

**A Project Report on**  
**Chronic Kidney Disease Diagnosis using Machine Learning**  
**Department of Cyber Security & Data Science**

submitted in partial fulfillment for the award of

**Bachelor of Technology**

in

**Data Science**

by

**G. Jaya Lakshmi (Y20ADS409)      G. Durgesh Reddy (Y20ADS408)**

**A. Devi Sesha Sai (Y20ADS401)      K. Pallavi Ramani (Y20ADS418)**



Under the guidance of  
**Ass. Prof. P.V. Naga Srinivas.**

Department of Cyber Security & Data science  
**Bapatla Engineering College**  
(Autonomous)  
(Affiliated to Acharya Nagarjuna University)  
**BAPATLA – 522 102, Andhra Pradesh, INDIA**  
**2023-2024**

**Department of  
Cyber Security & Data Science**



**CERTIFICATE**

This is to certify that the project report entitled **Chronic Kidney Disease Diagnosis using Machine Learning** that is being submitted by G. Jaya Lakshmi (Y20ADS409), G. Durgesh Reddy (Y20ADS408), A. Devi Sesha Sai (Y20ADS401), and K. Pallavi Ramani (Y20ADS418) in partial fulfillment for the award of the Degree of Bachelor of Technology in Data Science to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

**Signature of the Guide**  
**P. V. Naga Srinivas**  
**Ass. Prof.**

**Signature of the HOD**  
**V. Chakradhar**  
**Prof. & Head**

## **DECLARATION**

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

**G. Jaya Lakshmi (Y20ADS409)**

**G. Durgesh Reddy (Y20ADS408)**

**A. Devi Sesha Sai (Y20ADS401)**

**K. Pallavi Ramani (Y20ADS418)**

## Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for the successful completion of the work.

We are deeply indebted to our most respected guide **P. V. Naga Srinivas**, Ass. Prof., Department of Cyber Security & Data Science, for his/her valuable and inspiring guidance, comments, suggestions, and encouragement.

We extend our sincere thanks to **V. Chakradhar**, Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. Nazeer Shaik** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **G.V. Leela Kumari**, Ass. Prof. Dept. of Cyber Security & Data Science for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

**G. Jaya Lakshmi (Y20ADS409)**  
**G. Durgesh Reddy (Y20ADS408)**  
**A. Devi Sessa Sai (Y20ADS401)**  
**K. Pallavi Ramani (Y20ADS418)**

# Table of Contents

List of Figures .....	vii
List of Equations.....	viii
Abstract.....	ix
1 Introduction .....	1
1.1 Stages of CKD .....	2
1.1.1 Early stages of CKD.....	2
1.1.2 CKD in its advanced stages .....	2
2 Literature Survey.....	4
3 Problem Statement .....	7
4 Features .....	8
4.1 Feature Selection.....	8
4.1.1 Filter Methods.....	8
4.1.2 Wrapper methods .....	10
4.1.3 Embedded methods .....	11
5 Machine Learning .....	12
5.1 Supervised Learning.....	13
5.1.1 Regression .....	14
5.1.2 Classification .....	14
5.2 Unsupervised Learning.....	15
5.2.1 Clustering .....	15
5.2.2 Association .....	16
5.3 Semi-supervised Learning .....	16
5.4 Reinforcement Learning .....	17

6	Classification Algorithms .....	18
6.1	AdaBoost Classifier .....	19
6.2	Random Forest Classifier .....	20
6.3	Gradient Boosting Classifier.....	21
6.4	XgBoost Classifier .....	22
6.5	Extra Tree Classifier.....	23
6.6	Light Gradient Boosting Classifier.....	24
6.7	Decision Tree Classifier .....	25
6.8	Support Vector Machine Classifier.....	26
7	Software and Hardware Requirements .....	27
7.1	Software Requirements .....	27
7.2	Hardware Requirements .....	27
8	Implementation Review.....	28
8.1	Software Implementation .....	28
8.1.1	NumPy .....	28
8.1.2	Pandas .....	29
8.1.3	Scikit-learn .....	30
8.1.4	Matplotlib.....	31
8.1.5	Plotly .....	31
8.1.6	Flask.....	32
8.2	Machine Learning Model Implementation .....	34
8.3	Dataset collection and pre-processing .....	35
8.4	Data Splitting.....	36
8.5	Building the model.....	37
8.6	Training the model .....	38

9	Bibliography .....	39
---	--------------------	----

## List of Figures

Figure 1.1 Chronic Kidney Disease Symptoms .....	1
Figure 1.2 Stages of Kidney Disease .....	2
Figure 5.1 Supervised Learning Model .....	13
Figure 5.2 Types of Supervised Learning.....	14
Figure 5.3 Unsupervised Learning Model .....	15
Figure 5.4 Types of Unsupervised Learning.....	15
Figure 5.5 Semi supervised Learning model .....	16
Figure 6.1 Adaboost classifier Model .....	19
Figure 6.2 Random Forest Classifier Model.....	20
Figure 6.3 Gradient Boosting Model.....	21
Figure 6.4 XgBoost classifier Model .....	22
Figure 6.5 Extra Tree Classifiers model.....	23
Figure 6.6 Light Gradient Boosting Model .....	24
Figure 6.7 Decision Tree classifiers model.....	25
Figure 6.8 SVM Classifier.....	26
Figure 8.1 Machine Learning Model.....	34



## List of Equations

Equation 1 Chi-Square Equation .....	9
--------------------------------------	---

# Abstract

Chronic kidney disease (CKD) is a dangerous ailment that can last a person's entire life and is caused by either kidney malignancy or decreased kidney functioning. It is feasible to halt or slow the progression of this chronic disease to an end-stage wherein dialysis or surgical intervention is the only method to preserve a patient's life. Earlier detection and appropriate therapy can increase the likelihood of this happening. Throughout this research, the potential of several different machine learning approaches for providing an early diagnosis of CKD has been investigated. There has been a significant amount of research conducted on this topic. Nevertheless, we are bolstering our approach by making use of predictive modeling. Therefore, in our approach, we investigate the link that exists between data factors as well as the characteristics of the target class. We are capable of constructing a collection of prediction models with the help of machine learning and predictive analytics, thanks to the better measures of attributes that can be introduced using predictive modeling. This study starts with 25 variables in addition to the class property, but by the end, it has narrowed the list down to 65% of those parameters as the best subset to identify CKD. Seven different machine learning-based classifiers have been tested in a supervised learning environment. Within the confines of a supervised learning environment, a total of 7 different machine learning-based classifiers have indeed been examined, with the greatest performance indicators being an accuracy of 1.00, a precision of 0.98, a recall of 0.98, and an F1-score of 0.98 for the Light Gradient Boosting classifier. The way the research was done leads to the conclusion that recent improvements in machine learning, along with the help of predictive modeling, make for an interesting way to find new solutions that can then be used to test the accuracy of prediction in the field of kidney disease and beyond.

# 1 Introduction

Chronic kidney disease, or CKD, is a condition in which the kidneys are so damaged that they can't filter blood as well as they should. The kidneys' main job is to get rid of waste and extra water from the blood. This is how urine is made. CKD means that waste has built up in the body. This condition is called chronic because the damage happens slowly over a long period of time. It is a disease that affects people all over the world. Because of CKD, you might experience various difficulties with your health. Diabetes, high blood pressure, and heart disease are only 3 of the many conditions that can lead to CKD. In addition to these serious health problems, age and gender also play a role in who gets a CKD. If one or both of your kidneys aren't working right, you may have several symptoms, such as back pain, stomach pain, diarrhea, fever, nosebleeds, rash, and vomiting. The 2 most common illnesses that might cause long-term damage to the kidneys are diabetes and high blood pressure. Therefore, the prevention of CKD can be thought of as the control of these 2 diseases [1]. Because chronic kidney disease (CKD) does not often present any symptoms until it has progressed to a more advanced state, many people who have it do not realize they have it until it is too late. The dataset consists of several columns such as Age, Blood Pressure, Sugar, Albumin, Specifi\_gravity, Blood Glucose, Blood Urea, Serum Creatine, sodium, Potassium, Haemoglobin, Packed cell volume, Red Blood cell count, Hypertension, Anaemia, etc...

