5-2023

# A Machine Learning Approach to Evaluate the Effect of Sodium-Glucose Cotransporter-2 Inhibitors on Chronic Kidney Disease in Diabetes Patients

Solomon Eshun
*The University of Texas Rio Grande Valley*

# A MACHINE LEARNING APPROACH TO EVALUATE THE EFFECT OF SODIUM-GLUCOSE COTRANSPORTER-2 INHIBITORS ON CHRONIC KIDNEY DISEASE IN DIABETES PATIENTS

A Thesis

by

SOLOMON ESHUN

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: Applied Statistics and Data Science

The University of Texas Rio Grande Valley

May 2023

# A MACHINE LEARNING APPROACH TO EVALUATE THE EFFECT OF SODIUM-GLUCOSE COTRANSPORTER-2 INHIBITORS ON CHRONIC KIDNEY DISEASE IN DIABETES PATIENTS

A Thesis
by
SOLOMON ESHUN

COMMITTEE MEMBERS

Dr. Tamer Oraby
Chair of Committee

Dr. George Yanev
Committee Member

Dr. Hansapani Rodrigo
Committee Member

Dr. Zhuanzhuan Ma
Committee Member

May 2023

ABSTRACT

Eshun, Solomon, A Machine Learning Approach to Evaluate the Effect of Sodium-Glucose Cotransporter-2 Inhibitors on Chronic Kidney Disease in Diabetes Patients. Master of Science (MS), May, 2023, 79 pp., 6 tables, 33 figures, references, 27 titles.

Chronic kidney disease (CKD) is a significant complication that contributes to diabetes-related mortality in the United States, and there is growing evidence that sodium-glucose cotransporter 2 inhibitors (SGLT2i) can slow its progression. However, observational studies may suffer from confounding by indication, where patient characteristics and disease severity influence the decision to prescribe SGLT2i. This study utilized electronic health records of individuals with diabetes (from TriNetX) to investigate the effectiveness of SGLT2i on CKD progression. The database provided detailed information on patients' CKD status, demographics, diagnosis, procedures, and medications, along with corresponding dates of diagnosis and prescription. The study comprised of $38,776$ patients aged 18 years and above with 1 (or $>$) year history of diabetes who initiated treatment with SGLT2i between May 9, 2013, and July 7, 2021. To address potential confounding by indication in observational studies, we utilized propensity score matching. We also addressed the issue of imbalanced classification in medical datasets by applying balanced bagging (with bootstrap aggregation) and Synthetic Minority Oversampling Technique (SMOTE). By overcoming the underrepresented minority class and reducing bias in the machine learning (ML) models, we ensured accurate causal identification of CKD prevalence by employing the following ML models: logistic regression, decision tree, random forest, extreme gradient boosting, support vector classifier and artificial neural network. Our results suggest that SGLT2i have a protective effect on CKD outcomes, providing valuable insights into the practical efficacy of this treatment and potentially serving as a clinical decision-making tool. Our findings have significant clinical implications for the healthcare sector and suggest that these techniques can improve the reliability of ML methods.

DEDICATION

In loving reminiscence of my late father Mr. Matthew Kwame Eshun.

ACKNOWLEDGMENTS

My sincere thanks go first and foremost to the Almighty God, whose unfailing mercy and protection have helped me to persevere in my studies. Glory be to Him.

I would like to express my sincere gratitude to my thesis supervisor, Dr. Tamer Oraby, for his guidance, support, and valuable feedback throughout the entire research process. He has been a constant source of inspiration and encouragement. His expertise and dedication have been invaluable in shaping my research interests and refining my ideas.

A special gratitude I give to Dr. Mohammad Abdullah Al-Mamun, in the department of Pharmaceutical Systems and Policy at West Virginia University, whose contribution in interesting suggestions and encouragement, helped me to organise and improve the content of this work. He played an integral role in the initiation and scoping of my thesis, and provided me with the necessary resources and tools to complete my work, including access to relevant data.

I would like to acknowledge with much appreciation the vital role of my thesis committee members, Dr. George Yanev, Dr. Hansapani Rodrigo and Dr. Zhuanzhuan Ma. Their insights and suggestions have been critical in shaping my research and improving the quality of this thesis. Their expertise and willingness to share their knowledge have been truly inspiring.

Notwithstanding, I would like to take this opportunity to express my sincere gratitude to the Office for Sustainability at the University of Texas Rio Grande Valley for their invaluable support in my research work. Thank you for your continued support.

I would also like to express my heartfelt gratitude to Dr. Francis Kofi Andoh-Baidoo for his advice and career guidance throughout my academic journey. His support and mentorship have been instrumental in shaping my professional goals.

Lastly, I would like to thank my family and friends for their unwavering support and encouragement throughout my academic journey. Thank you all for your contribution to this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER I

# INTRODUCTION

## 1.1 Background

The increasing prevalence of diabetes in the United States (US) and the associated risks of complications such as chronic kidney disease (CKD) places a heavy burden on individual patients and on national health budgets (L. Chen, Magliano, and Zimmet 2012). Diabetes and its complications, deaths, and societal costs have a huge and rapidly growing impact on the United States. Between 1990 and 2010 the number of people living with diabetes tripled and the number of new cases annually (incidence) doubled (Gregg, Williams, and Geiss 2014). Reducing this burden will require efforts on many fronts—from appropriate medical care to significant public health efforts and individual behavior change across the nation, through state- and community-specific efforts. Public awareness is a key first step.

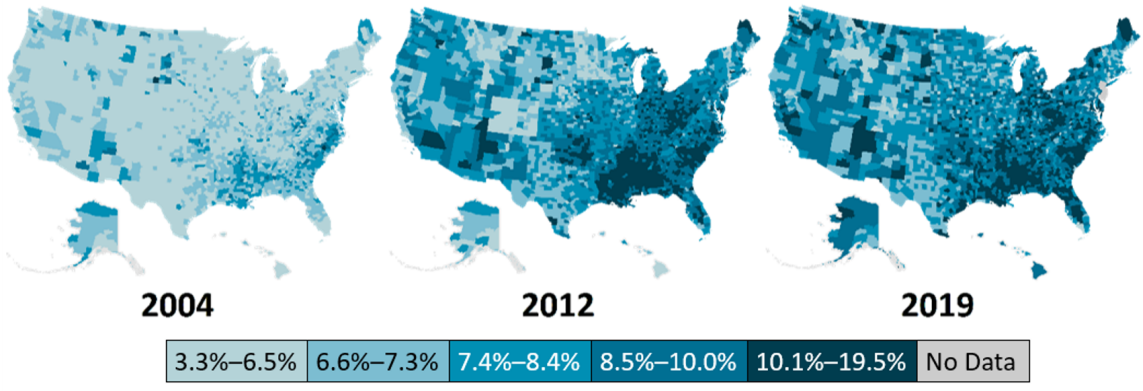


Figure 1.1: Prevalence of Diabetes Cases in the United States from 2004 - 2019 (Source CDC: www.cdc.gov)

Diabetes can lead to a number of complications including CKD. CKD is a general term for

any disorder that causes a gradual deterioration in kidney function or structure. It is a common complication for individuals with type 2 diabetes mellitus (T2DM) and is characterised by diminished renal function and/or increased urine albumin excretion. It is also closely linked to excess morbidity and cardiovascular as well as all-cause mortality (Afkarian et al. 2013; Thomas et al. 2015; Persson and Rossing 2018; Webster et al. 2017).

Diabetic kidney disease (DKD) is a CKD presumed to be caused by diabetes over time (Reutens 2013) and is mostly responsible for the higher mortality risk in diabetes patients (Afkarian et al. 2013). Generally speaking, this sickness, in its early stages, does not have obvious signs until the kidney has lost about 25% of its functionality. As a result, individuals with CKD may not be aware that they are experiencing the beginning stages of the disease (Polat, Danaei Mehr, and Cetin 2017).

The human body is a highly complex system in which the various organs and tissues are interdependent, relying on each other to function optimally. Despite having specific roles and responsibilities, each component is intimately connected to the others and works in concert to maintain a delicate balance. When one organ is not functioning as it should, it imposes stress on other organs and causes them to cease functioning properly as well. One illustration of how the organs are interconnected is the link between CKD, diabetes, and heart disease (also known as the cardiovascular disease).

The body uses a hormone called insulin to transport blood sugar into the cells where it may be used as energy. When a person develops diabetes, their pancreas cannot use the insulin that is produced as well as it should or doesn't produce enough of it. As a result, their kidneys are unable to remove waste and toxins from their blood as effectively as they need to, putting more stress on the heart. When someone has CKD, their heart needs to pump harder to get blood to the kidneys. This can lead to heart disease, the most common cause of mortality in the United States. Change in blood pressure is also a CKD complication that can lead to heart disease. Fortunately, controlling or preventing one illness can help control or prevent the others and reduce the likelihood of further complications.

With approximately half of patients with T2DM developing CKD, the rise in T2DM imposes a significant cost to patients as well as healthcare systems. CKD is diagnosed clinically by examining for persistently abnormal urine albumin excretion (UAE $\geq$ 30 mg per 24 hours), which is determined by at least two abnormal specimens in a period of three to six months, as well as by checking for a decreased estimated glomerular filtration rate (eGFR $<$ 60 mL/min $/1.73\text{m}^2$) (Reutens 2013; Allen et al. 2022).

Despite the fact that these parameters are fundamental clinical and laboratory measurements, CKD routine screening is not universally feasible; this can lead to missed or delayed diagnoses (Allen et al. 2022). Since those with T2DM are more likely to develop CKD, it is critical to quickly identify those who are at high risk to improve prognosis of these patients, but it is still unclear how to identify their risk.

## 1.2 Statement of the Problem

Despite the significant progress made in the understanding the pathophysiology of CKD and the development of various interventions, including lifestyle modifications, pharmacotherapy, and renal replacement therapy, the burden of CKD remains high. Therefore, continuous efforts are necessary to develop new interventions that can effectively control the incidence and progression of CKD. This requires a multi-disciplinary approach involving researchers, clinicians, policymakers, and patients, to identify the most promising strategies for the prevention and treatment of CKD.

Sodium-glucose cotransporter-2 (SGLT2) inhibitors are a class of drugs that have emerged as a promising treatment option for reducing CKD progression, with evidence suggesting that they may also protect the kidneys by reducing the risk of CKD progression (Birkeland et al. 2017).

The absorption of SGLT2 is visually depicted in Figure 1.2, providing an insightful understanding of the intricate processes involved in its function. By detailing the various stages and interactions that occur during this mechanism, the diagram presents a comprehensive overview of SGLT2 absorption.

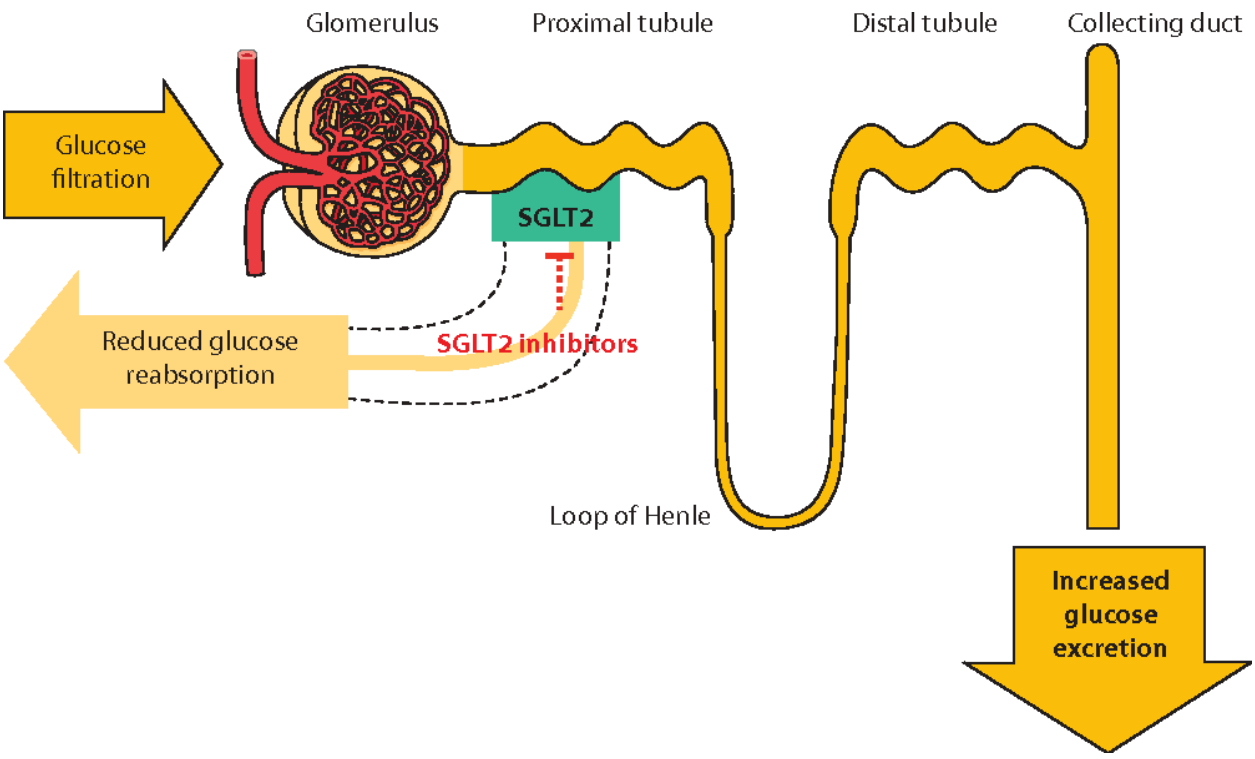


Figure 1.2: The figure illustrates the working mechanism of SGLT2 inhibitors, showcasing the key steps and interactions involved in this process (Vivian 2014).

However, observational studies to assess the effect of SGLT2 on CKD may suffer from confounding by indication, where patient characteristics and disease severity influence the decision to prescribe SGLT2. To overcome this issue, this study aims to address potential confounding by indication through propensity score matching (PSM) and various machine learning techniques, such as Synthetic Minority Oversampling Technique (SMOTE) and balanced bagging classifier (with bootstrap aggregation), to ensure accurate causal identification of CKD prevalence in the classification tasks. The results of this study aim to provide valuable insights into the practical efficacy of SGLT2 inhibitors and potentially serve as a clinical decision-making tool for the treatment of CKD in individuals with diabetes, addressing a significant problem in the healthcare industry.

## 1.3 Research Objectives

The general objective of this study is to investigate the effectiveness of SGLT2 inhibitors in slowing the progression of CKD. The specific Specific objectives are:

1. To investigate the effectiveness of SGLT2 inhibitors in slowing the progression of CKD in individuals with diabetes, by addresing the issue of potential confounding by indication through propensity score matching.
2. To overcome the issue of imbalanced classification by applying various techniques such as balanced bagging classifier (with bootstrap aggregation) and Synthetic Minority Oversampling Technique (SMOTE).
3. To ensure accurate causal identification of CKD prevalence in the classification tasks through the application of various machine learning models, including logistic regression, decision tree, random forest, extreme gradient boosting, support vector classifier and artificial neural network.
4. To provide valuable insights into the practical efficacy of SGLT2 inhibitors for the treatment of CKD using the SHapley Additive exPlanation (SHAP)analysis.

## 1.4 Relevance of the Study

The results of this study have important clinical implications for the healthcare industry, as they provide valuable insights into the practical efficacy of SGLT2 inhibitors and potentially serve as a clinical decision-making tool for the treatment of CKD in individuals with diabetes. By contributing to the body of knowledge on the effectiveness of SGLT2 inhibitors in the treatment of CKD, this study has the potential to improve patient outcomes, reduce healthcare costs, and enhance the quality of care for individuals with diabetes and CKD. Also, the best model attained could act as a major predictive indicator to assist medical professionals in identifying the prevalence of CKD in diabetic patients from the onset, to facilitate intervention and improve prognosis of the disorders.

## 1.5 Organisation of Thesis

The thesis is comprised of five parts. In Chapter 1, we introduce the topic of the study, providing background information and identifying the specific problem being investigated. This chapter also outlines the objectives of the study and the methods that will be used to achieve them.

Moving onto Chapter 2, we present a comprehensive review of the literature related to the

topic of the study. This includes a discussion of relevant theories, previous research studies, and any other information that is pertinent to the research question.

In Chapter 3, we provide an in-depth discussion of the methods used to build the thesis. Here, we detail the methods and procedures used to gather and analyze data, as well as any tools or techniques utilized throughout the project.

In Chapter 4, we present the results and discussions of the study, using visual representations such as tables and graphs to illustrate key findings. This chapter also includes a detailed discussion of the results, interpreting and contextualizing them within the broader context of the study.

Finally, in Chapter 5, we provide the conclusions of the study, highlighting the key findings and their implications. This chapter also includes a discussion of the limitations of the work and recommendations for future research directions.

Overall, this thesis provides a comprehensive and detailed investigation of the chosen topic, using a range of methods and techniques to explore and analyze key research questions.

# CHAPTER II

# LITERATURE REVIEW

## 2.1 Introduction

CKD is a significant public health problem worldwide, and its prevalence is higher in patients with diabetes. CKD in diabetes patients is associated with a higher risk of cardiovascular disease, hospitalization, and mortality. Early diagnosis and management of CKD are critical to preventing its progression and its associated complications.

In recent years, machine learning algorithms have been increasingly used in healthcare for the diagnosis and management of chronic diseases.

This chapter provides an overview of the current literature on the use of machine learning algorithms to diagnose CKD in diabetes patients.

## 2.2 Traditional Approaches to Diagnose CKD in Diabetes Patients

Traditional approaches to diagnose CKD in diabetes patients include clinical assessment and laboratory testing.

Clinical assessment involves evaluation of medical history and physical examination to identify signs and symptoms of kidney disease. Patients with diabetes are at higher risk of developing CKD, and regular monitoring of blood pressure, urine albumin-to-creatinine ratio (ACR), and estimated glomerular filtration rate (eGFR) is recommended ("Introduction: Standards of Medical Care in Diabetes-2021" 2020).

Laboratory testing includes blood and urine tests that can help identify early signs of kidney damage. Blood tests such as serum creatinine and blood urea nitrogen (BUN) are commonly used to estimate kidney function. eGFR is calculated using serum creatinine levels, age, gender, and

race (Levey and Coresh 2012). A urine ACR test measures the amount of protein in the urine and is an indicator of kidney damage.

The combination of eGFR and urine ACR testing is commonly used to diagnose CKD. The National Kidney Foundation recommends that patients with diabetes should be screened for CKD using both tests annually (Bilous et al. 2012). A diagnosis of CKD is made when eGFR is less than 60 mL/min/1.73 m² or urine ACR is greater than 30 mg/g.

Other laboratory tests such as hemoglobin A1c (HbA1c) and lipid profile can also help identify risk factors for CKD in diabetes patients. HbA1c measures the average blood glucose levels over the past 2-3 months and is used to monitor diabetes control. High levels of HbA1c are associated with an increased risk of developing CKD ("Introduction: Standards of Medical Care in Diabetes-2021" 2020). Lipid profile measures the levels of cholesterol and triglycerides in the blood, which are risk factors for cardiovascular disease and CKD.

Overall, traditional approaches to diagnose CKD in diabetes patients involve a combination of clinical assessment and laboratory testing. Regular monitoring of blood pressure, eGFR, urine ACR, HbA1c, and lipid profile is recommended for early detection and management of CKD in diabetes patients.

### 2.3 Previous Researches Concerning the Concept

The quest to pinpoint the risk of CKD in diabetes patients has been the subject of extensive research to date. A significant number of research have focused in particular on developing machine learning (ML) models to predict the risk of CKD in the early-stage T2DM patients due to their scientific advancement and classification abilities in medical diagnosis. ML algorithms can analyze large datasets to identify patterns and predict outcomes, which can improve the accuracy and efficiency of CKD diagnosis.

Below are some of the works of some authors on the definition of the concept and various research on diagnosing the risk of CKD in diabetes patients.

Allen *et al*. (Allen et al. 2022) developed and validated an XGBoost and a random forest model that predicts stages of CKD within 5 years upon diagnosis of T2DM. To assess performance,

they compared the two methods' prediction of CKD risk with the Centre for Disease Control and Prevention (CDC) risk score. The models were validated on a hold-out test set as well as an external dataset sourced from separate facilities. The ML algorithms outperformed the CDC risk score in both the hold-out test and external datasets by attaining an area under the receiver operating characteristic curve (AUROC) of 0.75 on the hold-out set for prediction of any-stage CKD and an AUROC of over 0.82 for more severe endpoints, compared with the CDC risk score with an AUROC $< 0.70$ on all test sets and endpoints. Their retrospective analysis indicated that an ML algorithm can provide timely predictions of CKD among patients with recently diagnosed T2DM.

Qin *et al*. (Qin et al. 2020) applied six ML algorithms (logistic regression, random forest, support vector machine, *k*-nearest neighbour, Naïve Bayes and feed forward neural network). Among these methods, random forest achieved the best performance with 99.75% diagnosis accuracy. They then created an integrated model that incorporates logistic regression and random forest (models with better performances) by utilising the perceptron learning approach, which after ten simulations obtained an average accuracy of 99.83%. This was done by analysing the errors produced by the established models. Their conclusion was that the method would have relevance in diagnosing complex diseases in clinical settings.

Yang *et al*. (Yang et al. 2022), on the other hand, used ML technique to predict acute kidney injury (AKI) risk in patients with type 2 diabetes (T2D) prescribed sodium-glucose co-transporter two inhibitors (SGLT2i). They used a 5% random sample of medicare claims data, and identified $17,694$ patients who filled $\geq 1$ prescriptions for canagliflozin, dapagliflozin and empagliflozin in 2013–2016. They measured 65 predictor candidates using claims data from the year prior to SGLT2i initiation (including information on socio-demographics like age, sex, race region of residence), diabetes duration, comorbidities, and other medications, and then applied three machine learning models, including random forests, elastic net and least absolute shrinkage and selection operator (LASSO) for risk prediction. Among these three machine learning methods, random forests produced the best prediction (C-statistic $= 0.72$), followed by LASSO and elastic net (both C-statistics $= 0.69$).

Makino *et al*. (Makino et al. 2019) constructed a predictive model for CKD using Artificial Intelligence (AI), processing natural language and longitudinal data with big data machine learning, based on the electronic medical records (EMR) of 64,059 diabetes patients. Using a convolutional auto-encoder, AI extracted raw features from the preceding six months as the reference period and selected 24 factors to uncover time series patterns linked to six-month CKD aggravation. A logistic regression was used by AI to build the predictive model with $3,073$ features, including time series data; it predicted a CKD aggravation with 71% accuracy. Furthermore, they observed that the group with CKD aggravation had a significantly higher incidence of hemodialysis than the non-aggravation group, over 10 years ($N = 2,900$). Their predictive model by AI was able to detect progression of CKD and may contribute to more effective and accurate intervention to reduce hemodialysis.

Following the review of relevant literature on the suggested topic, this work seeks to utilize machine learning techniques to evaluate the effectiveness of SGLT2 inhibitors on CKD in patients with diabetes. In order to aid in the early diagnosis of CKD, the actual investigation would seek to find a functional link between diabetes control (i.e., initiating SGLT2) and the aforementioned disorder, and then assess the effectiveness and performance of the methods for the data under consideration.

# CHAPTER III

# METHODS

## 3.1 Introduction

This chapter of the study discusses the research design, data and variables included in the study, estimation procedures and method in building the model, and some relevant statistical computations carried out.

## 3.2 Data Source and description

The data for this study is a de-identified dataset obtained from TriNetX, which includes medical records of $38,776$ patients with a history of diabetes spanning one year or more.

The dataset includes a total of 264 features, providing comprehensive information on patients' diabetes status, CKD status, heart failure status, and SGLT2 use, as well as their demographics, including age ($\geq$ 18 years), sex, race, and ethnicity.

The dataset also contains information on diagnosis, medications, and clinical procedures, classified using International Classification of Diseases, 9th and 10th Revision (ICD-9 and ICD-10) diagnosis codes, Current Procedural Terminology (CPT) codes, RxNorm, and National Drug Code (NDC) codes.

