Early-stage Detection of Kidney Disease using Machine Learning

Austin Yan [*]

March 28, 2022

Abstract

Chronic kidney disease is one of the most widespread diseases on Earth, at any given moment, it is estimated that around 697.5 million people are diagnosed with CKD. There are 5 stages of CKD, each stage gets progressively worse, until stage 5, where dialysis or a kidney transplant is needed to maintain life. If detected in the early stages (stage 1-2), treatments can slow or even stop the progression of CKD. However, detecting early stages of CKD is very difficult because patients usually do not have symptoms until stage 4, by then, the kidney's function is cut in half. In this paper, we demonstrated utilization of data efficient machine learning (ML) algorithms as Decision Tree, SVM, and Logistic Regression on publicly available data to predict CKD. Further interpretation of the algorithms revealed significance of certain attributes (e.g. blood pressure, age) and we were able to get 95 percent + accuracy using certain features about patients.

1 Introduction

The kidneys are located to the rear of the abdominal cavity on either side of the spine. Their main functions are to regulate extracellular fluid volume, regulate ion concentrations and pH levels in the body, excrete waste and toxins, and produce hormones (regulates blood pressure levels). Every day, the kidney filters roughly 200 liters of blood and also produces roughly 1-2 liters of urine. Chronic kidney disease (known as CKD) is a disease that gradually weakens the kidneys to the point that wastes can build up leading to complications like high blood pressure, anemia, weak bones, and nerve damage. CKD may lead to kidney failure, which requires dialysis or a kidney transplant to maintain life. Each year 37 million Americans suffer from CKD; it is estimated that there are 697.5 million cases of CKD in the world as of 2017 (9.1 percent of the world population). Early detection can substantially decrease the damage done to the kidneys. However, CKD is oftentimes asymptomatic in early stages [Web17].

[*]Advised by: Perman Jorayev

When determining the severity of a patient's CKD, doctors will test the patient's blood for creatinine, a waste product in blood (eGRF). There are 5 stages of CKD and depending on the results, the patient will fall into 1 of the 5 stages (1 is mild kidney disease and 5 is kidney failure) [ea07]. A complete blood count (CBC) is a set of tests that provide information about the cells in a person's blood. A CBC shows the counts of red blood cells, white blood cells, and platelets, the concentration of hemoglobin, and the hematocrit (the percent of red blood cells in blood). Both the CBC and the creatinine test are used to diagnose CKD. However, during an annual doctor's checkup, these tests are rarely done unless the patient shows symptoms of CKD or has a family history of CKD. Unfortunately, CKD is oftentimes asymptomatic in early stages up until stage 4, at that point, the kidney's functions have been cut in half [ea03] [Atk05].

Early detection and early treatment are essential for CKD because stage 4 CKD is one stage away from kidney failure and the implementation of dialysis or even worse, a kidney transplant. Blood tests and creatinine tests are only done if there are reasons to believe a patient has some sort of kidney disease, but CKD in early stages is often asymptomatic and patients will be oblivious of their condition. This is why we have to use the data in doctor's office to the best of our abilities. During the doctor's visit, doctors may test for patients: blood pressure, sugar intake, normal or abnormal amounts of red blood cells and pus cells, blood glucose random, etc.

The data from a simple doctor's visit might seem meaningless to the human eye, but utilizing the data efficient machine learning algorithms, we might be able to identify signals for early-stage detection of kidney disease. The recent increase in use of ML algorithms demonstrated the applications of ML models ranging from drug discovery to agriculture to disease detection. If the algorithm suggests that the patient may have kidney disease, now we have a reason to believe and run further tests like CBC and creatinine test to tell us definitively. If we can predict kidney disease way before we show any symptoms, therefore, we can treat it at an earlier stage and prevent it from worsening [Bre21]. Here, we demonstrated that using different levels of datasets (i.e. from 6 attributes to 24 attributes) depending on the availability of the data for the patient, showing flexibility of our workflow, we utilized several classification algorithms for above 90 percent accuracy in our predictions, even on a dataset that is easily accessible at home. Further model interpretations such as confusion matrix revealed the individual performance of the models.

