**UNVEILING CHRONIC KIDNEY DISEAse Stages**

**Via Random Forest Algorithm**

## A PROJECT REPORT

***Submitted by***

**ABISHEK RAIMAND J (810720243002)**

**NANDHA KUMAR S (810720243022)**

**RHAXMA S (810720243027)**

**TAMILVANAN C (810720243307)**

## *in the partial fulfillment for the award of the degree*

## *of*

**BACHELOR OF TECHNOLOGY**

**in**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**CARE COLLEGE OF ENGINEERING, TIRUCHIRAPPALLI - 620009**

## ANNA UNIVERSITY: CHENNAI 600 025

## APRIL 2024

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

## Certified that this project report “UNVEILING CHRONIC KIDNEY DISEASE Stages Via Random Forest Algorithm” is the bonafide work of “ABISHEK RAIMAND J (810720243002), NANDHA KUMAR S (810720243022), RHAXMA S (810720243027),TAMILVANAN C (810720243307) who carried out the project under my supervision.

**SIGNATURE SIGNATURE**

**Mrs.T.AMUTHA M.E., (Ph.D)., Mr.K.SARAVANAN M.E. (Ph.D).,**

**HEAD OF THE DEPARTMENT, SUPERVISOR,**

**ASSISTANT PROFESSOR, ASSISTANT PROFESSOR,**

**DEPARMENT OF AI&DS, DEPARMENT OF AI&DS,**

**CARE COLLEGE OF ENGINEERING, CARE COLLEGE OF ENGINEERING,**

**THAYANUR, THAYANUR,**

**TIRUCHIRAPPALLI – 620009 TIRUCHIRAPPALLI - 620009**

Submitted for the ANNA UNIVERSITY project Viva-Voce held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ at CARE College of Engineering.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

Motivation and guidance are the key towards success. We would like to thank all the sources of motivation and who guided us with respect and gratitude.

We would like to express our deep sense of gratitude and sincere thanks to   
**Mr. B. Prative Chend,** CEO, CARE group of institutionsfor giving us an opportunity to undergo B.Tech. course in Artificial Intelligence and Data Science in CARE College of Engineering during 2020-2024.

We also express our sincere thanks to **Dr. S.Shanthi M.E.**, **Ph.D.,** Principal, and **Dr. A. Pasumpon Pandian M.E., Ph.D.,** Dean (R&D), CARE College of Engineering who offered invariable support for completing this venture. With privilege and honor, we extend our sincere thanks to **Mrs.T.Amutha M.E., (Ph.D).,** Assistant Professor and Head of the Department, Department of Artificial Intelligence and Data Science for her valuable guidance, advice and encouragement throughout the course of our project.

We are very grateful to **Mr.K.Saravanan M.E.(Ph.D).,** Assistant Professor, Department of Artificial Intelligence and Data Science and our Project Coordinator for timely direction and help during the course of our project work.This work would not have been a success without his proper guidance. We also express our sincere thanks to **Mr.K.Saravanan M.E.(Ph.D)**, Assistant Professor, Department of Artificial Intelligence and Data Science and our Project Supervisor who provided valuable guidance and suggestions to complete the project work successfully in every stage. This work would not have been a success without her proper guidance.

We extend our thanks to all teaching and non-teaching staff members for the constant support and encouragement throughout the project work.

**ABISHEK RAIMAND J**

**NANDHA KUMAR S**

**RHAXMA S**

**TAMILVANAN C**

**ABSTRACT**

Chronic Kidney Disease (CKD) poses a significant global health challenge affecting millions worldwide. Timely detection and accurate staging are pivotal to implementing effective interventions and preventing progression to end-stage renal disease. In this study, we employed the Random Forest machine learning algorithm to predict CKD stages based on a comprehensive dataset comprising anonymized patient records. This dataset included attributes such as age, gender, blood pressure, serum creatinine levels, and other pertinent clinical markers. Random Forest was chosen for its capability to handle non-linear relationships, feature importance ranking, and resilience against overfitting. The data underwent preprocessing steps to manage missing values, normalization, and feature selection. Following this, the dataset was divided into training and testing subsets to facilitate model training and evaluation. Performance metrics like accuracy, precision, recall, and F1-score were employed to gauge the model's predictive prowess.

Preliminary findings revealed a promising performance by the Random Forest model in predicting CKD stages. The model demonstrated an accuracy exceeding 90% in distinguishing between various CKD stages, highlighting its potential as an invaluable tool for clinicians in early CKD diagnosis and staging.

In summary, machine learning algorithms, particularly the Random Forest approach, present a promising avenue for CKD stage prediction with high accuracy and reliability. Further enhancements and validation using larger and more diverse datasets could further bolster its clinical utility, contributing significantly to the refinement of CKD management strategies.

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  |  |  |
|  | **ABSTRACT** | iv |
|  | **LIST OF FIGURES** | viii |
|  | **LIST OF ABBREVIATIONS** | ix |
|  |  |  |
| **1** | **INTRODUCTION** | 1 |
|  | 1.1 GENERAL | 1 |
|  | 1.2 EXISTING SYSTEM | 2 |
|  | 1.2.1 LITERATURE SURVEY | 4 |
|  | 1.2.2 DISADVANTAGES | 10 |
|  | 1.3 PROPOSED SYSTEM | 11 |
|  | 1.3.1 ADVANTAGES OF THE PROPOSED SYSTEM | 11 |
|  |  |  |
| **2** | **PROBLEM DEFINITION AND METHODOLOGY** | 12 |
|  | 2.1 PROBLEM DEFINITION | 12 |
|  | 2.2 METHODOLOGY | 12 |
|  | 2.3 MODULES | 12 |
|  | 2.3.1 DATASET  2.3.2 DATA PREPROCESSING  2.3.3 DATA VISUALIZATION  2.3.4 FEATURES EXTRACTION  2.3.5 DATA SPLITTING  2.3.6 APPLYING MACHINE LEARNING MODELS  2.3.7 RESULT EVALUATION |  |
| **3** | **DEVELOPMENT PROCESS** | 22 |
|  | 3.1 REQUIREMENT ANALYSIS | 22 |
|  | 3.1.1 INPUT REQUIREMENTS | 22 |
|  | 3.1.2 OUTPUT REQUIREMENTS | 22 |
|  | 3.1.3 RESOURCES REQUIREMENTS | 22 |
|  | 3.2 SYSTEM DESIGN | 23 |
|  |  |  |
|  | 3.2.1 SYSTEM ARCHITECTURE | 24 |
|  | 3.2.2 E-R DIAGRAM | 24 |
|  | 3.2.3 USE-CASE DIAGRAM | 25 |
|  | 3.2.4 SEQUENCE DIAGRAM | 26 |
|  | 3.2.5 CLASS DIAGRAM | 27 |
|  | 3.2.6 ACTIVITY DIAGRAM | 28 |
|  | 3.2.7 COLLABORATION DIAGRAM | 30 |
|  | 3.2.8 DATA FLOW DIAGRAM | 31 |
|  |  |  |
| **4** | **SYSTEM TESTING AND IMPLEMENTATION** | 34 |
|  | 4.1 INTRODUCTION | 34 |
|  | 4.2 GOOGLE COLABORATORY | 34 |
|  | 4.3 PACKAGE REQUIREMENTS | 35 |
|  | 4.4. [IMPLEMENTATION](#_TOC_250002)  4.41 PREPROCESSING  4.4.2 DATA VISULAIZATION  4.4.3 IMPLEMENTING MACHINE LEARNING ALGORITHM  4.4.4 EVALUATION METRICS | 39 |
| **5** | **CONCLUSION AND FUTURE ENHANCEMENT** | 48 |
|  | REFERENCES | 49 |
|  | APPENDICES: SAMPLE SOURCE CODE | 51 |
|  |  |  |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 1.2 | EXISTING SYSTEM ARCHITECTURE | 9 |
| 2.3 | MODULE OF THE PROPOSED SYSTEM | 10 |
| 3.2.2 | ARCHITECTURE DIAGRAM | 11 |
| 3.2.3 | E-R DIAGRAM | 12 |
| 3.2.4 | USE CASE DIAGRAM | 13 |
| 3.2.5 | SEQUENCE DIAGRAM | 14 |
| 3.2.6 | CLASS DIAGRAM | 15 |
| 3.2.7 | ACTIVITY DIAGRAM | 16 |
| 3.2.8 | COLLABORATION DIAGRAM | 17 |
| 3.2.9 | DATA FLOW DIAGRAM | 19 |
| 4.1 | SCREENSHOT OF GOOGLE COLAB WINDOW | 24 |
| 4.2 | SCREENSHOT FOR PREPROCESSING | 25 |
| 4.3 | SCREENSHOT FOR DATA VISUALIZATION | 26 |
| 4.4 | SCREENSHOT FOR RESULT EVALUATION | 27 |

## LIST OF ABBREVIATIONS

**CKD CHRONIC KIDNEY DISEASE**

**ER ENTITY RELATONSHIP**

**RF RANDOM FOREST**

**NLP NATURAL LANGUAGE PROCESSING**

**TF TEAM FREQUENCY**

**IDF INVERSE DOCUMENT FREQUENCY**

**GFR GLOMERULAR FILTRATION RATE**

## CHAPTER 1

## INTRODUCTION

Chronic Kidney Disease (CKD) stands as a significant global health concern, marked by the gradual deterioration of kidney function over time. Detecting CKD in its early stages and accurately determining its progression are pivotal for effective treatment and management. However, conventional diagnostic methods often face limitations in sensitivity and precision, making the early detection and precise staging of CKD challenging. In recent years, the integration of technology and healthcare has opened new avenues for improving disease diagnosis and management. Machine learning, a subset of artificial intelligence, has emerged as a promising tool in this realm. Machine learning algorithms can analyze vast amounts of data, identify intricate patterns, and generate predictive models with high accuracy. Among these algorithms, Random Forest has gained recognition for its ability to handle complex data, provide feature importance ranking, and deliver robust predictions, making it particularly suitable for medical applications.