This dataset serves as a valuable resource for investigating the relationships between diabetes and various health outcomes. The extensive patient information available allows for comprehensive analysis and provides insights into potential treatment options and interventions for patients with diabetes. The utilization of the standardized classification systems ensures accuracy and consistency in the analysis of the data.

### 3.3 Propensity Score Matching (PSM) in Observational Studies

Causal inference is a vital aspect of medical research that helps to establish causal relationships between different variables. It is the process of making claims about cause-and-effect relationships in a given system or phenomenon. It is an essential component of scientific research, particularly in fields such as medicine, economics, and social sciences, where experiments that manipulate variables are often not feasible or ethical. In these cases, researchers often rely on comparative experimental studies.

Comparative experimental studies are a type of research design used to evaluate the effectiveness and safety of medical interventions, procedures, or treatments. These studies involve randomly assigning patients to different treatment groups, and then comparing the outcomes of these groups. The gold standard of comparative experimental studies is the randomized controlled trial (RCT), which involves randomly assigning patients to either an experimental group that receives the intervention, or a control group that does not. RCTs are designed to minimize the effects of bias and confounding variables, as randomization helps to ensure that the groups are similar in all relevant factors except for the intervention.

Observational studies become important in fields such as medicine, economics, and social sciences where it may not be feasible or ethical to conduct experiments that manipulate variables. However, making causal inferences in observational studies can be challenging due to limitations such as random selection of subjects but not random allocation of treatments to subjects. This makes it difficult to determine whether the difference in outcome between treated and untreated subjects is due to the treatment or differences in other characteristics of the subjects.

Another limitation is self-selection, where individuals opt for a particular treatment for specific reasons, making it difficult to compare them directly with those who did not receive the treatment. For instance, individuals from affluent backgrounds may have ease of access to SGLT2, a medication used for treating diabetes, and reducing the risk of CKD. Therefore, it is essential to consider the influence of confounding factors if we want to accurately measure the effect of SGLT2 on CKD.

There is an increasing interest in estimating the causal effects of treatment using observational or nonrandomized data. In observational studies, the baseline characteristics of treated or exposed subjects often differ systematically from those of untreated or unexposed subjects. Essential to the production of high-quality evidence to inform decision-making is the ability to minimize the effect of confounding. An increasingly frequent approach to minimizing bias when estimating causal treatment effects is based on the propensity score (the likelihood of receiving treatment based on some outlines subject characteristics) usually obtained using a logit model (Rosenbaum and Rubin 1983).

Propensity score matching have gained much relevance across disciplines to estimate causal effects using observational data. This technique attempt to replicate the ideal of randomized experiments as closely as possible. It a quasi-experimental method that aims to search for counterfactual unit that is comparable with the treated unit among many untreated units.

By matching individuals based on their propensity score, PSM seeks to reduce the influence of confounding bias and increases the internal validity of the study. This allows the researcher to make causal inferences about the treatment effect without having to randomly assign participants to treatment or control groups. Before conducting PSM, it is important to identify potential confounding variables that may affect both the treatment and the outcome of interest, and include them in the analysis.

The following are some of the general procedures involved in conducting PSM:

1. Estimate the propensity score, which is the probability of receiving the treatment/exposure given the confounding variables. The logit model is defined in Equation 3.1 below

$$\text{logit}\left[\frac{\text{ps}}{1-\text{ps}}\right] = \beta_0 + \beta_1 \text{X}_1 + \beta_2 \text{X}_2 + \cdots + \beta_\text{n} \text{X}_\text{n} \tag{3.1}$$

   where, ps is the probability of receiving a treatment and $\text{X}_1, \text{X}_2, \ldots, \text{X}_\text{n}$ are confounding variables.

2. Match the treated and untreated groups based on the estimated propensity score. Matching

can be done using various methods such as nearest-neighbor matching, caliper matching, and kernel matching.

3. After conducting PSM, it is important to assess the balance of covariates between the treated and untreated groups to ensure that the matching was successful. There are several ways to assess balance, including using standardized mean differences (SMD) or other measures of balance, such as the Kolmogorov-Smirnov test. Visualizing the distribution of covariates can also be helpful in assessing balance. Violin plots, histograms, bar charts, and other graphical displays can be used to visualize the distribution of covariates before and after matching. If the distributions of covariates in the treated and untreated groups are similar after matching, it suggests that the matching was successful in achieving balance.

4. Estimate the treatment effect by comparing the outcomes between the treated and untreated groups after matching. The treatment effect can be estimated by comparing the outcomes between the treated and untreated groups. The most common approach is to calculate the average treatment effect (ATE), which is the difference in the mean outcome between the treated and untreated groups. In addition to point estimates, it is also important to report confidence intervals or standard errors to indicate the precision of the estimates.

## 3.4 Machine Learning Techniques

### 3.4.1 Logistic Regression

Logistic Regression is a classification technique used in machine learning. It uses a logistic function to model the dependent variable. The dependent variable is dichotomous in nature, i.e. there could only be two possible classes (eg. either 0 or 1). As a result, this technique is used while dealing with binary data.

In logistic regression, in order to map the predicted values to probabilities, sigmoid function is used. The sigmoid function/logistic function is a function that resembles an “S” shaped curve when plotted on a graph. It takes values between 0 and 1 and “squishes” them towards the margins at the top and bottom, labeling them as 0 or 1.

The decision for converting a predicted probability into a class label is decided by the parameter known as threshold. A value above that threshold indicates one class while the one below indicates the other.

### 3.4.2 Decision Tree Classifier

Decision Tree is a popular machine learning algorithm that presents classifications in the form of a tree. It works by breaking down the input data set into smaller sub-data sets based on the values of its features. This process is repeated recursively, until the sub-data sets are small enough to be classified with certainty. At the last level of the tree, the result is revealed in the form of class labels.

The tree is made up of nodes that represent the various features in the dataset, and it starts at the root node, which represents the entire dataset. As the algorithm progresses, the root node is split into two or more child nodes, depending on the conditions set for each split. The process of dividing the root node is called splitting, and it continues recursively until the tree reaches a point where it cannot be divided further. This is called a leaf node, which represents the final output of the algorithm.

In a tree structure depicted in Figure 3.1, the leaves of the tree represent the class labels while the branches represent conjunctions of features that lead to these class labels. The algorithm works by recursively splitting the data into subsets based on the most informative feature at each step until a stopping criterion is met. This allows the algorithm to capture complex relationships between features and class labels, and makes it suitable for both classification and regression tasks.

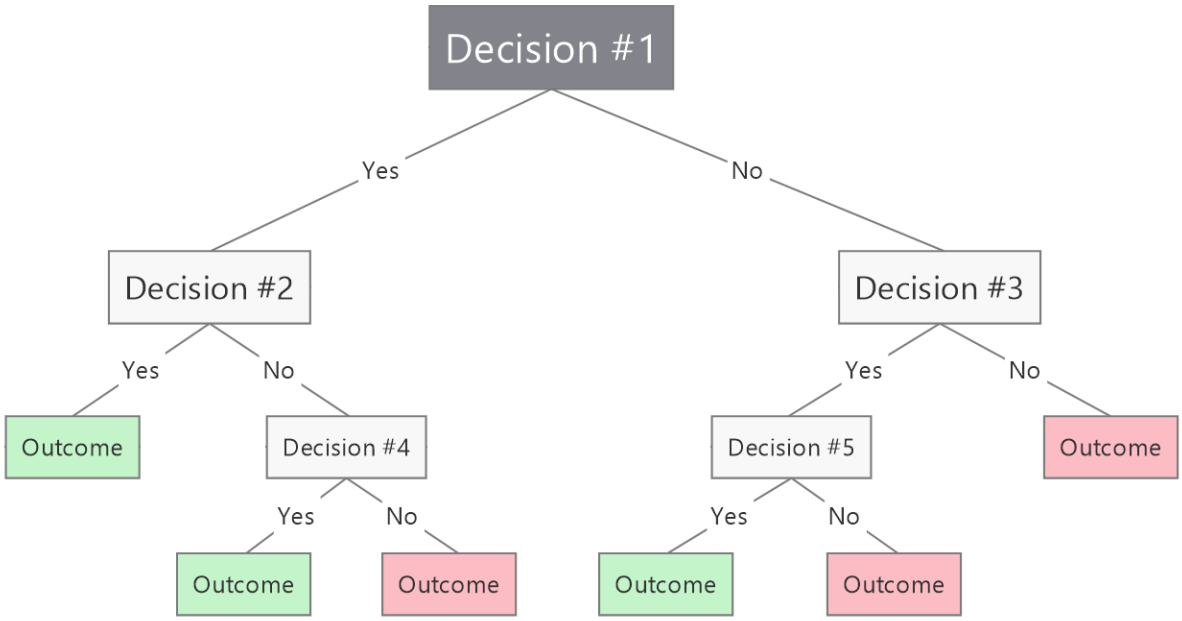


Figure 3.1: Architecture of decision tree model. The tree is made up of nodes that represent the various features in the dataset, and it starts at the root node, which represents the entire dataset.

One of the advantages of using decision tree is that it is not sensitive to noise, as the algorithm ignores irrelevant features and focuses only on the most informative ones. Additionally, decision tree provides a graphical representation of all the possible solutions to a problem based on given conditions. This makes it easier for users to understand the decision-making process and interpret the results.

Branches or sub-trees are formed by the splitting process, where each branch represents a subset of the original dataset. The sub-trees continue to split recursively until the algorithm reaches a point where further division is not possible, and a leaf node is formed.

Pruning is the process of removing unwanted branches from the tree to optimize its performance. It helps to reduce overfitting, which is the tendency of the algorithm to fit the training data too closely and perform poorly on new data. By removing unwanted branches, pruning helps to simplify the decision-making process and improve the accuracy of the algorithm.

In a decision tree, the root node is the parent node, and the child nodes are the nodes that result from the splitting process. The child nodes are further split into smaller sub-trees, which

form the branches of the decision tree. The leaf nodes are the final output of the algorithm, and they represent the class labels or predictions for the given input.

Decision tree is a powerful machine learning algorithm that is widely used in various domains, such as healthcare, finance, and marketing. It is intuitive, easy to use, and provides accurate results for many real-world problems.

### 3.4.3 Random Forest Classifier

Random forest is a technique implemented through bringing together a number of simple decision trees that operate as an ensemble system. Using the various trees in the ensemble, predictions are made based on the class, which is the response variable, and the class with the highest number of predictions is selected to be in the final predicted value. It works by creating a large number of decision trees, each based on a random subset of the features and training data. The algorithm then combines the output of these decision trees to make a final prediction.

Random Forest is a versatile algorithm that can be used for both classification and regression tasks. It is particularly useful for dealing with high-dimensional data and can handle missing values and noisy data. The algorithm also provides a measure of feature importance, which can be used for feature selection and to interpret the results.

One of the main advantages of using Random Forest is its ability to reduce overfitting, a common problem with decision trees. By creating multiple trees and averaging their output, random forest can provide more accurate predictions on new data than a single decision tree. Additionally, random forest can handle a large number of input features and can identify the most important features for the model.

Figure 3.2 shows the architecture of a random forest model. The algorithm starts by selecting a random subset of features from the input data and training multiple decision trees using these features. Each tree is trained on a different subset of the data, chosen randomly with replacement from the original dataset. This process is called bagging, and it helps to reduce the variance of the model.

During the testing phase, the algorithm combines the output of all the decision trees to make

a final prediction. To make a final prediction for a new sample, each tree in the random forest independently predicts the class label. The final prediction is then made based on the majority vote of the predictions from all the trees. That is, the class with the highest number of votes is chosen as the final prediction. For example, if a random forest consists of 100 decision trees and 70 of them predict the class label to be A, while 30 of them predict it to be B, then the final prediction will be A.

The majority voting mechanism helps to reduce the effect of over-fitting and increases the generalization ability of the model. It also makes the model more robust to noisy data and outliers.

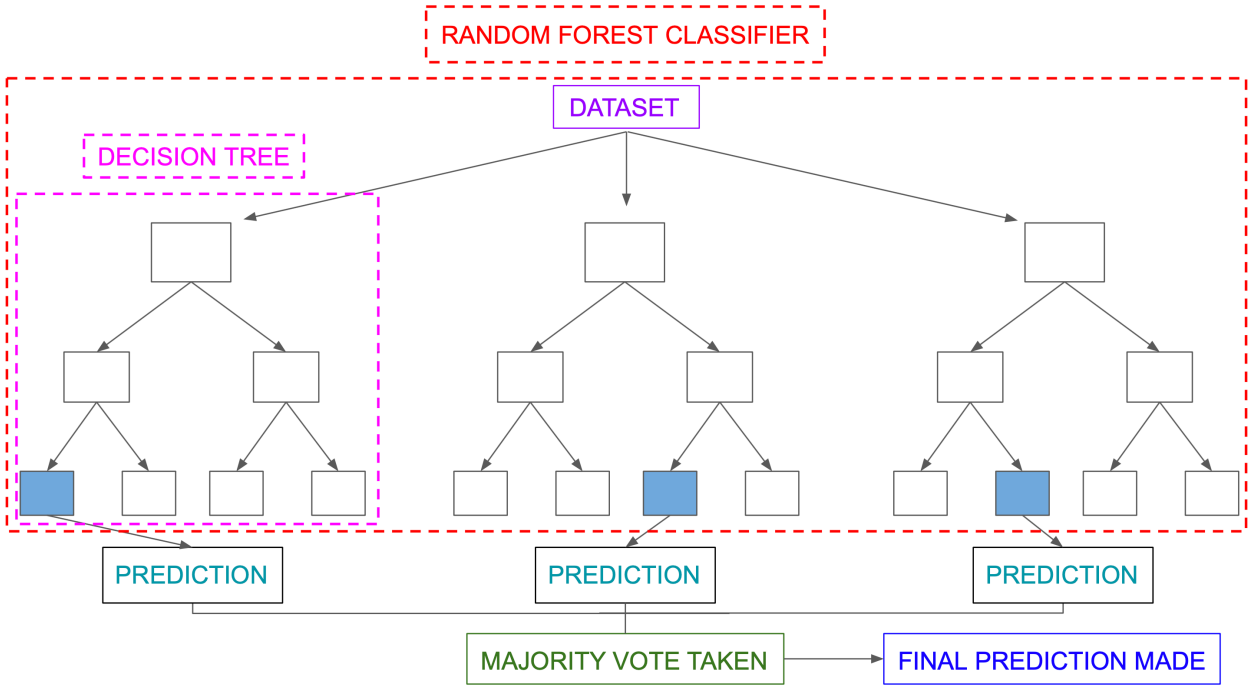


Figure 3.2: An illustration of the architecture of a random forest model, showing how it is constructed by combining multiple decision trees to make predictions. Each decision tree is built using a random subset of the features and a random subset of the training data, and the final prediction is made by aggregating the predictions of all the individual trees.

Random Forest has become one of the most popular machine learning algorithms due to its accuracy, robustness, and ability to handle a wide range of data types. It is used in various fields, such as finance, healthcare, and marketing, to make predictions and identify patterns in data.

### 3.4.4 Extreme Gradient Boosting

Extreme Gradient Boosting also known as the XGBoost, is a powerful and efficient gradient boosting framework that has been widely used in machine learning competitions and real-world applications. It is an extension of the Gradient Boosting algorithm and has gained popularity due to its speed and performance in handling large datasets (T. Chen and Guestrin 2016; Ke et al. 2017).

The XGBoost algorithm uses a combination of decision trees to build a predictive model. Each decision tree is built using a different random subset of the training data and the features, which helps to reduce overfitting and improve the generalization of the model (Ke et al. 2017). The algorithm then makes a final prediction by aggregating the predictions of all the decision trees using a weighted sum or majority voting scheme (T. Chen and Guestrin 2016).

The process starts with a naive model that assigns a probability of 0.5 to each class. Then, the errors of the naive model are calculated, and a new decision tree is trained to predict these errors. This tree is then added to the ensemble model, and the process is repeated with the updated ensemble as the new "naive" model. The goal of each new tree is to improve the classification accuracy of the ensemble model by predicting the errors of the previous trees.

During the training process, XGBoost optimizes a loss function that measures the difference between the predicted class probabilities and the actual class labels. The algorithm uses gradient boosting to minimize this loss function, which involves iteratively adding new decision trees to the ensemble in a way that reduces the error on the training data.

One of the key strengths of XGBoost is its ability to handle missing data and outliers effectively. It achieves this by using regularization techniques and incorporating the missing values as a separate category in the split criteria of the decision trees (T. Chen and Guestrin 2016).

Moreover, XGBoost provides several hyperparameters that can be tuned to improve the performance of the model. These include the learning rate, number of trees, maximum depth of the trees, and subsample ratio of the training instances (Ke et al. 2017).

This is a powerful algorithm that has proven to be effective in a wide range of machine learning tasks. Its ability to handle missing data and outliers, as well as its efficient implementation

and tuning options, make it a popular choice among practitioners and researchers alike.

### 3.4.5 Support Vector Classifier

Support Vector Classifier is a supervised machine learning algorithm that is widely used for classification tasks. The algorithm aims to find a hyperplane in an N-dimensional space that can effectively separate the data points into different classes. The dimension of the hyperplane is dependent on the number of input features. When there are two input features, the hyperplane is just a line, and when there are three input features, the hyperplane becomes a 2-D plane. As the number of features increases, it becomes challenging to visualize the hyperplane.

The figure 3.3 illustrates the possible hyperplanes for a support vector classifier. There can be several possible hyperplanes that can separate the data points into different classes. However, the objective of the support vector classifier algorithm is to identify a hyperplane that has maximum margin, which is the maximum distance between the data points of both classes. By maximizing the margin distance, the algorithm provides more reinforcement to classify the future data points with more confidence.

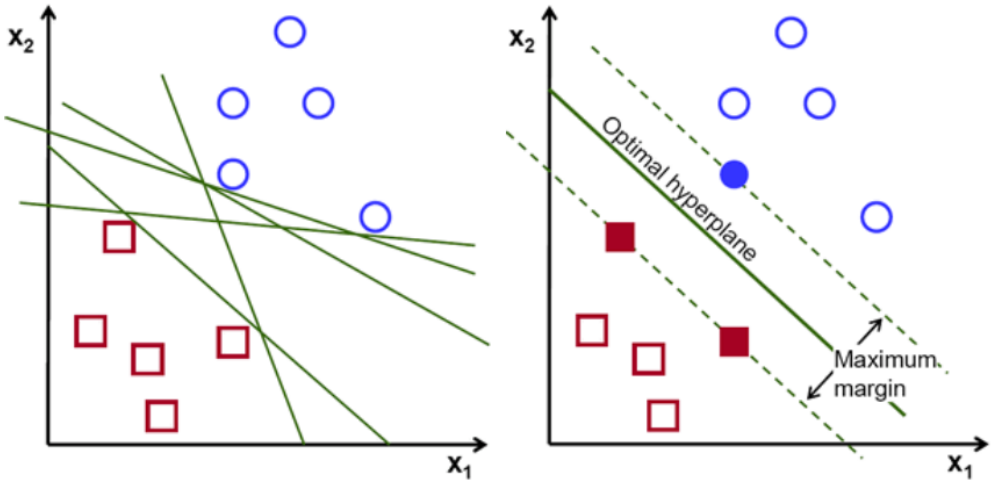


Figure 3.3: Illustration of the two possible hyperplanes for a support vector classifier in a two-dimensional feature space. The right plot represents the hyperplane that maximizes the margin between the two classes, while the left represents alternative hyperplanes that also separates the two classes but with a smaller margin.

The hyperplanes are decision boundaries that help classify the data points into different

classes. The data points that fall on either side of the hyperplane can be attributed to different classes. Therefore, the hyperplane that maximizes the margin provides better separation between the data points of different classes. The SVC algorithm can handle both linearly separable and non-linearly separable data.

One of the advantages of using this algorithm is that it is robust to noise in the dataset. The algorithm ignores the noisy data points and focuses on identifying the hyperplane that provides the maximum margin. Additionally, the SVC algorithm can handle high-dimensional data and is suitable for solving complex classification problems.

### 3.4.6 Artificial Neural Network

Artificial Neural Network is an information technology that imitates the human brain and nervous system. It is capable of representing knowledge through massive parallel processing and pattern recognition based on past experience or examples. ANNs have been widely studied and used in time series forecasting and classification tasks in business applications.

The theory of neural network computation is based on the assumption that information processing occurs at several simple elements called neurons, which are connected to each other through connection links. Each connection link has an associated weight that multiplies the signal transmitted. The neurons apply an activation function to their net input, which is the sum of the weighted input signals, to determine their output signal.

Through a replicative learning process and associative memory, the neural network model can accurately classify information as a pre-specified pattern. A typical neural network consists of a number of simple processing elements called neurons, which are connected to each other through directed communication links. Neural networks are usually modeled into one input layer, one or several hidden layers, and one output layer.

As shown in Figure 3.4, the components of neural network include neurons, connections and weights, and propagation function. Each artificial neuron has inputs and produces a single output that can be sent to multiple other neurons. The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons.

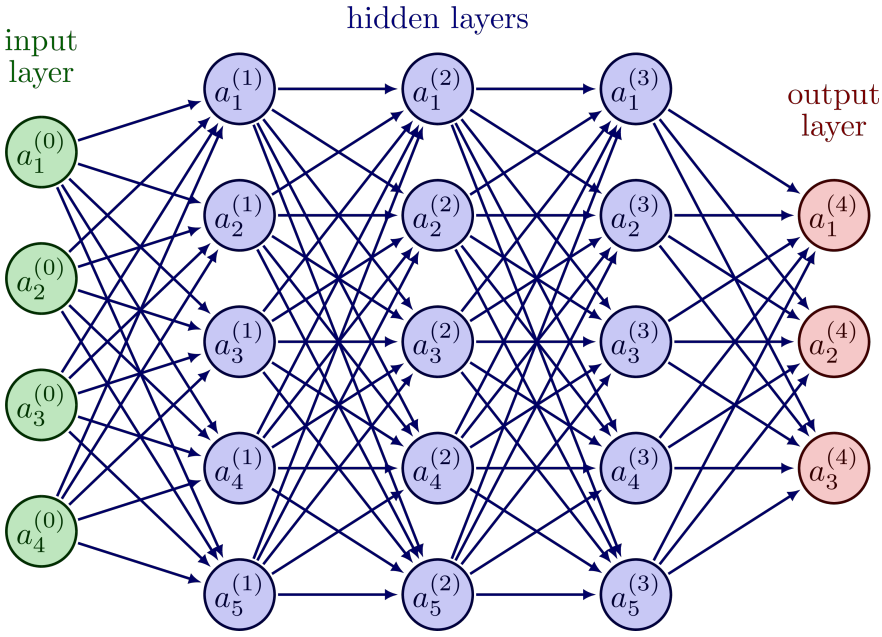


Figure 3.4: Architecture of a neural network. The neural network is composed of artificial neurons, connections, weights, and a propagation function. Each neuron has inputs and produces a single output which can be sent to multiple other neurons.

Neural networks have the ability to recognize high-level features, and this pattern recognition ability makes them a good alternative classification tool in medical applications.

### 3.5 Cross-Validation (CV)

Cross-validation is a crucial evaluation technique in machine learning that is used to assess the generalization ability of a model. The technique helps to detect flaws such as over-fitting or selection bias, and predict how the model would generalize to a different dataset. The technique involves re-sampling the data by using different portions of the data to test and train the model on successive iterations. K-fold cross-validation is one of the popular types of cross-validation used in machine learning.

In K-fold cross-validation, the training data of size N is randomly partitioned into K equal subsets. K-1 subsets are used as the training set, while the remaining subset is used as the test set. This process is repeated K times, with each iteration using a different fold for testing and the remaining K-1 folds for training. The mean of the values computed in the loop becomes the performance metric for the K-fold cross-validation. Figure 3.5 provides an illustration of the 5-Fold

Cross Validation mechanism.

