Report

Chronic Kidney Disease Detection using Machine Learning

System Development Project in Chronic Kidney Disease

using Machine learning

deep learning,

Student: Noor Muhammad Tamim

Roll: 1807087

Course: CSE 3200

3rd Year 2nd term

Date of Submission: 28.12.2022

Abstract

Chronic kidney disease (CKD) is among the top 20 causes of death worldwide and affects approximately 10% of the world adult population. The WHO has shown that CKD is a serious disease, ranked as one of the top twenty causes of death. It is recognized that2 million people worldwide suffer from kidney failure and the number of patients diagnosed with CDK continues to expand at a rate of 5-7% annually. Early and accurate detection of the stages of CKD is believed to be vital to minimize impacts of patient's health complications such as hypertension, anemia (low blood count), low white blood cell count poor nutritional health with timely intervention through appropriate medications. CKD is a serious life-threatening disease, with high rates of morbidity and mortality. Therefore, artificial intelligence techniques are of great importance in the early detection of CKD. These techniques are supportive of experts and doctors in early diagnosis to avoid developing kidney failure. The aim of this project is to predict chronic kidney disease using Support Vector Machine(svm), Random Forest, K-Nearest Neighbour(KNN), Logistic Regression, AdaBoost with Naive Bayes, XGBoost, Majority Voting on the basis of accuracy.

Contents

1 Introduction **1**
1.1 Causes of Chronic Kidney Disease 1

2 Background **1**
2.1 Objectives 2

3 Data **2**

4 Design Procedure **2**
4.1 Pre Processing Data 2
4.2 Materials and Methods 3

5 Models **3**
5.1 Logistic Regression 3
5.2 KNN 3
5.3 Naive Bayes 4
5.4 Random Forest 5
5.5 Support Vector Machine 6
5.6 AdaBoost 7
5.7 XGBoost 7
5.8 Voting Classifier 8
5.9 Setup 9

6 Approach **9**
6.1 Deployment 10

7 Results **11**
7.1 Confusion Matrices 11
7.2 Suggestions 12

8 Discussion **12**
8.1 Limitations and Challenges 12

9 Conclusion **13**

10 Reference **13**

List of Figures

1 Logistic Regression 3
2 K-Nearest Neighbor 4
3 Naive Bayes 5
4 Random Forest 6
5 Support Vector Machine 7
6 XGBoost 8
7 Majority Voting 9
8 (a) Deployment using Streamlit 2 (b) Deployment using Streamlit 2 (c) Selectbox 10
9 (a) Logistic Regression (b) Support Vector Machine (c) KNN 11
10 (a) Random Forest (b) AdaBoost (c) XGBoost 12

1. Introduction

Chronic kidney disease, also called chronic kidney failure, involves a gradual loss of kidney function. Chronic kidney disease includes conditions that damage our kidneys and decrease their ability to keep you healthy by filtering wastes from your blood. Our kidneys filter wastes and excess fluids from Our blood, which are then removed in the urine. If kidney disease worsens, wastes can build to high levels in our blood and make us feel sick.

Kidney disease also increases our risk of having heart and blood vessel disease. These problems may happen slowly over a long time. Advanced chronic kidney disease can cause dangerous levels of fluid, electrolytes and wastes to build up in our body.

Early detection and treatment can often keep chronic kidney disease from getting worse. When kidney disease progresses, it may eventually lead to kidney failure, which requires dialysis or a kidney transplant to maintain life. In the early stages of chronic kidney disease, one might have few signs or symptoms. We might not realize that someone has kidney disease until the condition is advanced.

Treatment for chronic kidney disease focuses on slowing the progression of kidney damage, usually by controlling the cause. But, even controlling the cause might not keep kidney damage from progressing. Chronic kidney disease can progress to end-stage kidney failure, which is fatal without artificial filtering (dialysis) or a kidney transplant.