2 Methodology

2.1 Data Preparation and Workflow

Our objective in this project is not to build a model with perfect attributes to predict CKD while still achieving high accuracy. Instead, it is to develop a generalizable and flexible workflow that could reliably predict CKD with the least number of attributes. The dataset we obtained from UCI repository [Rub15] was not clean and complete. Several missing attributes, duplicates, and comments were found in the raw dataset. This required several manual and automated steps of data pre-processing.

The original dataset was from the UCI machine learning repository and the data was collected in India. The dataset had 280 rows and 25 attributes, 174 patients had CKD and 106 patients did not have CKD, showing enough diversity to avoid biases towards a certain class in the data (Fig. 2). The main issues were that some data points were missing, some attributes were too correlated to CKD that the implementation of machine learning would have been biased. There were two different data types - some data points were nominal (normal, abnormal, good, poor, yes, no, etc.), while others were numerical (1, 2, 3, 4, etc.). For the code to work, all data types must be in the same format, meaning we must convert the nominal data into numerical data. In the dataset we were able to manually locate the nominal data and replace it with numbers using the find and replace tool. For example, nominal data like good and poor can be replaced with numbers like 1 and 2.

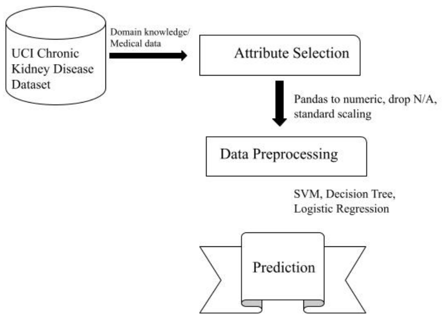

UCI Chronic Kidney Disease Dataset
Domain knowledge/ Medical data
Attribute Selection
Pandas to numeric, drop N/A, standard scaling
Data Preprocessing
SVM, Decision Tree, Logistic Regression
Prediction

Figure 1: Workflow of ML-based disease prediction.

2.2 Filtering and Selection

Next, to filter out the attributes that give tell-tale signs of CKD, we created a table, gave each attribute a difficulty rating, and explained why it deserved that rating. Afterwards, we discovered that the attributes with the highest difficulty rating were also the ones that had too much correlation to kidney disease; it would not be practical as a way to predict CKD. We filtered 25 attributes into 9. Finally, to make sure there were no gaps in the data, we ran a for loop to remove any row that had a gap in it, therefore, the cleaned dataset no longer had any gaps, the original dataset had 280 rows and after cleaning it, the cleaned dataset had 147 rows. The objective is to be able to accurately classify if a patient has CKD with the dataset, which is why we need a model that will be able to classify CKD in two categories, CKD and notCKD.