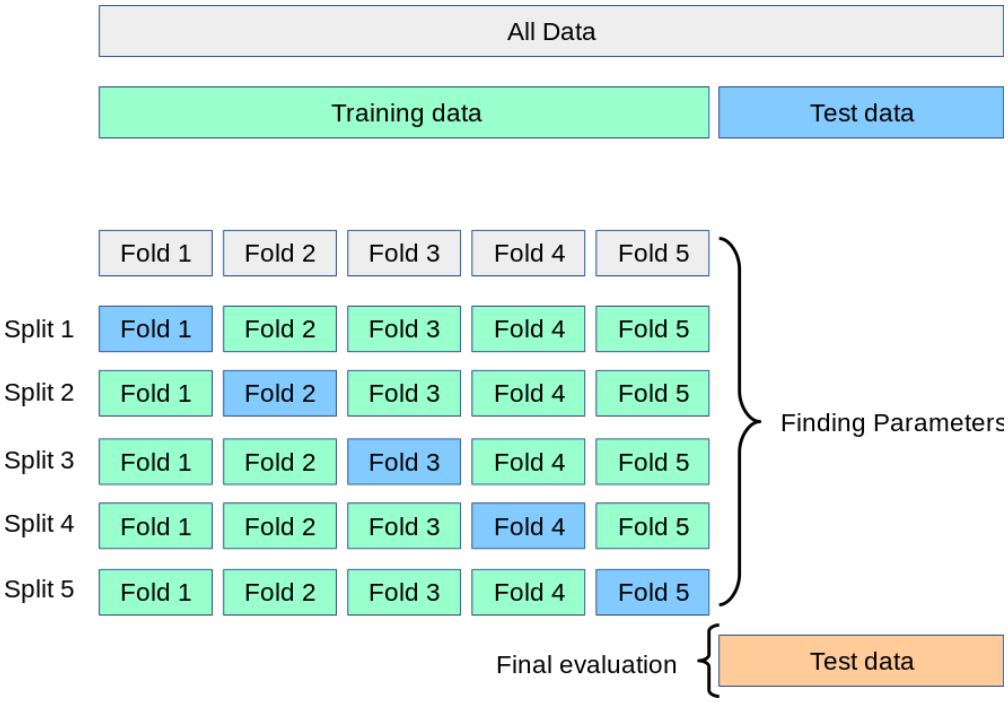


Figure 3.5: Illustration of 5-Fold Cross Validation. The dataset is randomly split into five equally sized folds. In each iteration, one of the folds is held out as the validation set, and the remaining four folds are used for training the model.

### 3.6 Recursive Feature Elimination with Cross-Validation (RFECV)

RFECV is a machine learning technique that is used to select the most important features from a dataset. The purpose of this technique is to reduce the dimensionality of the data while retaining the most relevant information for the prediction task.

This is a recursive algorithm that starts with all the features in the dataset and selects a subset of features that are most important for the prediction task. The algorithm then removes the least important feature from the subset and repeats the process until the desired number of features is reached.

The key difference between RFECV and traditional feature selection methods is that RFECV uses cross-validation to estimate the performance of the model with different subsets of features. This is important because some features may perform well in a particular subset, but not in others. By using cross-validation, RFECV ensures that the selected features are robust and can perform

well in different scenarios.

RFECV is commonly used in conjunction with linear models such as linear regression or logistic regression. These models are sensitive to the number of features used, and RFECV can help to identify the optimal number of features for the model. In addition, RFECV can also be used with non-linear models such as decision trees, support vector machines or random forest. The following are the steps involved in RFECV:

1. First, you need to choose a machine learning model that you want to use for feature selection. RFECV can be used with both linear and non-linear models.

2. Next, you need to divide the dataset into a training set and a testing set. The training set will be used to train the model, while the testing set will be used to evaluate the performance of the model.

3. You need to choose a performance metric that will be used to evaluate the performance of the model. This can be accuracy, precision, recall, or F1-score, depending on the nature of the problem.

4. The next step is to perform feature ranking using the chosen machine learning model. This involves fitting the model to the training set and evaluating the importance of each feature. The importance of each feature is determined based on how much it contributes to the overall performance of the model.

5. The least important feature is eliminated from the dataset, and the model is retrained using the remaining features.

6. The performance of the model is evaluated using cross-validation. This involves dividing the training set into multiple folds and training the model on different combinations of folds. The performance of the model is then evaluated on the testing set.

7. Steps 4-6 are repeated until the desired number of features is reached or the model's performance no longer improves.

Throughout this process, the optimal number of features is chosen based on the performance of the model on the testing set. This ensures that the selected features are robust and can perform well in different scenarios. Finally, the selected features are used to train the model on the entire dataset, and the performance of the model is evaluated on new data to ensure that it is generalizable.

One of the advantages of RFECV is that it is a data-driven approach that does not require any assumptions about the data or the model. It is also computationally efficient and can be used with large datasets. Another advantage is that RFECV can help to identify potential interactions between features, which can be useful for understanding the underlying relationships in the data.

In conclusion, RFECV is a powerful machine learning technique for feature selection that can improve the performance and interpretability of predictive models. By selecting the most important features and using cross-validation to estimate the model's performance, RFECV can help to reduce overfitting and ensure that the model is robust and generalizable to new data.

### 3.7 Balanced Bagging Classifier (with Bootstrapping Aggregation)

Balanced bagging (with bootstrapping aggregation) is a technique that combines bagging and boosting to improve the accuracy of classifiers in imbalanced datasets. It does not involve the generation of synthetic data.

As illustrated in Figure 3.6, the bagging component involves randomly selecting subsets of the majority and minority classes from the original dataset, with replacement. Multiple base classifiers are then trained on these bootstrapped samples, and their predictions are combined using aggregation methods such as majority voting or averaging.

The boosting component involves applying weights to the instances in the minority class, based on their classification errors. Instances that are misclassified by the base classifiers are assigned higher weights, while instances that are correctly classified are assigned lower weights. The base classifiers are then retrained on the modified dataset, with the weights serving as a form of oversampling for the minority class.

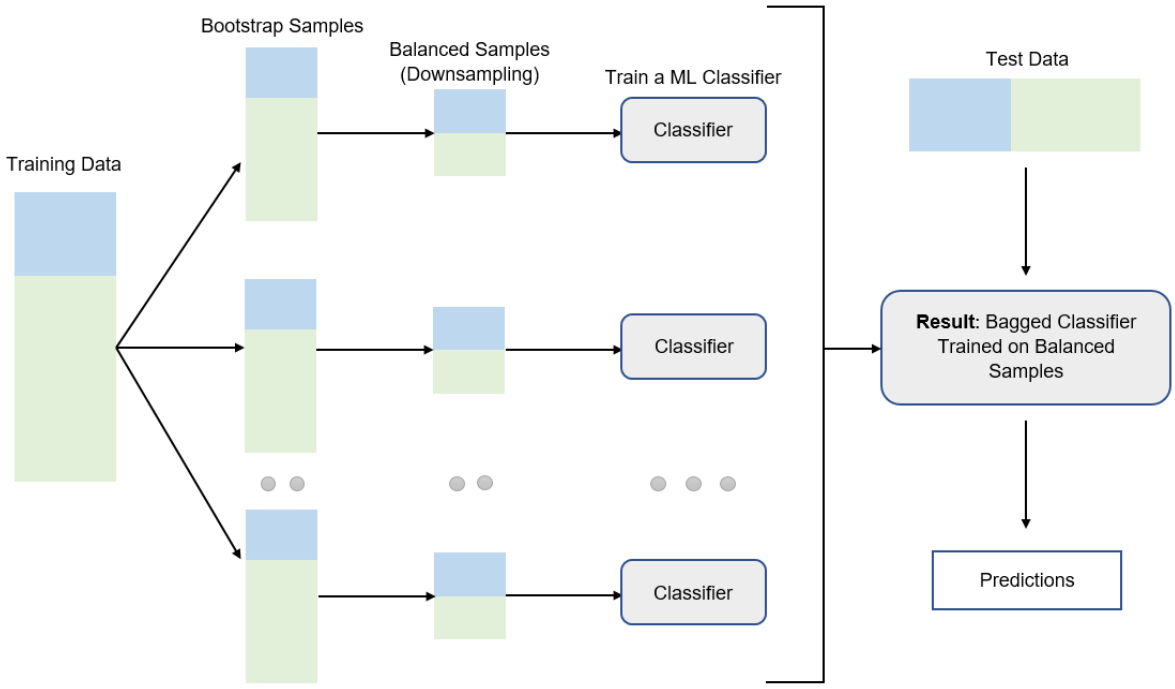


Figure 3.6: Illustration of the balanced bagging (with bootstrapping aggregation) technique for improving model performance in imbalanced datasets.

By combining bagging and boosting, this technique is able to improve the performance of the classifier by reducing the bias towards the majority class, while also addressing the issue of high variance in imbalanced datasets.

Balanced bagging has been shown to be effective in a variety of real-world applications, including fraud detection, medical diagnosis, and credit scoring. It is particularly useful in situations where the cost of misclassifying the minority class is high, such as in medical diagnosis where a false negative can have severe consequences.

### 3.8 Synthetic Minority Oversampling Techniques (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) is a popular oversampling method for imbalanced datasets in machine learning. SMOTE generates synthetic samples of the minority class by interpolating between pairs of existing minority class samples. This technique has been shown to improve the performance of classifiers on imbalanced datasets.

The entire process is shown in Figure 3.7. SMOTE works by randomly selecting a minority

class sample and finding its k nearest neighbors in feature space. The algorithm then selects one of these neighbors at random and generates a synthetic sample by interpolating between the two points. This process is repeated until the desired number of synthetic samples has been generated.

The interpolation process involves selecting a random point along the line connecting the two samples and creating a new sample at that point. By adding these synthetic samples to the original dataset, the SMOTE algorithm is able to balance the class distribution and improve the accuracy of classifiers.

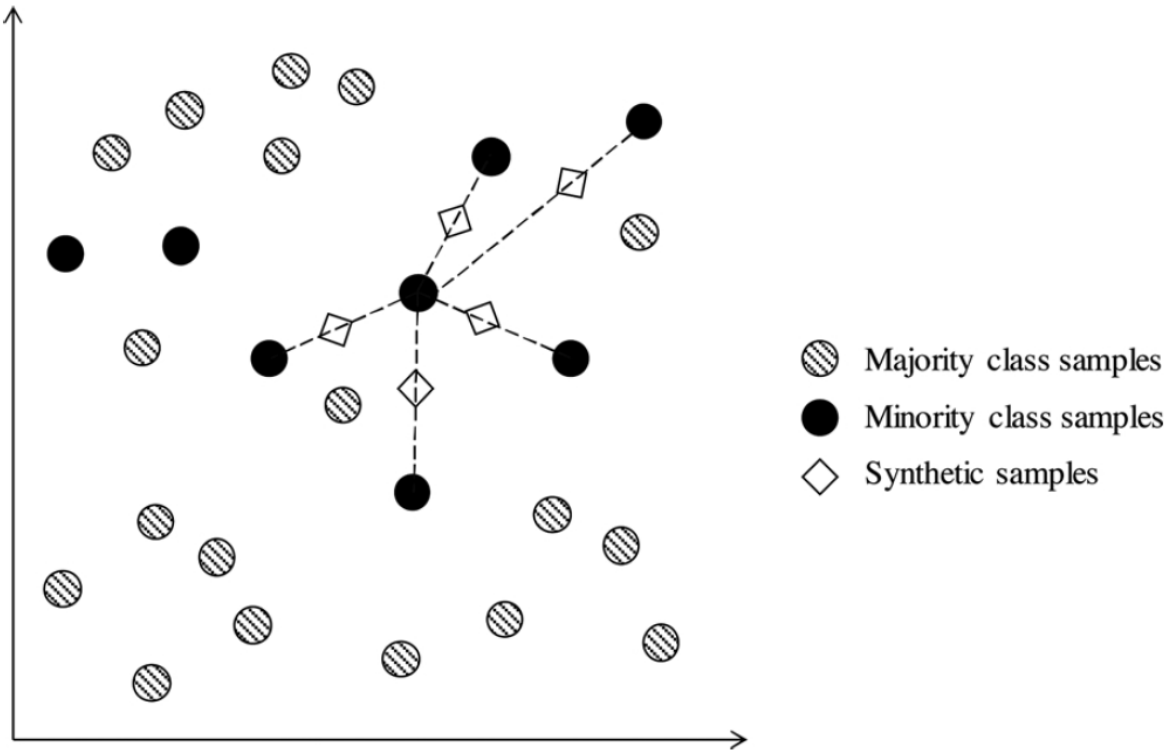


Figure 3.7: Illustration of the SMOTE technique for improving model performance in imbalanced datasets. The SMOTE algorithm generates synthetic samples of the minority class by interpolating between pairs of existing minority class samples.

### 3.9 Evaluation Metrics

In machine learning, it is essential to measure the performance of a model to determine whether it is accurate and reliable enough to be deployed. Different evaluation metrics are used for this purpose, and they provide information on how well the model performs, what needs to be improved, and what errors it makes.

One of the most common metrics for evaluating classification models is the confusion matrix. It is a table with two dimensions: "Actual" and "Predicted". Both dimensions have "True Positives", "True Negatives", "False Positives", and "False Negatives". The confusion matrix shows the number of correct and incorrect predictions made by the model, and it provides insights into the model's performance. The layout of a confusion matrix is shown in Figure 3.8.

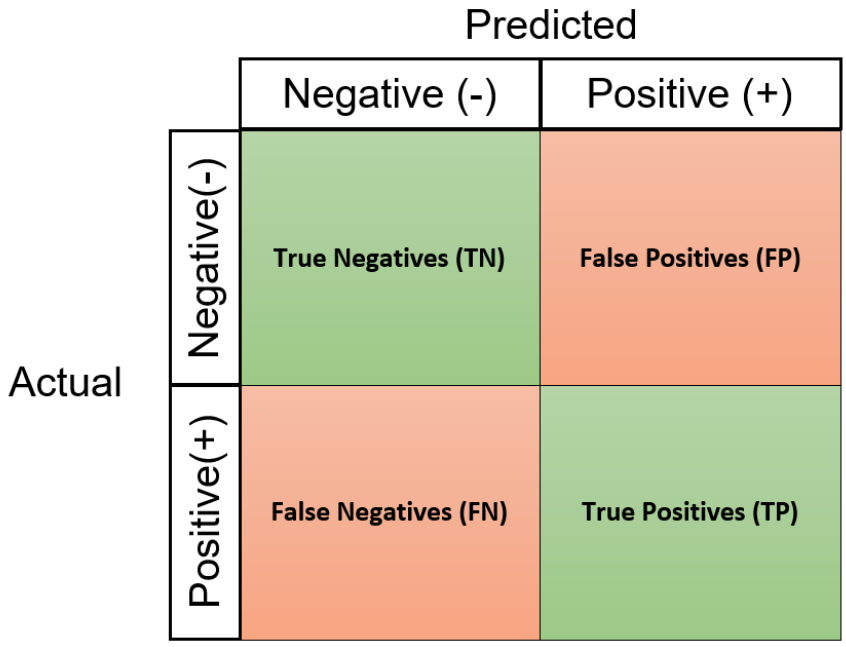


Figure 3.8: Layout of a confusion matrix. The matrix shows the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions made by the model. The actual class labels are shown in the rows, while the predicted class labels are shown in the columns.

The True Positive (TP) value represents the correctly predicted positive class outcome of the model, while the True Negative (TN) value represents the correctly predicted negative class outcome of the model. The False Positive (FP) value is the incorrectly predicted positive class outcome of the model, and the False Negative (FN) value is the incorrectly predicted negative class outcome of the model.

Sensitivity, also known as Recall, is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could be made by the model. Mathematically, it is given as TP/(TP + FN). Specificity measures the proportion of correctly identified negatives

over the total negative prediction made by the model. It is given as TN / (TN + FP).

Precision is a metric that quantifies the number of correct positive predictions made out of positive predictions made by the model. Precision calculates the accuracy of the True Positive. It is given as TP/(TP + FP). F1-Score is a metric that keeps the balance between precision and recall. It is often used when the class distribution is uneven. The F1-Score is defined as 2 * (Precision * Recall) / (Precision + Recall).

Evaluation metrics are crucial for measuring the performance of machine learning models. The confusion matrix and its various metrics, such as sensitivity, specificity, precision, and F1-Score, provide insights into the model's reliability, and help in identifying areas that need improvement.

### 3.10 SHapley Additive exPlanations (SHAP)

SHAP is a game-theoretic approach to explain the predictions of machine learning models. It was first introduced in a 2017 paper by Lundberg and Lee (Lundberg and S.-I. Lee 2017). The SHAP values provide a way to attribute the contribution of each feature to the final prediction made by a model.

SHAP values are based on the concept of Shapley values, which were developed in the context of cooperative game theory. Shapley values provide a way to distribute the payoff of a game among its players based on their contributions to the game. In the context of machine learning, the "game" is the prediction made by a model, and the "players" are the input features (Štrumbelj and Kononenko 2014). The SHAP framework is made up of model prediction as a sum of SHAP values of each feature as shown in Equation 3.2.

$$\phi_j^S = \frac{1}{K}\sum_{i=1}^{K}\left[f(x_i) - \frac{1}{|S_{i,j}|}\sum_{s\in S_{i,j}} f(z_{i,j}^s)\right] \tag{3.2}$$

where, $\phi_j^S$ is the SHAP value for feature $j$ on the subset of data $S$, $K$ is the number of samples in the dataset, $f$ is the model's prediction function, $x_i$ is the $i$th sample in the dataset, $z_{i,j}^s$ is the $i$th sample in the dataset with feature $j$ replaced by its value in the $s$th sample in $S_{i,j}$ and $S_{i,j}$ is the set

of samples that differ from $x_i$ only in the value of feature $j$. The equation calculates the difference between the model's prediction for a given sample $x_i$ and the average prediction over all possible combinations of values for feature $j$. This difference is then averaged across all samples in the dataset to obtain the SHAP value for feature $j$ on the subset of data $S$.

One of the key benefits of using SHAP values is that they provide a global and local interpretation of the model's predictions. The global interpretation helps to understand the overall importance of each feature in the model, while the local interpretation helps to understand the contribution of each feature to a specific prediction (Shrikumar, Greenside, and Kundaje 2017).

In summary, SHAP provides a powerful and flexible approach to interpreting the predictions of machine learning models. By attributing the contribution of each feature to the final prediction, SHAP values can provide valuable insights into the model's decision-making process (Molnar 2019).

# CHAPTER IV

# RESULTS AND DISCUSSIONS

## 4.1 Introduction

In this section, we presented the results obtained from the analysis of the data. The analysis consists of various techniques, including descriptive analysis, propensity score matching, machine learning analysis, results comparison, and diagnostic analysis.

The analysis aimed to provide insights into the potential protective effect of SGLT2 inhibitors against CKD incidence, to identify potential risk factors, and to develop predictive models that can aid in early detection and prevention of the disease. The results obtained will provide a better understanding of the factors that contribute to CKD, and will be useful in guiding healthcare practices and policies towards the prevention and management of this chronic condition.

## 4.2 Results and Discussions

### 4.2.1 Descriptive Analysis

Figure 4.1 shows that, out of the 19,786 patients who developed Chronic Kidney Disease (CKD), a significant majority of 76.7% were aged 65 years and above. A proportion of 19.7% fell in the age range of 45 to 64 years. Only a very small percentage of 3.6% were in the age range of 18 to 44 years. These findings may suggest that the risk of developing CKD increases with age, with older individuals being at a higher risk.

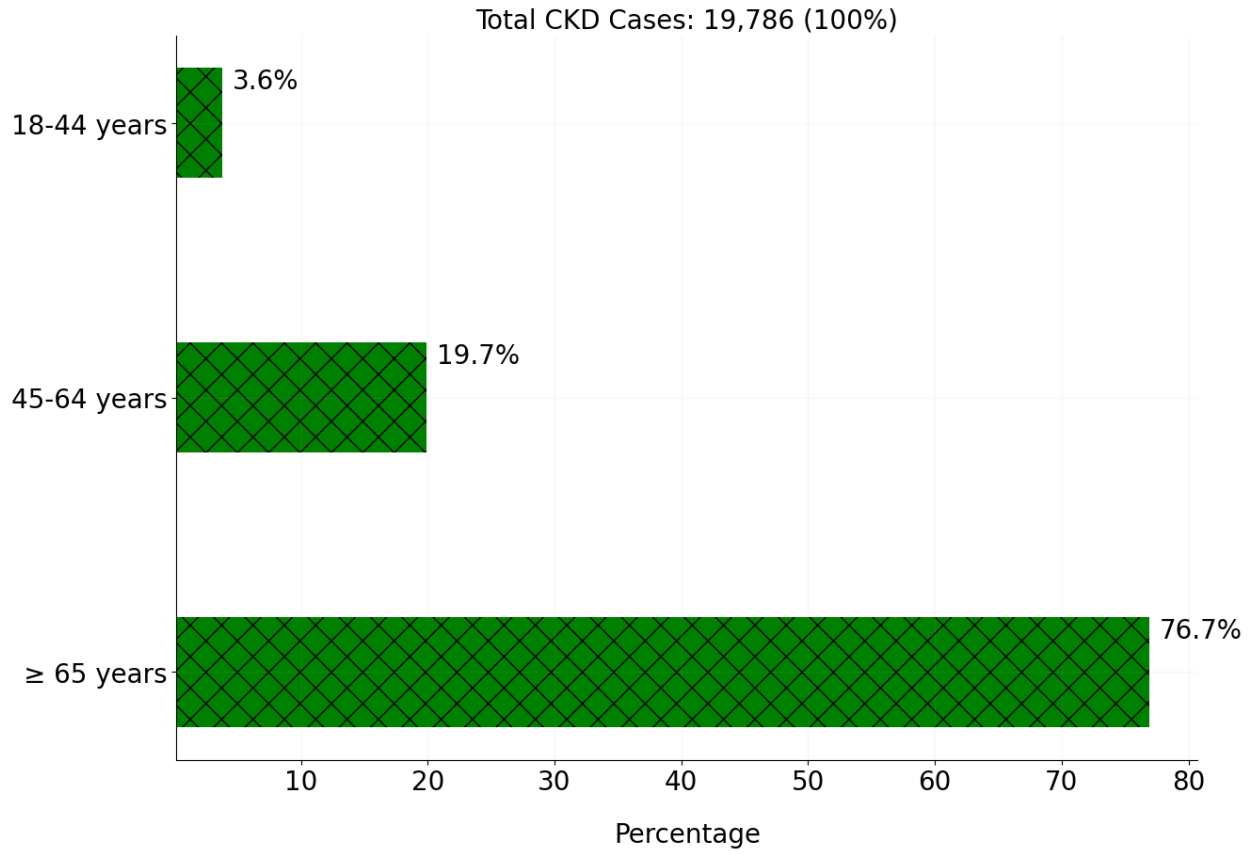


Figure 4.1: Age distribution of patients with Chronic Kidney Disease. The figure shows the percentage of CKD cases in different age brackets. The majority of CKD cases were observed in patients aged 65 years old and above (76.7%), followed by those aged 45-64 (19.7%), and a small percentage in the 18-44 age group (3.6%).

According to various studies, certain racial and ethnic groups, such as Black or African Americans, Hispanic/Latino Americans, and Native Americans, are more susceptible to CKD compared to other groups. These disparities in incidence rates may be attributed to various factors, including genetic predispositions, differences in healthcare access and quality, lifestyle factors, and socioeconomic factors. Therefore, it is important to consider race as a significant factor when studying the incidence and prevalence of CKD.

In this work, the race of the patients was considered to be a crucial demographic factor that needed to be included. Previous literature suggests that race is an important issue when it comes to the incidence of Chronic Kidney Disease. The following output displays the results of CKD cases in each racial group.

Based on the data presented in Figure 4.2, it was observed that CKD is more prevalent in Black or African American individuals, with a percentage of 60.5%. This indicates that a larger proportion of Black or African American individuals in the study were reported to have experienced CKD, compared to individuals of other race.