1.1 Causes of Chronic Kidney Disease

Diabetes and high blood pressure, or hypertension, are responsible for two-thirds of chronic kidney disease cases.

Diabetes: Diabetes occurs when your blood sugar remains too high. Over time, unmanaged blood sugar can cause damage to many organs in your body, including the kidneys and heart and blood vessels, nerves, and eyes.

High blood pressure: High blood pressure occurs when your blood pressure against the walls of your blood vessels increases. If uncontrolled or poorly controlled, high blood pressure can be a leading cause of heart attacks, strokes, and chronic kidney disease. Also, chronic kidney disease can cause high blood pressure.

2. Background

Renal failure can be avoided if chronic kidney disease is detected and treated early. The best course of action for treating chronic kidney disease is early diagnosis, but waiting until it has progressed to this point will result in renal failure, which necessitates ongoing dialysis or kidney transplantation to sustain a normal life. A blood test to evaluate the glomerular filtrate or a urine test to assess albumin are the two medical procedures used to diagnose chronic kidney disease (CKD). There is a need for computer-assisted diagnostics to aid with the growing number of chronic renal patients, the lack of specialized doctors, and the high costs of diagnosis and treatment, particularly in developing nations.

2.1 Objectives

- To detect Chronic kidney Disease using machine learning
- To be familiar with machine learning classifiers
- To learn how to deploy machine learning models as a web application

3. Data

Given 25 health related attributes taken in 2-month period of 400 patients, using the information of the 150 patients with complete records to predict the outcome (i.e. whether one has chronic kidney disease) of the remaining 250 patients (with missing values in their records).

The dataset used in this project can be found at: https://www.kaggle.com/code/csyhuang/predicting-chronic-kidney-disease/notebook

4. Design Procedure

4.1 Pre Processing Data

The dataset was pre processed to make it more applicable for the machine learning models.

1. There were a number of data with missing parameters. We replaced the missing data with Not a Number(NaN) for future identification.
2. We replaced the NaN values with most frequent data of respective columns. We used simple imputer for this purpose.
3. We used label encoding to convert categorical values to numerical values.
4. We then corrected data types of respective columns.
5. Checked whether the data was normally distributed or not.
6. We spllit the whole dataset into two dataframes, one consisting of the classifier column and the other consisting of all the other columns
7. Finally we split the whole dataset into two subsets; train subset and test subset. Train subset is used to train the model and the test subset is used make predictions and compare to the expected values. Train subset contains 80% data and test subset contains 20% data of the total dataset.

4.2 Materials and Methods

The work in our project uses six classification techniques to predict chronic kidney disease of a person. We have used Logistic Regression, Gaussian Naive Bayes, KNN, Random Forest, AdaBoost, XGBoost, Sequential Model.

5. Models

5.1 Logistic Regression

Logistic Regression is implemented as a linear model for classification rather than regression in terms of the scikit-learn/ML nomenclature. The logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.
This implementation can fit binary, One-vs-Rest, or multinomial logistic regression with optional ℓ_1,ℓ_2 or Elastic-Net regularization.
Logistic regression is a special case of Generalized Linear Models with a Binomial / Bernoulli conditional distribution and a Logit link. The numerical output of the logistic regression, which is the predicted probability, can be used as a classifier by applying a threshold (by default 0.5) to it. This is how it is implemented in scikit-learn, so it expects a categorical target, making the Logistic Regression a classifier.

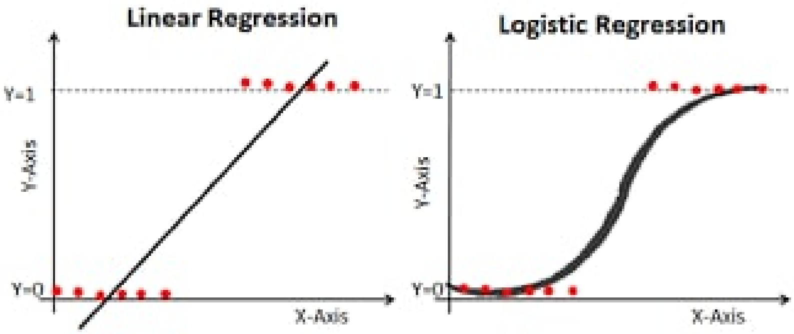


Figure 1: Logistic Regression

5.2 KNN

The k-nearest neighbors algorithm (k-NN) is a non-parametric supervised learning method. The input consists of the k closest training examples in a data set.