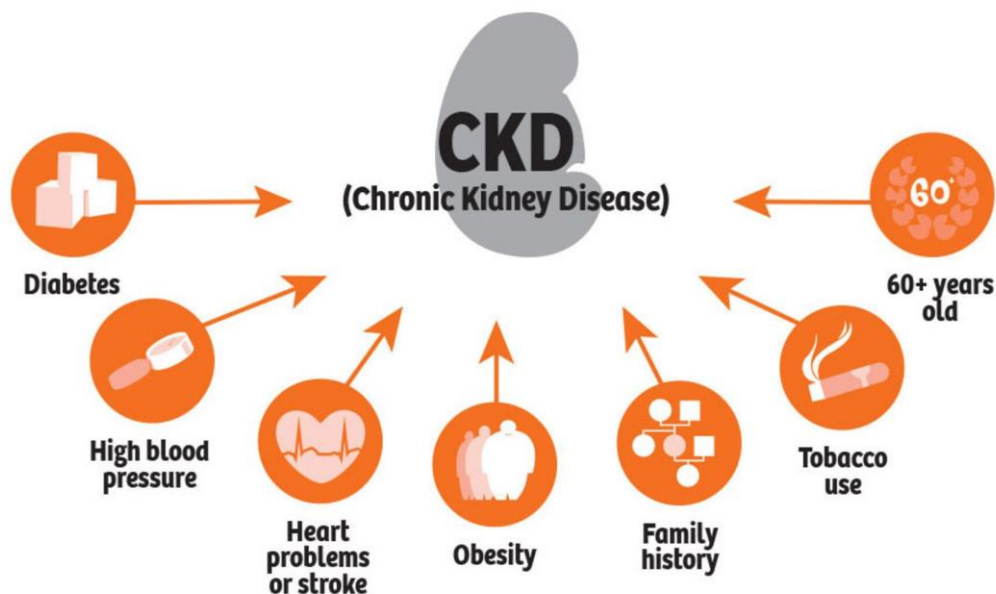


Figure 1.1 Chronic Kidney Disease Symptoms






## 1.1 Stages of CKD

### 1.1.1 Early stages of CKD

CKD in its early stages typically does not present any symptoms. This is due to the fact that the human body can typically adjust to a large decrease in the function of the kidneys. It is common for kidney disease to not be diagnosed until this stage unless a routine test for another issue, such as a test of the blood or urine, discovers a potential problem. If it is discovered at an early stage, treatment with medication and ongoing monitoring with routine tests may help prevent it from progressing to a more advanced state.

### 1.1.2 CKD in its advanced stages

If kidney disease isn't caught early or keeps getting worse even after treatment, there may be few signs.

5 Stages Of Kidney Disease					
Stage 1	Stage 2	Stage 3A	Stage 3B	Stage 4	Stage 5
$\text{GFR} \geq 90$	$89 \geq \text{GFR} \geq 60$	$59 \geq \text{GFR} \geq 40$	$44 \geq \text{GFR} \geq 30$	$29 \geq \text{GFR} \geq 15$	$\text{GFR} < 15$
					
Normal or high function	Mildly decreased function	Mild to moderately decreased function		Severely decreased function	Kidney failure

*Figure 1.2 Stages of Kidney Disease*

Kidney failure is the last stage of CKD. It is also called end-stage renal disease or established renal failure. It is possible that dialysis or a kidney transplant will be needed at some point.

#### 1.1.2.1 When to see a physician

If you have signs or symptoms of renal illness, make an appointment with your doctor. Renal disease could be prevented from progressing to kidney failure if detected early. During office visits, your doctor may use urine and blood tests to check your blood pressure and kidney function if you have a health condition that makes you more likely to get renal disease. Ask your physician if these tests are required for you.

### **1.1.2.2 Tests for CKD**

Chronic kidney disease is when a disease or condition makes it hard for the kidneys to work, causing the damage to the kidneys to get worse over time. This can occur when the kidneys are affected by another disease or condition. Studies show that the number of people with CKD who are admitted to hospitals is going up by 6.23 percent every year, even though the global death rate has stayed the same. There are just a few diagnostic tests available to check the status of CKD, including:

1. estimated glomerular filtration rate (eGFR).
2. a urine tests.
3. a blood pressure reading.
4. Other tests for CKD.

#### **1.1.2.2.1 eGFR**

The eGFR value provides information on how well your kidneys cleanse the blood. If your eGFR number is higher than 90, it means that your kidneys are working well. If the value of your eGFR is less than 60, this indicates that you have CKD.

#### **1.1.2.2.2 Urine Tests**

In order to evaluate kidney function, the physician also requests a urine sample. Urine is produced by the kidneys. If your urine contains blood and protein, it is an indication that one or both of your kidneys are not functioning normally.

#### **1.1.2.2.3 Blood Pressure**

The doctor takes your blood pressure because the range of your blood pressure reveals how well your heart is pumping blood. If the patient's eGFR value falls below 15, this means they have reached the end stage of kidney disease. There are just two treatments that are now available for renal failure:

- (i) dialysis and
- (ii) kidney transplantation.

The patient's life expectancy after dialysis is contingent on a number of characteristics, including age, gender, the frequency and length of dialysis treatments, the patient's level of physical mobility, and their

mental state. Kidney transplantation is the only option left for the doctor to consider if dialysis cannot be performed successfully. Nevertheless, the price is exorbitantly high.

#### **1.1.2.2.4 Other Tests for CKD**

When determining the extent of the damage to your kidneys, it is not uncommon for additional tests to be performed. These may include an ultrasound scan, a magnetic resonance imaging scan, or a computed tomography scan. Their purpose is to look at the kidneys and see if there are any blockages. A needle is used to take a small piece of kidney tissue, and the cells are looked at under a microscope to look for signs of kidney disease. This is done in order to diagnose kidney conditions. The field of medicine is an extremely important area for the application of intellectually sophisticated systems. Then, data mining could be a big part of finding hidden information in the huge amount of patient medical and treatment data. This is information that doctors often get from their patients to learn more about their symptoms and make more accurate treatment plans.

## **2 Literature Survey**

Data mining is the process of using specialized software to find hidden information in a large set of data. [1]. Data mining techniques are linked to each other and used in a wide range of places and situations. With data mining technologies, we can make predictions, sort the data, filter it, and put it into groups. The goal of the algorithm is to process a training set that has a collection of attributes and targets, and the objective describes how this should be done. If the dataset is very big, data mining is a good way to find patterns in it. If the dataset is very small, however, we can still reach the same goal with the help of machine learning.

Data analysis and pattern recognition are 2 further capabilities of machine learning. Because there is such a wide diversity of health datasets, machine learning algorithms are the most appropriate method for enhancing the accuracy of diagnosis prediction. The prevalence of machine learning algorithms in the healthcare industry is growing as a direct result of the rapid growth of electronic healthcare datasets. Using information mining techniques, a variety of types of studies have been carried out in order to extract useful information from datasets pertaining to chronic kidney disease. This was done in order to cut down on the amount of time spent conducting the analysis, and in addition to that, it would increase the precision of the forecast with the assistance of the information mining categorization technique.

Data mining is also applied in the treatment and diagnosis of a number of diseases and conditions. Using techniques for information accumulation, different kinds of work have been done to get useful information out of the dataset on chronic kidney disease. Polat et al. offered directions that combined a total of 6 classifiers and 3 outfit measures. K-Nearest Neighbors (KNN), Naive Bayes (NB), support vector machine (SVM), preference tables, random forest (RF), and J48 were some of the classifiers that were used. The authors are looked into a number of possible treatments for chronic renal disease by using the k-means algorithm and Apriori. A test that uses SVM, DT, NB, and KNN computations was developed in order to diagnose chronic kidney disease (CKD) [2]. The dataset of patients with CKD is used to make these predictive models. Then, the performances of these models are compared to find out which classifier is the best at predicting which patients will get CKD.

The UCI machine learning repository's CKD dataset has many missing values. KNN imputation was used to fill in missing values, which finds full samples with identical measurements for each incomplete sample. Real-life medical scenarios often have missing values because, patients miss measurements. After completing the dataset, 6 machine learning algorithms (LR, RF, SVM, KNN, and NB) were used to create models. With 99.75% accuracy, RF performed best. By examining the established models' misjudgements, they created a model that combines LR and RF utilizing perceptron's, which achieved 99.83% accuracy after 10 simulations.

They hypothesized that this technology may be used to diagnose complex diseases using clinical data. Almasoud and Ward wants to test machine learning algorithms' capacity to forecast CKD with the fewest features. Analysis of variance (ANOVA), Pearson's correlation, and Cramer's V tests were used to remove redundant features. The gradient boosting classifier has a 100% accuracy rate. Hemoglobin is more important for RF and gradient boosting in identifying CKD. Their results are high compared to earlier studies, although they've reached fewer characteristics. Throughout this study, we tried to evaluate the possibility of a variety of machine learning algorithms, each of which could potentially provide an early diagnosis of CKD. [2] There has been a substantial amount of research carried out on this subject; nonetheless, we are strengthening our strategy by making use of predictive modeling.

As a result, in our methodology, we study the link that exists between the data variables and the characteristics of the target class. Because predictive modeling allows for a more accurate measurement of attributes to be introduced, we are able to use machine learning and predictive analytics to compile a

set of prediction models. This is made possible by the improved ability of predictive modeling to introduce new attributes or identify the most important features responsible for CKD.