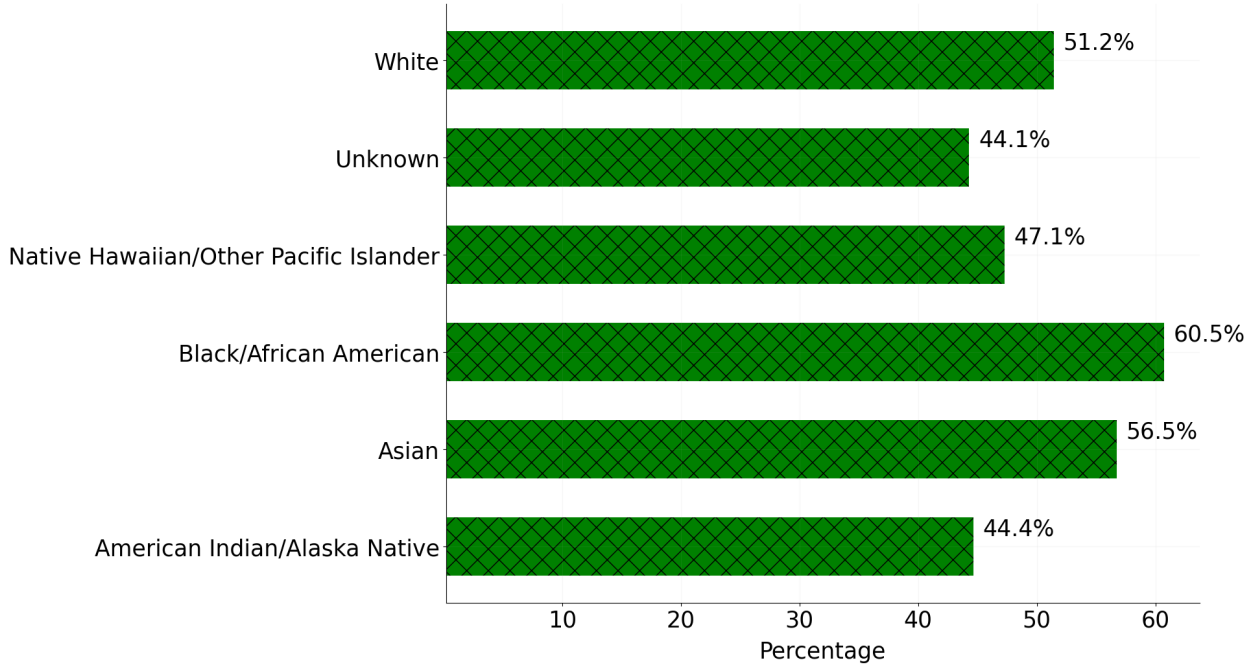


Figure 4.2: Percentage of CKD cases in different racial groups: The figure underscores the significance of race as a potential risk factor for CKD and emphasizes the need for targeted interventions to address health disparities.

The study investigated the prevalence of CKD across different ethnic groups, including Hispanic or Latinos, Non-Hispanicor Latinos, and Unknown ethnicity. A barchart was plotted to visualize the distribution of CKD cases among the groups.

Figure 4.3 revealed that out of the total Hispanic or Latino population considered in the study, 50.4% developed CKD. Similarly, 51.6% of the Non-Hispanic or Latino group developed CKD. In the Unknown ethnicity group, 46.3% of the population developed CKD.

These findings suggest that there is no significant difference in the susceptibility to CKD between Hispanic or Latinos and Non-Hispanic or Latinos. However, the prevalence of CKD in both groups is higher than that of the general population. Further research is needed to understand the underlying factors contributing to the high prevalence of CKD in these groups.

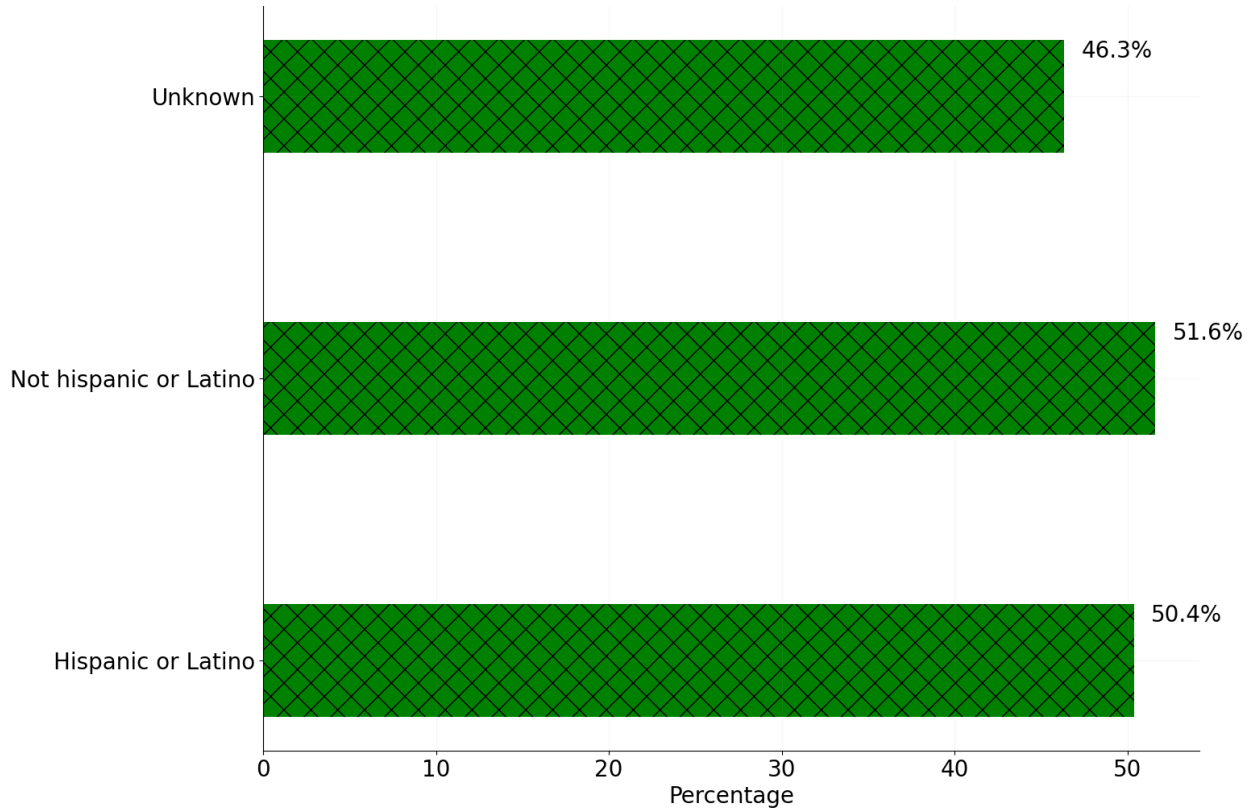


Figure 4.3: Ethnic distribution of CKD cases: This figure shows the percentage of CKD cases in different ethnic groups, based on the available data.

The incidence of CKD in patients who took SGLT2 inhibitors and those who did not was a crucial aspect of the study. Out of the total $38,776$ patients considered, only $1,450$ $(3.74\%)$ took the SGLT2 inhibitor, while $37,326$ $(96.26\%)$ did not take it, possibly due to several reasons known to the patients. Figure 4.4 shows the outcome of CKD cases in these two groups.

It was observed that only 213 $(14.7\%)$ out of the $1,450$ patients who took the inhibitor developed CKD, while $1,237$ $(85.3\%)$ did not develop the disease after taking the inhibitor. On the other hand, for patients who did not take the inhibitor, the incidence of CKD was found to be significantly high (about 52.44%). This could be an indication that the inhibitor has a protective effect on the outcome of CKD. However, further investigation was conducted in subsequent subsections to delve deeper into this matter.

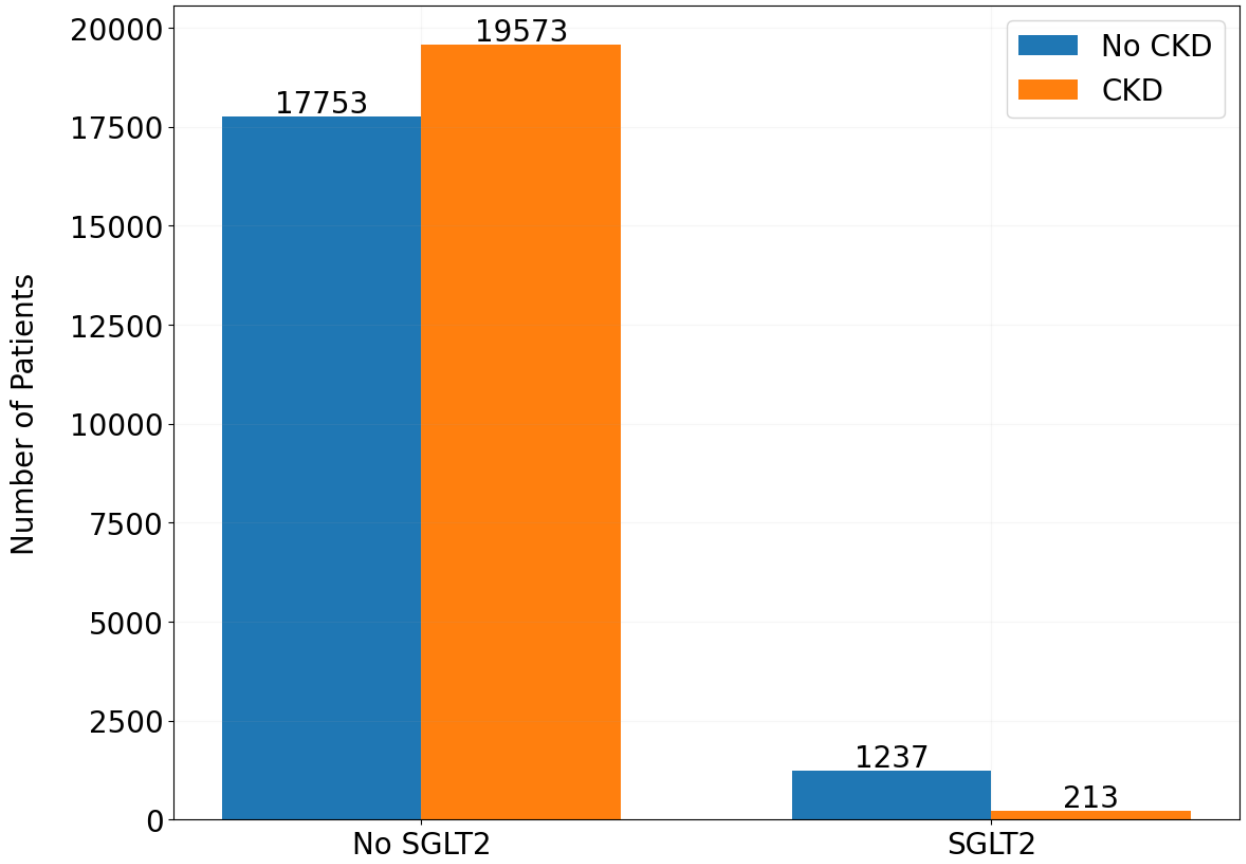


Figure 4.4: Distribution of Chronic Kidney Disease Cases by Sodium Glucose Cotransporter 2 (SGLT2) Inhibitor Intake.

Table 4.1 provides insights into the various demographic groups and the number of patients who took the SGLT2 inhibitor. Additionally, it presents the mean age of patients who took the inhibitor and those who did not. From the table, it is evident that across all demographic groups, the proportion of patients who took the inhibitor was significantly lower compared to those who did not.

This observation suggests that the use of SGLT2 inhibitors was not prevalent among patients belonging to diverse demographic groups, possibly due to reasons such as availability, cost, and prescribing practices of healthcare providers.

Table 4.1: Distribution of SGLT2 Intake by Demographic Characteristics of the Patients

| **Demographic** | | **No SGLT2** | **SGLT2** |
|---|---|---|---|
| Sex, n (%) | Male | 18897(50.63%) | 825(56.90%) |
| | Female | 18424(49.36%) | 625(43.10%) |
| | Unknown | 5(0.01%) | 0(0.00%) |
| Race, n (%) | White | 33470(89.67%) | 1326(91.45%) |
| | Black or African American | 1167(3.12%) | 31(2.14%) |
| | Unknown | 2613(7.00%) | 88(6.07%) |
| | Others | 76(0.21%) | 5(0.34%) |
| Ethnicity, n (%) | Hispanic or Latino | 133(0.36%) | 8(0.55%) |
| | Not Hispanic or Latino | 33172(88.87%) | 1339(92.34%) |
| | Unknown | 4021(10.77%) | 103(7.10%) |
| Age, Mean (SD) | | 70.24(12.85) | 64.49(10.72) |

## 4.3 Propensity Score Matching

In order to evaluate the effect of SGLT2 on CKD, we used propensity score matching (PSM) to reduce selection bias in our observational study. This involved estimating a propensity score for each participant, which was used to match individuals who received SGLT2 with those who did not based on their likelihood of receiving treatment. We then used the Average Treatment Effect (ATE) and Odds Ratio (OR) to estimate the effect of SGLT2 on CKD outcomes, while controlling for any differences between the treatment and control groups.

### 4.3.1 Propensity Scores Estimation and Matching

To estimate the propensity scores, we used logistic regression to model the probability of receiving SGLT2 treatment based on a set of covariates. We included the following covariates in the model: age, sex (Female), race (black or African American), race (White), race (Unknown), ethnicity (Not Hispanic or Latino) and ethnicity (Unknown). The logistic regression model was fit, and the resulting predicted probabilities were used to estimate the propensity scores for each

participant. Equation 4.1 illustration the nature of the logit model.

$$\log\left[\frac{\mathrm{P(T=1)}}{1-\mathrm{P(T=1)}}\right] = \beta_0 + \beta_1(\text{Age}) + \beta_2(\text{Female}) + \beta_3(\text{Black}) + \beta_4(\text{Unknown Race}) + \beta_5(\text{White}) + \beta_6(\text{not Hispanic or Latino}) + \beta_7(\text{Unknown Ethnicity}) \quad (4.1)$$

where: $\mathrm{P(T=1)}$ is the probability of receiving SGLT2 treatment, $\beta_0$ is the intercept term and $\beta_1 - \beta_7$ are the coefficients for each covariate (age, female, black or African American, white, unknown race, not Hispanic or Latino, and unknown ethnicity).

One essential consideration when using propensity score matching is the overlap of the propensity score distributions between the treatment and control groups. This is because if there is little overlap between the two distributions, it may be difficult or impossible to find suitable matches for some individuals in the treatment group, which could bias the results, and the two groups may not be comparable in clinical settings. A lack of overlap in the distribution of propensity scores indicates that the characteristics of the two groups are substantially different, which can lead to biased estimates of treatment effects. As a result, it is important to examine the distribution of propensity scores of the two groups, as well as to assess the extent of overlap before using the estimated propensity scores for further analysis.

Therefore, we examine the distribution of propensity scores and the degree of overlap before using them for further analysis to ensure that the treatment and control groups are comparable. It was observed in Figure 4.5 that the distributions of propensity scores between the SGLT2 and non-SGLT2 groups overlap sufficiently. Since there is significant overlapping between the distributions of propensity scores of the two groups, it indicates that the treatment and control groups are comparable with respect to the observed covariates.

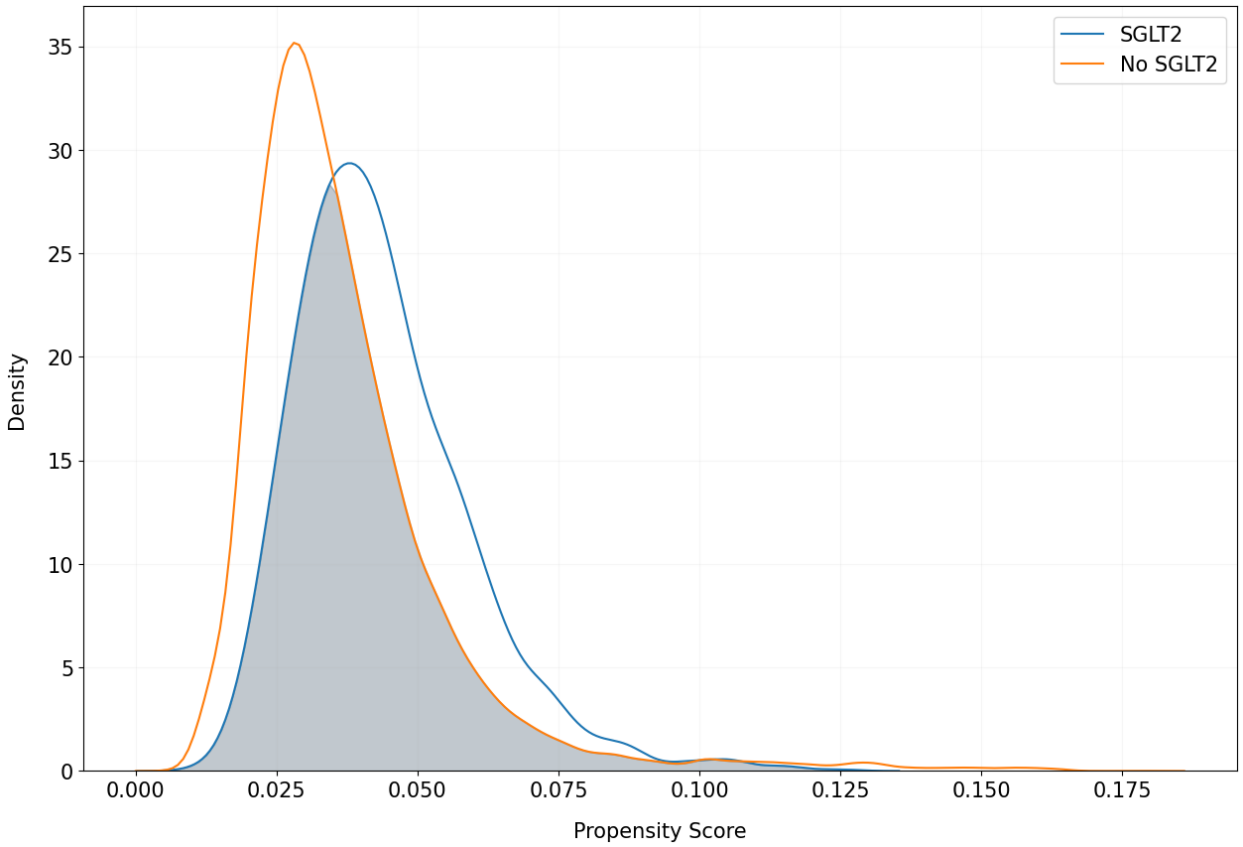


Figure 4.5: Density plot of propensity scores for the SGLT2-treated and untreated groups. The plot shows the distribution of estimated propensity scores for each group.

We then used propensity score matching with K-nearest neighbor (KNN) method to balance the covariates between the SGLT2-treated and untreated groups. Specifically, we performed matching with 10 nearest neighbors using the estimated propensity scores from the logistic regression model. The nearest neighbor matching algorithm goes through the potential matches in the untreated (non-SGLT2) samples and selects the closest unmatched subject in terms of similar propensity score to match the treated (SGLT2) subject. However, the KNN matching is at risk of bad matches when the closest neighbor is far away so we set a caliper of 0.25 times the standard deviation of the PS. The caliper imposed a tolerance level on the maximum PS distance. Only NNs within the caliper size can be matched.

This resulted in a matched sample of patients that were balanced on age, sex, race, and ethnicity between the two treatment groups in Figure 4.6. The matching process was well-defined

and consistently produced high-quality matches across different effect sizes. This approach ensured that any observed differences in the outcome of interest, such as chronic kidney disease (CKD), could be attributed to the treatment itself and not to differences in the baseline characteristics of the patients.

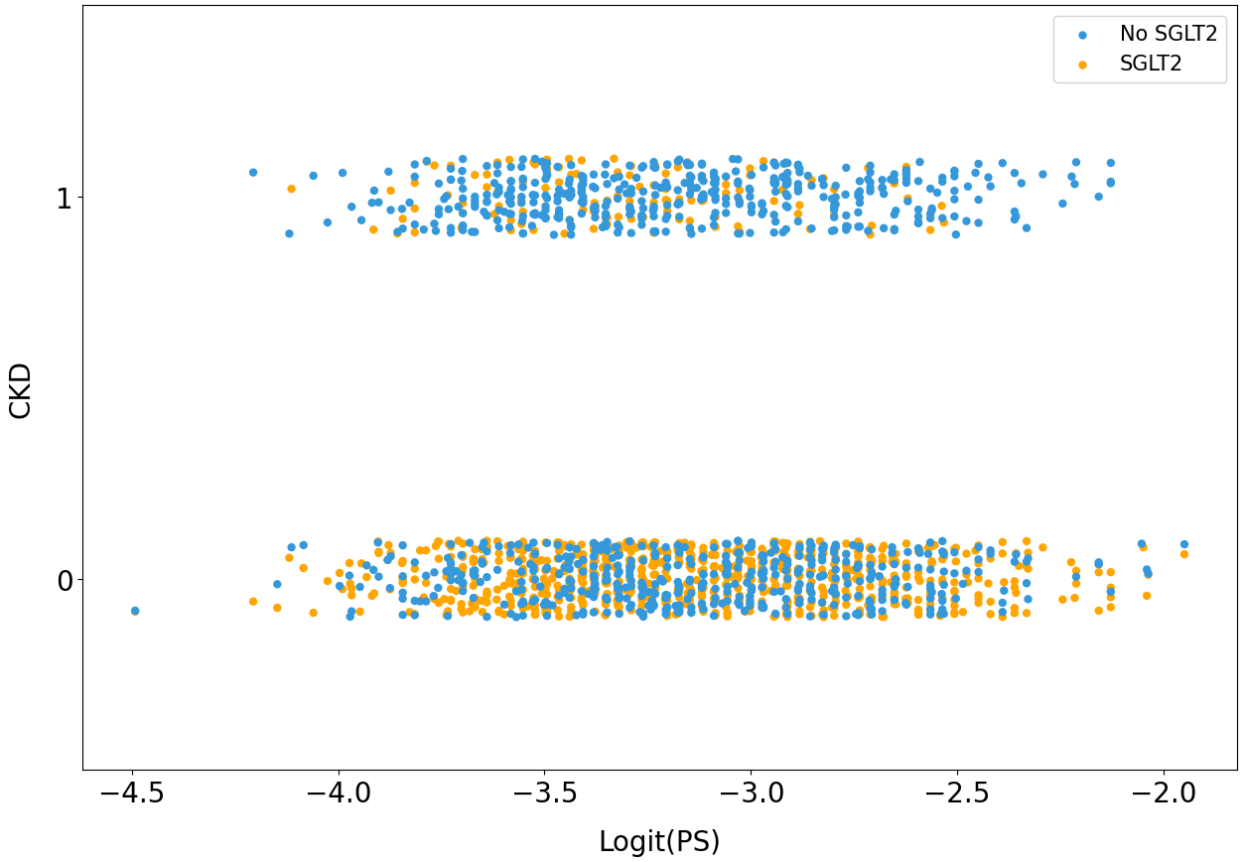


Figure 4.6: The figure shows the distribution of the logit of the propensity scores for the SGLT2 and non-SGLT2 groups, with some degree of overlaps after the matching.

The next approach, is to carefully assess goodness of the matching by performing diagnostics checks in the next subsection.

### 4.3.2 Balanced Diagnostics

After the matching, the effectiveness of the matching can be judged by the degree of balance in all the measured baseline covariates between the two groups after matching. The standardized mean difference (SMD) that is not affected by the samples size and represents properties of the sample, is proposed as an adequate method to assess this balance (Ho et al. 2007). Hypothesis

testing and p-values are not recommended to check the balance between groups after propensity score matching, since a failure to reject the null hypothesis does not guarantee successful balance of covariates between the two groups. Notwithstanding, the level of significance by itself does not predict effect size. Unlike significance tests, effect size is independent of sample size. Statistical significance, on the other hand, depends upon both sample size and effect size. For this reason, the interpretation of p-values are considered to be confounded because of their dependence on sample size (Sullivan and Feinn 2012).

Based on the results in Table 4.2, there were significant differences in the covariates before the matching, with SMD greater than 0.1, a threshold recommended for declaring imbalance (Stuart, B. K. Lee, and Leacy 2013). A larger effect size suggests a stronger association between the covariate and the outcome, and may indicate a greater potential for bias or confounding in the study results.