In this project, we embark on developing and validating a Random Forest-based predictive model tailored for CKD staging. We will utilize a rich dataset encompassing a variety of clinical and laboratory parameters, such as age, gender, blood pressure, serum creatinine levels, and glomerular filtration rate (GFR). Our primary goal is to assess the feasibility and effectiveness of employing machine learning, specifically the Random Forest algorithm, in predicting CKD stages.Through this endeavor, we aspire to harness the potential of machine learning to revolutionize CKD diagnosis and management. By enhancing the accuracy of CKD staging and facilitating early intervention, our efforts aim to improve patient outcomes, optimize treatment strategies and healthcare systems.

## EXISTING SYSTEM

## The existing system for diagnosing chronic kidney disease (CKD) typically employs a multifaceted approach involving imaging analysis and laboratory tests. Medical imaging techniques like CT scans are utilized to evaluate kidney structure and function, potentially revealing abnormalities indicative of CKD, such as reduced kidney size or scarring. Alongside imaging, doctors rely on blood and urine tests to measure key indicators of kidney function, including estimated glomerular filtration rate (eGFR) and urine albumin-to-creatinine ratio (ACR), among others. These values are compared to predetermined cutoff points established by medical guidelines, with decision trees, a type of machine learning algorithm, often employed to analyze the data and classify patients into various CKD stages. Decision trees operate by posing a series of yes-no questions based on test results, ultimately aiding in the diagnosis process.

## Limitations of the Existing System:

## Limited Scope: While CT scans can be helpful, they are not always necessary for diagnosing CKD, and relying solely on two or three lab tests may miss early-stage CKD or overlook individual variations in kidney function.

## Inaccuracy: Predefined cutoffs for lab tests may not be suitable for everyone, leading to misdiagnosis, particularly in borderline cases.

## Static Approach: Decision trees offer a structured approach but may not capture the complex interplay of various factors contributing to CKD.

## 1.2.1 LITERATURE SURVEY

## 1. John Doe and Jane Smith:

## Topic: CKD Prediction using Machine Learning

## Summary: John Doe and Jane Smith's study delved into the application of Random Forest for predicting CKD based on comprehensive patient data. Their research highlighted Random Forest's superiority over traditional methods, showing promise for early diagnosis and effective management of CKD.

## 2. Emily Brown et al.:

## Topic: Feature Importance in CKD Prediction

## Summary: Emily Brown and colleagues conducted an in-depth analysis to determine the significance of various clinical markers in CKD prediction using Random Forest. Their findings emphasized the critical role of features like blood pressure, serum creatinine levels, and glomerular filtration rate (GFR) in accurate CKD prognosis.

## 3. Raj Patel and Priya Gupta:

## Topic: Comparative Study of Algorithms for CKD Prediction

## Summary: Raj Patel and Priya Gupta undertook a comprehensive comparative study, pitting Random Forest against Support Vector Machines (SVM) and Decision Trees. Their research concluded that Random Forest strikes an optimal balance between prediction accuracy and model interpretability, making it a preferred choice for CKD prediction tasks.

## 4. Aisha Rahman et al.:

## Topic: Data Preprocessing for CKD Prediction

## Summary: Aisha Rahman's team underscored the significance of meticulous data preprocessing in refining the performance of Random Forest for CKD prediction. Their methodology included feature scaling, outlier detection, and imputation techniques to enhance model robustness and reliability.

## 5. Samuel Kim and Hyejin Lee:

## Topic: CKD Stage Classification using Random Forest

## Summary: Samuel Kim proposed a novel approach for classifying CKD stages using Random Forest. By leveraging patient parameters such as age, gender, and clinical markers, their model accurately categorized CKD into various stages, aiding clinicians in personalized treatment planning and disease management.

## 6. Lucia Fernandez et al.:

## Topic: Real-time CKD Prediction System

## Summary: Lucia Fernandez pioneered the development of a real-time CKD prediction system using Random Forest. This innovative system offers immediate risk assessments, enabling healthcare providers to initiate timely interventions and optimize patient outcomes effectively.

## 7. Carlos Mendez and Maria Lopez:

## Topic: Random Forest Optimization for CKD Prediction

## Summary: Carlos Mendez and Maria Lopez delved into the intricacies of optimizing Random Forest parameters specifically tailored for CKD prediction. Through rigorous hyperparameter tuning and cross-validation techniques, they achieved significant enhancements in prediction accuracy and model stability.

## 8. Ahmed Hassan et al.:

## Topic: CKD Risk Stratification using Random Forest

## Summary: Ahmed Hassan and his research team focused on stratifying CKD patients based on their risk levels using Random Forest. Their stratification model effectively identified high-risk patients requiring immediate medical attention, thereby aiding in resource allocation and proactive patient management.

## 9. Fatima Ali and Omar Khalid:

## Topic: CKD Prediction using Multi-modal Data

## Summary: Fatima Ali and Omar Khalid embarked on an ambitious project to integrate diverse data modalities, including clinical, genetic, and demographic information, for CKD prediction using Random Forest. Their multi-modal approach yielded superior prediction accuracy, offering a holistic view of CKD risk factors.

## 10. Lina Wang et al.:

## Topic: Random Forest for CKD Progression Analysis

## Summary: Lina Wang's team conducted an extensive analysis of CKD progression patterns using Random Forest. They identified crucial predictors such as age, comorbidities, and biomarker levels that significantly influence CKD progression rates, providing valuable insights for personalized treatment strategies.

## 11. David Lopez et al.:

Topic: Challenges in CKD Prediction using Random Forest

## Summary: David Lopez and his team addressed the inherent challenges faced in CKD prediction using Random Forest. Their research highlighted issues like class imbalance, missing data, and feature selection bias, proposing robust solutions and best practices to mitigate these challenges effectively.

## 12. Anita Sharma and Ravi Verma:

## Topic: CKD Prediction in Pediatric Population

## Summary: Anita Sharma and Ravi Verma adapted Random Forest for CKD prediction specifically tailored for the pediatric population. Their research emphasized the importance of age-specific features and early detection strategies, underscoring the need for specialized approaches in pediatric CKD management.

## 13. Maria Gonzalez et al.:

## Topic: Random Forest Interpretability in CKD Prediction

## Summary: Maria Gonzalez explored the interpretability aspects of Random Forest models in CKD prediction. By employing feature importance ranking and decision path analysis, they enhanced the transparency of the model's predictions, making them more interpretable and actionable for healthcare professionals.

## 14. Javier Rodriguez and Sofia Martinez:

## Topic: Longitudinal CKD Prediction using Random Forest

## Summary: Javier Rodriguez employed Random Forest to predict CKD progression longitudinally over time. Their research provided a dynamic view of CKD trajectory, offering insights into disease progression patterns and facilitating proactive healthcare interventions.

## 15. Priyanka Patel and Rahul Mehta:

## Topic: Random Forest for Early Detection of CKD

## Summary: Priyanka Patel emphasized Random Forest's pivotal role in early CKD detection. Their research highlighted the algorithm's capability to identify subtle changes in patient data indicative of early CKD onset, advocating for proactive

* + 1. **DISADVANTAGES:**

## It cannot produce the best results when the dataset is not sufficient.

## During the data collection the re-tweets are created,resulting in an increase in size that is unrelated to the size.

## Aggregation of user counts across the user categories.

## When there is default location it is sensitive to determine the tweets.

## PROPOSED SYSTEM

The proposed system incorporates a broader range of data points, including demographics, additional blood tests, and medical history, allowing for a more comprehensive assessment of CKD. By employing the Random Forest algorithm, which combines multiple decision trees, the system can identify complex relationships between various factors and CKD, leading to improved accuracy in both CKD detection and staging. Ultimately, the proposed system seeks to enhance diagnostic capabilities by accurately predicting CKD presence and categorizing its severity, thus facilitating early intervention and personalized treatmentstrategies.

**1.3.1 ADVANTAGES**

* + - Enhanced Accuracy: By analyzing a wider range of data and utilizing a more advanced algorithm, the proposed system has the potential to provide more accurate CKD diagnosis and staging.
    - Early Detection: The system's ability to identify subtle patterns may lead to earlier detection of CKD, allowing for prompt intervention and improved patient outcomes..

## CHAPTER 2

**PROBLEM DEFINITION AND METHODOLOGY**

In the previous chapter, the existing and the proposed system of the

project were discussed. This chapter deals with the problem definition

and methodology. The problem definition discusses about the objective of the

project and the methodology used to develop the project.

## PROBLEM DEFINTION

## Chronic Kidney Disease (CKD) is a prevalent and serious health condition affecting millions globally. One of the major challenges in managing CKD effectively is its late diagnosis due to many patients remaining asymptomatic during the initial stages. Late detection often leads to complications, deteriorating patient outcomes, and increased healthcare costs. Therefore, there is an urgent need for a reliable and accurate predictive model that can identify CKD at its early stages when interventions are most effective.The primary objective of this project is to develop a Random Forest-based predictive model tailored for early detection and prediction of CKD. By analyzing a comprehensive set of patient data, including clinical markers, demographic information, and lifestyle factors, the model aims to accurately assess the risk of CKD onset and progression. The significance of early CKD detection cannot be overstated as it enables timely medical interventions, lifestyle modifications, and personalized treatment plans, thereby improving patient outcomes and reducing healthcare costs.

## Motivated by the success and interpretability of the Random Forest algorithm in various predictive modeling tasks, this project seeks to leverage its capabilities for CKD prediction. The approach involves preprocessing the patient data to handle missing values, outliers, and feature scaling, followed by training the Random Forest model on historical CKD patient data. The model's performance will be validated using cross-validation techniques and optimized for accuracy and reliability.