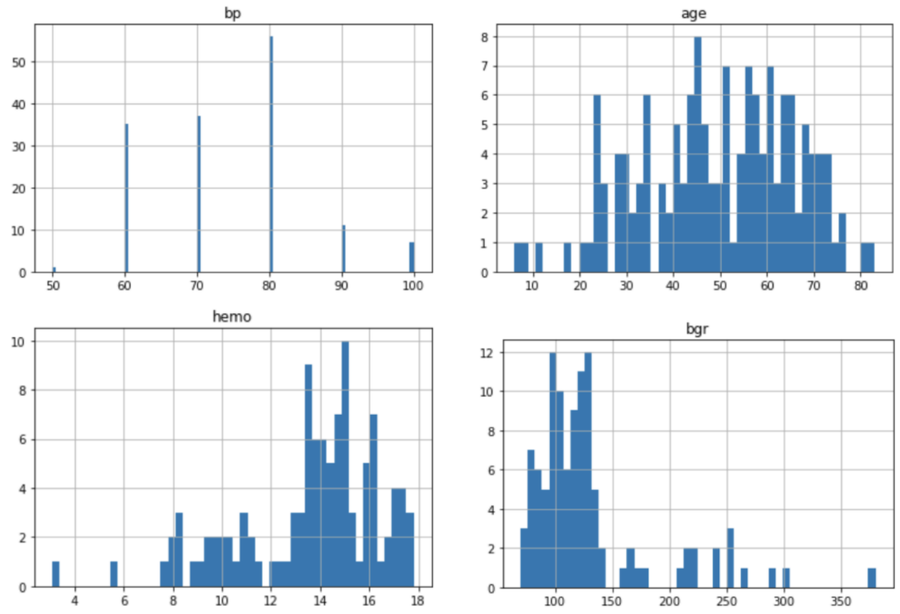


Figure 2: Data distribution across different attributes (bp: blood pressure; hemo: hemoglobin; bgr: blood glucose random) to confirm a diverse data without a significant bias towards a certain attribute range.

2.3 Machine Learning Algorithms

2.3.1 Decision Tree

Decision Tree is a machine learning algorithm that creates classification models that form a tree structure. Similar to how the human mind makes decisions, Decision Tree breaks down a more complicated dataset and evaluates in smaller subsets. Each subset leads to an unique subset, until the complicated dataset is fully broken down. The split criteria in a tree (i.e. nod) can be decided using either gini or entropy model hyperparameter. In both cases, the objective of the model is to maximize the information gain after split (i.e. achieve better separation of classes). We also set the max-depth hyperparameter to 10 to give enough flexibility, but also avoiding model overfitting. Afterwards with testing data, we can run it through the Decision Tree model that was built on the training data to predict the new data.

2.3.2 Support Vector Machine (SVM)

Support Vector Machine is a model to solve classification problems by creating a line or hyperplane which separates the data into classes. Sometimes data may look impossible to separate the classes with only one line, but kernel hyperparameters such as linear, poly, and rbf can transfer the data into higher dimension spaces. In our workflow, we implemented all three kernels and compared their performances (Figure 3). Only then will the hyperplane be able to separate the data into different classes.

2.3.3 Logistic Regression

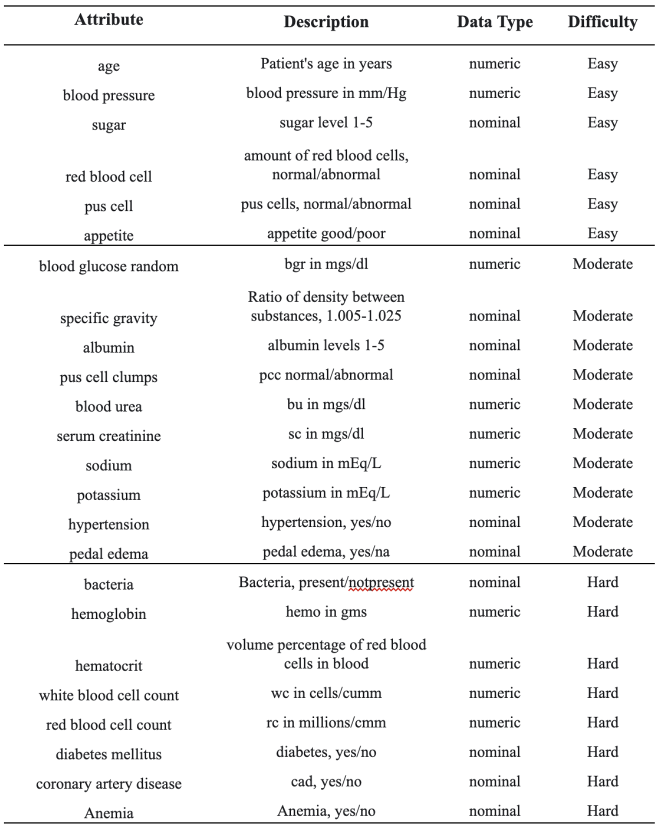
Logistic Regression is a machine learning algorithm used for classification. Logistic Regression fits a sigmoid function to the data; the sigmoid function exists only between 0 and 1 and observes a discrete set of classes (CKD and not CKD). Moreover, the sigmoid function can tell the probability a patient has CKD based on the various attributes we assign to the x variable, which shows the confidence in our predictions. For instance, a patient with 95 percent confidence of having a disease would surely need urgent testing, while a prediction with 60 percent probability would need to repeat the test.