The research on the detection of CKD is based on 1 dataset which is available in the UCI machine learning repository. The dataset includes 24 input features used by the maximum research mentioned above. No work has been found for detecting CKD based on the least number of predictors. In addition to improving the accuracy using this dataset, we tried to reduce the number of input features using SelectKBest method in scikit-learn and tried to develop a machine learning- based model showing the highest accuracy.



### 3 Problem Statement

Healthcare sector is totally different from other industry. It is on high priority sector and people expect highest level of care and services regardless of cost. After the success of machine learning in other real-world application, it is also providing exciting solutions with good accuracy for medical imaging and is a key method for future applications in health sector. Kidney is an organ that is used to eliminate excess bodily fluid, salts and byproducts of metabolism – this makes kidneys key in the regulation of acid-base balance, blood pressure, and many other homeostatic parameters. Recognition of automated chronic kidney disease in Magnetic Resonance Imaging (MRI) is a difficult task due to complexity of size and location variability [5]. In this Problem, we reduced attributes or features using SelectKBest method in Scikit-learn. The results produced by this approach will increase the accuracy. MRI contains fine information for treatment. Texture of MRI contains information of size, shape, colour, and brightness that texture properties help to detect texture extraction. The main theme of this application is to increase the Accuracy of a model or application.

## 4 Features

Feature Selection is the process of reducing the number of input variables when developing a model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performances of the model.

### 4.1 Feature Selection

Statistical-based feature selection methods involve evaluating the relationship between each input variable and the target variable using statistics and selecting those input variables that have the strongest relationship with the target variable. These methods can be fast and effective, although the choice of statistical measures depends on the data type of both the input and output variables.

Feature selection helps in improving the accuracy of the prediction as the feature which contribute more to the classification are selected through it.

Some popular techniques of feature selection in machine learning are:

- Filter methods
- Wrapper methods
- Embedded methods

#### 4.1.1 Filter Methods

These methods are generally used while doing the pre-processing step. These methods select features from the dataset irrespective of the use of any machine learning algorithm. In terms of computation, they are very fast and inexpensive and are very good for removing duplicated, correlated, redundant features but these methods do not remove multicollinearity. Selection of feature is evaluated individually which can sometimes help when features are in isolation (don't have a dependency on other features) but will lag when a combination of features can lead to increase in the overall performance of the model.

Set of all features → Selecting the best subset → Learning algorithm → Performance

Some techniques used are:

- **Information Gain** – It is defined as the amount of information provided by the feature for identifying the target value and measures reduction in the entropy values. Information gain of each attribute is calculated considering the target values for feature selection.
- **Chi-square test** — Chi-square method ( $\chi^2$ ) is generally used to test the relationship between categorical variables. It compares the observed values from different attributes of the dataset to its expected value.

$$\chi^2 = \sum \frac{(\text{Observed value} - \text{Expected value})^2}{\text{Expected value}}$$

*Equation 1 Chi-Square Equation*

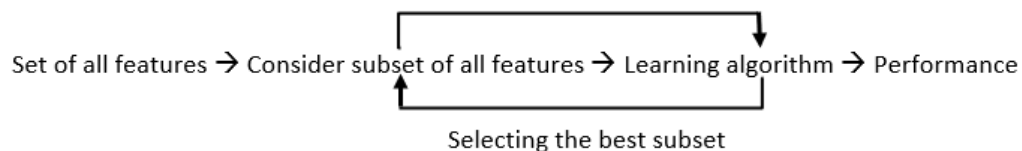
- **Fisher's Score** – Fisher's Score selects each feature independently according to their scores under Fisher criterion leading to a suboptimal set of features. The larger the Fisher's score is, the better is the selected feature.
- **Correlation Coefficient** – Pearson's Correlation Coefficient is a measure of quantifying the association between the two continuous variables and the direction of the relationship with its values ranging from  $-1$  to  $1$ .
- **Variance Threshold** – It is an approach where all features are removed whose variance doesn't meet the specific threshold. By default, this method removes features having zero variance. The assumption made using this method is higher variance features are likely to contain more information.
- **Mean Absolute Difference (MAD)** – This method is similar to variance threshold method, but the difference is there is no square in MAD. This method calculates the mean absolute difference from the mean value.
- **Dispersion Ratio** – Dispersion ratio is defined as the ratio of the Arithmetic mean (AM) to that of Geometric mean (GM) for a given feature. Its value ranges from  $+1$  to  $\infty$  as  $AM \geq GM$  for a given feature. Higher dispersion ratio implies a more relevant feature.
- **Mutual Dependence** – This method measures if two variables are mutually dependent, and thus provides the amount of information obtained for one variable on observing the other variable.

Depending on the presence/absence of a feature, it measures the amount of information that feature contributes to making the target prediction.

- **Relief** – This method measures the quality of attributes by randomly sampling an instance from the dataset and updating each feature and distinguishing between instances that are near to each other based on the difference between the selected instance and two nearest instances of same and opposite classes.

### 4.1.2 Wrapper methods

Wrapper methods, also referred to as greedy algorithms train the algorithm by using a subset of features in an iterative manner. Based on the conclusions made from training in prior to the model, addition and removal of features takes place. Stopping criteria for selecting the best subset are usually pre-defined by the person training the model such as when the performance of the model decreases or a specific number of features has been achieved. The main advantage of wrapper methods over the filter methods is that they provide an optimal set of features for training the model, thus resulting in better accuracy than the filter methods but are computationally more expensive.



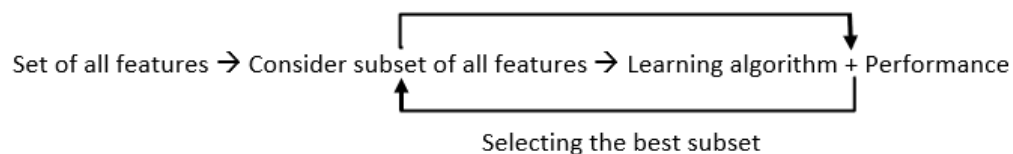
Some techniques used are:

- **Forward selection** – This method is an iterative approach where we initially start with an empty set of features and keep adding a feature which best improves our model after each iteration. The stopping criterion is till the addition of a new variable does not improve the performance of the model.
- **Backward elimination** – This method is also an iterative approach where we initially start with all features and after each iteration, we remove the least significant feature. The stopping criterion is till no improvement in the performance of the model is observed after the feature is removed.
- **Bi-directional elimination** – This method uses both forward selection and backward elimination technique simultaneously to reach one unique solution.

- **Exhaustive selection** – This technique is considered as the brute force approach for the evaluation of feature subsets. It creates all possible subsets and builds a learning algorithm for each subset and selects the subset whose model's performance is best.
- **Recursive elimination** – This greedy optimization method selects features by recursively considering the smaller and smaller set of features. The estimator is trained on an initial set of features and their importance is obtained using `feature_importance_attribute`. The least important features are then removed from the current set of features till we are left with the required number of features.

### 4.1.3 Embedded methods

In embedded methods, the feature selection algorithm is blended as part of the learning algorithm, thus having its own built-in feature selection methods. Embedded methods encounter the drawbacks of filter and wrapper methods and merge their advantages. These methods are faster like those of filter methods and more accurate than the filter methods and take into consideration a combination of features as well.



Some techniques used are:

- **Regularization** – This method adds a penalty to different parameters of the machine learning model to avoid over-fitting of the model. This approach of feature selection uses Lasso (L1 regularization) and Elastic nets (L1 and L2 regularization). The penalty is applied over the coefficients, thus bringing down some coefficients to zero. The features having zero coefficient can be removed from the dataset.
- **Tree-based methods** – These methods such as Random Forest, Gradient Boosting provides us feature importance as a way to select features as well. Feature importance tells us which features are more important in making an impact on the target feature.

## 5 Machine Learning

Machine Learning is defined as a technology that is used to train machines to perform various actions such as predictions, recommendations, estimations, etc., based on historical data or past experience.

Machine Learning enables computers to behave like human beings by training them with the help of past experience and predicted data.

There are tens of thousands of machine learning algorithms and hundreds of new algorithms are developed every year.

Every machine learning algorithm has three components:

- **Representation:** how to represent knowledge. Examples include decision trees, sets of rules, instances, graphical models, neural networks, support vector machines, model ensembles and others.
- **Evaluation:** the way to evaluate candidate programs (hypotheses). Examples include accuracy, prediction and recall, squared error, likelihood, posterior probability, cost, margin, entropy k-L divergence and others.
- **Optimization:** the way candidate programs are generated known as the search process. For example, combinatorial optimization, convex optimization, constrained optimization.

All machine learning algorithms are combinations of these three components. A framework for understanding all algorithms.