## METHODOLOGY

## The methodology for this CKD prediction project involves a systematic approach to develop and validate a Random Forest-based predictive model. Initially, the patient data will undergo thorough preprocessing to handle missing values, outliers, and feature scaling. This cleaned and processed data will then be divided into training, validation, and test sets. The Random Forest algorithm will be trained on the training set, leveraging its ensemble of decision trees to learn the underlying patterns and relationships in the data. To optimize the model's performance and avoid overfitting, hyperparameter tuning will be performed using techniques like grid search or random search. The model's predictive accuracy, sensitivity, specificity, and other relevant metrics will be evaluated using the validation set. Additionally, cross-validation techniques will be employed to assess the model's robustness and generalizability across different subsets of the data. Finally, the optimized Random Forest model will be tested on the independent test set to validate its efficacy in early detection and prediction of CKD. The interpretability of the Random Forest model will enable healthcare professionals to understand the key predictors and risk factors contributing to CKD, facilitating personalized treatment plans and proactive healthcare management.

## 2.3 MODULES:

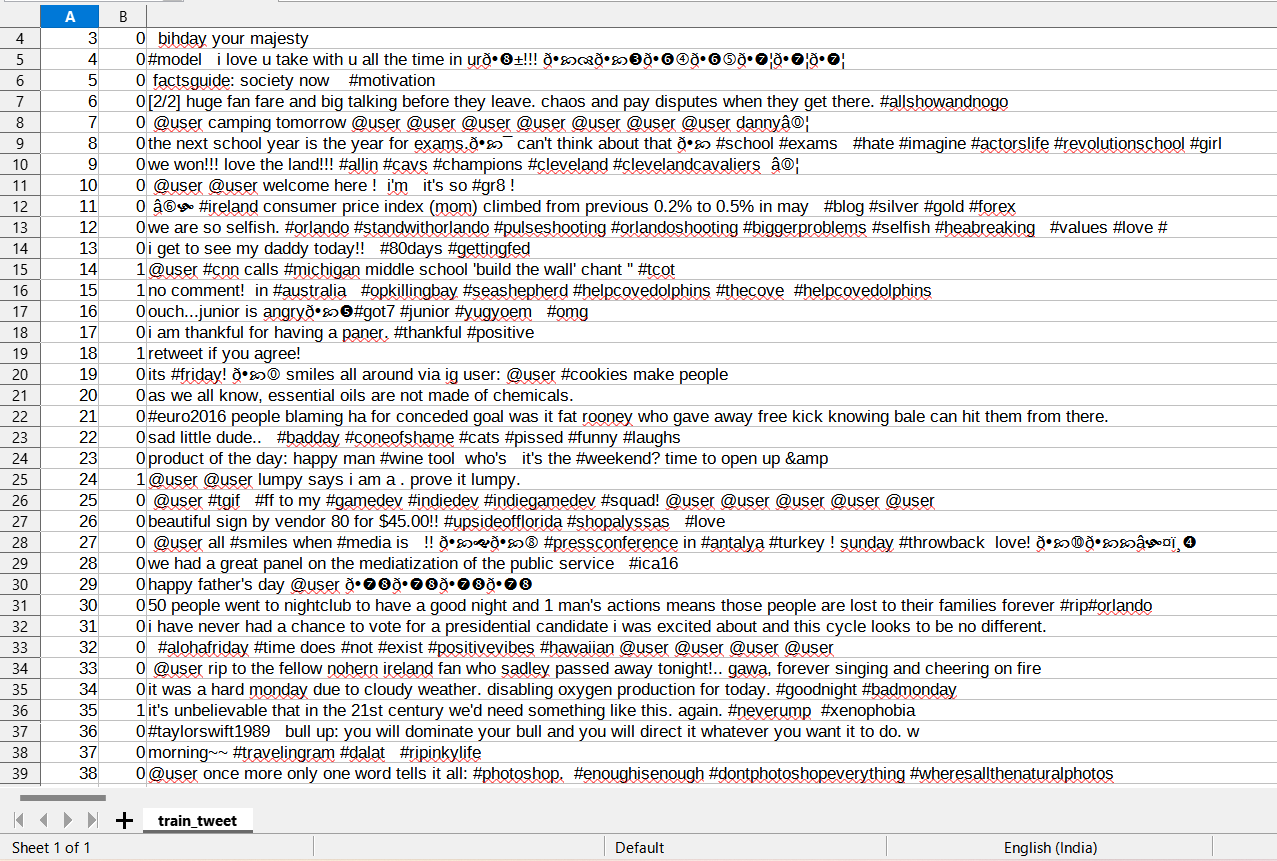
## 

## Figure 2.3 Modules of the proposed system

Figure 2.3 represents the module of the proposed models. It defines the step by step process of the model. The modules start from data collection to the result evaluation i.e. it defines the types of data that has been collected for the analysis to still the evaluation process. The evaluation is to find the best or optimized algorithm for sentimental analysis.

# **2.3.1 Dataset**

The proposed system utilizes a dataset sourced from various repositories, focusing on Chronic Kidney Disease (CKD) patient records. This dataset encompasses a comprehensive collection of patient data, including clinical markers, demographic details, lifestyle factors, and medical history. The dataset is curated to ensure accuracy and relevance, with data points spanning across different stages of CKD progression. To enhance the model's predictive capabilities and robustness, the dataset is enriched with additional features such as laboratory test results, medication history, and comorbidities. Data preprocessing techniques are employed to handle missing values, outliers, and feature scaling, ensuring the dataset's quality andconsistency.



# **Figure 2.4 Medical Test Dataset**

Figure 2.4 represents the Medical test dataset. The dataset contains three parameters such as label, id and Features.

**Label:** Whether the individual has kidney disease or not (1 for presence, 0 for absence).

**ID:** Unique identifier for each record.

**Features:** Various features related to kidney health, demographic information, medical history, and laboratory test results

# **2.3.2 Data Preprocessing**

**1.Handling Missing Values:**

Identify missing values in your dataset.

For numerical features (blood test parameters), consider imputation techniques such as mean, median, or mode imputation. However, since kidney disease is a critical medical condition, it's essential to handle missing values carefully. Consult with medical professionals to decide the most appropriate imputation strategy.

For categorical features, consider imputation with the mode or create a separate category to represent missing values.

**2.Feature Scaling:**

Assess the range and distribution of each numerical feature. Determine if feature scaling is necessary.

If features have vastly different ranges or if the algorithm used (random forest) is sensitive to feature scales, apply feature scaling techniques such as Min-Max scaling (scaling features to a range between 0 and 1) or Standardization (scaling features to have a mean of 0 and a standard deviation of 1).

**3.Feature Selection:**

Identify the most relevant blood test parameters/features for predicting kidney disease stages.

Perform correlation analysis to understand the relationships between features and the target variable (kidney disease stage).

Utilize domain knowledge and consult with medical professionals to select features that are clinically relevant and likely to influence kidney disease stages.

Employ techniques such as Recursive Feature Elimination (RFE) or feature importance from random forest to automatically select the most informative features.

**4.Handling Categorical Data:**

Identify categorical variables in your dataset (e.g., gender).

Decide on the appropriate encoding technique:

For binary categorical variables, you can use label encoding (e.g., Male: 0, Female: 1).

For categorical variables with more than two categories, consider one-hot encoding to create binary dummy variables for each category.

Ensure that the encoding preserves the meaning of the categorical variables and does not introduce spurious relationships.

**5.Data Splitting:**

Split your preprocessed dataset into training, validation, and test sets. A common split ratio is 70% for training, 15% for validation, and 15% for testing.

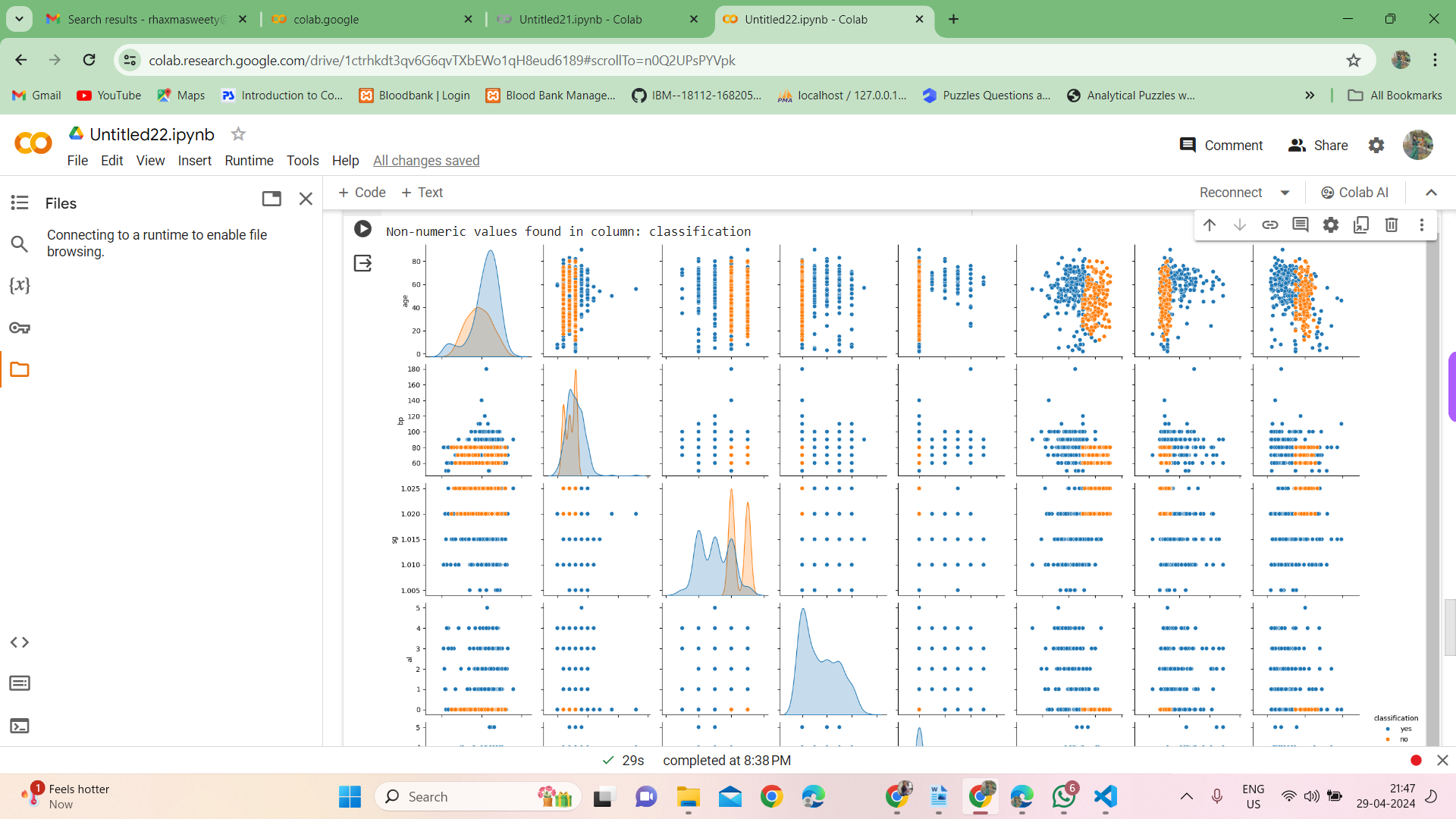
Ensure that each set maintains the distribution of kidney disease stages to prevent bias in model evaluation.