2.4 Feature Scaling and Train Test Split

When we model a data, we need a certain part of the data, usually 70-80 percent to be used for training an ML model and the rest (i.e. test or a holdout data) to validate our prediction on a dataset the model has not seen before. This is where the train-test-split comes in handy. This technique starts with taking a dataset and splitting it into 2 subsets (train and test). The train dataset is used to fit the machine learning model and it is tasked to find a pattern or correlation between the data. The knowledge from the train dataset is then applied to the test data to create a prediction. That prediction is then compared to the actual result to determine the accuracy, precision, etc. This paper used 80 percent train data and 20 percent test data. Imagine a game of Wheel of Fortune and 80 percent of the phrase has already been filled in, it would be quite easy to presume the other 20 percent just by intuition. The same can be said for the train-test-split technique, the purpose is not to use anecdotal evidence (bias) from the train data to predict the test data. To eliminate the bias, feature scaling is applied to train-test split so that each feature contributes a proportionate amount to the prediction.

2.5 Evaluation of Different Data Formatting

The original dataset had many nominal attributes that had to be converted into numerical data. During this process, for all attributes excluding classification, the nominal data that was positive (good appetite, normal red blood cell, etc.) was replaced with 1, while all negative data (cad and dm presence, abnormal pus cell, etc.) would be replaced with 2. Likewise, CKD was replaced with 0 and notCKD was replaced with 1. To understand the ML algorithms better, we wanted to see if changing all nominal data to 0 and 1 vs 1 and 2 exclusively, would alter the results of the classification. After changing all the nominal attributes to either 0 and 1 or 1 and 2, we run the same 3 ML algorithms and the results are the exact same. All of the confusion matrices were also the same, meaning regardless of what number we set the variables to, the model still correctly identified the same points, demonstrating flexibility of the ML algorithms on different data formatting that can be used to describe nominal (categorical) data.



Attribute	Description	Data Type	Difficulty
age	Patient's age in years	numeric	Easy
blood pressure	blood pressure in mm/Hg	numeric	Easy
sugar	sugar level 1-5	nominal	Easy
red blood cell	amount of red blood cells, normal/abnormal	nominal	Easy
pus cell	pus cells, normal/abnormal	nominal	Easy
appetite	appetite good/poor	nominal	Easy
blood glucose random	bgr in mgs/dl	numeric	Moderate
specific gravity	Ratio of density between substances, 1.005-1.025	nominal	Moderate
albumin	albumin levels 1-5	nominal	Moderate
pus cell clumps	pcc normal/abnormal	nominal	Moderate
blood urea	bu in mgs/dl	numeric	Moderate
serum creatinine	sc in mgs/dl	numeric	Moderate
sodium	sodium in mEq/L	numeric	Moderate
potassium	potassium in mEq/L	numeric	Moderate
hypertension	hypertension, yes/no	nominal	Moderate
pedal edema	pedal edema, yes/na	nominal	Moderate
bacteria	Bacteria, present/notpresent	nominal	Hard
hemoglobin	hemo in gms	numeric	Hard
hematocrit	volume percentage of red blood cells in blood	numeric	Hard
white blood cell count	wc in cells/cumm	numeric	Hard
red blood cell count	rc in millions/cmm	numeric	Hard
diabetes mellitus	diabetes, yes/no	nominal	Hard
coronary artery disease	cad, yes/no	nominal	Hard
Anemia	Anemia, yes/no	nominal	Hard

Figure 3: Summary of attributes from the original dataset, their description, their format, and the difficult level of obtaining the attributes.

3 Results and Discussion

Running the model with the cleaned dataset with 25 attributes resulted in a perfect prediction across all models. Using this dataset would not give an accurate representation of the machine learning model, rather it shows that the attributes are too correlated with the classification, learning to model bias. Moving forward, all attributes rated a “Hard” will be discarded, as these attributes show too much correlation to CKD presence.