There are four types of machine learning:

- **Supervised learning:** (also called inductive learning) Training data includes desired outputs. This is spam this is not, learning is supervised.
- **Unsupervised learning:** Training data does not include desired outputs. Example is clustering. It is hard to tell what good learning is and what is not.
- **Semi-supervised learning:** Training data includes a few desired outputs.
- **Reinforcement learning:** Rewards from a sequence of actions. AI types like it, it is the most ambitious type of learning.

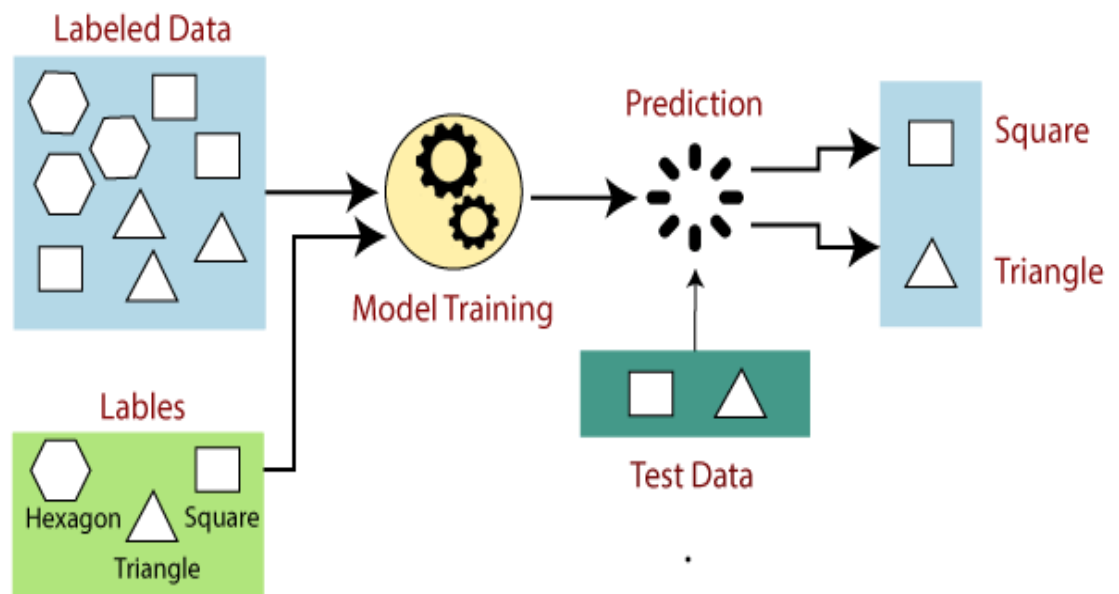
## 5.1 Supervised Learning

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

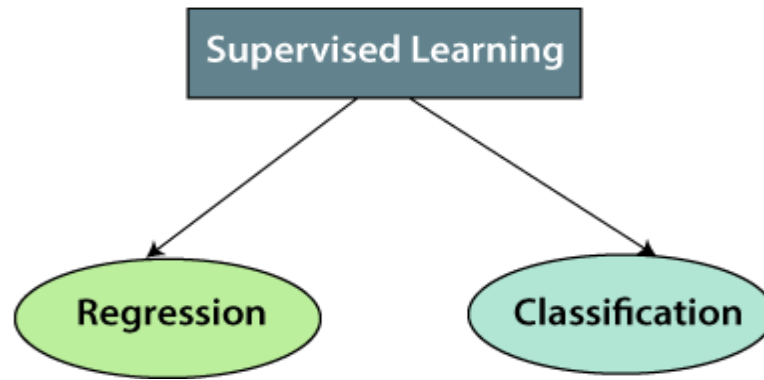
Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable( $x$ ) with the output variable( $y$ ).

In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.



*Figure 5.1 Supervised Learning Model*

Supervised learning can be further divided into two types of problems:



*Figure 5.2 Types of Supervised Learning*

### 5.1.1 Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

### 5.1.2 Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Random Forest
- Decision Trees
- Logistic Regression
- Support Vector Machines



## 5.2 Unsupervised Learning

Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision. Unsupervised learning is a type of machine learning where the algorithm learns to identify patterns or structures in data without explicit supervision or labeled examples. Unlike supervised learning, where the algorithm is trained on labeled data (input-output pairs), unsupervised learning algorithms explore the data on their own to discover hidden patterns or intrinsic structures.

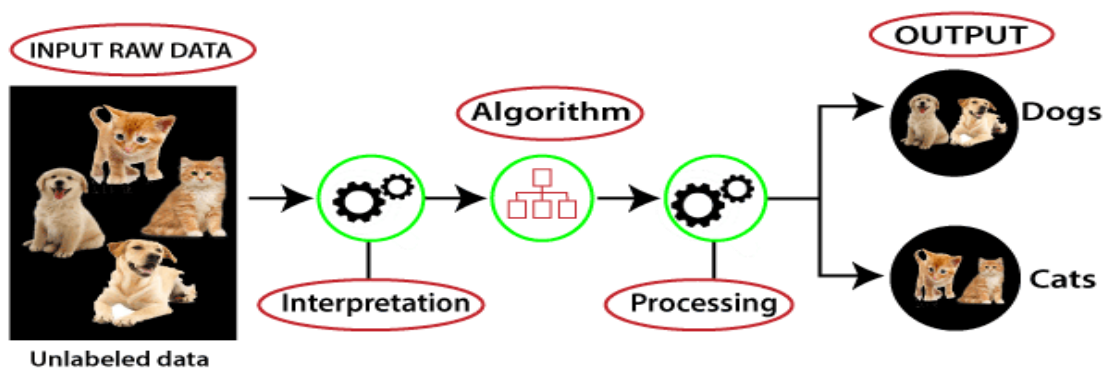


Figure 5.3 Unsupervised Learning Model

The unsupervised learning algorithm can be further categorized into two types of problems:

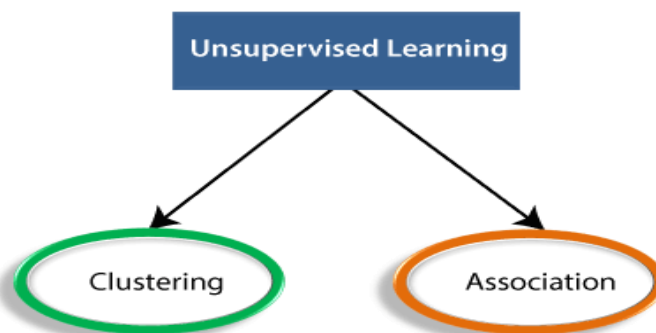


Figure 5.4 Types of Unsupervised Learning

### 5.2.1 Clustering

Clustering is a method of grouping the objects into clusters such that objects with most similarities remain in a group and have less or no similarities with the objects of another group. Cluster analysis

finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

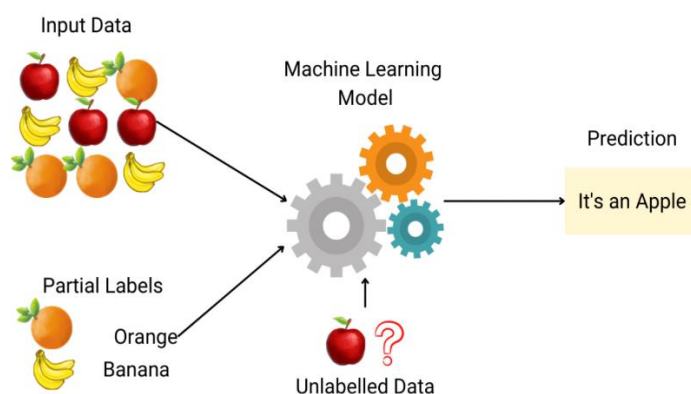
### 5.2.2 Association

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

## 5.3 Semi-supervised Learning

Semi-Supervised learning is a type of Machine Learning algorithm that represents the intermediate ground between Supervised and Unsupervised learning algorithms. It uses the combination of labeled and unlabeled datasets during the training period. To overcome these drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced.

In this algorithm, training data is a combination of both labeled and unlabeled data. However, labeled data exists with a very small amount while it consists of a huge amount of unlabeled data. Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labeled data. It is why label data is a comparatively, more expensive acquisition than unlabeled data.



*Figure 5.5 Semi supervised Learning model*

## 5.4 Reinforcement Learning

Reinforcement Learning (RL) is the science of decision making. It is about learning the optimal behavior in an environment to obtain maximum reward. In RL, the data is accumulated from machine learning systems that use a trial-and-error method. Data is not part of the input that we would find in supervised or unsupervised machine learning.

Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next. After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect. It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance.

Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error. It performs actions with the aim of maximizing rewards, or in other words, it is learning by doing in order to achieve the best outcomes.

There are two types of Reinforcement:

- **Positive:** Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behaviour. In other words, it has a positive effect on behaviour.
- **Negative:** Negative Reinforcement is defined as strengthening of behaviour because a negative condition is stopped or avoided.