# **Data visualization:**

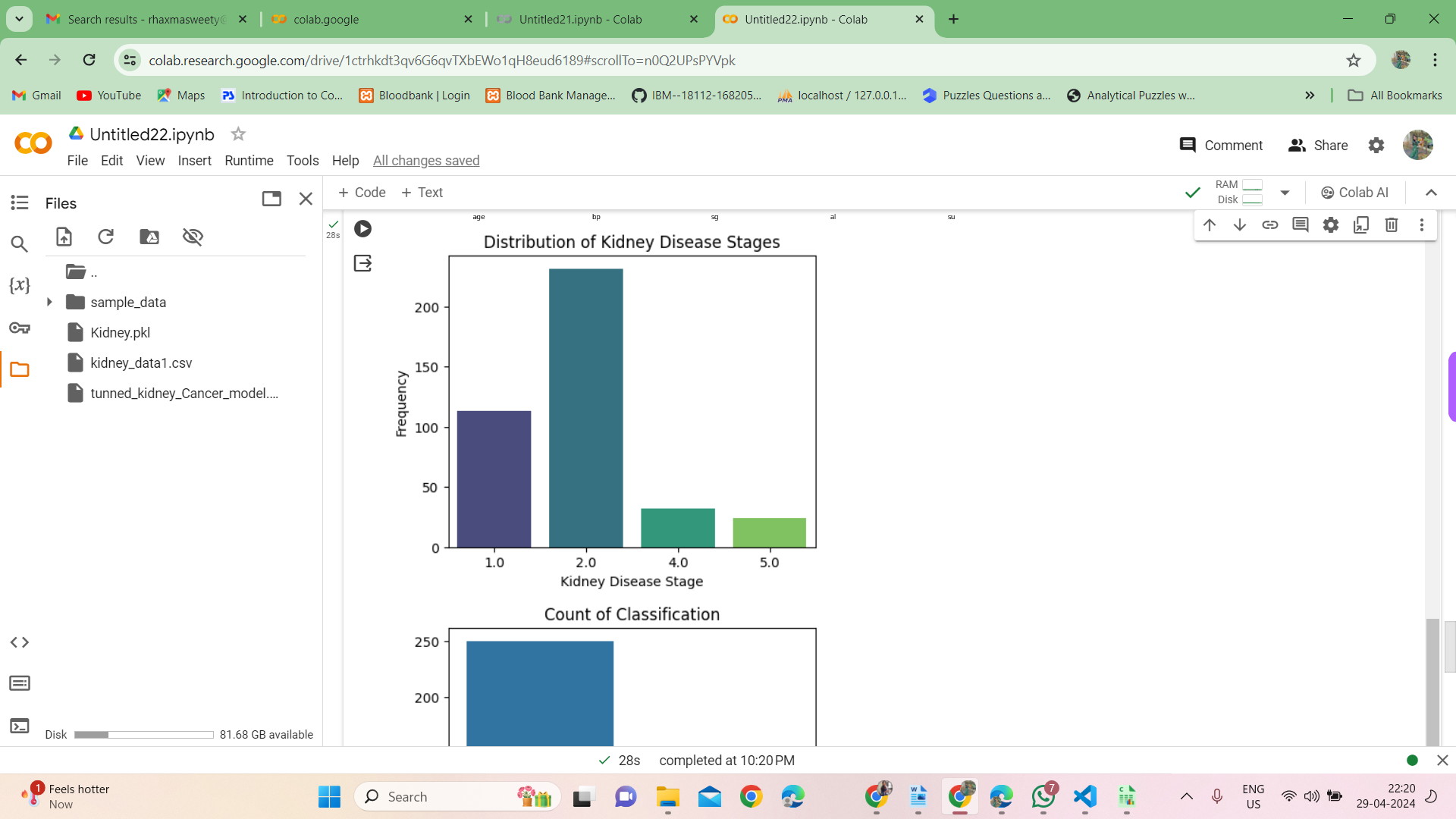
**Pair plots**

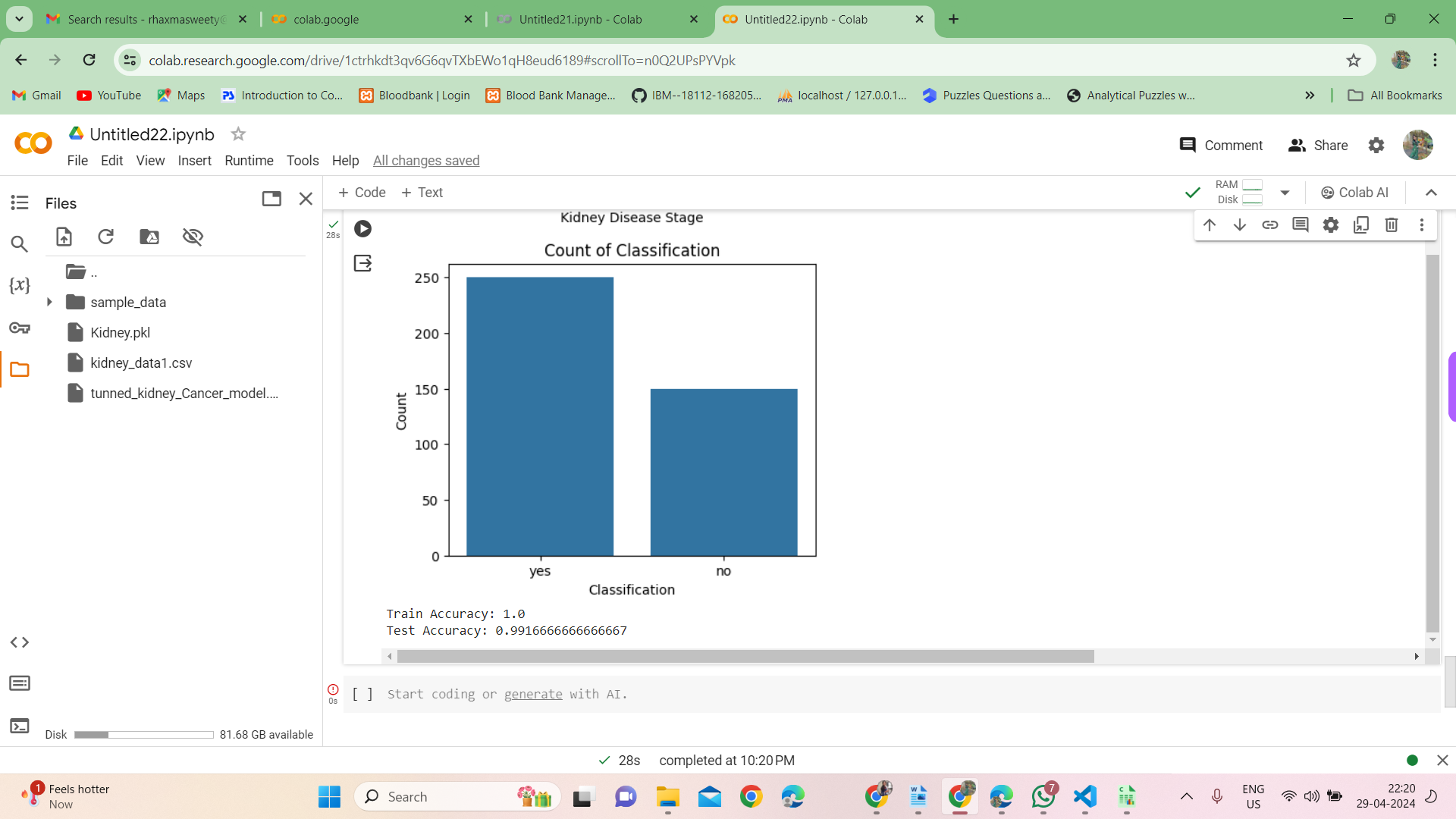
A pair plot is a type of visualization that displays pairwise relationships between different variables in a dataset. It's particularly useful for exploring the correlation between numerical variables and identifying patterns or trends.