Table 4.2: Standardized Mean Difference and Covariate Description before and after Matching

| | **Before PSM** | | | **After PSM** | | |
|---|---|---|---|---|---|---|
| **Covariate** | SGLT2 (n = 1450) | No SGLT2 (n = 37326) | SMD | SGLT2 (n = 901) | No SGLT2 (n = 901) | SMD |
| Sex (Female), (%) | 43.10 | 49.36 | −0.125 | 47.50 | 47.39 | 0.002 |
| Race (Black), (%) | 2.14 | 3.13 | −0.057 | 3.44 | 3.33 | 0.006 |
| Race (White), (%) | 91.45 | 89.67 | 0.059 | 86.35 | 86.57 | −0.006 |
| Race (Unknown), (%) | 6.07 | 7.00 | −0.037 | 9.77 | 9.99 | −0.007 |
| Not Hispanic/Latino, (%) | 92.34 | 88.87 | 0.111 | 87.79 | 87.46 | 0.010 |
| Ethnicity (Unknown), (%) | 7.10 | 10.77 | −0.119 | 11.32 | 11.88 | −0.017 |
| Age, (mean) | 64.49 | 70.24 | −0.449 | 63.72 | 63.62 | 0.007 |

After the matching, however, there was a decrease in the standardized mean differences for all the covariates, with each covariate having a value below the threshold. The difference in averages between the two groups being compared is relatively minor in magnitude across all the covariates under consideration, indicating a small effect size. This suggests that the groups being

compared are comparable with regard to the covariates being measured. Figure 4.7 shows the graphical representation of the SMDs before and after and the balanced threshold point.

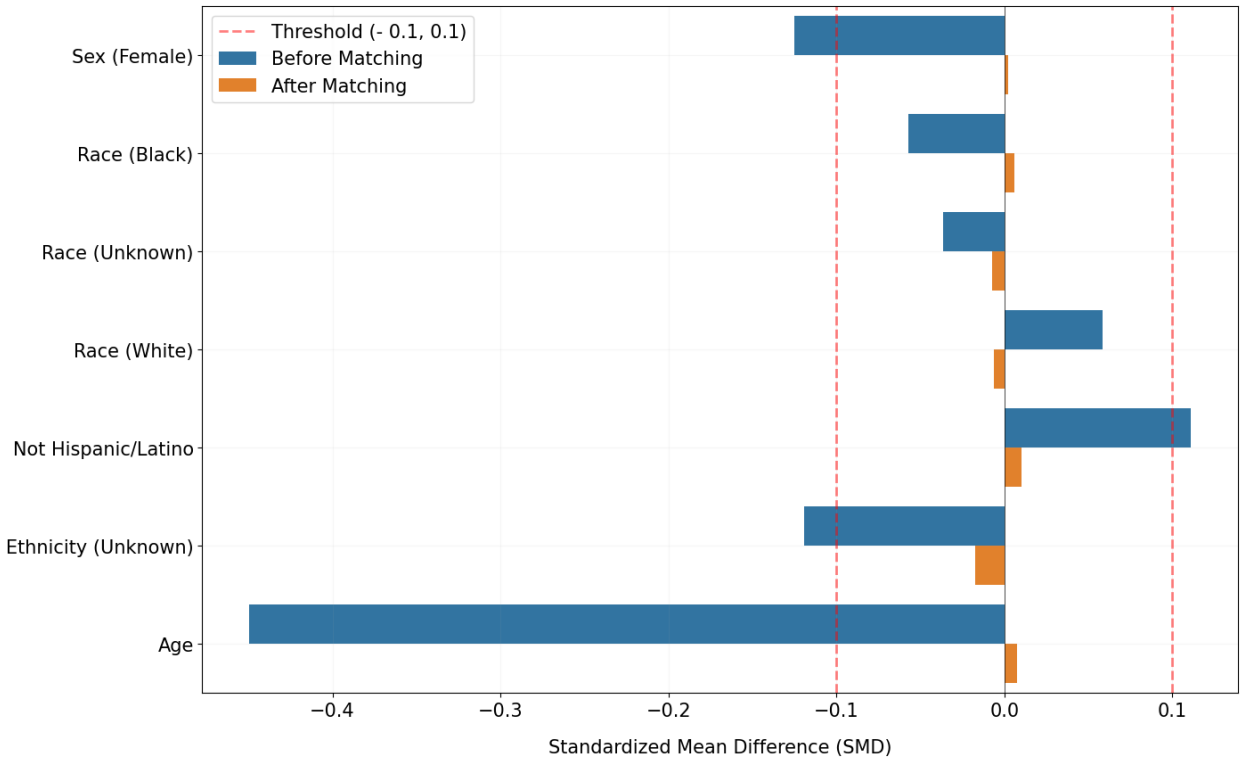


Figure 4.7: Plot of Standardized Mean Difference of Covariates Before and After Matching.

The Violin plot in Figure 4.8 shows the distribution of propensity scores in the two groups before and after matching. It was observed that before PSM, there was a significant difference between the distribution of propensity scores in the SGLT2 and Non-SGLT2 group. However, after PSM, the distribution of propensity scores in the two groups appears to be more similar. The violin plot suggests that the PSM process was effective in reducing the imbalance in the distribution of propensity scores between the two groups.

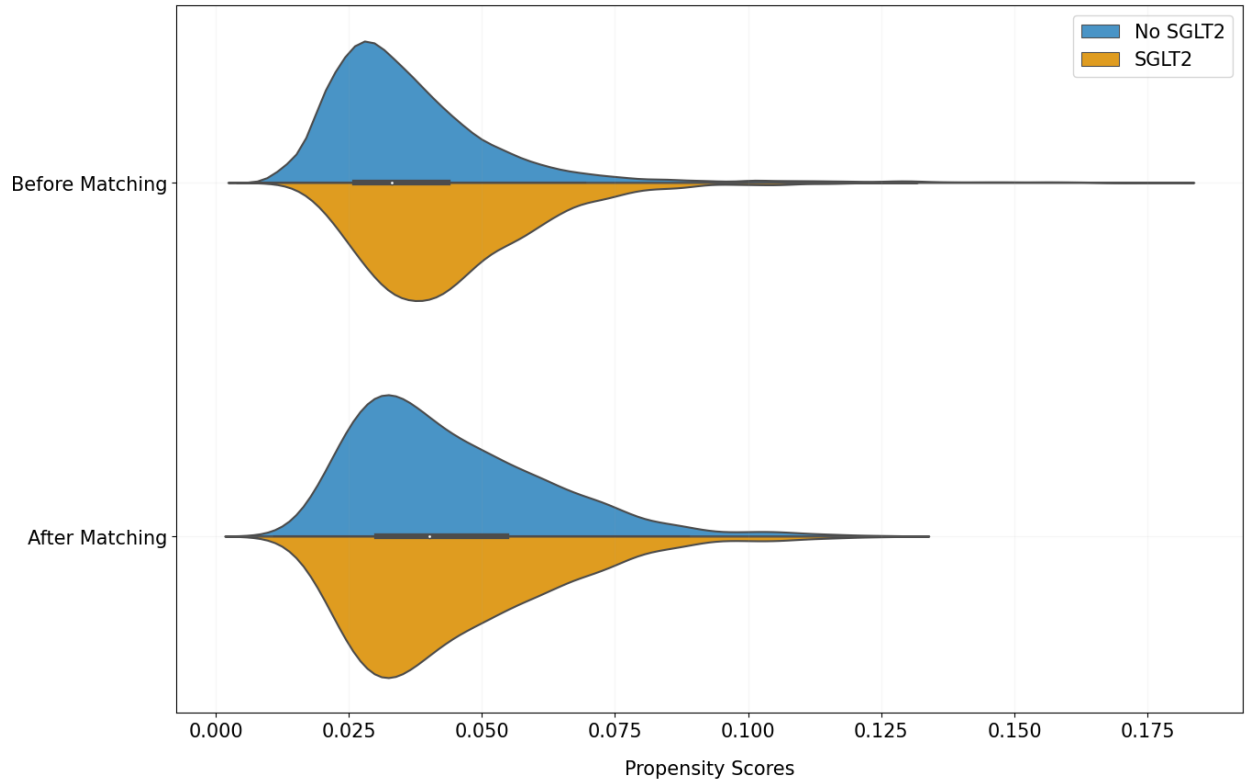


Figure 4.8: Distribution of Propensity Scores in the Treatment (SGLT2) and Control (No SGLT2) Group Before and After Matching.

Aside the SMD results, we examine the balance in the age graphically in Figure 4.9. It shows a balance in the age between the two groups after the matching as explained in the previous paragraphs.

Our propensity score matching analysis indicated that the balance in age between the two groups was satisfactory, as confirmed by both statistical measures (SMD results) and graphical methods (Figure 4.9). This finding suggests that any observed differences in age between the two groups were likely due to chance rather than systematic bias. By ensuring balance in age, our study is better able to isolate the effect of the intervention/exposure being studied and reduce the likelihood of confounding. Therefore, our results are more robust and reliable, allowing us to draw more accurate conclusions about the relationship between SGLT2 and the outcome of CKD. However, the Kolomogorov-Smirnov (KS) and the Wilcoxon rank sum non-parametric tests for equality of distributions were used to verify these observations.

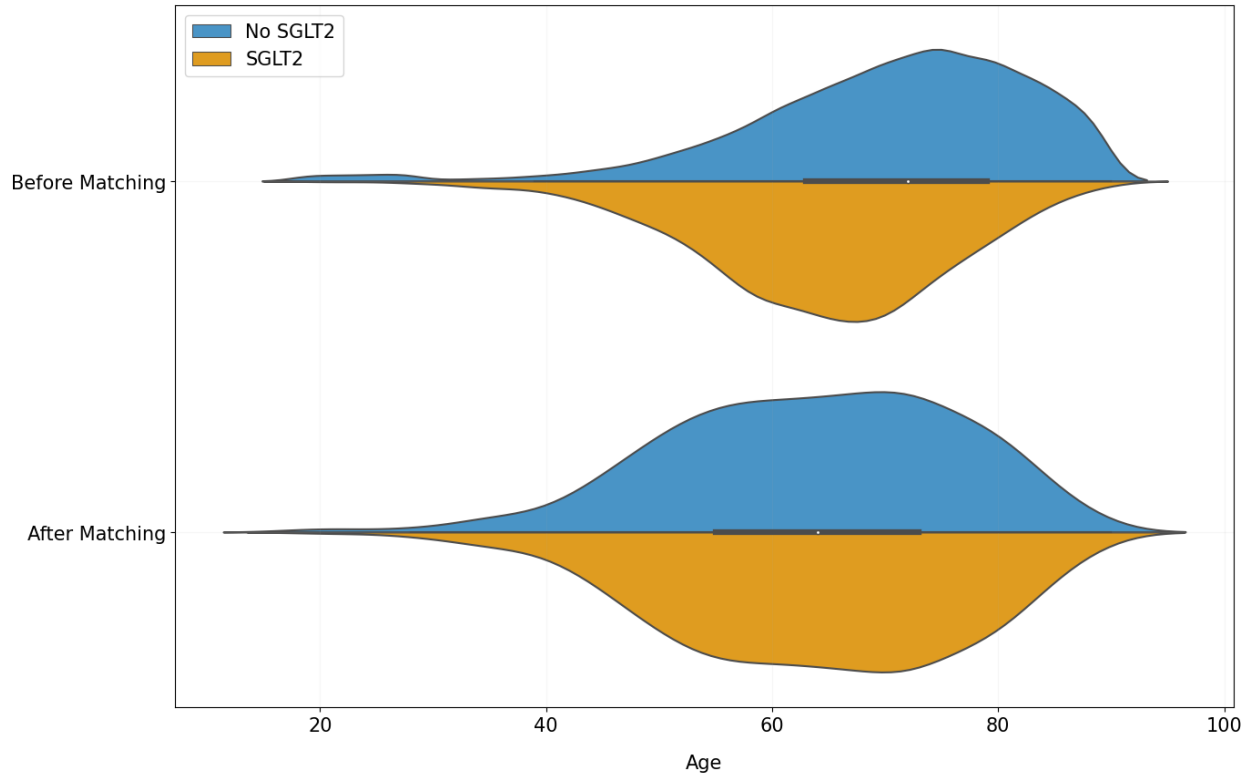


Figure 4.9: Distribution of Age in the Treatment (SGLT2) and Control (No SGLT2) Group Before and After Matching.

We carry out the KS test at $\alpha = 5\%$, against the null hypothesis, $H_0$ that the distribution of propensity scores between the two groups are equal.

The decision rule for this test is that, if the p-value is greater than or equal to the significance level ($\alpha$), then we fail to reject the null hypothesis and conclude that there is not enough evidence to suggest that the two samples are drawn from different distributions, otherwise reject $H_0$ and conclude that there is enough evidence to suggest that the two samples are drawn from different distributions.

As depicted in Figure 4.10, the KS test results from Table 4.3 confirmed a significant difference in distribution between the SGLT2 and Non-SGLT2 groups before matching, with a p-value less than 0.05, this suggests that the groups were not balanced with respect to their propensity scores and age distribution. However, the test results show no significant difference in the distribution of propensity scores and age between the groups after matching, with a p-value $\geq 0.05$. This con-

firms that the matching process was successful in reducing the imbalance in the propensity score distribution between the groups.

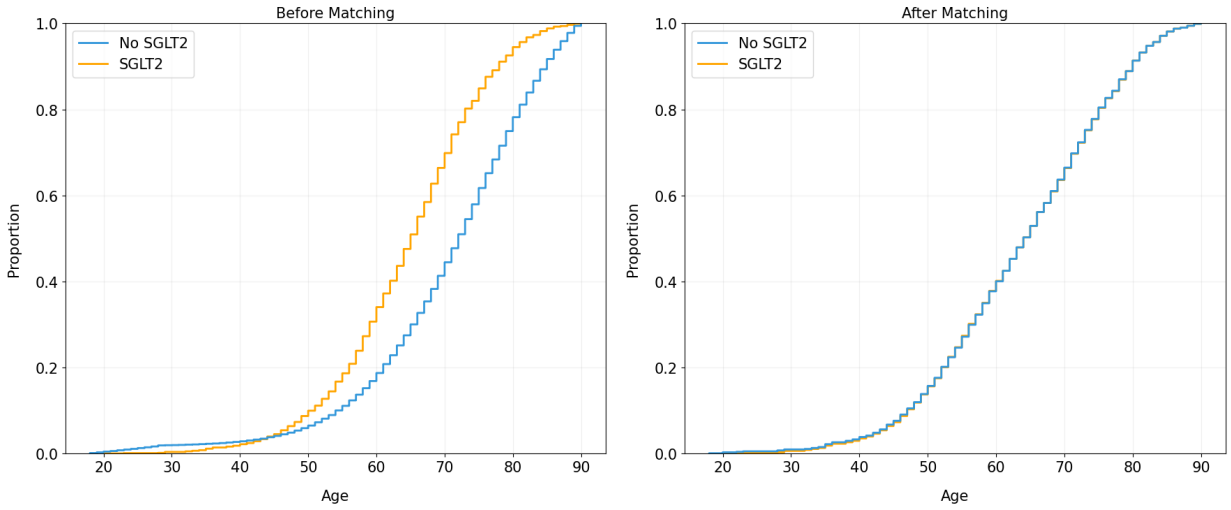


Figure 4.10: Empirical Cumulative Distribution Functions (ECDFs) comparing the age distribution of SGLT2 inhibitor users and non-users before and after PSM. The left plot shows the distribution of age before PSM, while the right plot shows the distribution after PSM. The ECDFs provide a visual representation of the cumulative proportion of individuals in each group at a given age, allowing for a comparison of the age distributions between groups before and after matching.

Table 4.3: KS Test Results for Age and Propensity Scores (PS) before and after Matching

| | **Before PSM** | | **After PSM** | |
|---|---|---|---|---|
| | KS Test Statistic, D | p-value | KS Test Statistic, D | p-value |
| Propensity Scores | 0.2577 | 2.8279e-82 | 0.0011 | 0.9997 |
| Age | 0.2646 | 9.7113e-87 | 0.0044 | 0.9999 |

For the Wilcoxon rank sum, we test at $\alpha = 5\%$, against the null hypothesis, that the distribution of propensity scores and age between the SGLT2 and non-SGLT2 groups come from the same population.

Table 4.4 presents the results of the Wilcoxon rank sum test before and after propensity score matching (PSM) for two variables, PS and Age.

Before PSM, the Wilcoxon rank sum test showed a statistically significant difference between the SGLT2 and non-SGLT2 groups for PS and Age. These results indicate that the distribu-

tion of PS and age differs significantly between the two groups before PSM.

After PSM, the Wilcoxon rank sum test showed no statistically significant difference in PS and age between the two groups. These results indicate that after PSM, the distribution of variable values is not significantly different between the two groups, and this finding is consistent with the results of the graphical diagnosis and the Kolmogorov-Smirnov test, which also showed no significant differences in the distributions of the two variables between the two groups after PSM.

These results suggest that propensity score matching was successful in balancing the distributions of age and propensity scores between the two groups, which is an important assumption in many causal inference methods. This is a valuable finding that supports the use of PSM in this work.

Table 4.4: Wilcoxon Test Results for Age and Propensity Scores (PS) before and after Matching

| | **Before PSM** | | **After PSM** | |
|---|---|---|---|---|
| | Wilcoxon Statistic | p-value | Wilcoxon Statistic | p-value |
| Propensity Scores | 21.0328 | 3.2866e-98 | −0.0001 | 0.9999 |
| Age | −20.9376 | 2.4347e-97 | 0.0343 | 0.9726 |

To further examine the balance in covariates between the two groups, we plotted a bar chart to visualize the distribution of the dichotomous covariates before and after propensity score matching. The chart showed that the two groups were imbalanced with respect to the covariates prior to PSM, but after PSM, the distribution of the covariates were similar between the two groups. This finding was further supported by our SMD analysis and graphical examination of balance in covariates in Figures 4.12 and 4.11.

The use of the bar chart allowed us to quickly and easily assess the balance in the dichotomous variables before and after PSM. This method of visual inspection is a useful tool in identifying potential imbalances in dichotomous covariate distributions and evaluating the effectiveness of PSM.

The balance in covariates is an important consideration in observational studies, as imbalances can lead to confounding and inaccurate estimates of the exposure or intervention effect. By

using PSM, we were able to control for potential confounding by ensuring that the distribution of covariates was similar between the two groups. This approach is particularly useful in situations where randomization is not possible or ethical, such as in many epidemiological studies.

Overall, our use of both statistical measures and graphical methods to evaluate balance in covariates provides a comprehensive assessment of the effectiveness of our PSM analysis. These findings strengthen the validity of our study and support the accuracy of our estimates of the effect of the exposure or intervention of interest.

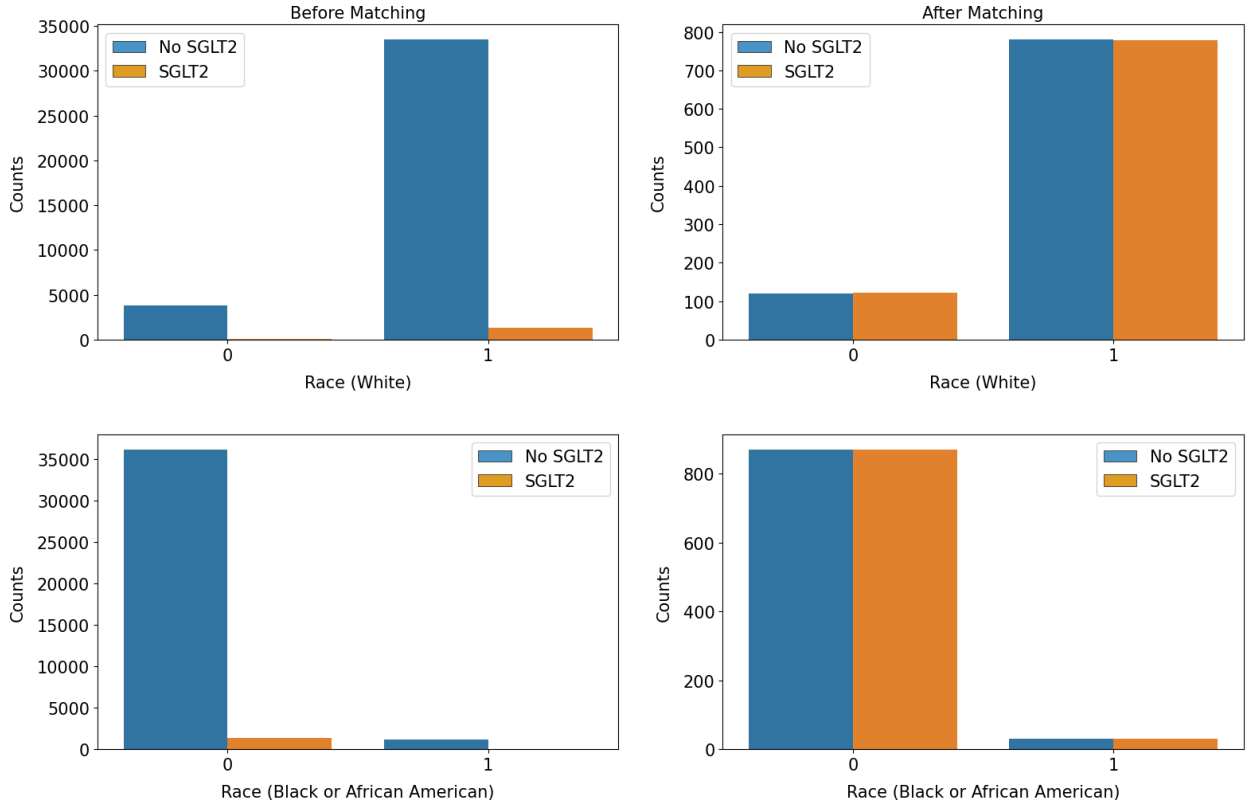


Figure 4.11: (a) Distribution of Dichotomous Variables Before and After Matching. The left graph shows the frequency of each category in the unmatched dataset, while the right shows the corresponding values in the matched dataset.

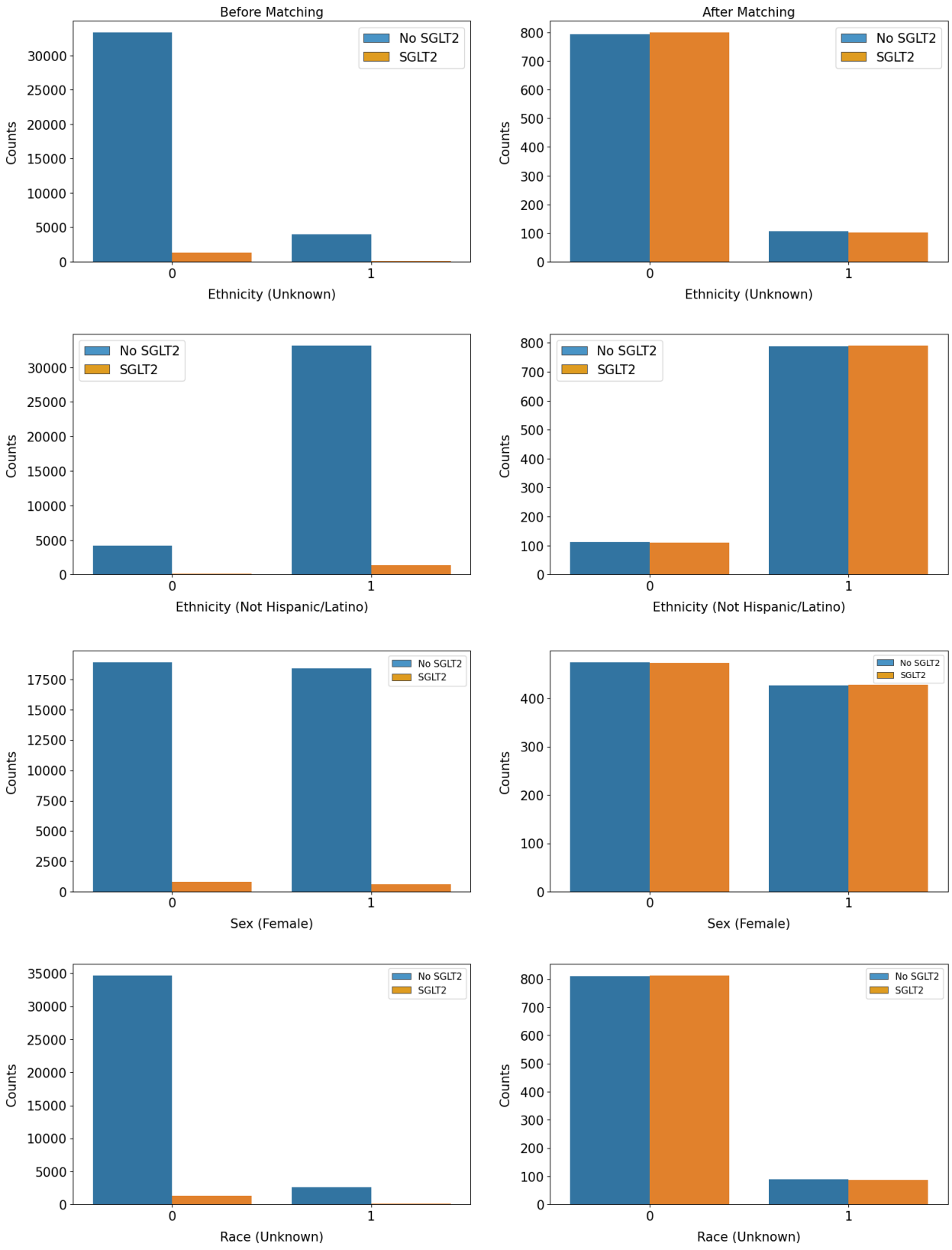


Figure 4.12: (b) Distribution of Dichotomous Variables Before and After Matching. The left graph shows the frequency of each category in the unmatched dataset, while the right shows the corresponding values in the matched dataset.