Reinforcement learning elements are as follows:

- Policy
- Reward function.
- Value function
- Model of the environment

## 6 Classification Algorithms

This Project is based on Classification Algorithms, and it is a classification model or application. The classification Algorithms are.

- AdaBoost Classifier
- Random Forest Classifier
- Gradient Boosting Classifier
- XgBoost Classifier
- Extra Tree Classifier
- Light Gradient Boosting Classifier
- Decision Tree Classifier
- Support Vector Machine Classifier

Types of Classification algorithms:

- **Linear Models:** Algorithms like Logistic Regression and Linear Discriminant Analysis (LDA) create linear decision boundaries between classes.
- **Tree-Based Models:** Decision Trees, Random Forests, and Gradient Boosting Machines (GBMs) partition the feature space into regions using hierarchical tree structures.
- **Support Vector Machines:** SVM constructs hyperplanes that separate classes in the feature space, maximizing the margin between classes.
- **Neural Networks:** Deep Learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) learn hierarchical representations for classification tasks.
- **Instance-Based Methods:** k-Nearest Neighbours (k-NN) classifies data points based on the majority vote of their nearest neighbours in the feature space.

## 6.1 AdaBoost Classifier

AdaBoost is an excellent machine learning approach for creating highly accurate prediction rules by combining and boosting relatively weak and inaccurate rules. It has a compact mathematical basis and increases the efficiency of multiclass classifier problems in practical applications. AdaBoost takes an iterative approach in order to improve the performance of weak classifiers by allowing them to study and improve from their own mistakes. The ability to reduce noise is improved when the AdaBoost is put into the stopping condition.

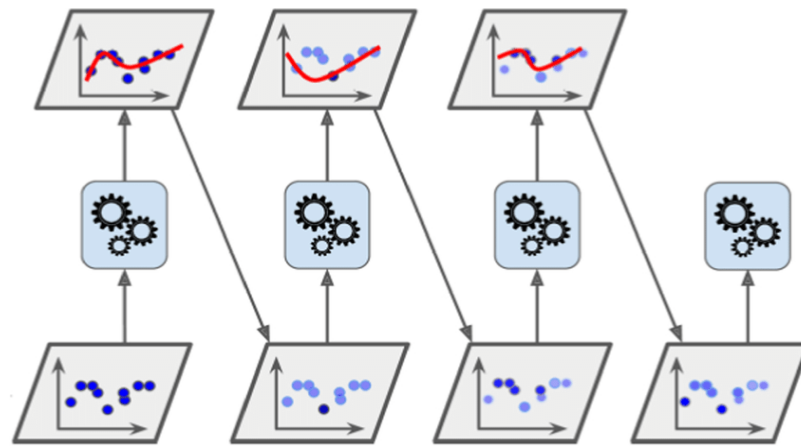


Figure 6.1 Adaboost classifier Model

It has several advantages:

- High Accuracy
- Versatility
- Less Prone to Overfitting
- Automatic Feature Selection
- Handles Class Imbalance
- No prior Knowledge Needed
- Less Sensitive to Noisy Data
- Interpretability
- Works well with Large Datasets

## 6.2 Random Forest Classifier

Random forest is a supervised learning classifier that associates a series of decision tree algorithms with various subsets of the provided datasets. It is capable enough to be used for large-scale problems and simple enough to be customized for various ad hoc learning tasks. To improve the prediction accuracy of the given dataset, it takes the average value from each tree and predicts the final outputs. A probabilistic machine learning technique called NB, which is based on the Bayesian theorem, has been successfully employed for a wide range of tasks, but it excels in solving natural language processing (NLP) issues. It used a simple mathematical formula for calculating conditional probabilities. However, its classification efficiency gradually falls if features are not independent and when the attributes are not independent, and it cannot handle continuous non-parametric characteristics.

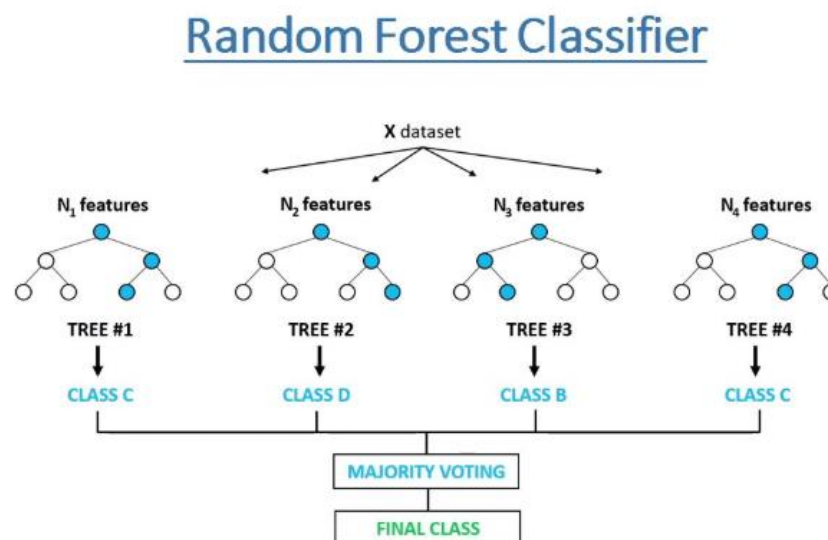


Figure 6.2 Random Forest Classifier Model

It has several advantages:

- High Accuracy
- Reduced Overfitting
- Handles High Dimensional Data
- Parallelizable and Scalable
- Less Sensitivity to Hyperparameters
- Interpretability
- Robustness to Noisy Data

## 6.3 Gradient Boosting Classifier

Gradient boosting (GB) classifiers are a type of machine learning method that brings together numerous weak learning models to develop a powerful predictive model. GB frequently makes use of decision trees. It is predicated on the hypothesis that when merged with earlier models, the best next model will minimize the overall prediction error. Setting the desired results for this subsequent model in order to reduce mistakes is the important concept. The gradient of the error with regard to the prediction is used to determine the goal outcomes for each case. Each model moves closer to making the best predictions feasible for each training example while minimizing prediction error.

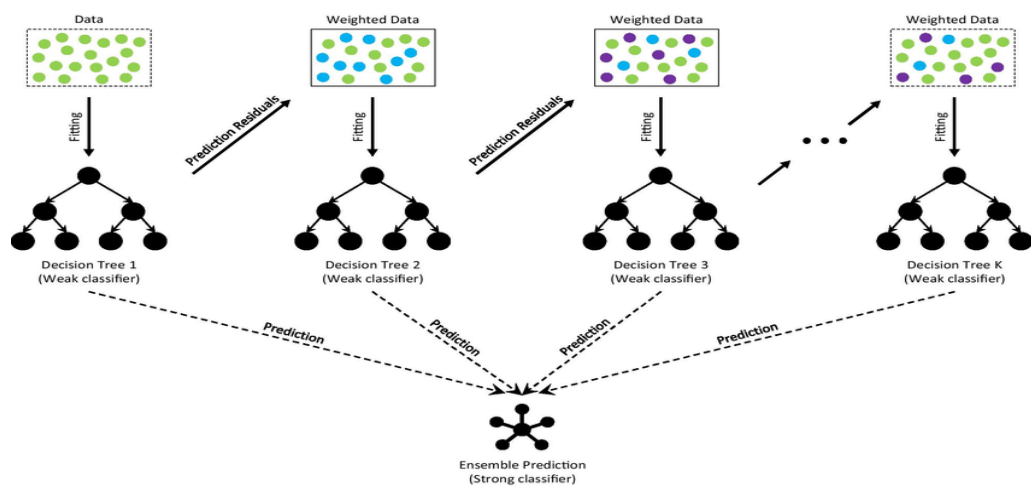


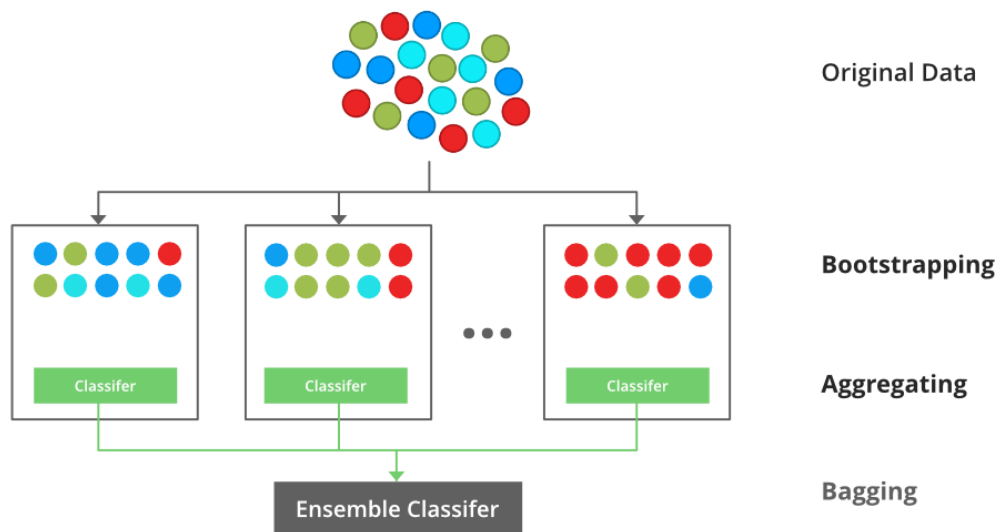
Figure 6.3 Gradient Boosting Model

It has several advantages:

- High accuracy
- Reduced Overfitting
- Handles High Dimensional Data
- Parallelizable and Scalable
- Less Sensitivity to Hyperparameters
- Interpretability
- Robustness to Noisy Data
- Versatility

## 6.4 XgBoost Classifier

XgBoost is commonly known to offer smart solutions to structured data problems through the implementation of the gradient boosted trees technique. Each regression tree in a gradient boosting regression setup act as the weak learner, and it does so by assigning a continuous score to each input data point in the form of a leaf. XgBoost reduces a formalized objective function by merging a convex loss function based on the difference between the observed and target outputs with a weighting parameter for model computational complexity. Adding new trees that forecast the residuals or errors of earlier trees, which are then integrated with earlier trees to produce the final prediction, is how the training process is carried out iteratively.



*Figure 6.4 XgBoost classifier Model*

It has several advantages:

- Highly Efficient
- High Predictive Accuracy
- Regularization techniques
- Tree Pruning
- Supports Custom Loss Functions
- Handles Class Imbalance



## 6.5 Extra Tree Classifier

The Extra Trees method functions by combining the forecasts from various decision tree algorithms. An extra-trees regression employs averaging to increase predictive accuracy and reduce over-fitting. It does this by implementing a meta predictor that fits a number of randomized decision trees on different subsamples of the dataset. The Extra Trees approach is faster and shortens the process overall in terms of computational cost and execution time, but it arbitrarily selects the separation point and does not choose the best one.

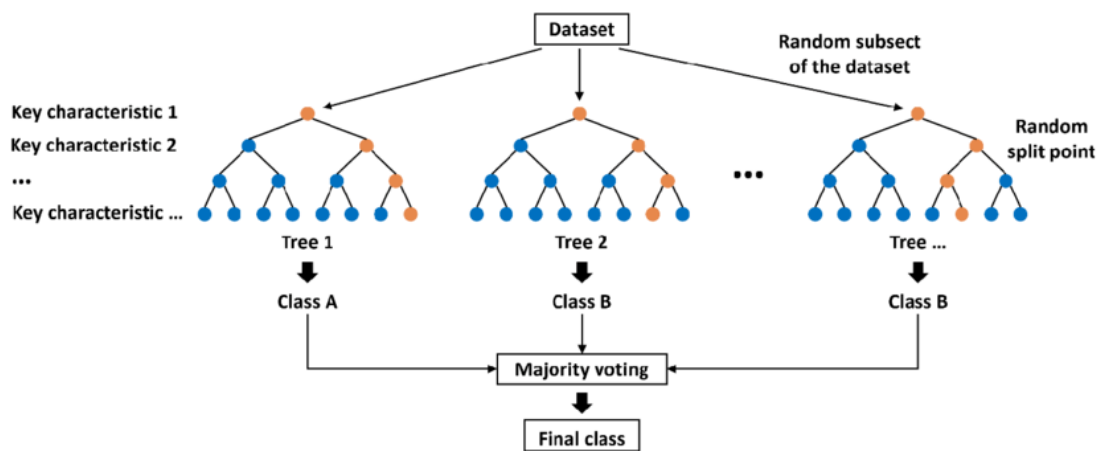


Figure 6.5 Extra Tree Classifiers model

It has several advantages:

- Reduced Variance
- Computationally Efficient
- Less sensitive to noisy data
- Parallelization
- Reduced Bias
- Handles Imbalanced Data
- No Hyperparameter tuning required.

## 6.6 Light Gradient Boosting Classifier

LGBM is a dispersed, strong gradient boosting framework for sorting, classification, as well as data science application development that is based on the decision tree method and it requires less RAM to run while handling massive data sizes. When compared to other algorithms, Light GBM grows trees vertically, or leaf-wise, as opposed to other algorithms, which grow trees horizontally. It will be decided to grow the leaf with the highest delta loss. A leaf-wise method can reduce loss more than a level-wise strategy when expanding the same leaf.

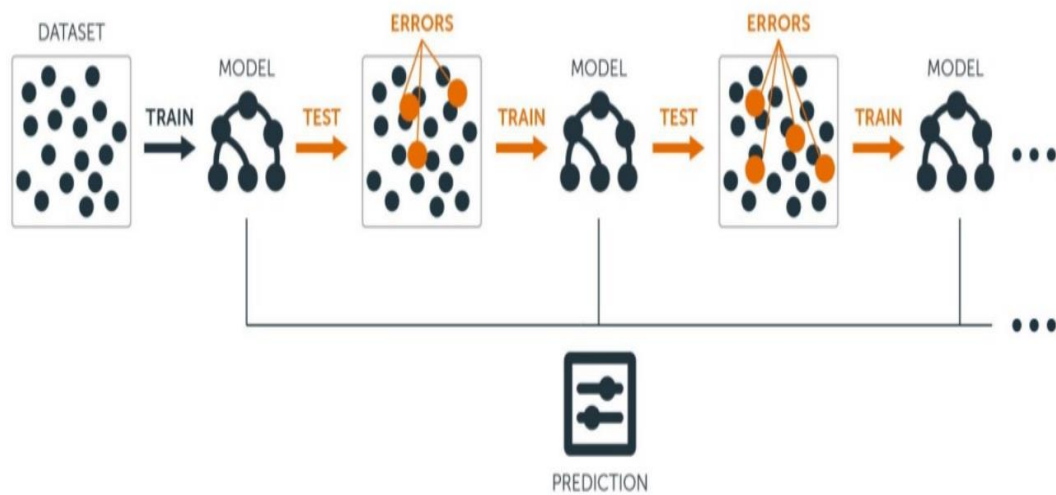


Figure 6.6 Light Gradient Boosting Model

It has several advantages:

- High Efficiency
- Fast Training Speed
- Low Memory Usage
- Handles Large-scale Data
- High Accuracy
- Optimized for Categorical Features
- Tuning Flexibility
- Robustness to Overfitting

## 6.7 Decision Tree Classifier

When it comes to solving categorization issues, one of the most effective and widely used strategies for supervised machine learning is known as the decision tree. A decision tree is a type of tree structure that is similar to a flowchart. In a decision tree, each internal node represents a test that is performed on a feature, each branch represents the outcome of the test, and each leaf node contains a class label.<sup>24</sup> The decision tree starts with the root node of the tree, compares the value of various variables, and then moves to the next branch until it reaches the end leaf node. In classification issues, the decision tree enquires, and based on the answers, it splits the data into subsequent sub branches. It makes use of many techniques to examine the population divide and parameters that allow for the most homogeneous sets.

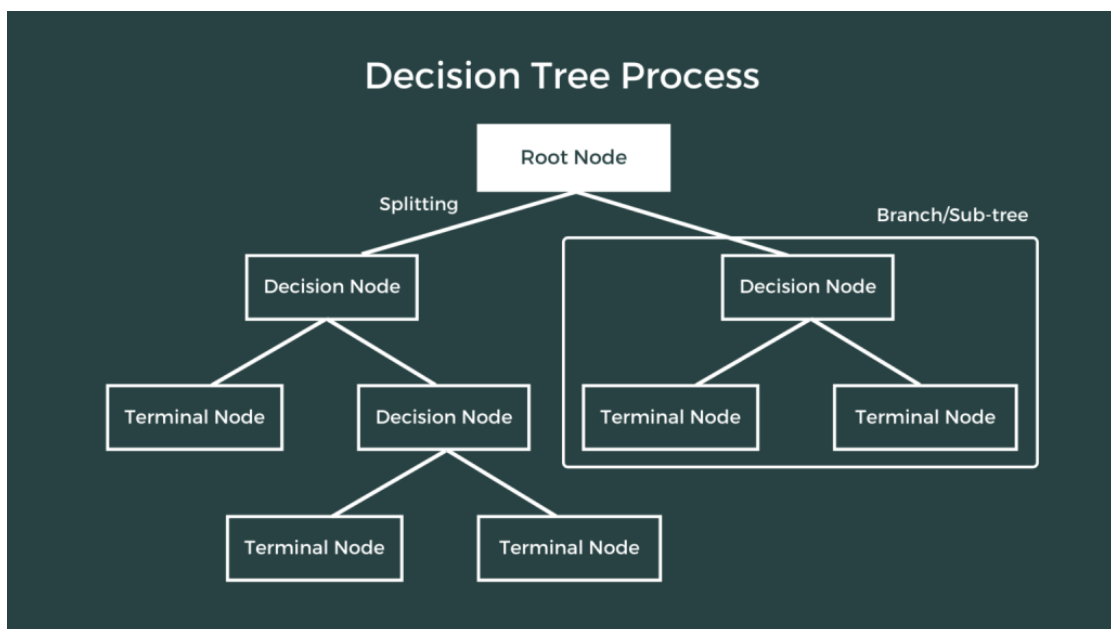


Figure 6.7 Decision Tree classifiers model

It has several advantages:

- Interpretability
- No assumptions about Data Distribution
- Handles Non-linear Relationships
- Robust to outliers and missing values
- Scalability
- Ensemble methods
- Versatility

## 6.8 Support Vector Machine Classifier

Support vector machine (SVM) is a promising classical learning method for classification and regression problems and also solves various linear, non linear, and practical difficulties. The statistical learning theory is the foundation of SVM and it projects targeted data using a kernel function to categorize in a high-dimensional feature space so that data points can be classified even though they are linearly non-separable.