- In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically.
Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.
The neighbors are taken from a set of objects for which the class (for k-NN classification) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.
A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.

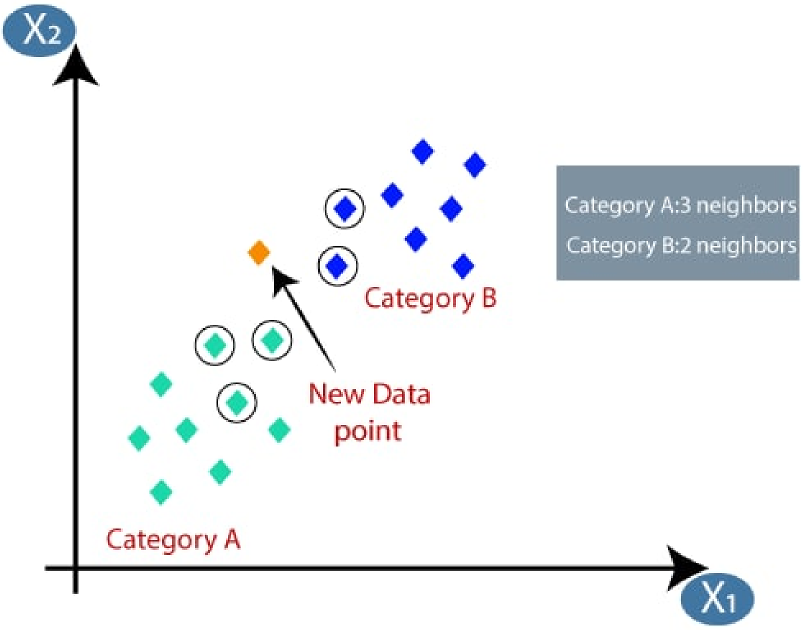


Figure 2: K-Nearest Neighbor

5.3 Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable and dependent feature vector through , : The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i \mid y)$. In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