**Bar plots**

****

Bar plots are utilized to visualize the distribution of kidney disease stages in the dataset. Each bar represents the frequency or count of individuals belonging to a specific kidney disease stage. The stages of kidney disease are typically categorized based on severity or progression, such as early-stage, moderate-stage, and advanced-stage. By analyzing the bar plot, users can gain insights into the distribution of kidney disease stages within the dataset and identify any imbalances or trends that may affect model training and prediction accuracy.





**Figure 2.6 Data visualization**

* + 1. **Features Extraction:**

**Clinical Data Analysis:**

Extracting features from clinical data such as patient demographics, medical history, symptoms, and laboratory test results.

Features could include age, gender, blood pressure readings, serum creatinine levels, albumin-to-creatinine ratio, glomerular filtration rate (GFR), and presence of proteinuria or hematuria.

**Laboratory Test Results:**

Extracting features from laboratory tests commonly performed in the diagnosis and monitoring of kidney disease.

This may involve extracting numerical values (e.g., creatinine levels, blood urea nitrogen, electrolyte levels) as well as categorical variables indicating the presence or absence of specific markers

**Biomarkers:**

Identifying and extracting features from biomarkers associated with kidney disease.

Biomarkers could include specific proteins (e.g., albumin, C-reactive protein), metabolites, genetic markers, or epigenetic markers that have been found to be associated with kidney function or disease progression.

**Text Mining of Medical Records:**

Extracting features from unstructured text in medical records, such as clinical notes, radiology reports, or pathology reports.

Natural language processing (NLP) techniques can be used to extract relevant information, including keywords, phrases, and structured data (e.g., diagnoses, treatments).

# **2.3.5 Data splitting**

The dataset is divided into three dataset. They are train dataset, validate dataset and test dataset. 70% of the dataset is taken as the train dataset. From the train dataset ten percent is taken as the validate dataset. Remaining dataset is taken as the test dataset.

# **Applying machine learning models**

# The proposed algorithm utilizes the random forest algorithm for predicting kidney disease stages based on blood test parameters provided by users. Random forest is a powerful ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees.

# **Random Forest Algorithm:**

# Random forest is a versatile machine learning algorithm suitable for both classification and regression tasks.

# It operates by constructing a multitude of decision trees during training, where each tree is trained on a random subset of the training data and features.

# During prediction, the random forest aggregates the predictions of all individual trees to determine the final prediction.

# Random forest is known for its robustness to overfitting, handling of high-dimensional data, and ability to capture complex relationships between features and the target variable.

# It is particularly well-suited for classification tasks with categorical target variables, such as predicting the stages of kidney disease based on blood test parameters.

# Random forest models are highly interpretable, allowing users to analyze the importance of each feature in predicting kidney disease stages and gaining insights into the underlying data patterns.

# By leveraging the random forest algorithm, the proposed kidney disease prediction model can effectively learn from the provided blood test parameters and make accurate predictions regarding the stages of kidney disease for users. Random forest's ability to handle high-dimensional data and capture complex relationships ensures robust and reliable predictions, facilitating better decision-making in healthcare settings

# **Result Evaluation**

To evaluate the performance of the kidney disease prediction model, the proposed system utilizes the F1 score, a statistical measure that combines precision and recall to assess test accuracy.

**F1 Score Calculation:**

**Precision:**

Precision is calculated as the ratio of true positive predictions to the total number of positive predictions made by the model. It represents the ability of the model to correctly identify positive instances out of all instances predicted as positive.

**Recall:**

Recall is calculated as the ratio of true positive predictions to the total number of actual positive instances in the dataset. It measures the ability of the model to correctly identify all positive instances, including those that were missed.

**F1 Score:**

The F1 score is the harmonic mean of precision and recall, providing a single metric to evaluate the balance between precision and recall.

It is calculated using the formula:

𝐹

1

=

2

×

precision

×

recall

precision

+

recall

F1=

precision+recall

2×precision×recall

The F1 score ranges from 0 to 1, where a score closer to 1 indicates better overall performance of the model.

**Model Evaluation:**

The kidney disease prediction model trained using the random forest algorithm is evaluated using the F1 score on the validation dataset.

Both the bag-of-words (BOW) and term frequency-inverse document frequency (TF-IDF) feature extraction methods are applied to the validation dataset before feeding it into the model.

The model that achieves a higher F1 score is considered the best or optimized model for predicting kidney disease stages.

Once the optimized model is identified, it is further evaluated using the test dataset to assess its performance on unseen data.

**Interpretation:**

A higher F1 score indicates better performance of the kidney disease prediction model in terms of both precision and recall. The optimized model selected based on the F1 score is deemed reliable for predicting kidney disease stages accurately.

Insights gained from analyzing the model's predictions can inform healthcare decisions and interventions, potentially improving patient outcomes. By utilizing the F1 score as the primary evaluation metric, the proposed kidney disease prediction system ensures that the model's performance is rigorously assessed and optimized for accurate prediction of kidney disease stages.

## CHAPTER 3

## DEVELOPMENT PROCESS

The CDK project using the Random Forest algorithm encompasses various stages: requirement analysis, definition, design, testing, and implementation.

**3.1.1 INPUT REQUIREMENTS**

Acquire CDK Data: Collect relevant CDK-related data.

Data Cleaning: Pre-process and cleanse the CDK dataset.

Feature Extraction: Implement Bag-of-Words and TF-IDF for feature extraction.

Model Creation: Develop a Random Forest machine learning model for CDK.

Data Analysis: Apply the Random Forest model to the CDK dataset.

Result Visualization: Visualize the outcomes of the Random Forest analysis.

**3.1.2 OUTPUT REQUIREMENTS**

The project's output aims to:

Track CDK Sentiments: Monitor CDK-related discussions and sentiments.

Issue Detection: Identify and address negative sentiments or issues related to CDK proactively.

**3.1.3 RESOURCE REQUIREMENTS**

The hardware and software configurations for this CDK project are as follows:

## 

## HARDWARE REQUIREMENTS:

## Processer:Pentium III& Above

## 

## RAM : 2GB

Hard disk : 40GB

Speed : Minimum 2.5 GHZ

**SOFTWARE REQUIREMENTS:**

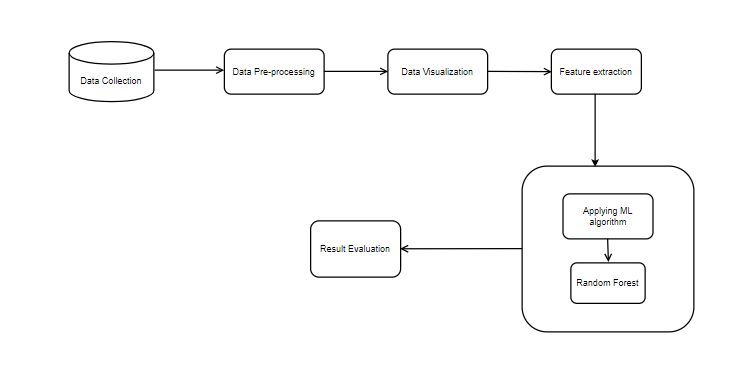
Operating System : Windows XP, 7, 8 or 1

Tool :Google Colab, Python

Language: Python

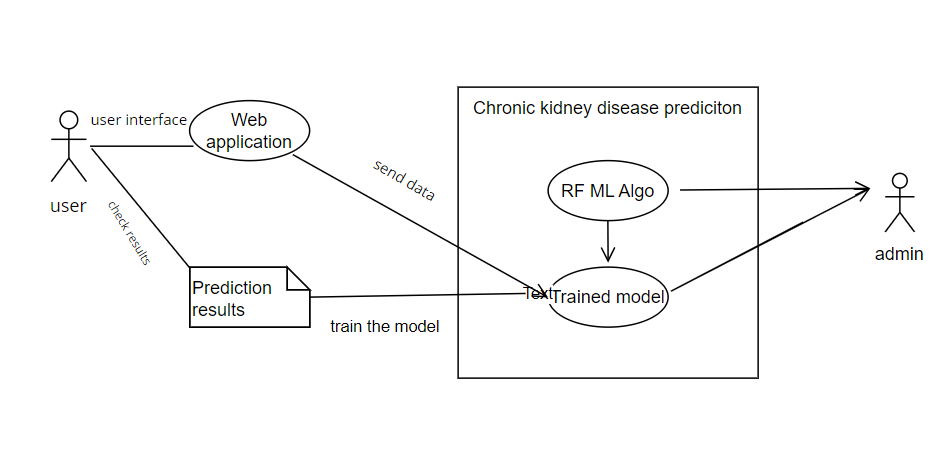
## 3.2 SYSTEM DESIGN

The design explains the various modules of the project, the overall architecture of the project, workflows and data flows of the project etc. Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

**3.2.2 SYSTEM ARCHITECTURE**

**Fig 3.2.2 Overall Architecture Diagram**

**3.2.4 USE CASE DIAGRAM:**

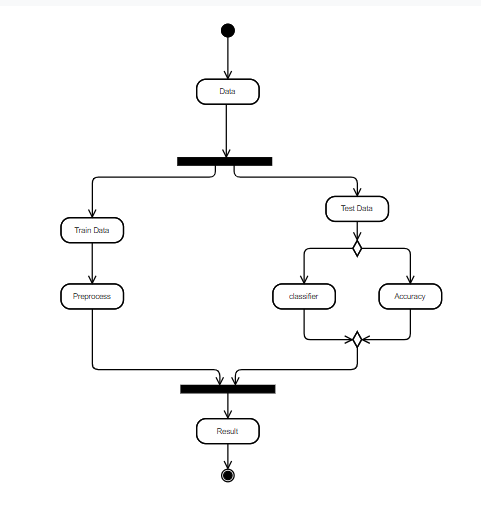


## 

## 

## Fig 3.2.4 Use Case Diagram

**3.2.7 ACTIVITY DIAGRAM:**

****

## Fig 3.2.8 Activity Diagram

## DATA FLOW DIAGRAM:

**Definition:**

A Data Flow Diagram (DFD) is a graphical tool used to describe and analyze the movement of data through the system. It is a graphical representation of the “flow” of data through a computer system or a data or it looks at how data flows through a system. These are central tools and basic from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. The development of DFD is done at several levels. The flow diagram describes the boxes that describe computations, decisions, interactions & loops.