To further examine the balance in the dichotomous variables before and after PSM, the chi-square ($\chi^2$) test was used. The null hypothesis ($H_0$) states that there is no significant difference in the distribution of each dichotomous variable between the SGLT2 and no SGLT2 groups with a decision rule of rejecting $H_0$ if the p-value is less than 5%.

Based on the provided data in Table 4.5, it was observed that before PSM, five out of six dichotomous variables showed a statistically significant difference between the two groups, as evidenced by their low p-values. This indicates that the groups were not comparable, and there may be confounding variables that could affect the results. The only variable that did not show a significant difference was race (unknown), suggesting that the distribution of this variable in both groups was similar even before PSM.

After PSM, however, all variables showed no statistically significant differences between the two groups, as evidenced by their high p-values. This suggests that PSM was successful in reducing the effect of confounding variables, resulting in comparable treatment and control groups.

Regarding race (unknown), the $\chi^2$ statistic before PSM was 1.727950, while after PSM, it was 0.006234. This suggests that there was an improvement in the balance of race (unknown) between the SGLT2 and no SGLT2 groups after PSM, as the $\chi^2$ statistic was smaller after PSM than before.

Table 4.5: $\chi^2$ Test Results for Dichotomous Variables before and after Matching

| | **Before PSM** | | **After PSM** | |
|---|---|---|---|---|
| | $\chi^2$ Statistic | p-value | $\chi^2$ Statistic | p-value |
| Sex (Female) | 21.609731 | 0.000003 | 0.000000 | 1.000000 |
| Race (White) | 4.604175 | 0.031894 | 0.004740 | 0.945110 |
| Race (Black) | 4.231684 | 0.039676 | 0.000000 | 1.000000 |
| Race (Unknown) | 1.727950 | 0.188673 | 0.006234 | 0.937069 |
| Not Hispanic/Latino | 16.852791 | 0.000040 | 0.020470 | 0.886231 |
| Ethnicity (Unknown) | 19.387304 | 0.000011 | 0.086599 | 0.768546 |

### 4.3.3 Outcome Analysis

After confirming that our propensity score matching procedure resulted in a balanced distribution of covariates between the treatment and control groups, we performed outcome analyses to examine the treatment effect of SGLT2 on our outcome of interest (CKD). Specifically, we estimated the odds ratio (OR) and the treatment effect of SGLT2 on CKD, using logistic regression models.

From Figure 4.13, our results showed that patients treated with SGLT2 had a significantly lower odds of developing, compared to those who did not receive SGLT2 treatment (OR $= 0.156$, 95% CI $[0.123, 0.197]$, $p < 0.05$). Additionally, the treatment effect (ATE) of SGLT2 was estimated to be $-0.354 \pm 0.022$ (95% CI $[-0.394, -0.315]$, $p < 0.05$), indicating a statistically significant reduction in the outcome among patients treated with SGLT2, compared to the control group.

The interpretation of the ATE means that the proportion of positive outcomes (CKD = 1) is lower in the treatment group (SGLT2) compared to the control group (since it is negative). The magnitude of the ATE also provides an estimate of the average difference in the outcome between the treated group (receiving SGLT2) and the control group. In this case, a decrease of $0.354 \pm 0.022$ units in CKD can be attributed to the treatment with SGLT2.

This means that, on average, individuals who use SGLT2 have a lower odds of having CKD than those who do not use SGLT2.

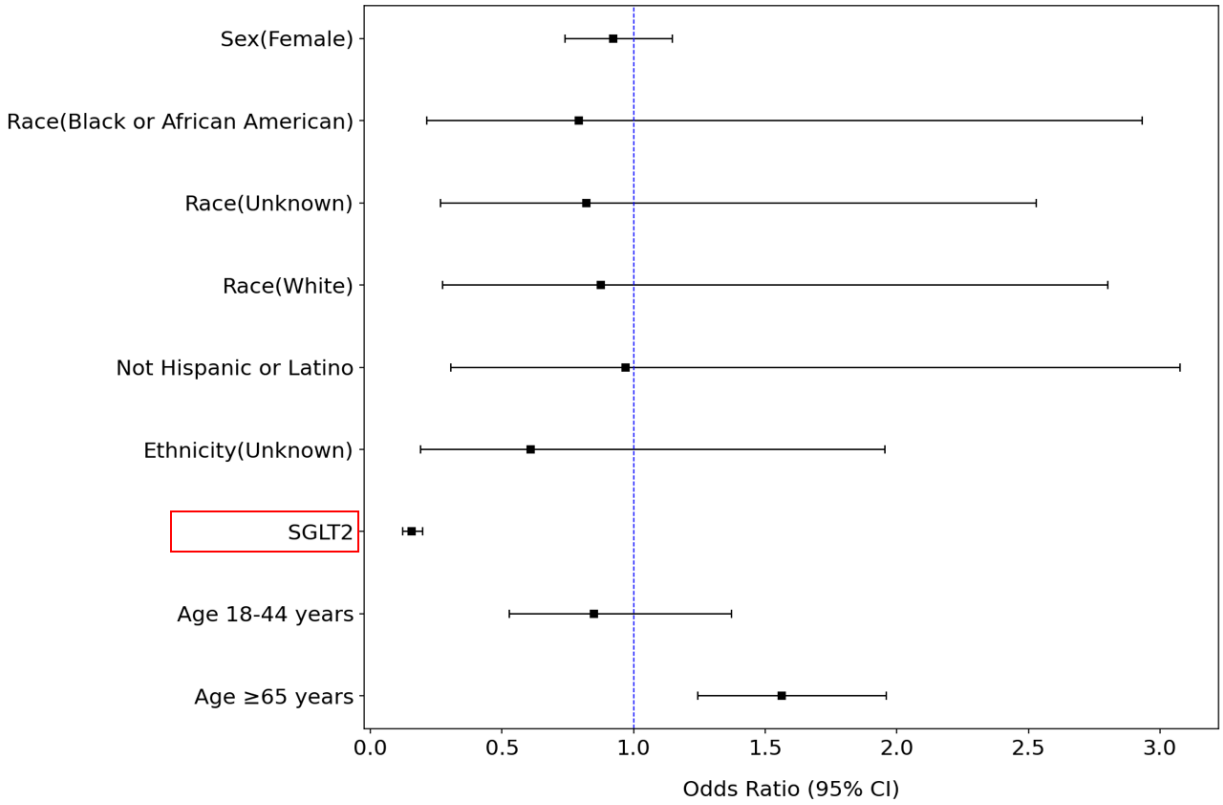


Figure 4.13: Odds Ratios of Covariates: A visual representation of the strength of association between the covariates and the outcome variable, expressed as odds ratios. The bars represent the 95% confidence intervals, and the dotted line at 1.0 indicates no association. Covariates with odds ratios above 1.0 are positively associated with the CKD, while those below 1.0 are negatively associated (decrease the prevalence of CKD cases).

Overall, our findings suggest that SGLT2 treatment is associated with a beneficial effect on the outcome of interest, after accounting for potential confounding factors through propensity score matching and logistic regression analyses.

### 4.4 Machine Learning Models

This section presents an analysis of the effect of Sodium-glucose cotransporter 2 inhibitors (SGLT2) on the prediction of chronic kidney disease (CKD) using machine learning models and the SHapley Additive exPlanations (SHAP) approach.

The overview of how the various machine learning models were trained and evaluated is illustrated in Figure 4.14.

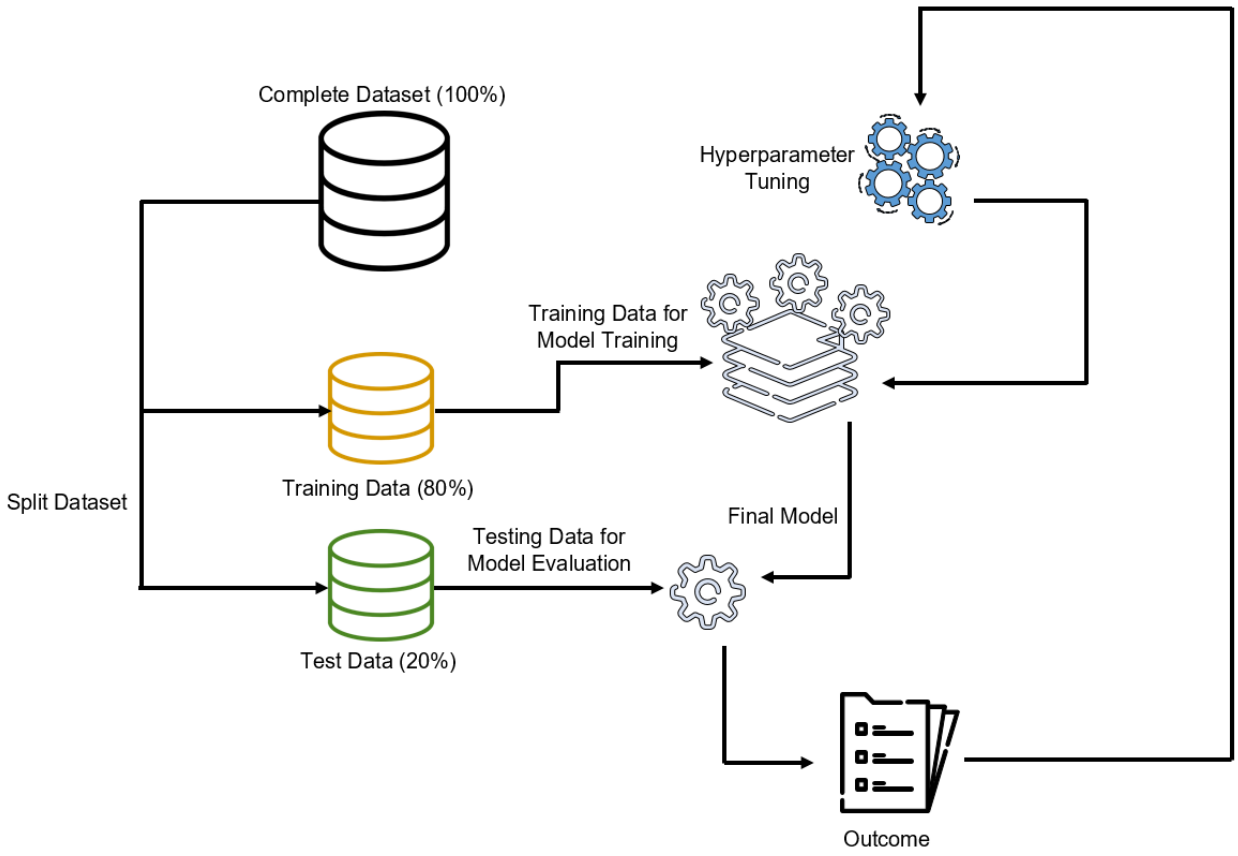


Figure 4.14: Overview of the machine learning models used to predict CKD. The architecture includes data splitting, feature selection, model training, hyperparameter tuning, and model evaluation.

### 4.4.1 Recursive Feature Selection with Random Forest (RFE+RF)

In this study, we used the RFE+RF approach of the wrapper method to select the best features for predicting CKD. This approach involves a greedy search algorithm that iteratively selects features that improve the model's performance until the optimal (or desired) number of features is reached.

We chose to use the random forest (RF) model with RFE because it has a well-known internal method for measuring feature importance. This method can be used with the first model fit within RFE, where the entire predictor set is used to compute the feature rankings. RFE+RF internally applies a filter-based feature selection method to rank the features by importance, discarding the least important features. It then performs cross-validation using 80% of the data for training

and 20% for testing, and re-fits the model through an iteration of each of the predictors.

While this approach is considered the best way for effective and efficient feature selection, it is also the most expensive feature selection method and requires heavy computational power. To ensure robustness and reliability of our results, we used 5-Fold Cross-Validation, and selected the best results. This allowed us to obtain a set of features that were most relevant for predicting CKD, while minimizing the risk of overfitting and obtaining a more robust and generalizable model.

Figure 4.15 shows that the optimal number of features that produced the best predictive performance was determined to be 56. The results in Figure 4.15 demonstrates the relationship between the number of features and their respective cross-validation score. This provides a visual representation of the feature selection process and highlights the importance of selecting the optimal set of features for accurate CKD prediction. The RFECV+RF results can aid in the development of more robust CKD prediction models by providing essential insights into the most significant predictors of CKD.

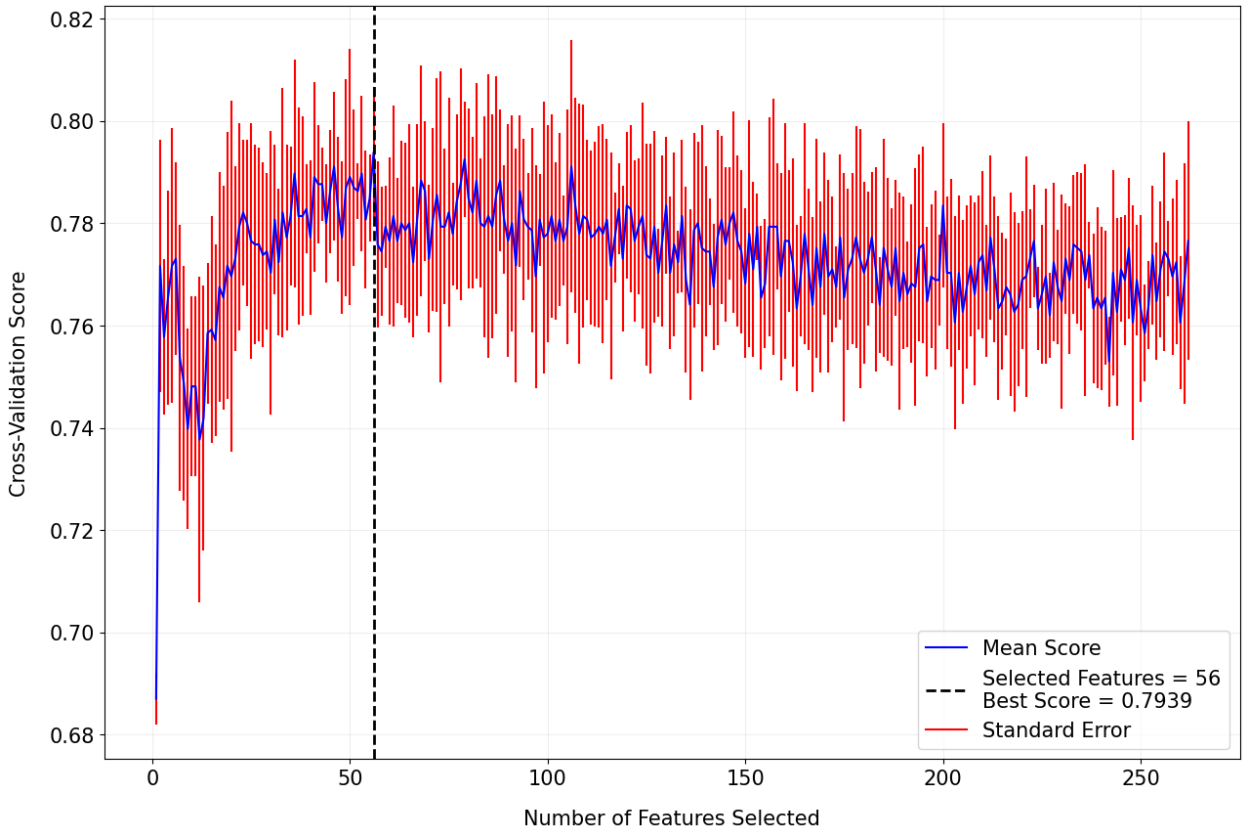


Figure 4.15: The figure presents the results of Recursive Feature Elimination (RFE) combined with Random Forest (RF) using 5-Fold Cross-Validation.

Among these selected 56 optimal features, SGLT2, our intervention of interest, was also included. While the RFECV+RF technique did not directly evaluate the importance of SGLT2, its contribution to the predictive performance of the model was assessed using the SHAP analysis. The SHAP analysis provided more insights into the role of SGLT2 as a potential protective factor for CKD. Therefore, the RFECV technique was an essential first step in selecting the optimal features for predicting CKD, and the subsequent ML training and SHAP analysis provided a more comprehensive understanding of the role of SGLT2, in the prediction of CKD.

### 4.4.2 Model Training and Evaluation

In our predictive analysis, we used the holdout method to split our dataset into a training set (80%) and a testing set (20%). This approach provided the best results for our analysis. We employed six known machine learning algorithms to perform the predictive analysis. The algorithm

that performed the best on the test set based on overall performance metrics such as test accuracy, Area Under the Curve (AUC), and model recall and precision would be selected.

We first ran the algorithms on the original dataset and recorded their performance. We then applied the bootstrapping approach, allowing the models to learn more from the training data, and recorded their performance again. Finally, we used the Synthetic Minority Over-sampling Technique (SMOTE) to create synthetic data for the minority class (CKD) to balance the training data.

Overall, our approach aimed to optimize the performance of our predictive models by selecting the best algorithm and incorporating techniques such as bootstrapping and SMOTE to improve the quality of the training data. By doing so, we were able to make more positive predictions for the dataset.

Table 4.6 provided a comprehensive overview of the performance of the different models before and after balancing the train dataset. The table presents a comparative analysis of the various machine learning models' scores, including logistic regression, decision tree, random forest, XGBoost, support vector classifier (SVC) and neural network.

It also highlight the performances of these models after using the bootstrapping and SMOTE techniques to create a balanced train dataset. This comprehensive analysis of the models' performance helped in identifying the most effective strategy for improving the models' performances in each case. For a more in-depth understanding, detailed figures were provided in the subsection subsections. These figures provided a visual representation of the models' performance, making it easier to interpret the results and gain deeper insights into the data. This analysis is critical in identifying the most effective strategy for improving the models' performance in each case. The results of this analysis can help make better decisions based on the data and improve the overall accuracy of the machine learning models.

Table 4.6: Evaluation of Model Performance (in CKD Prediction)

| | ML Algorithm | Precision | Recall | F1-Score | Accuracy |
|---|---|---|---|---|---|
| Original Dataset | Logistic Regression | 0.669 | 0.791 | 0.725 | 0.817 |
| | Random Forest | 0.689 | 0.746 | 0.716 | 0.820 |
| | Decision Tree | 0.617 | 0.718 | 0.664 | 0.778 |
| | SVC | 0.664 | 0.791 | 0.722 | 0.814 |
| | XGBoost | 0.706 | 0.764 | 0.734 | 0.831 |
| | Neural Network | 0.691 | 0.773 | 0.730 | 0.825 |
| Bootstrapping | Logistic Regression | 0.674 | 0.809 | 0.736 | 0.823 |
| | Random Forest | 0.669 | 0.773 | 0.717 | 0.814 |
| | Decision Tree | 0.701 | 0.746 | 0.723 | 0.825 |
| | SVC | 0.664 | 0.809 | 0.730 | 0.817 |
| | XGBoost | 0.677 | 0.800 | 0.733 | 0.823 |
| | Neural Network | 0.659 | 0.791 | 0.719 | 0.817 |
| SMOTE | Logistic Regression | 0.652 | 0.800 | 0.718 | 0.809 |
| | Random Forest | 0.944 | 0.927 | 0.936 | 0.961 |
| | Decision Tree | 0.872 | 0.864 | 0.868 | 0.920 |
| | SVC | 0.817 | 0.855 | 0.836 | 0.898 |
| | XGBoost | 0.934 | 0.900 | 0.917 | 0.950 |
| | Neural Network | 0.682 | 0.800 | 0.736 | 0.825 |

Figure Figure 4.16 displays a heatmap that compares the performance metrics for using the original dataset and balancing it with SMOTE. The heatmap provides a visual representation of the comparison between the two approaches, with the rows representing the performance metrics and the columns representing the ML algorithms.

This observation is consistent with the fact that SMOTE oversamples the minority class, which may lead to misclassification of some samples. However, the increase in the performance metrics for the minority class more than compensates for a slight decrease in accuracy.


Figure 4.16: Comparison of Machine Learning Models' Performance on Original and Balanced Training Datasets using SMOTE for CKD Prediction.

In addition to the results presented in Table 4.6, we also evaluated the models' performance using the area under the receiver operating characteristic curve (AUC-ROC). The ROC curve is a probability curve that represents the model's true positive rate (sensitivity) against the false-positive rate (1-specificity) for different classification thresholds.

The AUC-ROC score defined how well the models were capable of separating the positive cases from the negative cases in CKD prediction. A score of 1 indicates perfect classification, while a score of 0.5 indicates random guessing. Therefore, the higher the AUC, the better the model for classification, and vice versa.

Evaluating the models' performance using AUC-ROC is crucial in determining the model's

capability to classify positive and negative cases correctly. This evaluation metric complements the precision, recall and accuracy scores presented in Table 4.6 and provides a more comprehensive understanding of the models' performance.

The results in Figure 4.17 indicated that all the models had good AUC scores, with Logistic Regression, XGBoost, and neural network having the highest scores of 0.810, 0.812, and 0.811, respectively. Random Forest and SVM also had high AUC scores of 0.799 and 0.808, respectively, while decision tree had a slightly lower AUC score of 0.761.

These results suggest that the evaluated machine learning models had a good discriminative ability and were effective in CKD prediction for the original dataset, with some models having slightly better performance than others. Therefore, the models with higher AUC scores may be preferred for CKD prediction in clinical practice.