GaussianNB implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

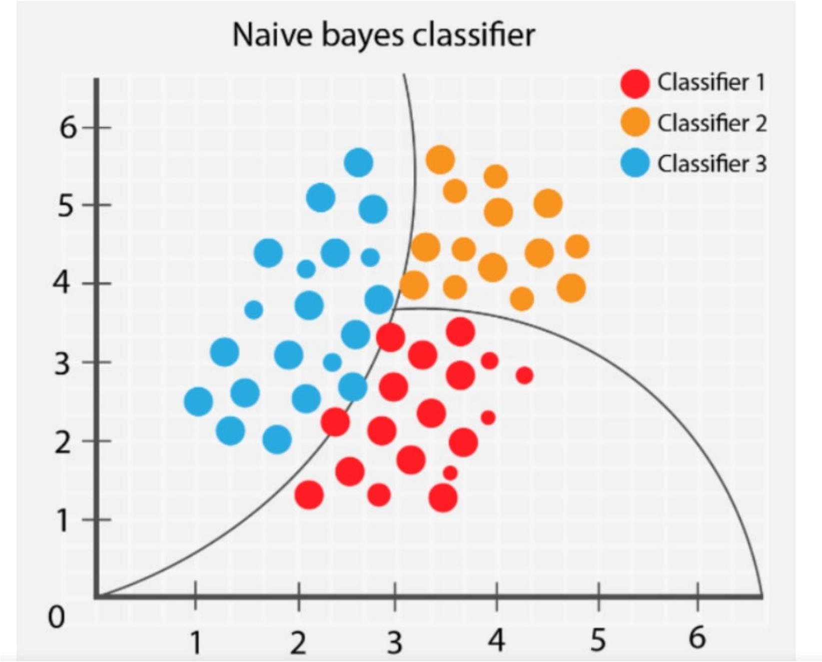


Figure 3: Naive Bayes

5.4 Random Forest

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

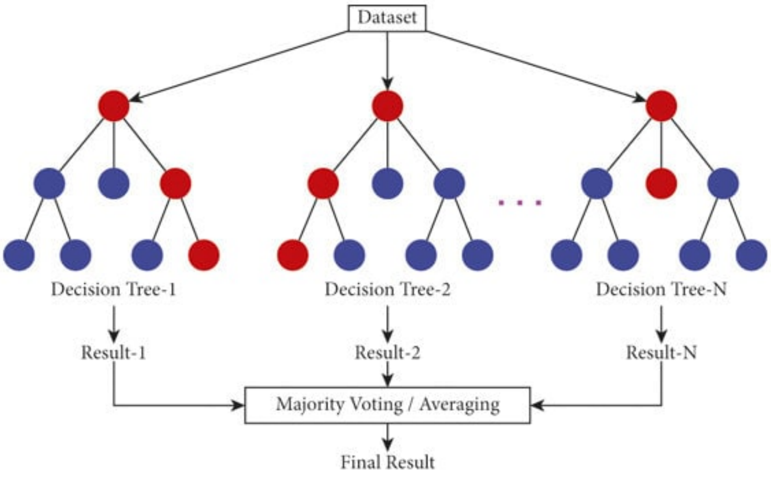


Figure 4: Random Forest

5.5 Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

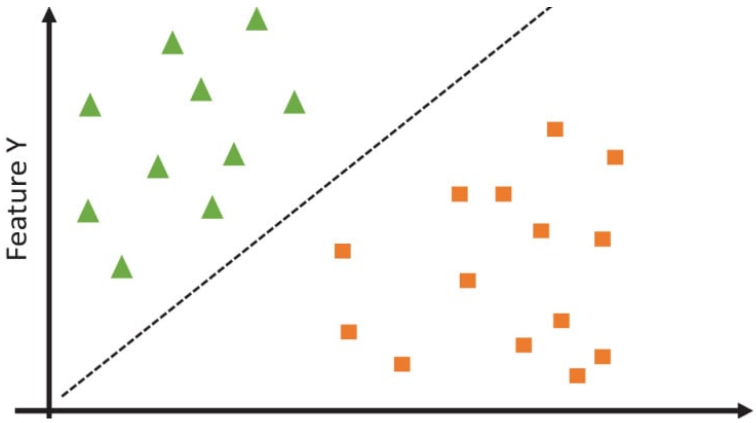


Figure 5: Support Vector Machine

5.6 AdaBoost

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying weights w_1, w_2, $\ldots$,w_N to each of the training samples. Initially, those weights are all set to $w_i = 1/N$, so that the first step simply trains a weak learner on the original data. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the re-weighted data. At a given step, those training examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. As iterations proceed, examples that are difficult to predict receive ever-increasing influence. Each subsequent weak learner is thereby forced to concentrate on the examples that are missed by the previous ones in the sequence

5.7 XGBoost

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting. Supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features and then uses the trained model to predict the labels on a new dataset's features. Decision trees create a model that predicts the label by evaluating a tree of if-then-else true/false feature questions, and estimating the minimum number of questions needed to assess the probability of making a correct decision. A Gradient Boosting Decision Trees (GBDT) is a decision tree ensemble learning algorithm similar to random forest, for classification and regression. Ensemble learning algorithms combine multiple machine learning algorithms to obtain a better model.

In XGBoost, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model.