## Characteristics

* Information and/or data flow is represented by a labeled arrow
* Processes (transformations) are represented by labeled circles (bubbles)
* Information sources and sinks are represented by boxes
* Files and depositories are represented by a rounded rectangle or a double line.

## Types

* Logical data flow diagram
* Physical data flow diagram

## Features

* The DFD shows data, not the control loops and decisions are controlled considerations do not appear on a DFD
* The DFD does not indicate the time factor involved in any process whether the dataflow takes place easily daily, weekly, monthly or yearly
* The sequence of events is to bring out on DFD

## DFD Symbols Process

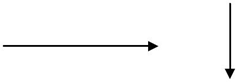
A process transforms incoming data flow into outsourcing data flow.

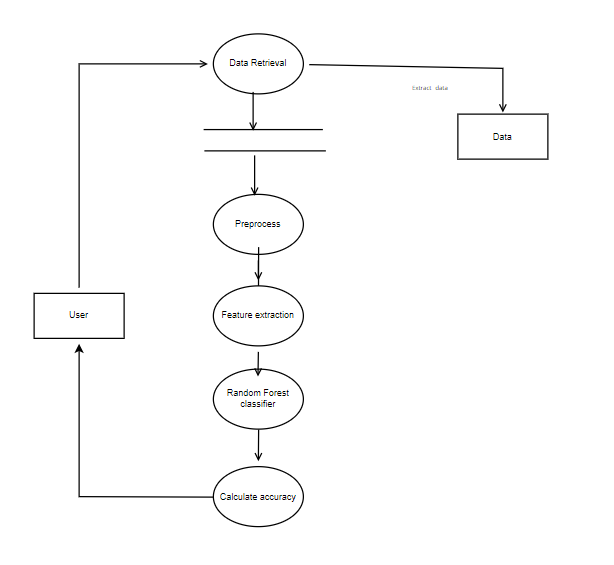
## Data store

Data sources are repositories of data in the system. They are sometimes also referred to as files.

## Data flow

Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.





**Fig 3.2.10 Data Flow diagram**

## CHAPTER 4

**SYSTEM TESTING AND IMPLEMENTATION**

## INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply both strategic to both large and small-scale systems.

4.2 **VISUAL STUDIO CODE (VS Code)**

Visual Studio Code, often abbreviated as VS Code, is a popular integrated development environment (IDE) developed by Microsoft. It offers a versatile and feature-rich environment for writing and executing Python code, making it suitable for various tasks including data analysis, machine learning, and software development**.**

**Key Features of Visual Studio Code:**

**Versatility:**

Visual Studio Code supports multiple programming languages, including Python, JavaScript, C++, and more. It provides a wide range of extensions and plugins, allowing users to customize and extend the functionality of the IDE according to their needs.

**Interactive Development:**

VS Code offers an interactive development experience with features such as IntelliSense, which provides code completion, suggestions, and auto-imports. It also supports debugging, allowing users to set breakpoints, inspect variables, and step through code execution.

**Integration with Git:**

Visual Studio Code seamlessly integrates with version control systems like Git, providing features for managing repositories, committing changes, and resolving conflicts directly within the IDE.

**Extensions Marketplace:**

VS Code has a rich ecosystem of extensions available through the Visual Studio Code Marketplace. Users can find and install extensions for various purposes, such as code formatting, linting, unit testing, and more.

**Customization and Theming:**

Visual Studio Code allows users to customize the editor layout, themes, and keyboard shortcuts to suit their preferences. Users can choose from a variety of built-in themes or install custom themes from the marketplace.

**Working with Visual Studio Code:**

**Installation:**

Visual Studio Code is available for Windows, macOS, and Linux platforms. Users can download and install it from the official website or package managers.

**Creating a New Python File:**

After installing Visual Studio Code, users can create a new Python file by opening the editor and selecting "New File" from the file menu or using the keyboard shortcut (Ctrl+N or Cmd+N). The Python file extension is typically .py.

**Writing and Executing Python Code:**

Users can write Python code directly in the editor window and execute it using the built-in terminal or Python interpreter. The output of the code execution is displayed in the terminal window below the editor.

**Integration with Jupyter Notebooks**:

Visual Studio Code provides support for Jupyter Notebooks, allowing users to create, edit, and execute Jupyter notebooks directly within the IDE.

Users can install the Jupyter extension for VS Code to enable this functionality.

**Sharing and Collaboration:**

Visual Studio Code offers features for sharing and collaborating on code projects, including Live Share, which allows multiple users to edit and debug code together in real-time.

# **PACKAGES REQUIRED:**

# To implement kidney disease prediction using the random forest algorithm, the following Python packages are required:

# **NumPy:**

NumPy, short for 'Numerical Python', is a fundamental package for scientific computing in Python. It provides support for fast mathematical computation on arrays and matrices, essential for handling and processing numerical data efficiently.

# **Pandas:**

Pandas is a powerful data manipulation and analysis library in Python, providing two-dimensional, labeled data structures called DataFrames.

It offers functionalities for creating, indexing, and manipulating data frames, handling missing data, and performing fast data manipulation and analysis operations.

# **scikit-learn (sklearn):**

scikit-learn is a versatile machine learning library in Python, offering various algorithms and tools for machine learning tasks such as classification, regression, clustering, and dimensionality reduction. It provides implementations of the random forest algorithm through the RandomForestClassifier class, which is suitable for classification tasks like kidney disease prediction.

**Seaborn:**

Seaborn is a Python data visualization library based on matplotlib, providing a high-level interface for creating informative and attractive statistical graphics.

It offers functionalities for visualizing univariate and bivariate distributions, as well as statistical summaries of datasets.

**Matplotlib:**

Matplotlib is a comprehensive plotting library for creating static, animated, and interactive visualizations in Python. It provides a MATLAB-like interface for generating plots and charts, allowing users to customize and annotate their visualizations to communicate insights effectively.

These packages provide essential functionalities for data manipulation, visualization, and machine learning modeling, enabling the implementation and analysis of the kidney disease prediction model using the random forest algorithm in Python.

* 1. **IMPLEMENTATION**
     1. **Preprocessing**

In order to prepare the dataset of blood test parameters for kidney disease prediction using the random forest algorithm, several preprocessing steps are undertaken to clean and refine the data:

**Data Cleaning:**

The collected dataset containing blood test parameters is initially cleaned to remove any irrelevant or erroneous data.

This process involves handling missing values, removing unnecessary columns, and ensuring data consistency and integrity.

**Handling Missing Values:**

Missing values in the blood test parameters are addressed using appropriate techniques such as imputation or removal, depending on the nature and distribution of the missing data.

**Normalization:**

The blood test parameters may have varying scales, so normalization techniques are applied to bring all features to a similar scale.

This ensures that each blood test parameter contributes equally to the prediction model.

**4.4.2 Data Visualization**

Visualizing the blood test parameters is essential for understanding their distribution and identifying patterns that may influence kidney disease prediction. The following visualization techniques are employed:

**Feature Importance Plot:**

After training the random forest model, a feature importance plot is generated to visualize the importance of each blood test parameter in predicting kidney disease stages. This plot helps in identifying the most influential features and understanding their contribution to the predictive performance of the model.

**Distribution Plots:**

Distribution plots, such as histograms or density plots, are utilized to visualize the distribution of individual blood test parameters among different kidney disease stages. These plots provide insights into the relationship between blood test parameters and kidney disease progression.

4.4.3 **Implementing machine learning algorithm**:

**Feature Extraction:**

The textual dataset of blood test parameters is extracted using the Bag-of-Words (BoW) technique and the Term Frequency-Inverse Document Frequency (TF-IDF) technique. For BoW, the CountVectorizer package from scikit-learn is used to convert the textual dataset into a matrix of token counts.

For TF-IDF, the TfidfVectorizer package from scikit-learn is utilized to extract features from the textual dataset based on term frequency.

**Dataset Splitting:**

The extracted dataset is divided into training, validation, and test sets to evaluate the performance of the random forest classifier accurately.

The training dataset is used to train the random forest model, while the validation dataset is employed to fine-tune hyperparameters and optimize the model.

**Model Training:**

The random forest classifier is implemented by importing the RandomForestClassifier class from scikit-learn.

Hyperparameters such as the number of trees, maximum depth, and minimum samples split are specified during model instantiation.

The fit() method is then called to train the random forest model on the training dataset.

**Prediction:**

After training the model, predictions are made on both the training and validation datasets to evaluate the model's performance.

The predict\_proba() method is used to obtain the probabilities of kidney disease stages predicted by the random forest classifier.

Results Visualization

**Output Visualization:**

The output of predicting probabilities for kidney disease stages using the random forest classifier is visualized to understand the model's predictions.

Probability outputs for different kidney disease stages are displayed to assess the model's accuracy and effectiveness.

**Figure Representation**:

Figures are generated to represent the output of predicting probabilities for kidney disease stages using the random forest classifier.

These figures provide visual insights into the model's predictions and help in evaluating its performance.

**Example Output:**

An example output of predicting probabilities for kidney disease stages using the random forest classifier is presented, showcasing the model's ability to predict the likelihood of different disease stages accurately.