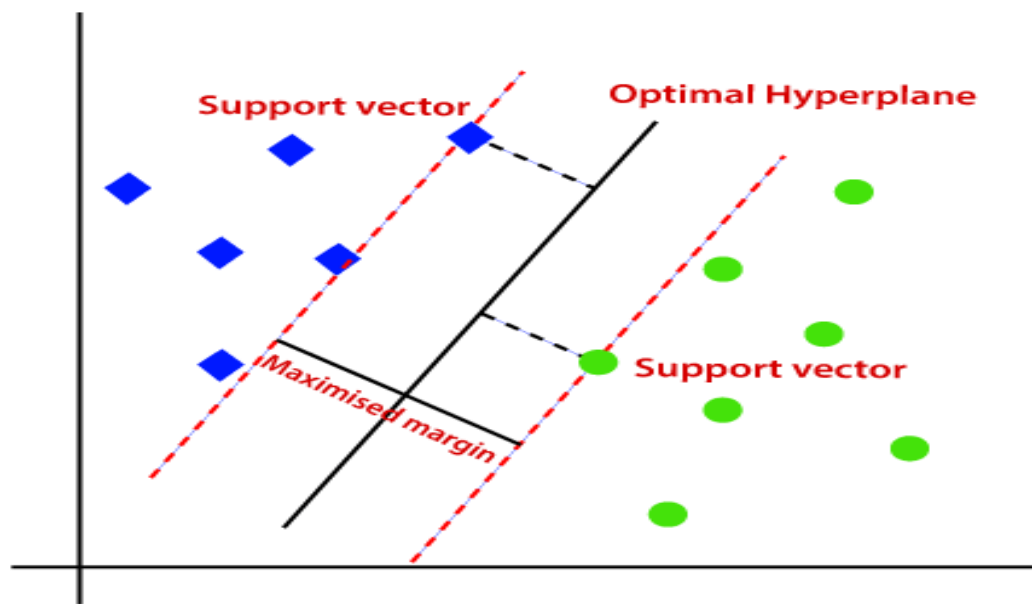


Figure 6.8 SVM Classifier

It has several advantages:

- Effective in High-Dimensional spaces
- Memory Efficient
- Versatile Kernel Functions
- Robust to Overfitting
- Global Optimum Solution
- Effective with Small Sample Sizes
- Controlled Complexity
- Effective in Non-linear Separable cases
- Interpretability

## 7 Software and Hardware Requirements

### 7.1 Software Requirements

- i. Google Colaboratory
- ii. Anaconda3
- iii. Python 3.6.0 or above versions
- iv. Modules like
  - a) Scikit-learn.
  - b) Pandas
  - c) NumPy
  - d) Matplotlib
  - e) Seaborn
  - f) Flask
  - g) Plotly

### 7.2 Hardware Requirements

- i. A minimum of 4 GB RAM
- ii. A hard disk space of 512 GB or more
- iii. Processor: I5 or Ryzen5.

## 8 Implementation Review

### 8.1 Software Implementation

We will look at the following important concepts here which go into building this model:

- NumPy
- Pandas
- Scikit-learn.
- Matplotlib
- Seaborn
- Plotly
- Flask

#### 8.1.1 NumPy

NumPy is a python library that adds support for huge, multi-dimensional arrays and matrices, as well as many high-level mathematical functions to run on these arrays.

Furthermore, NumPy is the backbone of the Machine Learning Stack. The most used NumPy operations are included in this paper.

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

This behaviour is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also, it is optimized to work with latest CPU architectures.

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

## 8.1.2 Pandas

Pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labelled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Furthermore, Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel.

Processing, such as restructuring, cleaning, merging, etc., is necessary for data analysis. NumPy, SciPy, Cython, and Pandas are just a few of the fast data processing tools available. Yet, we incline toward Pandas since working with Pandas is quick, basic and more expressive than different apparatuses.

### Key Features of Pandas:

- It has a DataFrame object that is quick and effective, with both standard and custom indexing.
- Utilized for reshaping and turning of the informational indexes.
- For aggregations and transformations, group by data.
- It is used to align the data and integrate the data that is missing.
- Provide Time Series functionality.
- Process a variety of data sets in various formats, such as matrix data, heterogeneous tabular data, and time series.
- Manage the data sets' multiple operations, including subsetting, slicing, filtering, group By, reordering, and reshaping.
- It incorporates with different libraries like SciPy, and scikit-learn.
- Performs quickly, and the Cython can be used to accelerate it even further.

The following are the advantages of pandas overusing other languages:

- **Representation of Data:** Through its DataFrame and Series, it presents the data in a manner that is appropriate for data analysis.
- **Clear code:** Pandas' clear API lets you concentrate on the most important part of the code. In this way, it gives clear and brief code to the client.

DataFrame and Series are the two data structures that Pandas provides for processing data.

### 8.1.3 Scikit-learn

**Scikit-learn** (formerly **scikits.learn** and also known as **sklearn**) is a free software machine learning library for the Python programming language.

It features various classification, regression and clustering algorithms which including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. It's built upon some of the technology you might already be familiar with, like NumPy, pandas, and Matplotlib.

An open-source Python package to implement machine learning models in Python is called Scikit-learn. This library supports modern algorithms like KNN, random forest, XgBoost, and SVC. It is constructed over NumPy. Both well-known software companies and the Kaggle competition frequently employ Scikit-learn. It aids in various processes of model building, like model selection, regression, classification, clustering, and dimensionality reduction (parameter selection).

Scikit-learn is simple to work with and delivers successful performance. Scikit Learn, though, does not enable parallel processing. We can implement deep learning algorithms in sklearn, though it is not a wise choice, especially if using TensorFlow is an available option.

We can divide the complete dataset into two parts-a training dataset and a testing dataset-to spare some unseen data to check the model's accuracy. Use the testing dataset to test or validate the model once it has been trained using the training set. Then, we can assess how well the trained model performed.

This example will divide the data into a 70:30 ratio, meaning that 70% of the data will be used for training the model, and 30% will be used for testing the model.

The functionality that scikit-learn provides include:

- **Regression**, including Linear and Logistic Regression
- **Classification**, including K-Nearest Neighbours
- **Clustering**, including K-Means and K-Means++



### 8.1.4 Matplotlib

Matplotlib is a library for plotting data. pyplot is a collection of command-style functions that enable matplotlib to behave similarly to MATLAB. Each pyplot function modifies a diagram in some way, such as creating a figure, a plotting area in a figure, plotting some lines in a plotting area, decorating the plot with labels, and so on.

Various states are stored through function calls in matplotlib. Pyplot, allowing it to keep track of details like the current figure and plotting field, as well as directing plotting functions to the current axes.

Matplotlib is a comprehensive library for creating static, interactive, and animated visualizations in Python. It is widely used for generating plots, charts, histograms, and other types of graphical representations of data. Matplotlib provides a high-level interface for creating publication-quality visualizations with ease.

#### **Key Features of Matplotlib:**

- Wide Range of Plot Types
- High-Quality Output
- Interactive and GUI-based Plotting
- Seamless Integration with NumPy
- Multiple Backends
- Customizable Layouts
- Integration with pandas
- Support for LaTeX text Rendering
- Extensibility and Customization
- Rich Documentation and Community Support

### 8.1.5 Plotly

**Plotly** library in Python is an open-source library that can be used for data visualization and understanding data simply and easily. Plotly supports various types of plots like line charts, scatter plots,

histograms, box plots, etc. So you all must be wondering why Plotly is over other visualization tools or libraries. So here are some reasons :

- Plotly has hover tool capabilities that allow us to detect any outliers or anomalies in a large number of data points.
- It is visually attractive and can be accepted by a wide range of audiences.
- Plotly generally allows us endless customization of our graphs and makes our plot more meaningful and understandable for others.