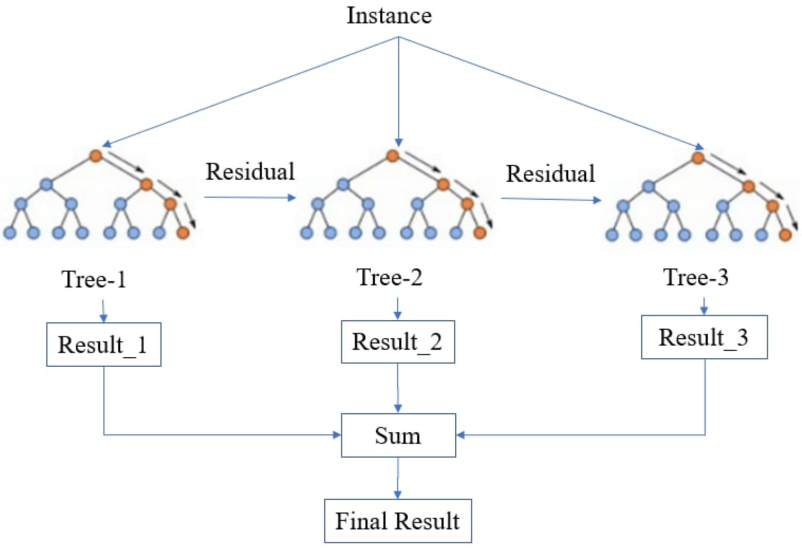


Figure 6: XGBoost

5.8 Voting Classifier

The idea behind the **VotingClassifier** is to combine conceptually different machine learning classifiers and use a majority vote or the average predicted probabilities (soft vote) to predict the class labels. Such a classifier can be useful for a set of equally well performing models in order to balance out their individual weaknesses.

5.8.1 Majority Class Labels (Majority/Hard Voting)

In majority voting, the predicted class label for a particular sample is the class label that represents the majority (mode) of the class labels predicted by each individual classifier.

E.g., if the prediction for a given sample is

- classifier 1 $->$ class 1
- classifier 2 $->$ class 1
- classifier 3 $->$ class 2

the VotingClassifier (with voting='hard') would classify the sample as "class 1" based on the majority class label.

In the cases of a tie, the VotingClassifier will select the class based on the ascending sort order. E.g., in the following scenario

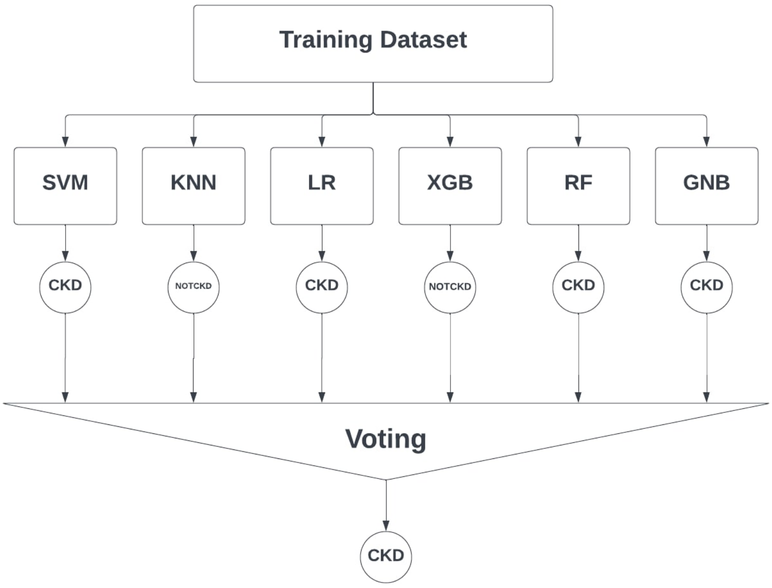


Figure 7: Majority Voting

5.9 Setup

We have used python as the primary language for running our models. Jupyter Notebook was used as a tool for running the codes.