The original dataset had 24 attributes, but many were difficult to obtain because some attributes required lab tests which costs both money and time. The idea of building a machine learning model is to make it as convenient and quick as possible without having to spend money. Some attributes in this dataset defeat the purpose of a machine learning model because some of the tests used to get the attributes are the same tests doctors use to diagnose CKD as given with some attributes with df-moderate results on Figure 4. This is why we narrowed down 24 attributes to 6 attributes in df-easy.

The attributes include age, blood pressure, sugar level, red blood cell, pus cell, and appetite, which all can be tested during a regular doctor’s office visit. The original dataset had a lot of gaps, to get rid of the gaps, we had to delete all the rows that had a gap in it. This leaves us with a complete dataset without any gaps; after getting rid of all the gaps in df-easy, there were 147 remaining rows. But the original dataset only had 107 rows after we cleaned it, that means that a lot of data from the attributes rated as moderate or hard were difficult to get. As we can see from the results, the 6 attributes that were the easiest to get still had good accuracy and precision across all three models. This result and workflow show that even with an easily accessible data we can detect CKD presence with high accuracy and precision using ML algorithms.

3.1 Performance Evaluation

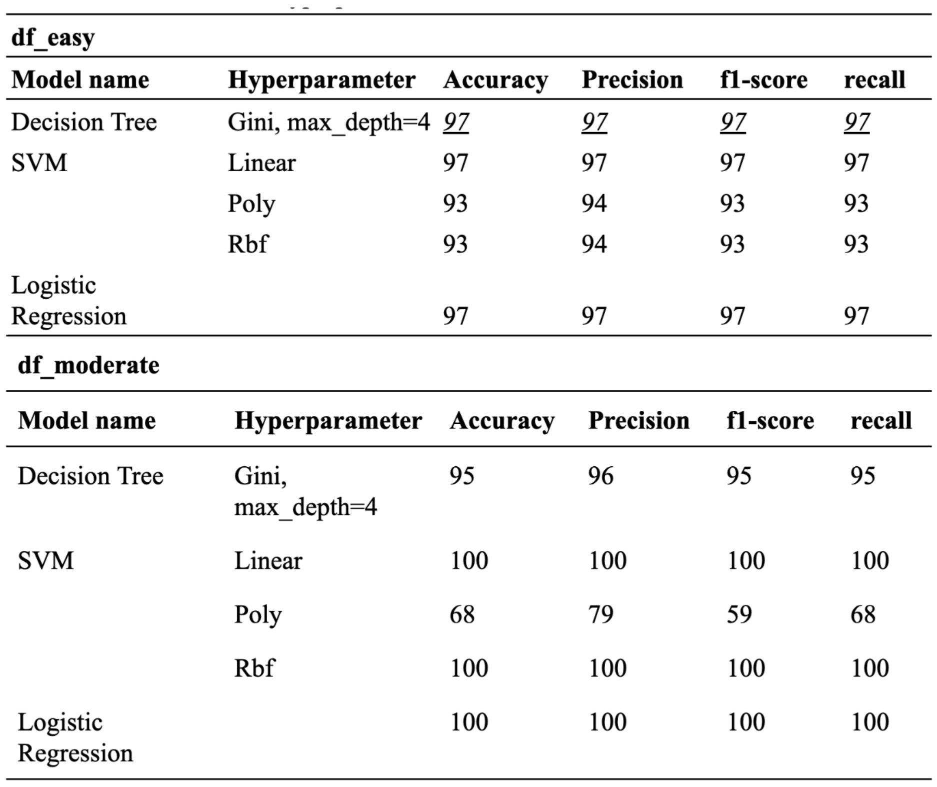
The metrics used to evaluate different models and hyperparameters are given in Figure 4. Given the objective is to classify whether a patient would show CKD symptoms, we used classification metrics as accuracy, precision, f1-score, and recall. The reason to provide multiple evaluation metrics is to give flexibility to evaluate different outcomes depending on the sensitivity of misdiagnosis for different diseases.