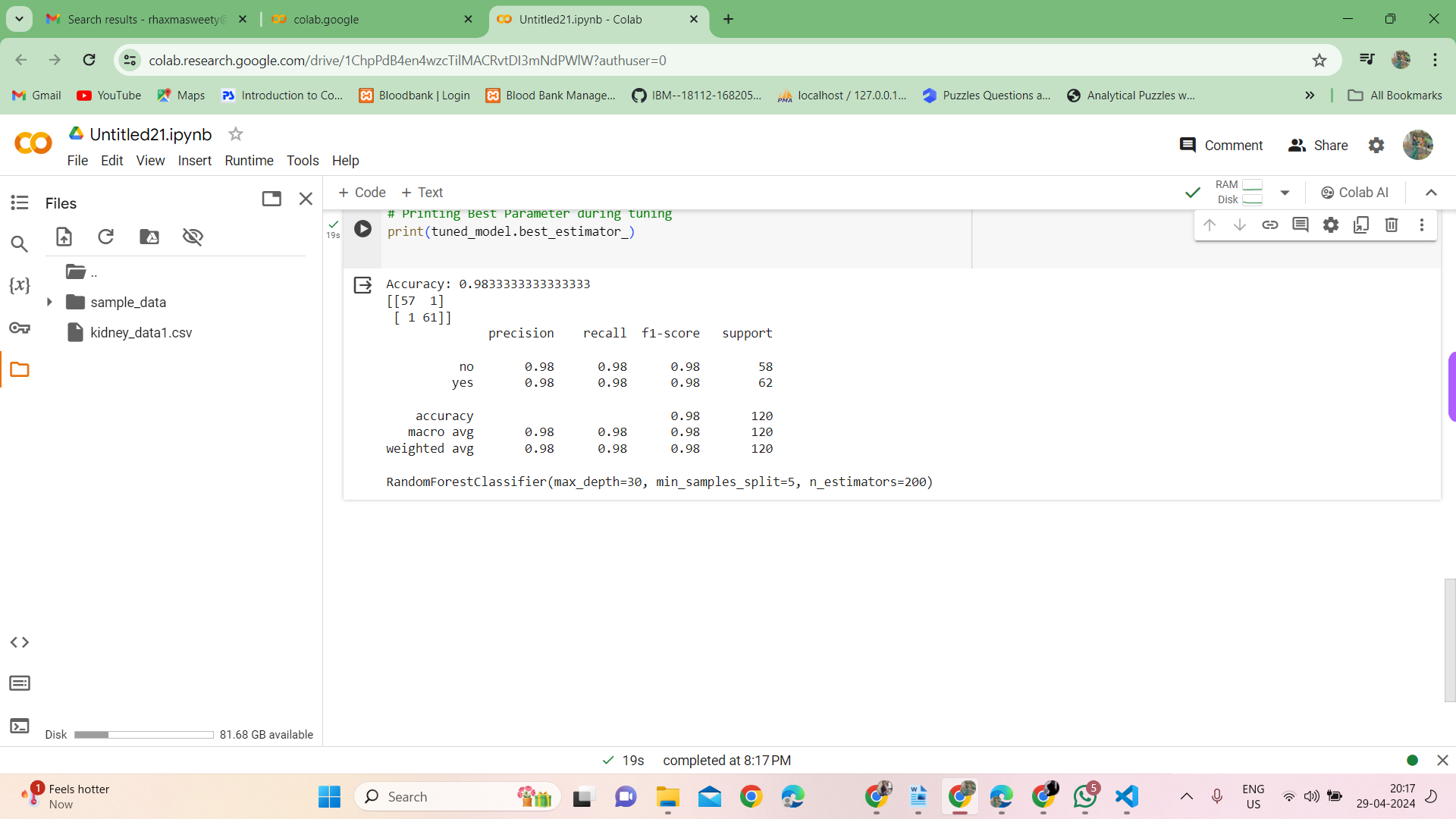
4.4.4 **Evaluation Metrics:**

After implementing the extracted training data into the machine learning algorithm for learning, apply the validation data for prediction. After predicting the probabilities of a tweet, calculate the F1\_score for each model. F1\_scores is used for evaluation metric instead of accuracy. Accuracy may encounter a higher number of false positives compared to F1\_scores. Thus F1\_score is used as an evaluation metric.

# **F1\_score comparison using features from Bag-of-Words:**

F1\_score for each model is calculated. The F1\_score is implemented by importing packages from sklearn. And it's given by