6. Approach

After pre processing the data we checked for correlations to measure of the strength of association between two variables. We applied Support Vector Machine Classifier to train our machine learning model. We acquired moderate accuracy with basic parameters so we applied hyper parameter tuning(GridSearchCV) to find the best parameters for support vector machine. The best parameter for SVM is **C= 100, gamma= 'scale', kernel= 'linear')**. The best training accuracy for SVM was.

Secondly, we applied Logistic Regression Classifier to train the machine learning model. After applying Hyper parameter tuning for logistic regression classifier we found out the best parameters were **(C = 100, max_iter = 2000, penalty= 'l2', solver = 'newton-cg')**. The best Training accuracy was 1.000 and Test accuracy was 0.988. Hyper parameter tuning checks combination of all the given parameters for the optimal result between those parameters.

Then we applied K-Nearest Neighbour Classifier and applied GridSearchCV to find the best parameters to train the machine learning model. The best parameters were **algorithm = 'ball_tree', n_jobs= 1, n_neighbors= 6, weights= 'uniform')**. The best Training accuracy was 0.834 and Test accuracy was 0.662.

After that, we applied Random Forest Classifier and used GridSearchCV for Hyper parameter tuning. After hyper paramter tuning we found the best parameters to be **(bootstrap =True, criterion = 'entropy', max_features = 'auto', min_samples_leaf = 1, min_samples_split = 4, n_estimators = 100)**

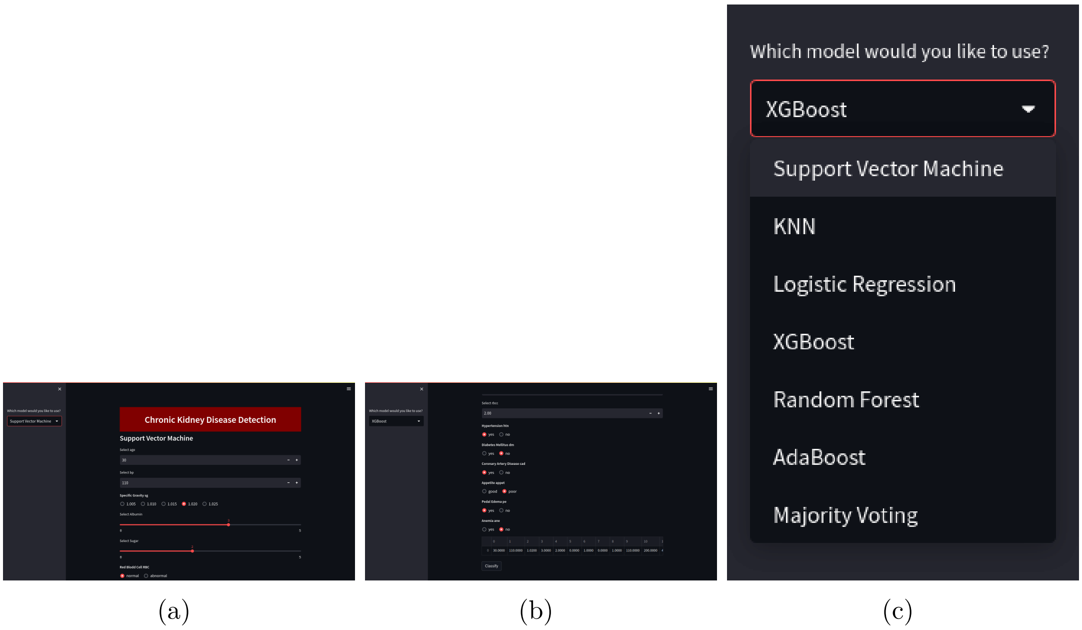
Then we applied AdaBoost Classifier with Gaussian Naive Bayes and applied GridSearchCV to find the best parameters to train our machine learning model. The best parameters were **(base_estimator=GaussianNB(), learning_rate = 0.2, n_estimators = 10)**. The best Training accuracy was 1.000 and Test accuracy was 0.988.