$$Recall = \frac{True\ positive}{True\ Positive + False\ Negative}$$

$$F1 - Score = \frac{2*Precision*Recall}{Precision + Recall}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{All\ predictions}$$



df_easy					
Model name	**Hyperparameter**	**Accuracy**	**Precision**	**f1-score**	**recall**
Decision Tree	Gini, max_depth=4	*97*	*97*	*97*	*97*
SVM	Linear	97	97	97	97
	Poly	93	94	93	93
	Rbf	93	94	93	93
Logistic Regression		97	97	97	97

df_moderate					
Model name	**Hyperparameter**	**Accuracy**	**Precision**	**f1-score**	**recall**
Decision Tree	Gini, max_depth=4	95	96	95	95
SVM	Linear	100	100	100	100
	Poly	68	79	59	68
	Rbf	100	100	100	100
Logistic Regression		100	100	100	100

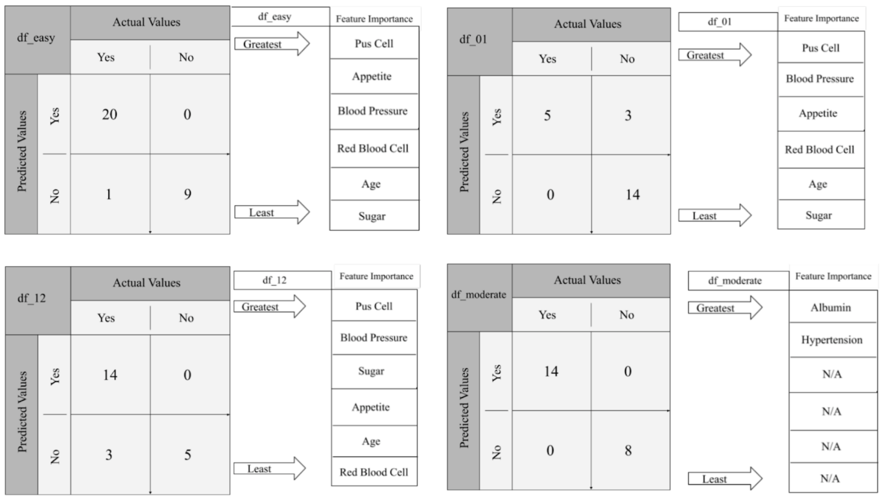
Figure 4: ML models and hyperparameters used to achieve the results.

Results on Figure 4 show the model hyperparameters and model accuracy, recall, precision, and f-1 score for different models and different datasets. Using the easily available 6 attributes in the df-easy dataset, we can achieve up to 97 percent accuracy in predicting CKD presence in the patient data. This was achieved both with Decision Tree, SVM model (using linear kernel), and Logistic Regression model. Moreover, as given in the next section, we can interpret the model results using feature-importance- in Decision Tree, which shows the contribution of different attributes to the final outcome. In our case, we found that pus cell attribute had the most important contribution in predicting CKD presence. Next two attributes were blood pressure and sugar or appetite level depending on the model iteration. Given the appetite values are subjective based on the individual, there is a higher confidence in the accuracy of sugar level in the patient. This was followed by red blood cell count and age. Furthermore, in the case of Logistic Regression, predict-proba gives the probability of the predicted result. This is an important feature since a prediction with 95 percent vs 60 percent confidence interval would not be the same, even though the prediction would result in the same class. While 95 percent confidence in the prediction accuracy can be trusted, 60 percent confidence level would require taking the test again to further validate the model prediction results in the patient.

As mentioned in the beginning of Results and Discussion, some attributes are highly correlated with the CKD presence and some of them are the tests taken to predict CKD. Even after removing the aforementioned attributes, we still see a high correlation for the case of df-moderate, which includes attributes that have the accessibility level of easy and moderate (Appendix). As given in Figure 4 for df-moderate, we achieve 100 percent prediction accuracy in SVM (both for linear and rbf kernels) and with Logistic Regression, probably due to the presence of highly correlated attributes. One surprising result from df-moderate is 95 percent accuracy using Decision Tree. This is probably due to the nature of the model where the model predicts the outcome using one feature at a time and sequentially choosing the features. This makes 100 percent accuracy difficult unless the split at all the trees are fully accurate.

