: Conducting robust longitudinal studies and utilizing real-world evidence to validate findings from controlled clinical trials will be crucial in translating research discoveries into clinical practice. Long- term follow-up of CKD cohorts can provide insights into the long-term effectiveness and safety of interventions.

In conclusion, future enhancements in the management of chronic kidney disease should focus on advancing personalized medicine approaches, leveraging technology for remote monitoring and interventions, exploring novel biomarkers and therapies, addressing socioeconomic determinants of health, and prioritizing patient-centered care. These efforts have the potential to transform CKD management, improve patient outcomes, and reduce the global burden of kidney disease.

**References**

* + <https://openai.com/>
  + <https://ieeexplore.ieee.org/>
  + <https://www.kaggle.com/>
  + <https://colab.research.google.com/>
  + <https://ieeexplore.ieee.org/document/10100868/>
  + <https://www.irjet.net/archives/V9/i7/IRJET-V9I7123.pdf>
  + <https://link.springer.com/chapter/10.1007/978-3-031-55048-5_10>