Finally, we applied Logistic Regression Classifier to train the machine learning model. After applying Hyper parameter tuning for logistic regression classifier we found out the best parameters were **(colsample_bytree = 0.5, gamma= 0.0, learning_rate = 0.15, max_depth = 5, min_child_weight = 1)**. The best Training accuracy was 1.000 and Test accuracy was 1.000.

After applyig all the classifiers, we used majority voting to find the best classifier among the tested classifiers.

6.1 Deployment

After training our machine learning models with our taken classifiers we deployed our machine learning model as a web application. We deployed our model with the help of streamlit library of python. Streamlit is an open-source app framework for Machine Learning and Data Science.



Which model would you like to use?
Support Vector Machine

Chronic Kidney Disease Detection

Support Vector Machine

Select age
30

Select bp
110

Specific Gravity sg
○ 1.005 ○ 1.010 ○ 1.015 ● 1.020 ○ 1.025

Select Albumin
3 (0–5)

Select Sugar
2 (0–5)

Red Blodd Cell RBC
● normal ○ abnormal

(a)

Which model would you like to use?
XGBoost

Select rbcc
2.00

Hypertension htn
● yes ○ no

Diabetes Mellitus dm
○ yes ● no

Coronary Artery Disease cad
● yes ○ no

Appetite appet
○ good ● poor

Pedal Edema pe
● yes ○ no

Anemia ane
○ yes ● no

	0	1	2	3	4	5	6	7	8	9	10	1
0	30.0000	110.0000	1.0200	3.0000	2.0000	0.0000	1.0000	0.0000	1.0000	110.0000	200.0000	4

Classify

(b)

Which model would you like to use?
XGBoost

(c)

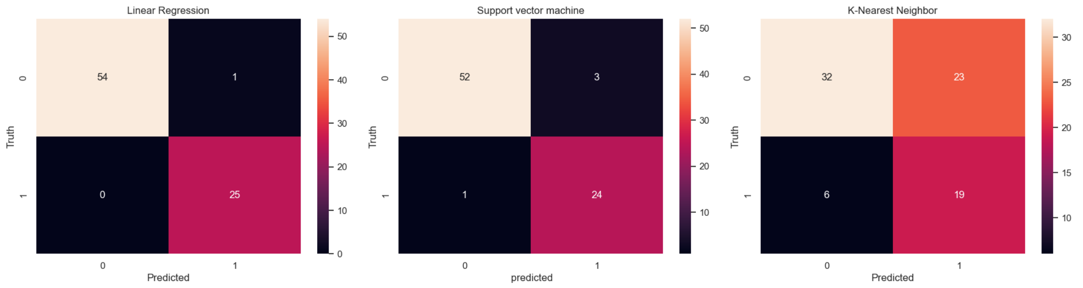
Figure 8: (a) Deployment using Streamlit 2 (b) Deployment using Streamlit 2 (c) Selectbox

7. Results

We applied Logistic Regression, K-Nearest Neighbour, Support Vector Machine, Random Forest, AdaBoost with Naive Bayes, XGBoost classifiers to predict the chance of having chronic kidney disease of a patient. After that we used majority voting to combine different machine learning classifiers and check every individual classifiers vote for a class, and decide according to the majority vote. The overall performance of KNN is unsatisfactory. The random forest algorithm classified all positive and negative samples correctly, as positive samples were correctly classified 250 samples, and all negative samples were classified for 150 samples correctly. While the SVM, KNN, Logistic Regression, AdaBoost and XGBoost algorithms rated the positive samples by 98.1 percent, 83.4 percent, 100 percent, 100 percent and 100 percent.