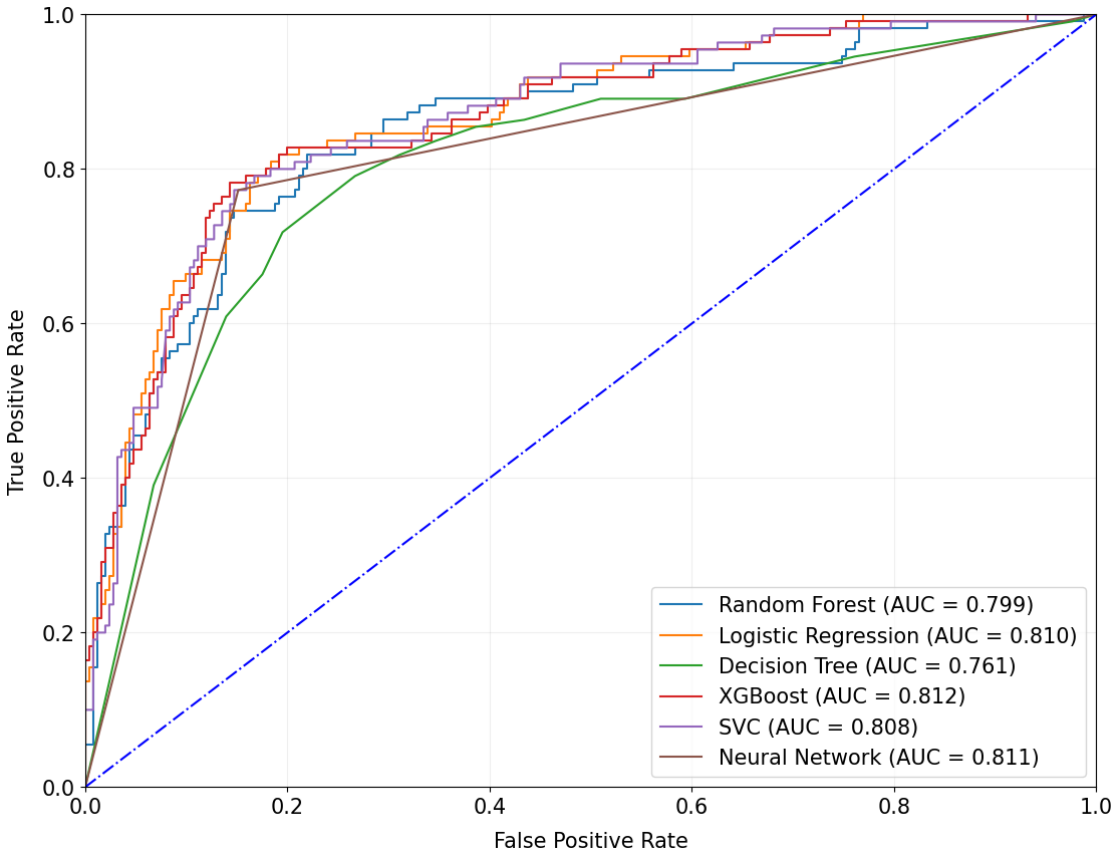


Figure 4.17: ROC-AUC Scores for Machine Learning Models on Original Dataset: The figure presents the ROC-AUC scores for various machine learning models evaluated on the original dataset.

When the models were trained on the balanced dataset using SMOTE, the AUC values showed a significant improvement compared to the original dataset. The Random Forest, Decision Tree, and XGBoost models demonstrated the most substantial improvement, with AUC values of 0.952, 0.904, and 0.936, respectively. The AUC values for Logistic Regression, SVC, and Neural Network models were also higher than the original dataset, but their performance improvement was not as significant, with AUC values ranging from 0.806 to 0.823.

The evaluation of the ML models in this study revealed that the Random Forest model exhibited the best predictive accuracy. However, despite using RFECV with Random Forest in the feature selection process, it was found that this did not have a significant impact on why Random Forest was the best-performing model.

Further examination of the Random Forest model's performance was carried out on both the original and balanced datasets. The balanced dataset was obtained using the Balanced Bagging Classifier with Bootstrap Aggregation. The evaluation revealed that Random Forest's performance was poor in both cases. However, the results changed significantly after applying the SMOTE technique to balance the dataset, which oversampled the minority class.

These findings suggest that Random Forest's effectiveness as the best model cannot be attributed solely to the feature selection process. Instead, the dataset balancing technique used played a crucial role in improving its performance. This highlights the importance of employing appropriate dataset balancing techniques to improve the performance of machine learning models, especially in imbalanced datasets such as the CKD dataset considered in this study.

Overall, these results in Figure 4.18 demonstrate the potential of machine learning models in aiding the prediction of CKD complications. The improved performance of the Random Forest and XGBoost models, which had the highest AUC values in both the original and balanced datasets, highlights the importance of balancing the training dataset. However, external validation using other datasets is necessary to ensure the generalizability of the models to other populations.

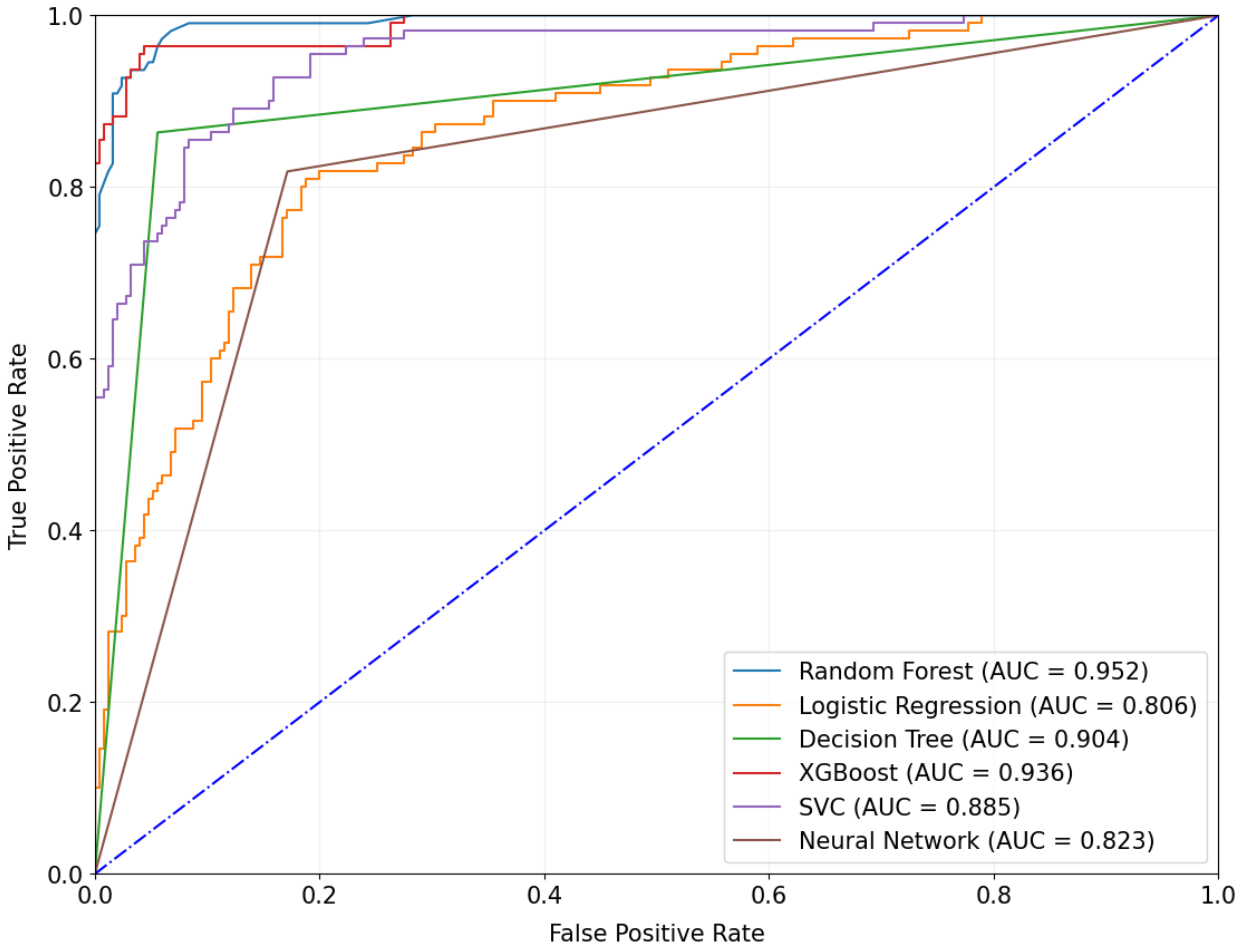


Figure 4.18: Receiver operating characteristic curve Area Under the Curve for the ML Models after balancing the training dataset with SMOTE.

Additionally, the confusion matrices in Figure 4.20 provided a detailed breakdown of the performance of the machine learning models after applying the bootstrapping technique and balancing the training data with SMOTE. It presents the number of true positives, true negatives, false positives, and false negatives for each model. These metrics are important in evaluating the model's accuracy, precision, recall, and F1 score. A visual representation of the confusion matrix can help in identifying which models are better at identifying positive and negative cases of CKD, and which models may require further tuning to improve their performance.

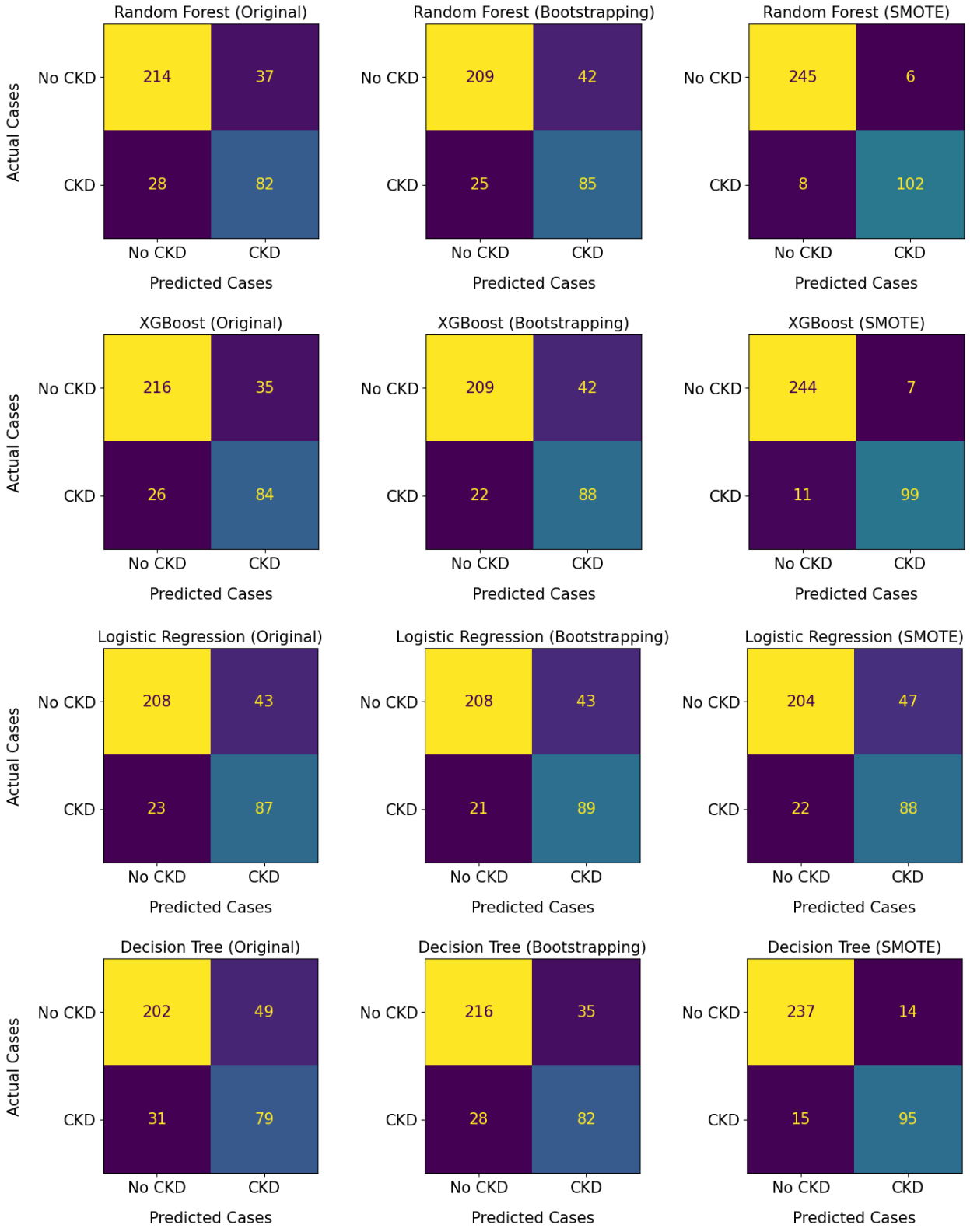


Figure 4.19: (a) Confusion matrices for the evaluated ML models on the original dataset, using bootstrapping and using the SMOTE balanced training dataset. The diagonal elements represent the number of correctly classified CKD cases, while the off-diagonal elements indicate misclassified CKD cases. A cut-off value of 0.5 was used.

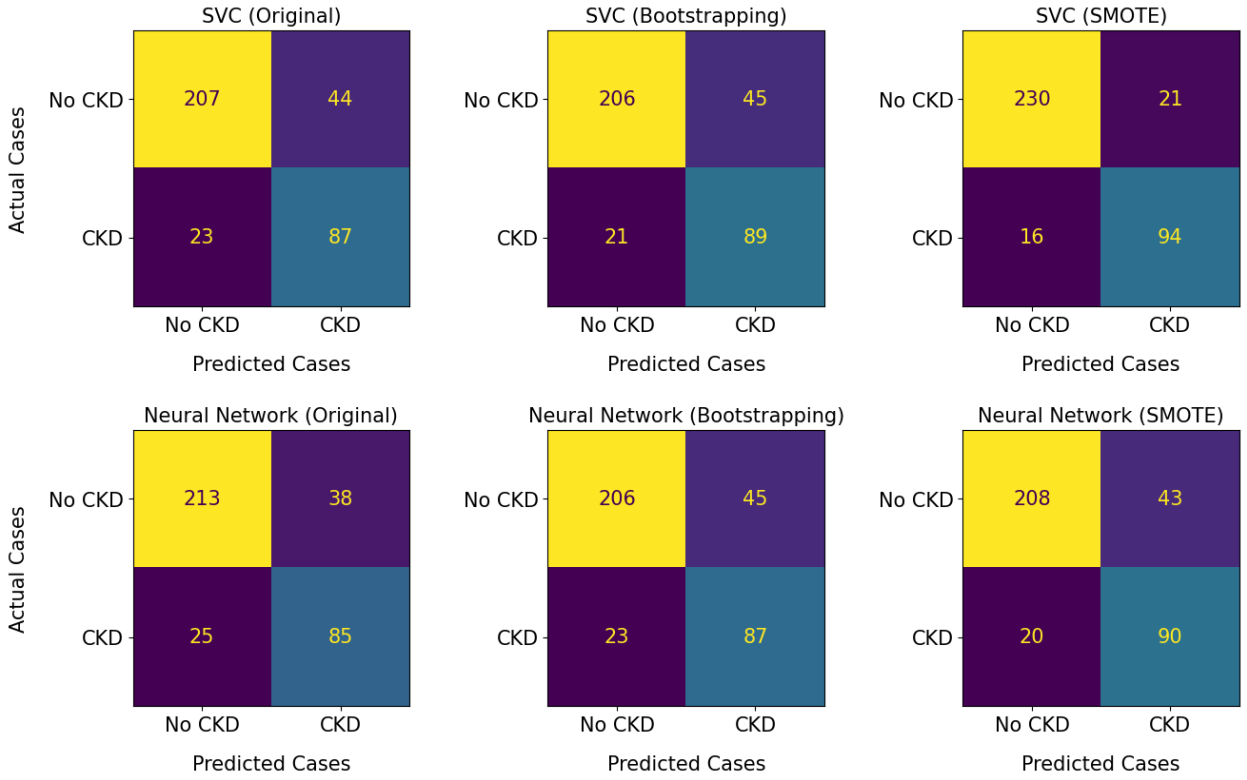


Figure 4.20: (b) Confusion matrices for the evaluated ML models on the original dataset, using bootstrapping and using the SMOTE balanced training dataset. The diagonal elements represent the number of correctly classified CKD cases, while the off-diagonal elements indicate misclassified CKD cases. A cut-off value of 0.5 was used.

### 4.4.3 SHapley Additive exPlanations (SHAP) Analysis

In this subsection, we used the SHAP analysis to investigate how SGLT2 inhibitors impacted CKD prediction. For the SHAP analysis, we will focus on the best-performing machine learning models identified in the previous subsection, namely the Random Forest and the XGBoost. These models demonstrated high AUC values and good performance in CKD prediction, making them suitable for exploring the impact of the SGLT2 feature on the models' predictions.

By applying SHAP analysis to our best performing models, we aim to gain a better understanding of how SGLT2 inhibitors affected the CKD prediction and to identify other important features that may affect CKD predictions. The SHAP analysis of the two best performing models (Random Forest and XGBoost) in Figure 4.21 revealed that SGLT2 was the most important feature for CKD prediction, with a mean SHAP value of approximately 0.16 in both models. This suggests

that SGLT2 played a crucial role in predicting CKD.

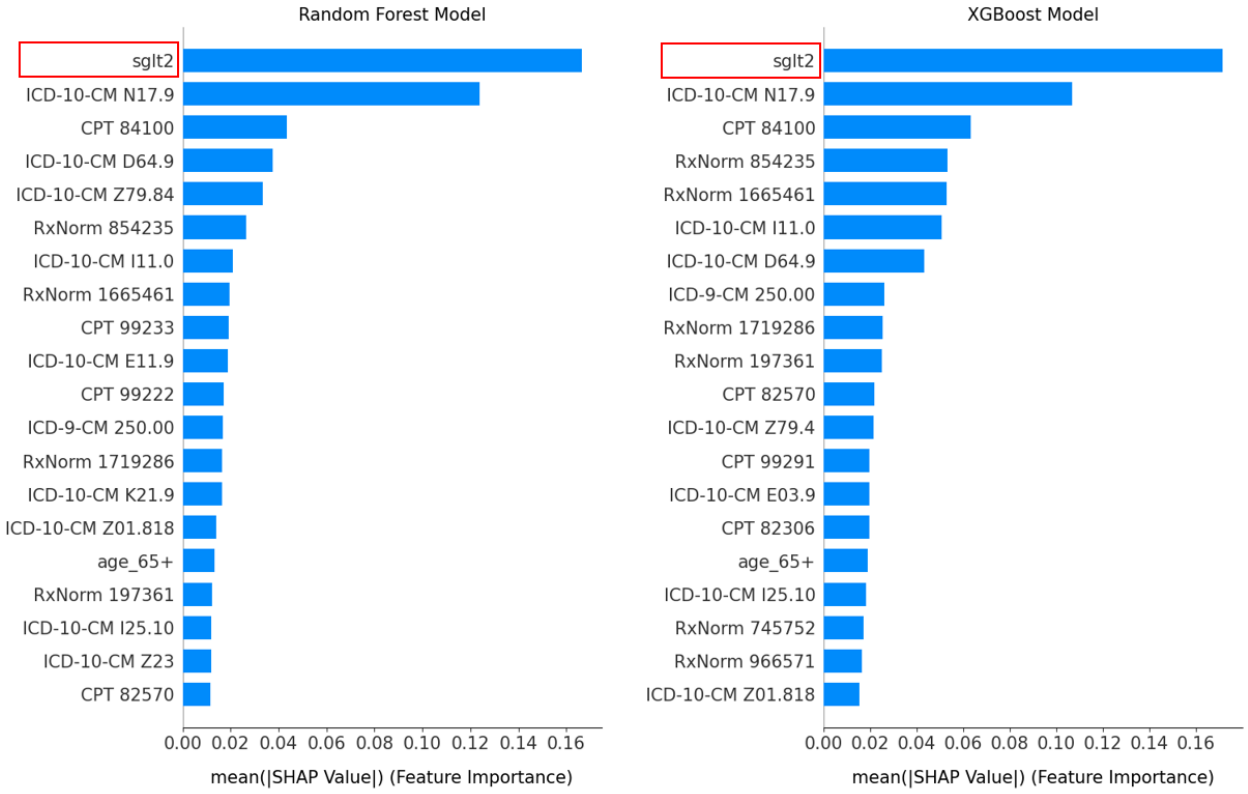


Figure 4.21: Feature importance as determined by the random forest and XGBoost models. This is measured based on the SHAP values, with higher values indicating good performance.

After identifying SGLT2 as a top performing feature, it was important to examine the direction of its impact on the CKD prediction. This helped in gaining a deeper understanding of the underlying mechanism and biological plausibility of the feature in the prediction. Therefore, we proceeded to conduct a SHAP analysis to examine the direction of impact of SGLT2 on the prediction.

In interpreting the SHAP plot, it is important to note that previous analysis using Propensity Score Matching have shown that SGLT2 inhibitors have a protective effect on CKD development.

The beeswarm plot in Figure 4.22 shows that high values of SGLT2 move towards the left (negative) side of the plot and low values move towards the right (positive) side, indicating that SGLT2 has a negative impact on CKD prediction. This finding is consistent with the previously established protective effect, where higher levels of SGLT2 are associated with lower risk of CKD,

while lower levels of SGLT2 are associated with higher risk of CKD.

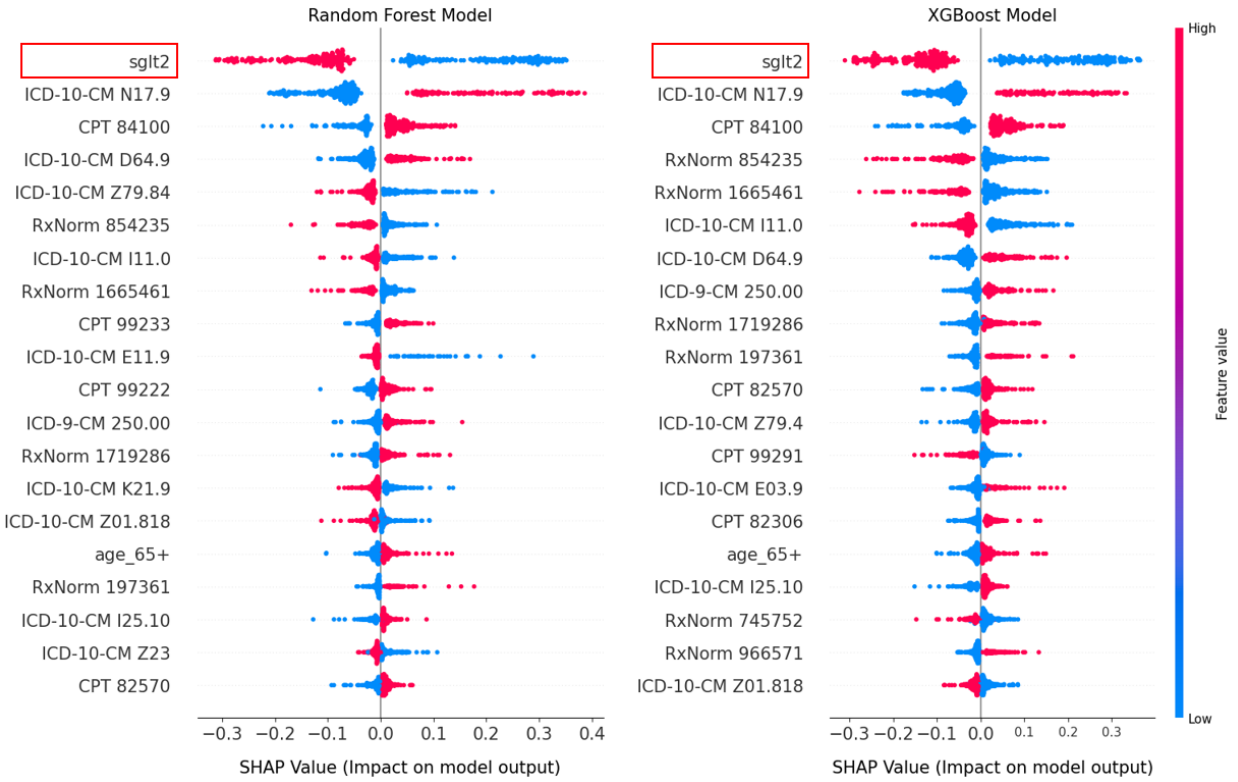


Figure 4.22: Beeswarm plot, ranked by mean absolute SHAP value. This provides a rich overview of how the variables (especially SGLT2) impacted the models' prediction.

The SHAP plot in Figure 4.23 revealed that the SGLT2 feature has a negative impact on the prediction of a positive CKD case. This means that the presence of SGLT2 in a patient's medical profile makes it less likely for the model to predict a positive CKD case. Both models agree on the negative contribution of SGLT2 in predicting a positive CKD case. This consistency between the models provides confidence in the importance of SGLT2 in reducing a positive CKD case.