Plotly is a comprehensive data visualization library that enables users to create interactive and high-quality visualizations in Python, R, and JavaScript. It offers a wide range of chart types, including line plots, scatter plots, bar charts, pie charts, 3D plots, geographic maps, and more. Plotly is widely used in various domains, including data analysis, scientific research, finance, and business intelligence.

### 8.1.6 Flask

Flask is a lightweight and flexible web framework for Python. It's designed to make getting started quick and easy, with the ability to scale up to complex applications. Flask provides tools, libraries, and best practices that allow developers to build web applications quickly and efficiently.

Flask offers comprehensive resources to help developers learn and use the framework effectively.

#### Key Features of Flask:

**Minimalistic Design:** Flask follows a minimalist design philosophy, providing only the essential components needed for web development. This simplicity makes Flask lightweight and easy to understand, ideal for beginners and experienced developers alike.

**Routing:** Flask uses a decorator-based routing system to map URLs to Python functions (view functions). Developers can define routes using the `@app.route` decorator, specifying the URL pattern and associated view function to handle incoming requests.

**Template Engine:** Flask includes a built-in template engine called Jinja2, which enables developers to create dynamic HTML pages by embedding Python code within HTML templates. Jinja2 templates support inheritance, macros, filters, and other features for building reusable and maintainable web templates.

**HTTP Request Handling:** Flask provides built-in support for handling HTTP requests and responses. Developers can access request data (e.g., form data, query parameters, headers) using the `request` object and return HTTP responses using the `Response` class or convenience functions like `render_template` and `redirect`.

**URL Building:** Flask includes a URL building mechanism that generates URLs for view functions based on their names and arguments. This feature simplifies URL management and ensures consistency when linking to different parts of the application.

**Extension Ecosystem:** Flask has a rich ecosystem of extensions that provide additional functionality and integrations with third-party services. Extensions cover a wide range of features, including authentication, database integration, form validation, caching, and more.

**Development Server:** Flask includes a built-in development server for testing and debugging web applications locally. The development server automatically reloads the application when code changes are detected, making the development process faster and more efficient.

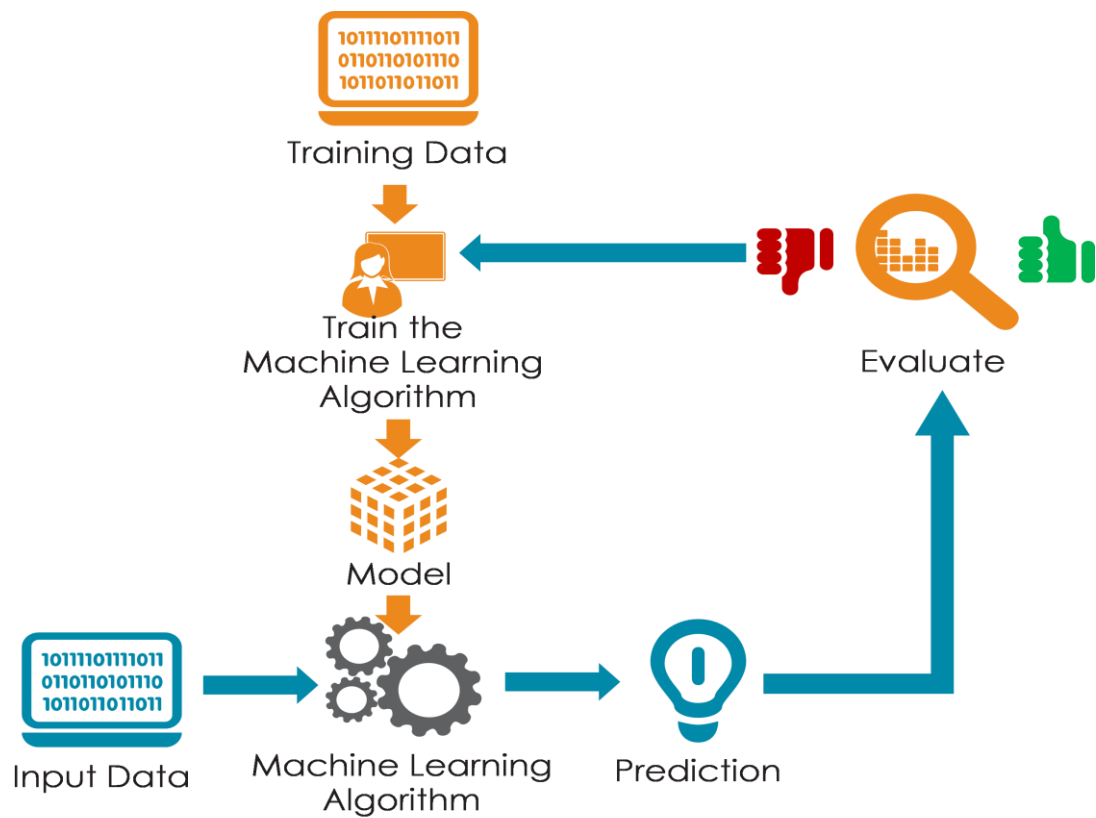
**WSGI Compliance:** Flask is WSGI-compliant, meaning it adheres to the Web Server Gateway Interface (WSGI) standard for Python web applications. This compatibility allows Flask applications to run on any WSGI-compliant web server, including popular options like Gunicorn, uWSGI, and Apache with `mod_wsgi`.

**Testing Support:** Flask provides utilities and conventions for writing automated tests for web applications. Developers can use testing frameworks like Flask-Testing and Werkzeug's test client to simulate HTTP requests, test route handlers, and verify application behavior.

**Community and Documentation:** Flask has a vibrant community of developers who contribute plugins, tutorials, and resources to support Flask development. The official Flask documentation is

comprehensive and well-maintained, offering guidance on getting started, advanced features, and best practices.

## 8.2 Machine Learning Model Implementation



*Figure 8.1 Machine Learning Model*

You will follow the general machine learning workflow.

1. Examine and understand the data
2. Build an input pipeline
3. Compose the model
4. Load in the pretrained base model (and pretrained weights)
5. Train the model
6. Evaluate model

## 8.3 Dataset collection and pre-processing

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px


import warnings

warnings.filterwarnings('ignore')

# loading data

df= pd.read_csv('/content/kidney_disease.csv')

df.head()

# checking for null values

df.isna().sum().sort_values(ascending = False)
```

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	no	no	good	no	no	ckd

5 rows × 26 columns

## 8.4 Data Splitting

```
# filling null values, we will use two methods, random sampling for higher null values and

# mean/mode sampling for lower null values

def random_value_imputation(feature):

    random_sample = df[feature].dropna().sample(df[feature].isna().sum())

    random_sample.index = df[df[feature].isnull()].index

    df.loc[df[feature].isnull(), feature] = random_sample

print(df['age'].mode())

print(df['age'].mode()[0])

def impute_mode(feature):

    mode = df[feature].mode()[0]

    df[feature] = df[feature].fillna(mode)

# filling "red_blood_cells" and "pus_cell" using random sampling method and rest of
cat_cols using mode imputation

random_value_imputation('red_blood_cells')

random_value_imputation('pus_cell')

for col in cat_cols:

    impute_mode(col)
```

## 8.5 Building the model

```
ind_col = [col for col in df.columns if col != 'class']

dep_col = 'class'

X = df[ind_col]

y = df[dep_col]

# splitting data into training and test set

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 0)

#light gradient Boosting

from lightgbm import LGBMClassifier

lgbm = LGBMClassifier(learning_rate = 1)

lgbm.fit(X_train, y_train)

# accuracy score, confusion matrix and classification report of lgbm classifier

lgbm_acc = accuracy_score(y_test, lgbm.predict(X_test))

print(f"Training Accuracy of LGBM Classifier is {accuracy_score(y_train, lgbm.predict(X_train))}")

print(f"Test Accuracy of LGBM Classifier is {lgbm_acc} \n")

print(f"{confusion_matrix(y_test, lgbm.predict(X_test))}\n")

print(classification_report(y_test, lgbm.predict(X_test)))
```

## 8.6 Training the model

```
# splitting data into training and test set
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 0)
```



## 9 Bibliography

- [1] K. I. Supplements, “Kidney Disease: Improving Global Outcomes (KDIGO).,” National Kidney Foundation.
- [2] A. S. B. C. I. L. A. e. a. Levey, “Glomerular filtration rate and albuminuria for detection and staging of acute and chronic kidney disease in adults: a systematic review.,” *JAMA - Journal of the American Medical Association*, 2015.
- [3] C. Zoccali, “The systemic nature of CKD”.
- [4] N. R. F. S. T. O. J. L. e. a. Hill, “Global prevalence of chronic kidney disease - a systematic review and meta-analysis.,” 2016.
- [5] S. J. Gilbert, “Nephrology.,” Oxford Press, 2014.