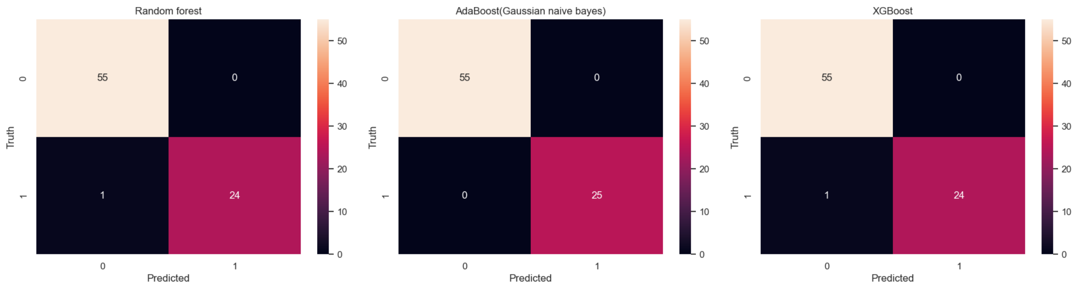
Results of different classifiers		
Model Name	Training Accuracy	Testing Accuracy
KNN	0.834	0.662
Logistic Regression	1.000	0.988
Support Vector Machine	0.981	0.988
Random Forest	1.000	1.000
AdaBoost	1.000	0.988
XGBoost	1.000	1.000
Majority Voting	0.98750	0.98750

7.1 Confusion Matrices



(a) (b) (c)

Figure 9: (a) Logistic Regression (b) Support Vector Machine (c) KNN



(a) (b) (c)

Figure 10: (a) Random Forest (b) AdaBoost (c) XGBoost

7.2 Suggestions

How do you deal with kidney problems? Ways to manage chronic kidney disease

- Control blood pressure.
- Meet blood glucose goal if you have diabetes.
- Work with health care team to monitor your kidney health.
- Take medicines as prescribed.
- Work with a dietitian to develop a meal plan.
- Make physical activity part of routine.
- Aim for a healthy weight.

8. Discussion

We used Machine Learning Classification techniques to determine chronic kidney disease of a patient. We used KNN, SVM, Logistic Regression, Random Forest, AdaBoost, XGBoost. We also used Majority Voting to get the best classifier among all of our classifiers. Machine learning algorithms are used for the early diagnosis of CKD. Many factors affect kidney performance, which induce CKD, like diabetes, blood pressure, heart disease, some kind of food, and family history. This project helped kidney patients to get their kidney health status. Patients could be checked for chronic kidney disease and get proper treatment. Detection of kidney disease at an early stage can help save the lives of many people.

8.1 Limitations and Challenges

If our dataset had been larger we could have trained the machine more efficiently.

9. Conclusion

The development of this application and the public sharing of it might minimize the restrictions of future study on it. The proposed models produces binary classification, which helps to determine chronic kidney disease patients and help in their sound health. In future, we wish to use our project to get better accuracy. We also wish to implement the project on a much larger scale to impact the lives of millions of people.

10. Reference

1. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
2. A. S. Levey, R. Atkins, J. Coresh et al., "Chronic kidney disease as a global public health problem: approaches and initiatives–a position statement from kidney disease improving global outcomes," Kidney International, vol. 72, no. 3, pp. 247–259, 2007.
3. V. Jha, G. Garcia-Garcia, K. Iseki et al., "Chronic kidney disease: global dimension and perspectives," The Lancet, vol. 382, no. 9888, pp. 260–272, 2013.
4. N. R. Hill, S. T. Fatoba, J. L. Oke et al., "Global prevalence of chronic kidney disease – a systematic review and meta-analysis," PLoS One, vol. 11, no. 7, article e0158765, 2016.
5. H. Nasri, "World kidney day 2014; chronic kidney disease and aging: a global health alert," Iranian Journal of Public Health, vol. 43, no. 1, pp. 126-127, 2014.
6. L. Ali, K. Fatema, Z. Abedin et al., "Screening for chronic kidney diseases among an adult population," Saudi Journal of Kidney Diseases and Transplantation, vol. 24, no. 3, p. 534, 2013.

Bibliography