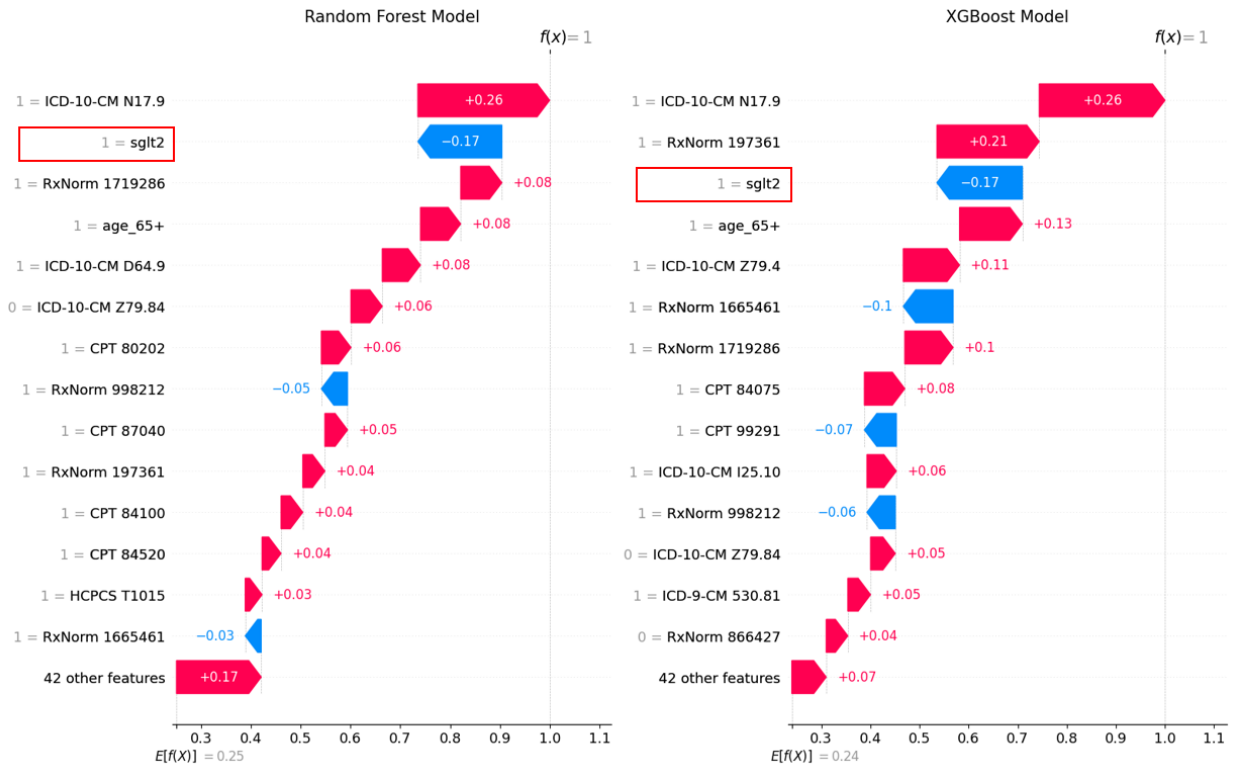


Figure 4.23: Comparison of feature impact on predicting a positive CKD case as determined by the random forest and XGBoost models.

These results suggested that SGLT2 use may be an important factor to consider in CKD risk control. The combination of PSM and SHAP analysis provided a comprehensive understanding of the impact of SGLT2 on CKD prediction, which can inform clinical decision-making and improve patient outcomes.

Furthermore, the feature importance of the random forest and XGBoost models further reinforce the significance of SGLT2 as a top-performing feature for CKD prediction. This highlights the potential of ML models in identifying important clinical features for CKD risk management.

# CHAPTER V

# CONCLUSION, LIMITATIONS AND FUTURE STUDIES

## 5.1 Conclusion

In conclusion, our study provides evidence that SGLT2 inhibitors have a protective effect on CKD outcomes in individuals with diabetes. The results obtained from both the PSM and machine learning models suggest that SGLT2 inhibitors may be an important factor to consider in CKD risk stratification and management. Additionally, our study highlights the potential utility of machine learning models in identifying important clinical features for CKD risk assessment and the importance of addressing issues such as imbalanced classification in medical datasets.

## 5.2 Limitations

This thesis has several limitations that should be acknowledged. First, the use of EHR data may be subject to incomplete or inaccurate recording of patient information, which may have affected the accuracy of our analyses. Although we have used advanced statistical techniques such as PSM to minimize confounding, confounding by indication may still be present due to the nature of observational studies. Additionally, our study only focused on individuals with diabetes, and the generalizability of our findings to other patient populations is unknown.

Also, the large and disperse number of files and records in the dataset can be overwhelming and difficult to manage. Replications in data and multiple patient entries can lead to redundancies and potential inaccuracies in the analysis. The large number of procedures involved in the study can make it difficult to identify and isolate specific factors that may be driving the outcome of CKD.

Finally, this study lacks information on key factors that are known to influence the development of CKD. The dataset used in this study did not include variables such as eGFR, blood pressure,

and urine ACR, which are important indicators of kidney function and damage. These factors are commonly used in clinical practice to diagnose and monitor CKD, and they are also used to assess the efficacy of treatments such as SGLT2 inhibitors. Therefore, the absence of this information may limit the ability of the machine learning model to accurately predict the effect of SGLT2 on CKD.

Despite these limitations, our study has successfully demonstrated the protective impact of SGLT2 inhibitors on CKD outcomes and has provided valuable insights into the potential of utilizing machine learning models for CKD prediction. These findings open up a new avenue for future research that could further explore the effectiveness of SGLT2 inhibitors and machine learning models for CKD prevention and management. It is important for researchers to consider the limitations of our study when interpreting the results and drawing conclusions.

### 5.3 Recommendations for Future Studies

Future studies could consider using a larger and more diverse dataset to improve the generalizability of the findings beyond individuals with diabetes.

In order to address the issue of incomplete or inaccurate recording of patient information in EHR data, future research could explore the use of additional data sources, such as patient-reported outcomes or genetic information.

To minimize redundancies and potential inaccuracies in the analysis, future studies could focus on developing more efficient data management strategies or utilizing data cleaning techniques to identify and eliminate duplicate records.

In order to better identify and isolate specific factors that may be driving the observed outcomes, further research with a more comprehensive dataset that includes key factors such as eGFR, blood pressure, and urine ACR may be necessary to provide a more accurate assessment of the effect of SGLT2 inhibitors on CKD.

Finally, future research could investigate the use of machine learning models in combination with other clinical tools, such as biomarkers or imaging, to improve the accuracy of disease diagnosis and prediction.

REFERENCES


Afkarian, Maryam et al. (2013). "Kidney Disease and Increased Mortality Risk in Type 2 Diabetes". In: *Journal of Theoretical Biology* 24.2, pp. 302–308.

Allen, Angier et al. (2022). "Prediction of Diabetic Kidney Disease with Machine Learning Algorithms, upon the Initial Diagnosis of Type 2 Diabetes Mellitus". In: *BMJ Open Diabetes Research and Care* 10.1, e002560.

Bilous, Rudolf W et al. (2012). "KDOQI Clinical Practice Guideline for Diabetes and CKD: 2012 update". In: *American Journal of Kidney Diseases*.

Birkeland, Kåre I et al. (2017). "Cardiovascular Mortality and Morbidity in Patients with Type 2 Diabetes Following Initiation of Sodium-Glucose Co-transporter-2 Inhibitors versus other Glucose-lowering Drugs (CVD-REAL Nordic): a Multinational Observational Analysis". In: *The lancet Diabetes & Endocrinology* 5.9, pp. 709–717.

Chen, Lei, Dianna J Magliano, and Paul Z Zimmet (2012). "The Worldwide Epidemiology of Type 2 Diabetes Mellitus— Present and Future Perspectives". In: *Nature Reviews Endocrinology* 8.4, pp. 228–236.

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Gregg, Edward W, Desmond E Williams, and Linda Geiss (2014). "Changes in Diabetes-Related Complications in the United States." In: *The New England Journal of Medicine* 371.3, pp. 286–287.

Ho, Daniel E et al. (2007). "Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference". In: *Political Analysis* 15.3, pp. 199–236.

"Introduction: Standards of Medical Care in Diabetes-2021" (Dec. 2020). In: *Diabetes Care* 44.Supplement$_1$, S1–S2. ISSN: 0149-5992. DOI: 10.2337/dc21-Sint. eprint: https://diabetesjournals.org/care/article-pdf/44/Supplement_1/S1/551699/dc21sint.pdf. URL: https://doi.org/10.2337/dc21-Sint.

Ke, Guolin et al. (2017). "Lightgbm: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems* 30.

Levey, Andrew S and Josef Coresh (2012). “Chronic Kidney Disease”. In: *The Lancet* 379.9811, pp. 165–180.

Lundberg, Scott M and Su-In Lee (2017). “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems* 30.

Makino, Masaki et al. (2019). “Artificial Intelligence Predicts the Progression of Diabetic Kidney Disease Using Big Data Machine Learning”. In: *Scientific Reports* 9.1, pp. 1–9.

Molnar, Christoph (2019). “Interpretable Machine Learning: A Guide for Making Black Box Models Explainable”. In: URL: `https://christophm.github.io/interpretable-ml-book`.

Persson, Frederik and Peter Rossing (2018). “Diagnosis of Diabetic kidney Disease: State of the Art and Future Perspective”. In: *Kidney International Supplements* 8.1, pp. 2–7.

Polat, Huseyin, Homay Danaei Mehr, and Aydin Cetin (2017). “Diagnosis of Chronic Kidney Disease Based on Support Vector Machine by Feature Selection Methods”. In: *Journal of Medical Systems* 41.4, pp. 1–11.

Qin, Jiongming et al. (2020). “A Machine Learning Methodology for Diagnosing Chronic Kidney Disease”. In: *IEEE Access* 8, pp. 20991–21002. DOI: `10.1109/ACCESS.2019.2963053`.

Reutens, Anne T (2013). “Epidemiology of Diabetic Kidney Disease”. In: *Medical Clinics* 97.1, pp. 1–18.

Rosenbaum, Paul R and Donald B Rubin (1983). “The Central Role of the Propensity Score in Observational Studies for causal effects”. In: *Biometrika* 70.1, pp. 41–55.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2017). “Learning Important Features through Propagating Activation Differences”. In: *International Conference on Machine Learning*. PMLR, pp. 3145–3153.

Štrumbelj, Erik and Igor Kononenko (2014). “Explaining Prediction Models and Individual Predictions with Feature Contributions”. In: *Knowledge and Information Systems* 41, pp. 647–665.

Stuart, Elizabeth A, Brian K Lee, and Finbarr P Leacy (2013). “Prognostic Score–Based Balance Measures Can be a Useful Diagnostic for Propensity Score Methods in Comparative Effectiveness Research”. In: *Journal of Clinical Epidemiology* 66.8, S84–S90.

Sullivan, Gail M and Richard Feinn (2012). “Using effect Size or Why the P Value is not Enough”. In: *Journal of Graduate Medical education* 4.3, pp. 279–282.

Thomas, Merlin C et al. (2015). “Diabetic Kidney Disease”. In: *Nature Reviews Disease Primers* 1.1, pp. 1–20.

Vivian, Eva M. (2014). “Sodium-Glucose Co-transporter 2 (SGLT2) Inhibitors: A Growing Class of Antidiabetic Agents”. In: *Drugs in Context* 3.

Webster, Angela C et al. (2017). “Chronic Kidney Disease”. In: *The Lancet* 389.10075, pp. 1238–1252.

Yang, Lanting et al. (2022). “Identifying Patients at Risk of Acute Kidney Injury Among Medicare Beneficiaries With Type 2 Diabetes Initiating SGLT2 Inhibitors: A Machine Learning Approach.” In: *Frontiers in Pharmacology* 13, pp. 834743–834743.

# APPENDIX A

# APPENDIX A

## PROPENSITY SCORE MATCHING IN PYTHON

```
[ ]  import numpy as np; import pandas as pd
     from sklearn.linear_model import LogisticRegression
     from sklearn.neighbors import NearestNeighbors
     from sklearn import metrics
     import matplotlib.pyplot as plt
     import seaborn as sns
     import math
     from scipy.stats import ttest_ind
```

```
[ ]  data = pd.read_csv('data.csv') # Read data
     data.groupby('sglt2').mean() # Check the means for control and treatment

     # separate control and treatment
     data_control, data_treatment = data[data.sglt2==0], data[data.sglt2==1]

     # choose features for propensity score calculation
     X = data[['age', 'sex_F', 'race_White', 'race_Black or African American',
               'race_Unknown', 'ethnicity_not Hispanic or Latino', 'ethnicity_Unknown']]
     y = data['sglt2']
```

```
     # use logistic regression to calculate the propensity scores
     log_model = LogisticRegression()
     log_model.fit(X, y)

     # combine features and coefficients into a dataframe
     coeffs = pd.DataFrame({'column':X.columns.to_numpy(), 'coeff':log_model.coef_.ravel(),})

     # prediction
     pred_binary = log_model.predict(X)  # binary 0 control, 1, treatment
     pred_prob = log_model.predict_proba(X)  # probabilities for classes

     # the propensity score (ps) is the probability of being 1 (i.e., in the sglt2 group)
     data['ps'] = pred_prob[:, 1]

     # calculate the logit of the propensity score for matching if needed
     def logit(p):
         return (math.log(p / (1-p)))

     data['ps_logit'] = data.ps.apply(lambda x: logit(x))

     sns.histplot(data=data, x='ps', hue='sglt2')  # check the overlap of ps
```

```
# use 25% of standard deviation of the propensity score as the caliper/radius

caliper = np.std(data.ps) * 0.25
n_neighbors = 10

# setup knn
knn = NearestNeighbors(n_neighbors=n_neighbors, radius=caliper)

ps = data[['ps']]  # double brackets as a dataframe
knn.fit(ps)

distances, neighbor_indexes = knn.kneighbors(ps)
```

```
[ ] # for each point in treatment, we find a matching point in control without replacement
    # note the 10 neighbors may include both points in treatment and control

    matched_control = []  # keep track of the matched observations in control

    for current_index, row in data.iterrows():  # iterate over the dataframe
        if row.treatment == 0:  # the current row is in the control group
            data.loc[current_index, 'matched'] = np.nan  # set matched to nan
        else:
            for idx in neighbor_indexes[current_index, :]: # for each row in treatment, find the k neighbors
                # make sure the current row is not the idx - don't match to itself
                # and the neighbor is in the control
                if (current_index != idx) and (data.loc[idx].treatment == 0):
                    if idx not in matched_control:  # this control has not been matched yet
                        data.loc[current_index, 'matched'] = idx  # record the matching
                        matched_control.append(idx)  # add the matched to the list
                        break
```

```
[ ] print('total observations in treatment:', len(datd[data.sglt2==1]))
    print('total matched observations in control:', len(matched_control))

    treatment_matched = data.dropna(subset=['matched'])  # drop not matched

    # matched control observation indexes
    control_matched_idx = treatment_matched.matched.astype(int)
    control_matched = data.loc[control_matched_idx, :]  # select matched control observations

    # combine the matched treatment and control
    df_matched = pd.concat([treatment_matched, control_matched])

    print(df_matched.treatment.value_counts())

    # matched control and treatment
    df_matched_control = df_matched[df_matched.sglt2==0]
    df_matched_treatment = df_matched[df_matched.sglt2==1]
```

```
[ ]  from numpy import mean, var
     from math import sqrt

     # function to calculate Cohen's d for independent samples
     def cohen_d(d1, d2):
       n1, n2 = len(d1), len(d2)
       s1, s2 = var(d1, ddof=1), var(d2, ddof=1)
       s = sqrt(((n1 - 1) * s1 + (n2 - 1) * s2) / (n1 + n2 - 2))
       u1, u2 = mean(d1), mean(d2)
       return (u1 - u2) / s

     effect_sizes = []; cols = X.columns

     for cl in cols:
         _, p_before = ttest_ind(data_control[cl], data_treatment[cl])
         _, p_after = ttest_ind(df_matched_control[cl], df_matched_treatment[cl])
         cohen_d_before = cohen_d(df_treatment[cl], df_control[cl])
         cohen_d_after = cohen_d(df_matched_treatment[cl], df_matched_control[cl])
         effect_sizes.append([cl,'before', cohen_d_before, p_before])
         effect_sizes.append([cl,'after', cohen_d_after, p_after])

     df_effect_sizes = pd.DataFrame(effect_sizes, columns=['feature', 'matching', 'effect_size', 'p-value'])
     df_effect_sizes

     fig, ax = plt.subplots(figsize=(15, 5))
     sns.barplot(data=df_effect_sizes, x='effect_size', y='feature', hue='matching', orient='h')
```

APPENDIX B

# APPENDIX B

## MACHINE LEARNING

```
[ ]  import matplotlib.pyplot as plt
     from sklearn.metrics import classification_report, confusion_matrix
     import pydotplus
     from sklearn.metrics import accuracy_score
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.svm import SVC
     from sklearn.pipeline import Pipeline
     from sklearn.model_selection import GridSearchCV
     from sklearn.neural_network import MLPClassifier
     from xgboost import XGBClassifier
     from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import roc_curve, auc, precision_score, f1_score
     from sklearn.metrics import confusion_matrix, roc_auc_score, recall_score
     from sklearn.model_selection import train_test_split
```

```
[ ]  K = data
     K = K[K['patient_id'].isin(df_matched['patient_id'])]
```

```
[ ]  X = K.drop(['patient_id', 'CKD'], axis=1)
     y=K[['CKD']]

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 109)
```

```
     #Recursive Feature Elimination with Random Forest
     from sklearn.feature_selection import RFECV
     from sklearn.model_selection import StratifiedKFold

     random_seed = 12344

     rfecv_rf = RFECV(
         estimator=RandomForestClassifier(random_state=random_seed),
         step=1,
         cv=StratifiedKFold(5, random_state=random_seed, shuffle=True),
         scoring="accuracy",
         min_features_to_select=1,
         n_jobs=2)

     rfecv_rf.fit(X_train, y_train)
```

```
selected_features_rf = X_train.columns[rfecv_rf.support_]
n_scores_rf = len(rfecv_rf.cv_results_["mean_test_score"])
bs = round(np.max(rfecv_rf.cv_results_['mean_test_score']),4)

plt.figure(figsize=(15,10))
plt.xlabel("Number of Features Selected", fontsize=15, labelpad=15)
plt.ylabel("Cross-Validation Score", fontsize=15, labelpad=15)

plt.errorbar(
    range(rfecv_rf.min_features_to_select, n_scores_rf + rfecv_rf.min_features_to_select),
    rfecv_rf.cv_results_["mean_test_score"],
    yerr=rfecv_rf.cv_results_["std_test_score"],
    fmt=' ', ecolor='red', color='blue')

plt.plot(
    range(rfecv_rf.min_features_to_select,
          n_scores_rf + rfecv_rf.min_features_to_select),
         rfecv_rf.cv_results_["mean_test_score"],
         linestyle='-', marker='', markersize=5, color='blue', label='Mean Score')

plt.axvline(x = rfecv_rf.n_features_, color = 'black', ls='--',
            lw=2, label=f'Selected Features = {rfecv_rf.n_features_}\nBest Score = {bs}')
plt.xticks(size=15); plt.yticks(size=15)
plt.plot([], [], color='red', label='Standard Error')
plt.legend(fontsize=15, loc='lower right')
plt.show()
```

```
# Random Forest Model
rf = RandomForestClassifier(criterion='gini', max_features=56,
                            n_estimators=56,
                            class_weight='balanced', max_depth=6)
rf.fit(X_train[selected_features_rf], y_train)
rf_pred = rf.predict(X_test[selected_features_rf])
```

```
[ ] from imblearn.ensemble import BalancedBaggingClassifier

    # Balanced Bagging with Random Forest as Base Estimator
    # Base Estimator can be Varied
    bb = BalancedBaggingClassifier(RandomForestClassifier(criterion='gini', max_features=56,
                                                          n_estimators=56, max_depth=6,
                                                          class_weight='balanced'),
                                   n_estimators=500, sampling_strategy='auto',
                                   max_samples=1000, bootstrap=True)

    bb.fit(X_train[selected_features_rf], y_train)
    pp = bb.predict(X_test[selected_features_rf])
```

```
[ ]  # SMOTE
     from imblearn.over_sampling import SMOTE
     from collections import Counter

     sampling = SMOTE(random_state = 100)
     X_train_smote, y_train_smote = sampling.fit_resample(X_train.values, y_train.values.ravel())

     # Shuffle the data
     perms = np.random.permutation(X_train_smote.shape[0])
     X_train_smote = X_train_smote[perms]
     X_train_smote = pd.DataFrame(data = X_train_smote, columns = X_train.columns)
     y_train_smote = y_train_smote[perms]
     y_train_smote = pd.DataFrame(data = y_train_smote)
```

```
     rf_model = RandomForestClassifier(criterion='gini', max_features=56,
                                       n_estimators=56, max_depth=6,
                                       class_weight='balanced')
     rf_model.fit(X_train_smote[selected_features_rf], y_train_smote)
     rf_pred = rf_model.predict(X_test[selected_features_rf])
     print(classification_report(y_test, rf_pred))
     print(confusion_matrix(y_test, rf_pred))
```

```
     # SHAP Analysis

     import shap
     import transformers

     model_shap = RandomForestClassifier(criterion='gini', max_features=56,
                                         n_estimators=56, class_weight='balanced',
                                         max_depth=6)
     model_shap.fit(X_train[selected_features_rf], y_train)
     explainer = shap.Explainer(model_shap.predict, X_test[selected_features_rf])
     shap_values = explainer(X_test[selected_features_rf], max_evals = 6500)
```

```
[ ]  shap.summary_plot(shap_values, max_display=15, show=False, plot_size=(12, 8))
     shap.summary_plot(shap_values, X_test[selected_features_rf], max_display=15, plot_type='bar')
     shap.plots.bar(shap_values[1], show = False, max_display=15)
     shap.plots.waterfall(shap_values[0])
```

**Model Hyperparameters**

- Logistic Regression:

  solver = ‘newton-cg’, C=0.1, penalty=‘l2’, class_weight=‘balanced’

- Random Forest:

  criterion=‘gini’, max_features=56, max_depth=6, n_estimator=56, class_weight=‘balanced’

- Decision Tree:

  max_depth=4,min_sample_leaf=4, cc_alpha=0.001, max_features=99, class_weight=‘balanced’

- XGBoost:

  n_estimator=63, max_depth=3, learning_rate=0.14, scale_pos_weight=1.853

- Support Vector Classifier:

  C=0.3, kernel='linear', degree=2, gamma=0.01

- Neural Network:

  Input layer with 56 neurons

  3 hidden layers with 32, 16, and 32 neurons respectively

  Output layer with 2 neurons

  Activation functions: softmax and sigmoid

# BIOGRAPHICAL SKETCH

Solomon Eshun is a first-generation college graduate, who has excelled in the fields of mathematics, data science, and computational biology. He completed his Master's degree in the Applied Statistics and Data Science program at the University of Texas Rio Grande Valley (UTRGV) in May 2023, where he acquired advanced skills in data analysis and modeling. Prior to that, Solomon earned a Bachelor of Science in Mathematics degree from the University of Mines and Technology (UMaT), Ghana, where he emerged as the Best Mathematical Sciences Student, the Best Faculty of Engineering Student, and ultimately, the Valedictorian of his graduating class. Solomon was awarded the Sustainability Research Fellowship award in research by the Office for Sustainability, UTRGV for the Fall 2022 to Spring 2023 term.

During the summer of 2022, Solomon worked as a computational science Graduate Researcher in the Theoretical Biology and Biophysics Group (T-6) at the prestigious Los Alamos National Laboratory, where he worked on projects related to vector-borne diseases. Together with his team, they analyzed the effect of temperature on the spread of these diseases. This experience further honed Solomon's expertise in data analysis and modeling, and reinforced his commitment to using his skills to help solve real-world problems.

Solomon's research interests encompass various fields such as mathematical epidemiology, biomedical data science, machine learning, dynamical systems, and operations research. He is passionate about applying his skills to make meaningful contributions in public health and disease control.

Feel free to send him an email at the following address: eshunsolomon5@gmail.com.