3.2 Bridging the Gap Between ML Results and Medical Knowledge.

Looking at the feature importance table with df-moderate, the two most important features are Albumin and Hypertension. The rest of the features contribute an insignificant amount to the decision making of the model. Unsurprisingly, the two most important features are also rated a moderate which means that both features are relatively hard to obtain. Albumin is a protein found in blood, a healthy kidney does not let Albumin pass into urine. An increase in Albumin levels directly correlates with an increase in probability of CKD. Hypertension is prevalent in many cases of CKD and it is reported that 85 percent to 95 percent of all patients with CKD have hypertension. Besides the highly-correlated albumin level and hypertension for df-moderate, we see that pus cell, build-up of dead white blood cells that can be obtained using a dipstick at home, has the highest importance in df-easy dataset. Not surprisingly, this is followed by blood pressure and sugar levels. The next attributes were red blood cell and age. The sequence of selected features might not be directly correlated, meaning it is the combination of attributes that lead to the final prediction, not individual attributes themselves (to avoid highly correlated attributes). Furthermore, we can also see the df-easy and df-12 (df-easy with 1 and 2 data formatting) had a tendency to choose false negatives, meaning it often identified CKD cases as not CKD. On the other hand, df-01 (df-easy with 0 and 1 data formatting) had a tendency towards false positives, meaning the CKD patients were often not detected. This is important to know as it will help decide whether certain tests need repeating or further investigation to confirm the case.



df_easy		Actual Values	
		Yes	No
Predicted Values	Yes	20	0
	No	1	9

df_easy	Feature Importance
Greatest	Pus Cell
	Appetite
	Blood Pressure
	Red Blood Cell
	Age
Least	Sugar

df_01		Actual Values	
		Yes	No
Predicted Values	Yes	5	3
	No	0	14

df_01	Feature Importance
Greatest	Pus Cell
	Blood Pressure
	Appetite
	Red Blood Cell
	Age
Least	Sugar

df_12		Actual Values	
		Yes	No
Predicted Values	Yes	14	0
	No	3	5

df_12	Feature Importance
Greatest	Pus Cell
	Blood Pressure
	Sugar
	Appetite
	Age
Least	Red Blood Cell

df_moderate		Actual Values	
		Yes	No
Predicted Values	Yes	14	0
	No	0	8

df_moderate	Feature Importance
Greatest	Albumin
	Hypertension
	N/A
	N/A
	N/A
Least	N/A

Figure 5: Feature importance of Decision Tree for different datasets.

4 Conclusion

Using several machine learning algorithms and a generalizable workflow we developed, the prediction of CKD using preliminary data was possible and actually very practical as the results (accuracy, precision, recall, etc.) were high. We demonstrated 95 percent + using different models and model hyperparameters. We also demonstrated the flexibility of our workflow in terms of adjusting number of attributes, dataset size, and choice of the algorithm. We need to note that the data was collected exclusively in India and the data may vary from region to region. To improve the model, data should be gathered from all across the world and should be solely in the attributes categorized as easy. If successful, hospitals from all across the world can adopt these models and predict early stages of CKD, reducing the risk of severe stages of CKD to millions of people.

Horizon
Academic

Figure 6:

References

[Atk05] Robert Atkins. The epidemiology of chronic kidney disease. *Kidney International*, 2005.

[Bre21] Dan Brennan. What is stage 3 kidney disease life expectancy? *WebMD*, 2021.

[ea03] Andrew Levey et al. National kidney foundation practice guidelines for chronic kidney disease: evaluation, classification, and stratification. *Annals of internal medicine*, 2003.

[ea07] Ernesto Schiffrin et al. Chronic kidney disease: effects on the cardiovascular system. *Circulation*, 2007.

[Rub15] L.Jerlin Rubini. Chronic kidney disease data set. *UCI Machine Learning Repository*, 2015.

[Web17] Angela Webster. Chronic kidney disease. *The Lancet*, 2017.