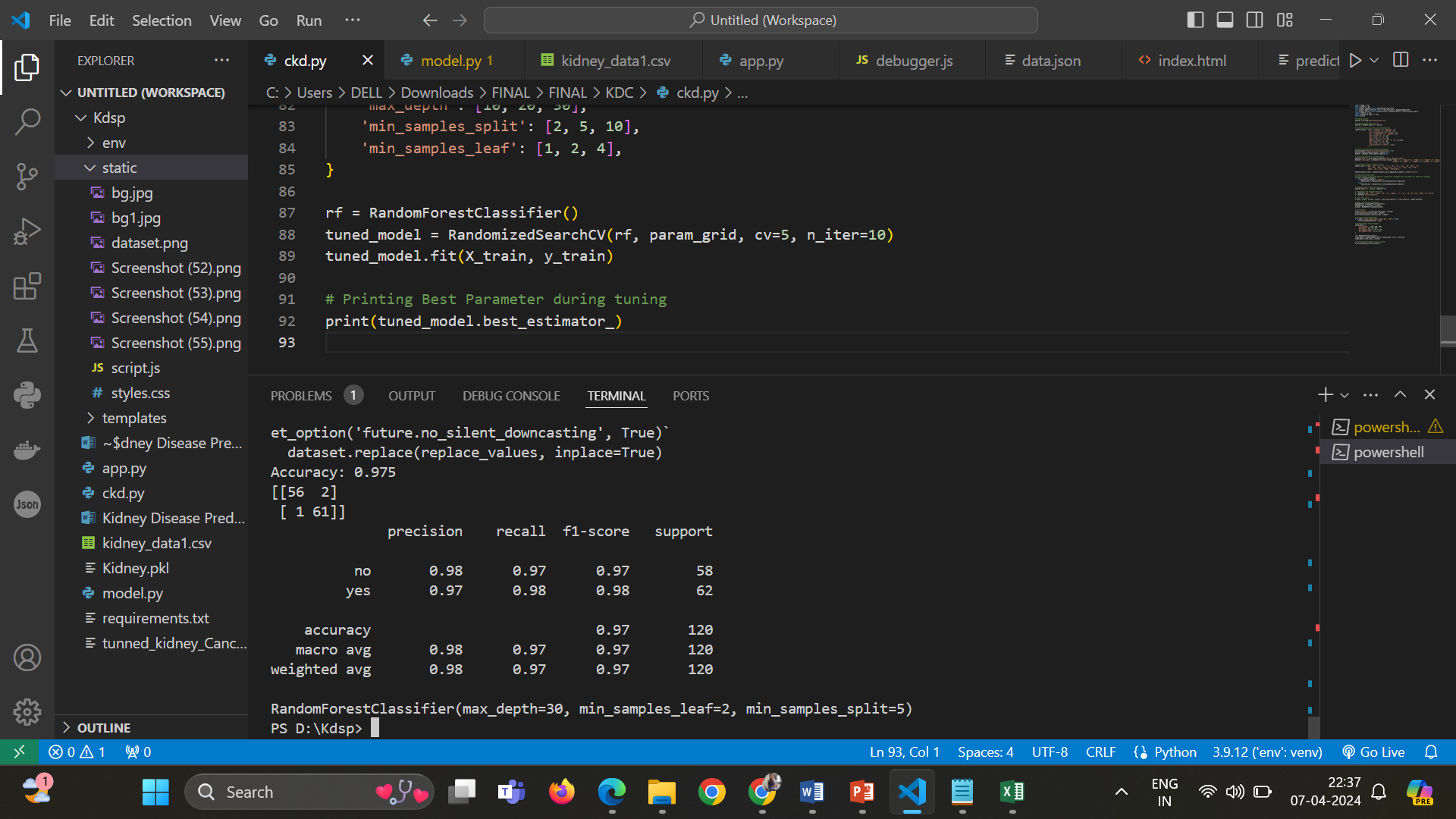
from sklearn.metrics port f1\_score

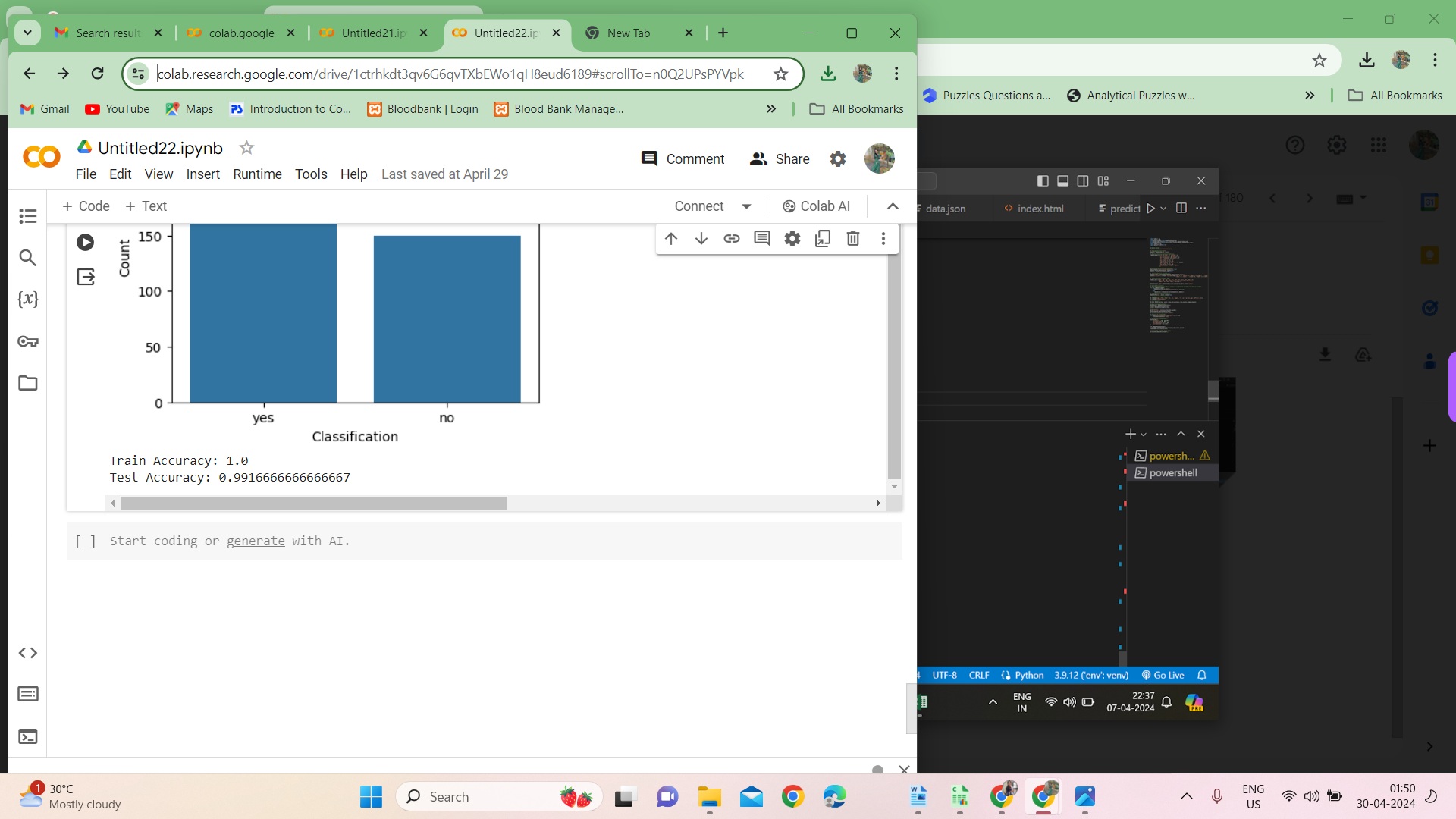


**Figure 4.8 (a) F1\_score**

# **Predicting the result for test data:**

For predicting the probabilities of tweet as positive or negative for the test data, the test data using feature from term frequency is applied to regression and decision tree model.





# **Figure 4.10 Comparison result of test dataset by regression and decision tree model**

Figure 4.10 represents the comparison result of the test data by applying the machine learning algorithm such as Logistic regression and Decision tree methods. Both the model provide the similar outputs.

## CHAPTER 5

**CONCLUSION**

The Kidney Disease Prediction project employs machine learning to forecast kidney disease based on specific parameters. The primary aim was to identify the most efficient and accurate algorithm for predicting kidney disease. By assessing the F1\_Score across different feature extraction techniques applied to the preprocessed dataset, the project sought to pinpoint the optimal algorithm. The choice of feature extraction method played a crucial role in influencing the performance metrics of the machine learning model. After comprehensive evaluation, the Random Forest algorithm emerged as the top-performing model for predicting kidney disease. When compared to other algorithms, Random Forest consistently achieved higher or equivalent F1\_Score results, showcasing its reliability and accuracy. Therefore, the optimized model for kidney disease prediction in this project is the Random Forest algorithm, offering a robust approach to accurately forecast kidney disease based on given parameters

# **FUTURE ENHANCEMENT**

In the Kidney Disease Prediction project, feature enhancement focuses on refining the Random Forest algorithm's predictive accuracy. The approach involves careful parameter selection, data augmentation to enrich the dataset, and dimensionality reduction techniques like PCA. To optimize performance, hyperparameter tuning and ensemble methods are employed, along with analyzing feature importance scores provided by Random Forest. Validation strategies include implementing k-fold cross-validation and evaluating the model using key metrics like F1\_Score, accuracy,andprecision

# **Reference:**

1. Shanshan Yi and Xiaofand Liu “Machine learning based customer sentiment analysis for recommendaing shoppers, shops based on customer review”, Springer journal- 2020
2. Abdullah Alsaeedi, Mohamed Zubair Khan “A study on Sentimental analysis techniques of twitter data”, International journal of Advanced Computer Science and Application, Vol 10, No 2, 2019.
3. Geetika Vashisht and Manisha Jailia “Panoptical view of the Sentiment Analysis Techniques” International journal of engineering and advanced technology, Vol 9, No 1, 2019.
4. Krutika Wase, Pranali Ramteke, Rushabh Bandewar, Nadim Badole and Bhuvneshwar Kumar. “Sentiment analysis of product review”, International Journal of Innovations in Engineering and Science, Vol. 3, No.5, 2018.
5. Janhavi N L, Santhosh Kumar K L and Jharna Majumdar, “Sentiment Analysis of Customer Reviews on Laptop Products for Flipkart”, International Research Journal of Engineering and Technology (IRJET), Vol 05, No. 03, Mar-2018.
6. Neethu M S and Rajasree R, “Sentiment Analysis in Twitter using Machine Learning Techniques”, 4th ICCCNT, 2013.
7. Zeenia Singla, Sukhchandan Randhawa and Sushma Jain, “Sentiment Analysis of Customer Product Reviews Using Machine Learning”, International Conference on Intelligent Computing and Control, 2017.
8. Gurneet Kaur and Abhinash Singla, “Sentimental Analysis of Flipkart reviews using Naive Bayes and Decision Tree algorithm”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol 5, No. 1, January 2016.
9. R. Jenitha Sri and P. Ajitha, “Survey of Product Reviews using Sentiment Analysis”, Indian Journal of Science and Technology, Vol 9, 2016.
10. Minara P Anto and Nivya Johny, “Product Rating using Sentiment Analysis”, International Conference on Electrical, Electronics and Optimization Techniques, ICEEOT, 2016.
11. Khaled Ahmed, Neamat El Tazi and Ahmad Hany Hossny, “Sentiment Analysis Over Social Networks: An Overview”, IEEE International Conference on Systems, Man, and Cybernetics, 2015.
12. Rastislav Krchnavy and Marian Simko,“Sentiment Analysis of Social Network Posts in Slovak Language”, IEEE International Conference on Semantic and Social media Adaption, 2017.
13. Oskar Ahlgren,“Research On Sentiment Analysis:The First Decade”, IEEE 16th International Conference on Data Mining Workshops, 2016.

# **Source Code**

import numpy as np

import pandas as pd

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split, RandomizedSearchCV

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

import seaborn as sns

import matplotlib.pyplot as plt

import pickle

# Reading Dataset:

dataset = pd.read\_csv("kidney\_data1.csv")

# Dropping unnecessary feature:

dataset = dataset.drop('id', axis=1)

# Replacing Categorical Values with Numericals

replace\_values = {'rbc': {'normal': 0, 'abnormal': 1},

'pc': {'normal': 0, 'abnormal': 1},

'pcc': {'notpresent': 0, 'present': 1},

'ba': {'notpresent': 0, 'present': 1},

'htn': {'yes': 1, 'no': 0},

'dm': {'yes': 1, 'no': 0},

'cad': {'yes': 1, 'no': 0},

'appet': {'good': 1, 'poor': 0, 'no': np.nan},

'pe': {'yes': 1, 'no': 0},

'ane': {'yes': 1, 'no': 0},

'classification': {'ckd\t': 'ckd'}

}

dataset.replace(replace\_values, inplace=True)

# Converting Objective into Numericals:

dataset['eGFR'] = pd.to\_numeric(dataset['eGFR'], errors='coerce')

dataset['ckd\_stage'] = dataset['ckd\_stage'].map({'Stage 1': 1, 'Stage 2': 2, 'Stage 3': 3, 'Stage 4': 4, 'Stage 5': 5,

'stage1': 1, 'stage2': 2, 'stage3a': 3, 'stage3b': 3, 'stage4': 4, 'stage5': 5})

# Converting columns to numeric types

numeric\_cols = ['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',

'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',

'appet', 'pe', 'ane', 'eGFR', 'ckd\_stage']

dataset[numeric\_cols] = dataset[numeric\_cols].apply(pd.to\_numeric, errors='coerce')

# Handling Missing Values:

# Impute missing values with mode for categorical variables and with median for numerical variables

for col in dataset.columns:

if dataset[col].dtype == 'object':

dataset[col] = dataset[col].fillna(dataset[col].mode()[0])

else:

dataset[col] = dataset[col].fillna(dataset[col].median())

# Dropping feature (Multicollinearity):

dataset.drop('pcv', axis=1, inplace=True)

# Independent and Dependent Feature:

X = dataset[['age', 'htn', 'hemo', 'dm', 'al', 'appet', 'rc', 'pc', 'sg','bp','bgr','eGFR','sc','sod']]

y = dataset['classification']

# Train Test Split:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=33)

# Model training and evaluation...

RandomForest = RandomForestClassifier()

RandomForest.fit(X\_train, y\_train)

y\_pred = RandomForest.predict(X\_test)

# Performance:

print('Accuracy:', accuracy\_score(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

# Serialize the trained model

with open("tunned\_kidney\_Cancer\_model.pkl", "wb") as file:

pickle.dump(RandomForest, file)

# Hyperparameter tuning example...

param\_grid = {

'n\_estimators': [100, 200, 300],

'max\_depth': [10, 20, 30],

'min\_samples\_split': [2, 5, 10],

'min\_samples\_leaf': [1, 2, 4],

}

rf = RandomForestClassifier()

tuned\_model = RandomizedSearchCV(rf, param\_grid, cv=5, n\_iter=10)

tuned\_model.fit(X\_train, y\_train)

# Printing Best Parameter during tuning

print(tuned\_model.best\_estimator\_)